SIEMENS

8-Bit Microcontroller Handbook
1994

8-Bit Single-Chip Microcontroller Handbook 1994

Published by Siemens Components Inc., Integrated Circuit Division
10950 North Tantau Avenue Cupertino, CA 95014-0716

For questions on technology, delivery, and prices please contact the Offices of Siemens Components.

8-Bit Single-Chip Microcontrollers

SAB 8031A
SAB 8032B
SAB 8051A
SAB 8052B
SAB 80515
SAB 80C515
SAB 80C515A
SAB 80C517
SAB 80C517A
SAB 80535
SAB 80C535
SAB 80C537
SAB 83C515A-5
SAB 83C517A-5

8-Bit Single-Chip Microcontrollers Extended Temperature Range

SAB 80515
SAB 80C515
SAB 80C515A
SAB 80C517
SAB 80C517A
SAB 80535
SAB 80C535
SAB 80C537
SAB 83C515A
SAB 83C517A

General Information

Type Designation Code of ICs

IC type designations are based on the European Pro Electron System. The code system is summarized on page 3-3 and explained in the Pro Electron brochure D 15, edition 1985, available at:

Pro Electron, Avenue Louise, 430 (B.12)
B-1060 Brussels, Belgium

Mounting Instructions

Plastic Package (Dual In-Line)

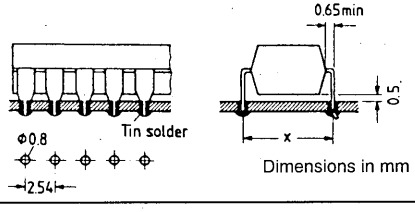
The 90° pins fit into holes with a diameter of 0.7mm to 0.9mm, spaced 2.54mm apart. See spacing x in Figure 1.

The bottom of the package will not touch the PC board after insertion because the pins have shoulders just below the package (see Figure 1).

After insertion of the package into the PC board it is advisable to bend the ends of two pins at an angle of approx. 30° to the board so that the package does not have to be pressed down during soldering. Plastic packages are soldered on that side of the PCB facing away from the package.

The maximum permissible soldering temperature is 350°C (max. 3s) for hand soldering and 260°C (max. 10s) for dip soldering and wave soldering.

Figure 1
Plastic Package (Dual-In-Line)



Plastic Packages (SO and PLCC) for Surface Mounting (SMD)

Iron Soldering:
Soldering temperature 350°C for max. 3s; minimum distance between package and soldering point 1.5mm package temperature max. 150°C; no mechanical stress on the pins

Vapor phase soldering:
Soldering temperature 215°C, max. soldering time 40s

Wave soldering: (pins and package are dipped into the tin bath). Soldering temperature 260°C, max. soldering time 8s.

2

Storage, Pretreatment before Processing

The components are to be stored in a dry environment. When solder methods causing solder heat shock stresses are used (reflow soldering where the component is dipped into the solder bath, vapor-phase soldering) it is recommendable to subject IC's in plastic packages to a 24-hour drying phase at 125°C.

Other Points to Note

Ensure that no current is able to flow between the solder bath or soldering iron and the PCB. It is advisable to ground the pins that are to be soldered as well as the solder bath or soldering iron.

When the pins are being prepared and inserted in a PCB, circuits should be protected against static charge. Under no circumstances should the components be removed or inserted while the operating voltage is switched on.

The increase in chip temperature during the soldering process results in a temporary increase in electrostatic sensitivity of integrated circuits. Special precautions should therefore be taken against line transients, e.g. through the switching of

inductances on magnetic chutes, etc.

Processing Guidelines for ICs

Integrated circuits (ICs) are electrostatic-sensitive (ESS) devices. The requirement for greater packing density has led to increasingly small structures on semiconductor chips with the result that today every IC, whether bipolar, MOS, or CMOS, has to be protected against electrostatics.

MOS and CMOS devices generally have integrated protective circuits and it is hardly possible any more for them to be destroyed by purely static electricity. On the other hand, there is acute danger from electrostatic discharge (ESD).

Of the multiple of possible sources of discharge, charged devices should be mentioned in addition to charged persons. With low-resistive discharges it is possible for peak power amounting to kilowatts to be produced.

For the protection of devices the following principles should be observed:

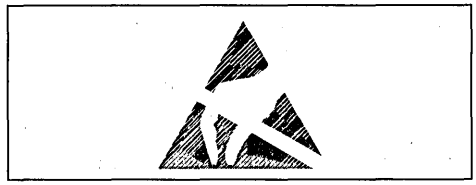
a. Reduction of charging voltage, below 200V if possible. Means which are effective here are an increase in relative humidity to ≥ 60% and the replacement of highly charging plastics by antistatic materials.

b. With every kind of contact with the device pins a charge equalization is to be expected. This should always be highly resistive (ideally $R = 10^6$ OHM to 10^8 Ω).

All in all this means that ICs call for special handling, because uncontrolled charges, voltages from ungrounded equipment or persons, surge voltage spikes and similar influences can destroy a device. Even if devices have protective circuits (e.g. protective diodes) on their inputs, the following guidelines for their handling should nevertheless be observed.

Identification

The packing of ESS devices is provided with the following label by the manufacturer:



Scope

The guidelines apply to the storage, transport, testing, and processing of all kinds of ICs, as well as the soldered circuit boards equipped with such components.

Handling of Devices

1. ICs must be left in their containers until they are processed.
2. ICs may only be handled at specially equipped work stations. These stations must have work surfaces covered with a conductive material of the order of 10^6 OHM/cm. to 10^9 Ω/cm.
3. With humidity of >50% a coat of pure cotton is sufficient. In the case of chargeable synthetic fibers the clothing should be worn close-fitting. The wrist strap must be worn snugly on the skin and be grounded through a resistor of 50 k Ω to 100 k Ω.
4. If conductive floors, $R = 5 \times 10^4$ Ω to 10^7 Ω are provided, further protection can be achieved by using so-called MOS chairs and shoes with a conductive sole ($R \approx 10^5$ Ω to 10^7 Ω).
5. All transport containers for ESS devices and assembled circuit boards must first be brought to the same potential by being placed on the work surface or touched by the operator before the individual devices may be handled. The potential equalization should be through a resistor of 10^6 Ω to 10^8 Ω.
6. When loading machines and production devices it should be noted that the devices come out of the transport magazine

charged and can be damaged if they touch metal, e.g. machine parts.

Example 1. Conductive (black) tubes.

The devices may be destroyed in the tube by charged persons or come out of the tube charged if this is emptied by a charged person. Conductive tubes may only be handled at ESS work stations (high-resistance work-station and person grounding).

Example 2. Anti-static (transparent) tubes.

The devices cannot be destroyed by charged persons in the tube (there may be a rare exception in the case of custom ICs with unprotected gate pins). The devices can be endangered as in 1) when the tube is emptied if the latter, especially at low humidity, is no longer sufficiently anti-static after a long period of storage (> 1 year).

In both cases damage can be avoided by discharging the devices through a grounded adapter of high-resistance material ($\approx 10^6$ Ω/cm to 10^8 Ω/cm) between the tube and the machine.

The use of metal tubes—especially of anodized aluminum—is not advisable because of the danger of low-resistance device discharge.

Storage

ESS devices should only be stored in identified locations provided for the purpose. During storage the devices should remain in the package in which they are supplied. The storage temperature should not exceed 60°C.

Transport

ESS devices in approved packing tubes should only be transported in suitable containers of conductive or long-term anti-static-treated plastic or possibly unvarnished wood. Containers of high-charging plastic or very low-resistance materials are likewise unsuitable.

Transfer cars and their rollers should exhibit adequate electrical conductivity ($R < 10^6$ Ω). Sliding contacts and grounding chains will not reliably eliminate charges.

Incoming Inspection

In incoming inspection the above guidelines should be observed. Otherwise any right for refund or replacement if devices fail inspection may be lost.

Material and Mounting

1. The drive belts of machines used for the processing of the devices, in as much as they come into contact with them (e.g. bending and cutting machines, conveyor belts), should be treated with anti-static spray (e.g. anti-static spray 100 from Kontaktchemie). It is better, however, to avoid the contact completely.
2. If ESS devices have to be soldered or de-soldered manually, soldering irons with thyristor control cannot be used. Siemens EMI-suppression capacitors of the type B 81711-B31—B36 have proven very effective against line transients.
3. Circuit boards fitted and soldered with ESS devices are always to be considered as endangered.

Electrical Tests

1. The devices should be processed with observation of these guidelines. Before assembled and soldered circuit boards are tested, remove any shorting ring.
2. Test sockets must not be conducting any voltage when individual devices or assembled circuit boards are inserted or withdrawn, unless works' specifications

state otherwise. Ensure that the test devices do not produce any voltage spikes, either when being turned on and off in normal operation or if the power fuse blows or other fuses respond.

3. Signal voltages may only be applied to the inputs of ICs when or after the supply voltage is turned on. They must be disconnected before or when the supply voltage is turned off.
4. Observe any notes and instructions in the respective data books/sheets.

Packing of Assembled PC Boards or Flatpack Units

The packing material should exhibit low volume conductivity:

$10^5\ \Omega/cm < p < 10^{10}\ \Omega/cm.$

In most cases—especially with humidity of > 40%—this requirement is fulfilled using simple corrugated board. Better protection is obtained with bags of conductive polyethylene foam (e.g. RCAS 1200 from Richmond of Redlands, California).

It must always be ensured that boards do not touch.

In special cases it may be necessary to provide protection against strong electric fields, such as can be generated by conveyor belts for example. For this purpose a sheath of aluminum foil is recommended, although direct contact between the film and the PCB must be avoided. Cardboard boxes with an aluminum-foil lining, such as those used for shipping of our devices, are available from Laber of Munich.

Ultrasonic Cleaning of ICs

The following recommendation applies to plastic packages. For cavity packages (metal and also ceramic) separate regulations have to be observed.

Freon and isopropyl alcohol (trade name: propanol) can be used as solvents. These solvents can also be used for plastic packages because they do not eat into the plastic material.

An ultrasonic bath in double halfwave operation is advisable because of the low component stress.

The ultrasonic limits are as follows:

sound frequency	$f > 40$ kHz
exposure	$t < 2$ min
alternating sound pressure	$p < 0.29$ bar
sound power	$N < 0.5$ W/cm^2/liter

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics will apply at $T_A = 25°C$ and for the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description will be fulfilled.

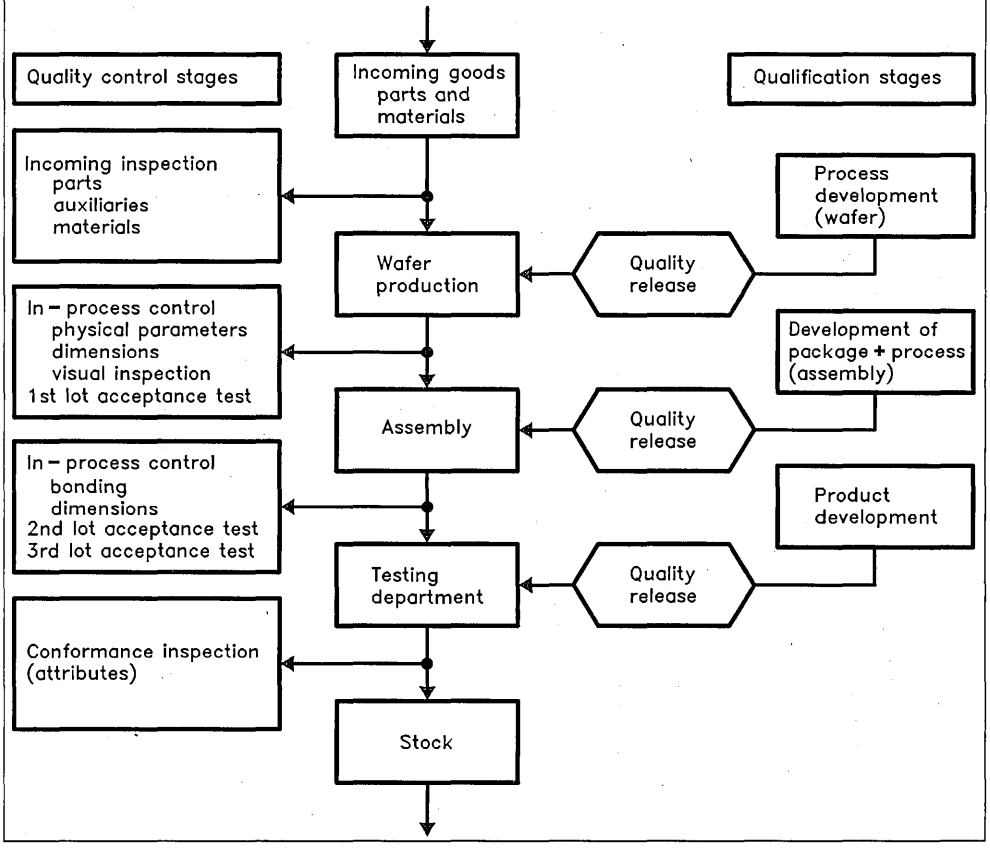
Quality Assurance System

The high quality and reliability of integrated circuits from Siemens is the result of a carefully arranged production which is systematically checked and controlled at each production stage.

The procedures are subject to a quality assurance system; full details are given in the brochure "Siemens Quality Assurance—Integrated Circuits' (SQS-IC).

Figure 2 shows the most important stages of the "SQS-IC". A quality assurance (QA) department which is independent of production and development, is responsible for the selected control measures, acceptance procedures, and information feedback loops. This department has state-of-the-art test and measuring equipment at its disposal, works according to approved methods of statistical quality control, and is provided with facilities for accelerated life and environmental tests used for both qualification and routine monitoring test.

Figure 2
Quality Assurance System



The latest methods and equipment for preparation and analysis are employed to achieve continuity of quality and reliability.

Conformance

Each integrated circuit is subjected to a final test at the end of the production process. These tests are carried out by computer-controlled, automatic test systems because hundreds of thousands of operating conditions as well as a large number of static and dynamic parameters have to be considered. Moreover, the test systems are extremely reliable and reproducible. The quality assurance department carries out a final check in the form of a lot-by-lot sampling inspection to additionally ensure this minimum percent defectives as well as the acceptable quality level (AQL). Sampling inspection is performed in accordance with the inspection plans of DIN 40080, as well as of the identical MIL-STD-105 or IEC 410.

Reliability

Measures Taken during Development

The reliability of ICs is already considerably influenced at the development stage. Siemens has, therefore, fixed certain design standards for the development of circuit and layout, specifying e.g. minimum width and spacing of conductive layers on a chip, dimensions and electrical parameters of protective circuits for electrostatic charge, etc. An examination with the aid of carefully arranged programs operated on large-scale computers, guarantees the immediate identification and elimination of unintentional violations of these design standards.

In-Process Control during Production

The manufacturing of integrated circuits comprises several hundred production steps. As each step is to be executed with utmost accuracy, the in-process control is of outstanding importance. Some processes require more than a hundred different test measures. The tests have been arranged such that the individual process steps can be reproduced continuously.

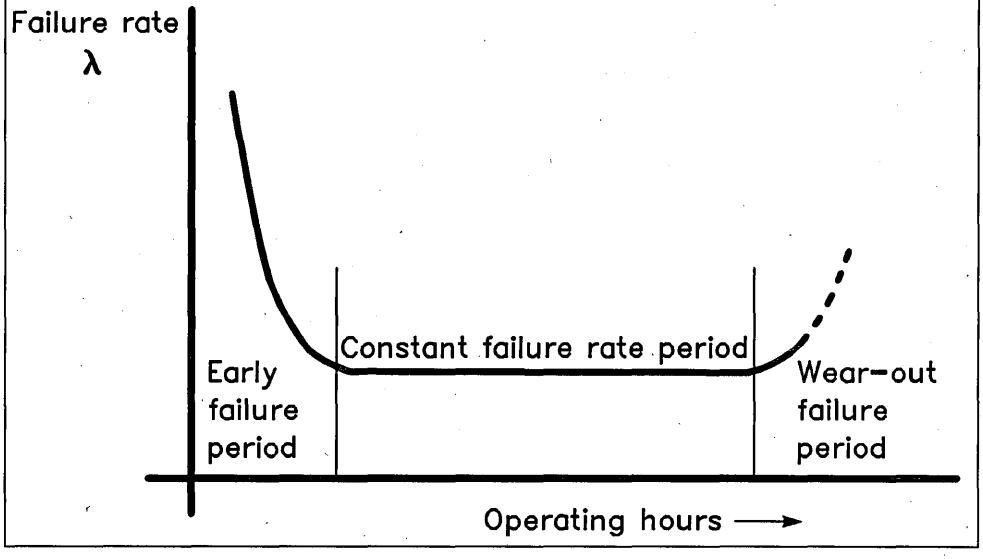
The decreasing failure rates reflect the never ending effort in this direction; they have been reduced considerably despite an immense rise in the IC's complexity.

Reliability Monitoring

The general course of the IC's failure rate versus time is shown by a so-called "bathtub" curve (Figure 3). The failure rate has its peak during the first few operating hours (early failure period). After the early failure period has decayed, the "constant" failure rate period starts during which the failures may occur at an approximately uniform rate. This period ends with a repeated rise of the curve during the wear-out failure period. For ICs, however, the latter period usually lies far beyond the service life specified for the individual equipment.

Reliability tests for ICs are usually destructive examinations. They are, therefore, carried out with samples. Most failure mechanisms can be accelerated by means of higher temperatures. Due to the temperature dependence of the failure mechanisms, it is possible to simulate future operational behavior within a short time by applying high temperatures; this is called accelerated life testing.

Figure 3
Reliability "Bath-Tub" Curve



The acceleration factor B for the life test can be obtained from the Arrhenius equation

$$B = \exp \frac{E_A}{k}\left[\frac{1}{T_1} - \frac{1}{T_2}\right]$$

where T_2 is the temperature at which the life test is performed, T_1 is the assumed operating temperature, and k is the Boltzmann constant.

Important for factor B is the activation energy E_A. It lies between 0.3V and 1.3 eV and differs considerably for individual failure mechanisms.

For all Siemens ICs, the reliability data from life tests is converted to an operating temperature of T_A = 55°C, assuming an average activation energy of 0.5 eV. The acceleration factor for life tests at 125°C is thus 22.3, compared with operational behavior. This method considers also failure mechanisms with low activation energy, i.e. which are only slightly accelerated by the temperature effect.

Various reliability tests are periodically performed with IC types that are representative of a certain production line—this is described in the brochure "SQS-IC". Such tests are e.g. humidity test at 85°C and 85% relative humidity, pressure cooker test, as well as life tests up to 1000 hours and more. Test results are available in the form of summary reports.

SAB-51 Architectural Overview

Siemens SAB-51 family of 8-bit microcontrollers consists of the devices listed in Table 1, all of which are based on the SAB-8051 architecture shown in Figure 1. The original 8051 was built in N-channel, silicon gate, Siemens MYMOS II technology and packaged in a 40-pin DIP.

The 8051A which is in the advanced N-channel, silicon gate Siemens MYMOS III process, is the device currently in production.

All other microcontrollers listed in this book are backward compatible with the SAB-8051A.

Figure 1
SAB 8051 Architectural Block Diagram

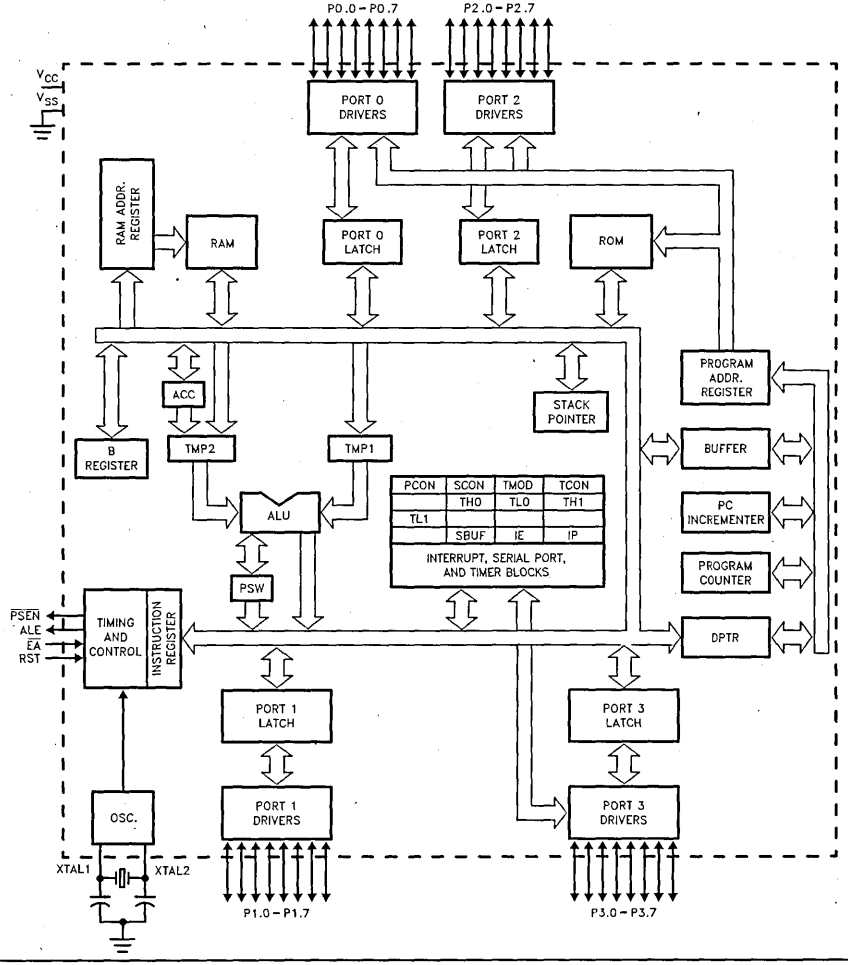


Table 1
The SAB-51 Family of Microcontrollers

Device *=ROMless version †=Hybrid EEPROM Version	Clock Rate (MHz)	ROM (KB)	RAM (Byte)	I/O-Ports (8-Bit)	ADC Inputs/ Resol.	Timer/ Counter (16-Bit)	Watch Dog Timers	Inter-rupt Sources/ Levels	Serial I/O	High Speed Outputs	Div/ Mult Unit	Data Pointers (16-Bit)	Package	Process
SAB 8051 SAB 8031*	12, 16, 20	4 –	128	4	–	2	–	5/2	USART	–	–	1	PDIP 40 PLCC 44	NMOS
SAB 8052 SAB 8032*	12, 16, 20	8 –	256	4	–	3	–	6/2	USART	–	–	1	PDIP 40 PLCC 44	NMOS
SAB C501-1R SAB 501-L*	12, 20, 40	8 –	256	4	–	3	–	6/2	USART	–	–	1	PDIP 40 PLCC 44	CMOS
SAB C502-2R† SAB 502-L*	12, 18	16 –	256 + 256	4	–	3	2	6/2	USART	–	–	8	PDIP 40 PLCC 44	CMOS
SAB C503-1R† SAB 503-L*	12, 18	8 –	256	4	8/10 Bit	3	2	6/2	USART	–	–	1	PLCC 44	CMOS
SAB 80515 SAB 80535*	12	8 –	256	6	8/8 Bit	3	1	12/4	USART	4-ch	–	1	PLCC 68	NMOS
SAB 80C515† SAB 80C535*	12, 16	8 –	256	6(I/O) + 1(I)	8/8 Bit	3	1	12/4	USART	4-ch	–	1	PLCC 68	CMOS
SAB 83C515A-5† SAB 80C515A*	18	32 –	256 + 1K	6(I/O) + 1(I)	8/10 Bit	3	2	12/4	USART	4-ch	–	1	PLCC 68	CMOS
SAB 80C517† SAB 80C537*	12, 16	8 –	256	7 (I/O) + 1 1/2(I)	12/8 Bit	4	2	14/4	USART + UART	21-ch	Yes	8	PLCC 84 PQFP-100	CMOS
SAB 80C517A-5† SAB 80C517A*	18	32 –	256 + 2K	7 (I/O) + 1 1/2(I)	12/10 Bit	4	2	17/4	USART + UART	21-ch	Yes	8	PLCC 84 PQFP-100	CMOS

*=ROMless version

†=Also available in E²PROM Hybrid version

Standardized Micro-Controller Part Numbering System

To conform to Pro Electron Standards the following part numbering system has been adopted for all microcontrollers introduced after 1991.

Features:

- SAx Prefix: this prefix is now incorporated (with x = B, F, H or K for temperature range specifications). The purpose is to conform to Pro Electron Standards.
- Intel Prefix: the secondary Intel prefix "80/83/87/.." will no longer be used.
- Speed/Package Type: Clock Frequency (in MHz) and standard Siemens Package Type indicators will always be a part of the final suffix to conform with other HL IS part numbering schemes.
- CMOS: all new devices are planned to be CMOS. We will incorporate the "C" designator in the Part Number Core to promote this technology.
- ROM Type: the ROM type and size is clearly specified in the Part Number.
- Part Number Length: including spaces and dashes, the maximum length of a part number is 16 characters.

Standardized Part Numbering Scheme

A typical part number will have seven fields and a maximum of 16 characters (as shown below).

Prefix via Pro Electron (T)	Basic Core (U)	Option Field (V)	—	ROM Size (W)	ROM Type (X)	Speed (Y)	Package Type (Z)

Example:

SAF C517 AB - 6R18N

Size		Variable Index
3	T	Pro Electron Temperature Range
1		Space (always appears)
2-4	U	Basic Core Type (with CMOS indicator)
0-4	V	Functionality Option Field (optional)
1		Dash (always appears)
0-1	W	ROM Size
1	X	ROM Type
2	Y	Clock Frequency (MHz)
1	Z	Package Type

11 to a maximum of 16 characters

Meaning of Indices:

Index T	Index U	Index V**
Prefix + Temp. Range	Basic Core Type	Optional Field
SAB = 0C to +70C	51, 52, 515, 517,	examples: A,
SAF = -40C to +85C	C515, C517,	ABS, ATT, C2,
SAH = -40C to +110C	C165, C166,	FGR, USA, GTI,
SAK = -40C to +125C	C167 . . .	LUX, Z1, VW,
		Z1, " ", MX, GBR
		. . .

Index W	Index X	Index Y	Index Z	
ROM Size	ROM Type	Clock Frequency	Package	
" " = 0K	L = ROMless	always in MHz	Type	C = CDIP
1 = 8K	R = with ROM		A = CPGA	N = PLCC
2 = 16K	E = with		J = SOJ	R = CLCC
. . . .	EPROM		P = PDIP	M = MQFP
9 = 72K	F = with Flash		S = PQFP	
	EPROM			

**Note:The Option Field (Index V) can be increased as desired as long as the total part number length does not exceed 16.

SAB 8051A/8031A,
SAB 8051A-16/8031A-16

The SAB8051A is the original member of the SAB-51 family. Among the features of the SAB 8051A are:

- 8-Bit CPU Optimized for Control Applications
- Extensive Boolean Processing (Single-Bit Logic) Capabilities
- 32 Bidirectional and Individually Addressable I/O Lines
- 128 Bytes of On-Chip Data RAM
- RAM Power-Down Supply
- Two 16-Bit Timer/Counters
- Full Duplex UART
- 5-Source Interrupt Structure with 2 Priority Levels
- On-Chip Clock Oscillator
- 4 Kbytes of On-Chip Program Memory
- 64K Program Memory Address Space
- 64K Data Memory Address Space

The SAB 8051A/8031A is a standalone, high-performance single-chip micro-controller fabricated in +5V advanced Siemens MYMOS (III) technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PL-CC-44) package.

The SAB8031A differs from the SAB8051A, in not having the on-chip program ROM. Instead, the SAB 8031A fetches all instructions from external memory.

The SAB8051A-16 differs from the SAB8051A only in the speed of operation. The SAB8051A can run with a clock oscillator frequency from 1.2 MHz to 12 MHz whereas the SAB 8051A-16 can run up to a clock oscillator frequency of 16 MHz.

The extended temperature versions of these parts are also available.

SAB 8052B/8032B,
SAB 8052B-16/SAB 8032B-16,
SAB 8032B-20

The SAB 8052B/8032B is identical to the SAB 8051A/8031A and is also fabricated in advanced N-channel, Silicon gate Siemens MYMOS III technology. It is pin for pin compatible with the SAB8051A. Its enhancements over the SAB8051A are as

follows:
- 256 Bytes of On-Chip Data RAM
- Three Timer/Counters
- 6-Source Interrupt Structure
- 8 Kbytes of On-Chip Program ROM

The ROMless version of the 8052B is the 8032B. The 16 MHz version is the SAB 8052B-16/8032B-16. The ROMless version is also available in 20 MHz speed called the SAB 8032B-20. The extended temperature versions of these parts are also available. The earlier version of the SAB 8052B/8032B was fabricated in N-channel, silicon gate Siemens MYMOS II technology and was called the SAB 8052A/8032A. Whatever applies to the SAB 8052A/8032A also applies to the SAB 8052B/8032B except that the SAB 8052B/8032B is in the advanced N-channel, silicon gate Siemens MYMOS III technology.

SAB-C501-L/C501-1R
SAB-C501-L20/C501-1R20
SAB-C501-L40/C501-1R40

The SAB C501 is a standalone, high-performance CMOS single-chip microcontroller. The C501-L and C501-1R are drop-in replacements for the SAB 80C32 and SAB 80C52 respectively. The C501-L40 is capable of running with up to 40MHz crystal oscillator. The C501-L20 on the other hand runs with up to 20 MHz oscillator.

The SAB-C501-1R contains a non-volatile 8KX8 read-only program memory and can be used for all present SAB 80C52 applications. The SAB-C501-L on the other hand, lacks the program memory on-chip. The SAB-C501 is also available in extended temperature range. The part comes in both 40-pin DIP and 44 pin PLCC packages.

SAB-C502-L/C502-2R
SAB-C502-L20/C502-2R20

The SAB-C502 is an extended version of the SAB-C501. In addition to the C501 features, the SAB-C502 has 256 bytes of XRAM, 8 datapointers and a watchdog timer.

The SAB-C502-2R contains a non-volatile 16KX8 read-only program memory and the SAB-C502-L is the ROMless version of the same. The parts can also be used as drop-in replacements for the SAB 80C52 and 80C32 applications. The SAB-C502-L20/C502-2R20 run with up to 20 MHz oscillator. Extended temperature versions of these devices are also available. The devices come in both 40-pin DIP and 44-pin PLCC packages.

SAB-C503-L/C503-1R
SAB-C503-L20/C503-1R20

The SAB-C503 is another high-performance member of the C501 family. The device is functionally compatible to the 80C52/32 and is enhanced by the presence of an on-chip 8 inputs 10-bit A/D converter and two watchdog timers.

The SAB-C503-1R contains a non-volatile 8K Bytes of read-only program memory and the SAB-C503-L is the ROMless version of the same. The SAB-C503-L20/C503-2R20 run with up to 20 MHz oscillator. Extended temperature versions of these devices are also available.

SAB 80515/80535

The SAB 80515/80535 is a powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in +5V N-channel, silicon-gate Siemens MYMOS technology.

The SAB 80515/80535 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051 architecture. While maintaining all the SAB 8051 operating characteristics, the SAB 80515/80535 incorporates several enhancements which significantly increase design flexibility and overall system performance. These features are:
- 8K X 8 ROM (SAB 80515 Only)
- 256 X 8 RAM
- Six 8-Bit I/O Ports, One 8-Bit Input Port for Analog Signals

- Three 16-Bit Timer/Counters
- Programmable Timer/Counter Register-Array with Compare/Capture
- Auto Reload
- Pulse Width Modulator Capabilities
- Full-Duplex Serial Channel
- Twelve Interrupt Vectors, Four Priority Levels
- 8-Bit A/D Converter with Eight Multiplexed Inputs and Programmable Internal Reference Voltages
- 16-Bit Watchdog Timer
- VPD Provides Standby Current for 40 Bytes of RAM
- Boolean Processor
- 256 Bit-Addressable Locations
- Most Instructions Executed in: 1 µs (SAB 80515/80535)
- 4 µs Multiply and Divide
- Backwardly Compatible with SAB 8051
- 68-Pin Plastic Leaded Chip Carrier Package (PLCC 68)

The SAB 80535 is identical with the SAB 80515 except that it lacks the on-chip program memory. The SAB 80515/80535 is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68). The SAB 80515/80535 operates up to 12 MHz crystal oscillator frequency. The SAB 80515/80535 is also available in industrial temperature range (-40°C to +85°C) and in the automotive temperature range (-40°C to +110°C).

SAB 80C515/80C535, SAB 80C515-16/80C535-16

The SAB 80C515/80C535 is a new, powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80515/80535 devices designed in MYMOS technology.

In addition, the SAB 80C515 has two software-selectable power saving modes: idle mode and the power-down mode. These modes replace the power-down supply mode via pin VPD of the SAB 80515 (NMOS).

In the case of the SAB 80C515 the analog port can also be used as a digital input port.

The SAB 80C535 is identical with the SAB 80C515 except that it lacks the on-chip program memory. The SAB 80C515-16/80C535-16 operates up to 16 MHz crystal oscillator frequency. The SAB 80C515/80C535 is supplied in a 68-pin plastic leaded chip carrier package (PLCC 68). For the industrial temperature range -40°C to +85°C, the SAB 80C515/80C535-T40/85 and SAB 80C515/80C535-T40/85-16 are also available.

SAB 80C515A/83C515A-5

The SAB 80C515A/83C515A-5 is a high-end member of the Siemens SAB 8051 microcontroller family. It is designed in Siemens ACMOS technology and based on the SAB 8051 architecture. ACMOS is a technology which combines high-speed and density characteristics with low-power consumption and dissipation.

While maintaining all the SAB 80C515 features and operating characteristics the SAB 80C515A/83C515A-5 contains more on-chip RAM/ROM. Furthermore a new 10-bit A/D-Converter is implemented as well as extended security mechanisms. The SAB 80C515A is identical with the SAB 83C515A-5 except that it lacks the on-chip program memory. The SAB 80C515A/83C515A-5 is supplied in a 68-pin plastic leaded chip carrier package (P-LCC-68). The features are:

- SAB 80C515A/83515A-5, up to 18 MHz operation frequency
- 32K x 8 ROM (SAB 83C515A-5 only, ROM-Protection available)
- 256 x 8 on-chip RAM (XRAM)
- Superset of SAB 80C51 architecture:
  - 1µs instruction cycle time at 12 MHz
  - 666 ns instruction cycle time at 18 MHz
  - 256 directly addressable bits
  - Boolean processor
  - 64 Kbyte external data and program memory addressing
- Three 16-bit timer/counters

- Versatile "fail-safe" provisions
- Twelve interrupt vectors, four priority levels selectable
- 10-bit A/D converter with 8 multiplexed inputs
- Full duplex serial interface with programmable Baudrate-Generator
- Functionally compatible with SAB 80C515
- Extended power saving modes
- Fast Power-On Reset
- Six ports: 48 I/O lines, 8 input lines
- Three temperature ranges available
  0 to 70 °C
  -40 to 85 °C
  -40 to 110 °C
- Plastic package: PLCC-68

SAB 80C517/80C537
SAB 80C517-16/80C537-16

The SAB80C517/80C537 is a new and the most powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80C515/80C535 devices. The SAB 80C517-16/80C537-16 operates up to 16 MHz crystal oscillator frequency. While maintaining all the SAB 80C51 operating characteristics, the SAB 80C517/80C537 incorporates several enhancements which significantly increase design flexibility and overall system performance. These features are:

- 8 Kbyte On-Chip Program Memory
- 256 Byte On-Chip RAM
- 256 Directly Addressable Bits
- 1 µs Instruction Cycle at 12 MHz
- 750 µs Instruction Cycle at 16 MHz
- 64 of 111 Instructions Executed in One Cycle
- External Program and Data Memory Expandable to 64 Kbyte Each
  - 8-Bit A/D Converter
  - 12 Multiplexed Inputs
  - Programmable Reference Voltages
  - External/Internal Start of Conversion
- Two General Purpose 16-Bit Timers/ Counters (Timer 0, Timer 1)
- Compare Capture Unit (CCU
  - One 16-Bit Timer/Counter, 1 MHz Clock
  - One 16-Bit Compare-Timer, 6 MHz Clock with Dedicated Reload Register
  - One 16-Bit Reload/Capture/Compare Register
  - Four 16-Bit Capture/Compare Register
  - Eight 16-Bit Compare Register
  - Concurrent Compare
  - Pulse Width Modulation or High Speed Output Possible on up to 21 Channels
  - Fine Capture Input Channels
- Two Full Duplex Serial Interfaces with Own Baud Rate Generator
- Four Priority Level Interrupt System, 14 Interrupt Sources
- Extended Arithmetic Capabilities for Division and Multiplication (Mul./Div. Unit Operations to Fast 16/32-Bit)
- Eight Datapointers for Indirect Addressing
- Extended Fail Safe Mechanism
  - 16-Bit Programmable Watchdog Time
  - Oscillator Watchdog
  - Hardware Disable for Power Saving Modes
- Extended Power Saving Modes (Slow Down, Idle, Power-Down)
- Nine Port
  - Seven Bidirectional 8-Bit Port
  - One 8-Bit, One 4-Bit Input Port
- 84 Pin PLCC Package

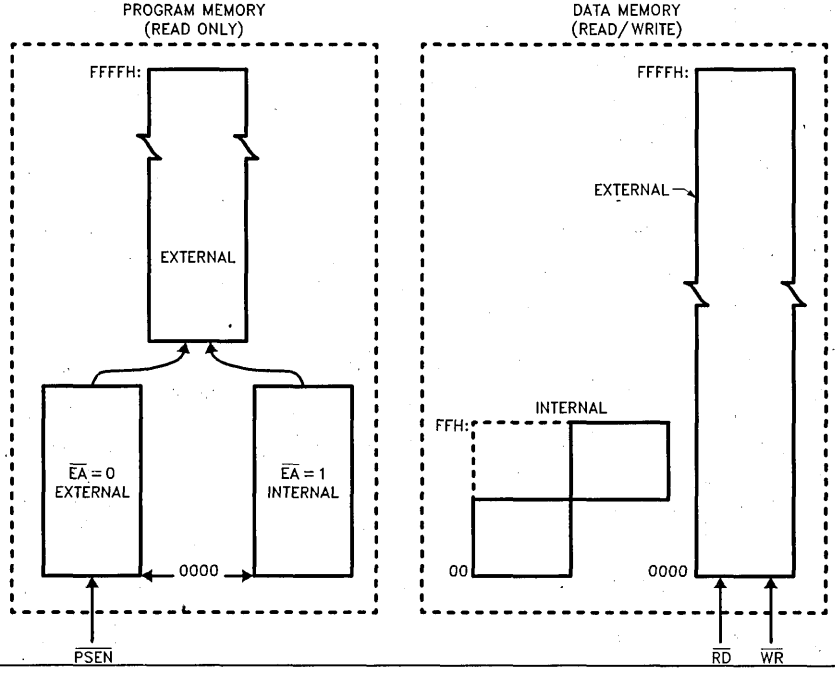
SAB 80C517A/83C517A-5

The SAB 80C517A/83C517A-5 is a high-end member of the Siemens SAB 8051 family of microcontrollers. It is designed in Siemens ACMOS technology and based on SAB 8051 architecture. ACMOS is a technology which combines high-speed and density characteristics with low-power consumption or dissipation.

While maintaining all the SAB 80C517 features and operating characteristics the SAB 80C517A is expanded in its "fail-safe" characteristics and timer capabilities. The SAB 80C517A is identical with the SAB 83C517A-5 except that it lacks the on-chip program memory. The SAB 80C517A/ 83C517A-5 is supplied in a 84-pin plastic leaded chip carrier package (P-LCC-84) and in a 100-pin plastic quad flat package (P-MQFP-100). The features are:

- SAB 80C517A/83C517A-5, up to 18 MHz operation
- 32 K x 8 ROM (SAB 83C517A-5 only, ROM-Protection available)
- 256 x 8 on-chip RAM (XRAM)
- Superset of SAB 80C51 architecture:
  - 1 µs instruction cycle time at 12 MHz
  - 666 ns instruction cycle time at 18 MHz
  - 256 directly addressable bits
  - Boolean processor
  - 64 Kbyte external data and program memory addressing
- Four 16-bit timer/counters
- Powerful 16-bit compare/capture unit (CCU) with up to 21 high-speed or PWM output channels and 5 capture inputs
- Versatile "fail-safe" provisions
- Fast 32-bit division, 16-bit multiplication, 32-bit normalize and shift by peripheral MUL/DIV unit (MDU)
- Eight data pointers for external memory addressing
- Seventeen interrupt vectors, four priority levels selectable
- 10-bit A/D converter with 12 multiplexed inputs
- Two full duplex serial interfaces with programmable Baudrate-Generators
- Fully upward compatible with SAB 80C515m SAB 80517, SAB 80C515A
- Extended power saving modes
- Fast Power-On Reset
- Nine ports: 56 I/O lines, 12 input lines
- Three temperature ranges available:
  0 to 70°C
  -40 to 85°C
  -40 to 110°C
- Plastic packages: P-LCC-84, P-MQFP-100

Figure 2
SAB-51 Memory Structure



Memory Organization in SAB-51 Devices

Logical Separation of Program and Data Memory

All SAB-51 devices have separate address spaces for Program and Data Memory, as shown in Figure 2. The logical separation of Program and Data Memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

Program Memory can only be read, not written to. There can be up to 64 Kbytes of Program Memory. In the 8051A, 80C51 and 80512 the lowest 4 Kbytes of Program Memory are on-chip. The 8052B, 80515, 80C515 and 80C517 provide 8 Kbytes of on-chip Program Memory storage. The SAB 80513 has 16K of on-chip program memory.

In the ROMless versions all Program Memory is external. The read strobe for external Program Memory is the signal $\overline{\text{PSEN}}$ (Program Store Enable).

Data Memory occupies a separate address space from Program Memory. Up to 64 Kbytes of external RAM can be addressed in the external Data Memory space. The CPU generates read and write signals, $\overline{\text{RD}}$ and $\overline{\text{WR}}$, as needed during external Data Memory accesses.

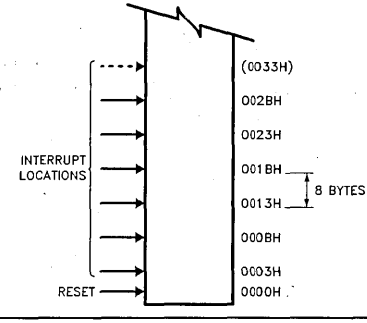
External Program Memory and external Data Memory may be combined if desired by applying the RD and PSEN signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

Figure 3
SAB-51 Program Memory



The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4K (or 8K, in the 8052B, 80515 and 80C517) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the EA (External Access) pin to either V_{CC} or V_{SS}.

In 8051A, 80C51 and 80512, if the $\overline{EA}$ pin is strapped to V_{CC}, then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

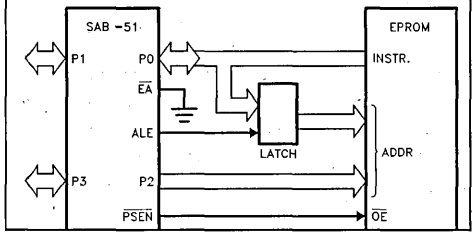
In the 8052B and the other 8K ROM parts, $\overline{EA}$ = V_{CC} selects addresses 0000H through 1FFFH to be internal, and addresses 2000H through FFFFH to be external.

In the 80513, $\overline{EA}$ = V_{CC} selects addresses 0000H through 3FFFH to be internal and addresses 4000H through FFFH to be external.

If the $\overline{EA}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts 8031A, 8032B, 80532 etc. must have this pin externally strapped to V_{SS} to enable them to execute from external Program Memory.

The read strobe to external ROM, $\overline{PSEN}$, is used for all external program fetches. $\overline{PSEN}$ is not activated for internal program fetches.

Figure 4
Executing from External Program Memory



The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). The $\overline{PSEN}$ strobes the EPROM and the code byte is read into the microcontroller.

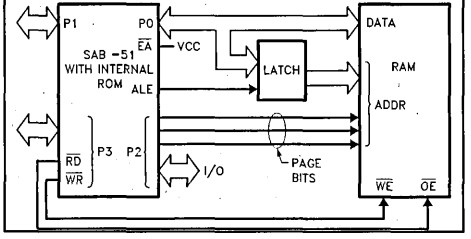
Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64 Kbytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the SAB-51 user.

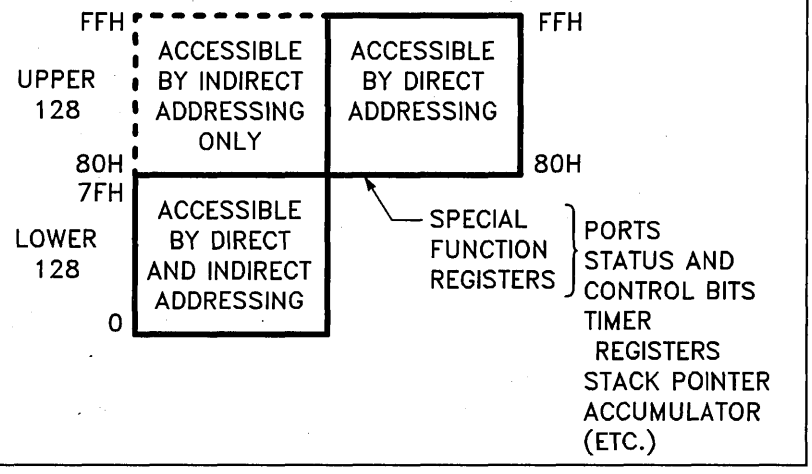
Figure 5 shows a hardware configuration for accessing up to 2 Kbytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM and 3 lines of Port 2 are being used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses.

Figure 5
Accessing External Data Memory If the Program Memory is Internal, the Other Bits of P2 are Available as I/O.



There can be up to 64 Kbytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5. Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

Figure 6
Internal Data Memory

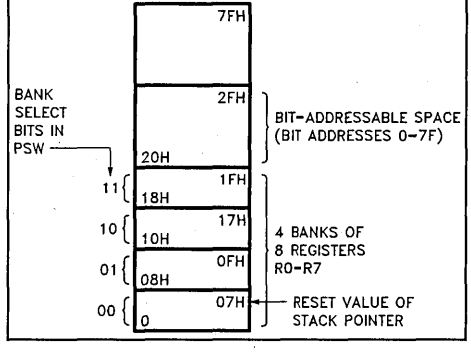


Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128 and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The Lower 128 bytes of RAM are present in all SAB-51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

Figure 7
The Lower 128 Bytes of Internal RAM



The next 16 bytes above the register banks form a block of bit-addressable memory space. The SAB-51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

Figure 8
The Upper 128 Bytes of Internal RAM

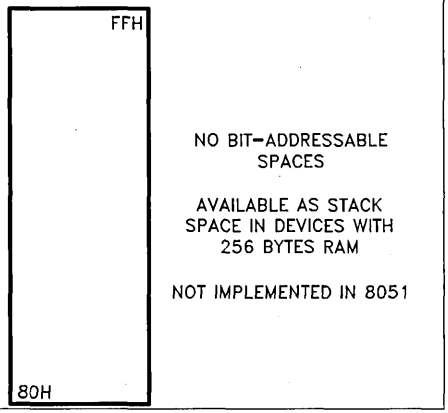


Figure 9
SFR Space

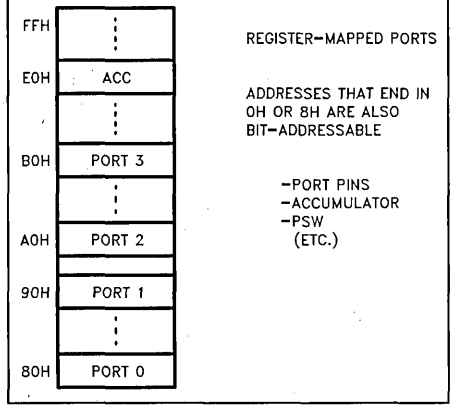
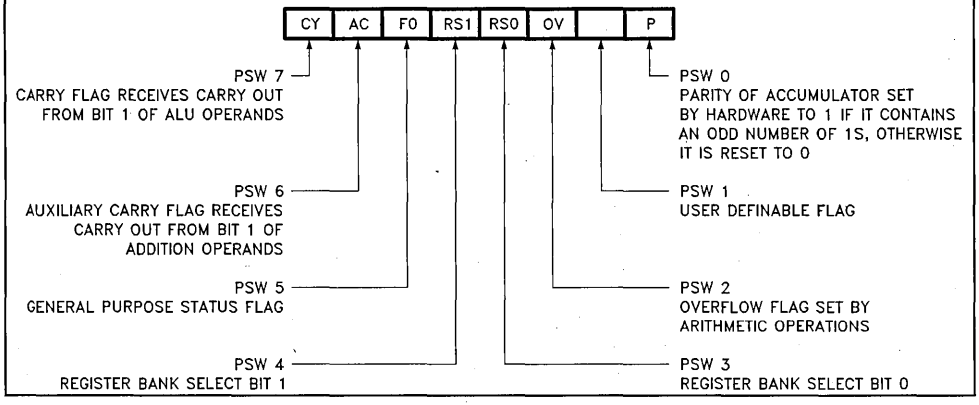


Figure 10
PSW (Program Status Word) Register SAB-51 Devices



All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are not implemented in the 8051A, but are in the 8052B, 80515 and 80C517.

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFR include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all SAB-51 microcontrollers have the same SFRs as the 8051. However, enhancements to the 8051 have additional SFRs that are not present in the 8051, nor perhaps in other proliferations of the family.

The bit addresses in this area are 80H through FFH.

The SAB-51 Instruction Set

All members of the SAB-51 family execute the same instruction set. SAB-51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

An overview of the SAB-51 instruction set is presented below, with a brief description of how certain instructions might be used. Refer to the chapter on instruction set in this book for detailed information on the instructions.

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the functions of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four banks is being referred to is made on the basis of the bits RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and may be used as general purpose status flags.

Addressing Modes

The addressing modes in the SAB-51 instruction set are as follows:

Direct Addressing

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

Indirect Addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected register bank or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

Register Instructions

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

Register-Specific Instructions

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumulator as A assemble as accumulator-specific opcodes.

Immediate Constants

The value of a constant can follow the opcode in Program Memory. For example,

MOV A, #100

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

Indexed Addressing

Only Program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

With the on-chip arithmetic unit the SAB 80C517 has a special provision for fast multiplication and division. Refer to the SAB 80C517 User's Manual for detailed information on this logic. However, the following information applies to all the members of the SAB-51 family.

The menu of arithmetic instructions is listed in Table 2. The table indicates the addressing modes that can be used with each instruction to access the byte operand. For example, the ADD A, <byte> instruction can be written as:

ADD	A,7FH	(direct addressing)
ADD	A,@R0	(indirect addressing)
ADD	A,R7	(register addressing)
ADD	A,#127	(immediate constant)

The execution times listed in Table 2 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 µs except the INC DPTR instruction, which takes 2 µs, and the Multiply and Divide instructions, which take 4 µs.

Note that any byte in the internal Data Memory space can be incremented or decremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

Table 2
A List of the SAB-51 Arithmetic Instructions

Mnemonic		Operation	Addressing Modes				Execution Time (µs)
			Dir	Ind	Reg	Imm	
ADD	A, <byte>	A = A + <byte>	X	X	X	X	1
ADDC	A, <byte>	A = A + <byte> + C	X	X	X	X	1
SUBB	A, <byte>	A = A - <byte> - C	X	X	X	X	1
INC	A	A = A + 1	Accumulator Only				1
INC	<byte>	<byte> = <byte> + 1	X	X	X	X	1
INC	DPTR	DPTR = DPTR + 1	Data Pointer Only				2
DEC	A	A = A - 1	Accumulator Only				1
DEC	<byte>	<byte> = <byte> - 1	X	X	X	X	1
MUL	AB	B:A = B X A	ACC and B Only				4
DIV	AB	A = Int [A/B]	ACC and B Only				4
		B = Mod [A/B]					4
DA	A	Decimal Adjust	Accumulator Only				1

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 µs and leaves the B register holding the bits that were shifted out.

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Logical Instructions

Table 3 shows the list of SAB-51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and byte contains 01010011B, then

ANL A,<byte>

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the byte operand are listed in Table 3. Thus, the ANL A, <byte> instruction may take any of the forms:

```
ANL   A,7FH      (direct addressing)
ANL   A,@R1      (indirect addressing)
ANL   A,R6       (register addressing)
ANL   A,#53H     (immediate constant)
```

Table 3
A List of the SAB-51 Logical Instructions

Mnemonic		Operation	Addressing Modes				Execution Time (s)
			Dir	Ind	Reg	Imm	
ANL	A, <byte>	A = A.AND. <byte>	X	X	X	X	1
ANL	<byte>, A	<byte> = <byte> .AND.A	X				1
ANL	<byte>, #data	<byte> = <byte> .AND.#data	X				2
ORL	A,<byte>	A = A.OR. <byte>	X	X	X	X	1
ORL	<byte>,A	<byte> = <byte> .OR.A	X				1
ORL	<byte>,#data	<byte> = <byte>.OR.#data	X				2
XRL	A,<byte>	A = A.XOR.<byte>	X	X	X	1	1
XRL	<byte>, A	<byte> = <byte> .XOR.A	X				1
XRL	<byte>, #data	<byte> = <byte> .XOR.#data	X				2
CRL	A	A = 00H	Accumulator Only				1
CPL	A	A = .NOT.A	Accumulator Only				1
RL	A	Rotate ACC Left 1 Bit	Accumulator Only				1
RLC	A	Rotate Left through Carry	Accumulator Only				1
RR	A	Rotate ACC Right 1 Bit	Accumulator Only				1
RRC	A	Rotate Right through Carry	Accumulator Only				1
SWAP	A	Swap Nibbles in A	Accumulator Only				1

All of the logical instructions that are Accumulator-specific execute in 1 µs (using a 12 MHz clock). The others take 2 µs.

Note that Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in:

```
XRL     R1,#0FFH
```

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to stack it in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code.

```
MOV     B, #10
DIV     AB
SWAP    A
ADD     A, B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator and the ones digit to the low nibble.

Data Transfers

Internal RAM

Table 4 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12 MHz clock, all of these instructions execute in either 1 µs or 2 µs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember the Upper 128 bytes of data RAM can be accessed only by indirect addressing and SFR space only by direct addressing.

Note that in all SAB-51 devices, the stack resides in on-chip RAM, and grows upwards.

The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

Table 4
A List of the SAB-51 Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic		Operation	Addressing Modes				Execution Time (s)
			Dir	Ind	Reg	Imm	
MOV	A, <src>	A = <src>	X	X	X	X	1
MOV	<dest>, A	<dest> = A	X	X	X		1
MOV	<dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV	DPTR, #data16	DPTR = 16-Bit Immediate Constant				X	2
PUSH	<src>	INC SP: MOV "@SP", <src>	X				2
POP	<dest>	MOV <dest>, "@SP": DEC SP	X				2
XCH	A, <byte>	ACC and <byte> Exchange Data	X	X	X		1
XCHD	A, @Ri	ACC and @Ri Exchange Low Nibbles		X			1

The Upper 128 Bytes are not implemented in the 8051A, 80C51 and 80512, nor in their ROMless counterparts. With these devices, if the SP points to the Upper 128, PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses. Refer to the SAB 80C517 Data Sheet for extended Data Pointer Capabilities.

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A, @Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

Figure 11
Shifting a BCD Number Two Digits to the Right

	2A	2B	2C	2D	2E	ACC
MOV A, 2EH	00	12	34	56	78	78
MOV 2EH, 2DH	00	12	34	56	56	78
MOV 2DH, 2CH	00	12	34	34	56	78
MOV 2CH, 2BH	00	12	12	34	56	78
MOV 2BH, #0	00	00	12	34	56	78
(a) Using Direct MOVs: 14 Bytes, 9 µs						
	2A	2B	2C	2D	2E	ACC
CLR A	00	12	34	56	78	00
XCH A, 2BH	00	00	34	56	78	12
XCH A, 2CH	00	00	12	56	78	34
XCH A, 2DH	00	00	12	34	78	56
XCH A, 2EH	00	00	12	34	56	78
(b) Using XCH_{S1} 9 Bytes, 5 µs						

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 µs of execution time (assuming a 12 MHz clock). The same operation with XCHs uses less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

Figure 12
Shifting a BCD Number One Digit to the Right

	2A	2B	2C	2D	2E	ACC
MOV R1, #2EH	00	12	34	56	78	XX
MOV R0, #2DH	00	12	34	56	78	XX
Loop for R1 = 2EH:						
LOOP: MOV A, @R1	00	12	34	56	78	76
XCHD A, @R0	00	12	34	58	78	76
SWAP A	00	12	34	58	78	67
MOV @R1, A	00	12	34	58	67	67
DEC R1	00	12	34	58	67	67
DEC R0	00	12	34	58	67	67
CJNE R1, #2AH, LOOP						
Loop for R1 = 2DH:	00	12	38	45	67	45
Loop for R1 = 2CH:	00	18	23	45	67	23
Loop for R1 = 2BH:	08	01	23	45	67	01
CLR A	08	01	23	45	67	00
XCH A, 2AH	00	01	23	45	67	08

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 2EH, 2DH, 2CH and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

External RAM

Table 5 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few Kbytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few Kbytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2.

All of these instructions execute in 2 µs, with a 12 MHz clock.

Table 5
A List of the SAB-51 Data Transfer Instructions that Access External Data Memory Space

Address Width	Mnemonic	Operation	Execution Time (µs)
8 Bits	MOVX A, @Ri	Read External RAM @Ri	2
8 Bits	MOVX @Ri, A	Write External RAM @Ri	2
16 Bits	MOVXA, @DPTR	Read External RAM @DPTR	2
16 Bits	MOVX@DPTR, A	Write External RAM @DPTR	2

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines. More about that later.

Lookup Tables

Table 6 shows the two instructions that are available for reading loopup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated. The mnemonic is MOVC for "move constant".

If the table access is to external Program Memory, then the read strobe is $\overline{PSEN}$.

Table 6
The SAB-51 Lookup Table Read Instructions

Mnemonic	Operation	Execution Time (µs)
MOVC A, @A +DPTR	Read PGM Memory at (A + DPTR)	2
MOVC A, @A + PC	Read PGM Memory at (A + PC)	2

The first MOVC instruction in Table 6 can accommodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then

```
MOVC   A, @A + DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
MOV    A, ENTRY_ NUMBER
CALL   TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE: MOVC    A, @ A + PC
       RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This

type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

SAB-51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

Table 7
A List of the SAB-51 Boolean Instructions

Mnemonic		Operation	Execution Time (µs)
ANL	C,Bit	C = C.AND.Bit	2
ANL	C,/Bit	C = C.AND..NOT.Bit	2
ORL	C,Bit	C = C.OR.Bit	2
ORL	C,/Bit	C = C.OR..NOT.Bit	2
MOV	C,Bit	C = Bit	1
MOV	Bit, C	Bit = C	2
CLR	C	C = 0	1
CLR	Bit	Bit = 0	1
SETB	C	C = 1	1
SETB	Bit	Bit = 1	1
CPL	C	C = .NOT.C	1
CPL	Bit	Bit = .NOT.Bit	1
JC	Rel	Jump if C = 1	2
JNC	Rel	Jump if C = 0	2
JB	Bit,Rel	Jump if Bit = 1	2
JNB	Bit,Rel	Jump if Bit = 0	2
JBC	Bit,Rel	Jump if Bit = 1; CLR Bit	2

The instruction set for the Boolean processor is shown in Table 7. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128 and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV     C,FLAG
MOV     P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

C = Bit1 .XRL. bit 2

The software to do that could be as follows:

```
MOV     C,bit1
JNB     bit2,OVER
CPL     C
```

OVER: (Continue)

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit 2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a

series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0 the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program Memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore -128 to +127 Program Memory bytes relative to the first byte following the instruction.

Jump Instructions

Table 8 shows the list of unconditional jumps.

Table 8
Unconditional Jumps in SAB-51 Devices

Mnemonic	Operation	Execution Time (µs)
JMP addr	Jump to addr	2
JMP @A + DPTR	Jump to A + DPTR	2
CALL Addr	Call Subroutine at addr	2
RET	Return from Subroutine	2
RETI	Return from Interrupt	2
NOP	No Operation	1

The Table lists a single "JMP addr" instruction, but in fact there are three— SJMP, LJMP and AJMP— which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of -128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A + DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the

Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV     DPTR,#JUMP_TABLE
MOV     A,INDEX_NUMBER
RL      A
JMP     @A + DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP_TABLE:
        AJMP    CASE_0
        AJMP    CASE_1
        AJMP    CASE_2
        AJMP    CASE_3
        AJMP    CASE_4
```

Table 8 shows a single "CALL addr' "instruction, but there are two of them—LCALL and ACALL—which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 9
Conditional Jumps in SAB-51 Devices

Mnemonic		Operation	Addressing Modes				Execution Time (s)
			Dir	Ind	Reg	Imm	
JZ	Rel	Jump if A = 0	Accumulator Only				2
JNZ	Rel	Jump if A ≠ 0	Accumulator Only				2
DJNZ	<byte>,rel	Decrement and Jump if Not Zero	X		X		2
CJNE	A, <byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE	<byte>, #data, rel	Jump if <byte> ≠ #data		X	X		2

Table 9 shows the list of conditional jumps available to the SAB-51 user. All of these jumps specify the destination address by the relative offset method; and so are limited to a jump distance of -128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
        MOV COUNTER, #10
LOOP:   (begin loop)
        •
        •
        •
        (end loop)
        DJNZ COUNTER, LOOP
```

(Continue)

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop as executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

3

CPU Timing

All SAB-51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller and capacitors to ground as shown in Figure 13.

Figure 13
Using the On-Chip Oscillator

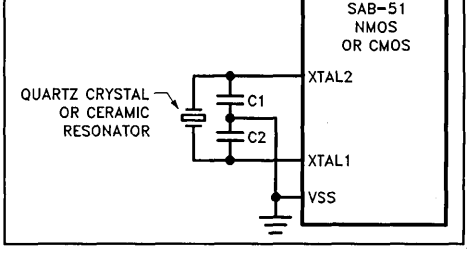
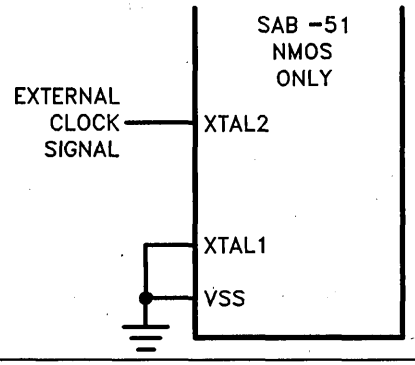


Figure 14
Using an External Clock



Example of how to drive the clock with an external oscillator is shown in Figure 14. Note that in the NMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CMOS devices 80C51, 80C52 the signal at the XTAL1 pin drives the internal clock generator, whereas in 80C515 and 80C517 the signal at the XTAL2 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin. For 80C515 and 80C517, refer to their data-sheets for the description of the clock oscillator pins.

The internal clock generator defines the sequence of states that make up the SAB-51 machine cycle.

Machine Cycles

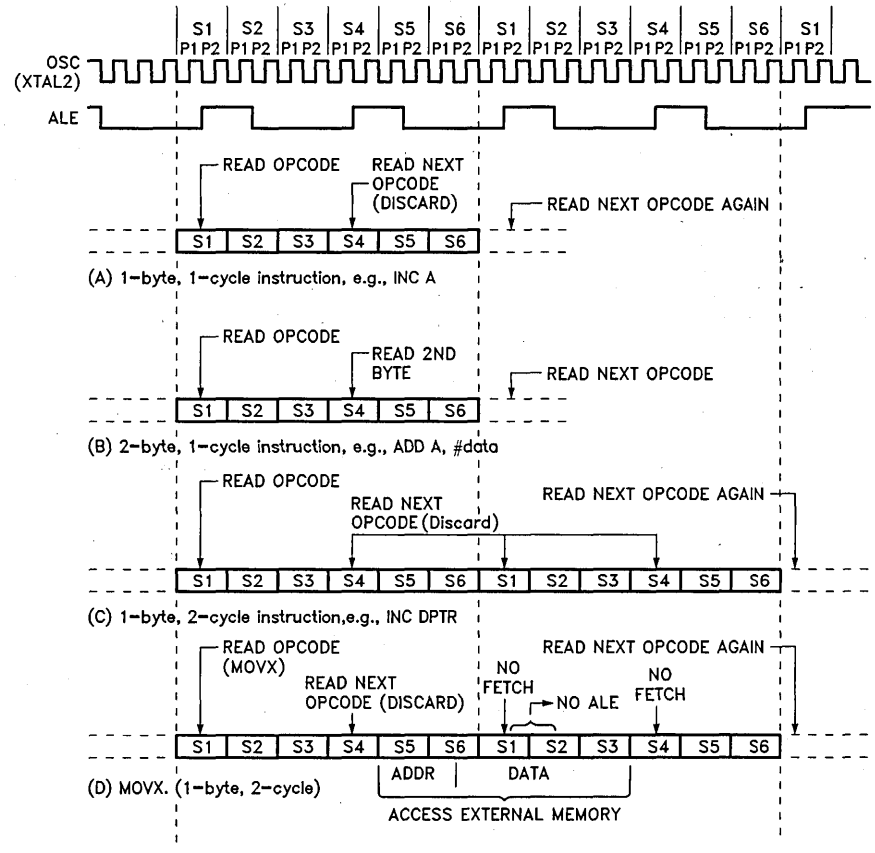
A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 µs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/ execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/ execute sequence for MOVX instructions is shown in Figure 15.

Figure 15
State Sequences in SAB-51 Devices



The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe $\overline{PSEN}$ is normally activated twice per machine cycle, as shown in Figure 16(A).

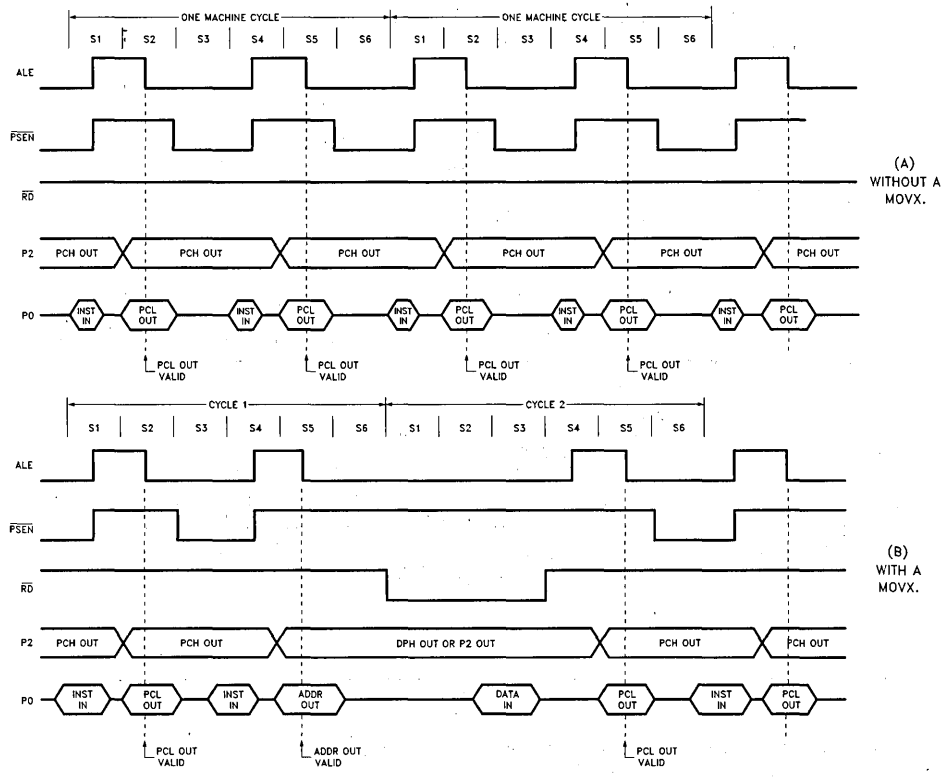
If an access to external Data Memory occurs, as shown in Figure 16(B), two $\overline{PSEN}$s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and $\overline{PSEN}$. ALE is

used to latch the low address byte from P0 into the address latch.

When the CPU is executing from internal Program Memory, $\overline{PSEN}$ is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Figure 16
Bus Cycles in SAB-51 Devices Executing from External Program Memory



Interrupt Structure

The 8051A and 80C51 and their ROMless versions, provide 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt. The 8052B and 80C52 provide these 5 plus a sixth interrupt that is associated with the third timer/counter which is present in the device. Additional interrupts are available on the 80512, 80515, 80C515 and 80C517. Refer to the appropriate chapters on these devices for further information on their interrupts.

What follows is an overview of the interrupt structure for these devices. More detailed information for specific members of the SAB-51 family is provided in the chapters of this handbook that describe the specific devices.

Interrupt Enables

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 80528 and 80C52.

Figure 17
IE (Interrupt Enable) Register in the 8052B and 80C52

(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	reserved.
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

Interrupt Priorities

Each interrupt source can also be individually programmed to one of two priority levels (SAB 80515, 80C515 and 80C517 have four priority levels for the interrupts, please refer to their respective chapters for detailed information). By setting clearing a bit in the SFR named IP (Interrupt Priority) Figure 18 shows the IP register in the 8052B and 80C52.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another

low-priority interrupt. A high-priority can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 8052B, how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

Figure 18
IP (Interrupt Priority) Register in the 8052B and 80C52

(MSB)							(LSB)
–	–	PT2	PS	PT1	PX1	PT0	PX0

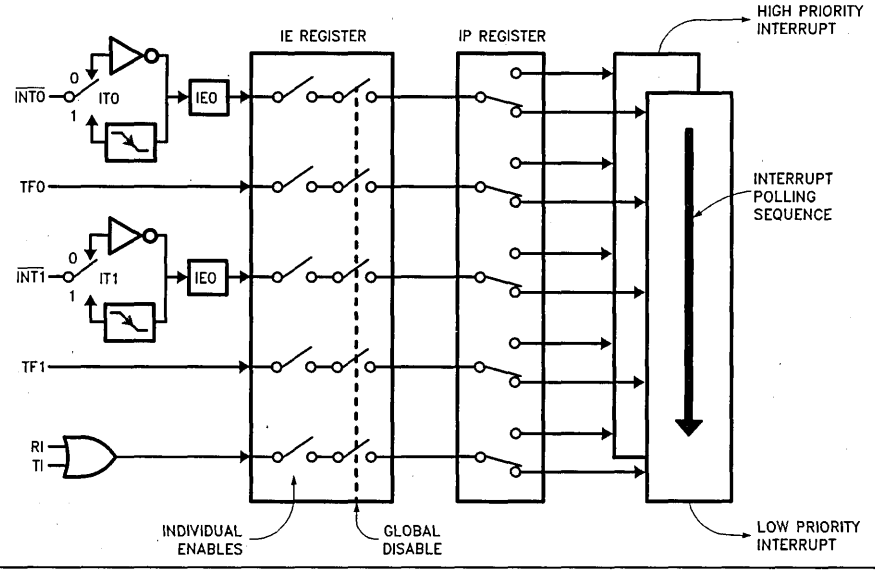
Symbol	Position	Function
–	IP.7	reserved
–	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 program it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the External Interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level, PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC be automatically saved allows the programmer to decide how much time to spend saving which other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, for example, or reloading a timer, or unloading a serial buffer—can often be completed in less time than it takes other architectures to commence them.

Figure 19
8052 Interrupt Control System



Simulating a Third Priority Level in Software

Some applications require more than the two priority levels that are provided by on-chip hardware in SAB-51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level.

First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the IP (Interrupt Priority) register. The service routines for priority 1 interrupts that are supposed to be interruptible by "priority 2" interrupts are written to include the following code:

```
        PUSH    IE
        MOV     IE,#MASK
        CALL    LABEL
                *******
        (execute service routine)
                *******
        POP     IE
        RET
LABEL:  RETI
```

As soon as any priority 1 interrupt is acknowledged, the IE (Interrupt Enable) register is re-defined so as to disable all but "priority 2" interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only "priority 2" interrupts are enabled. POPping IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 µs (at 12 MHz) to priority 1 interrupts.

8-Bit Single-Chip Microcontroller

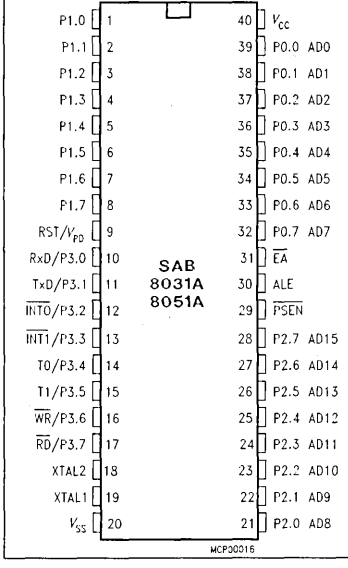
SAB 8051A Microcontroller with factory mask-programmable ROM

SAB 8031A Microcontroller for external ROM

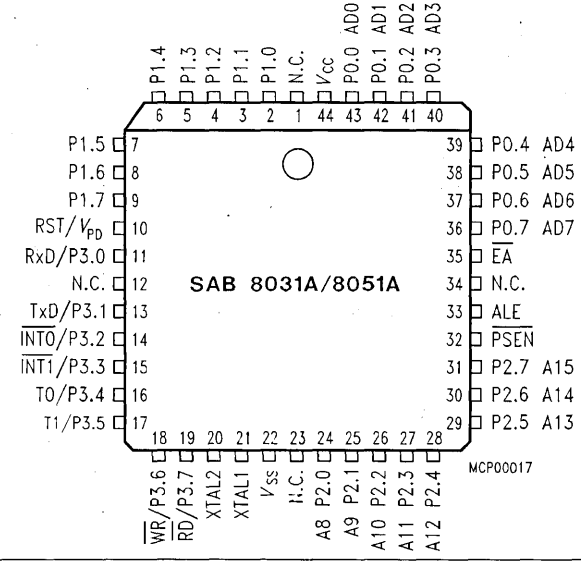
- Version for 12MHz/16MHz/ 20 MHz operating frequency
- 4K × 8 ROM
- 128 × 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- High-performance full-duplex serial channel with flexible transmit/receive bound rate capability
- External memory expandable up to 128 Kbyte
- Compatible with SAB 8080/8085 peripherals
- Boolean processor
- 218 user bit-addressable locations
- Most instructions execute in:
  1 µs instruction cycle time at 12 MHz
  750 ns instruction cycle time at 16 MHz
  600 ns instruction cycle time at 20 MHz
- 4 µs (3 µs, 2.4 µs) multiply and divide
- Packages P-DIP-40 and PL-CC-44
- Two temperature ranges available
  0 to 70 °C
  –40 to 85 °C: T40/85

4

Figure 1
Pin Configuration
P-DIP-40



PL-CC-44



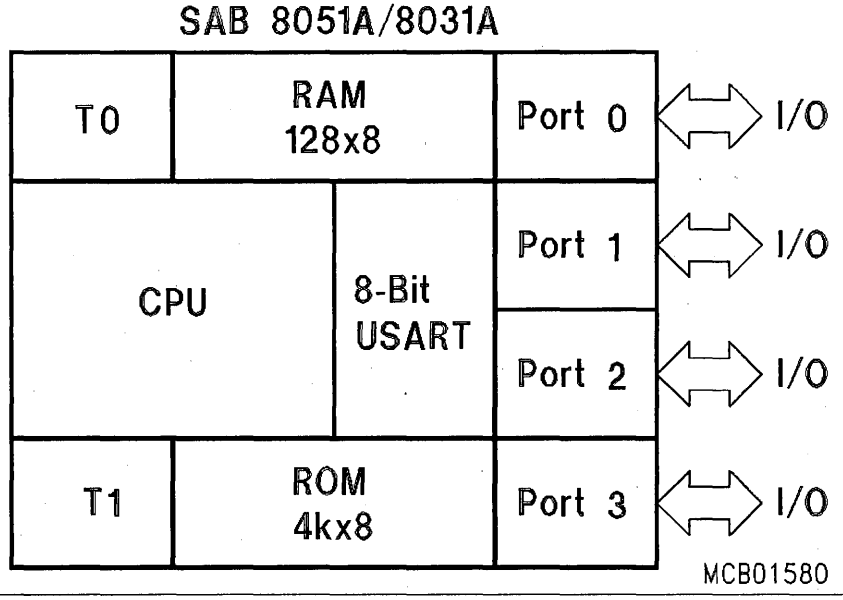
The SAB 8051A/8031A Family are standalone, high-performance single-chip microcontrollers fabricated in + 5 V advanced N-channel, silicon-gate Siemens MYMOS technology and supplied in a 40-pin plastic P-DIP or 44-pin plastic leaded chip carrier (PL-CC-44) package. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64 Kbytes of program memory and/or up to 64 Kbytes of data storage.

The SAB 8051A contains a non-volatile 4K × 8 read-only program memory; a volatile 128 × 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full-duplex UART; and on-chip oscillator and clock circuits. The SAB 8031A is identical with the SAB 8051A, except that it lacks the program memory.

For systems that require extra capability, the SAB 8051A can be expanded using standard TTL-compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals.

The parts are available for standard temperature range (0 to 70 °C) and extended temperature range (T40/85: – 40 to 85 °C).

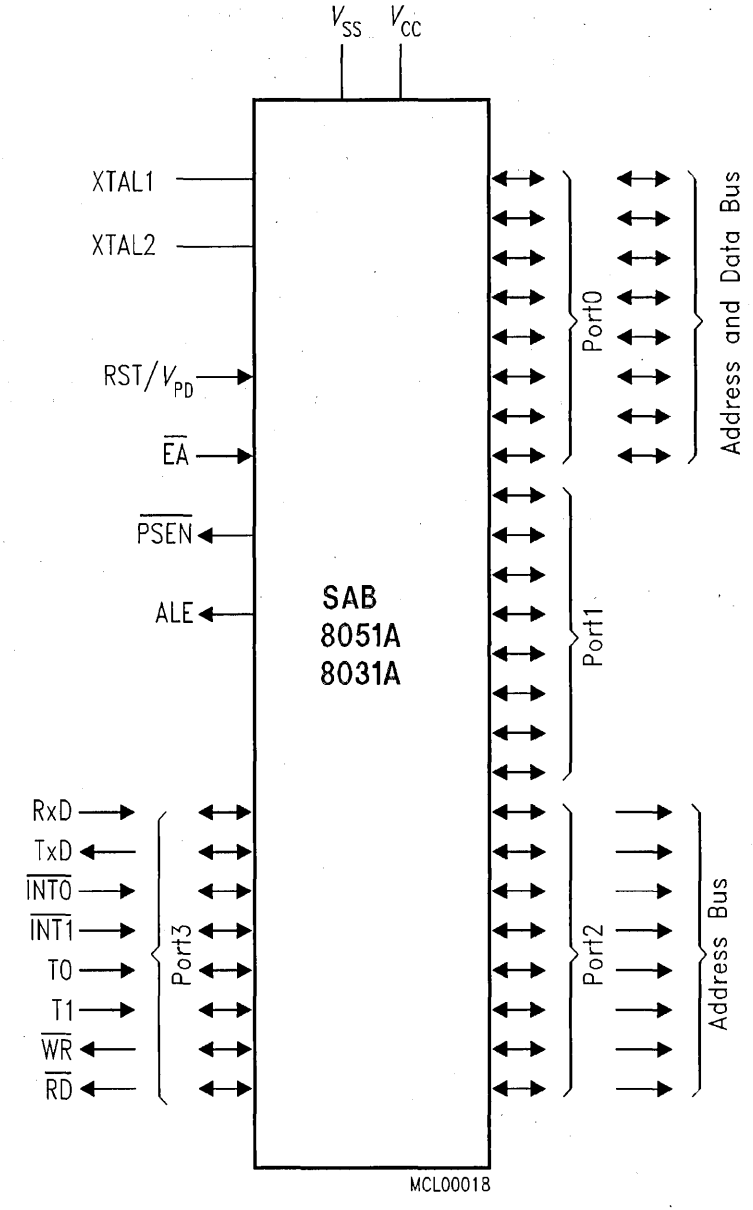
Figure 2
SAB 8051A/8031A Family



Ordering Information

Type	Package	Description (8-bit single-chip microcontroller)
SAB 8031A-P-40/85	P-DIP-40	for external memory, 12 MHz, ext. Temp.
SAB 8031A-N-40/85	PL-CC-44	
SAB 8031A-P	P-DIP-40	for external memory, 12 MHz
SAB 8031A-N	PL-CC-44	
SAB 8031A-16-P	P-DIP-40	for external memory, 16 MHz
SAB 8031A-16-N	PL-CC-44	
SAB 8031A-20-P	P-DIP-40	for external memory, 20 MHz
SAB 8031A-20-N	PL-CC-44	
SAB 8051A-P-40/85	P-DIP-40	with 4-KByte mask-programmable ROM 12 MHz, ext. Temp.
SAB 8051A-P	P-DIP-40	with 4-KByte mask-programmable ROM 12 MHz
SAB 8051A-N	PL-CC-44	
SAB 8051A-16-P	P-DIP-40	with 4-KByte mask-programmable ROM 16 MHz
SAB 8051A-16-N	PL-CC-44	
SAB 8051A-20-P	P-DIP-40	with 4-KByte mask-programmable ROM 20 MHz
SAB 8051A-20-N	PL-CC-44	

Figure 3
Logic Symbol



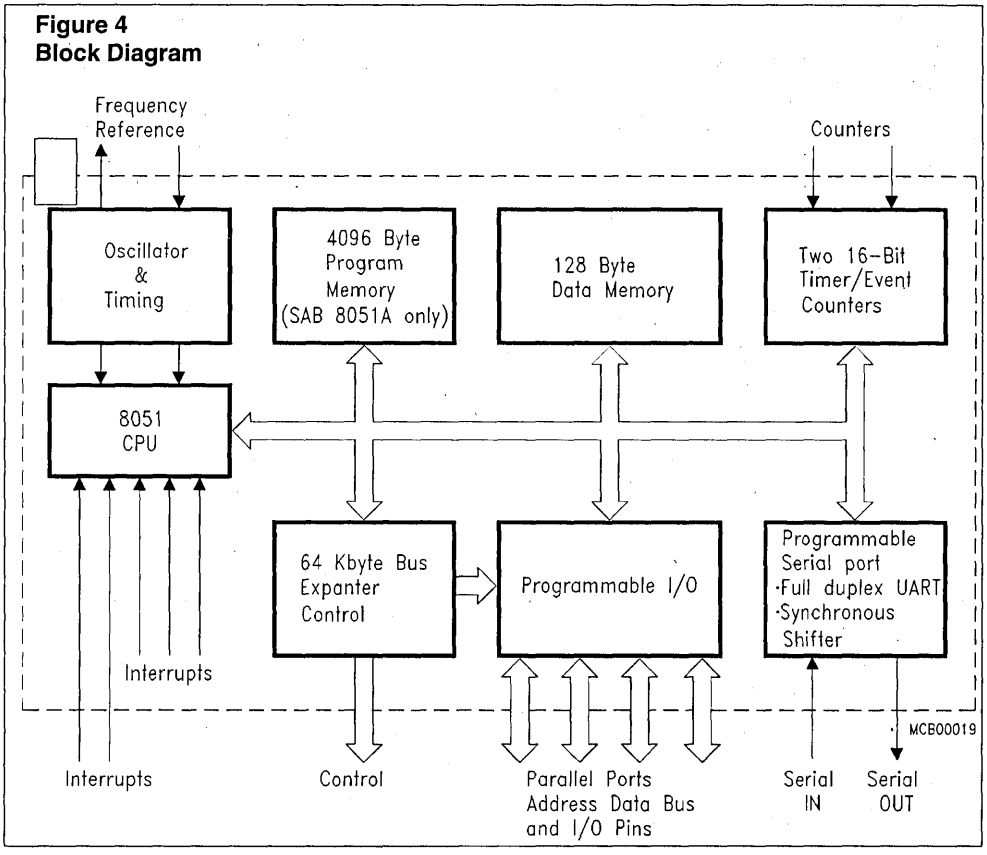
Pin Definitions and Functions

Symbol	Pin		Input (I) Output (O)	Function
	P-DIP-40	PL-CC-44		
P1.0-P1.7	1-8	2-9	I/O	**PORT 1** is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
RST/V_{PD}	9	10	I	**RESET** A high level on this pin resets the SAB 8051A. A small internal pulldown resistor permits power-on reset using only a capacitor connected to V_{CC}. If V_{PD} is held within its spec while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
P3.0-P3.7	10-17	11, 13-19	I/O	**PORT 3** is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of port 3, as follows: –RxD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). –TxD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). –$\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. –$\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. –T0 (P3.4). Input to counter 0. –T1 (P3.5). Input to counter 1. –$\overline{WR}$ (P3.6). The write control signal latches the data byte from port 0 into the external data memory. –$\overline{RD}$ (P3.7). The read control signal enables external data memory to port 0.
XTAL 1 XTAL 2	19 18	21 20		**XTAL 1** input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on XTAL 2. **XTAL 2** output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.
P2.0-P2.7	21-28	24-31	I/O	**PORT 2** is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
$\overline{PSEN}$	29	32	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

Pin Definitions and Functions (cont'd)

Symbol	Pin		Input (I) Output (O)	Function
	P-DIP-40	PL-CC-44		
ALE	30	33	O	Provides **Address Latch Enable** output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	31	35	I	**External Latch Enable** when held at a TTL high level, the SAB 8051A executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the SAB 8051A fetches all instructions from external program memory. For the SAB 8031A this pin must be tied low.
P0.0-P0.7	39-32	43-36	I/O	**Port 0** is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
V_{CC}	40	44		+ 5 V **Power Supply** during operation and program verification.
V_{SS}	20	22		**Ground** (0 V)
NC	–	1, 12 23, 34	–	**No Connection**

Figure 4
Block Diagram



4

Absolute Maximum Ratings

Ambient temperature under bias .....
SAB 8051A/8031A..... 0 to 70 °C
SAB 8051A/8031A-T40/85 ..... – 40 to 85 °C
Storage temperature..... – 65 to 150 °C
Voltage on V_{CC} pins with
respect to ground (V_{SS})..... – 0.5 to 7 V
Power dissipation ..... 2 W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to 70 °C for the SAB 8051A/8031A
T_A = – 40 to 85 °C for the SAB 8051A/8031A-T40/85

Symbol	Parameter	Limit Values		Unit	Test Condition
		min.	max.		
V_{IL}	Input low voltage	– 0.5	0.8	V	–
V_{IH}	Input high voltage (except RST/VPD and XTAL 2)	2.0	V_{CC} + 0.5	V	–
V_{IH1}	Input high voltage to RST/VPD for reset, XTAL 2	2.5	V_{CC} + 0.5	V	XTAL1 to V_{SS}
V_{PD}	Power down voltage to RST/VPD	4.5	5.5	V	V_{CC} = 0 V
V_{OL}	Output low voltage Ports 1, 2, 3	–	0.45	V	I_{OL} = 1.6 mA
V_{OL1}	Output low voltage Port 0, ALE, $\overline{PSEN}$	–	0.45	V	I_{OL} = 3.2 mA
V_{OH}	Output high voltage Ports 1, 2, 3	2.4	–	V	I_{OH} = – 80 µA
V_{OH1}	Output high voltage Port 0, ALE, $\overline{PSEN}$	2.4	–	V	I_{OH} = – 400 µA

DC Characteristics (cont'd)

Symbol	Parameter	Limit Values		Unit	Test Condition
		min.	max.		
I_{IL}	Logical 0 input current Ports 1, 2, 3	–	– 500	µA	V_{IL} = 0.45 V
I_{IL2}	Logical 0 input current XTAL 2 SAB 8051A/8031A-12/16/20 SAB 8051A/8031A-T40/85	 – –	 – 3.2 – 2.5	 mA mA	 XTAL1 = V_{SS} V_{IL} = 0.45 V
I_{IH1}	Input high current to RST/V_{PD} for reset	–	500	µA	V_{IN} = V_{CC} – 1.5 V
I_{LI}	Input leakage current to port 0, $\overline{EA}$	–	± 10	µA	0 V < V_{IN} < V_{CC}
I_{CC}	Power supply current SAB 8051A/8031A SAB 8051A/8031A-16 SAB 8051A/8031A-20	 – – –	 125 140 140	 mA mA mA	All outputs disconnected
I_{PD}	Power down current SAB 8051A/8031A-12/16/20 SAB 8051A/8031A-T40/85	 – –	 10 15	 mA mA	 V_{CC} = 0 V V_{PD} = 4.5 ... 5.5 V
C_{IO}	Capacitance of I/O buffer	–	10	pF	f_c = 1 MHz

4

AC Characteristics for SAB 8051A/8031A

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and PSEN outputs = 100 pF; C_L for all other outputs = 80 pF)
T_A = 0 to 70 °C for the SAB 8051A/8031A
T_A = – 40 to 85 °C for the SAB 8051A-T3/8031A-T40/85

Program Memory Characteristics

Symbol	Parameter	Limit Values				Unit
		Clock 12 MHz clock		Variable clock 1/t_{CLCL} = 1.2 MHz to 12 MHz		
		min.	max.	min.	max.	
t_{LHLL}	ALE pulse width	127	–	$2t_{CLCL}-40$	–	ns
t_{AVLL}	Address setup to ALE	53	–	$t_{CLCL}-30$	–	ns
t_{LLAX1}	Address hold after ALE	48	–	$t_{CLCL}-35$	–	ns
t_{LLIV}	ALE to valid instruction in	–	233	–	$4t_{CLCL}-100$	ns
t_{LLPL}	ALE to $\overline{PSEN}$	58	–	$t_{CLCL}-25$	–	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	215	–	$3t_{CLCL}-35$	–	ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in	–	150	–	$3t_{CLCL}-100$	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXIZ}*)	Input instruction float after $\overline{PSEN}$	–	63	–	$t_{CLCL}-20$	ns
t_{PXAV}*)	Address valid after $\overline{PSEN}$	75	–	$t_{CLCL}-8$	–	ns
t_{AVIV}	Address to valid instruction in	–	302	–	$5t_{CLCL}-115$	ns
t_{AZPL}	Address float to $\overline{PSEN}$	0	–	0	–	ns

*) Interfacing the SAB8051A to devices with float times up to 75 ns is permissible. The limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

t_{RLRH}	RD pulse width	400	–	$6t_{CLCL}-100$	–	ns
t_{WLWH}	$\overline{WR}$ pulse width	400	–	$6t_{CLCL}-100$	–	ns
t_{LLAX2}	Address hold after ALE	132	–	$2t_{CLCL}-35$	–	ns
t_{RLDV}	$\overline{RD}$ to valid data in	–	252	–	$5t_{CLCL}-165$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	Data float after $\overline{RD}$	–	97	–	$2t_{CLCL}-70$	ns
t_{LLDV}	ALE to valid data in	–	517	–	$8t_{CLCL}-150$	ns
t_{AVDV}	Address to valid data in	–	585	–	$9t_{CLCL}-165$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$	203	–	$4t_{CLCL}-130$	–	ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ high to ALE high	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	33	–	$t_{CLCL}-50$	–	ns
t_{QVWH}	Data setup before $\overline{WR}$	433	–	$7t_{CLCL}-150$	–	ns
t_{WHQX}	Data hold after $\overline{WR}$	33	–	$t_{CLCL}-50$	–	ns
t_{RLAZ}	Address float after $\overline{RD}$	–	0	–	0	ns

External Clock Drive XTAL2

t_{CLCL}	Oscillator period	–	–	83.3	833.3	ns
t_{CHCX}	High time	–	–	20	$t_{CLCL}-t_{CLCX}$	ns
t_{CLCX}	Low time	–	–	20	$t_{CLCL}-t_{CHCX}$	ns
t_{CLCH}	Rise time	–	–	–	20	ns
t_{CHCL}	Fall time	–	–	–	20	ns

4

AC Characteristics for SAB 8051A/8031A-16

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and PSEN outputs = 100 pF; C_L for all other outputs = 80 pF)
T_A = 0 to + 70 °C for SAB 8051 A/8031A-16

Program Memory Characteristics

Symbol	Parameter	Limit Values				Unit
		Clock 16 MHz clock		Variable clock 1/t_{CLCL} = 1.2 MHz to 16 MHz		
		min.	max.	min.	max.	
t_{LHLL}	ALE pulse width	85	–	$2t_{CLCL}$–40	–	ns
t_{AVLL}	Address setup to ALE	33	–	t_{CLCL}–30	–	ns
t_{LLAX1}	Address hold after ALE	28	–	t_{CLCL}–35	–	ns
t_{LLIV}	ALE to valid instruction in	–	150	–	$4t_{CLCL}$–100	ns
t_{LLPL}	ALE to $\overline{PSEN}$	38	–	t_{CLCL}–25	–	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	153	–	$3t_{CLCL}$–35	–	ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in	–	88	–	$3t_{CLCL}$–100	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXIZ}*)	Input instruction float after $\overline{PSEN}$	–	48	–	t_{CLCL}–15	ns
t_{PXAV}*)	Address valid after $\overline{PSEN}$	60	–	t_{CLCL}–3	–	ns
t_{AVIV}	Address to valid instruction in	–	223	–	$5t_{CLCL}$–90	ns
t_{AZPL}	Address float to $\overline{PSEN}$	0	–	0	–	ns

*) Interfacing the SAB 8051A-16 to devices with float times up to 55 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics for SAB 8051A/8031A-16 (cont'd)

External Data Memory Characteristics

Symbol	Parameter	Limit Values				Unit
		Clock 16 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 16 MHz		
		min.	max.	min.	max.	
t_{RLRH}	$\overline{RD}$ pulse width	275	–	$6t_{CLCL}-100$	–	ns
t_{WLWH}	$\overline{WR}$ pulse width	275	–	$6t_{CLCL}-100$	–	ns
t_{LLAX2}	Address hold after ALE	90	–	$2t_{CLCL}-35$	–	ns
t_{RLDV}	$\overline{RD}$ to valid data in	–	148	–	$5t_{CLCL}-165$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	Data float after $\overline{RD}$	–	55	–	$2t_{CLCL}-70$	ns
t_{LLDV}	ALE to valid data in	–	350	–	$8t_{CLCL}-150$	ns
t_{AVDV}	Address to valid data in	–	398	–	$9t_{CLCL}-165$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$	120	–	$4t_{CLCL}-130$	–	ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	13	–	$t_{CLCL}-50$	–	ns
t_{QVWH}	Data setup before $\overline{WR}$	288	–	$7t_{CLCL}-150$	–	ns
t_{WHQX}	Data hold after $\overline{WR}$	13	–	$t_{CLCL}-50$	–	ns
t_{RLAZ}	Address float after $\overline{RD}$	–	0	–	0	ns

External Clock Drive XTAL2

t_{CLCL}	Oscillator period	–	–	62.5	833.3	ns
t_{CHCX}	High time	–	–	15	$t_{CLCL}-t_{CLCX}$	ns
t_{CLCX}	Low time	–	–	15	$t_{CLCL}-t_{CHCX}$	ns
t_{CLCH}	Rise time	–	–	–	15	ns
t_{CHCL}	Fall time	–	–	–	15	ns

AC Characteristics for SAB 8051A/8031A-20

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)
T_A = 0 to + 70 °C for SAB 8051A/8031A-20

Program Memory Characteristics

Symbol	Parameter	Clock 20 MHz clock		Variable clock 1/t_{CLCL} = 1.2 MHz to 20 MHz		Unit
		min.	max.	min.	max.	
t_{LHLL}	ALE pulse width	60	–	$2t_{CLCL}$–40	–	ns
t_{AVLL}	Address setup to ALE	20	–	t_{CLCL}–30	–	ns
t_{LLAX1}	Address hold after ALE	20	–	t_{CLCL}–35	–	ns
t_{LLIV}	ALE to valid instruction in	–	100	–	$4t_{CLCL}$–100	ns
t_{LLPL}	ALE to $\overline{PSEN}$	25	–	t_{CLCL}–25	–	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	115	–	$3t_{CLCL}$–35	–	ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in	–	75	–	$3t_{CLCL}$–75	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXAV}*)	Address valid after $\overline{PSEN}$	47	–	t_{CLCL}–3	–	ns
t_{PXIZ}*)	Input instruction float after $\overline{PSEN}$	–	40	–	t_{CLCL}–10	ns
t_{AVIV}	Address to valid instruction in	–	175	–	$5t_{CLCL}$–75	ns
t_{AZPL}	Address float to $\overline{PSEN}$	0	–	0	–	ns

*) Interfacing the SAB 8051A-20 to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

Symbol	Parameter	Clock 20 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 20 MHz		Unit
		min.	max.	min.	max.	
t_{RLRH}	$\overline{RD}$ pulse width	200	–	$6t_{CLCL}-100$	–	ns
t_{WLWH}	$\overline{WR}$ pulse width	200	–	$6t_{CLCL}-100$	–	ns
t_{LLAX2}	Address hold after ALE	70	–	$2t_{CLCL}-30$	–	ns
t_{RLDV}	$\overline{RD}$ to valid data in	–	100	–	$5t_{CLCL}-150$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	Data float after $\overline{RD}$	–	40	–	$2t_{CLCL}-60$	ns
t_{LLDV}	ALE to valid data in	–	250	–	$8t_{CLCL}-150$	ns
t_{AVDV}	Address to valid data in	–	285	–	$9t_{CLCL}-165$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	100	200	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$	70	–	$4t_{CLCL}-130$	–	ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ high to ALE high	20	80	$t_{CLCL}-30$	$t_{CLCL}+30$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	5	–	$t_{CLCL}-45$	–	ns
t_{QVWH}	Data setup before $\overline{WR}$	200	–	$7t_{CLCL}-150$	–	ns
t_{WHQX}	Data hold after $\overline{WR}$	10	–	$t_{CLCL}-40$	–	ns
t_{RLAZ}	Address float after RD	–	0	–	0	ns

External Clock Drive

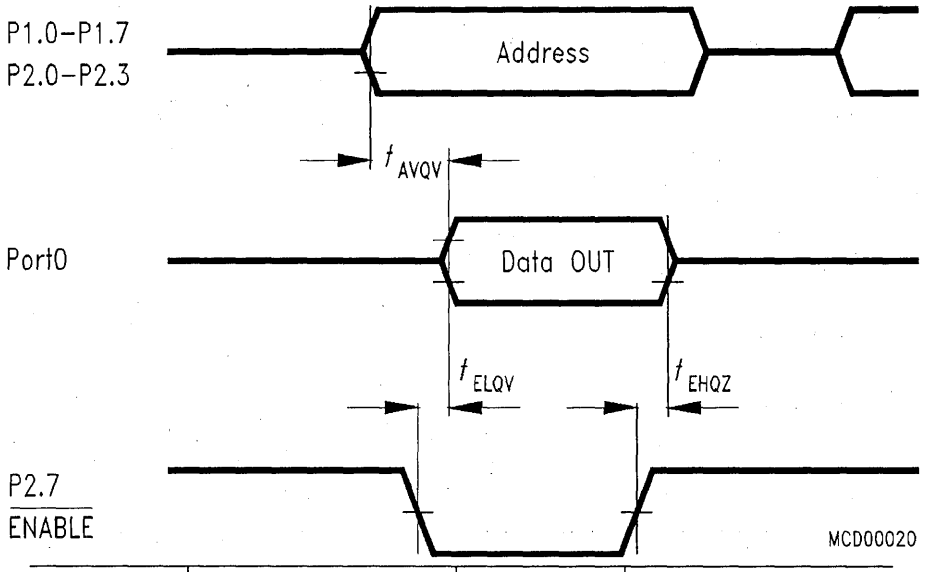
Symbol	Parameter	Clock 20 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 20 MHz		Unit
		min.	max.	min.	max.	
t_{CLCL}	Oscillator period	–	–	50	833.3	ns
t_{CHCX}	High time	–	–	15	$t_{CLCL}-t_{CLCX}$	ns
t_{CLCX}	Low time	–	–	15	$t_{CLCL}-t_{CHCX}$	ns
t_{CLCH}	Rise time	–	–	–	15	ns
t_{CHCL}	Fall time	–	–	–	15	ns

ROM Verification Characteristics for SAB 8051A/8031A Family

T_A = 25 °C ± 5 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V

Symbol	Parameter	Limit Values		Unit
		min.	max.	
t_{AVQV}	Address to valid data	–	48 t_{CLCL}	ns
t_{ELQV}	$\overline{\text{ENABLE}}$ to valid data	–	48 t_{CLCL}	ns
t_{EHQZ}	Data float after $\overline{\text{ENABLE}}$	0	48 t_{CLCL}	ns
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz

Figure 5
ROM Verification



Microcontroller	Address	Data	Inputs
SAB 8051A 4K x 8	P1.0 – P1.7 = A0 – A7 P2.0 – P2.3 = A8 – A11	Port 0= D0 – D7	P2.4 – P2.6, PSEN= V_{SS} ALE, EA = V_{IH} R_{ST}/V_{PD} = V_{IH1}

Waveforms

Figure 6
Program Memory Read Cycle

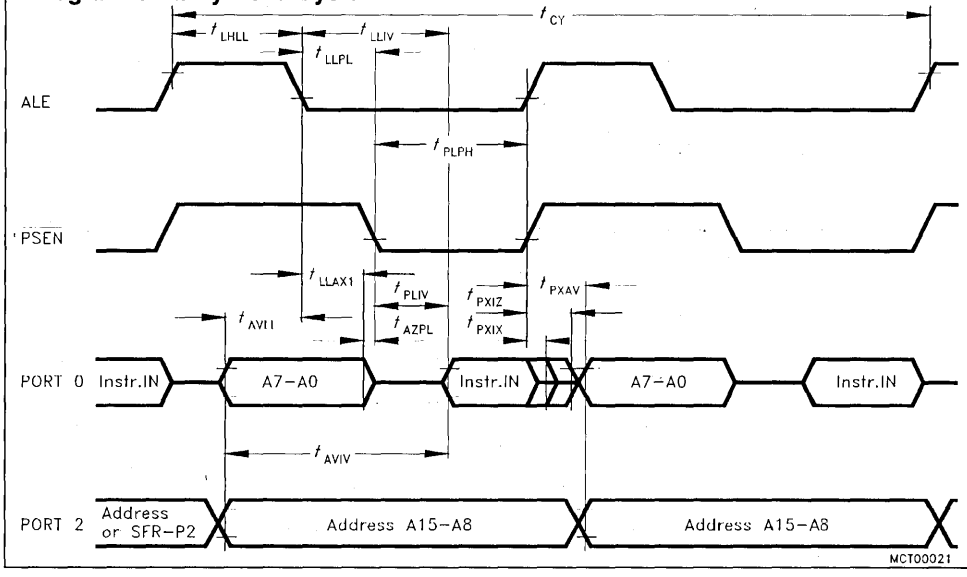
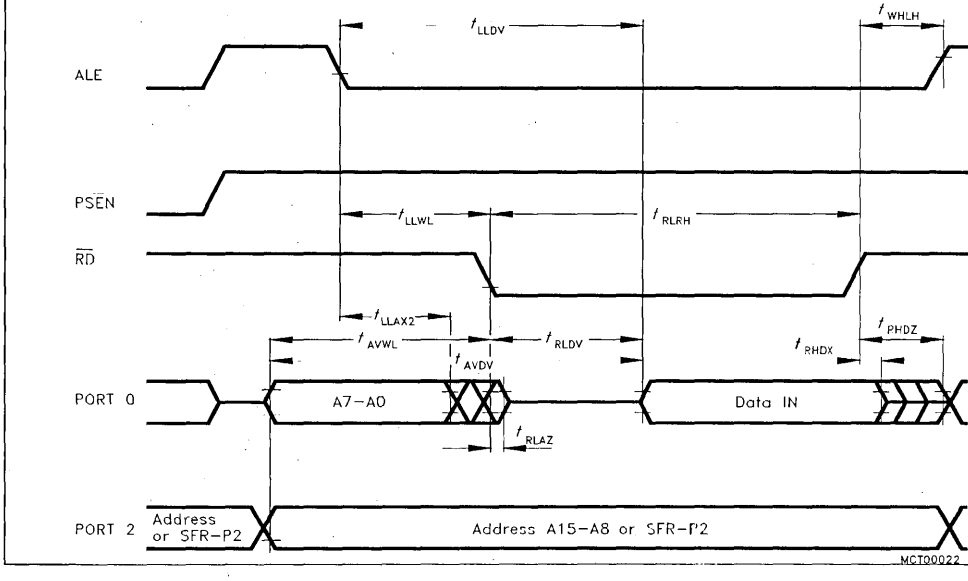


Figure 7
Data Memory Read Cycle



4

Figure 8
Data Memory Write Cycle

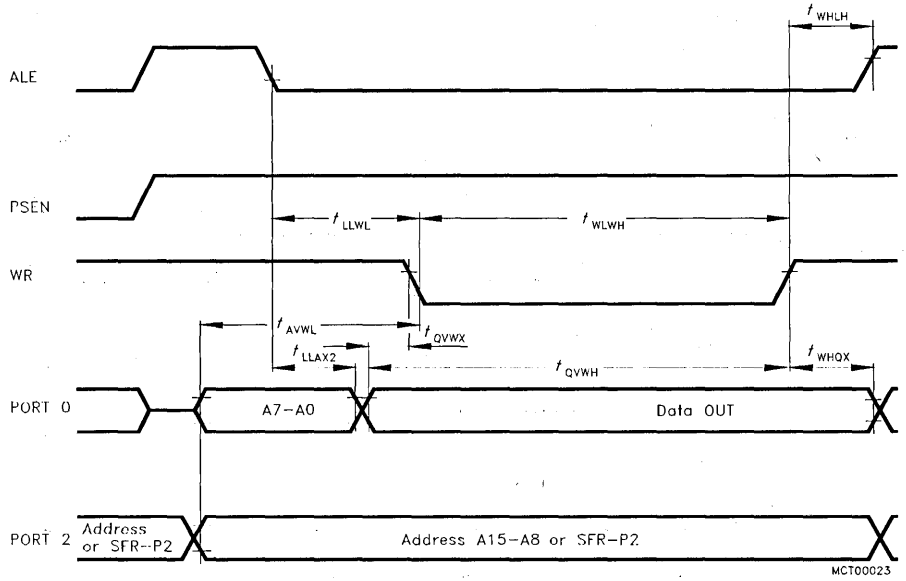
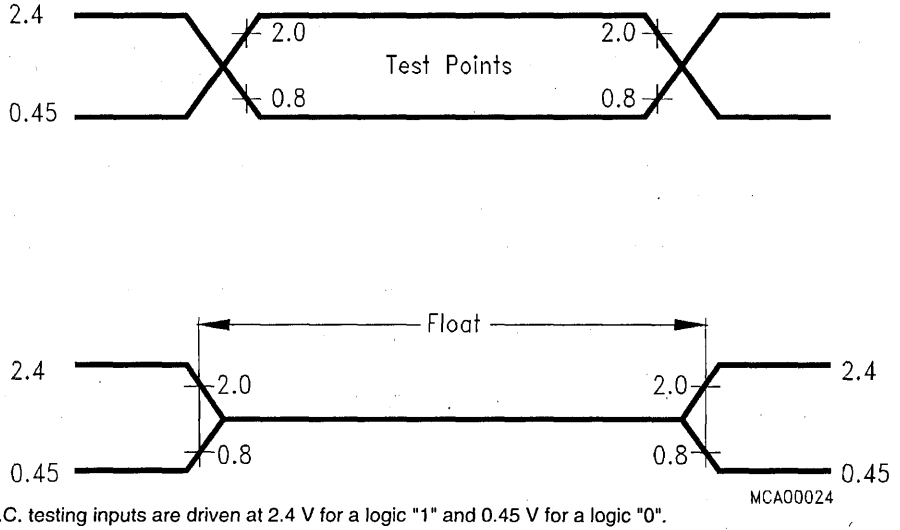
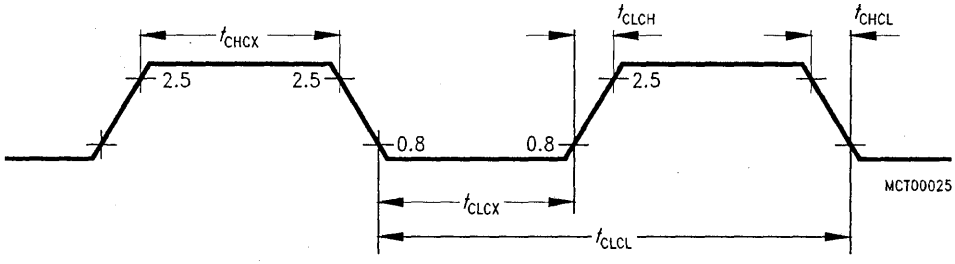


Figure 9
AC Testing Input, Output, Float Waveforms



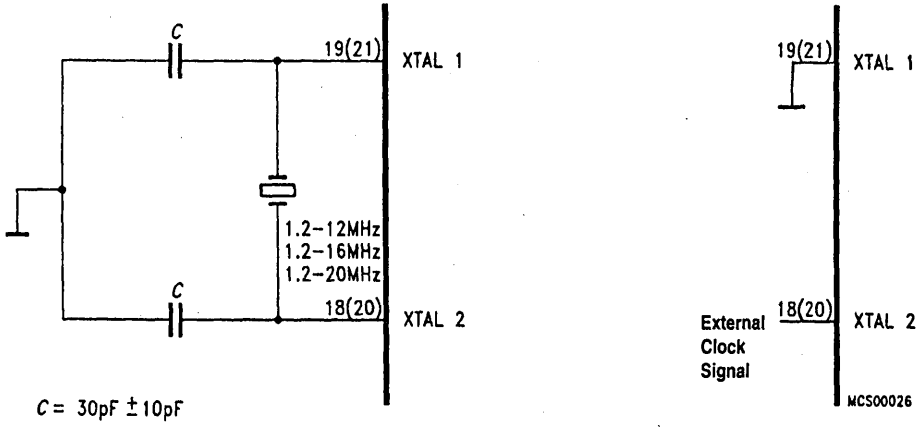
A.C. testing inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0".
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0".
For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2 mA or sources 400 µA at the voltage test levels.

Figure 10
External Clock Cycle



4

Figure 11
Recommended Oscillator Circuits



Pin number in (. . .) are for PL-CC-44 package

8-Bit Single-Chip Microcontroller

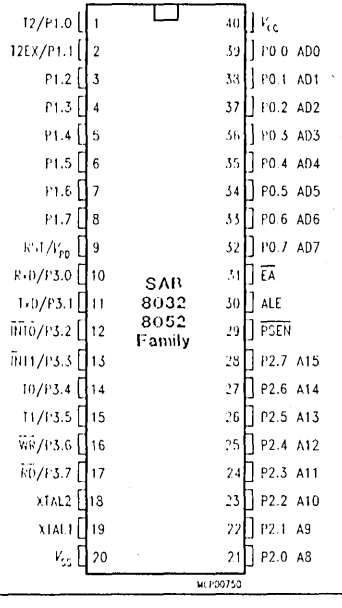
SAB 8052B Microcontroller with factory mask programmable ROM (8K)

SAB 8032B Microcontroller for external ROM

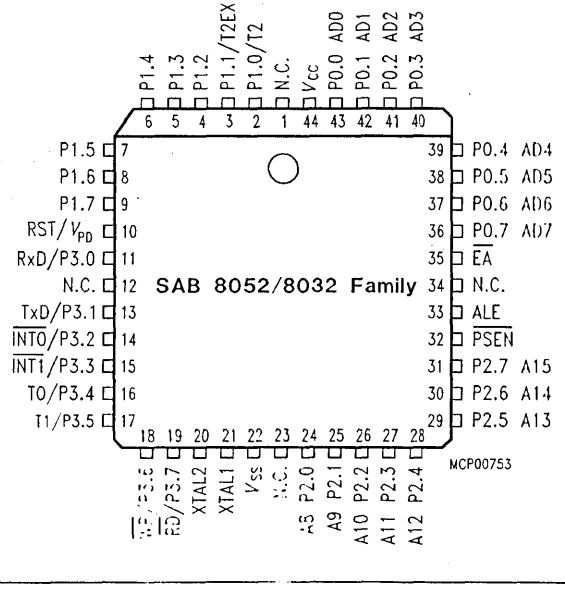
- Versions for 12 MHz / 16 MHz / 20 MHz operating frequency
- 8 K × 8 ROM (SAB 8052B only)
- 256 × 8 RAM
- Four 8-bit ports, 32 I/O lines
- Three 16-bit timer/event counters
- High-performance full-duplex serial channel with flexible transmit/receive baud rate capability
- External memory expandable up to 128 Kbytes
- Boolean processor
- Most instructions execute in:
  1 µs instruction cycle time at 12 MHz
  750 ns instruction cycle time at 16 MHz
  600 ns instruction cycle time at 20 MHz
- Multiply and divide in 4 µs/3 µs/2.4 µs
- Six interrupt vectors, two priority levels
- RAM power-down supply
- Packages P-DIP-40 and PL-CC-44
- Full backward compatibility with SAB 8051/8031
- Three temperature ranges available
  0 to 70 °C
  –40 to 85 °C
  –40 to 110 °C

4

Figure 12
Pin Configuration
P-DIP-40



PL-CC-44



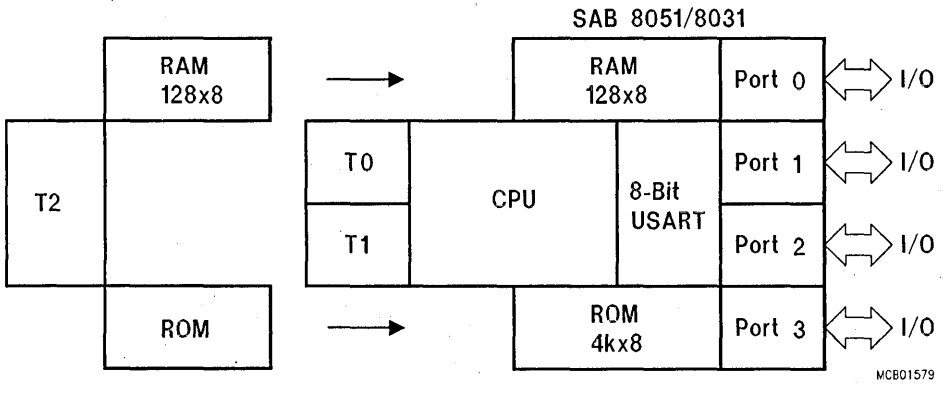
The SAB 8052/8032 family are standalone, high-performance single-chip microcontrollers fabricated in +5V advanced N-channel, silicon-gate Siemens MYMOS technology, packaged in a 40-pin plastic dual-in-line package (P-DIP-40) or 44-pin plastic leaded chip carrier (PL-CC-44) package. It is backwardly compatible with the SAB 8051A/8031A and provides the hardware features, architectural enhancements, and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64 Kbytes of program memory and/or up to 64 Kbytes of data memory.

The controllers of the SAB 8052 / 8032 family contain a non-volatile 8 K × 8 read-only program memory, a volatile 256 × 8 read/write data memory, 32 I/ O lines, three 16-bit timer/counters, a six-source, two-priority-level nested interrupt structure, a serial I/O port for either multiprocessor communications, I/O expansion, or full-duplex UART, as well as an on-chip oscillator and clock circuits.

For systems that require extra capability, the standard TTL compatible memories and the byte-oriented SAB 8080 and SAB 8085 peripherals can be used to expand the SAB 8052 / 8032 family.

The parts are available for standard temperature range (0 to 70 °C) and extended temperature ranges (– 40 to 85 °C and – 40 to 110 °C).

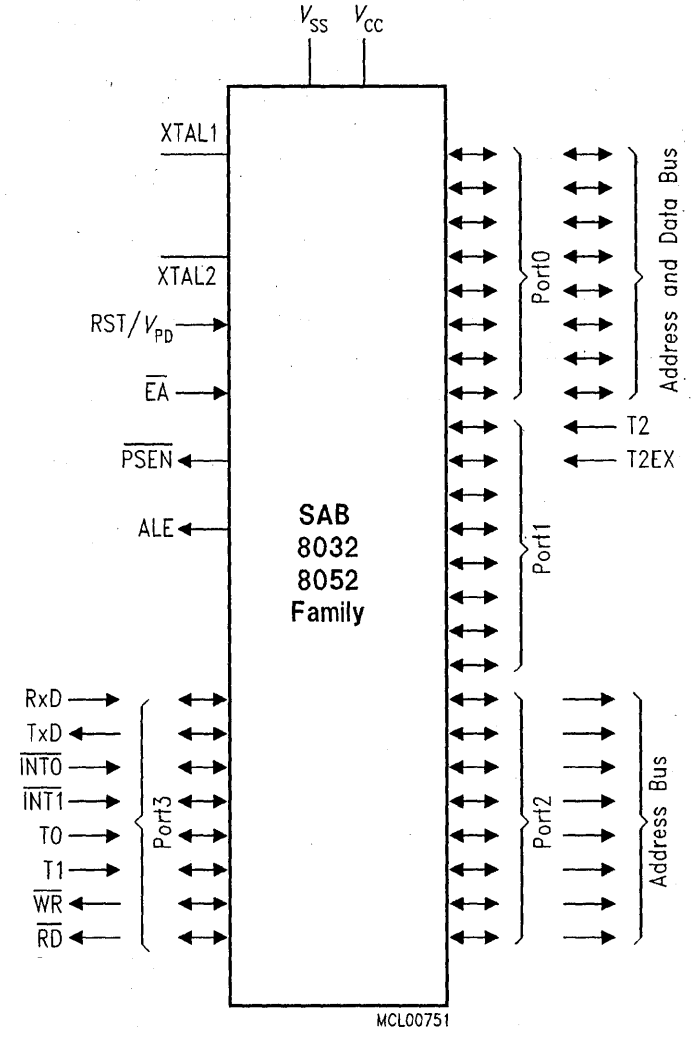
Figure 13
SAB 8052 / 8032 family



Ordering Information

Type	Package	Description (8-bit single- microcontroller)
SAB 8032B-P	P-DIP-40	for external memory, 12 MHz.
SAB 8032B-N	PL-CC-44	
SAB 8032B-16-P	P-DIP-40	for external memory, 16 MHz
SAB 8032B-16-N	PL-CC-44	
SAB 8032B-20-P	P-DIP-4	for external memory, 20 MHz
SAB 8032B-20-N	PL-CC-44	
SAB 8052B-P	P-DIP-40	with 8-KByte mask-programmable ROM, 12 MHz
SAB 8052B-N	PL-CC-44	
SAB 8052B-16-P	P-DIP-40	with 8-KByte mask-programmable ROM, 16 MHz
SAB 8052B-16-N	PL-CC-44	

Figure 14
Logic Symbol



Pin Definitions and Functions

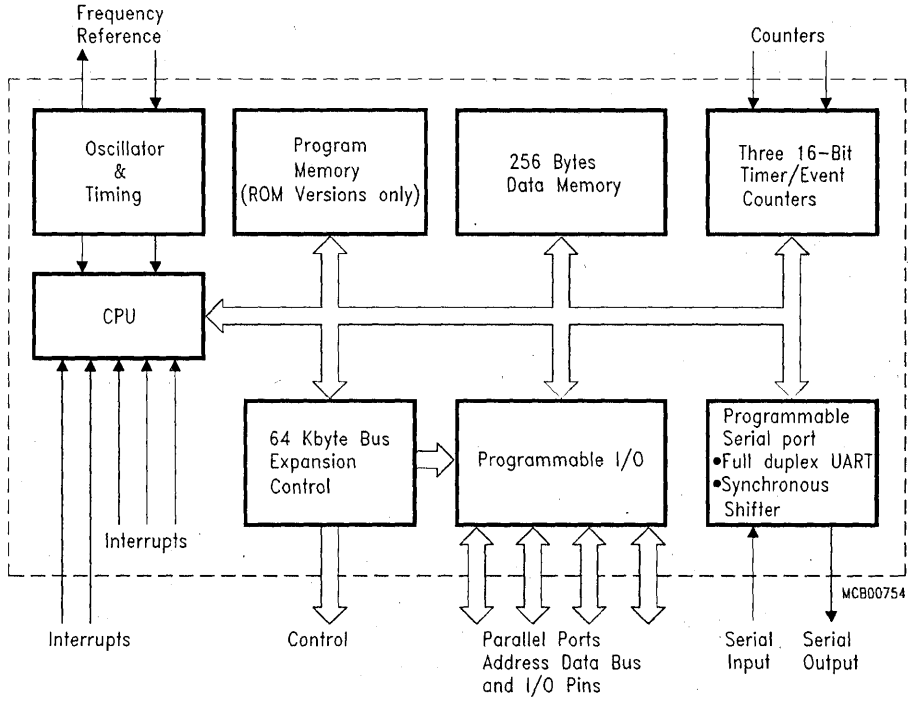
Symbol	Pins		Input (I) Output (O)	Function
	P-DIP-40	PL-CC-44		
P1.0-P1.7	1-8	2-9	I/O	**PORT 1** is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads. Port 1 also contains the timer 2 pins as a secondary function. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 1, as follows: – T2 (P1.0). Input to counter 2. – T2 (EX (P1.1). Capture/Reload trigger of timer 2.
RST/V_{PD}	9	10	I	**RESET** input. A high level on this pin resets the SAB 8052B. A small internal pulldown resistor permits power-on reset using only a capacitor connected to V_{CC}. If V_{PD} is held within its spec while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
P3.0-P3.7	10-17	11 13-19	I/O	**PORT 3** is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of port 3, as follows: – RxD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). – TxD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). – $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. – $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. – T0 (P3.4). Input to counter 0. – T1 (P3.5). Input to counter 1. – $\overline{WR}$ (P3.6). The write control signal latches the data byte from port 0 into the external data memory. – $\overline{RD}$ (P3.7). The read control signal enables external data memory to port 0.
XTAL1 XTAL2	19 18	21 20		**XTAL 1** input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on XTAL 2. **XTAL 2** output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.
P2.0-P2.7	21-28	24-31	I/O	**PORT 2** is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.

4

Pin Definitions and Functions (continued)

Symbol	Pins		Input (I) Output (O)	Function
	P-DIP-40	PL-CC-44		
$\overline{PSEN}$	29	32	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
ALE	30	33	O	Provides **Address Latch Enable** output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	31	35	I	**External Access** enable. When held at a TTL high level, the ROM-versions executes instructions from the internal ROM when the PC points to the internal ROM address space. When held at a TTL low level, the ROM-versions fetch all instructions from external program memory. For the ROM-less versions this pin must be tied low.
P0.0-P0.7	39-32	43-36	I/O	**Port 0** is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
V_{CC}	40	44	–	+ 5 V **Power Supply** during operation and program verification.
V_{SS}	20	22	–	Circuit **Ground** potential
NC	–	1,12, 23,34	–	**No Connection**

Figure 15
Block Diagram



4

Absolute Maximum Ratings

Ambient temperature under bias
SAB 8052B/8032B ..... 0 to +70 °C
Storage temperature.....– 65 to + 150 °C
Voltage on any pin with
respect to ground (V_{SS}) .....– 0.5 to + 7 V
Power dissipation .....2 W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB 8052B/8032B

Symbol	Parameter	Limit Values		Unit	Test Conditions
		min.	max.		
V_{IL}	Input low voltage	– 0.5	0.8	V	–
V_{IH}	Input high voltage (except RST/V_{PD} and XTAL 2)	2.0	V_{CC} + 0.5	V	–
V_{IH1}	Input high voltage to RST/V_{PD} for reset, XTAL 2	2.5	V_{CC} + 0.5	V	XTAL1 to V_{SS}
V_{PD}	Power down voltage to RST/V_{PD}	4.5	5.5	V	V_{CC} = 0 V
V_{OL}	Output low voltage Ports 1, 2, 3	–	0.45	V	I_{OL} = 1.6 mA
V_{OL1}	Output low voltage Port 0, ALE, $\overline{PSEN}$	–	0.45	V	I_{OL} = 3.2 mA
V_{OH}	Output high voltage Ports 1, 2, 3	2.4	–	V	I_{OH} = – 80 µA
V_{OH1}	Output high voltage Port 0, ALE, $\overline{PSEN}$	2.4	–	V	I_{OH} = – 400 µA

DC Characteristics (cont'd)

Symbol	Parameter	Limit Values		Unit	Test Conditions
		min.	max.		
I_{IL}	Logical 0 input current Ports 1, 2, 3	–	– 500	µA	V_{IL} = 0.45 V
I_{IL2}	Logical 0 input current XTAL 2 SAB 8052B/8032B - 12/16/20	–	– 3.2	mA	XTAL1 = V_{SS} V_{IL} = 0.45 V
I_{IH1}	Input high current to RST/V_{PD} for reset	–	500	µA	V_{IN} = V_{CC} – 1.5 V
I_{LI}	Input leakage current to port 0, $\overline{EA}$	–	± 10	µA	0 V < V_{IN} < V_{CC}
I_{CC}	Power supply current SAB 8052B/8032B SAB 8052B-16/8032B-16 SAB 8052B-20/8032B-20	 – – –	 175 175 175	 mA mA mA	All outputs disconnected
I_{PD}	Power down current	–	15	mA	V_{CC} = 0 V; V_{PD} = 4.5 ... 5.5 V
C_{IO}	Capacitance of I/O buffer	–	10	pF	f_C = 1 MHz

4

AC Characteristics for SAB 8052B/8032B, 12 MHz

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)
T_A = 0 to + 70 °C for SAB 8052B/8032B

Please refer to AC characteristics for SAB 8051A/8031A, 12MHz

AC Characteristics for SAB 8052B/8032B, 16 MHz

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and PSEN outputs = 100 pF; C_L for all other outputs = 80 pF)
T_A = 0 to 70 °C; for SAB 8052B/8032B-16

Please refer for AC characteristics to the SAB 8051A/8031A-16, 16MHz

AC Characteristics for SAB 8032B-20, 20 MHz

T_A = 0 to 70 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V
(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Symbol	Parameter	Limit Values				Unit
		Clock 20 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 20 MHz		
		min.	max.	min.	max.	

Program Memory Characteristics

t_{LHLL}	ALE pulse width	60	–	$2t_{CLCL}$–40	–	ns
t_{AVLL}	Address setup to ALE	20	–	t_{CLCL}–30	–	ns
t_{LLAX1}	Address hold after ALE	20	–	t_{CLCL}–30	–	ns
t_{LLIV}	ALE to valid instruction in	–	100	–	$4t_{CLCL}$–100	ns
t_{LLPL}	ALE to $\overline{PSEN}$	25	–	t_{CLCL}–25	–	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	115	–	$3t_{CLCL}$–35	–	ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in	–	75	–	$3t_{CLCL}$–75	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXIZ}*)	Input instruction float after $\overline{PSEN}$	–	40	–	t_{CLCL}–10	ns
t_{PXAV}*)	Address valid after $\overline{PSEN}$	47	–	t_{CLCL}–3	–	ns
t_{AVIV}	Address to valid instruction in	–	190	–	$5t_{CLCL}$–60	ns
t_{AZPL}	Address float to $\overline{PSEN}$	0	–	0	–	ns

*) Interfacing the SAB 8032B-20 to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics for SAB 8032B-20, 20 MHz (cont'd)

Symbol	Parameter	Limit Values				Unit
		Clock 20 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 20 MHz		
		min.	max.	min.	max.	

External Data Memory Characteristics

t_{RLRH}	$\overline{RD}$ pulse width	200	–	$6t_{CLCL}-100$	–	ns
t_{WLWH}	$\overline{WR}$ pulse width	200	–	$6t_{CLCL}-100$	–	ns
t_{LLAX2}	Address hold after ALE	70	–	$2t_{CLCL}-30$	–	ns
t_{RLDV}	$\overline{RD}$ to valid data in	–	100	–	$5t_{CLCL}-150$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	Data float after $\overline{RD}$	–	40	–	$2t_{CLCL}-60$	ns
t_{LLDV}	ALE to valid data in	–	250	–	$8t_{CLCL}-150$	ns
t_{AVDV}	Address to valid data in	–	285	–	$9t_{CLCL}-165$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	100	200	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$	70	–	$4t_{CLCL}-130$	–	ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ high to ALE high	20	80	$t_{CLCL}-30$	$t_{CLCL}+30$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	5	–	$t_{CLCL}-45$	–	ns
t_{QVWH}	Data setup before $\overline{WR}$	200	–	$7t_{CLCL}-150$	–	ns
t_{WHQX}	Data hold after $\overline{WR}$	10	–	$t_{CLCL}-40$	–	ns
t_{RLAZ}	Address float after $\overline{RD}$	–	0	–	0	ns

External Clock Drive XTAL2

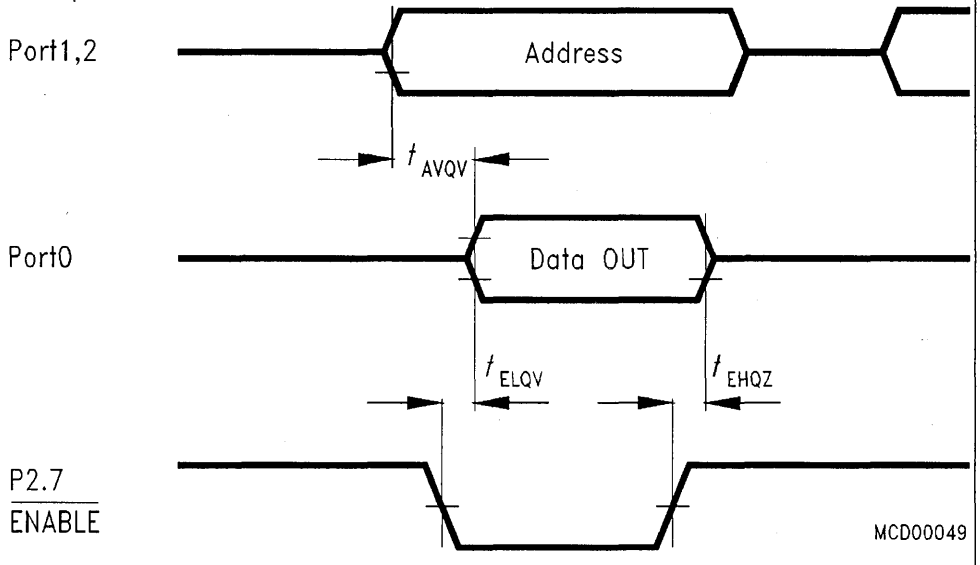
t_{CLCL}	Oscillator period	–	–	50	833.3	ns
t_{CHCX}	High time	–	–	15	$t_{CLCL}-t_{CLCX}$	ns
t_{CLCX}	Low time	–	–	15	$t_{CLCL}-t_{CHCX}$	ns
t_{CLCH}	Rise time	–	–	–	15	ns
t_{CHCL}	Fall time	–	–	–	15	ns

ROM Verification Characteristics for SAB 8052B/8032B Family

T_A = 25 °C ± 5 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V

Symbol	Parameter	Limit Values		Unit
		min.	max.	
t_{AVQV}	Address to valid data	–	48 t_{CLCL}	ns
t_{ELQV}	$\overline{\text{ENABLE}}$ to valid data	–	48 t_{CLCL}	ns
t_{EHQZ}	Data float after $\overline{\text{ENABLE}}$	0	48 t_{CLCL}	ns
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz

Figure 16
ROM Verification



Microcontroller	Address	Data	Inputs	
SAB 8052B 8K × 8	P1.0 – P1.7 = A0 – A7 P2.0 – P2.4 = A8 – A12	Port 0 = D0 – D7	P2.5 – P2.6, $\overline{\text{PSEN}}$ ALE, $\overline{\text{EA}}$ R_{ST}/V_{PD}	= V_{SS} = V_{IH} = V_{IH1}

Waveforms

Please refer to SAB 8051A/8031A for AC waveforms.

8-Bit Single Chip Microcontroller

SAB 80515 Microcontroller with factory mask-programmable ROM

SAB 80535 Microcontroller for external ROM

- 8K × 8 ROM (SAB 80515 only)
- 256 × 8 RAM
- Six 8-bit I/O ports, one 8-bit input port for analog signals
- Three 16-bit timer/counters
- Highly flexible reload, capture, compare capabilities
- Full-duplex serial channel
- Twelve interrupt vectors, four priority levels
- 8-bit A/D converter with eight multiplexed inputs and programmable internal reference voltages
- 16-bit watchdog timer
- V_{PD} provides standby current for 40 bytes of RAM
- Boolean processor
- 256 bit-addressable locations
- Most instructions execute in 1 µs
- 4 µs multiply and divide
- External memory expandable up to 128 Kbyte
- Backwardly compatible with SAB 8051
- 68-pin plastic leaded chip carrier package (PL-CC-68)
- Three temperature ranges available:
  0 to 70°C
  -40 to 85°C (T40/85)
  -40 to 110°C (T40/110)

4

Figure 17
Logic Symbol

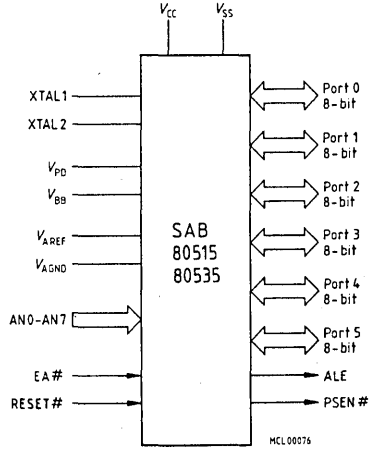
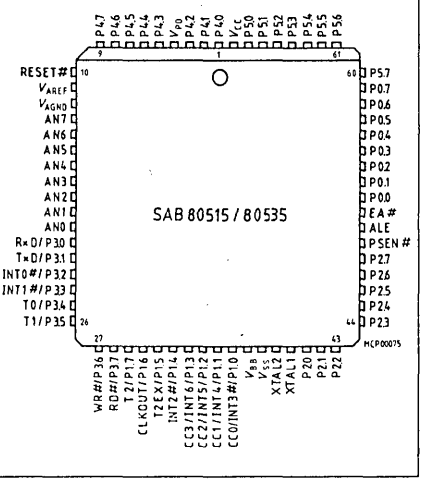


Figure 18
Pin Configuration (PL-CC-68)



The SAB 80515/80535 is a powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is fabricated in + 5 V N-channel, silicon-gate Siemens MYMOS technology. The SAB 80515/80535 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051 architecture. While maintaining all the SAB 8051 operating characteristics, the SAB 80515/80535 incorporates several enhancements which significantly increase design flexibility and overall system performance.
The SAB 80535 is identical with the SAB 80515 except that it lacks the on-chip program memory. The SAB 80515/80535 is supplied in a 68-pin plastic leaded chip carrier package (P-LCC-68).

Ordering Information

Type	Description
SAB 80515-N	8-bit single-chip microcontroller with mask-programmable ROM
SAB 80535-N	for external memory
SAB 80515-N-T40/85	with mask-programmable ROM, EXT. Temperature
SAB 80535-N-40/85	for external memory, EXT. Temperature
SAB 80515-N-T40/110	with mask-programmable ROM, EXT. Temperature
SAB 80535-N-T40/110	for external memory, EXT. Temperature

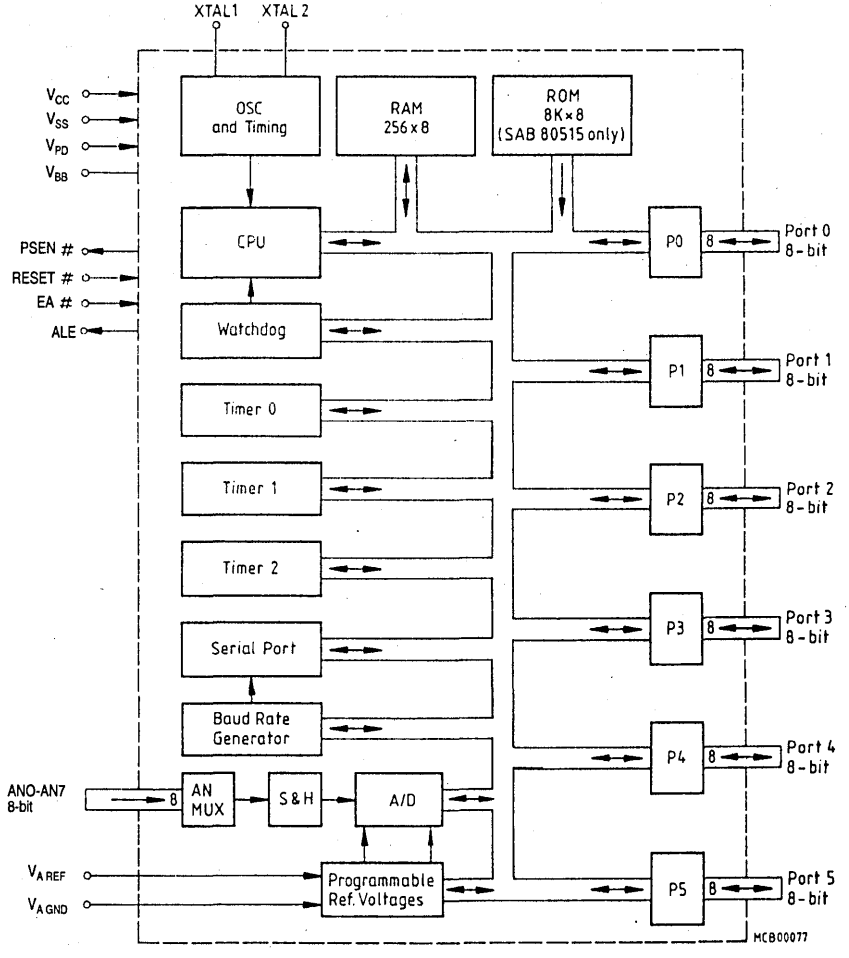
Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P4.0-P4.7	1-3, 5-9	I/O	Port 4 is an 8-bit quasi-bidirectional I/O port. Port 4 can sink/source 4 LS-TTL loads.
V_{PD}	4		Power down supply. If V_{PD} is held within its specs while V_{CC} drops below specs, V_{PD} will provide standby power to 40 byte of the internal RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}.
$\overline{\text{RESET}}$	10	I	A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80515. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
V_{AREF}	11		Reference voltage for the A/D converter
V_{AGND}	12		Reference ground for the A/D converter
AN7-AN0	13-20	I	Multiplexed analog inputs
P3.0-P3.7	21-28	I/O	Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source 4 LS-TTL loads. The secondary functions are assigned to the pins of port 3, as follows: – RxD (P3.0): serial port's receiver data input (asynchronous) or data input/output (synchronous) – TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) – $\overline{\text{INT0}}$ (P3.2): interrupt 0 input / timer 0 gate control input – $\overline{\text{INT1}}$ (P3.3): interrupt 1 input / timer 1 gate control input – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – $\overline{\text{WR}}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory – $\overline{\text{RD}}$ (P3.7): the read control signal enables the external data memory to port 0
P1.7-P1.0	29-36	I/O	Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. It also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch must be programmed to a one (1) for that function to operate (except when used for the compare functions). Port 1 can sink/source 4 LS-TTL loads. The secondary functions are assigned to the port 1 pins, as follows: – $\overline{\text{INT3}}$/CC0 (P1.0): interrupt 3 input/compare 0 output/ capture 0 input – INT4/CC1 (P1.1): interrupt 4 input/compare 1 output/ capture 1 input – INT5/CC2 (P1.2): interrupt 5 input/compare 2 output/ capture 2 input – INT6/CC3 (P1.3): interrupt 6 input/compare 3 output/ capture 3 input

Pin Definitions and Functions (cont'd)

Symbol	Pin	Input (I) Output (O)	Function
P1.7-P1.0 (cont'd)			– $\overline{INT2}$ (P1.4): interrupt 2 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input
V_{BB}	37		Substrate pin. Must be connected to V_{SS} through a capacitor (47 to 100 nF) for proper operation of the A/D converter.
XTAL2	39		XTAL2 is the output from the oscillator's amplifier.Input to the internal timing circuitry. A crystal, ceramic resonator, or external source can be used.
XTAL1	40		XTAL1 is the input to the oscillator's high gain amplifier. Required when a crystal or ceramic resonator is used. Connect to V_{SS} when external source is used on XTAL2.
P2.0-P2.7	41-48	I/O	Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source 4 LS-TTL loads.
$\overline{PSEN}$	49	O	The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
ALE	50	O	Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	51	I	When held at a TTL high level, the SAB 80515 executes instructions from the internal ROM when the PC is less than 8192. When held at a TTL low level, the SAB 80515 fetches all instructions from external program memory. For the SAB 80535 this pin must be tied low.
P0.0-P0.7	52-59	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source 8 LS-TTL loads.
P5.7-P5.0	60-67	I/O	Port 5 is an 8-bit quasi-bidirectional I/O port. Port 5 can sink/source 4 LS-TTL loads.
V_{CC}	68		POWER SUPPLY (+ 5 V power supply during normal operation and program verification)
V_{SS}	38		GROUND (0 V)

Figure 19
Block Diagram



4

Functional Description

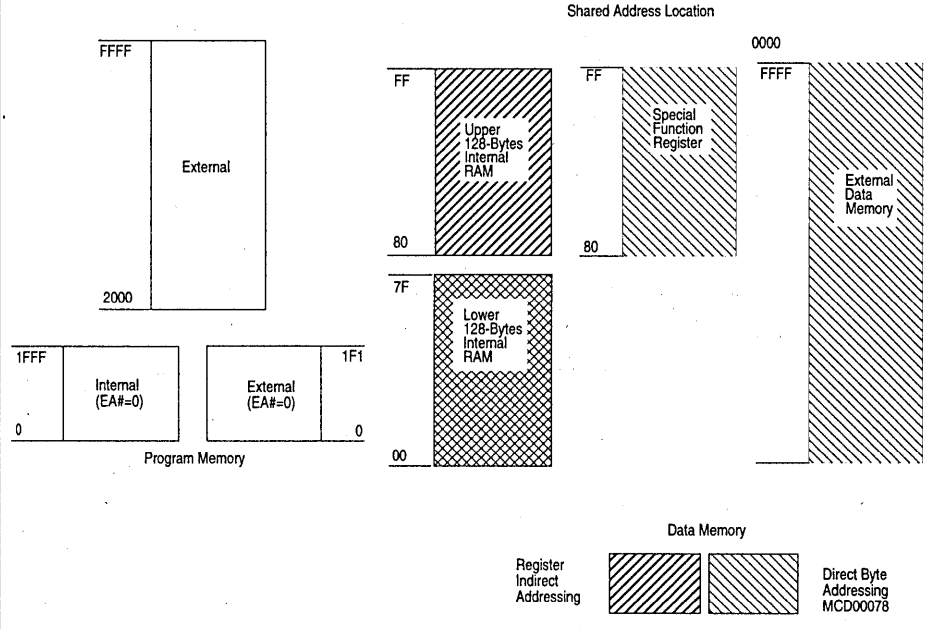
Principles of Architecture

The architecture of the SAB 80515 is based on the SAB 8051 microcontroller family. The following features of the SAB 80515 are fully compatible with the SAB 8051 features:

- Instruction set
- External memory expansion interface (port 0 and port 2)
- Full-duplex serial port
- Timer/counter 0 and 1
- Alternate functions on port 3
- The lower 128 bytes of internal RAM and the lower 4 Kbytes of internal ROM

The SAB 80515 additionally contains 128 bytes of internal RAM and 4 Kbytes of internal ROM, which results in a total of 256 bytes of RAM and 8 Kbytes of ROM on chip. The SAB 80515 has a new 16-bit timer/counter with a 2:1 prescaler, reload mode, compare and capture capability. It also contains a 16-bit watchdog timer, an 8-bit A/D converter with programmable reference voltages, two additional quasi-bidirectional 8-bit ports, one 8-bit input port for analog signals, and a programmable clock output ($f_{OSC}/12$).
Furthermore, the SAB 80515 has a powerful interrupt structure with 12 vectors and 4 programmable priority levels.
Figure 19 Block Diagram shows a block diagram of the SAB 80515.

Figure 20
Memory Address Spaces



Absolute Maximum Ratings

Ambient temperature under bias
SAB 80515/80535 ..... – 0 to +70°C
Storage temperature ..... – 65 to 150 °C
Voltage on any pin with respect
to ground (V_{SS}) .....– 0.5 to + 7 V
Power Dissipation .....2 W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_A = – 0 to + 70 °C; for SAB 80515/80535
T_A= – 40 to 85 °C; for SAB 80515/80535-T40/85
T_A= – 40 to 110 °C; for SAB 80515/80535-T40/110

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Input low voltage	V_{IL}	– 0.5	0.8	V	–
Input high voltage (except $\overline{RESET}$ and XTAL2)	V_{IH}	2.0	V_{CC} + 0.5	V	–
Input high voltage to XTAL2	V_{IH1}	2.5	V_{CC} + 0.5	V	XTAL 1 to V_{SS}
Input high voltage to $\overline{RESET}$	V_{IH2}	3.0	–	V	–
Power-down voltage	V_{PD}	3	5.5	V	V_{CC} = 0 V
Output low voltage, ports 1, 2, 3, 4, 5	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA [1]
Output low voltage port 0, ALE, $\overline{PSEN}$	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA [1]
Output high voltage, ports 1, 2, 3, 4, 5	V_{OH}	2.4	–	V	I_{OH} = – 80 µA
Output high voltage port 0, ALE, $\overline{PSEN}$	V_{OH1}	2.4	–	V	I_{OH} = – 400 µA
Logic 0 input current ports 1, 2, 3, 4, 5	I_{IL}	–	– 800	µA	V_{IL} = 0.45 V
Logic 0 input current XTAL2	I_{IL2}	–	– 2.5	mA	XTAL1 = V_{SS} V_{IL} = 0.45 V
Input low current to $\overline{RESET}$ for reset	I_{IL3}	–	– 500	µA	V_{IL} = 0.45 V
Input leakage current to port 0, $\overline{EA}$, AN0 – AN7	I_{LI}	–	± 10	µA	0 V < V_{IN} < V_{CC}
Power supply current SAB 80515/80535 SAB 80515/80535- T40/85 SAB 80515/80535 - T40/110	 I_{CC} I_{CC} I_{CC}	 – – –	 210 230 230	 mA mA mA	All outputs disconnected
Power- down current	I_{PD}	–	3	mA	V_{CC} = 0 V
Capacitance of I/O buffer	C_{IO}	–	10	pF	f_c = 1 MHz

[1] Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on tho VOL of ALE and ports 1, 3, 4, 5. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-0 transactions during bus operation.

A/D Converter Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; V_{AREF} = V_{CC} = 5.0 V; V_{AGND} = V_{SS} = 0.0 V; $V_{IntAREF} - V_{IntAGND} \geq 1$ V;

Parameter	Symbol	Limit Values			Unit	Test Conditions
		min.	typ.	max.		
Analog input voltage	V_{AINPUT}	V_{AGND}- 0.2	–	V_{AREF+} 0.2	V	–
Analog input capacitance	C_I	–	25	–	pF	2)
Load time	t_L	–	–	2 t_{CY}	µs	–
Sample time (incl. load time)	t_S	–	–	5 t_{CY}	µs	–
Conversion time (including sample time)	t_C	–	–	15 t_{CY}	µs	–
Differential non-linearity	DNLE	–	± 1/2	± 1	LSB	$V_{IntAREF}$ =
Integral non-linearity	INLE	–	± 1/2	± 1	LSB	V_{AREF} = V_{CC}
Offset error			± 1/2	± 1	LSB	$V_{IntAGND}$ =
Gain error			± 1/2	± 1	LSB	V_{AGND} = V_{SS}
Total unadjusted error	TUE		± 1	± 2	LSB	2)
V_{AREF} supply current	I_{REF}	–	–	5	mA	3)
Internal reference error	$V_{IntREFER}$	–	± 5	± 30	mV	3)

2) The internal resistance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (t_L). After charging of the internal capacitance (C_I)in the load time (t_L) the analog input must be held constant for the rest of the sample time (t_S).

3) The differential impedance r_D of the analog reference voltage source must be less than 1 kΩ at reference supply voltage.

AC Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V;
T_A = 0 to + 70 °C; for SAB 80515/80535
T_A = -40 to + 85 °C; for SAB 80515/80535 - T40/85
T_A = -40 to + 110 °C; for SAB 80515/80535 - T40/110
(C_L for port 0, ALE and PSEN outputs = 100 pF; C_L for all other outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 12 MHz		
		min.	max.	min.	max.	
Cycle time	t_{CY}	1000	–	$12t_{CLCL}$	–	ns
ALE pulse width	t_{LHLL}	127	–	$2t_{CLCL}-40$	–	ns
Address setup to ALE	t_{AVLL}	53	–	$t_{CLCL}-30$	–	ns
Address hold after ALE	t_{LAX1}	48	–	$t_{CLCL}-35$	–	ns
ALE to valid instruction in	t_{LLIV}	–	233	–	$4t_{CLCL}-100$	ns
ALE to $\overline{PSEN}$	t_{LLPL}	58	–	$t_{CLCL}-25$	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	215	–	$3t_{CLCL}-35$	–	ns
$\overline{PSEN}$ to valid instruction in	t_{PLIV}	–	150	–	$3t_{CLCL}-100$	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	–	63	–	$t_{mCLCL}-20$	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	75	–	$t_{LCL}-8$	–	ns
Address to valid instruction in	t_{AVIV}	–	302	–	$5t_{CLCL}-115$	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the SAB 80515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

External Data Memory Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V;
T_A = 0 to + 70 °C; for SAB 80515/80535
T_A = -40 to + 85 °C; for SAB 80515/80535 - T40/85
T_A = -40 to + 110 °C; for SAB 80515/80535 - T40/110

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1,2 MHz to 12 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	400	–	$6t_{CLCL}-100$	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	400	–	$6t_{CLCL}-100$	–	ns
Address hold after ALE	t_{LLAX2}	132	–	$2t_{CLCL}-35$	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	252	–	$5t_{CLCL}-165$	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	97	–	$2t_{CLCL}-70$	ns
ALE to valid data in	t_{LLDV}	–	517	–	$8t_{CLCL}-150$	ns
Address to valid data in	t_{AVDV}	–	585	–	$9tnCLCL-165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
Address to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	203	–	$4t_{CLCL}-130$	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	43	123	$tnCLCL-40$	$t_{CLCL}+40$	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	33	–	$t_{CLCL}-50$	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	433	–	$7t_{CLCL}-150$	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	33	–	$t_{CLCL}-50$	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

Figure 21
Program Memory Read Cycle

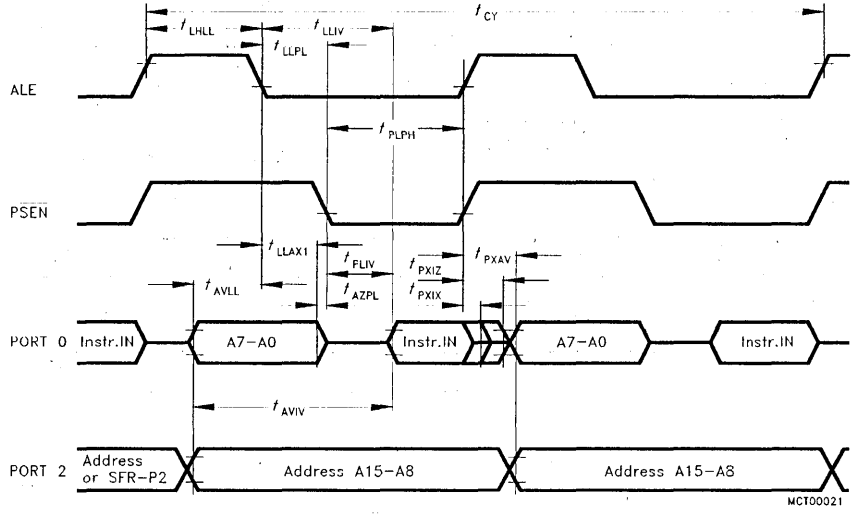
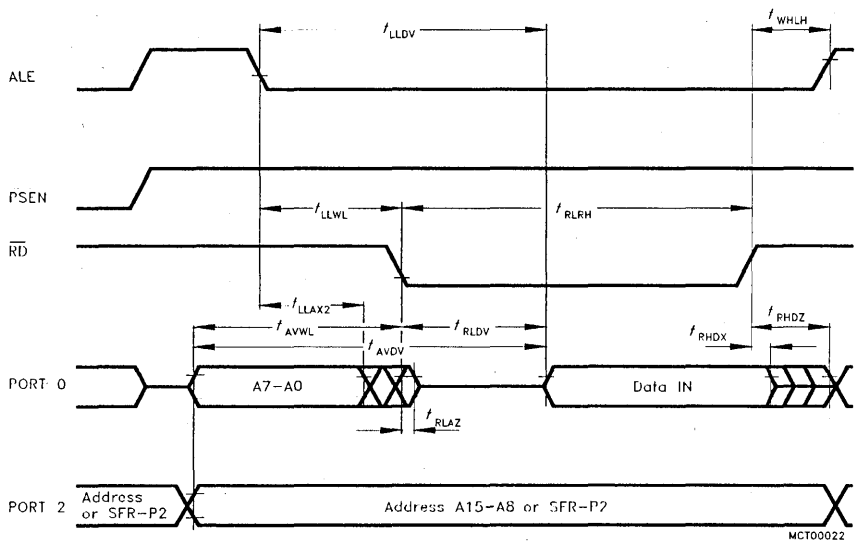


Figure 22
Data Memory Read Cycle



4

Figure 23
Data Memory Write Cycle

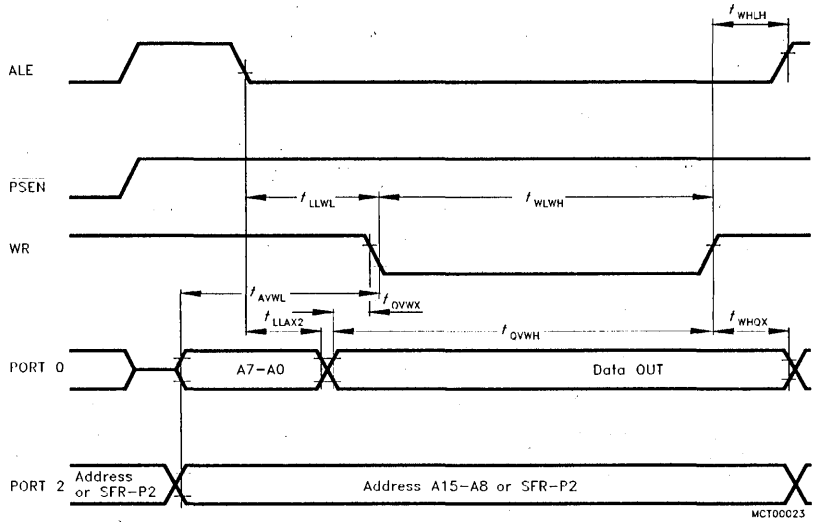
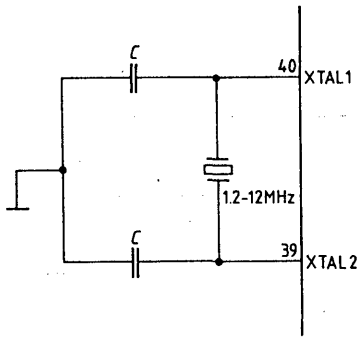
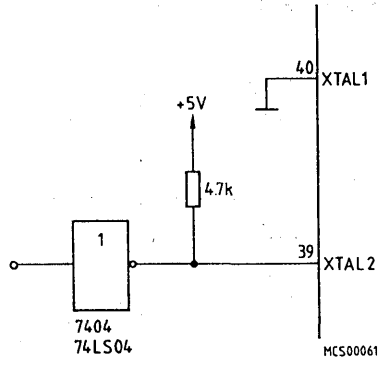


Figure 24
Recommended Oscillator Circuits



$C = 30pF \pm 10pF$

Crystal Oscillator Mode



Driving from External Source

External Clock Drive XTAL2

Parameter	Symbol	Limit Values		Unit
		Variable clock Freq = 1.2 MHz to 12 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	83.3	833.3	ns
High time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCK}	20	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

Figure 25
External Clock Cycle

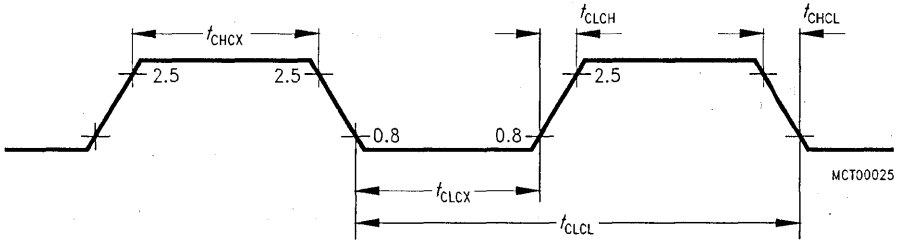
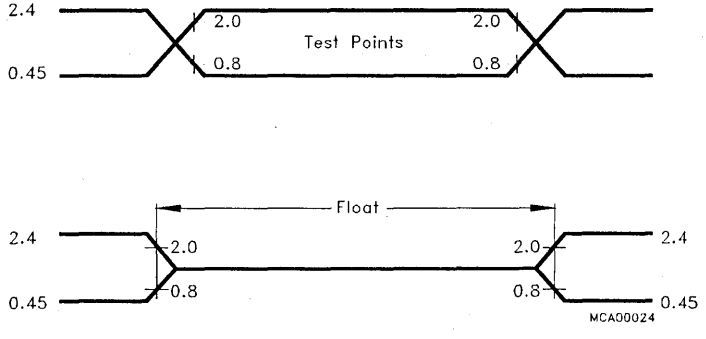


Figure 26
A.C. Testing Input, Output, Float Waveforms

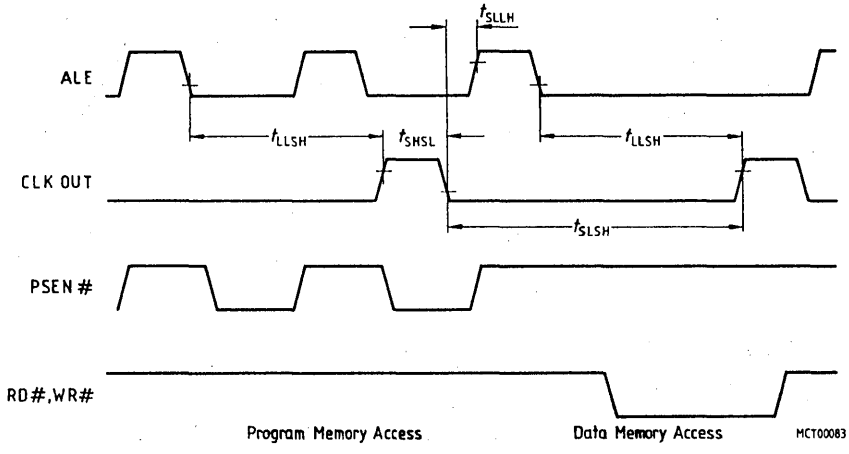


A.C. testing inputs are driven at 2.4 V for a logic "1" and at 0.45 V for a logic "0".
Timing measurements are made at 2.0 V for a logic "1" and at 0.8 V for a logic "0".
For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2 mA or sources 400 µA at the voltage test levels.

System Clock Timing

Parameter	Symbol	Limit Values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 1.2 MHz to 12 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	543	–	$7t_{CLCL}-40$	–	ns
CLKOUT high time	t_{SHSL}	127	–	$2t_{CLCL}-40$	–	ns
CLKOUT low time	t_{SLSH}	793	–	$10t_{CLCL}-40$	–	ns
CLKOUT low to ALE high	t_{SLLH}	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns

Figure 27
System Clock Timing

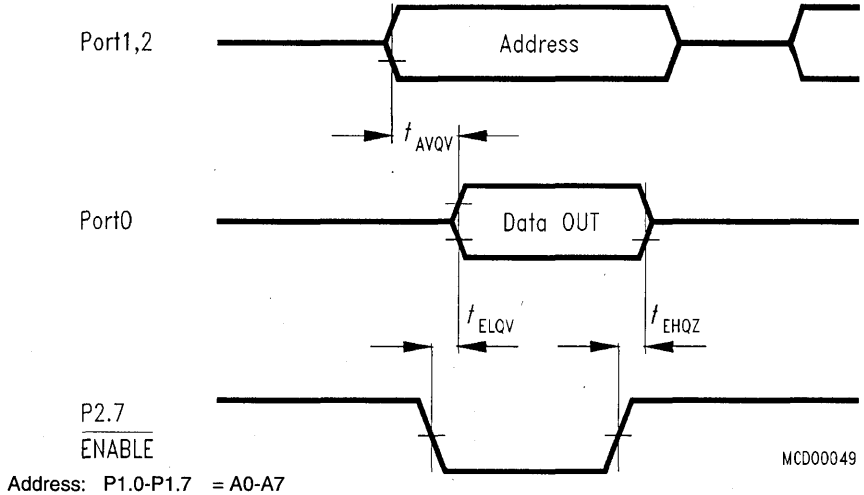


ROM Verification Characteristics

T_A = 25 °C ± °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	–	48 t_{CLCL}	ns
ENABLE to valid data	t_{ELQV}	–	48 t_{CLCL}	ns
Data float after ENABLE	t_{EHQZ}	0	48 t_{CLCL}	ns
Oscillator frequency	$1/t_{CLCL}$	4	6	MHz

Figure 28
ROM Verification



Address: P1.0-P1.7 = A0-A7
P2.0-P2.4 = A8-A12

Data: Port 0 = D0-D7

Inputs: P2.5-P2.6, $\overline{PSEN}$ = V_{SS}
ALE, $\overline{EA}$ = V_{IH}
$\overline{RESET}$ = V_{IL}

4

8-Bit CMOS Microcontroller

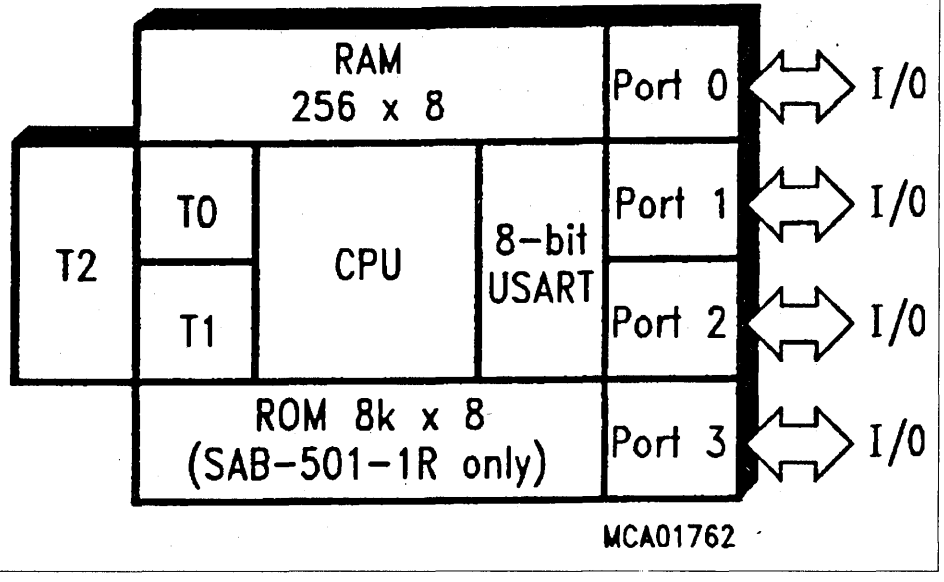
- Fully compatible to standard 8051 microcontroller
- Versions for 12/20/40 MHz operating frequency
- 8 K × 8 ROM (SAB-C501-1R only)
- 256 × 8 RAM
- Four 8-bit ports
- Three 16-bit Timers / Counters (Timer 2 with Up/Down Counter feature)
- USART
- Six interrupt sources, two priority levels
- Power Saving Modes
- P-DIP-40 and P-LCC-44 package
- Temperature ranges:
  SAB-C501 T_A: 0 °C to 70 °C
  SAF-C501 T_A: – 40 °C to 85 °C

The SAB-C501-L/C501-1R described in this document is compatible with the SAB 80C32/C52 and can be used for all present SAB 80C52 applications.

The SAB-C501-1R contains a non-volatile 8K × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four ports, three 16-bit timers counters, a seven source, two priority level interrupt structure, a serial port and versatile fail save mechanisms. The SAB-C501-L is identical, except that it lacks the program memory on chip. Therefore the term SAB-C501 refers to both versions within this specification unless otherwise noted.

4

Figure 29
SAB-C501

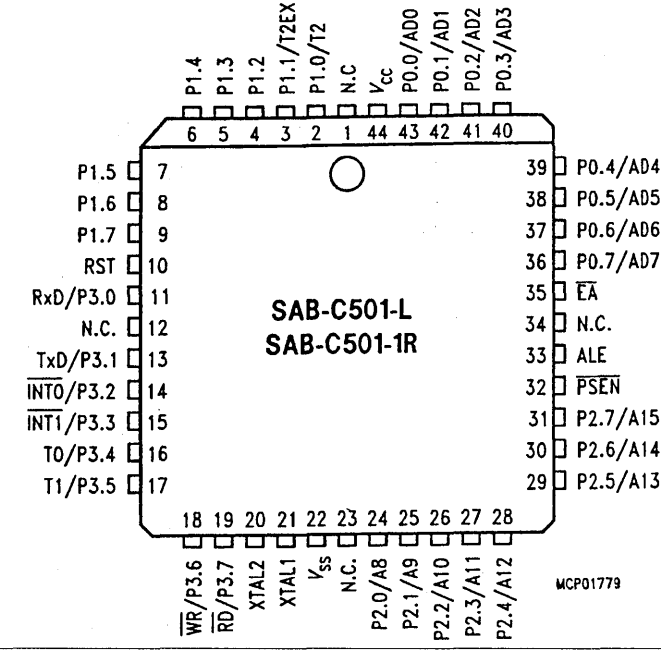


Ordering Information

Type	Package	Description (8-Bit CMOS microcontroller)
SAB-C501-LN SAB-C501-LP	P-LCC-44 P-DIP-40	for external memory 12 MHz
SAB-C501-1RN SAB-C501-1RP	P-LCC-44 P-DIP-40	with factory mask-programmable ROM 12 MHz
SAB-C501-L20N SAB-C501-L20P	P-LCC-44 P-DIP-40	for external memory 20 MHz
SAB-C501-1R20N SAB-C501-1R20P	P-LCC-44 P-DIP-40	with factory mask-programmable ROM 20 MHz
SAB-C501-L40N SAB-C501-L40P	P-LCC-44 P-DIP-40	for external memory 40 MHz
SAB-C501-1R40N SAB-C501-1R40P	P-LCC-44 P-DIP-40	with factory mask-programmable ROM 40 MHz
SAF-C501-LN SAF-C501-LP	P-LCC-44 P-DIP-40	for external memory, 12 MHz ext. temp. – 40 °C to 85 °C
SAF-C501-1RN SAF-C501-1RP	P-LCC-44 P-DIP-40	with factory mask-programmable ROM 12 MHz ext. temp. – 40 °C to 85 °C
SAF-C501-L20N SAF-C501-L20P	P-LCC-44 P-DIP-40	for external memory, 20 MHz ext. temp. – 40 °C to 85 °C
SAF-C501-1R20N SAF-C501-1R20P	P-LCC-44 P-DIP-40	with factory mask-programmable ROM 20 MHz ext. temp. – 40 °C to 85 °C

Note: Extended temperature range – 40 °C to 110 °C (SAH-C501) on request.

Figure 30
Pin Configuration (P-LCC-44)



4

Figure 31
Pin Configuration (P-DIP-40)

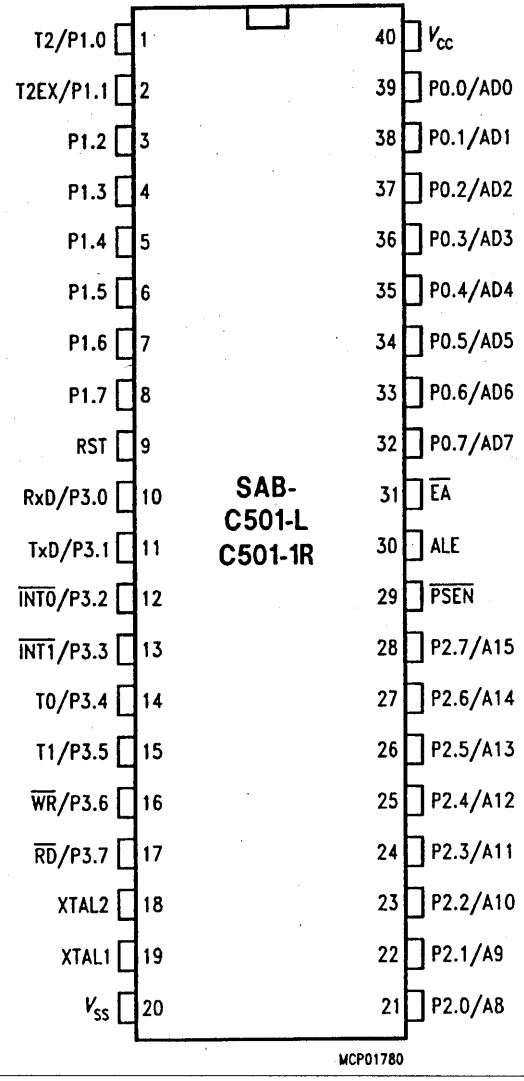
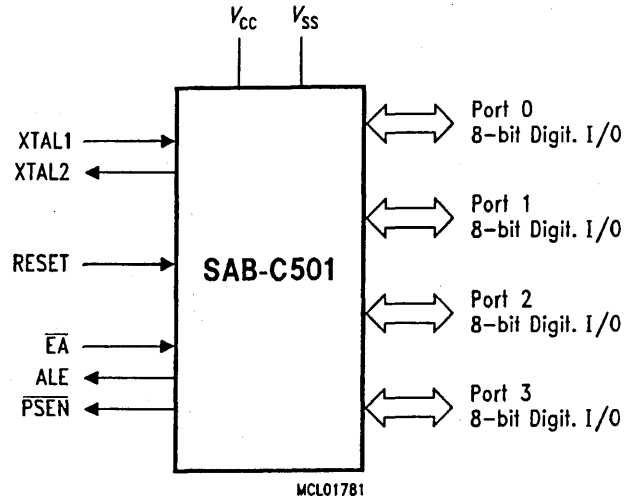


Figure 32
Logic Symbol



4

Pin Definitions and Functions

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
P1.7 – P1.0	2–9	1–8	I	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 1 also contains the timer 2 pins as secondary function. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate.
	2 3	1 2		The secondary functions are assigned to the pins of port 1, as follows: P1.0 T2 Input to counter 2 P1.1 T2EX Capture - Reload trigger of timer 2 / Up-Down count

*) I = Input
O = Output

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
P3.0 – P3.7	11, 13–19	10–17	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state they can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins which are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: P3.0 (11, 10) RxD – receiver data input (asynchronous) or data input output (synchronous) of serial interface 0 P3.1 (13, 11) TxD – transmitter data output (asynchronous) or clock output (synchronous) of the serial interface 0 P3.2 (14, 12) $\overline{INT0}$ – interrupt 0 input/timer 0 gate control P3.3 (15, 13) $\overline{INT1}$ – interrupt 1 input/timer 1 gate control P3.4 (16, 14) T0 – counter 0 input P3.5 (17, 15) T1 – counter 1 input P3.6 (18, 16) $\overline{WR}$ – the write control signal latches the data byte from port 0 into the external data memory P3.7 (19, 17) $\overline{RD}$ – the read control signal enables the external data memory to port 0
XTAL2	20	18	–	**XTAL2** Output of the inverting oscillator amplifier.

*) I = Input
O = Output

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
XTAL1	21	19	–	**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise and fall times specified in the AC characteristics must be observed.
P2.0 – P2.7	24–31	21–28	I/O	**Port 2** is a bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state they can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pull-up resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
$\overline{\text{PSEN}}$	32	29	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
RST	10	9	I	**RESET** A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{CC}.

*) I = Input
O = Output

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
ALE	33	30	O	The **Address Latch Enable** output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	35	31	I	**$\overline{External\ Access\ Enable}$** When held at high level, instructions are fetched from the internal ROM (SAB C501-1R only) when the PC is less than 2000H. When held at low level, the SAB C501 fetches all instructions from external program memory. For the SAB C501-L this pin must be tied low.
P0.0 – P0.7	43–36	39–32	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB C501-1R. External pull-up resistors are required during program verification.
V_{SS}	22	20	–	**Circuit ground potential**
V_{CC}	44	40	–	**Supply terminal** for all operating modes
N.C.	1, 12, 23, 24	–	–	**No connection**

*) I = Input
O = Output

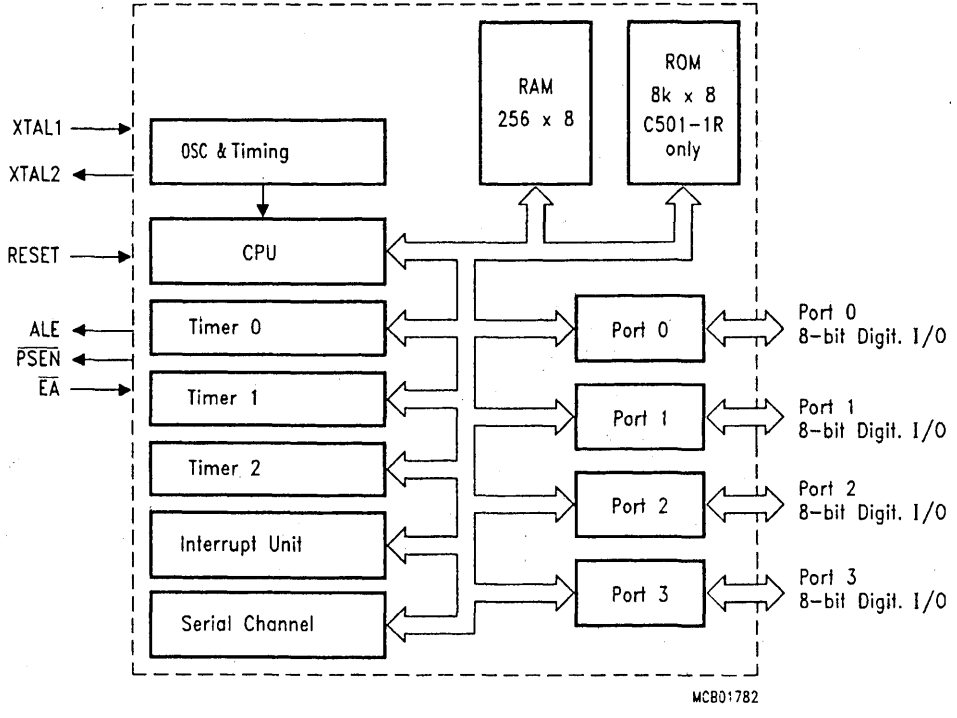
Functional Description

The SAB-C501 is fully compatible to standard 8051 microcontroller.

It is compatible with the SAB 80C52. While maintaining all architectural and operational characteristics of the SAB 80C52 the SAB-C501 incorporates some enhancements in the Timer2 Unit.

Figure 33 Block Diagram of the SAB-C501 shows a block diagram of the SAB-C501.

Figure 33
Block Diagram of the SAB-C501



4

CPU

The SAB-C501 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions are executed in 1.0 µs (20 MHz: 600 ns, 40 MHz: 300 ns).

Figure 34 Special Function Register PSW

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
Addr. 0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

Bit		Function
CY		**Carry Flag**
AC		**Auxiliary Carry Flag (for BCD operations)**
F0		**General Purpose Flag**
RS1	**RS0**	**Register Bank select control bits**
0	0	Bank 0 selected, data address 00H – 07H
0	1	Bank 1 selected, data address 08H – 0FH
1	0	Bank 2 selected, data address 10H – 17H
1	1	Bank 3 selected, data address 18H – 1FH
OV		**Overflow Flag**
F1		**General Purpose Flag**
P		**Parity Flag** Set/cleared by hardware each instruction cycle to indicate an odd/ even number of "one" bits in the accumulator, i.e. even parity.

Reset value of PSW is 00H.

Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area.

The 27 special function registers (SFR) include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. There are also 128 directly addressable bits within the SFR area.

All SFRs are listed in Table 1, Table 2, and Table 3.

In Table 1 they are organized in numeric order of their addresses. In Table 2 they are organized in groups which refer to the functional blocks of the SAB-C501. Table 3 illustrates the contents of the SFRs.

Table 1
Special Function Registers in numeric order of their addresses

Address	Register	Contents after Reset	Address	Register	Contents after Reset
80H	**P0** [1]	**0FFH**	**98H**	**SCON** [1]	**00H**
81H	SP	07H	99H	SBUF	XXH [2]
82H	DPL	00H	9AH	reserved	XXH [2]
83H	DPH	00H	9BH	reserved	XXH [2]
84H	reserved	XXH [2]	9CH	reserved	XXH [2]
85H	reserved	XXH [2]	9DH	reserved	XXH [2]
86H	reserved	XXH [2]	9EH	reserved	XXH [2]
87H	PCON	00H	9FH	reserved	XXH [2]
88H	**TCON** [1]	**00H**	**A0H**	**P2** [1]	**0FFH**
89H	TMOD	00H	A1H	reserved	XXH [2]
8AH	TL0	00H	A2H	reserved	XXH [2]
8BH	TL1	00H	A3H	reserved	XXH [2]
8CH	TH0	00H	A4H	reserved	XXH [2]
8DH	TH1	00H	A5H	reserved	XXH [2]
8EH	reserved	XXH [2]	A6H	reserved	XXH [2]
8FH	reserved	XXH [2]	A7H	reserved	XXH [2]
90H	**P1** [1]	**0FFH**	**A8H**	**IE** [1]	**00H**
91H	reserved	00H	A9H	reserved	XXH [2]
92H	reserved	XXH [2]	AAH	reserved	XXH [2]
93H	reserved	XXH [2]	ABH	reserved	XXH [2]
94H	reserved	XXH [2]	ACH	reserved	XXH [2]
95H	reserved	XXH [2]	ADH	reserved	XXH [2]
96H	reserved	XXH [2]	AEH	reserved	XXH [2]
97H	reserved	XXH [2]	AFH	reserved	XXH [2]

[1] Bit-addressable Special Function Register
[2] X means that the value is indeterminate and the location is reserved

Table 1
Special Function Registers in numeric order of their addresses (cont'd)

Address	Register	Contents after Reset	Address	Register	Contents after Reset
B0H	**P3** [1]	**0FFH**	**D8H**	reserved	XXH [2]
B1H	reserved	XXH [2]	D9H	reserved	XXH [2]
B2H	reserved	XXH [2]	DAH	reserved	XXH [2]
B3H	reserved	XXH [2]	DBH	reserved	XXH [2]
B4H	reserved	XXH [2]	DCH	reserved	XXH [2]
B5H	reserved	XXH [2]	DDH	reserved	XXH [2]
N6H	reserved	XXH [2]	DEH	reserved	XXH [2]
B7H	reserved	XXH [2]	DFH	reserved	XXH [2]
B8H	**IP** [1]	**X0000000B** [2]	**E0H**	**ACC** [1]	**00H**
B9H	reserved	XXH [2]	E1H	reserved	XXH [2]
BAH	reserved	XXH [2]	E2H	reserved	XXH [2]
BBH	reserved	XXH [2]	E3H	reserved	XXH [2]
BCH	reserved	XXH [2]	E4H	reserved	XXH [2]
BDH	reserved	XXH [2]	E5H	reserved	XXH [2]
BEH	reserved	XXH [2]	E6H	reserved	XXH [2]
BFH	reserved	XXH [2]	E7H	reserved	XXH [2]
C0H	reserved	XXH [2]	**E8H**	reserved	XXH [2]
C1H	reserved	XXH [2]	E9H	reserved	XXH [2]
C2H	reserved	XXH [2]	EAH	reserved	XXH [2]
C3H	reserved	XXH [2]	EBH	reserved	XXH [2]
C4H	reserved	XXH [2]	ECH	reserved	XXH [2]
C5H	reserved	XXH [2]	EDH	reserved	XXH [2]
C6H	reserved	XXH [2]	EEH	reserved	XXH [2]
C7H	reserved	XXH [2]	EFH	reserved	XXH [2]
C8H	**T2CON**	**00H**	**F0H**	**B** [1]	**00H**
C9H	T2MOD	XXXXXXX0 [2]	F1H	reserved	XXH [2]
CAH	RC2L	00H	F2H	reserved	XXH [2]
CBH	RC2H	00H	F3H	reserved	XXH [2]
CCH	TL2	00H	F4H	reserved	XXH [2]
CDH	TH2	00H	F5H	reserved	XXH [2]
CEH	reserved	XXH [2]	F6H	reserved	XXH [2]
CFH	reserved	XXH [2]	F7H	reserved	XXH [2]
D0H	**PSW** [1]	**00H**	**F8H**	reserved	XXH [2]
D1H	reserved	XXH [2]	F9H	reserved	XXH [2]
D2H	reserved	XXH [2]	FAH	reserved	XXH [2]
D3H	reserved	XXH [2]	FBH	reserved	XXH [2]
D4H	reserved	XXH [2]	FCH	reserved	XXH [2]
D5H	reserved	XXH [2]	FDH	reserved	XXH [2]
D6H	reserved	XXH [2]	FEH	reserved	XXH [2]
D7H	reserved	XXH [2]	FFH	reserved	XXH [2]

Table 2
Special Function Registers - Functional blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	0E0H [1]	00H
	B	B-Register	0F0H [1]	00H
	DPH	Data Pointer, High Byte	83H	00H
	DPL	Data Pointer, Low Byte	82H	00H
	PSW	Program Status Word Register	0D0H [1]	00H
	SP	Stack Pointer	81H	07H
Interrupt System	IE	Interrupt Enable Register	0A8H [1]	00H
	IP	Interrupt Priority Register	0B8H [1]	00H
Ports	P0	Port 0	80H [1]	0FFH
	P1	Port 1	90H [1]	0XXH [3]
	P2	Port 2	0A0H [1]	0FFH
	P3	Port 3	0B0H [1]	0FFH
Serial Channels	PCON [2]	Power Control Register	87H	00H
	SBUF	Serial Channel Buffer Reg.	99H	0XXH [3]
	SCON	Serial Channel 0 Control Reg.	98H [1]	00H
Timer 0 / Timer 1	TCON	Timer 0/1 Control Register	88H [1]	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Timer 2	T2CON	Timer 2 Control Register	0C8H [1]	00H
	T2MOD	Timer 2 Mode Register	0C9H	00H
	RC2H	Timer 2 Reload Capture Reg., High Byte	0CBH	00H
	RC2L	Timer 2 Reload Capture Reg., Low Byte.	0CAH	00H
	TH2	Timer 2, High Byte	0CDH	00H
	TL2	Timer 2, Low Byte	0CCH	00H
Pow.Sav.Modes	PCON	Power Control Register	87H	00H

[1] Bit-addressable special function registers
[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks
[3] X means that the value is indeterminate and the location is reserved

4

Table 3
Contents of SFRs, SFRs in numeric order

Address	Register	Bit 7	6	5	4	3	2	1	0
80H	P0								
81H	SP								
82H	DPL								
83H	DPH								
87H	PCON	SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE
88H	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89H	TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
8AH	TL0								
8BH	TL1								
8CH	TH0								
8DH	TH1								
90H	P1								
98H	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99H	SBUF								
A0H	P2								
A8H	IE	EA	–	ET2	ES	ET1	EX1	ET0	EX0
B0H	P3								
B8H	IP	–	–	PT2	PS	PT1	PX1	PT0	PX0
C8H	T2CON	TF2	EXF2	RCLK	TCLK	EXEN	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
C9H	T2MOD	–	–	–	–	–	–	–	DCEN

SFR bit and byte addressable

SFR not bit addressable

– : = this bit location is reserved

Table 3
Contents of SFRs, SFRs in numeric order (cont'd)

Address	Register	Bit 7	6	5	4	3	2	1	0
CAH	RC2L								
CBH	RC2H								
CCH	TL2								
CDH	TH2								
D0H	PSW	CY	AC	F0	RS1	RS0	OV	F1	P
E0H	ACC								
F0H	B								

SFR bit and byte addressable

SFR not bit addressable

– : = this bit location is reserved

4

Timer / Counter 0 and 1

Timer/Counter 0 and 1 can be used in four operating modes as listed in Table 4:

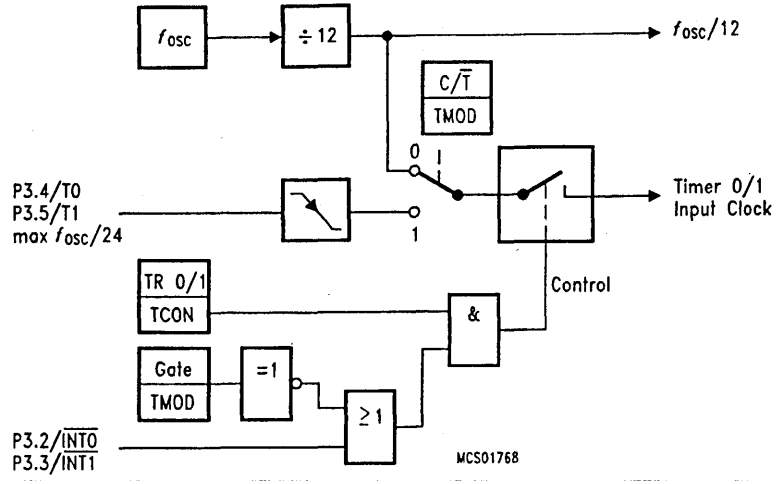
Table 4
Timer/Counter 0 and 1 operating modes

Mode	Description	TMOD				Input Clock	
		Gate	C/$\overline{T}$	M1	M0	internal	external (max)
0	8-bit timer/counter with a divide-by-32 prescaler	X	X	0	0	f_{osc}/12 x 32	f_{osc}/24 x 32
1	16-bit timer/counter	X	X	1	1	f_{osc}/12	f_{osc}/24
2	8-bit timer/counter with 8-bit autoreload	X	X	0	0	f_{osc}/12	f_{osc}/24
3	Timer/counter 0 used as one 8-bit timer/counter and one 8-bit timer Timer 1 stops	X	X	1	1	f_{osc}/12	f_{osc}/24

In the "timer" function (C/$\overline{T}$ = '0') the register is incremented every machine cycle. Therefore the count rate is f_{osc}/12.

In the "counter" function the register is incremented in response to a 1-to-0 transition at its corresponding external input pin (P3.4/T0, P3.5/T1). Since it takes two machine cycles to detect a falling edge the max. count rate is f_{osc}/24. External inputs $\overline{INT0}$ and $\overline{INT1}$ (P3.2, P3.3) can be programmed to function as a gate to facilitate pulse width measurements. Figure 35 Timer/Counter 0 and 1 input clock logic illustrates the input clock logic.

Figure 35
Timer/Counter 0 and 1 input clock logic



Timer 2

Timer 2 is a 16-bit Timer/Counter with an up/down count feature. It can operate either as timer or as an event counter which is selected by bit C/$\overline{T2}$ (T2CON.1). It has three operating modes as shown in Table 5.

Table 5
Timer/Counter 2 operating modes

Mode	T2CON			T2MOD	T2CON	P1.1/ T2EX	Remarks	Input Clock	
	R×CLK or T×CLK	CP/$\overline{RL2}$	TR2	DCEN	EXEN			internal	external (P1.0/T2)
16-bit Auto-reload	0	0	1	0	0	X	reload upon overflow	$f_{OSC}/12$	max $f_{OSC}/24$
	0	0	1	0	1	↓	reload trigger (falling edge)		
	0	0	1	1	X	0	Down counting		
	0	0	1	1	X	1	Up counting		
16-bit Capture	0	1	1	X	0	X	16 bit Timer/ Counter (only up-counting) capture TH2, TL2 → RC2H, RC2L	$f_{OSC}/12$	max $f_{OSC}/24$
	0	1	1	X	1	↓			
Baud Rate Generator	1	X	1	X	0	X	no overflow interrupt request (TF2) extra external interrupt ("Timer 2")	$f_{OSC}/2$	max $f_{OSC}/24$
	1	X	1	X	1	↓			
off	X	X	0	X	X	X	Timer 2 stops	–	–

Note: ↓ = falling edge

4

Serial Interface (USART)

The serial port is full duplex and can operate in four modes (one synchronous mode, three asynchronous modes) as illustrated in Table 6. The possible baudrates can be calculated using the formulas given in Table 7.

Table 6
USART operating modes

Mode	SCON		Baudrate	Description
	SM0	SM1		
0	0	0	$f_{osc}/12$	Serial data enters and exits through RxD. TxD outputs the shift clock. 8-bit are transmitted/received (LSB first)
1	0	1	Timer 1/2 overflow rate	8-bit UART 10 bits are transmitted (through TxD) or received (RxD)
2	1	0	$f_{osc}/32$ or $f_{osc}/64$	9-bit UART 11 bits are transmitted (TxD) or received (RxD)
3	1	1	Timer 1/2 overflow rate	9-bit UART Like mode 2 except the variable baud rate

Table 7
Formulas for calculating Baudrates

Baud rate derived from	Interface Mode	Baudrate
Oscillator	0	$f_{osc}/12$
	2	$(2^{SMOD} \times f_{osc}) / 64$
Timer 1 (16-bit timer)	1,3	$(2^{SMOD}$ × timer 1 overflow rate) /32
(8-bit timer with 8-bit autoreload)	1,3	$(2^{SMOD} \times f_{osc}) / (32 \times 12 \times (256\text{-TH1}))$
Timer 2	1,3	$f_{osc} / (32 \times (65536\text{-(RC2H, RC2L)}))$

Interrupt System

The SAB-C501 provides 6 interrupt sources with two priority levels.

Figure 36 Interrupt Request Sources gives a general overview of the interrupt sources and illustrates the request and control flags.

Figure 36
Interrupt Request Sources

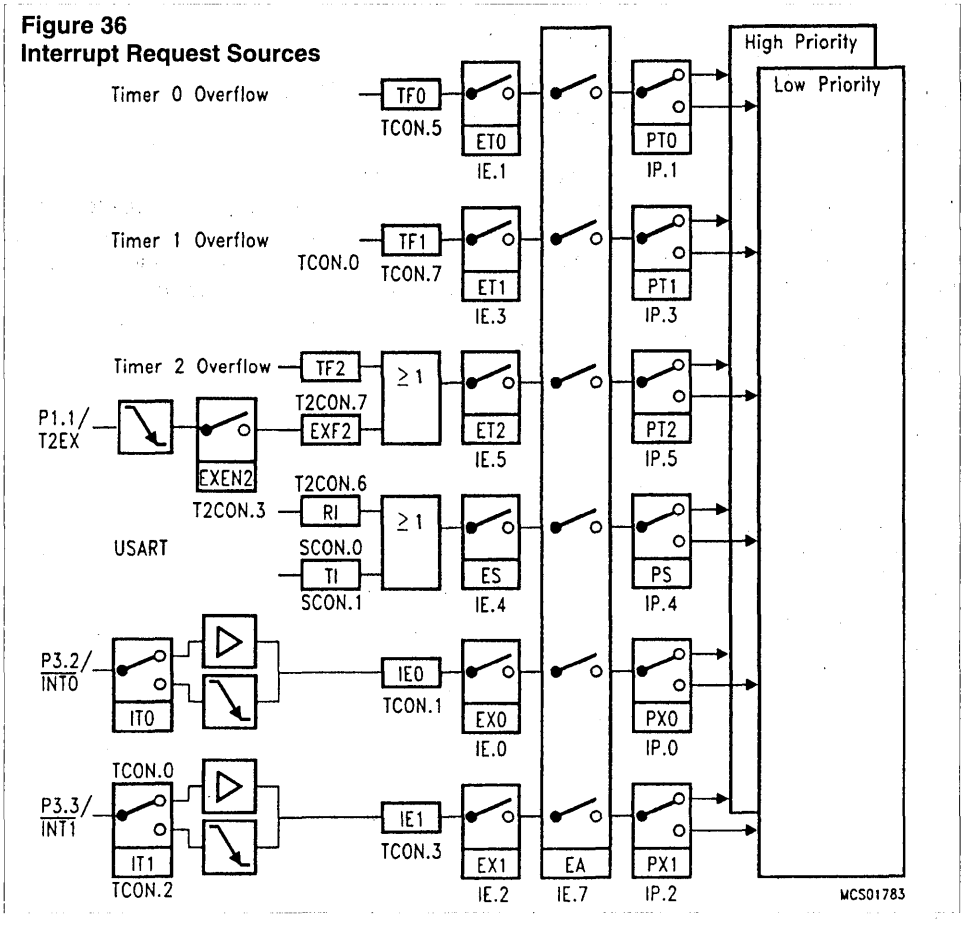


Table 8
Interrupt sources and their corresponding interrupt vectors

Source (Request Flags)	Vector	Vector Address
IE0	External interrupt 0	0003H
TF0	Timer 0 interrupt	000BH
IE1	External interrupt 1	0013H
TF1	Timer 1 interrupt	001BH
RI + TI	Serial port interrupt	0023H
TF2 + EXF2	Timer 2 interrupt	002BH

A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

If two requests of different priority level are received simultaneously, the request of higher priority is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as shown in Table 9.

Table 9
Interrupt priority-within-level

Interrupt Source		Priority
External Interrupt 0,	IE0	High
Timer 0 Interrupt,	TF0	
External Interrupt 1,	IE1	↓
Timer 1 Interrupt,	TF1	
Serial Channel,	RI + TI	
Timer 2 Interrupt,	TF2 + EXF2	Low

Power Saving Modes

Two power down modes are available, the Idle Mode and Power Down Mode.

The bits PDE, PDS and IDLE, IDLS select the Power Down mode or the Idle mode, respectively. If the Power Down mode and the Idle mode are set at the same time, the Power Down mode takes precedence. Table 10 gives a general overview of the power saving modes.

Table 10
Power Saving modes overview

Mode	Entering Example	Leaving by	Remarks
Idle mode	ORL PCON, #01H ORL PCON, #20H	– enabled interrupt – Hardware Reset	CPU is gated off CPU status registers maintain their data. Peripherals are active Double instruction sequence
Power-Down Mode	ORL PCON, #02H ORL PCON, #40H	Hardware Reset	Oscillator is stopped, contents of on-chip RAM and SFR's are maintained (leaving Power Down Mode means redefinition of SFR contents). Double instruction sequence

In the Power Down mode of operation, V_{CC} can be reduced to minimize power consumption. It must be ensured, however, that V_{CC} is not reduced before the Power Down mode is invoked, and that V_{CC} is restored to its normal operating level, before the Power Down mode is terminated. The reset signal that terminates the Power Down mode also restarts the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (simulator to power-on reset).

Absolute Maximum Ratings

Ambient temperature
under bias (T_A) ..... – 40 to + 85 °C
Storage temperature (T_{ST}) ..... – 65 to + 150 °C

Voltage on V_{CC} pins with
respect to ground (V_{SS}) ..... – 0.5 V to 6.5 V
Voltage on any pin with
respect to ground (V_{SS}) .. – 0.5 V to V_{CC} + 0.5 V

Input current on any pin during
overload condition ..... – 10 mA to + 10 mA
Absolute sum of all input currents
during overload condition ..... I 100 mA I

Power dissipation ..... TBD

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{CC} > V_{CC}$ or $V_{CC} < V_{CC}$) the Voltage on V_{CC} pins with respect to ground (V_{CC}) must not exceed the values defined by the absolute maximum ratings.*

DC Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{CC} = 0 V; T_A = 0 °C to 70 °C for the SAB-C501
T_A = – 40 °C to 85 °C for the SAF-C501

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, RESET)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage ($\overline{RESET}$)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except XTAL1, $\overline{EA}$, RESET)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL1	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	
Input high voltage to $\overline{EA}$, RESET	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (ports 1, 2, 3)	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA [1]
Output low voltage (port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA [1]
Output high voltage (ports 1, 2, 3)	V_{OH}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 80 µA, I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 800 µA [2], I_{OH} = – 80 µA [2]
Logic 0 input current (ports 1, 2, 3)	I_{IL}	– 10	– 50	µA	V_{IN} = 0.45 V

DC Characteristics (cont)

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Logical 1-to-0 transition current (ports 2, 3)	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{EA}$)	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 °C
Power supply current:					
Active mode, 12 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [4]
Idle mode, 12 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [5]
Active mode, 20 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [4]
Idle mode, 20 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [5]
Active mode, 40 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [4]
Idle mode, 40 MHz [7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V, [5]
Power Down Mode	I_{PD}	–	TBD	µA	V_{CC} = 2 ... 5.5 V, [3]

[1] Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and port 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

[2] Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall bellow the 0.9 V_{CC} specification when the address lines are stabilizing.

[3] I_{PD} (Power Down Mode) is measured under following conditions:
$\overline{EA}$ = RESET = Port0 = V_{CC}; XTAL2 = N.C.; XTAL1 = V_{SS}; all other pins are disconnected.

[4] I_{CC} (active mode) is measured with:

XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
$\overline{EA}$ = Port0 = RESET = V_{CC}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

[5] I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
RESET = $\overline{EA}$ = V_{SS}; Port0 = V_{CC}; all other pins are disconnected;

[7] $I_{CC\,Max}$ at other frequencies is given by:
active mode: TBD
idle mode: TBD
where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at v_{cc} = 5 V.

4

AC Characteristics for SAB-C501-LN / C501-RN

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to 70 °C for the SAB-C501
T_A = – 40 °C to 85 °C for the SAF-C501

(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	127	–	$2t_{CLCL} - 40$	–	ns
Address setup to ALE	t_{AVLL}	43	–	$t_{CLCL} - 40$	–	ns
Address hold after ALE	t_{LLAX}	60	–	$t_{CLCL} - 23$	–	ns
Address to valid instr in	t_{LLIV}	–	233	–	$4t_{CLCL} - 100$	ns
ALE to $\overline{PSEN}$	t_{LLPL}	58	–	$t_{CLCL} - 25$	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	215	–	$3t_{CLCL} - 35$	–	ns
$\overline{PSEN}$ to valid instr in	t_{PLIV}	–	150	–	$3t_{CLCL} - 100$	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	–	63	–	$t_{CLCL} - 20$	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	75	–	$t_{CLCL} - 8$	–	ns
Address to valid instr in	t_{AVIV}	–	302	–	$5t_{CLCL} - 115$	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the SAB-C501 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 Drivers.

AC Characteristics for SAB-C501-L / C501-1R

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	400	–	$6t_{CLCL} - 100$	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	400	–	$6t_{CLCL} - 100$	–	ns
Address hold after ALE	t_{LLAX2}	132	–	$2t_{CLCL} - 35$	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	252	–	$5t_{CLCL} - 165$	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	97	–	$2t_{CLCL} - 70$	ns
ALE to valid data in	t_{LLDV}	–	517	–	$8t_{CLCL} - 150$	ns
Address to valid data in	t_{AVDV}	–	585	–	$9t_{CLCL} - 165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	203	–	$4t_{CLCL} - 130$	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	33	–	$t_{CLCL} - 50$	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	433	–	$7t_{CLCL} - 150$	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	33	–	$t_{CLCL} - 50$	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 12 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	83.3	285.7	ns
High time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	20	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

AC Characteristics for SAB-C501-L20 / C501-1R20

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to 70 °C for the SAB-C501
T_A = – 40 °C to 85 °C for the SAF-C501

(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		20 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	60	–	$2t_{CLCL} - 40$	–	ns
Address setup to ALE	t_{AVLL}	20	–	$t_{CLCL} - 30$	–	ns
Address hold after ALE	t_{LLAX}	20	–	$t_{CLCL} - 30$	–	ns
Address to valid instr in	t_{LLIV}	–	100	–	$4t_{CLCL} - 100$	ns
ALE to $\overline{PSEN}$	t_{LLPL}	25	–	$t_{CLCL} - 25$	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	115	–	$3t_{CLCL} - 35$	–	ns
$\overline{PSEN}$ to valid instr in	t_{PLIV}	–	75	–	$3t_{CLCL} - 75$	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	–	40	–	$t_{CLCL} - 10$	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	47	–	$t_{CLCL} - 3$	–	ns
Address to valid instr in	t_{AVIV}	–	190	–	$5t_{CLCL} - 60$	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the SAB-C501 to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 Drivers.

AC Characteristics for SAB-C501-L20 / C501-1R20

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	200	–	$6t_{CLCL} - 100$	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	200	–	$6t_{CLCL} - 100$	–	ns
Address hold after ALE	t_{LLAX2}	65	–	$2t_{CLCL} - 35$	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	155	–	$5t_{CLCL} - 95$	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	40	–	$2t_{CLCL} - 60$	ns
ALE to valid data in	t_{LLDV}	–	250	–	$8t_{CLCL} - 150$	ns
Address to valid data in	t_{AVDV}	–	285	–	$9t_{CLCL} - 165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	100	200	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	70	–	$4t_{CLCL} - 130$	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	20	80	$t_{CLCL} - 30$	$t_{CLCL} + 30$	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	5	–	$t_{CLCL} - 45$	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	200	–	$7t_{CLCL} - 150$	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	10	–	$t_{CLCL} - 40$	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 20 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	50	285.7	ns
High time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	20	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

AC Characteristics for SAB-C501-L40 / C501-1R40

Advance Information

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to 70 °C for the SAB-C501
T_A = – 40 °C to 85 °C for the SAF-C501

(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		40 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 40 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	35	–	TBD	TBD	ns
Address setup to ALE	t_{AVLL}	10	–	TBD	TBD	ns
Address hold after ALE	t_{LLAX}	10	–	TBD	TBD	ns
Address to valid instr in	t_{LLIV}	–	60	TBD	TBD	ns
ALE to $\overline{PSEN}$	t_{LLPL}	10	–	TBD	TBD	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	60	–	TBD	TBD	ns
$\overline{PSEN}$ to valid instr in	t_{PLIV}	–	40	TBD	TBD	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	TBD	TBD	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	–	20	TBD	TBD	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	TBD	–	TBD	TBD	ns
Address to valid instr in	t_{AVIV}	–	90	TBD	TBD	ns
$\overline{PSEN}$ to Address float	t_{PLAZ}	–	10	TBD	TBD	ns

*) Interfacing the SAB-C501 to devices with float times up to TBD ns is permissible. This limited bus contention will not cause any damage to port 0 Drivers.

AC Characteristics for SAB-C501-L40 / C501-1R40 (cont'd)

Advance Information

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		40 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 40 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	120	–	TBD	TBD	ns
$\overline{WR}$ pulse width	t_{WLWH}	120	–	TBD	TBD	ns
Address hold after ALE	t_{LLAX2}	10	–	TBD	TBD	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	75	TBD	TBD	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	TBD	TBD	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	20	TBD	TBD	ns
ALE to valid data in	t_{LLDV}	–	150	TBD	TBD	ns
Address to valid data in	t_{AVDV}	–	150	TBD	TBD	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	60	100	TBD	TBD	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	70	–	TBD	TBD	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	10	40	TBD	TBD	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	5	–	TBD	TBD	ns
Data setup before $\overline{WR}$	t_{QVWH}	125	–	TBD	TBD	ns
Data hold after $\overline{WR}$	t_{WHQX}	5	–	TBD	TBD	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	TBD	TBD	ns

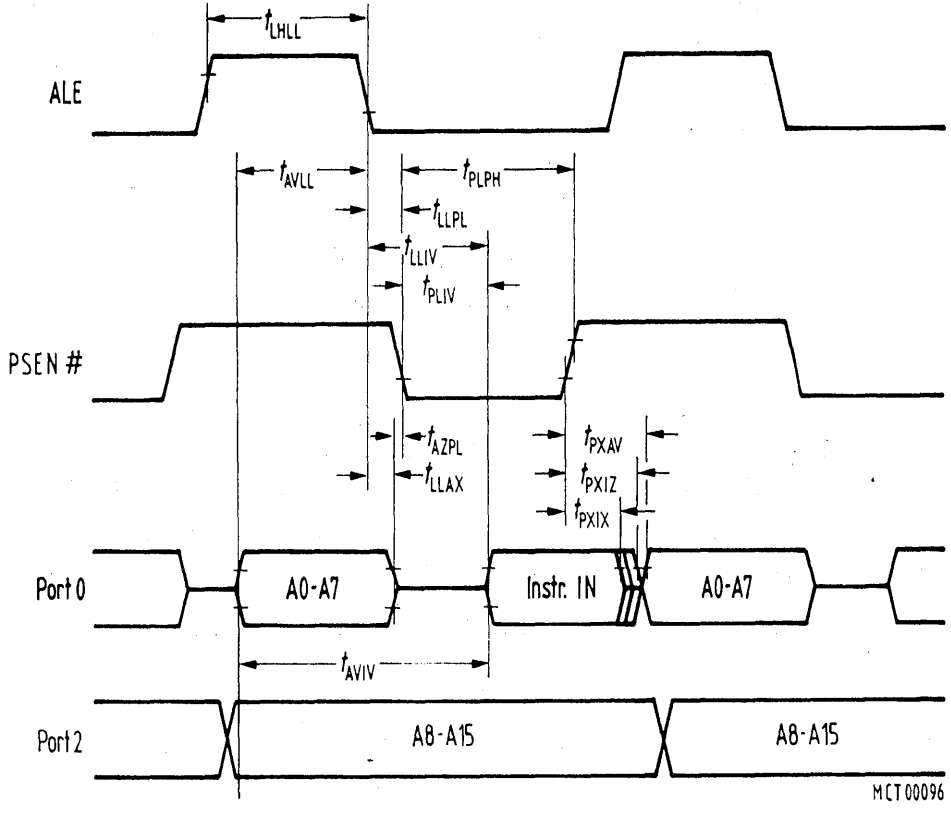
4

Advance Information

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 40 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	25	285.7	ns
High time	t_{CHCX}	TBD	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	TBD	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	TBD	ns
Fall time	t_{CHCL}	–	TBD	ns

Figure 37
Program Memory Read Cycle



4

Figure 38
Data Memory Read Cycle

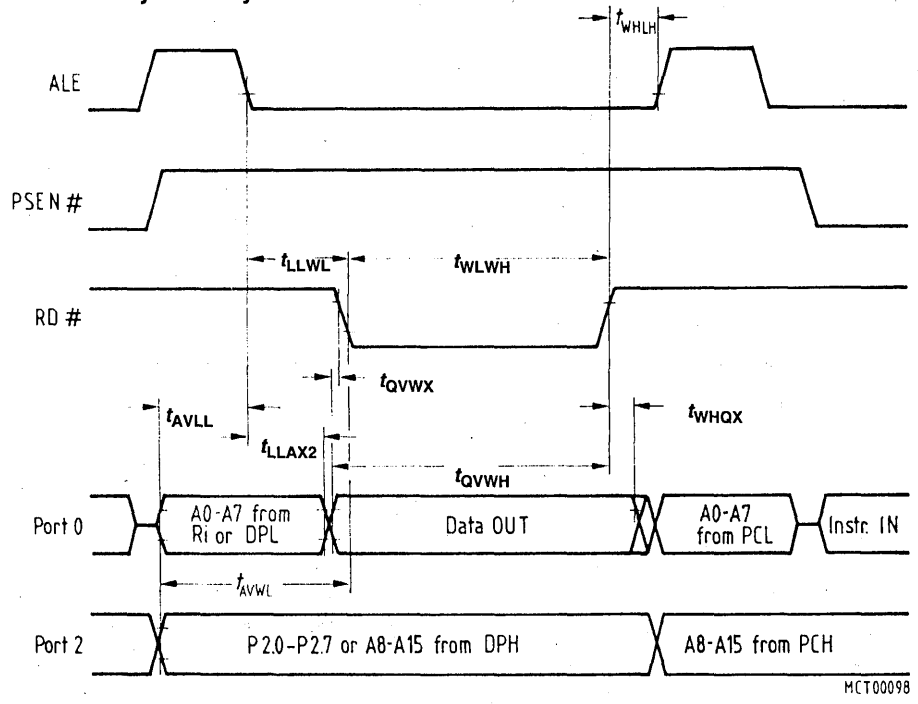
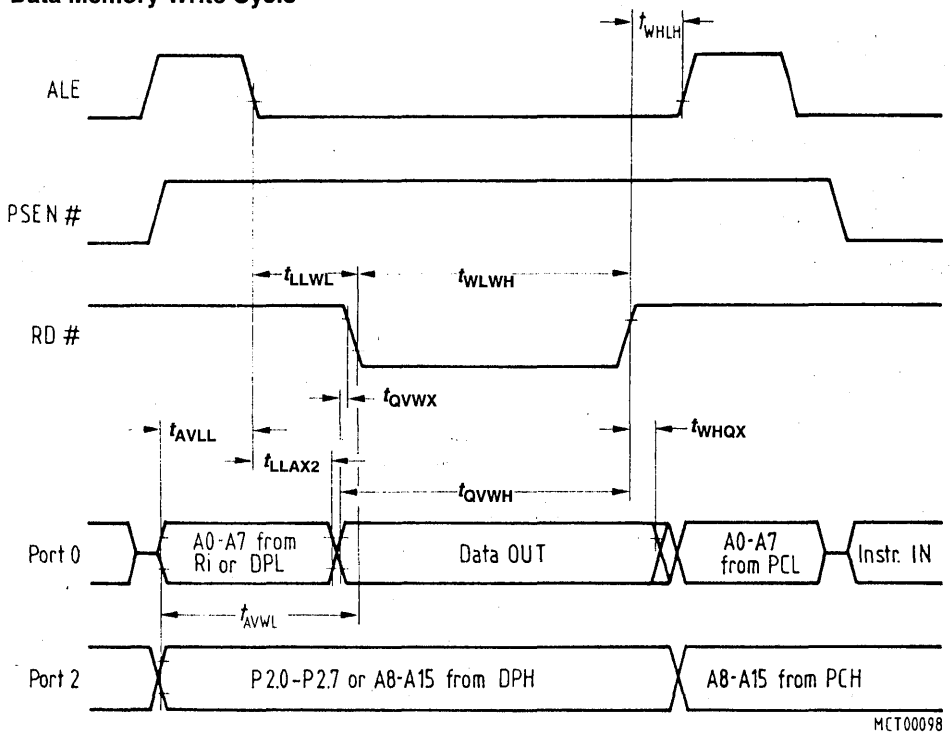


Figure 39
Data Memory Write Cycle



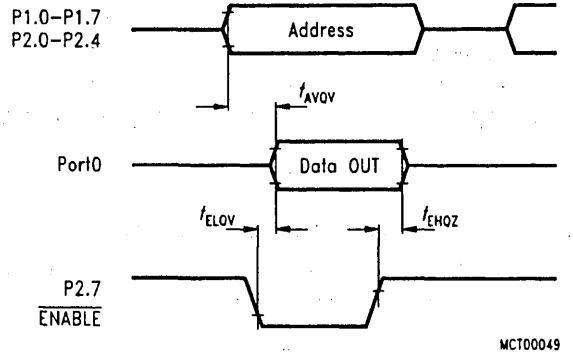
4

ROM Verification Characteristics for SAB-C501-1R

ROM Verification Mode 1

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	–	$48t_{CLCL}$	ns
ENABLE to valid data	t_{ELQV}	–	$48t_{CLCL}$	ns
Data float after ENABLE	t_{EHQZ}	0	$48t_{CLCL}$	ns
Oscillator frequency	$1/t_{CLCL}$	4	6	MHz

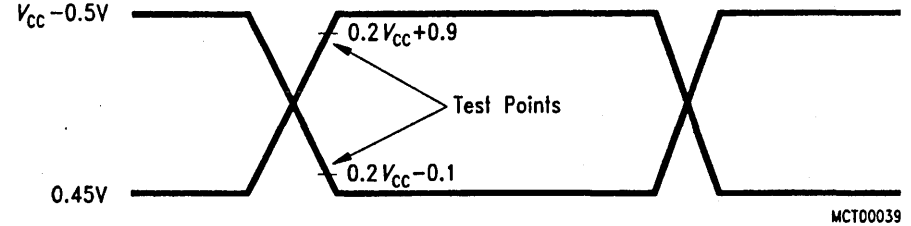
Figure 40
ROM Verification Mode 1



Address: P1.0–P1.7=A0–A7
P2.0–P2.4=A8–A12
Data: P0.0–P0.7=D0–D7

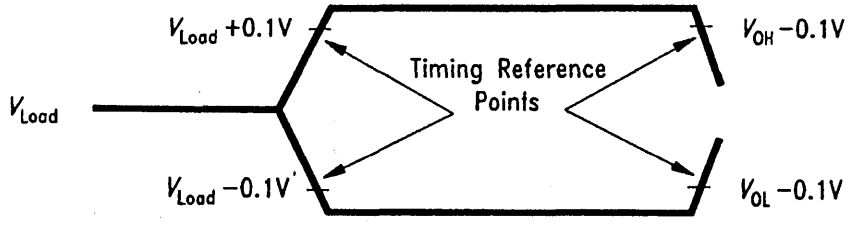
Inputs: P2.5–P2.6, $\overline{PSEN}=V_{SS}$
ALE, $\overline{EA}=V_{IH}$
RESET = V_{IH1}

Figure 41
AC Testing: Input, Output Waveforms



AC Inputs during testing are driven at V_{CC} – 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at V_{IHmin} for a logic '1' and V_{ILmax} for a logic '0'.

Figure 42
AC Testing: Float Waveforms



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH} / V_{OL} level occurs.

Figure 43
External Clock Cycle

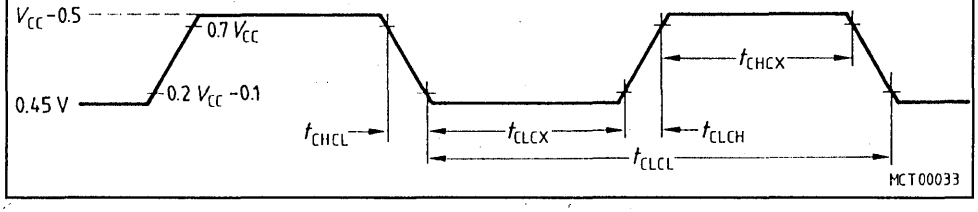
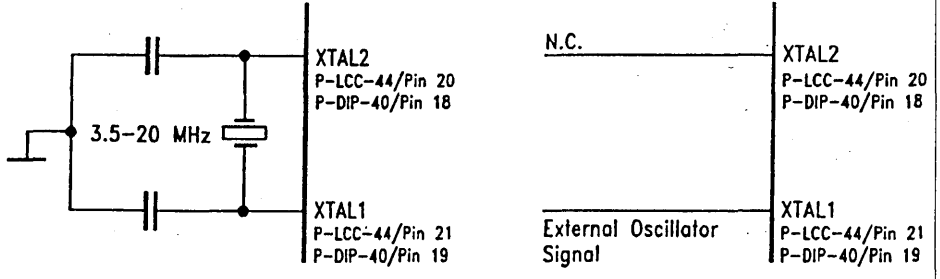


Figure 44
Recommended Oscillator Circuits



C = 30pF ± 10pF
(incl. stray capacitance)

Crystal Oscillator Mode

Driving from External Source

8-Bit CMOS Microcontroller

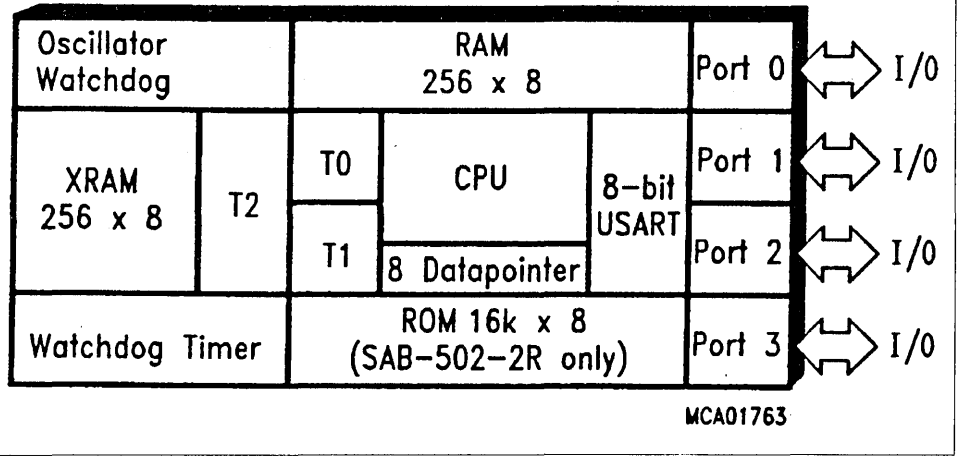
- Fully compatible to standard 8051 microcontroller
- Versions for 12 and 20 MHz operating frequency
- 16 K × 8 ROM (SAB-C502-2R only)
- 256 × 8 RAM
- 256 × 8 XRAM (additional on-chip RAM)
- Eight datapointers for indirect addressing of program and external data memory (including XRAM)
- Four 8-bit ports
- Three 16 -bit Timers / Counters (Timer 2 with Up/Down Counter feature)
- USART
- Six interrupt sources, two priority levels
- Programmable 15-bit Watchdog Timer
- Oscillator Watchdog
- Fast Power On Reset
- Power Saving Modes
- P-DIP-40 package and P-LCC-44 package
- Temperature ranges:
  SAB-C502 T_A: 0 °C to 70 °C
  SAF-C502 T_A: – 40 °C to 85 °C

The SAB-C502-L/C502-2R described in this document is compatible (not pin-compatible) with the SAB 80C52 and can be used for all present SAB 80C52 applications.

The SAB-C502-2R contains a non-volatile 16 K × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four ports, three 16-bit timers/counters, a six source, two priority level interrupt structure, a serial port and versatile fail save mechanisms. The SAB-C502-L/C502-2R incorporates 256 × 8 additional on-chip RAM called XRAM. For higher performance eight datapointers are implemented. The SAB-C502-L is identical, except that it lacks the program memory on chip. Therefore the term SAB-C502 refers to both versions within this specification unless otherwise noted.

4

Figure 45
SAB-C502

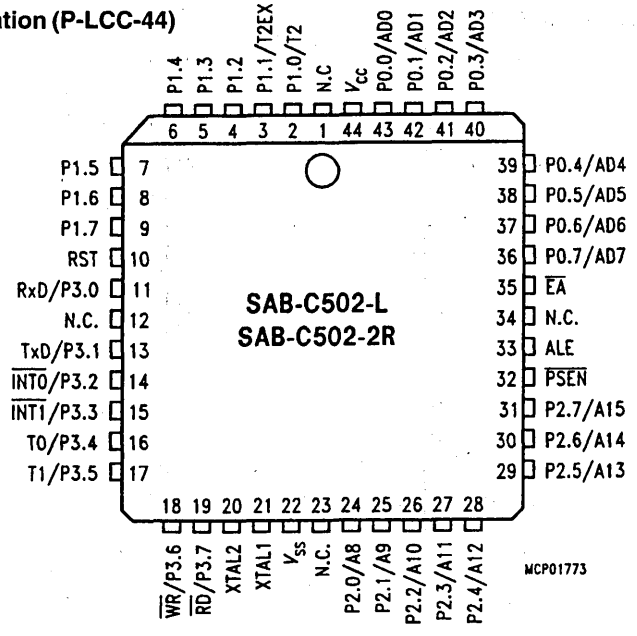


Ordering Information

Type	Package	Description (8-Bit CMOS microcontroller)
SAB-C502-LN SAB-C502-LP	P-LCC-44 P-DIP-40	for external memory 12 MHz
SAB-C502-2RN SAB-C502-2RP	P-LCC-44 P-DIP-40	with factory mask-programmable ROM, 12 MHz
SAB-C502-L20N SAB-C502-L20P	P-LCC-44 P-DIP-40	for external memory 20 MHz
SAB-C502-2R20N SAB-C502-2R20P	P-LCC-44 P-DIP-40	with factory mask-programmable ROM, 20 MHz
SAF-C502-LN SAF-C502-LP	P-LCC-44 P-DIP-40	for external ROM, 12 MHz, ext. temp. – 40 °C to 85 °C
SAF-C502-2RN SAF-C502-2RP	P-LCC-44 P-DIP-40	with factory mask-programmable ROM, 12 MHz, ext. temp. – 40 °C to 85 °C
SAF-C502-L20N SAF-C502-L20P	P-LCC-44 P-DIP-40	for external memory, 20 MHz, ext. temp. – 40 °C to 85 °C
SAF-C502-2R20N SAF-C502-2R20P	P-LCC-44 P-DIP-40	with factory mask-programmable ROM, 20 MHz, ext. temp. – 40 °C to 85 °C

Note: Extended temperature range – 40 °C to 110 °C (SAH-C502) on request.

Figure 46
Pin Configuration (P-LCC-44)



4

Figure 47
Pin Configuration (P-DIP-40)

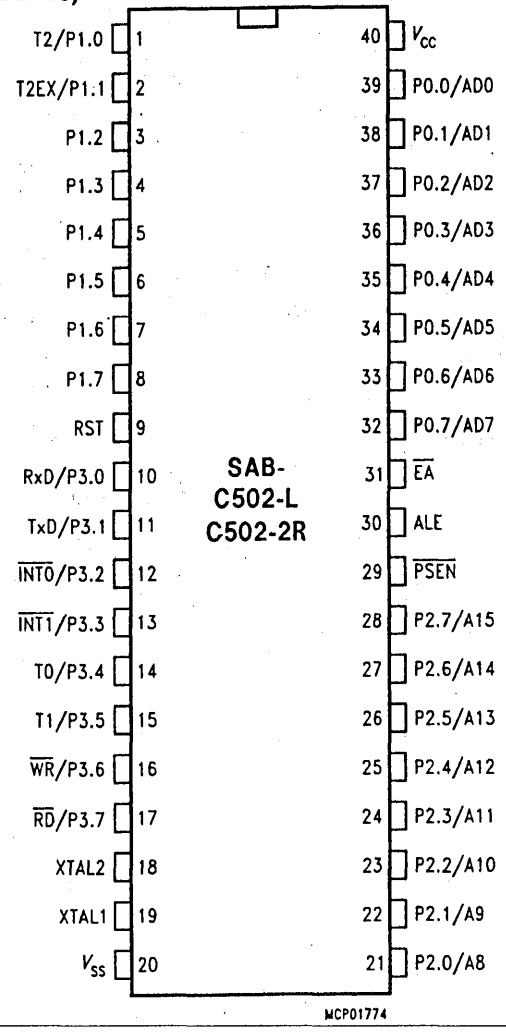
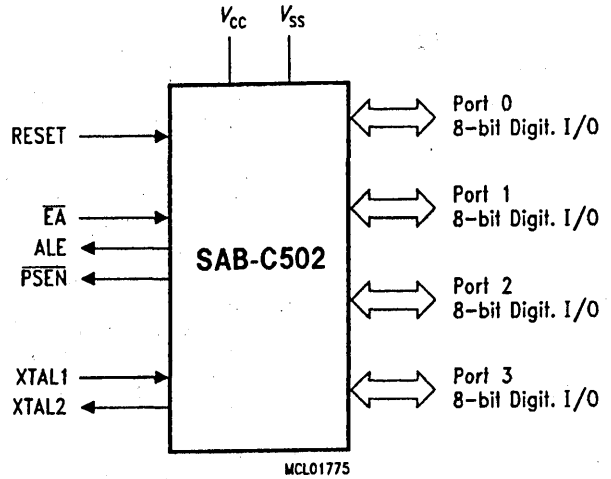


Figure 48
Logic Symbol



Pin Definitions and Functions

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
P1.7 – P1.0	2–9	1–8	I	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 1 also contains the timer 2 pins as secondary function. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate.
	2 3	1 2		The secondary functions are assigned to the pins of port 1, as follows: P1.0 T2Input to counter 2 P1.1 T2EXCapture - Reload trigger of timer 2 / Up-Down count

*) I = Input
O = Output

Pin Definitions and Functions (Continued)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
P3.0 – P3.7	11, 13–19	10–17	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows:
	11	10		P3.0 RxDreceiver data input (asynchronous) or data input/output (synchronous) of serial interface 0
	13	11		P3.1 TxDtransmitter data output (asynchronous) or clock output (synchronous) of the serial interface 0
	14	12		P3.2 $\overline{\text{INT0}}$interrupt 0 input/timer 0 gate control P3.3 $\overline{\text{INT1}}$interrupt 1 input/timer 1 gate control
	15	13		P3.4 T0counter 0 input
	16	14		P3.5 T1counter 1 input
	17	15		P3.6 $\overline{\text{WR}}$the write control signal latches the data byte from port 0 into the external data memory
	18	16		P3.7 $\overline{\text{RD}}$the read control signal enables the external data memory to port 0
	19	17		

*) I = Input
O = Output

Pin Definitions and Functions (Continued)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
XTAL1 XTAL2	21 20	19 18	– –	XTAL1 Input to the inverting oscillator amplifier and input to the internal clock generator circuits. **XTAL2** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise fall times specified in the AC characteristics must be observed.
P2.0 – P2.7	24–31	21–28	I/O	Port 2 is a bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pull-up resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
PSEN	32	29	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
RST	10	9	I	RESET A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{CC}.

4

Pin Definitions and Functions (Continued)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-DIP-40		
ALE	33	30	O	The **Address Latch Enable** output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
EA	35	31	I	External Access Enable When held at high level, instructions are fetched from the internal ROM (SAB-C502-2R only) when the PC is less than 4000H. When held at low level, the SAB-C502 fetches all instructions from external program memory. For the SAB-C502-L this pin must be tied low.
P0.0 – P0.7	43–36	39–32	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB-C502-2R if ROM-Protection was not enabled. External pull-up resistors are required during program verification.
V_{SS}	22	20	–	Circuit ground potential
V_{CC}	44	40	–	**Supply terminal** for all operating modes
N.C.	1, 12, 23, 24	–	–	No connection

*) I = Input
O = Output

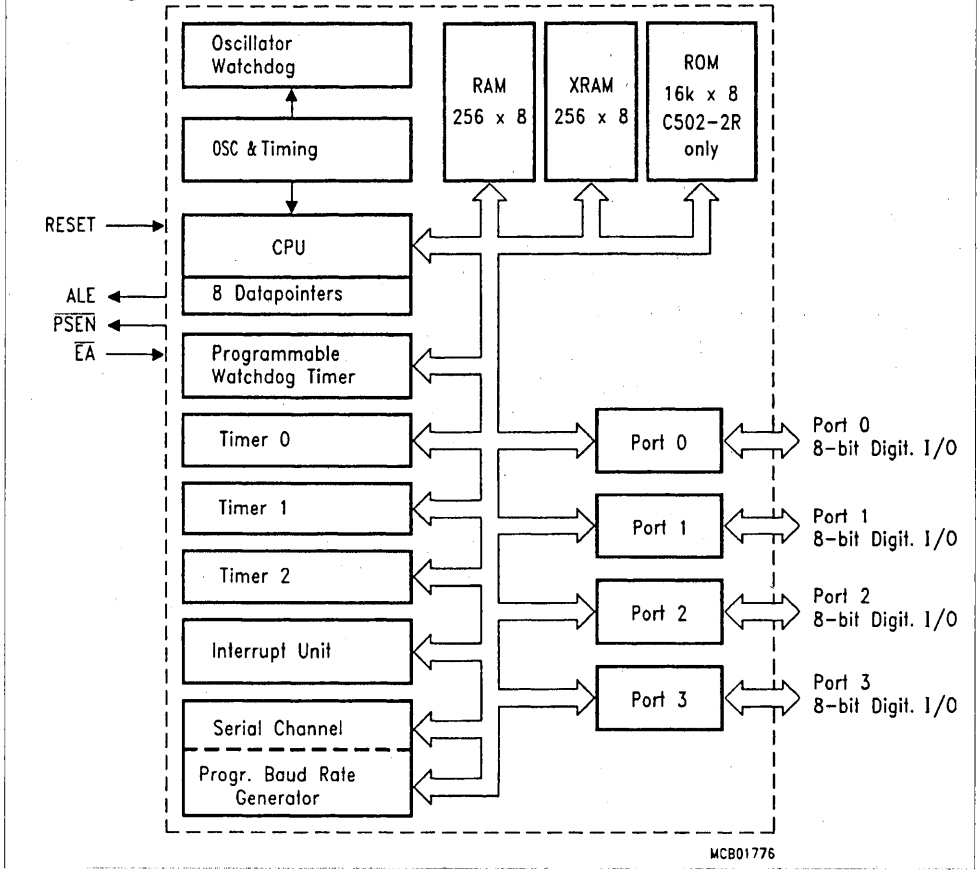
Functional Description

The SAB-C502 is fully compatible to standard 8051 microcontroller.

It is compatible with the SAB 80C52 but not pin-compatible. While maintaining all architectural and operational characteristics of the SAB 80C52 the SAB-C502 incorporates some enhancements in the Timer2 and Fail Save Mechanism Unit.

Figure 49 Block Diagram of the SAB-C502 shows a block diagram of the SAB-C502.

Figure 49
Block Diagram of the SAB-C502



4

CPU

The SAB-C502 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44 % one-byte, 41 % two-byte, and 15 % three-byte instructions. With a 12 MHz crystal, 58 % of the instructions execute in 1.0 µs (18 MHz : 667 ns).

Figure 50 Special Function Register PSW

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
Addr. 0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PS

Bit		Function
CY		**Carry Flag**
AC		**Auxiliary Carry Flag** (for BCD operations)
F0		**General Purpose Flag**
RS1	**RS0**	**Register Bank select control bits**
0	0	Bank 0 selected, data address 00H - 07H
0	1	Bank 1 selected, data address 08H - 0FH
1	0	Bank 2 selected, data address 10H - 17H
1	1	Bank 3 selected, data address 18H - 1FH
OV		**Overflow Flag**
F1		**General Purpose Flag**
P		**Parity Flag.** Set/cleared by hardware each instruction cycle to indicate an odd/ even number of "one" bits in the accumulator, i.e. even parity.

Reset value of PSW is 00H.

Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area.

The 36 special function register (SFR) include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. There are also 128 directly addressable bits within the SFR area.

All SFRs are listed in Table 11, Table 12 and Table 13. In Table 11 they are organized in numeric order of their addresses. In Table 12 they are organized in groups which refer to the functional blocks of the SAB-C502. Table 13 illustrates the contents of the SFRs.

Table 11
Special Function Register in numeric order of their addresses

Address	Register	Contents after Reset	Address	Register	Contents after Reset
80H	**P0** [1]	**0FFH**	**98H**	**SCON** [1]	**00H**
81H	SP	07H	99H	SBUF	XXH [2]
82H	DPL	00H	9AH	reserved	XXH [2]
83H	DPH	00H	9BH	reserved	XXH [2]
84H	(WDTL)		9CH	reserved	XXH [2]
85H	(WDTH)		9DH	reserved	XXH [2]
86H	WDTREL	00H	9EH	reserved	XXH [2]
87H	PCON	000X0000B [2]	9FH	reserved	XXH [2]
88H	**TCON** [1]	**00H**	**A0H**	**P2** [1]	**0FFH**
89H	TMOD	00H	A1H	reserved	XXH [2]
8AH	TL0	00H	A2H	reserved	XXH [2]
8BH	TL1	00H	A3H	reserved	XXH [2]
8CH	TH0	00H	A4H	reserved	XXH [2]
8DH	TH1	00H	A5H	reserved	XXH [2]
8EH	reserved	XXH [2]	A6H	reserved	XXH [2]
8FH	reserved	XXH [2]	A7H	reserved	XXH [2]
90H	**P1** [1]	**0FFH**	**A8H**	**IE** [1]	**0X000000B** [2]
91H	XPAGE	00H	A9H	reserved	XXH [2]
92H	DPSEL	XXXXX000B [2]	AAH	SRELL	0D9H
93H	reserved	XXH [2]	ABH	reserved	XXH [2]
94H	XCON	0F8H	ACH	reserved	XXH [2]
95H	reserved	XXH [2]	ADH	reserved	XXH [2]
96H	reserved	XXH [2]	AEH	reserved	XXH [2]
97H	reserved	XXH [2]	AFH	reserved	XXH [2]

[1] Bit-addressable Special Function Register

[2] X means that the value is indeterminate and the location is reserved

Table 11
Special Function Register in numeric order of their addresses (cont'd)

Address	Register	Contents after Reset	Address	Register	Contents after Reset
B0H	**P3** [1]	**0FFH**	**D8H**	**BAUD**	**0XXXXXXXB** [2]
B1H	SYSCON	XXXXXX01B [2]	D9H	reserved	XXH [2]
B2H	reserved	XXH [2]	DAH	reserved	XXH [2]
B3H	reserved	XXH [2]	DBH	reserved	XXH [2]
B4H	reserved	XXH [2]	DCH	reserved	XXH [2]
B5H	reserved	XXH [2]	DDH	reserved	XXH [2]
B6H	reserved	XXH [2]	DEH	reserved	XXH [2]
B7H	reserved	XXH [2]	DFH	reserved	XXH [2]
B8H	**IP** [1]	**XX000000B** [2]	**E0H**	**ACC** [1]	**00H**
B9H	reserved	XXH [2]	E1H	reserved	XXH [2]
BAH	SRELH	XXXXXX11B [2]	E2H	reserved	XXH [2]
BBH	reserved	XXH [2]	E3H	reserved	XXH [2]
BCH	reserved	XXH [2]	E4H	reserved	XXH [2]
BDH	reserved	XXH [2]	E5H	reserved	XXH [2]
BEH	reserved	XXH [2]	E6H	reserved	XXH [2]
BFH	reserved	XXH [2]	E7H	reserved	XXH [2]
C0H	**WDCON** [1]	**XXXX0000B** [2]	**E8H**	reserved	XXH [2]
C1H	reserved	XXH [2]	E9H	reserved	XXH [2]
C2H	reserved	XXH [2]	EAH	reserved	XXH [2]
C3H	reserved	XXH [2]	EBH	reserved	XXH [2]
C4H	reserved	XXH [2]	ECH	reserved	XXH [2]
C5H	reserved	XXH [2]	EDH	reserved	XXH [2]
C6H	reserved	XXH [2]	EEH	reserved	XXH [2]
C7H	reserved	XXH [2]	EFH	reserved	XXH [2]
C8H	**T2CON** [1]	**00H**	**F0H**	**B** [1]	**00H**
C9H	T2MOD	XXXXXXX0B [2]	F1H	reserved	XXH [2]
CAH	RC2L	00H	F2H	reserved	XXH [2]
CBH	RC2H	00H	F3H	reserved	XXH [2]
CCH	TL2	00H	F4H	reserved	XXH [2]
CDH	TH2	00H	F5H	reserved	XXH [2]
CEH	reserved	XXH [2]	F6H	reserved	XXH [2]
CFH	reserved	XXH [2]	F7H	reserved	XXH [2]
D0H	**PSW** [1]	**00H**	**F8H**	reserved	**XXH** [2]
D1H	reserved	XXH [2]	F9H	reserved	XXH [2]
D2H	reserved	XXH [2]	FAH	reserved	XXH [2]
D3H	reserved	XXH [2]	FBH	reserved	XXH [2]
D4H	reserved	XXH [2]	FCH	reserved	XXH [2]
D5H	reserved	XXH [2]	FDH	reserved	XXH [2]
D6H	reserved	XXH [2]	FEH	reserved	XXH [2]
D7H	reserved	XXH [2]	FFH	reserved	XXH [2]

[1] Bit-addressable Special Function Register

[2] X means that the value is indeterminate and the location is reserved

Table 12
Special Function Registers - Functional blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	0E0H[1)]	00H
	B	B-Register	0F0H[1)]	00H
	DPH	Data Pointer, High Byte	83H	00H
	DPL	Data Pointer, Low Byte	82H	00H
	DPSEL	Data pointer select register	92H	XXXX X000
	PSW	Program Status Word Register	0D0H[1)]	B[3)]
	SP	Stack Pointer	81H	00H
				07H
Interrupt System	IE	Interrupt Enable Register	0A8H[1)]	0X00 0000 B3)
	IP	Interrupt Priority Register	0B8H[1)]	XX00 0000 B[3)]
Ports	P0	Port 0	80H[1)]	0FFH
	P1	Port 1	90H[1)]	0FFH
	P2	Port 2	0A0H[1)]	0FFH
	P3	Port 3	0B0H[1)]	0FFH
XRAM	XPAGE	Page addr. reg. for XRAM	91H	00H
	XCON	XRAM startaddress (highbyte)	94H	0F8H
	SYSCON	XRAM control register	0B1H	XXXX XX01B[3)]
Serial Channels	PCON[2)]	Power Control Register	87H	00H
	SBUF	Serial Channel Buffer Reg.	99H	0XXH[3)]
	SCON	Serial Channel Control Reg.	98H[1)]	00H
	SRELL	Baudrate Generator Reloadvalue, Lowbyte	AAH	0D9H
	SRELH	Baudrate Generator Reloadvalue, Highbyte	BAH	XXXX XX11B[3)]
	BAUD	Baudrate Generator Enable Bit	0D8H[1)]	0XXX XXXXB[3)]
Timer 0/ Timer 1	TCON	Timer 0/1 Control Register	88H[1)]	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Timer 2	T2CON	Timer 2 Control Register	0C8H[1)]	00H
	T2MOD	Timer 2 Mode Register	0C9H	XXXX XXX0 B[3)]
	RC2L	Timer 2, Reload Capture Register, Low Byte	0CAH	00H
	RC2H	Timer 2, Reload Capture Register, High	0CBH	00H
	TH2	Byte	0CDH	00H
	TL2	Timer 2, High Byte	0CCH	00H
		Timer 2, Low Byte		
Watchdog	WDCON	Watchdog Timer Control Register	0C0H[1)]	XXXX 0000B[3)]
	WDTREL	Watchdog Timer Reload Reg.	86H	00H
Pow.Sav. Modes	PCON[2)]	Power Control Register	87H	000X 0000B[3)]

[1)] Bit-addressable special function registers

[2)] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.

[3)] X means that the value is indeterminate and the location is reserved

4

Table 13
Contents of SFR's, SFR's in numeric order

Address	Register	Bit 7	6	5	4	3	2	1	0
80H	P0								
81H	SP								
82H	DPL								
83H	DPH								
86H	WDTREL								
87H	PCON	SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE
88H	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89H	TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
8AH	TL0								
8BH	TL1								
8CH	TH0								
8DH	TH1								
90H	P1								
91H	XPAGE								
92H	DPSEL	–	–	–	–	–	.2	.1	.0
94H	XCON								
98H	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99H	SBUF								
A0H	P2								
A8H	IE	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
AAH	SRELL								

bit and byte addressable

not bit addressable

– = reserved

Table 13
Contents of SFRs, SFRs in numeric order (cont'd)

Address	Register	Bit 7	6	5	4	3	2	1	0
B0H	P3								
B1H	SYSCON	–	–	–	–	–	–	XMAP1	XMAP0
B8H	IP	–	PADC	PT2	PS	PT1	PX1	PT0	PX0
BAH	SRELL								
C0H	WDCON	–	–	–	–	OWDS	WDTS	WDT	SWDT
C8H	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
C9H	T2MOD	–	–	–	–	–	–	–	DCEN
CAH	RC2L								
CBH	RC2H								
CCH	TL2								
CDH	TH2								
D0H	PSW	CY	AC	F0	RS1	RS0	OV	F1	P
D8H	BAUD	BD	–	–	–	–	–	–	–
E0H	ACC								
F0H	B								

bit and byte addressable

not bit addressable

– = reserved

4

Timer/Counter 0 and 1

Timer/Counter 0 and 1 can be used in four operating modes as listed in Table 14:

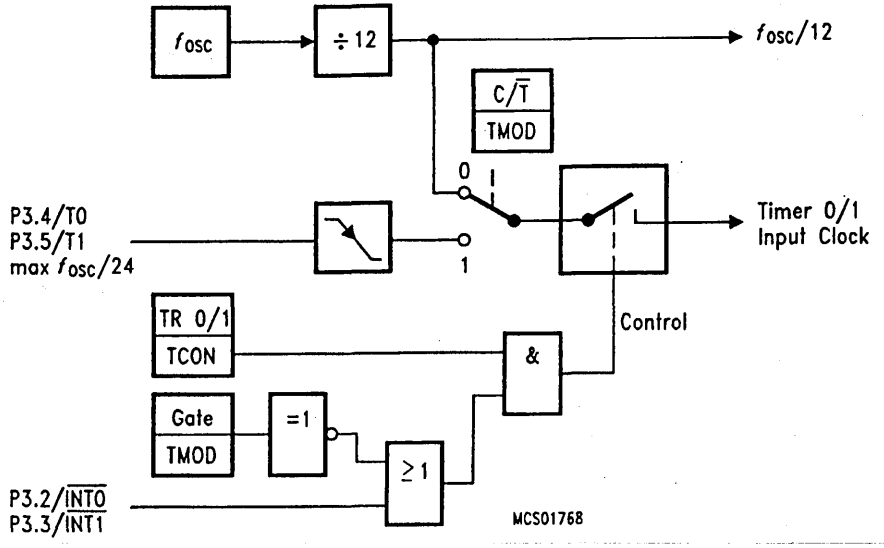
Table 14
Timer/Counter 0 and 1 operating modes

Mode	Description	TMOD				Input Clock	
		Gate	C/$\overline{T}$	M1	M0	internal	external (max)
0	8-bit timer/counter with a divide-by-32 prescaler	X	X	0	0	*f*osc/12 x 32	*f*osc/24 x 32
1	16-bit timer/counter	X	X	0	1	*f*osc/12	*f*osc/24
2	8-bit timer/counter with 8-bit auto-reload	X	X	1	0	*f*osc/12	*f*osc/24
3	Timer/counter 0 used as one 8-bit timer/counter and one 8-bit timer Timer 1 stops	X	X	1	1	*f*osc/12	*f*osc/24

In "timer" function (C/$\overline{T}$ = '0') the register is incremented every machine cycle. Therefore the count rate is *f*osc/12.

In "counter" function the register is incremented in response to a 1-to-0 transition at its corresponding external input pin (P3.4/T0, P3.5/T1). Since it takes two machine cycles to detect a falling edge the max. count rate is *f*osc/24. External inputs $\overline{INT0}$ and $\overline{INT1}$ (P3.2, P3.3) can be programmed to function as a gate to facilitate pulse width measurements. Figure 51 Timer/Counter 0 and 1 input clock logic illustrates the input clock logic.

Figure 51
Timer/Counter 0 and 1 input clock logic



Timer 2

Timer 2 is a 16-bit Timer/Counter with up/down count feature. It can operate either as timer or as an event counter which is selected by bit C/$\overline{T2}$ (T2CON.1). It has three operating modes as shown in Table 15.

Table 15
Timer/Counter 2 operating modes

Mode	T2CON			T2MOD	T2CON	P1.1/ T2EX	Remarks	Input Clock	
	R×CLK or T×CLK	CP/$\overline{RL2}$	TR2	DCEN	EXEN			internal	external (P1.0/T2)
16-bit Auto-reload	0	0	1	0	0	X	reload upon overflow	f_{osc}/12	max f_{osc}/24
	0	0	1	0	1	↓	reload trigger (falling edge)		
	0	0	1	1	X	0	Down counting		
	0	0	1	1	X	1	Up counting		
16-bit Capture	0	1	1	X	0	X	16-bit Timer/ Counter (only up-counting) capture TH2, TL2 → RC2H, RC2L	f_{osc}/12	max f_{osc}/24
	0	1	1	X	1	↓			
Baud Rate Generator	1	X	1	X	0	X	no overflow interrupt request (TF2) extra external interrupt ("Timer 2")	f_{osc}/2	max f_{osc}/24
	1	X	1	X	1	↓			
off	X	X	0	X	X	X	Timer 2 stops	–	–

Note: ↓ = falling edge

4

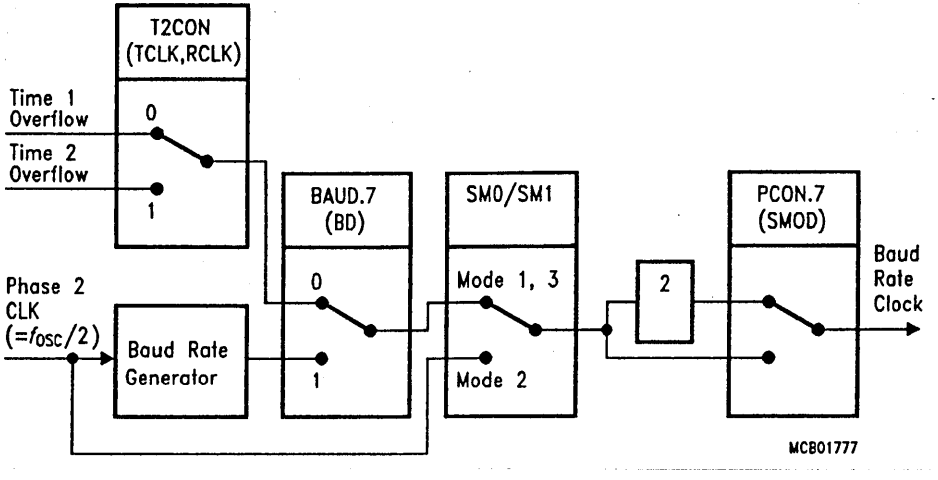
Serial Interface (USART)

The serial port is full duplex and can operate in four modes (one synchronous mode, three asynchronous modes) as illustrated in Table 16. Figure 3 illustrates the block diagram of Baudrate generation for the serial interface.

Table 16
USART operating modes

Mode	SCON		Baudrate	Description
	SM0	SM1		
0	0	0	*f*osc/12	Serial data enters and exits through RxD. TxD outputs the shift clock. 8-bit are transmitted/received (LSB first)
1	0	1	Timer 1/2 overflow rate or Baudrate Generator	8-bit UART 10 bits are transmitted (through TxD) or received (RxD)
2	1	0	*f*osc/32 or *f*osc/64	9-bit UART 11 bits are transmitted (TxD) or received (RxD)
3	1	1	Timer 1/2 overflow rate or Baudrate Generator	9-bit UART Like mode 2 except the variable baud rate

Figure 52
Block Diagram of Baud Rate Generation for Serial Interface



The possible baudrate can be calculated using the formulas given in Table 17.

Table 17
Baudrates

Baudrate derived from	Interface Mode	Baudrate
Oscillator	0 2	$f_{OSC}/12$ $(2^{SMOD} \times f_{OSC})/64$
Timer 1 (16-bit timer) (8-bit timer with 8-bit autoreload)	1,3 1,3	$(2^{SMOD} \times$ timer 1 overflow rate$)/32$ $(2^{SMOD} \times f_{OSC})/(32 \times 12 \times (256\text{-TH1}))$
Timer 2	1,3	$f_{OSC}/(32 \times (65536\text{-(RC2H, RC2L)}))$
Baudrate Generator	1,3	$(2^{SMOD} \times f_{OSC})/(64 \times (2^{10}\text{-SREL}))$

The internal baudrate generator consists of a free running 10-bit timer with $f_{OSC}/2$ input frequency.

The internal baudrate generator is selected by setting bit BD in SFR BAUD.

4

Additional On-Chip RAM - XRAM

The SAB-C502 contains another 256byte of On-Chip RAM additional to the 256bytes internal RAM. This RAM is called XRAM ('eXtended RAM') in this document.

The additional ON-Chip RAM is logically located in the external data memory range. The highbyte of the XRAM address range start address is programmable by SFR XCON (94H). The reset value of XCON is 0F8H (that is, XRAM address range F800H ... F8FFH).

The contents of the XRAM is not affected by a reset. After power up the contents is undefined, while it remains unchanged during and after reset as long as the power supply is not turned off. The XRAM is controlled by SFR SYSCON as shown in Table 18.

Table 18
Control of the XRAM

SFR SYSCON		Description
XMAP1	XMAP0	
0	1	Reset value. Access to XRAM is disabled. When cleared it can be set again only by a reset
0	0	XRAM enabled
1	0	XRAM enabled. The signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM

Because of the XRAM is used in the same way as external data memory the same instruction types must be used for accessing the XRAM. A general overview gives Table 19.

Table 19
Accessing the XRAM

Instruction using	Instruction	Remarks
DPTR	MOVX A @DPTR MOVX @ DPTR,A	Normally the use of these instructions would use a physically external memory. However, in the SAB-C502 the XRAM is accessed if it is enabled.
R0/R1 (page mode)	MOVX A, @Ri MOVX@Ri,A	Normally Port 2 serves as page register. However, the distinction, whether Port 2 is as general purpose I/O or as "page address" is made by the external design. Hence a special SFR XPAGE is implemented the serve the same function for the XRAM as Port 2 for external data memory.

Note: When writing the page address (in page mode) at Port2 the value is also written in XPAGE. However when writing XPAGE the value at PORT2 is not changed!

The behavior of Port0/Port2 and $\overline{RD}/\overline{WR}$ during MOVX accesses is shown in Table 20.

Table 20
Behavior of P0/P2 and $\overline{\text{RD}}$/$\overline{\text{WR}}$ during MOVX accesses

		$\overline{\text{EA}}$ = 0			$\overline{\text{EA}}$ = 1		
		XMAP1, XMAP0			XMAP1, XMAP0		
		00	10	X1	00	10	X1
MOVX @DPTR	DPTR outside XRAM address range (DPH ≠ XCON)	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used
	DPTR within XRAM address range (DPH = XCON)	a) P0/P2 → Bus ($\overline{\text{WR}}$-Data only) b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c) XRAM is used	a) P0/P2 → Bus ($\overline{\text{WR}}$-Data only) b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) XRAM is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c) XRAM is used	a) P0/P2 → Bus ($\overline{\text{WR}}$-Data only) b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) XRAM is used	a) P0/P2 → Bus b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used
MOVX @Ri	XPAGE outside XRAM addr. page range (XPAGE ≠ XCON)	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used
	XPAGE within XRAM addr. page range (XPAGE = XCON)	a) P0 → Bus ($\overline{\text{WR}}$-Data only) P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c) XRAM is used	a) P0 → Bus ($\overline{\text{WR}}$-Data only) P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) XRAM is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a) P0/P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c) XRAM is used	a) P0 → Bus ($\overline{\text{WR}}$-Data only) P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) XRAM is used	a) P0 → Bus P2 → I/O b) $\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used

Eight Datapointers for Faster External Bus Access

The SAB-C502 contains a set of eight 16-bit-Datapointer (DPTR) from which the actual DPTR can be selected.

This means that the user's program may keep up to eight 16-bit addresses resident in these registers, but only one register at a time is selected to be the datapointer. Thus the DPTR in turn is accessed (or selected) via indirect addressing. This indirect addressing is done through a special function register (SFR) called DPSEL (data pointer select register, Bits 0 to 2). All instructions of the SAB-C502 which handle the DPTR therefore affect only one of the eight pointers which is addressed by DPSEL at that very moment.

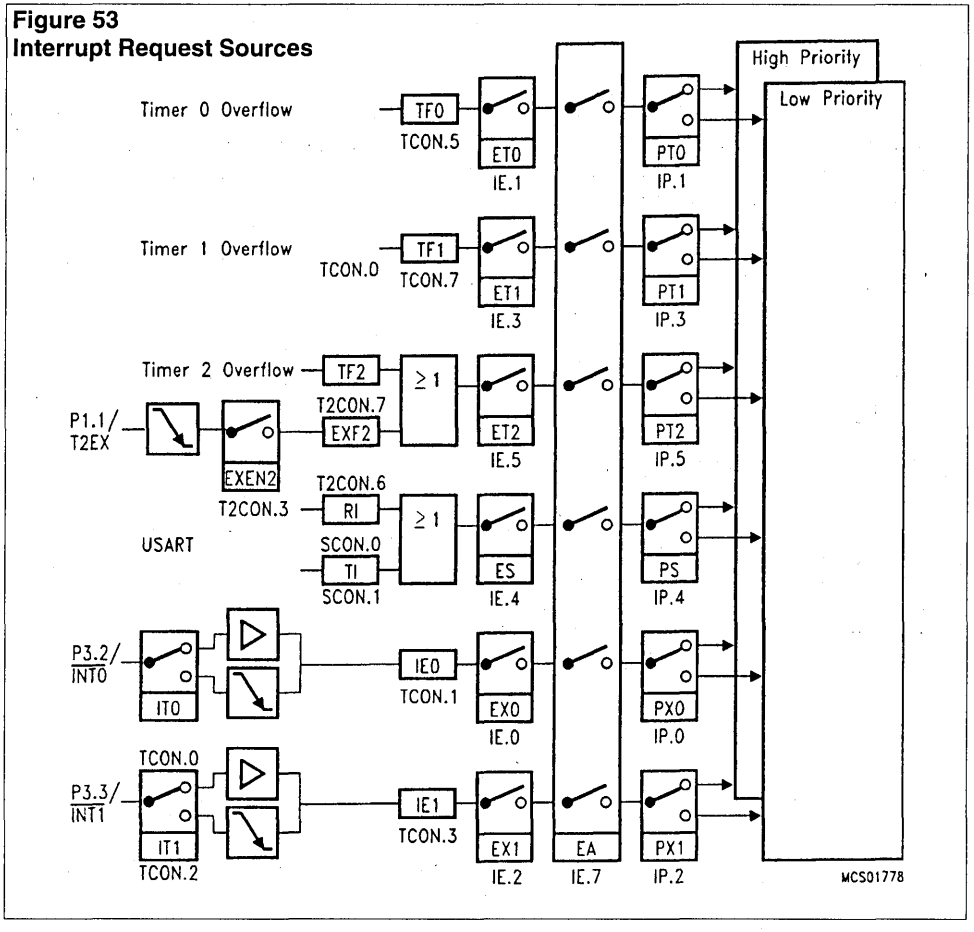
A 3-bit field in SFR DPSEL points to the currently used DPTRx:

DPSEL .2	.1	.0	selected DPTR
0	0	0	DPTR 0
0	0	1	DPTR 1
0	1	0	DPTR 2
0	1	1	DPTR 3
1	0	0	DPTR 4
1	0	1	DPTR 5
1	1	0	DPTR 6
1	1	1	DPTR 7

Interrupt System

The SAB-C502 provides 6 interrupt sources with two priority levels. Figure 53 Interrupt Request Sources gives a general overview of the interrupt sources and illustrates the request and control flags.

Figure 53
Interrupt Request Sources



4

Table 21
Interrupt sources and their corresponding interrupt vectors

Source (Request Flags)	Vector	Vector Address
IE0	External interrupt 0	0003H
TF0	Timer 0 interrupt	000BH
IE1	External interrupt 1	0013H
TF1	Timer 1 interrupt	001BH
RI + TI	Serial port interrupt	0023H
TF2 + EXF2	Timer 2 interrupt	002BH

A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

If two requests of different priority level are received simultaneously, the request of higher priority is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as shown in Table 22.

Table 22
Interrupt priority-within-level

Interrupt Source		Priority
External Interrupt 0,	IE0	High
Timer 0 Interrupt,	TF0	
External Interrupt 1,	IE1	↓
Timer 1 Interrupt,	TF1	
Serial Channel,	RI or TI	
Timer 2 Interrupt,	TF2 or EXF2	Low

Fail Safe Mechanisms

The SAB-C502 offers enhanced fail safe mechanisms, which allow an automatic recovery from software upset or hardware failure.

1) Watchdog Timer (15 bit, WDT)

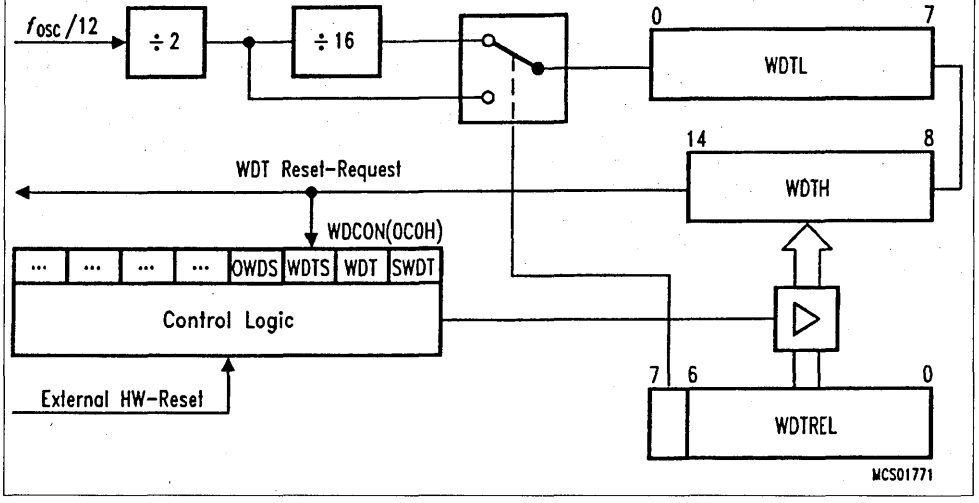
2) Oscillator Watchdog (OWD)

1) Watchdog Timer (WDT)

The Watchdog Timer in the SAB-C502 is a 15-bit timer, which is incremented by a count rate of either $f_{CYCLE}/2$ or $f_{CYCLE}/32$ ($f_{CYCLE} = f_{OSC}/12$). That is, the machine clock is divided by a series of arrangement of two prescalers, a divide-by-two and a divide-by-16 prescaler. The latter is enabled by setting bit WDTREL.7.

Figure 54 Block Diagram of the Programmable Watchdog Timer shows the block diagram of the programmable Watchdog Timer.

Figure 54
Block Diagram of the Programmable Watchdog Timer



4

Starting and refreshing the WDT

Table 23 gives an overview how to start and refresh the WDT. The mentioned bits are located in SFR WDCON.

Table 23
Starting and Refreshing the WDT

Function	Example		Remarks
Starting WD	SETB	SWDT	Cannot be stopped during active mode of the device. WDT is halted during idle mode, power down mode or the oscillator watchdog reset is active.
Refreshing WD	SETB SETB	WDT SWDT	Double instruction sequence (setting bit WDT and SWDT consecutively) to increase system security.

Watchdog reset and watchdog status flag (WDTS)

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state 7FFCH. The duration of the reset signal then depends on the prescaler selection (either 8 or 128 cycles). This internal reset differs from an external one in so far as the Watchdog Timer is not disabled and bit WDTS (SFR WDCON) is set. The WDTS is a flip-flop, which is set by a Watchdog Timer reset and can be cleared by an external hardware reset. Bit WDTS allows the software to examine from which source the reset was activated. The bit WDTS can also be cleared by software.

2) Oscillator Watchdog (OWD)

The OWD consists of an internal RC oscillator which provides the reference frequency for the comparison with the frequency of the on-chip oscillator.

Figure 55 Functional Block Diagram of the Oscillator Watchdog shows the block diagram of the oscillator watchdog unit while Table 14 shows the effect when the OWD becomes active/inactive.

Note: The OWD is always enabled!

Figure 55
Functional Block Diagram of the Oscillator Watchdog

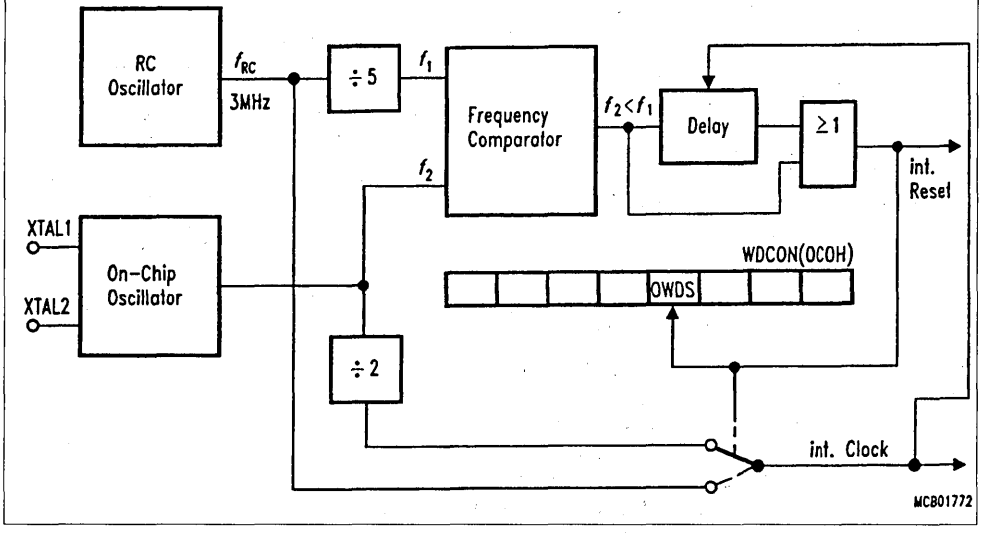


Table 24
Effects of the OWD

Conditions	Effect
$f_{OSC} < f_{RC}/5$	Switch input of internal clock system to RC oscillator output Activating internal reset at the same time (reset sequence is clocked by RC-oscillator). Exception from effects of a Hardware Reset: Watchdog Timer Status Flag, WDTS is not reset Oscillator Watchdog Status Flag, OWDS is set
$f_{OSC} > f_{RC}/5$	Input of internal clock system is $f_{OSC}/2$. When failure condition ($f_{OSC} < f_{RC}/5$) disappears the part executes a final reset phase of typ. 1 ms in order to allow the external oscillator to stabilize.

Power Saving Modes

Two power down modes are available, the Idle Mode and the Power Down Mode.

The bits PDE, PDS and IDLE, IDLS select the Power Down mode or the idle mode, respectively. If the Power Down mode and the idle mode are set at the same time, Power Down takes precedence. Table 25 gives a general overview of the power saving modes.

Table 25
Entering and leaving the power saving modes

Mode	Entering Example	Leaving by	Remarks
Idle mode	ORL PCON, #01H ORL PCON, #20H	– enabled interrupt – Hardware Reset	CPU is gated off CPU status registers maintain their data. Peripherals are active Double instruction sequence
Power Down Mode	ORL PCON, #02H ORL PCON, #40H	Hardware Reset	Oscillators are stopped. Contents of on-chip RAM and SFR's are maintained (leaving Power Down Mode means redefinition of SFR's contents.) Double instruction sequence

In the Power Down mode of operation, V_{CC} can be reduced to minimize power consumption. It must be ensured, however, that V_{CC} is not reduced before the Power Down mode is invoked, and that V_{CC} is restored to its normal operating level, before the Power Down mode is terminated. The reset signal that terminates the Power Down mode also restarts the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

Absolute Maximum Ratings

Ambient temperature under bias (T_A)..... – 40 °C to +85 °C
Storage temp. (T_{ST}) ..... – 65 °C to +150 °C
Voltage on V_{CC} pins with respect to ground (V_{SS})..... – 0.5 V to 6.5 V
Voltage on any pin with respect to ground (V_{SS}) ..... – 0.5 V to V_{CC} + 0.5 V
Input current on any pin during overload condition..... – 10 mA to + 10 mA
Absolute sum of all input currents during overload condition..... I100 mA I
Power dissipation..... TBD

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.*

4

DC Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V; T_A = 0 to + 70 °C for the SAB-C502
T_A = – 40 to + 85 °C for the SAF-C502

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, RESET)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage ($\overline{RESET}$)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except $\overline{EA}$, RESET, XTAL1)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL1	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	
Input high voltage to RESET, $\overline{EA}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (ports 2, 3)	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA[1]
Output low voltage (port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA[1]
Output high voltage (ports 2, 3)	V_{OH}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 80 µA I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 800 µA[2], I_{OH} = – 80 µA[2]
Logic 0 input current (ports 1, 2, 3)	I_{IL}	– 10	– 50	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (ports 2, 3)	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{EA}$, P1)	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 °C
Power supply current:					
Active mode, 12 MHz[7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V,[4]
Idle mode, 12 MHz[7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V,[5]
Active mode, 20 MHz[7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V,[4]
Idle mode, 20 MHz[7]	I_{CC}	–	TBD	mA	V_{CC} = 5 V,[5]
Power Down Mode	I_{PD}	–	50	µA	V_{CC} = 2 ... 5.5 V,[3]

[1] Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and port 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

[2] Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall bellow the 0.9 V_{CC} specification when the address lines are stabilizing.

[3] I_{PD} (Power Down Mode) is measured under following conditions:
$\overline{EA}$ = RESET = Port0 = V_{CC}; XTAL2 = N.C.;
XTAL1 = V_{SS}; all other pins are disconnected.

[4] I_{CC} (active mode) is measured with:
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
$\overline{EA}$ = Port0 = RESET = V_{CC}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

[5] I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
RESET = $\overline{EA}$ = V_{SS}; Port0 = V_{CC}; all other pins are disconnected;

[7] $I_{CC\ Max}$ at other frequencies is given by:
active mode:TBD
idle mode:TBD
where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V.

4

AC Characteristics for SAB-C502-L / C502-2R

Please refer to the SAB-C501 AC Characteristics and Waveforms for the 12 and 20MHz AC Characteristics and Waveforms for the SAB-C502.

8-Bit CMOS Microcontroller

- Compatible to standard 8051 microcontroller
- Versions for 12 and 20 MHz operating frequency
- 8 K × 8 ROM (SAB-C503-1R only)
- 256 × 8 RAM
- Four 8-bit ports, (including one input port for digital or analog input)
- Three 16-bit Timers / Counters (Timer 2 with Up/Down Counter feature)
- USART
- Genuine 10-bit A/D Converter with 8 multiplexed inputs
- Seven interrupt sources, two priority levels
- Programmable 15-bit Watchdog Timer
- Oscillator Watchdog
- Fast Power On Reset
- Power Saving Modes
- P-LCC-44 package

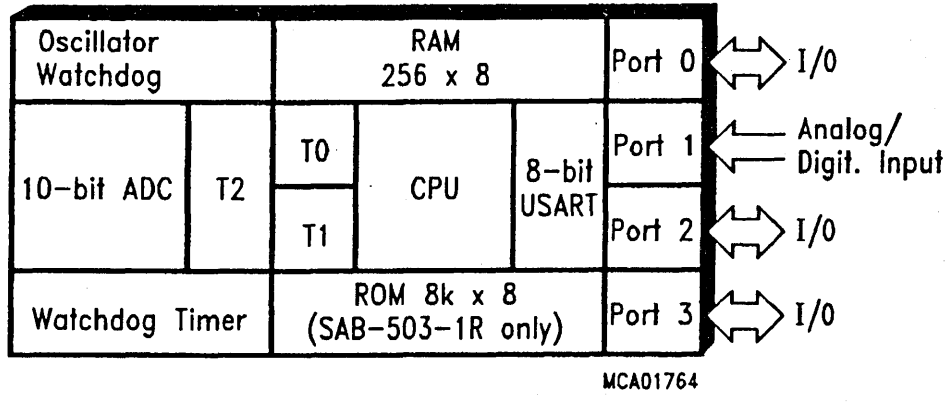
- Temperature ranges:
  SAB-C503 T_A: 0 °C to 70 °C
  SAF-C503 T_A: – 40 °C to 85 °C

The SAB-C503-L/C503-1R described in this document is compatible (not pin-compatible) with the SAB 80C32/C52 and can be used for all present SAB 80C52 applications.

The SAB-C503-1R contains a non-volatile 8 K x 8 read-only program memory, a volatile 256 x 8 read/write data memory, four ports, three 16-bit timers/counters, a seven source, two priority level interrupt structure, a serial port, versatile fail save mechanisms and a genuine 10-bit A/D Converter. The SAB-C503-L is identical, except that it lacks the program memory on chip. Therefore the term SAB-C503 refers to both versions within this specification unless otherwise noted.

4

Figure 56
SAB-C503

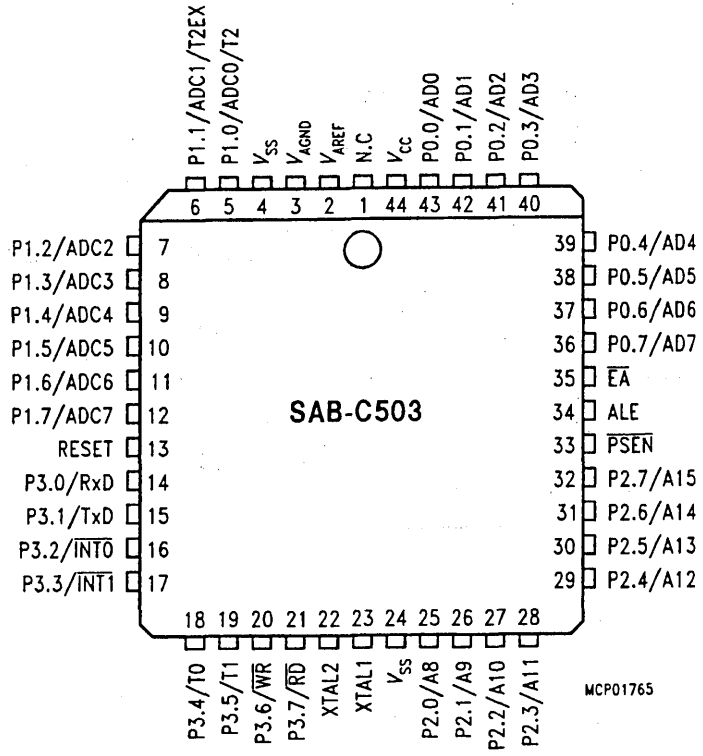


Ordering Information

Type	Package	Description (8-Bit CMOS microcontroller)
SAB-C503-LN	P-LCC-44	for external memory 12 MHz
SAB-C503-1RN	P-LCC-44	with factory mask-programmable ROM, 12 MHz
SAB-C503-L20N	P-LCC-44	for external memory 20 MHz
SAB-C503-1R20N	P-LCC-44	with factory mask-programmable ROM, 20 MHz
SAF-C503-LN	P-LCC-44	for external ROM, 12 MHz, ext. temp. – 40 °C to 85 °C
SAF-C503-1RN	P-LCC-44	with factory mask-programmable ROM, 12 MHz, ext. temp. – 40 °C to 85 °C
SAF-C503-L20N	P-LCC-44	for external memory, 20 MHz, ext. temp. – 40 °C to 85 °C
SAF-C503-1R20N	P-LCC-44	with factory mask-programmable ROM, 20 MHz, ext. temp. – 40 °C to 85 °C

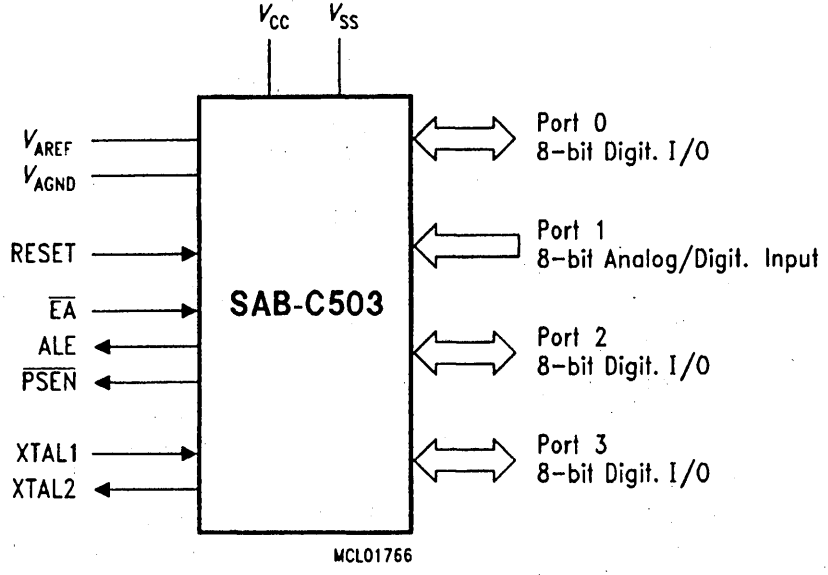
Note: extended temperature range – 40 °C to 110 °C (SAH-C503) on request.

Figure 57
Pin Configuration (P-LCC-44)



4

Figure 58
Logic Symbol



Pin Definitions and Functions

Symbol	Pin Number P-LCC-44	I/O*)	Function
P1.7 – P1.0	12–5	I	**Port 1** is an 8-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high low voltages, and for the multiplexed analog inputs of the A/D-Converter, simultaneously. Port 1 also contains the timer 2 pins as secondary function. The secondary functions are assigned to the pins of port 1, as follows: P1.0 AN0 / T2Analog input channel 0 Input to counter 2 P1.1 AN1/T2EXAnalog input channel 1 Capture - Reload trigger of timer 2 / Up-Down count P1.2 AN2Analog input channel 2 P1.3 AN3Analog input channel 3 P1.4 AN4Analog input channel 4 P1.5 AN5Analog input channel 5 P1.6 AN6Analog input channel 6 P1.7 AN7Analog input channel 7
P3.0 – P3.7	14–21	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state they can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins which are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – RxD (P3.0):receiver data input (asynchronous) or data input/output (synchronous) of serial interface – TxD (P3.1):transmitter data output (asynchronous) or clock output (synchronous) of serial interface – $\overline{\text{INT0}}$(P3.2):interrupt 0 input/timer 0 gate control – $\overline{\text{INT1}}$(P3.3):interrupt 1 input/timer 1 gate control – T0 (P3.4):counter 0 input – $\overline{\text{T1}}$ (P3.5):counter 1 input – $\overline{\text{WR}}$ (P3.6):the write control signal latches the data byte from port 0 into the external data memory – $\overline{\text{RD}}$ (P3.7):the read control signal enables the external data memory to port 0

*)I = Input
O = Output

Pin Definitions and Functions (Continued)

Symbol	Pin Number P-LCC-44	I/O*)	Function
XTAL2	22	–	**XTAL2** Output of the inverting oscillator amplifier.
XTAL1	23	–	XTAL1 Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times a well as rise/fall times specified in the AC characteristics must be observed.
P2.0 – P2.7	25–32	I/O	Port 2 is a bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state they can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pull-up resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
PSEN	33	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
RESET	13	I	RESET A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{CC}.

*)I = Input
O = Output

Pin Definitions and Functions (Continued)

Symbol	Pin Number P-LCC-44	I/O*)	Function
ALE	34	O	The **Address Latch Enable** output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
EA	35	I	External Access Enable When held at high level, instructions are fetched from the internal ROM (SAB-C503-1R only) when the PC is less than 2000H. When held at low level, the SAB-C503 fetches all instructions from external program memory. For the SAB-C503-L this pin must be tied low.
P0.0 – P0.7	43–36	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB-C503-1R. External pull-up resistors are required during program verification.
V_{AREF}	2		**Reference voltage** for the A/D converter.
V_{AGND}	3		**Reference ground** for the A/D converter.
V_{SS}	4, 24	–	Circuit ground potential
V_{CC}	44	–	**Supply terminal** for all operating modes
N.C.	1	–	No connection

*)I = Input
O = Output

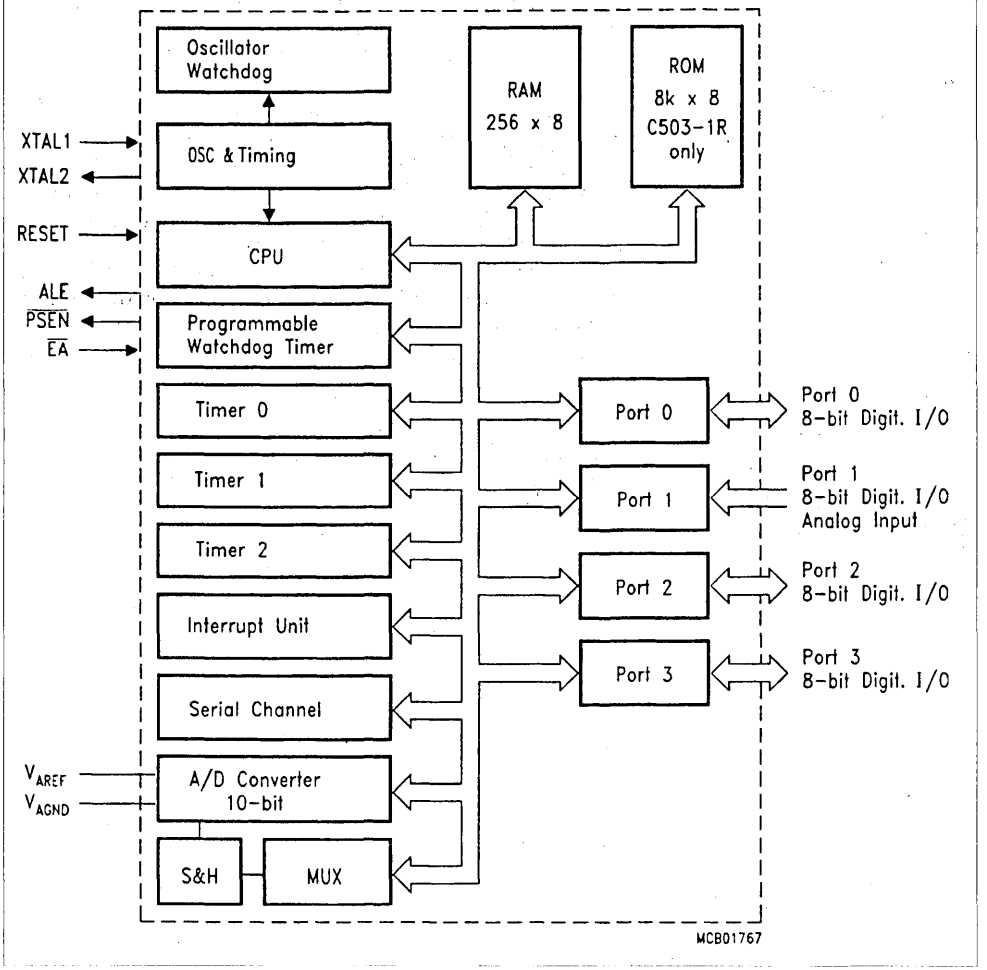
Functional Description

The SAB-C503 is fully compatible to standard 8051 microcontroller.

It is compatible with the SAB 80C52 but not pin-compatible. While maintaining all architectural and operational characteristics of the SAB 80C52 the SAB-C503 incorporates a genuine 10-bit A/D Converter as well as some enhancements in the Timer2 and Fail Save Mechanism Unit.

Figure 59 Block Diagram of the SAB-C503 shows a block diagram of the SAB-C503.

Figure 59
Block Diagram of the SAB-C503



CPU

The SAB-C503 is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44 % one-byte, 41 % two-byte, and 15 % three-byte instructions. With a 12 MHz crystal, 58 % of the instructions executed in 1.0 µs (20 MHz : 600 ns).

Figure 60 Special Function Register PSW

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
Addr. 0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PS

Bit		Function
CY		**Carry Flag**
AC		**Auxiliary Carry Flag** (for BCD operations)
F0		**General Purpose Flag**
RS1	**RS0**	**Register Bank select control bits**
0	0	Bank 0 selected, data address 00H - 07H
0	1	Bank 1 selected, data address 08H - 0FH
1	0	Bank 2 selected, data address 10H - 17H
1	1	Bank 3 selected, data address 18H - 1FH
OV		**Overflow Flag**
F1		**General Purpose Flag**
P		**Parity Flag** Set/cleared by hardware each instruction cycle to indicate an odd/ even number of "one" bits in the accumulator, i.e. even parity.

Reset value of PSW is 00H.

Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area.

The 33 special function registers (SFRs) include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. There are also 128 directly addressable bits within the SFR area.

All SFRs are listed in Table 26, Table 27 and Table 28.

In Table 26 they are organized in numeric order of their addresses. In Table 27 they are organized in groups which refer to the functional blocks of the SAB-C503. Table 28 illustrates the contents of the SFRs.

Table 26
Special Function Registers in numeric order of their addresses

Address	Register	Contents after Reset	Address	Register	Contents after Reset
80H	**P0** [1]	**0FFH**	**98H**	**SCON** [1]	**00H**
81H	SP	07H	99H	SBUF	XXH [2]
82H	DPL	00H	9AH	reserved	XXH [2]
83H	DPH	00H	9BH	reserved	XXH [2]
84H	(WDTL) [3]	00H	9CH	reserved	XXH [2]
85H	(WDTH) [3]	00H	9DH	reserved	XXH [2]
86H	WDTREL	00H	9EH	reserved	XXH [2]
87H	PCON	000X0000B [2]	9FH	reserved	XXH [2]
88H	**TCON** [1]	**00H**	**A0H**	**P2** [1]	**0FFH**
89H	TMOD	00H	A1H	reserved	XXH [2]
8AH	TL0	00H	A2H	reserved	XXH [2]
8BH	TL1	00H	A3H	reserved	XXH [2]
8CH	TH0	00H	A4H	reserved	XXH [2]
8DH	TH1	00H	A5H	reserved	XXH [2]
8EH	reserved	XXH [2]	A6H	reserved	XXH [2]
8FH	reserved	XXH [2]	A7H	reserved	XXH [2]
90H	**P1** [1]	**0FFH**	**A8H**	**IE** [1]	**00H**
91H	reserved	00H	A9H	reserved	XXH [2]
92H	reserved	XXH [2]	AAH	reserved	XXH [2]
93H	reserved	XXH [2]	ABH	reserved	XXH [2]
94H	reserved	XXH [2]	ACH	reserved	XXH [2]
95H	reserved	XXH [2]	ADH	reserved	XXH [2]
96H	reserved	XXH [2]	AEH	reserved	XXH [2]
97H	reserved	XXH [2]	AFH	reserved	XXH [2]

[1] Bit-addressable Special Function Register

[2] X means that the value is indeterminate and the location is reserved

[3] () ... SFR not user accessible

Table 26
Special Function Register in numeric order of their addresses (cont'd)

Address	Register	Contents after Reset	Address	Register	Contents after Reset
B0H	**P3** [1]	**0FFH**	**D8H**	**ADCON0** [1]	00H
B1H	reserved	XXH [2]	D9H	ADDATH	00H
B2H	reserved	XXH [2]	DAH	ADDATL	00H
B3H	reserved	XXH [2]	DBH	reserved	XXH [2]
B4H	reserved	XXH [2]	DCH	ADCON1	0XXX000B [2]
B5H	reserved	XXH [2]	DDH	reserved	XXH [2]
B6H	reserved	XXH [2]	DEH	reserved	XXH [2]
B7H	reserved	XXH [2]	DFH	reserved	XXH [2]
B8H	**IP** [1]	**X0000000B** [2]	**E0H**	**ACC** [1]	**00H**
B9H	reserved	XXH [2]	E1H	reserved	XXH [2]
BAH	reserved	XXH [2]	E2H	reserved	XXH [2]
BBH	reserved	XXH [2]	E3H	reserved	XXH [2]
BCH	reserved	XXH [2]	E4H	reserved	XXH [2]
BDH	reserved	XXH [2]	E5H	reserved	XXH [2]
BEH	reserved	XXH [2]	E6H	reserved	XXH [2]
BFH	reserved	XXH [2]	E7H	reserved	XXH [2]
C0H	**WDCON** [1]	**XXXX0000B** [2]	**E8H**	reserved	XXH [2]
C1H	reserved	XXH [2]	E9H	reserved	XXH [2]
C2H	reserved	XXH [2]	EAH	reserved	XXH [2]
C3H	reserved	XXH [2]	EBH	reserved	XXH [2]
C4H	reserved	XXH [2]	ECH	reserved	XXH [2]
C5H	reserved	XXH [2]	EDH	reserved	XXH [2]
C6H	reserved	XXH [2]	EEH	reserved	XXH [2]
C7H	reserved	XXH [2]	EFH	reserved	XXH [2]
C8H	**T2CON** [1]	**00H**	**F0H**	**B** [1]	**00H**
C9H	T2MOD	XXXXXXX0B [2]	F1H	reserved	XXH [2]
CAH	RC2L	00H	F2H	reserved	XXH [2]
CBH	RC2H	00H	F3H	reserved	XXH [2]
CCH	TL2	00H	F4H	reserved	XXH [2]
CDH	TH2	00H	F5H	reserved	XXH [2]
CEH	reserved	XXH [2]	F6H	reserved	XXH [2]
CFH	reserved	XXH [2]	F7H	reserved	XXH [2]
D0H	**PSW** [1]	**00H**	**F8H**	reserved	**XXH** [2]
D1H	reserved	XXH [2]	F9H	reserved	XXH [2]
D2H	reserved	XXH [2]	FAH	reserved	XXH [2]
D3H	reserved	XXH [2]	FBH	reserved	XXH [2]
D4H	reserved	XXH [2]	FCH	reserved	XXH [2]
D5H	reserved	XXH [2]	FDH	reserved	XXH [2]
D6H	reserved	XXH [2]	FEH	reserved	XXH [2]
D7H	reserved	XXH [2]	FFH	reserved	XXH [2]

[1] Bit-addressable Special Function Register

[2] X means that the value is indeterminate and the location is reserved

4

Table 27
Special Function Registers - Functional blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	0E0H [1]	00H
	B	B-Register	0F0H [1]	00H
	DPH	Data Pointer, High Byte	83H	00H
	DPL	Data Pointer, Low Byte	82H	00H
	PSW	Program Status Word Register	0D0H [1]	00H
	SP	Stack Pointer	81H	07H
Interrupt System	IE	Interrupt Enable Register	0A8H [1]	00H
	IP	Interrupt Priority Register	0B8H [1]	00H
Ports	P0	Port 0	80H [1]	0FFH
	P1	Port 1, Analog/Digital Input	90H [1]	0XXH [3]
	P2	Port 2	0A0H [1]	0FFH
	P3	Port 3	0B0H [1]	0FFH
A/D-Converter	ADCON0	A/D Converter Control Register 0	0D8H [1]	00H
	ADCON1	A/D Converter Control Register 1	0DCH	0XXX X000B [3]
	ADDATH	A/D Converter Data Register High Byte	0D9H	00H
	ADDATL	A/D Converter Data Register Low Byte	0DAH	00H
Serial Channels	PCON [2]	Power Control Register	87H	00H
	SBUF	Serial Channel Buffer Reg.	99H	0XXH [3]
	SCON	Serial Channel 0 Control Reg.	98H [1]	00H
Timer 0 / Timer 1	TCON	Timer 0/1 Control Register	88H [1]	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Timer 2	T2CON	Timer 2 Control Register	0C8H [1]	00H
	T2MOD	Timer 2 Mode Register	0C9H	XXXX XXX0B [3]
	RC2H	Timer 2 Reload Capture Reg., High	0CBH	00H
	RC2L	Byte	0CAH	00H
	TH2	Timer 2 Reload Capture Reg., Low Byte	0CDH	00H
	TL2	Timer 2, High Byte	0CCH	00H
		Timer 2, Low Byte		
Watchdog	WDCON	Watchdog Timer Control Register	0C0H [1]	XXXX 0000B [3]
	WDTREL	Watchdog Timer Reload Reg.	86H	00H
Pow.Sav. Modes	PCON	Power Control Register	87H	000X0000B [3]

[1] Bit-addressable special function registers

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks

[3] X means that the value is indeterminate and the location is reserved

Table 28
Contents of SFRs, SFRs in numeric order

Address	Register	Bit 7	6	5	4	3	2	1	0
80H	P0								
81H	SP								
82H	DPL								
83H	DPH								
86H	WDTREL								
87H	PCON	SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE
88H	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89H	TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
8AH	TL0								
8BH	TL1								
8CH	TH0								
8DH	TH1								
90H	P1								
98H	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99H	SBUF								
A0H	P2								
A8H	IE	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
B0H	P3								
B8H	IP	–	PADC	PT2	PS	PT1	PX1	PT0	PX0
C0H	WDCON	–	–	–	–	OWDS	WDTS	WDT	SWDT
C8H	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
C9H	T2MOD	–	–	–	–	–	–	–	DCEN

SFR bit and byte addressable

SFR not bit addressable

must not be used

– : = bit location is reserved

Table 28
Contents of SFRs, SFRs in numeric order (cont'd)

Address	Register	Bit 7	6	5	4	3	2	1	0
CAH	RC2L								
CBH	RC2H								
CCH	TL2								
CDH	TH2								
D0H	PSW	CY	AC	F0	RS1	RS0	OV	F1	P
D8H	ADCON0	–	–	IADC	BSY	ADM	MX2	MX1	MX0
D9H	ADDATH	MSB							
DAH	ADDATL		LSB	–	–	–	–	–	–
DCH	ADCON1	ADCL	–	–	–	MX3	MX2	MX1	MX0
E0H	ACC								
F0H	B								

SFR bit and byte addressable

SFR not bit addressable

must not be used

– : = bit location is reserved

Timer / Counter 0 and 1

Timer/Counter 0 and 1 can be used in four operating modes as listed in Table 29:

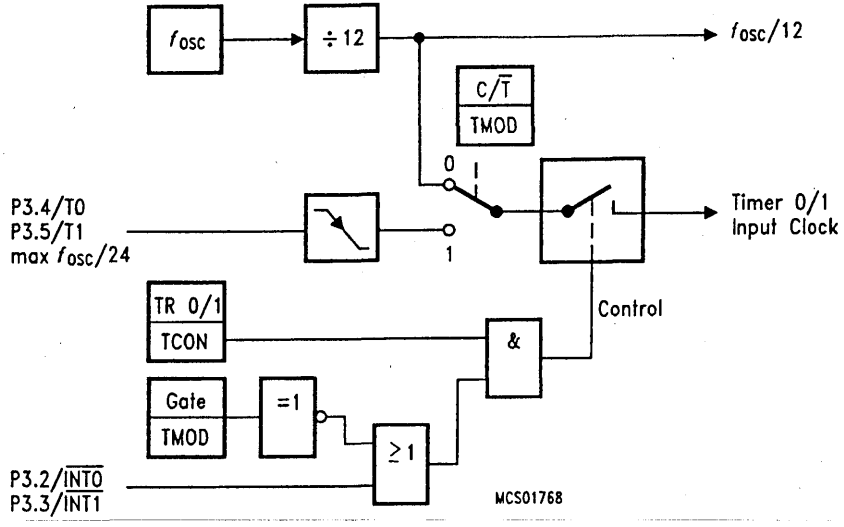
Table 29
Timer/Counter 0 and 1 operating modes

Mode	Description	TMOD				Input clock	
		Gate	C/$\overline{T}$	M1	M0	internal	external (max)
0	8-bit timer/counter with a divide-by-32 prescaler	X	X	0	0	*f*osc/12 x 32	*f*osc/24 x 32
1	16-bit timer/counter	X	X	0	1	*f*osc/12	*f*osc/24
2	8-bit timer/counter with 8-bit auto-reload	X	X	1	0	*f*osc/12	*f*osc/24
3	Timer/counter 0 used as one 8-bit timer/counter and one 8-bit timer Timer 1 stops	X	X	1	1	*f*osc/12	*f*osc/24

In the "timer" function (C/$\overline{T}$ = '0') the register is incremented every machine cycle. Therefore the count rate is *f*osc/12.

In the "counter" function the register is incremented in response to a 1-to-0 transition at its corresponding external input pin (P3.4/T0, P3.5/T1). Since it takes two machine cycles to detect a falling edge the max. count rate is *f*osc/24. External inputs $\overline{INT0}$ and $\overline{INT1}$ (P3.2, P3.3) can be programmed to function as a gate to facilitate pulse width measurements. Figure 61 Timer/Counter 0 and 1 input clock logic illustrates the input clock logic.

Figure 61
Timer/Counter 0 and 1 input clock logic



Timer 2

Timer 2 is a 16-bit Timer/Counter with an up/down count feature. It can operate either as timer or as an event counter which is selected by bit C/$\overline{T2}$ (T2CON.1). It has three operating modes as shown in Table 30.

Table 30
Timer/Counter 2 operating modes

Mode	T2CON			T2MOD	T2CON		Remarks	Input Clock	
	R×CLK or T×CLK	CP/$\overline{RL2}$	TR2	DCEN	EXEN	P1.1/ T2EX		internal	external (P1.0/T2)
16-bit Auto-reload	0	0	1	0	0	X	reload upon overflow	f_{osc}/12	max f_{osc}/24
	0	0	1	0	1	↓	reload trigger (falling edge)		
	0	0	1	1	X	0	Down counting		
	0	0	1	1	X	1	Up counting		
16-bit Cap-ture	0	1	1	X	0	X	16 bit Timer/ Counter (only up-counting)	f_{osc}/12	max f_{osc}/24
	0	1	1	X	1	↓	capture TH2, TL2 → RC2H, RC2L		
Baud Rate Gene-rator	1	X	1	X	0	X	no overflow interrupt request (TF2)	f_{osc}/2	max f_{osc}/24
	1	X	1	X	1	↓	extra external interrupt ("Timer 2")		
off	X	X	0	X	X	X	Timer 2 stops	–	–

Note: ↓ = falling edge

Serial Interface (USART)

The serial port is full duplex and can operate in four modes (one synchronous mode, three asynchronous modes) as illustrated in Table 31. The possible baud rates can be calculated using the formulas given in Table 32.

Table 31
USART operating modes

Mode	T2CON		Baudrate	Description
	SM0	SM1		
0	0	0	$f_{OSC}/12$	Serial data enters and exits through RxD. TxD outputs the shift clock. 8-bits are transmitted/received (LSB first)
1	0	1	Timer 1/2 overflow rate	8-bit UART 10 bits are transmitted (through TxD) or received (RxD)
2	1	0	$f_{OSC}/32$ or $f_{OSC}/64$	9-bit UART 11 bits are transmitted (TxD) or received (RxD)
3	1	1	Timer 1/2 overflow rate	9-bit UART Like mode 2 except the variable baud rate

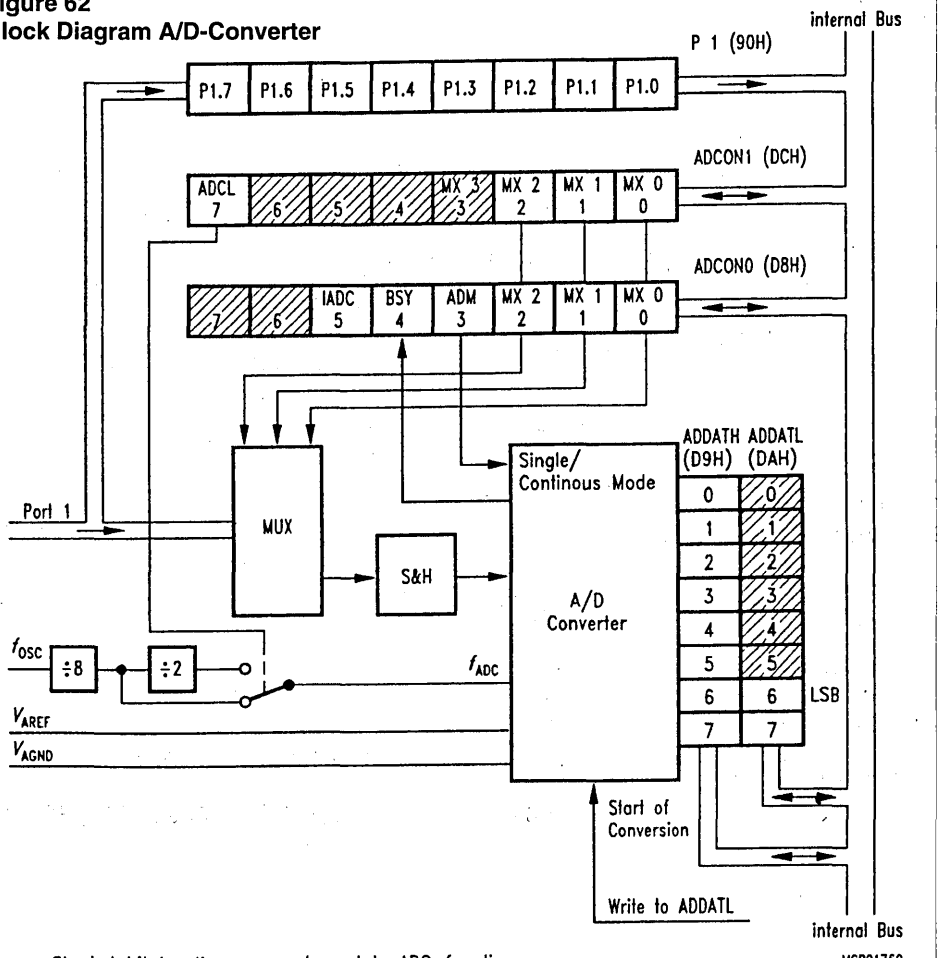
Table 32
Formulas for calculating Baud rates

Baud rate derived from	Interface Mode	Baud rate
Oscillator	0 2	$f_{OSC}/12$ $(2^{SMOD} \times f_{OSC}) / 64$
Timer 1 (16-bit timer) (8-bit timer with 8-bit autoreload)	1,3 1,3	$(2^{SMOD} \times$ timer 1 overflow rate$) /32$ $(2^{SMOD} \times f_{OSC}) / (32 \times 12 \times (256\text{-TH1}))$
Timer 2	1,3	$f_{OSC} / (32 \times (65536\text{-}(RC2H, RC2L))$

10-bit A/D Converter

In the SAB-C503 a high performance/high speed 8-channel 10-bit A/D-Converter (ADC) using the successive approximation technique is implemented.

Figure 62
Block Diagram A/D-Converter



Note that bit ADCL in SFR ADCON0 has to be set when f_{OSC} is higher than 16 MHz. Furthermore bit MX3 in SFR ADCON1 must not be set otherwise a not connected channel would be selected.

The formula for the conversion time is given by: $t_C = 14 \times (8 \times 2_{ADCL}) / f_{OSC}$

Interrupt System

The SAB-C503 provides 7 interrupt sources with two priority levels. Figure 63 Interrupt Request Sources gives a general overview of the interrupt sources and illustrates the request and control flags.

Figure 63
Interrupt Request Sources

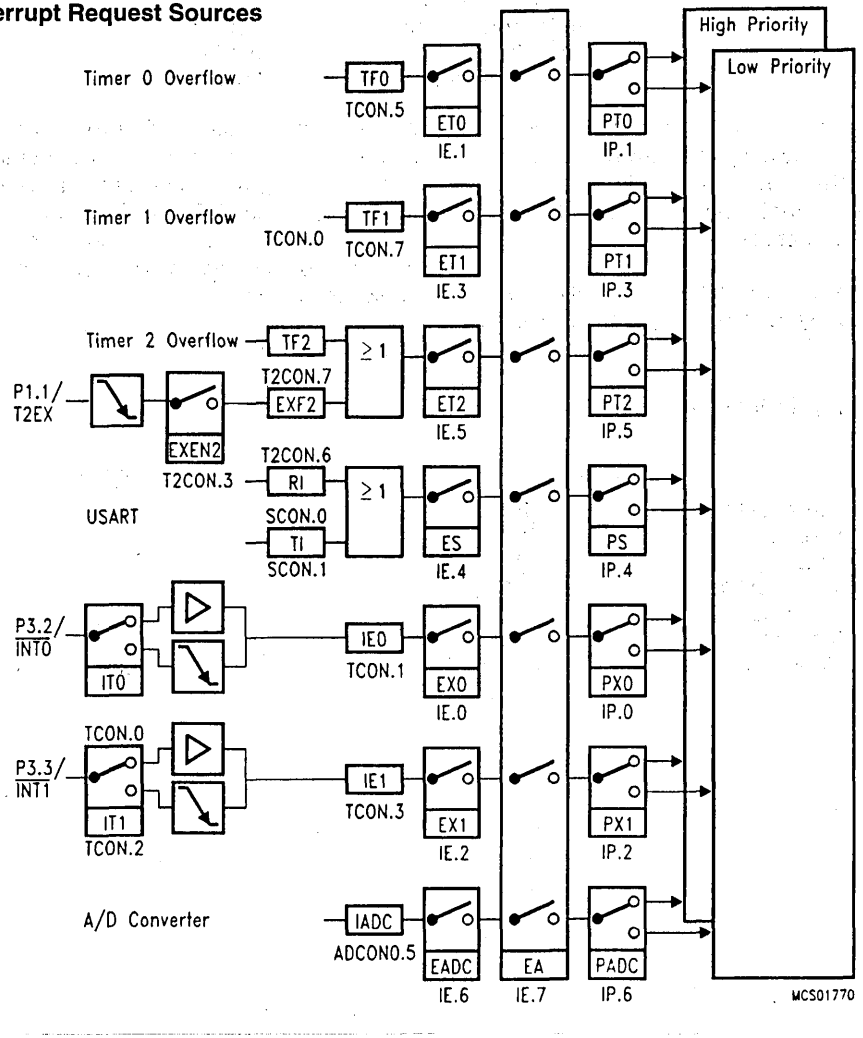


Table 33
Interrupt sources and their corresponding interrupt vectors

Source (Request Flags)	Vector	Vector Address
IE0	External interrupt 0	0003H
TF0	Timer 0 interrupt	000BH
IE1	External interrupt 1	0013H
TF1	Timer 1 interrupt	001BH
RI + TI	Serial port interrupt	0023H
TF2 + EXF2	Timer 2 interrupt	002BH
IADC	A/D converter interrupt	0043H

A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

If two requests of different priority level are received simultaneously, the request of higher priority is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as shown in Table 34.

Table 34
Interrupt priority-within-level

Interrupt Source		Priority
External Interrupt 0,	IE0	High
A/D Converter,	IADC	
Timer 0 Interrupt,	TF0	
External Interrupt 1,	IE1	↓
Timer 1 Interrupt,	TF1	
Serial Channel,	RI or TI	
Timer 2 Interrupt,	TF2 or EXF2	Low

Fail Safe Mechanisms

The SAB-C503 offers enhanced fail safe mechanisms, which allow an automatic recovery from software upset or hardware failure.

1)Watchdog Timer (15bit, WDT)

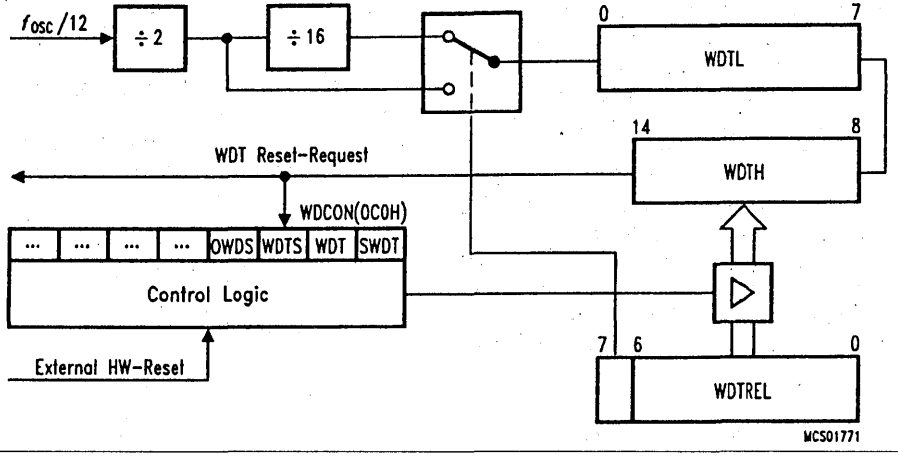
2)Oscillator Watchdog (OWD)

1) Watchdog Timer (WDT)

The Watchdog Timer in the SAB-C503 is a 15-bit timer, which is incremented by a count rate of either $f_{CYCLE}/2$ or $f_{CYCLE}/32$ ($f_{CYCLE} = f_{OSC}/12$). That is, the machine clock is divided by a series of arrangement of two prescalers, a divide-by-two and a divide-by-16 prescaler. The latter is enabled by setting bit WDTREL.7.

Figure 64 Block Diagram of the Programmable Watchdog Timer shows the block diagram of the programmable Watchdog Timer.

Figure 64
Block Diagram of the Programmable Watchdog Timer



– Starting and refreshing the WDT

Table 35 gives an overview how to start and refresh the WDT. The mentioned bits are located in SFR WDCON.

Table 35
Starting and refreshing the WDT

Function	Example	Remarks
Starting WD	SETB SWDT	Cannot be stopped during active mode of the device. WDT is halted during idle mode, power down mode or the oscillator watchdog reset is active.
Refreshing WD	SETB WDT SETB SWDT	Double instruction sequence (setting bit WDT and SWDT consecutively) to increase system security.

– Watchdog reset and watchdog status flag (WDTS)

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state 7FFCH. The duration of the reset signal then depends on the prescaler selection (either 8 or 128 cycles). This internal reset differs from an external one in so far as the Watchdog Timer is not disabled and bit WDTS (SFR WDCON) is set. The WDTS is a flip-flop, which is set by a Watchdog Timer reset and can be cleared by an external hardware reset. Bit WDTS allows the software to examine from which source the reset was activated. The bit WDTS can also be cleared by software.

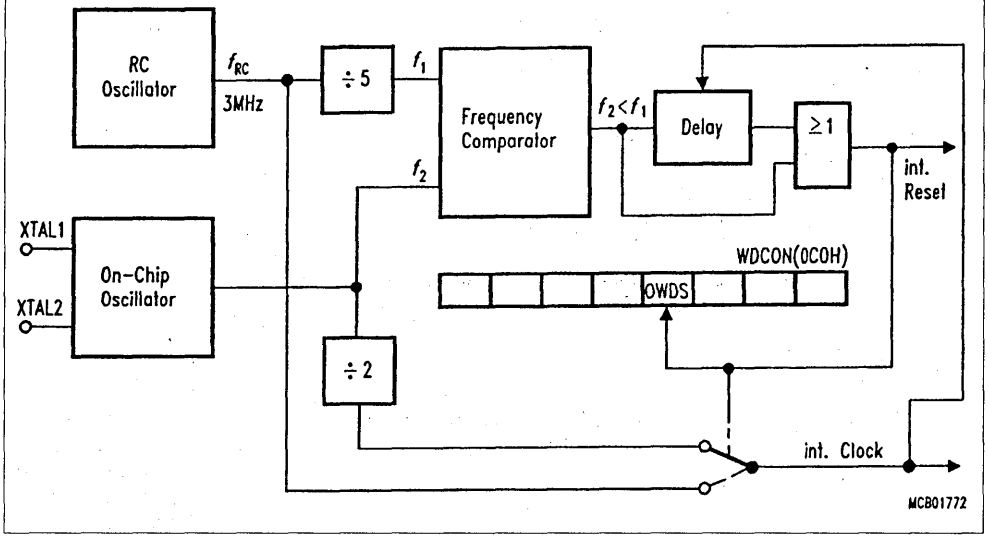
2) Oscillator Watchdog (OWD)

The OWD consists of an internal RC oscillator which provides the reference frequency for the comparison with the frequency of the on-chip oscillator.

Figure 65 Functional Block Diagram of the Oscillator Watchdog shows the block diagram of the oscillator watchdog unit while Table 36 shows the effect when the OWD becomes active/inactive.

Note: The OWD is always enabled!

Figure 65
Functional Block Diagram of the Oscillator Watchdog



4

Table 36
Effects of the OWD

Conditions	Effect
$f_{OSC} < f_{RC}/5$	Switch input of internal clock system to RC oscillator output Activating internal reset at the same time (reset sequence is clocked by RC-oscillator). Exception from effects of a Hardware Reset: Watchdog Timer Status Flag, WDTS is not reset Oscillator Watchdog Status Flag, OWDS is set
$f_{OSC} > f_{RC}/5$	Input of internal clock system is $f_{OSC}/2$. When failure condition ($f_{OSC} < f_{RC}/5$) disappears the part executes a final reset phase of typ. 1 ms in order to allow the external oscillator to stabilize.

Power Saving Modes

Two power down modes are available, the Idle Mode and Power Down Mode.

The bits PDE, PDS and IDLE, IDLS select the Power Down mode or the idle mode, respectively. If the Power Down mode and the idle mode are set at the same time, Power Down takes precedence. 37 gives a general overview of the power saving modes.

Table 37
Power Saving modes overview

Mode	Entering Example	Leaving by	Remarks
Idle mode	ORL PCON, #01H ORL PCON, #20H	– enabled interrupt – Hardware Reset	CPU is gated off CPU status registers maintain their data. Peripherals are active Double instruction sequence
Power-Down Mode	ORL PCON, #02H ORL PCON, #40H	Hardware Reset	Oscillator are stopped. Contents of on-chip RAM and SFR's are maintained (leaving Power Down Mode means redefinition of SFR's contents). Double instruction sequence

In the Power Down mode of operation, V_{CC} can be reduced to minimize power consumption. It must be ensured, however, that V_{CC} is not reduced before the Power Down mode is invoked, and that V_{CC} is restored to its normal operating level, before the Power Down mode is terminated. The reset signal that terminates the Power Down mode also restarts the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (simulator to power-on reset).

Absolute Maximum Ratings

Ambient temperature
under bias (T_A) ..... – 40 °C to + 85 °C
Storage temp. (T_{ST}) ..... – 65 °C to + 150 °C
Voltage on V_{CC} pins with
respect to ground (V_{SS})..... – 0.5 V to 6.5 V
Voltage on any pin with respect
to ground (V_{SS}) ..... – 0.5 V to V_{CC} + 0.5 V
Input current on any pin during
overload condition..... – 10 mA to + 10 mA
Absolute sum of all input currents
during overload condition..... | 100 mA |
Power dissipation..... TBD

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.*

4

DC Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V; T_A = 0 °C to 70 °C for the SAB-C503
T_A = – 40 °C to 85 °C for the SAF-C503

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, RESET)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage ($\overline{RESET}$)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except $\overline{RESET}$, XTAL1, $\overline{EA}$)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL1	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	
Input high voltage to $\overline{RESET}$, $\overline{EA}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (ports 2, 3)	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA 1)
Output low voltage (port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA 1)
Output high voltage (ports 2, 3)	V_{OH}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 80 µA, I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 800 µA 2), I_{OH} = – 80 µA 2)
Logic 0 input current (ports 1, 2, 3)	I_{IL}	– 10	– 50	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (ports 2, 3)	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{EA}$, P1)	I_{LI}	–	1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 °C
Power supply current: Active mode, 12 MHz 7) Idle mode, 12 MHz 7) Active mode, 20 MHz 7) Idle mode, 20 MHz 7) Power Down Mode	 I_{CC} I_{CC} I_{CC} I_{CC} I_{PD}	 – – – – –	 TBD TBD TBD TBD 50	 mA mA mA mA µA	 V_{CC} = 5 V, 4) V_{CC} = 5 V, 5) V_{CC} = 5 V, 4) V_{CC} = 5 V, 5) V_{CC} = 2 ... 5.5 V, 3)

1) Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and port 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

2) Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address lines are stabilizing.

3) I_{PD} (Power Down Mode) is measured under following conditions:
$\overline{EA}$ = RESET = Port 0 = Port 1 = V_{CC}; XTAL2 = N.C.; XTAL1 = V_{SS}; V_{AGND} = V_{SS}; all other pins are disconnected.

4) I_{CC} (active mode) is measured with:
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V, XTAL2 = N.C.; $\overline{EA}$ = Port 0 = Port 1 = RESET = V_{CC}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

5) I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.; RESET = $\overline{EA}$ = V_{SS}; Port 0 = Port 1 = V_{CC}; all other pins are disconnected;

7) $I_{CC\ Max}$ other frequencies is given by:
active mode:TBD
idle mode:TBD
where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V.

4

A/D Converter Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V
V_{AREF} = V_{CC} ± 5 %;

V_{AGND} = V_{SS} ± 0.2 V;
T_A = 0 °C to 70 °C for the SAB-C503
T_A = – 40 °C to 85 °C for the SAF-C503

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Analog input capacitance	C_I		25	70	pF	
Sample time (Inc. load time)	T_S			4 t_{CY} 1)	µs	2)
Conversion time (inc. sample time)	T_C			14 t_{CY} 1)	µs	3)
Total unadjusted error 4)	TUE			± 2	LSB	VAREF = V_{CC} VAGND = V_{SS}

1) t_{CY} = (8 × 2ADCL) / f_{OSC} (t_{CY} = 1 / f_{ADC}; f_{ADC} = f_{OSC} / (8 × 2ADCL)

2) This parameter specifies the time during the input capacitance C_I, can be charged/discharged by the external source. It must be guaranteed, that the input capacitance C_I, is fully loaded within this time. 4 TCY is 2 µs at the f_{OSC} = 16 MHz. After the end of the sample time T_S, changes of the analog input voltage have no effect on the conversion result.

3) This parameter includes the sample time T_S. 14 TCY is 7 µs at f_{OSC} = 16 MHz.

4) This parameter includes also the DNLE.

AC Characteristics for SAB-C503-LN / C503-1RN

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to 70 °C for the SAB-C503
T_A = – 40 °C to 85 °C for the SAF-C503

Please refer to the SAB-C501 AC Characteristics and Waveforms for the 12 and 20MHz AC Characteristics and Waveforms for the SAB-C503.

High-Performance 8-Bit CMOS Single-Chip Microcontroller

SAB 80C515/80C515-16 **CMOS microcontroller with factory mask-programmable ROM**
SAB 80C535/80C535-16 **CMOS microcontroller for external ROM**
SAB 80C515-T40/110,
SAB 80C535-T40/110 **Extended temperature range: – 40 to 110°C (for 12 MHz)**
SAB 80C515-T40/85
SAB 80C535-T40/85 **Extended temperature range: – 40 to 85°C (for 12 MHz)**
SAB 80C515-16-T40/85,
SAB 80C535-16-T40/85 **Extended temperature range: – 40 to 85°C (for 16 MHz)**

- 8 K × 8 ROM (SAB 80C515 only)
- 256 × 8 RAM
- Six 8-bit I/O ports, one input port for digital or analog input
- Three 16-bit timer/counters
- Highly flexible reload, capture, compare capabilities
- Full-duplex serial channel
- Twelve interrupt vectors, four priority levels
- 8-bit A/D converter with 8 multiplexed inputs and programmable internal reference voltages
- 16-bit watchdog timer
- Boolean processor
- 256-bit-addressable locations
- Most instructions execute in 1 µs (750 ns)
- 4 µs (3 µs) multiply and divide
- External memory expandable up to 128 Kbytes
- Backwardly compatible with SAB 8051
- Functionally compatible with SAB 80515
- Idle and power-down mode
- 68-pin plastic leaded chip carrier package (PL-CC-68)

The SAB 80C515/80C535 is a powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80515/80535 devices designed in MYMOS technology.

The SAB 80C515/80C535 is a stand-alone, high-performance single-chip microcontroller based on the SAB 8051/80C51 architecture. While maintaining all the SAB 80C51 operating characteristics, the SAB 80C515/80C535 incorporates several enhancements which significantly increase design flexibility and overall system performance.

In addition, the low-power properties of Siemens ACMOS technology allow applications where power consumption and dissipation are critical. Furthermore, the SAB 80C515/80C535 has two software-selectable modes of reduced activity for further power reduction: idle and power-down mode.

The SAB 80C535 is identical to the SAB 80C515 except that it lacks the on-chip program memory. The SAB 80C515/80C535 is supplied in a 68-pin plastic leaded chip carrier package (PL-CC-68). For the industrial temperature range – 40 to + 85°C, the SAB 80C515/80C535-T40/85 is available.

There are versions for 12 MHz operation and for 16 MHz operation available.

Ordering Information

Type	Package	Description)
SAB 80C515-N	PL-CC-68	with mask-programmable ROM, 12MHz
SAB 80C535-N	PL-CC-68	for external memory, 12MHz
SAB 80C515-N-T40/85	PL-CC-68	with mask-programmable ROM, 12MHz ext. temperature -40 to +85 °C
SAB 80C535-N-T40/85	PL-CC-68	for external memory, 12MHz ext. temperature -40 to +85 °C
SAB 80C515-N-T40/110	PL-CC-68	with mask-programmable ROM, 12MHz ext. temperature -40 to +110 °C
SAB 80C535-N-T40/110	PL-CC-68	for external memory, 12MHz ext. temperature -40 to +110 °C
SAB 80C515-16-N	PL-CC-68	with mask-programmable ROM, 16MHz
SAB 80C535-16-N	PL-CC-68	for external memory, 16MHz
SAB 80C535-16-N-T40/85	PL-CC-68	for external memory, 16MHz ext. temperature -40 to +85 °C

Figure 66
Pin Configuration (PL-CC-68)

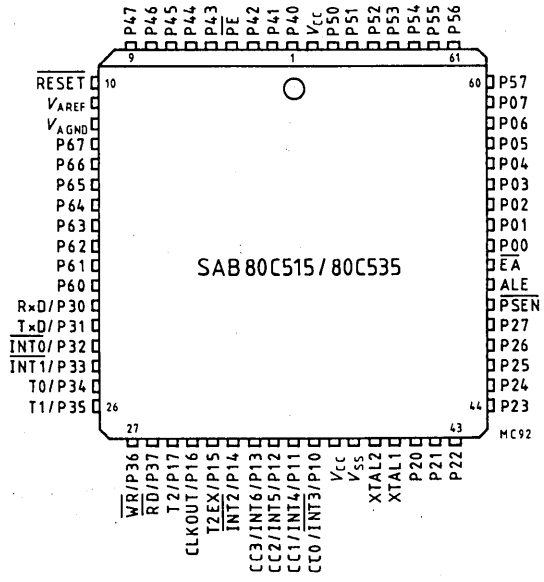
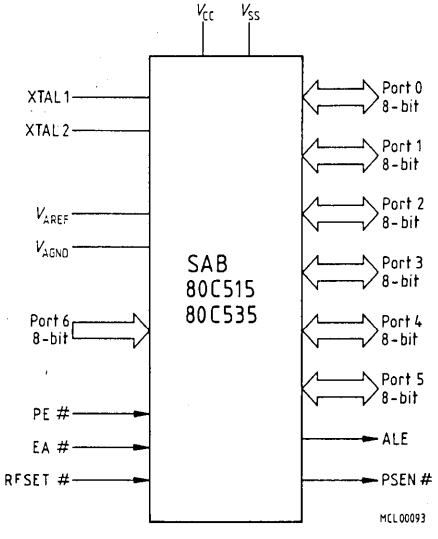


Figure 67
Logic Symbol



4

Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P4.0-P4.7	1-3, 5-9	I/O	**Port 4** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 4 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors.
PE#	4	I	**Power saving mode enable#** A low level on this pin enables the use of the power saving modes (idle mode and power-down mode). When PE# is held on high level it is impossible to enter the power saving modes.
RESET#	10	I	**Reset pin** A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80C515. A small internal pull-up resistor permits power-on reset using only a capacitor connected to V_{SS}.
V_{AREF}	11		**Reference voltage** for the A/D converter
V_{AGND}	12		**Reference ground** for the A/D converter
P6.7-P6.0	13-20	I	**Port 6** is an 8-bit unidirectional input port. Port pins can be used for digital input if voltage levels simultaneously meet the specifications for high/low input voltages and for the eight multiplexed analog inputs of the A/D converter.

Pin Definitions and Functions (cont'd)

Symbol	Pin	Input (I) Output (O)	Function
P3.0-P3.7	21-28	I/O	**Port 3** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 3 pins that have1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – RxD (P3.0): serial port's receiver data input (asynchronous) or data input/output (synchronous) – TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) – INT0# (P3.2): interrupt 0 input/timer 0 gate control input – INT1# (P3.3): interrupt 1 input/timer 1 gate control input – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – WR# (P3.6): the write control signal latches the data byte from port 0 into the external data memory – RD# (P3.7): the read control signal enables the external data memory to port 0

4

Pin Definitions and Functions (cont'd)

Symbol	Pin	Input (I) Output (O)	Function
P1.7 - P1.0	29 - 36	I/O	**Port 1** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pull-up resistors. The port is used for the low-order address byte during program verification. Port 1 also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: – INT3#/CC0 (P1.0): interrupt 3 input/ compare 0 output/capture 0 input – INT4/CC1 (P1.1): interrupt 4 input/ compare 1 output/capture 1 input – INT5/CC2 (P1.2): interrupt 5 input/ compare 2 output/capture 2 input – INT6/CC3 (P1.3): interrupt 6 input/ compare 3 output/capture 3 input – INT2# (P1.4): interrupt 2 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input

Pin Definitions and Functions (cont'd)

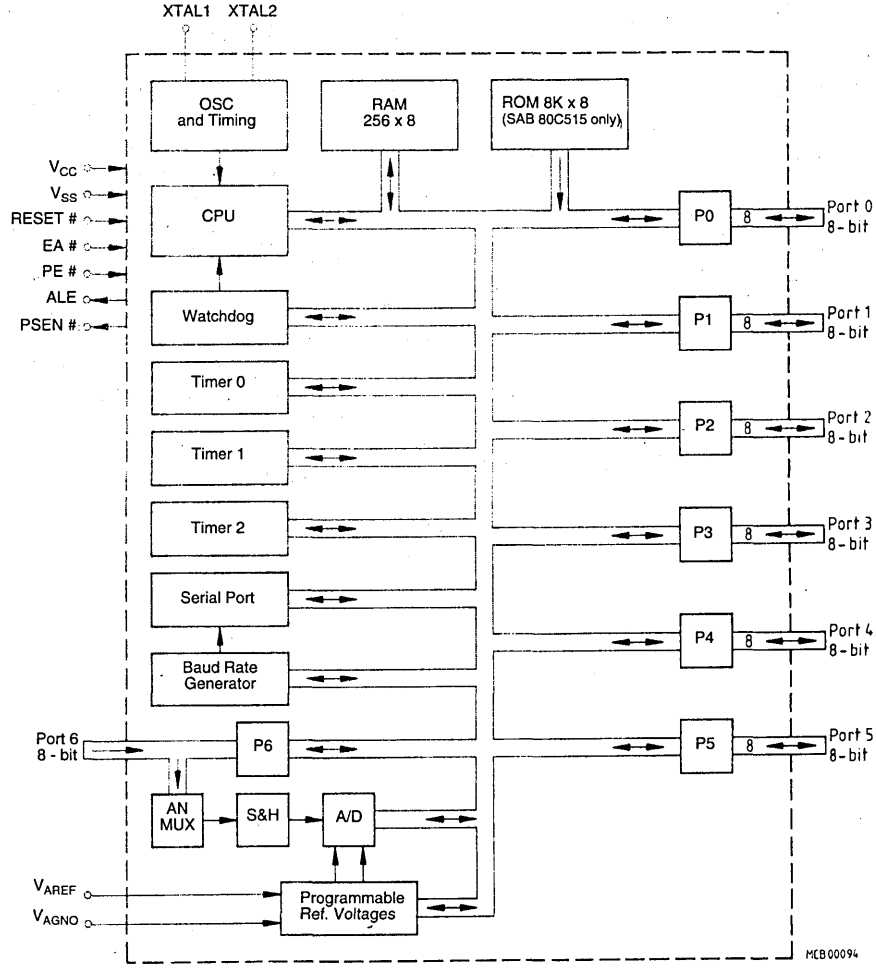
Symbol	Pin	Input (I) Output (O)	Function
XTAL2 XTAL1	39 49		**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. **XTAL1** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times and rise/fall times specified in the AC characteristics must be observed.
P2.0-P2.7	41- 48	I/O	**Port 2** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX@DPTR). In this application it uses strong internal pull-up resistors when issuing 1's. During accesses to external data memory that use 8-bit addresses (MOVX@Ri), port 2 issues the contents of the P2 special function register.
PSEN#	49	O	The **Program store enable#** Output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. The signal remains high during internal program execution.

4

Pin Definitions and Functions (cont'd)

Symbol	Pin	Input (I) Output (O)	Function
ALE	50	O	The **Address latch enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods, except during an external data memory access.
EA#	51	I	**External access enable#** When held high, the SAB 80C515 executes instructions from the internal ROM as long as the PC is less than 8192. When held low, the SAB 80C515 fetches all instructions from external program memory. For the SAB 80C535 this pin must be tied low.
P0.0-P0.7	52-59	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pull-up resistors when issuing 1's. Port 0 also outputs the code bytes during program verification in the SAB 80C515. External pull-up resistors are required during program verification.
P5.7-P5.0	60-67	I/O	**Port 5** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 5 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pull-up resistors.
V_{CC}	37		**Supply voltage** during normal, idle, and power-down operation. Internally connected to pin 68.
V_{SS}	38		**Ground** (0 V)
V_{CC}	68		**Supply voltage** during normal, idle, and power-down operation. Internally connected to pin 37.

Figure 68
Block Diagram



Functional Description

The members of the SAB 80515 family of microcontrollers are:

- SAB 80C515:
  Microcontroller, designed in Siemens ACMOS technology, with
  8 Kbyte factory mask-programmable ROM
- SAB 80C535:
  ROM-less version of the SAB 80C515
- SAB 80515:
  Microcontroller, designed in Siemens MYMOS technology, with
  8 Kbyte factory mask-programmable ROM
- SAB 80535:
  ROM-less version of the SAB 80515

Architecture

The architecture of the SAB 80C515 is based on the SAB 8051/SAB 80C51 microcontroller family. The following features of the SAB 80C515 are fully compatible with the SAB 80C51 features:

- Instruction set
- External memory expansion interface (port 0 and port 2)
- Full-duplex serial port
- Timer/counter 0 and 1
- Alternate functions on port 3
- The lower 128 bytes of internal RAM and the lower 4 Kbytes of internal ROM

The SAB 80C515 additionally contains 128 bytes of internal RAM and 4 Kbytes of internal ROM, which results in a total of 256 bytes of RAM and 8 Kbytes of ROM on-chip. The SAB 80C515 has a new 16-bit timer/counter with a 2:1 prescaler, reload mode, compare and capture capability. It also contains a 16-bit watchdog timer, an 8-bit A/D converter with programmable reference voltages, two additional quasi-bidirectional 8-bit ports, one 8-bit input port for analog or digital signals, and a programmable clock output ($f_{OSC}/12$). Furthermore, the SAB 80C515 has a powerful interrupt structure with 12 vectors and 4 programmable priority levels.

Figure 68 Block Diagram shows a block diagram of the SAB 80C515.

Differences in Pin Assignments of the SAB 80C515 and SAB 80515

Since the SAB 80C515 is designed in CMOS technology, this device requires no V_{BB} pin, because the die's substrate is internally connected to V_{CC}.

Furthermore, the RAM backup power supply via pin V_{PD} is replaced by the software controlled power-down mode and power supply via V_{CC}.

Therefore, pins V_{BB} and V_{PD} of the NMOS version SAB 80515 are used for other functions in the SAB 80C515.

Pin 4 (the former pin V_{PD}) is the new PE# pin which enables the use of the power saving modes.

Pin 37 (the former pin V_{BB}) becomes an additional V_{CC} pin. Thus, it is possible to insert a decoupling capacitor between pin 37 (V_{CC}) and pin 38 (V_{SS}) very close to the device, thereby avoiding long wiring and reducing the voltage distortion resulting from high dynamic current peaks.

There is a difference between the NMOS and CMOS version concerning the clock circuitry. When the device is driven from an external source, pin XTAL2 must be driven by the clock signal; pin XTAL1, however, must be left open in the SAB 80C515 (must be tied low in the NMOS version). When using the oscillator with a crystal there is no difference in the circuitry.

Thus, due to its pin compatibility the SAB 80C515 normally substitutes any SAB 80515 without redesign of the user's printed circuit board, but the user has to take care that the two V_{CC} pins are hardwired on-chip. In any case, it is recommended that power is supplied on both V_{CC} pins of the SAB 80C515 to improve the power supply to the chip. If the power saving modes are to be used, pin PE# must be tied low, otherwise these modes are disabled.

4

Absolute Maximum Ratings

Ambient temperature under bias
SAB 80C515 ..... 0 to 70 °C
SAB 80C515-T40/85 ..... – 40 to 85 °C
SAB 80C515-T40/110 ..... – 40 to 110 °C
Storage temperature ..... – 65 to 150 °C
Voltage on any pins with respect to ground (V_{SS}) ..... – 0.5 to V_{CC} 0.5 V
Voltage on V_{CC} to V_{SS} ..... – 0.5 to + 6.5
Power dissipation 2W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5 V + 10%, – 15%; V_{SS} = 0 V T_A = to 70 °C for the SAB 80C515/80C535
T_A = – 40 to 85 °C for the SAB 80C515/80C535-T40/85
T_A = – 40 to 110 °C for the SAB 80C515/80535-T40/110

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Input low voltage (except EA#)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage (EA#)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input high voltage (except RESET# and XTAL2)	V_{IH}	2.0 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL2	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	–
Input high voltage to RESET#	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage, ports 1, 2, 3, 4, 5	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA 1)
Output low voltage, port 0, ALE, PSEN#	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA 1)
Output high voltage, ports 1, 2, 3, 4, 5	V_{OH}	2.4 0.9 V_{CC}	– –	V V	I_{OH} = – 80 µA I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, PSEN)	V_{OH1}	2.4 0.9 V_{CC}	– –	V V	I_{OH} = – 400 µA I_{OH} = – 40 µA 2)

DC Characteristics (cont'd)

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Logic 0 input current, ports 1, 2, 3, 4, 5,	I_{IL}	– 10	– 70	µA	V_{IN} = 0.45 V
Input low current to RESET# for reset	I_{L2}	– 10	– 100	µA	V_{IN} = 0.45 V
Input low current (XTAL2#)	I_{L3}	– 10	–15	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current, ports 1, 2, 3, 4, 5	T_L	-65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, EA)	I_{LI}	–	± 10	µA	0.45 < V_{IN} < V_{CC} [1]
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25°C
Power-supply current:					
Active mode, 12 MHz [6]	I_{CC}	–	35	mA	V_{CC} = 5 V [4]
Idle mode, 12 MHz [6]	I_{CC}	–	13	mA	V_{CC} = 5 V [5]
Active mode, 16 MHz [6]	I_{CC}	–	46	mA	V_{CC} = 5 V [4]
Idle mode, 16 MHz [6]	I_{CC}	–	17	mA	V_{CC} = 5 V [5]
Power-down mode	I_{CC}	–	50	µA	V_{CC} = 2 V to 5.5 V [3]

[1] Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4 and 5. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation.
In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V.Then, it may be desirable to qualify ALE with a Schmitttrigger, or use an address latch with a Schmit trigger strobe input.

[2] Capacitive loading on ports 0 and 2 may cause the VOH on ALE and PSEN to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.

[3] Power-down I_{CC} is measured with: EA# = Port 0 = Port 6 = V_{CC};
XTAL1 = N.C.; XTAL2 = V_{SS}; RESET# = V_{CC}; V_{AGND} = V_{SS}; all other pins are disconnected.

[4] I_{CC} (active mode) is measured with: XTAL2 driven with the clock signal according to the figure below; XTAL1 = N.C.; EA# = Port 0 = Port 6 = V_{CC}; RESET# = V_{SS}; all other pins are disconnected. I_{CC} might be slightly higher if a crystal oscillator is used.

[5] I_{CC} (idle mode) is measured with: XTAL2 driven with the clock signal according to the figure below; XTAL1 = N.C.; EA# = V_{SS}; Port 0 = Port 6 V_{CC}; RESET# = V_{CC}; all other pins are disconnected; all on-chip peripherals are disabled.

[6] I_{CC} at other frequencies is given by:
Active mode: $I_{CC\,max}$ (mA) = 2.67 × f_{OSC} (MHz) + 3.00
Idle mode: $I_{CC\,max}$ (mA) = 0.88 × f_{OSC} (MHz) + 2.50
where f_{OSC} is the oscillator frequency in MHz.
$I_{CC\,max}$ is given in mA and measured at V_{CC} = 5 V (see also notes 4 and 5).

4

A/D Converter Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; V_{AREF} = V_{CC} = 5.0 V; V_{AGND} = V_{SS} = 0.0 V;
$V_{htAREF} - V_{htAGND} \geq 1$ V; T_A = 0 to 70°C for SAB 80C515/80C535
T_A = – 40 to 85°C for SAB 80C515/80C535-T40/85
T_A = – 40 to 110°C for SAB 80C515/80C535-T40/110

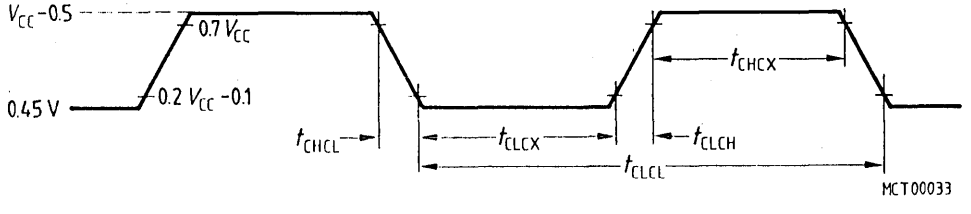
Parameter	Symbol	Limit Values			Unit	Test Conditions
		min.	typ.	max.		
Analog input voltage	V_{AINPUT}	V_{AGND} – 0.2	–	V_{AREF} + 0.2	V	9)
Analog input capacitance	C_I	–	25	45	pF	7)
Load time	t_L	–	–	2 t_{CY}	µs	–
Sample time (incl. load time)	t_S	–	–	7 t_{CY}	µs	–
Conversion time (incl. sample time)	t_C	–	–	13 t_{CY}	µs	–
Differential non-linearity	DNLE	–	± 1/2	± 1	LSB	V_{htAREF} =
Integral non-linearity	INLE	–	± 1/2	± 1	LSB	V_{AREF} = V_{CC}
Offset error			± 1/2	± 1	LSB	V_{htAGND} =
Gain error			± 1/2	± 1	LSB	V_{AGND} = V_{SS}
Total unadjusted error	TUE		± 1	± 2	LSB	7)
V_{AREF} supply current	I_{REF}	–	–	5	mA	8)
Internal reference error	$V_{htREFERR}$	–	± 5	± 30	mV	8)

7) The output impedance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (t_L). After charging of the internal capacitance (C_I) in the load time (t_L) the analog input must be held constant for the rest of the sample time (t_S).

8) The differential impedance r_D of the analog reference voltage source must be less than 1 kΩ at reference supply voltage.

9) Exceeding these limit values at one or more input channels will cause additional current which is sinked / sourced at these channels. This may also affect the accuracy of other channels which are operated within these specifications.

Figure 69
Clock Signal Waveform for I_{CC} Tests in Active and Idle Mode (t_{CLCH} = t_{CHCL} = 5 ns)



AC Characteristics for SAB 80C515/80C535

V_{CC} = 5 V ± 10%; V_{SS} = 0 V (C_L for Port 0, ALE and PSEN# outputs = 100 pF; C_L for all outputs = 80 pF)
T_A = 0 to 70°C for SAB 80C515/80C535
T_A = – 40 to 85°C for SAB 80C515/80C535-T40/85
T_A = – 40 to 110°C for SAB 80C515/80C535-T40/110

Parameter	Symbol	Limit Values				Unit
		12 MHz-clock		Variable clock $1/t_{CLCL}$ = 0.5 to 12 MHz		
		min.	max	min.	max.	

Program Memory Characteristics

ALE pulse width	t_{LHLL}	127	–	$2t_{CLCL} - 40$	–	ns
Address setup to ALE	t_{AVLL}	53	–	$t_{CLCL} - 30$	–	ns
Address hold after ALE	t_{LLAX}	48	–	$t_{CLCL} - 35$	–	ns
ALE to valid instruction in	t_{LLIV}	–	233	–	$4t_{CLCL} - 10$	ns
ALE to PSEN#	t_{LLPL}	58	–	$t_{CLCL} - 25$	–	ns
PSEN# pulse width	t_{PLPH}	215		$3t_{CLCL} - 35$		ns
PSEN# to valid instruction in	t_{PLIV}	–	150	–	$3t_{CLCL} - 100$	ns
Input instruction hold after PSEN#	t_{PXIX}	0	–	0	–	ns
Input instruction float after PSEN#	t_{PXIZ} [1]	–	63	–	$t_{CLCL} - 20$	ns
Address valid after PSEN#	t_{PXAV} [1]	75		$t_{CLCL} - 8$		ns
Address to valid instruction in	t_{AVIV}	–	302	–	$5t_{CLCL} - 115$	ns
Address float to PSEN#	t_{AZPL}	0	–	0	–	ns

[1] Interfacing the SAB 80C515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics for SAB 80C515/80C535 (cont'd)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V (C_L for Port 0, ALE and PSEN# outputs = 100 pF; C_L for all outputs = 80 pF)
T_A = 0 to 70°C for SAB 80C515/80C535
T_A = – 40 to 85°C for SAB 80C515/80C535-T40/85
T_A = – 40 to 110°C for SAB 80C515/80C535-T40/110

Parameter	Symbol	Limit Values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 0.5 to 12 MHz		
		min	max	min.	max.	

External Data Memory Characteristics

RD# pulse width	t_{RLRH}	400	–	6 t_{CLCL} – 100	–	ns
WR# pulse width	t_{WLWH}	400	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	132	–	2 t_{CLCL} – 35	–	ns
RD# to valid data in	t_{RLDV}	–	252	–	5 t_{CLCL} – 165	ns
DATA hold after RD#	t_{RHDX}	0	–	0		ns
Data float after RD#	t_{RHDZ}	–	97	–	2 t_{CLCL} – 70	ns
ALE to valid data in	t_{LLDV}	–	517	–	8 t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	585	–	9 t_{CLCL} – 165	ns
ALE to WR# or RD#	t_{LLWL}	200	300	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
WR# or RD# high to ALE high	t_{WHLH}	43	123	t_{CLCL} – 40	t_{CLCL} + 40	ns
Address valid to WR#	t_{AVWL}	203	–	4 t_{CLCL} – 130	–	ns
Data valid to WR# transition	t_{QVWX}	33	–	t_{CLCL} – 50	–	ns
Data setup before WR#	t_{QVWH}	433	–	7 t_{CLCL} – 150	–	ns
Data hold after WR#	t_{WHQX}	33	–	t_{CLCL} – 50	–	ns
Address float after RD#	t_{RLAZ}	–	0	–	0	ns

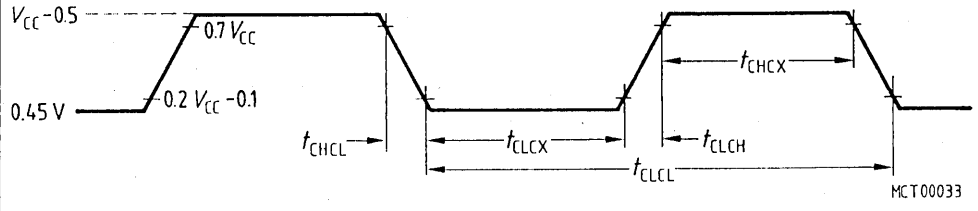
AC Characteristics for SAB 80C515/80C535 (cont'd)

Parameter	Symbol	Limit Values		Unit
		Variable clock Frequ. = 0.5 to 12 MHz		
		min.	max.	

External Clock Drive

Oscillator period	t_{CLCL}	83.3	2000	ns
Oscillator frequency	$1/t_{CLCL}$	0.5	12	MHz
High time	t_{CHCX}	20	–	ns
Low time	t_{CLCX}	20	–	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

Figure 70
External Clock Cycle

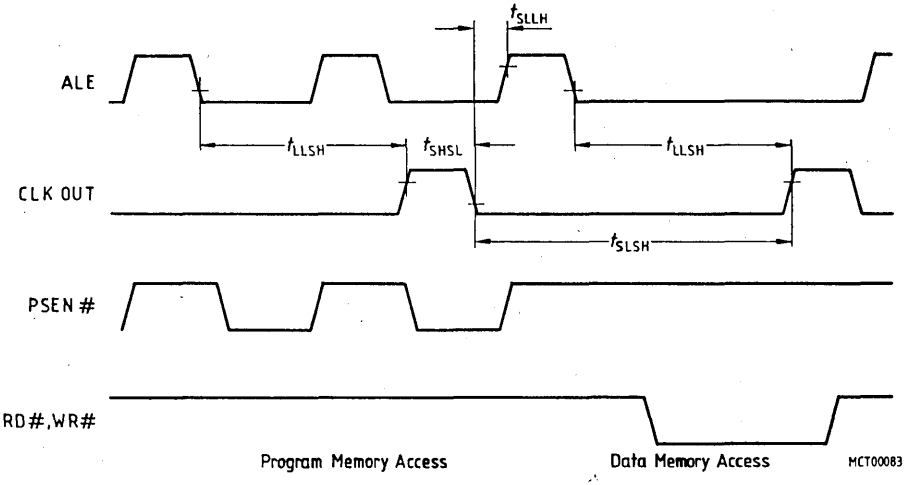


4

System Clock Timing for SAB 80C515/80C535

Parameter	Symbol	Limit Values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 0.5 to 12 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	543	–	$7\,t_{CLCL} - 40$	–	ns
CLKOUT high time	t_{SHSL}	127	–	$2\,t_{CLCL} - 40$	–	ns
CLKOUT low time	t_{SLSH}	793	–	$10\,t_{CLCL} - 40$	–	ns
CLKOUT low to ALE high	t_{SLLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns

Figure 71
System Clock Timing



AC Characteristics for SAB 80C515-16/80C535-16

V_{CC} = 5 V ± 10%; V_{SS} = 0 V (C_L for Port 0, ALE and PSEN# outputs = 100 pF; C_L for all outputs = 80 pF)
T_A = 0 to 70°C for SAB 80C515-16/80C535-16
T_A = – 40 to 85°C for SAB 80C515-16/80C535-16-T40/85

Parameter	Symbol	Limit Values				Unit
		16 MHz clock		Variable clock 1/t_{CLCL} = 0.5 to 16 MHz		
		min.	max.	min.	max.	

Program Memory Characteristics

ALE pulse width	t_{LHLL}	85	–	2 t_{CLCL} – 40	–	ns
Address setup to ALE	t_{AVLL}	33	–	t_{CLCL} – 30	–	ns
Address hold after ALE	t_{LLAX}	28	–	t_{CLCL} – 35	–	ns
ALE to valid instruction in	t_{LLIV}	–	150	–	4 t_{CLCL} – 100	ns
ALE to PSEN#	t_{LLPL}	38	–	t_{CLCL} – 25	–	ns
PSEN# pulse width	t_{PLPH}	153		3 t_{CLCL} – 35		ns
PSEN# to valid instruction in	t_{PLIV}	–	88	–	3 t_{CLCL} – 100	ns
Input instruction hold after PSEN#	t_{PXIX}	0	–	0	–	ns
Input instruction float after PSEN#	t_{PXIZ} [1]	–	43	–	t_{CLCL} – 20	ns
Address valid after PSEN#	t_{PXAV} [1]	55		t_{CLCL} – 8		ns
Address to valid instruction in	t_{AVIV}	–	198	–	5 t_{CLCL} – 115	ns
Address float to PSEN#	t_{AZPL}	0	–	0	–	ns

[1] Interfacing the SAB 80C515-16 to devices with float times up to 55 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

4

AC Characteristics for SAB 80C515-16/80C535-16 (cont'd)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V (C_L for Port 0, ALE and PSEN# outputs = 100 pF; C_L for all outputs = 80 pF)
T_A = 0 to 70°C for SAB 80C515-16/80C535-16
T_A = – 40 to 85°C for SAB 80C515-16/80C535-16-T40/85

Parameter	Symbol	Limit Values				Unit
		16 MHz clock		Variable clock $1/t_{CLCL}$ = 0.5 to 16 MHz		
		min.	max.	min.	max.	

External Data Memory Characteristics

RD# pulse width	t_{RLRH}	275	–	6 t_{CLCL} – 100	–	ns
WR# pulse width	t_{WLWH}	275	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	90	–	2 t_{CLCL} – 35	–	ns
RD# to valid data in	t_{RLDV}	–	148	–	5 t_{CLCL} – 165	ns
DATA hold after RD#	t_{RHDX}	0	–	0		ns
Data float after RD#	t_{RHDZ}	–	55	–	2 t_{CLCL} – 70	ns
ALE to valid data in	t_{LLDV}	–	350	–	8 t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	398	–	9t_{CLCL} – 165	ns
ALE to WR# or RD#	t_{LLWL}	138	238	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
WR# or RD# high to ALE high	t_{WHLH}	23	103	t_{CLCL} – 40	t_{CLCL} + 40	ns
Address valid to WR#	t_{AVWL}	120	–	4 t_{CLCL} – 130	–	ns
Data valid to WR# transition	t_{QVWX}	13	–	t_{CLCL} – 50	–	ns
Data setup before WR#	t_{QVWH}	288	–	7 t_{CLCL} – 150	–	ns
Data hold after WR#	t_{WHQX}	13	–	t_{CLCL} – 50	–	ns
Address float after RD#	t_{RLAZ}	–	0	–	0	ns

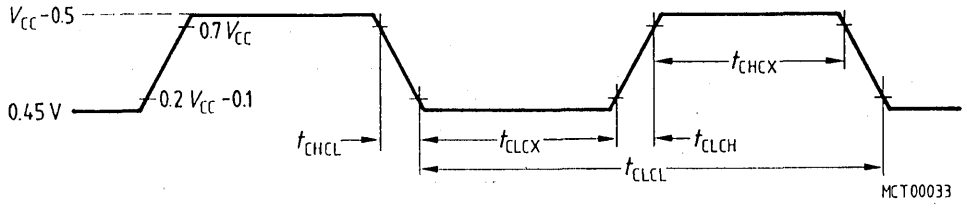
AC Characteristics for SAB 80C515-16/80C535-16 (cont'd)

Parameter	Symbol	Limit Values		Unit
		Variable clock Frequ. = 0.5 to 16 MHz		
		min.	max.	

External Clock Drive

Oscillator period	t_{CLCL}	62.5	2000	ns
Oscillator frequency	$1/t_{CLCL}$	0.5	16	MHz
High time	t_{CHCX}	15	–	ns
Low time	t_{CLCX}	15	–	ns
Rise time	t_{CLCH}	–	15	ns
Fall time	t_{CHCL}	–	15	ns

Figure 72
External Clock Cycle

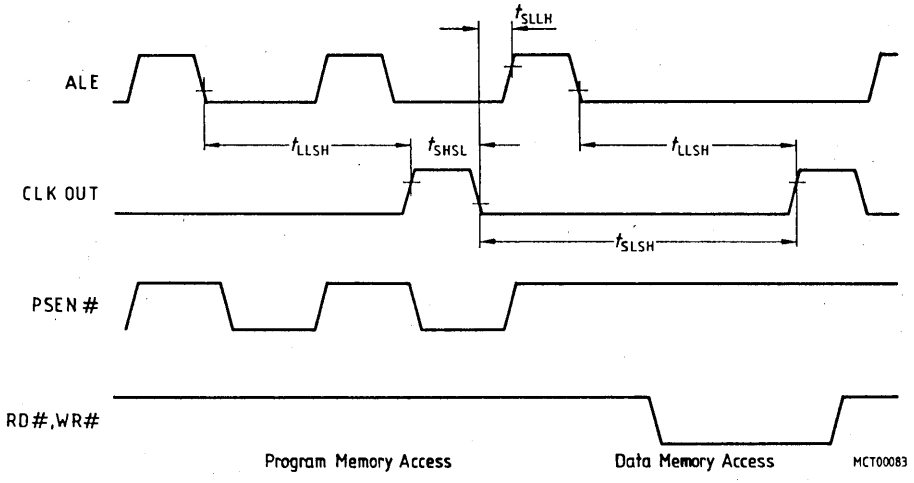


4

System Clock Timing for SAB 80C515-16/80C535-16

Parameter	Symbol	Limit Values				Unit
		16 MHz clock		Variable clock $1/t_{CLCL}$ = 0.5 to 16 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	398	–	$7\,t_{CLCL} - 40$	–	ns
CLKOUT high time	t_{SHSL}	85	–	$2\,t_{CLCL} - 40$	–	ns
CLKOUT low time	t_{SLSH}	585	–	$10\,t_{CLCL} - 40$	–	ns
CLKOUT low to ALE high	t_{SLLH}	23	103	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns

Figure 73
System Clock Timing

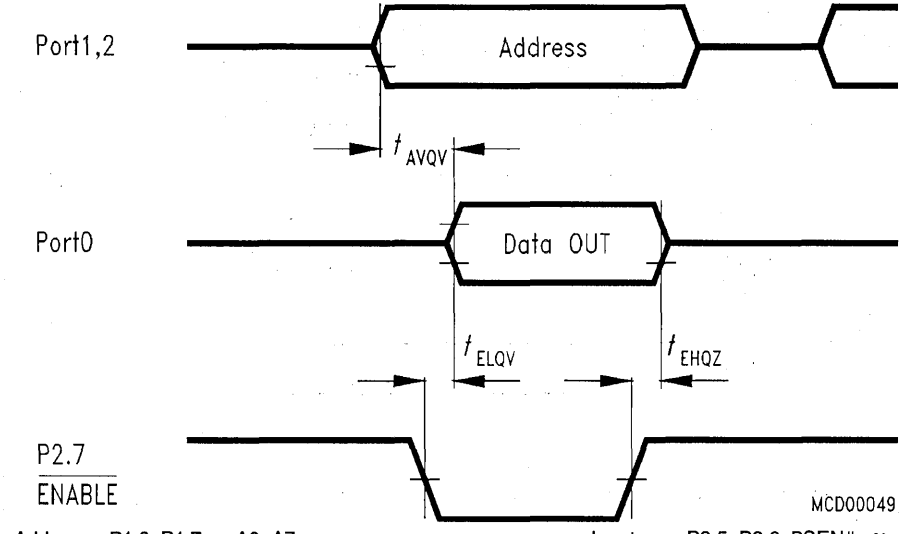


ROM Verification Characteristics

$T_A = 25°C \pm 5°C$; $V_{CC} = 5\ V \pm 10\%$; $V_{SS} = 0\ V$

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	–	48 t_{CLCL}	ns
ENABLE to valid data	t_{ELQV}	–	48 t_{CLCL}	ns
Data float after ENABLE	t_{EHQZ}	0	48 t_{CLCL}	ns
Oscillator frequency	$1/t_{CLCL}$	4	6	MHz

Figure 74
ROM Verification



Address: P1.0–P1.7 = A0–A7
P2.0–P2.4 = A8–A12
Data: Port 0 = D0–D7

Inputs: P2.5–P2.6, PSEN#= V_{SS}
ALE, EA# = V_{IH}
RESET# = V_{IH1}

4

Waveforms

Figure 75
Program Memory Read Cycle

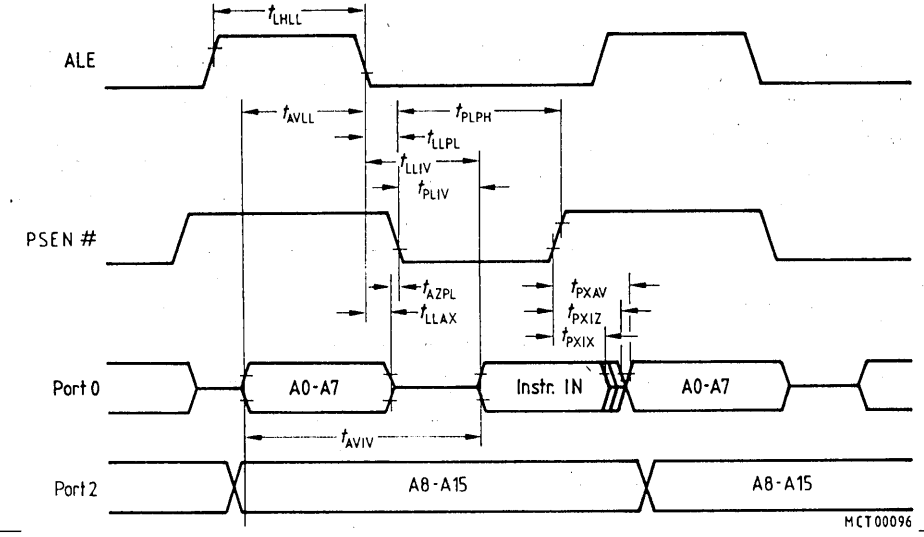


Figure 76
Data Memory Read Cycle

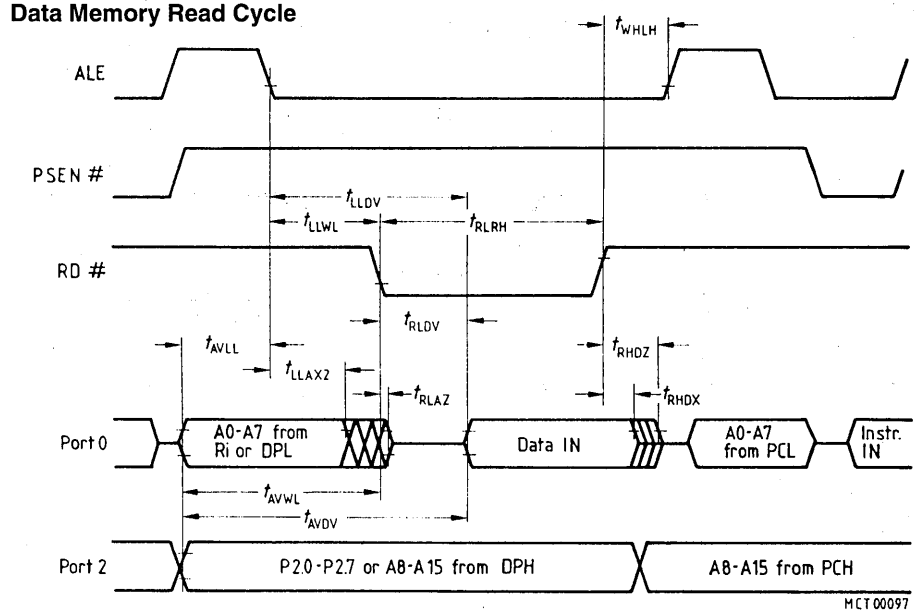


Figure 77
Data Memory Write Cycle

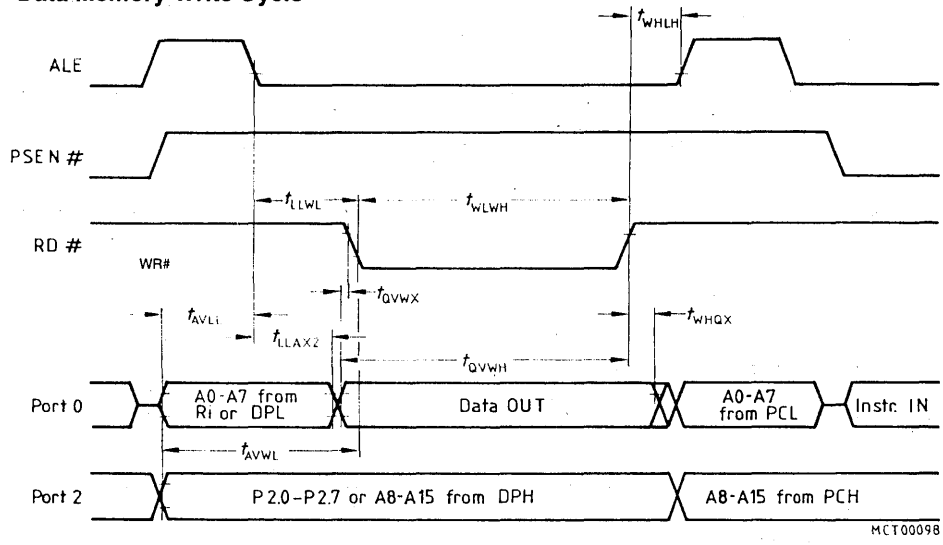
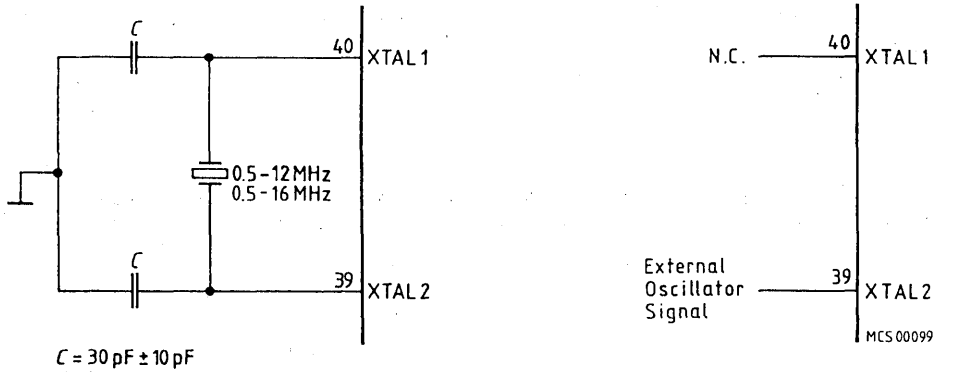


Figure 78
Recommended Oscillator Circuits

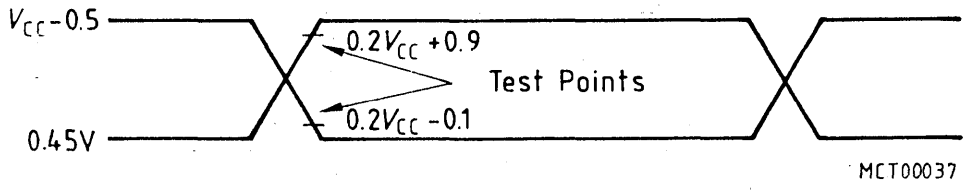


C = 30 pF ± 10 pF
(incl. stray capacitance)

Crystal Oscillator Mode

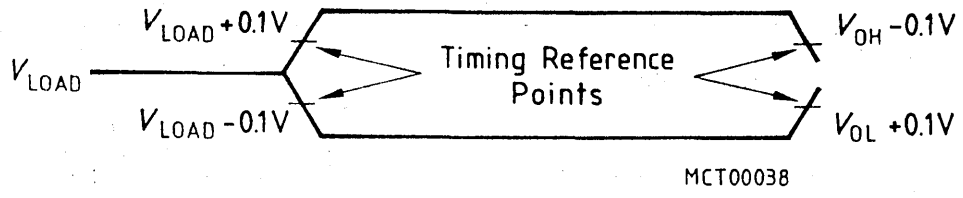
Driving from External Source

Figure 79
AC Testing: Input, Output Waveforms



AC inputs during testing are driven at V_{CC} – 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at $V_{IH\ min}$ for a logic '1' and $V_{IL\ max}$ for a logic '0'.

Figure 80
AC Testing: Float Waveforms



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV deviation from the load voltage V_{OH}/V_{OL} occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

High-Performance
8-Bit CMOS Single-Chip Microcontroller

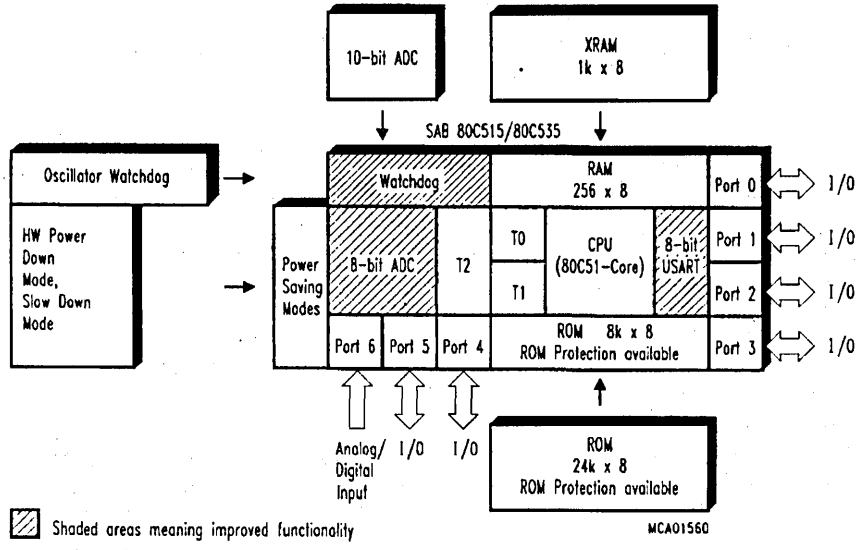
SAB 83C515A-5 Microcontroller with factory mask-programmable ROM
SAB 80C515A Microcontroller for external ROM

- SAB 80C515A/83C515A-5, up to 18 MHz operation frequency
- 32 K x 8 ROM (SAB 83C515A-5 only, ROM-Protection available)
- 256 x 8 on-chip RAM
- Additional 1 K x 8 on-chip RAM (XRAM)
- Superset of SAB 80C51 architecture:
  1 µs instruction cycle time at 12 MHz
  666ns instruction cycle time at 18 MHz
  256 directly addressable bits
  Boolean processor
  64 Kbyte external data and program memory addressing
- Three 16-bit timer/counters
- Versatile "fail-safe" provisions
- Twelve interrupt vectors, four priority levels selectable
- Genuine 10-bit A/D converter with 8 multiplexed inputs
- Full duplex serial interface with programmable Baudrate-Generator
- Functionally compatible with SAB 80C515
- Extended power saving mode
- Fast Power-On Reset
- Six ports: 48 I/O lines, 8 input lines
- Three temperature ranges available:
  0 to 70°C (T1)
  0 to 85°C (T3)
  40 to 110°C (T4)
- Plastic package: P-LCC-68

The SAB 80C515A/83C515A-5 is a high-end member of the Siemens SAB 8051 microcontroler family. It is designed in Siemens ACMOS technology and based on the SAB 8051 architecture. ACMOS is a technology which combines high-speed and density characteristics with low-power consumption or dissipation.

While maintaining all the SAB 80C515 features and operating characteristics the SAB 80C515A/83C515A-5 contains more on-chip RAM/ROM. Furthermore a new 10-bit A/D-Converter is implemented as well as extended security mechanisms. The SAB 80C515A is identical with the SAB 83C515A-5 except that it lacks the on-chip program memory. The SAB 80C515A/83C515A-5 is supplied in a 68-pin plastic leaded chip carrier package (P-LCC-68).

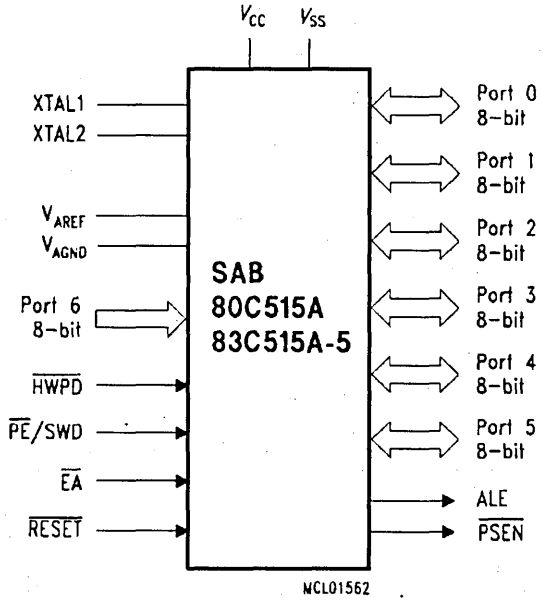
Figure 81
SAB 80C515A / 83C515A-5



Ordering Information

Type	Package	Description 8-bit CMOS- microcontroller
SAB 80C515A-N18	PL-CC-68	for external memory, 18 MHz
SAB 80C515A-5N18	PL-CC-68	with mask-programmable ROM, 18 MHz
SAB 80C515A-N18-T3	PL-CC-68	for external memory, 18 MHz ext. temperature -40 to 85°C
SAB 83C515A-5N18-T3	PL-CC-68	with mask-programmable ROM, 18 MHz ext. temperature -40 to 85°C

Figure 82
Logic Symbol

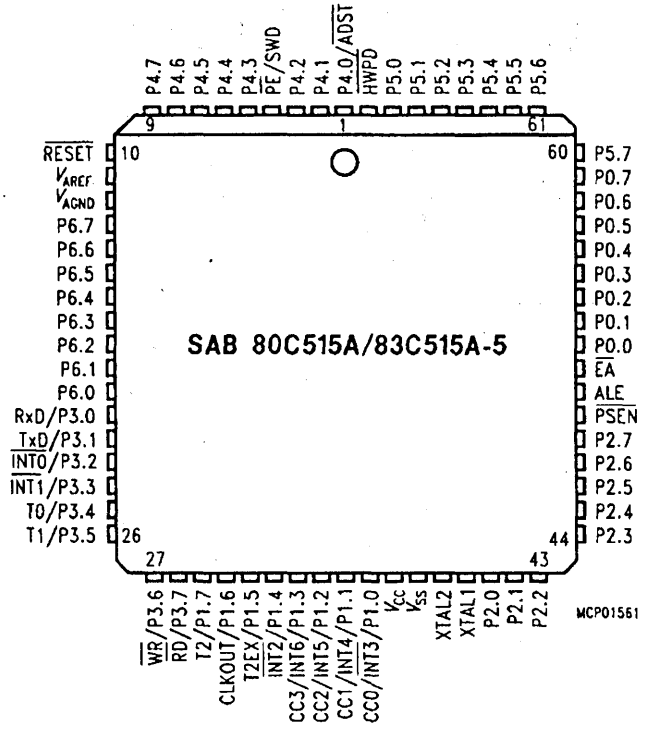


4

The pin functions of the SAB 80C515A are identical with those of the SAB 80C515 with following exception:

Pin	SAB 80C515A	SAB 80C515
68	$\overline{\text{HWPD}}$	V cc
1	P4.0/$\overline{\text{ADST}}$	P4.0
4	/SWD	$\overline{\text{PE}}$

Figure 83
Pin Configuration (P-LCC-68)



Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P4.0 - P4.7	1 -3, 5 - 9	I/O	**Port 4** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 4 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. P4 also contains the external A/D converter control pin. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary function assigned to port 6: - $\overline{ADST}$ (P4.0): external A/D converter start pin
$\overline{PE}$ / SWD	4	I	**Power saving modes enable/Start Watchdog Timer** A low level on this pin allows the software to enter the power down, idle and slow down mode. In case the low level is also seen during reset, the watchdog timer function is off on default. Use of the software controlled power saving modes is blocked, when this pin is held on high level. A high level during reset performs an automatic start of the watchdog timer immediately after reset. When left unconnected this pin is pulled high by a weak internal pull-up resistor.
$\overline{RESET}$	10	I	**$\overline{RESET}$ pin** A low level on this pin for the duration of one machine cycle while the oscillator is running resets the SAB 80C515A. A small internal pull-up resistor permits power-on reset using only a capacitor connected to Vss.
V_{AREF}	11		**Reference voltage** for the A/D converter.
V_{AGND}	12		**Reference ground** for the A/D converter.
P6.7 - P6.0	13 - 20	I	**Port 6** is an 8-bit unidirectional input port to the A/D converter. Port pins can be used for digital input, if voltage levels simultaneously meet the specifications high/low input voltages, and for the eight multiplexed analog inputs.

Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P3.0 - P3.7	21 - 28	I/O	**Port 3** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – R x D: serial port's receiver data input (asynchronous) of data input/ output (synchronous) – T x D (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) – $\overline{\text{INT0}}$ (P3.2): interrupt 0 input/timer 0 gate control input – $\overline{\text{INT1}}$ (P3.3): interrupt 1 input/timer 1 gate control input – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – $\overline{\text{WR}}$ (P3.6): the write control signal latches the e data byte from port 0 into the external data memory – $\overline{\text{RD}}$ (P3.7): the read control signal enables the external data memory to port 0

Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P1.7 - P1.0	29 - 36	I/O	**Port 1** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. The port is used for the low order address byte during program verification. Part 1 also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: – $\overline{\text{INT3}}$/CC0 (P1.0): interrupt 3 input/ compare 0 output/ capture 0 input – INT4/CC1 (P1.1): interrupt 4 input/ compare 1 output/ capture 1 input – INT5/CC2 (P1.2): interrupt 5 input/ compare 2 output/ capture 2 input – INT6/CC3 (P1.3): interrupt 6 input/ compare 3 output/ capture 3 input – $\overline{\text{INT2}}$ (P1.4): interrupt 2 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input
XTAL2	39	-	**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.

4

Pin Definitions and Functions

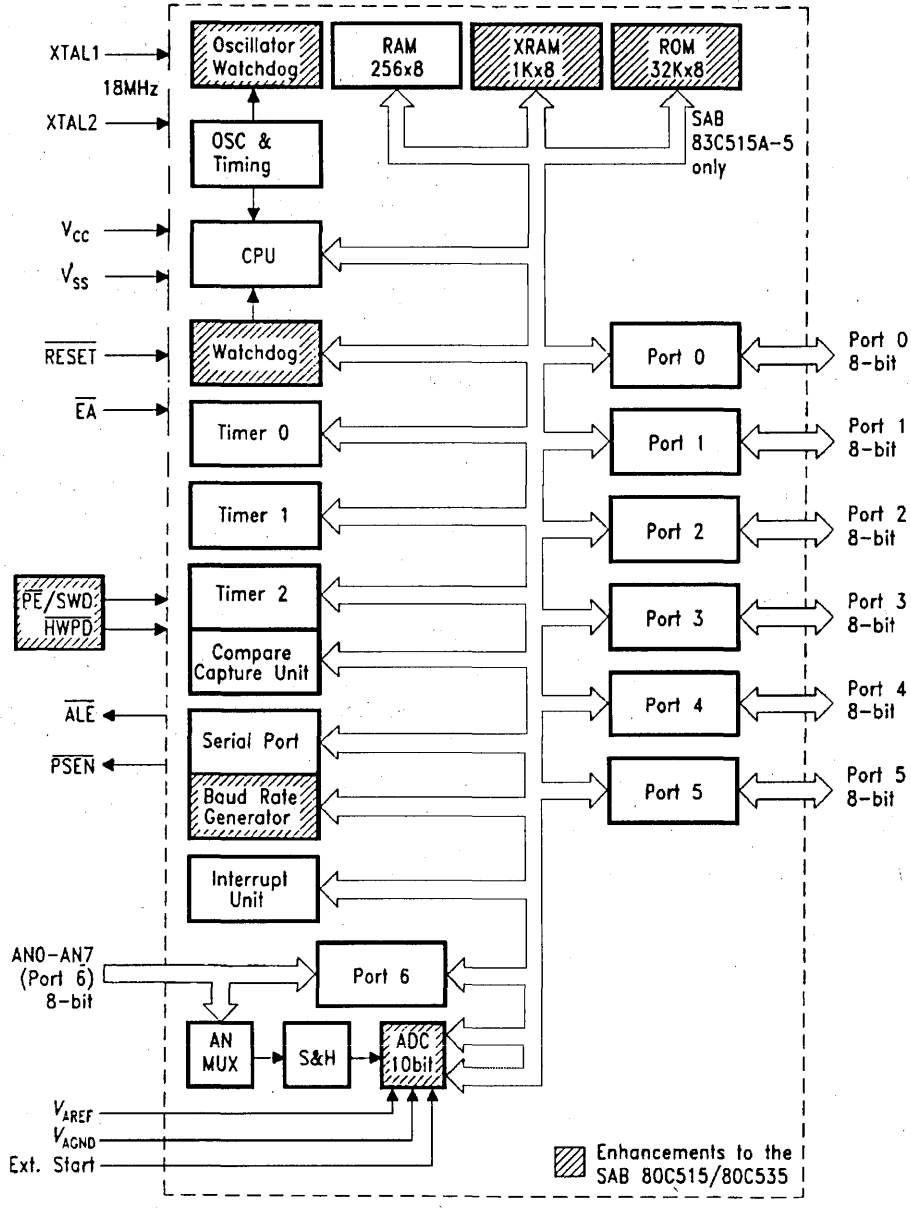
Symbol	Pin	Input (I) Output (O)	Function
XTAL1	40	-	**XTAL1** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times and rise/fall times specified in the AC characteristics must be taken into account.
P2.0 - P2.7	41 - 48	I/O	**Port2** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DOPTR). In this application it uses strong internal pull-up resistors when issuing 1's. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function execution.
$\overline{\text{PSEN}}$	49	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external data memory accesses. The signal remains high during internal program execution.
ALE	50	O	The **Address Latch Enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
EA	51	I	**External Access Enable** When held high, the SAB 80C515A executes instructions from the internal ROM as long as the PC is less than 8192. When held low, the SAB 80C515A-5 fetches all instructions from external program memory. For the SAB 83C515A-5 this pin must be tied low.

Pin Definitions and Functions

Symbol	Pin	Input (I) Output (O)	Function
P0.0 - P0.7	52 - 59	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1's. Port 0 also outputs the code bytes during program verification in the SAB 80C515A. External pull-up resistors are required during program verification.
P5.7 - P5.0	60 - 67	I/O	**Port 5** is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 5 pins that have 1's written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors.
$\overline{\text{HWPD}}$	68	I	**Hardware Power Down** A low level on this pin for the duration of one machine cycle while the oscillator is running resets the SAB 80C515A. A low level for a longer period will force the part to Power Down Mode with the pins floating.
V cc	37	-	**Supply voltage** during normal, idle, and power-down operation.
V ss	38	-	Ground (0 V)

4

Figure 84
Block Diagram



Functional Description

The SAB 80C515A is based on 8051 architecture. It is a fully compatible member of the Siemens SAB 8051/80C51 microcontroller family being an significantly enhanced SAB 80C515. The SAB 80C515A is therefore code compatible with the SAB 80C515.

Having an 8-bit CPU with extensive facilities for bit-handling and binary BCD arithmetics the SAB 80C515A is optimized for control applications. With a 18 MHz crystal, 58% of the instructions are executed in 666.67 ns.

While maintaining all architectural and operational characteristics of the SAB 80C515 the SAB 80C515A incorporates more on-chip RAM. A new 10-bit A/D-Converter is implemented as well as an oscillator watchdog unit. Also the maximum operating frequency of 18 MHz is higher than at the SAB 80C515.

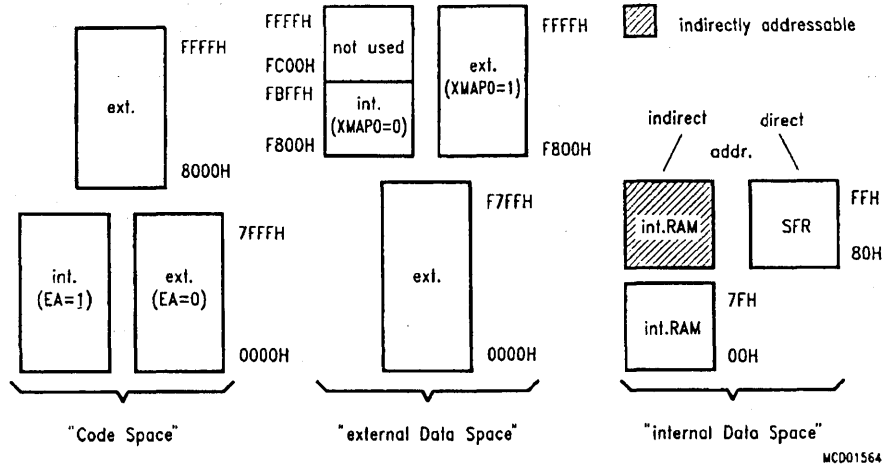
With exception of the ROM sizes both parts are identical. Therefore the therm SAB 80C515A refers to both versions within this specification unless otherwise noted.

Memory Organization

According to the SAB 8051 architecture, the SAB 80C515A has separate address spaces for program and data memory. Figure 85 Memory Map illustrates the mapping of the address spaces.

4

Figure 85
Memory Map



– Program Memory ('Code Space')

The SAB 83C515A-5 has 32 Kbyt of on-chip ROM, while the SAB 80C515A has no internal ROM. The program memory can externally be expanded up to 64 Kbyte. Pin EA determines whether program fetches below address 8000H are done from internal or external memory.

As a new feature the SAB 83C515A offers the possibility of protecting the internal ROM against unauthorized access. This protection is implemented in the ROM-Mask. Therefore, the decision ROM-Protection 'yes' or 'no' has to be made when delivering the ROM-Code. Once enabled, there is no way of disabling the ROM-Protection.

Effect: The access to internal ROM done by an externally fetched MOVC instruction is disabled. Nevertheless, an access from internal ROM to external ROM is possible.

To verify the read protected ROM-Code a special ROM-Verify-Mode is implemented. This mode also can be used to verify unprotected internal ROM.

ROM-Protection	ROM-Verification Mode (see 'AC Characteristics')	Restrictions
no	ROM-Verification Mode1 (standard 8051 Verification Mode) ROM-Verification Mode 2	
yes	ROM-Verification Mode 2	– standard 8051 Verification Mode is disabled – externally applied MOVC accessing internal ROM is disabled

– Data Memory ('Data Space')

The data memory space consists of an internal and an external memory space. The SAB 80C515A contains another 1 Kbyte on On-Chip RAM additional to the 256-bytes internal RAM of the base type SAB 80C515. This RAM is called XRAM ('eXtended RAM') in this document.

– External Data Memory

Up to 64 Kbyte external data memory can be addressed by instructions that use 8-bit or 16-bit indirect addressing. For 8-bit addressing MOVX instructions in combination with registers R0 and R1 can be used. A 16-bit external memory addressing is supported by a 16-bit datapointer. Registers XPAGE and SYSCON are controlling whether data fetches at addresses F800H to FBFFH are done from internal XRAM or from external data memory.

– Internal Data Memory

The internal data memory is divided into four physically distinct blocks:

- the lower 128 bytes of RAM including four register banks containing eight registers each the upper 128 byte of RAM
- the 128 byte special function register area.
- a 1 K x 8 area which is accessed like external RAM (MOVX-instructions), implemented on chip at the address range from F800H to FBFFH. Special Function Register SYSCON controls whether data is read from or written to XRAM or external RAM.

A map of the internal data memory is shown in Figure 85 Memory Map. The overlapping address spaces of the standard internal data memory (256 bytes) are accessed by different addressing modes (see User's Manual SAB 80C515). The stack can be located anywhere in the internal data memory.

Architecture of the XRAM

The contents of the XRAM is not affected by a reset or HW Power Down. After power-up the contents is undefined, while it remains unchanged during and after a reset or HW Power Down if the power supply is not turned off.

The additional On-Chip RAM is logically located in the "external data memory" range at the upper end of the 64 Kbyte address range (F800H-FBFFH). Nevertheless when XRAM is enabled the address range is F800H-FBFFH is occupied. This is done to assure software compatibility to SAB 80C517A. It is possible to enable and disable (only by reset) the XRAM. If it is disabled the device shows the same behavior as the parts without XRAM, i.e. all MOVX accesses use the external bus to physically external data memory.

4

Accesses to XRAM

Because the XRAM is used in the same way as external data memory the same instruction types must be used for accessing the XRAM.

Note: If a reset occurs during a write operation to XRAM, the effect on XRAM depends on the cycle which the reset is detected at (MOVX is a 2-cycle instruction):

Reset detection at cycle 1:
The new value will not be written to XRAM. The old value is not affected.

Reset detection at cycle 2:
The old value in XRAM is overwritten by the new value.

– Accesses to XRAM using the DPTR

There are a Read and a Write instruction from and to XRAM which use one of the 16-bit DPTR for indirect addressing. The instructions are:

MOVX A, @DPTR (Read)

MOVX @DPTR, A (Write)

Normally the use of these instructions would use a physically external memory. However, in the SAB 80C515A the XRAM is accessed if it is enabled and if the DPTR points to the XRAM address space (DPTR >=F800H).

– Accesses to XRAM using the Registers R0/R1

The 8051 architecture provides also instructions for accesses to external data memory range which use only an 8-bit address (indirect addressing with registers R0 or R1). The instructions are:

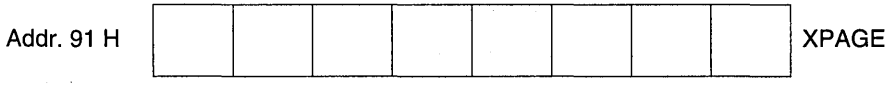
MOVX A, @Ri (Read)

MOVX @Ri, A (Write)

In application systems, either a real 8-bit bus (with 8-bit address) is used or Port 2 serves as page register which selects pages of 256-byte. However, the distinction, whether Port 2 is used as general purpose I/O or as "page address" is made by the external system design. From the device's point of view it cannot be decided whether the Port 2 data is used externally as address or as I/O data.

Hence, a special register is implemented into the SAB 80C515A to provide the possibility of accessing the XRAM also with the MOVX @Ri instructions, i.e. XPAGE serves the same function for the XRAM as Port 2 for external data memory.

Figure 86 Special Function Register XPAGE



The rest value of XPAGE is 00H
XPAGE can be set and read by software.

The register XPAGE provides the upper address byte for accesses to XRAM with MOVX @Ri instructions. If the address formed from XPAGE and Ri is less than the XRAM address range, then an external access is performed. For the SAB 80C515A the contents of XPAGE must be greater or equal then F8H in order to use the XRAM. Of course, the XRAM must be enabled if it shall be used with MOVX @Ri instructions.

Thus, the register XPAGE is used for addressing of the XRAM; additionally its contents are used for generating the internal XRAM select. If the contents of XPAGE is less than the XRAM address range then an external bus access is performed where the upper address byte is provided by P2 and not by XPAGE.

Therefore, the software has to distinguish two cases, if the MOVX @Ri instructions with paging shall be used:

a) Access to XRAM:
The upper address byte must be written to XPAGE or P2; both writes select the XRAM address range.

b) Access to external memory:
The upper address byte must be written to P2; XPAGE will be loaded with the same address in order to deselect the XRAM.

4

Figure 87 Special Function Register SYSCON

Addr. 0B1 H	–	–	–	–	–	–	XMAP1	XMAP0	SYSCON

Bit	Function
XMAP0	Global enable/disable bit for XRAM memory. XMAP0 = 0: The access to XRAM (= On-Chip XDATA memory) is enabled. XMAP0 = 1: The access to XRAM is disabled. All MOVX accesses are performed by the external bus (reset state).
XMAP1	Control bit for $\overline{RD}$/$\overline{WR}$ signals during accesses to XRAM; this bit has no effect if XRAM is disabled (XMAP0 = 1) or if addresses exceeding the XRAM address range are used for MOVX accesses. XMAP1 = 0: The signals $\overline{RD}$ and $\overline{WR}$ are not activated during accesses to XRAM. XMAP1 = 1: The signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM.

Control of XRAM in the SAB 80C515A

There are two control bits in register SYSCON which control the use and the bus operation during accesses to the additional On-Chip RAM (XRAM).

Reset value of SYSCON is XXXX XX01B.

The control bit XMAP0 is a global enable/disable bit for the additional On-Chip RAM (XRAM). If this bit is set, the XRAM is disabled, all MOVX accesses use external memory via the external bus. In this case SAB 80C515A does not use the additional On-Chip RAM and is compatible with the types without XRAM.

XMAP0 is hardware protected by an unsymmetric latch. An unintentional disabling of XRAM could be dangerous since indeterminate values would be read from the external bus. To avoid this the XMAP-bit is forced to '1' only by reset. Additionally, during reset an internal capacitor is loaded. So after reset state XRAM is disabled. Because of the load time of the capacitor XMAP0-bit once written to '0' (that is, discharging capacitor) cannot be set to '1' again by software. On the other hand any distortion (software hangup, noise, ...) is not able to load this capacitor, too. That is, the stable status is XRAM enabled. The only way to disable XRAM after it was enabled is a reset.

The clear instruction for XMAP0 should be integrated in the program initialization routine before XRAM is used. In extremely noisy systems the user may have redundant clear instructions.

The control bit XMAP1 is relevant only if the XRAM is accessed. In this case the external $\overline{RD}$ and $\overline{WR}$ signals at P3.6 and P3.7 are not activated during the access, if XMAP1 is cleared. For debug purposes it might be useful to have these signals and the addresses at Ports 0.2 available. This is performed if XMAP1 is set.

The behavior of Port 0 and P2 during a MOVX access depends on the control bits in register SYSCON and on the state of pin EA. The Table 38 lists the various operating conditions. It shows the following characteristics:

a) Use of P0 and P2 pins during the MOVX access.

Bus: The pins work as external address/data bus. If (internal) XRAM is accessed, the data written to the XRAM can be seen on the bus in debug mode.

I/O: The pins work as Input/Output lines under control of their latch.

b) Activation of the RD and WR pin during the access.

c) Use of internal or external XDATA memory.

The shaded areas describe the standard operation as each 8051 device without on-chip XRAM behaves.

4

Table 38:
Behavior of P0/P2 and RD/WR during MOVX accesses

		EA = 0			EA = 1		
		XMAP1, XMAP0			XMAP1, XMAP0		
		00	10	X1	00	10	X1
MOVX @DPTR	DPTR<XRAM address range	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used
	DPTR≥>XRAM address range	a) P0/P2→Bus ($\overline{WR}$ -Data only) b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	a) P0/P2→Bus ($\overline{WR}$ -Data only) b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used	a) P0/P2→I/O b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	a) P0/P2→Bus ($\overline{WR}$ -Data only) b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	▨ a) P0/P2→Bus b) $\overline{RD}/\overline{WR}$ active c)ext. memory is used
MOVX @Ri	XPAGE<XRAM addr. page range	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used
	XPAGE≥XRAM addr. page range	a) P0/P2→Bus ($\overline{WR}$ -Data only) P2→I/O b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	a) P0/P2→Bus ($\overline{WR}$ -Data only) P2→I/O b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used	a) P0/P2→I/O b) $\overline{RD}/\overline{WR}$ inactive c) XRAM is used	a) P0 Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) XRAM is used	▨ a) P0→Bus P2→I/O b) $\overline{RD}/\overline{WR}$ active c) ext. memory is used

▨ modes compatible to 8051 - family

Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area. The special function registers include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripherals. There are also 128 directly addressable bits within the SFR area. All special function registers are listed in Table 39 and Table 40.

In Table 39 they are organized in numeric order of their addresses. In Table 40 they are organized in groups which refer to the functional blocks of the SAB 80C515A.

Table 39
Special Function Register

Address	Register	Contents after Reset	Address	Register	Content after Reset
80H	P0[1]	OFFH	A0H	P2[1]	OFFH
81H	SP	07H	A1H	reserved	XXH[2]
82H	DPL	00H	A2H	reserved	XXH[2]
83H	DPH	00H	A3H	reserved	XXH[2]
84H	(WDTL)		A4H	reserved	XXH[2]
85H	(WDTH)		A5H	reserved	XXH[2]
86H	WDTREL	00H	A6H	reserved	XXH[2]
87H	PCON	00H	A7H	reserved	XXH[2]
88H	TCON[1]	00H	A8H	IEN0[1]	00H
89H	TNOD	00H	A9H	IP0	00H
8AH	TL0	00H	AAH	SRELL	0D9H
8BH	TL1	00H	ABH	reserved	XXH[2]
8CH	TH0	00H	ACH	reserved	XXH[2]
8DH	TH1	00H	ADH	reserved	XXH[2]
8EH	reserved	XXH[2]	AEH	reserved	XXH[2]
8FH	reserved	XXH[2]	AFH	reserved	XXH[2]
90H	P1[1]	OFFH	B0H	P3[1]	OFFH
91H	XPAGE	00H	B1H	SYSCON	XXXX
92H	reserved	XXH[2]	B2H	reserved	XX01B[2]
93H	reserved	XXH[2]	B3H	reserved	XXH[2]
94H	reserved	XXH[2]	B4H	reserved	XXH[2]
95H	reserved	XXH[2]	B5H	reserved	XXH[2]
96H	reserved	XXH[2]	B6H	reserved	XXH[2]
97H	reserved	XXH[2]	B7H	reserved	XXH[2]
98H	S0CON[1]	00H	B8H	IEN1[1]	00H
99H	SBUF	XXH[2]	B9H	IP1	XX00 000B[2]
9AH	reserved	XXH[2]	BAH	SRELH	XXXX
9BH	reserved	XXH[2]	BBH	reserved	XX11B[2]
9CH	reserved	XXH[2]	BCH	reserved	XXH[2]
9DH	reserved	XXH[2]	BDH	reserved	XXH[2]
9EH	reserved	XXH[2]	BEH	reserved	XXH[2]
9FH	reserved	XXH[2]	BFH	reserved	XXH[2]
					XXH[2]

[1] Bit-addressable Special Function Register
[2] X means that the value is indeterminate and the location is reserved

Table 39
Special Function Register

Address	Register	Contents after Reset	Address	Register	Content after Reset
C0H	IRCON[1]	00H	E0H	ACC[1]	00H
C1H	CCEN	00H	E1H	reserved	XXH[2]
C2H	CCL1	00H	E2H	reserved	XXH[2]
C3H	CCH1	00H	E3H	reserved	XXH[2]
C4H	CCL2	00H	E4H	reserved	XXH[2]
C5H	CCH2	00H	E5H	reserved	XXH[2]
C6H	CCL3	00H	E6H	reserved	XXH[2]
C7H	CCH3	00H	E7H	reserved	XXH[2]
C8H	T2CON[1]	00H	E8H	P4[1]	OFFH
C9H	reserved	XXH[2]	E9H	reserved	XXH[2]
CAH	CRCL	00H	EAH	reserved	XXH[2]
CBH	CRCH	00H	EBH	reserved	XXH[2]
CCH	TL1	00H	ECH	reserved	XXH[2]
CDH	TH2	00H	EDH	reserved	XXH[2]
CEH	reserved	XXH[2]	EEH	reserved	XXH[2]
CFH	reserved	XXH[2]	EFH	reserved	XXH[2]
D0H	PSW[1]	00H	F0H	B[1]	00H
D1H	reserved	XXH[2]	F1H	reserved	XXH[2]
D2H	reserved	XXH[2]	F2H	reserved	XXH[2]
D3H	reserved	XXH[2]	F3H	reserved	XXH[2]
D4H	reserved	XXH[2]	F4H	reserved	XXH[2]
D5H	reserved	XXH[2]	F5H	reserved	XXH[2]
D6H	reserved	XXH[2]	F6H	reserved	XXH[2]
D7H	reserved	XXH[2]	F7H	reserved	XXH[2]
D8H	ADCON0[1]	00H	F8H	P5	OFFH
D9H	ADDATH	00H	F9H	reserved	XXH[2]
DAH	ADDATL	00H	FAH	reserved	XXH[2]
DBH	P6	XXH[2]	FBH		
DCH	ADCON	XXXX 0000B	FCH		
DDH	reserved	XXH[2]	FDH		
DEH	reserved	XXH[2]	FEH		
DFH	reserved	XXH[2]	FFH		

[1] Bit-addressable Special Function Register
[2] X means that the value is indeterminate and the location is reserved

Table 40
Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	0E0H[1]	00H
	B	B-Register	0F0H[1]	00H
	DPH	Data Pointer, High Byte	83H	00H
	DPL	Data Pointer, Low Byte	82H	00H
	PSW	Program Status Word Register	0D0H[1]	00H
	SP	Stack Pointer	81H	00H
A/D-Converter	ADCON0	A/D Converter Control Register 0	0D8H[1]	00H
	ADCON1	A/D Converter Control Register 1	0DCH	OXXX 0000B[3]
	ADDATH	A/D Converter Control Register High Byte	0D9H	00H
			0DAH	00H
	ADDATL	A/D Converter Control Register Low Byte		00H
Interrupt System	IEN0	Interrupt Enable Register 0	0A8H[1]	00H
	IEN1	Interrupt Enable Register 1	0B8H[1]	00H
	IP0	Interrupt Priority Register 0	0A9H	00H
	IP1	Interrupt Priority Register 1	0B9H	XX00 0000B[3]
	IRCON	Interrupt Request Control Register	0C0H[1]	00H
	TCON[2]	Timer Control Register	88H[1]	00H
	T2CON[2]	Timer 2 Control Register	0C8H[1]	00H
Compare/ Capture Unit (CCU)	CCEN	Comp./Capture Enable Reg.	0C1H	00H
	CCH1	Comp./Capture Reg. 1, High Byte	0C3H	00H
	CCH2	Comp./Capture Reg. 2, High Byte	0C5H	00H
	CCH3	Comp./Capture Reg. 3, High Byte	0C7H	00H
	CCH4	Comp./Capture Reg. 4, High Byte	0CFH	00H
	CRCH	Comp./Capture Reg. 1, Low Byte	0C2H	00H
	CRCL	Comp./Capture Reg. 2, Low Byte	0C4H	00H
	TH2	Comp./Capture Reg. 3, Low Byte	0C6H	00H
	TL2	Comp./Capture Reg. 4, Low Byte	0CEH	00H
	T2CON	Timer 2, High Byte	0CBH	00H
		Timer 2, Low Byte	0CAH	00H
		Timer 2 Control Register	0CDH	00H
			0CCH	00H
			0C8H[1]	00H
XRAM	XPAGE	Page Address Register for Extened On Chip RAM	91H	00H
	SYSCON	XRAM Control Register	0B1H	XXXX XX01B[3]

[1] Bit-addressable special function registers

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.

[3] X means that the value is indeterminate and the location is reserved.

Table 40
Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
Ports	P0	Port 0	80H[1]	OFFH
	P1	Port 1	90H[1]	OFFH
	P2	Port 2	0A0H[1]	OFFH
	P3	Port 3	0B0H[1]	OFFH
	P4	Port 4	0E8H[1]	OFFH
	P5	Port 5	0F8H[1]	OFFH
	P6	Port 6, Analog/Digital Input	0DBH[1]	OFFH
Pow. Sav. Modes	PCON	Power Control Register	87H	00H
Serial Channels	ADCON[2]	A/D Converter Control Reg.	0D8H[1]	00H
	PCON[2]	Power Control Register	87H	00H
	SBUF	Serial Channel Buffer Reg.	99H	0XXH[3]
	SCON	Serial Channel 0 Control Reg.	98H[1]	00H
	SRELL	Serial Channel 0 Reload Reg., low byte	AAH	D9H
	SRELH	Serial Channel 0 Reload Reg., high byte	BAH	XXXX XX11B[3]
Timer 0/ Timer 1	TCON	Timer Control Register	88H[1]	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Watch-dog	IEN0[2]	Interrupt Enable Register 0	0A8H[1]	00H
	IEN1[2]	Interrupt Enable Register 1	0B8H[1]	00H
	IP0[2]	Interrupt Priority Register 0	0A9H	00H
	IP1[2]	Interrupt Priority Register 1	0B9H	XX00 0000B[3]
	WDTREL	Watchdog Timer Reload Reg.	86H	00H

[1] Bit-addressable special function registers

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.

[3] X means that the value is indeterminate and the location is reserved.

A/D Converter

In the SAB 80C515A a new high performance/high-speed 8-channel 10-bit A/D-Converter (ADC) is implemented. Its successive approximation technique provides 7 µs conversion time (f_{OSC} = 16 MHz). The conversion principle is upward compatible to the one used in the SAB 80C515. The main functional blocks are shown in Figure 88 Block Diagram A/D Converter.

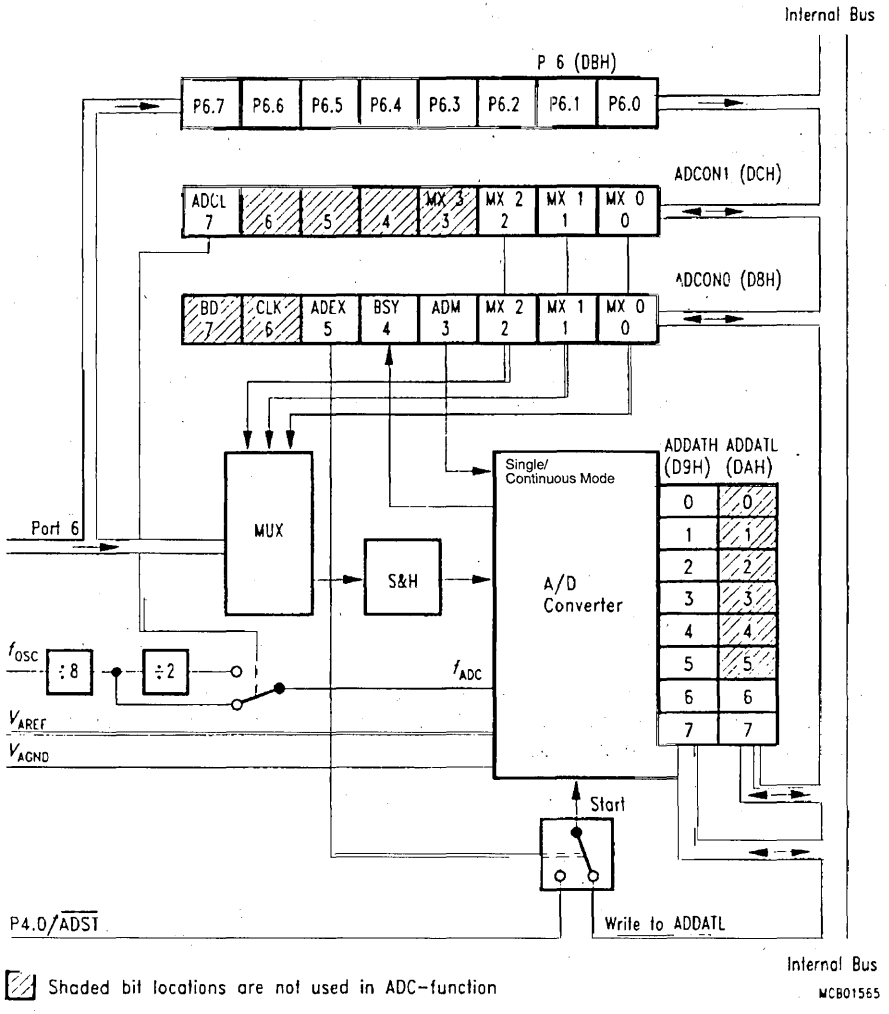
The comparator is a fully differential comparator for a high power supply rejection ratio and very low offset voltages. The capacitor network is binary weighted providing genuine 10-bit resolution.

The table below shows the sample time *Ts* and the conversion time *Tc*, which depend on f_{OSC} and the new prescaler.

f_{OSC} [MHz]	Prescaler	f_{ADC} [MHz]	Sample Time *Ts* [µs]	Conversion Time (incl. sample time) *T*c [µs]
12	÷ 8	1.5	2.67	9.33
	÷16	0.75	5.33	18.66
16	÷ 8	2.0	2.0	7.0
	÷16	1.0	4.0	14.0
18	÷ 8	-	-	-
	÷16	1.125	3.55	12.4

The ADC is clocked (f_{ADC}) with f_{OSC}/8. Because of the ADC's maximum clock frequency of 2 MHz the prescaler (divide-by-2) has to be enabled (set Bit ADCL in SFR ADCON 1) when the oscillator frequency (f_{OSC}) is higher than 16 MHz.

Figure 88
Block Diagram A/D Converter



Timers/Counters

The SAB 80C515A contains three 16-bit timers/counters which are useful in many applications for timing and counting. The input clock for the watch timer/counter is 1/12 of the oscillator frequency in the timer operation or can be taken from an external clock source for the counter operation (maximum count rate is 1/24 of the oscillator frequency).

– Timer/Counter 0 and 1

These timers/counters can operate in four modes:

Mode 0: 8-bit timer/counter with 32:1 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto-reload Mode 3: Timer/counter 0 is configured as one 8-bit timer/counter and one 8-bit timer; Timer/counter 1 in this mode holds its count.

External inputs $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ can be programmed to function as a gate for timer/counters 0 and 1 to facilitate pulse width measurements.

– Timer/Counter 2

Timer/counter 2 of the SAB 80C515A is a 16-bit timer/counter with several additional features. It offers a 2:1 prescaler, a selectable gate function, and compare, capture and reload functions. Corresponding to the 16-bit timer register there are four 16-bit capture/compare registers, one of them can be used to perform a 16-bit reload on a timer overflow or external event. Each of these registers corresponds to a pin or port 1 for capture input/compare output.

Figure 89 Block Diagram of Timer/Counter 2 shows a block diagram of timer/counter 2.

Reload

A 16-bit reload can be performed with the 16-bit CRC register consisting of CRCL and CRCH. There are two modes from which to select:

Mode 0: Reload is caused by a timer 2 overflow (auto-reload).

Mode 1: Reload is caused in response to a negative transition at pin T2EX (P1.5), which can also request an interrupt.

Capture

This feature permits saving of the actual timer/counter contents into a selected register upon an external event or a software write operation. Two modes are provided to latch the current 16-bit value of timer 2 registers TL2 and TH2 into a dedicated capture register.

Mode 0: Capture is performed in response to a transition at the corresponding port 1 pins CC0 to CC3.

Mode 1: Write operation into the low-order byte of the dedicated capture register causes the timer 2 contents to be latched into this register.

Compare

In compare mode, the 16-bit values stored in the dedicated compare registers are compared to the contents of the timer 2 registers. If the count value in the timer 2 registers matches one of the stored values, an appropriate output signal is generated and an interrupt is requested. Two compare modes are provided:

Mode 0: Upon a match the output signal changes from low to high. It goes back to low level when timer 2 overflows.

Mode 1: The transition of the output signal can be determined by software. A timer 2 overflow causes no output change.

Figure 89
Block Diagram of Timer/Counter 2

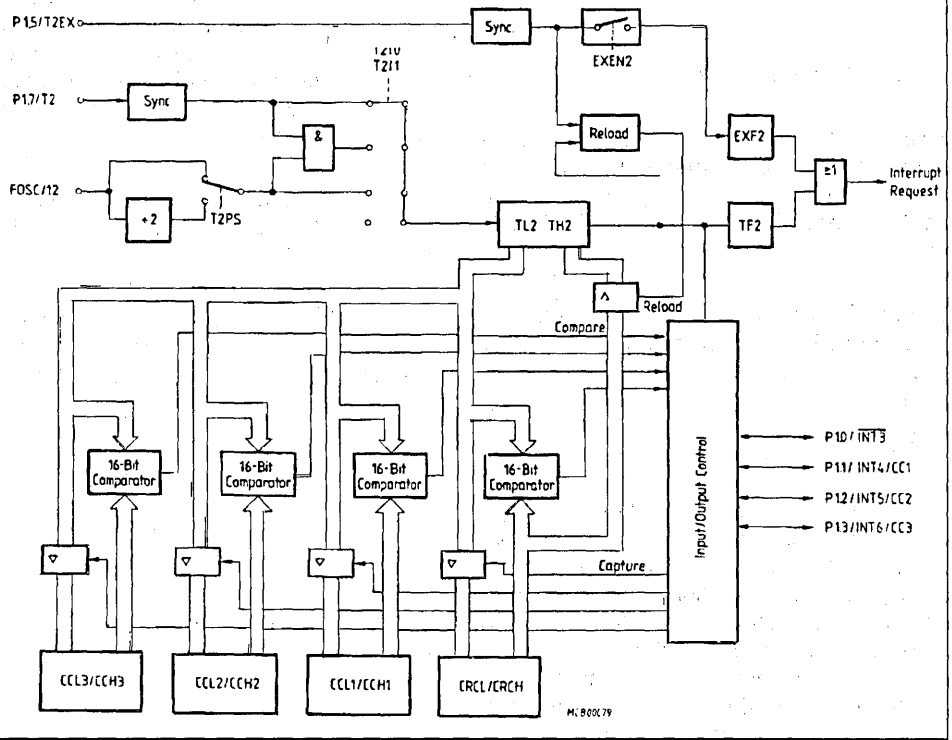


Table 41
Interrupt Sources and Vectors

Source (Request Flags)	Vector Address	Vector
IE0	0003H	External interrupt 0
TF0	000BH	Timer 0 interrupt
IE1	0013H	External interrupt 1
TF1	001BH	Timer 1 interrupt
RI/TI	0023H	Serial port interrupt
TF2+EFX2	002BH	Timer 2 interrupt
IADC	0043H	A/D converter interrupt
IEX2	004BH	External interrupt 2
IEX3	0053H	External interrupt 3
IEX4	005BH	External interrupt 4
IEX5	0063H	External interrupt 5
IEX6	006BH	External interrupt 6

Interrupt Structure

The SAB 80C515A has 12 interrupt vectors with the following vector addresses and request flags.

Each interrupt vector can be individually enabled/disabled. The minimum response time to an interrupt request is more than 3 machine cycles and less than 9 machine cycles, if no other interrupt of the same or a higher priority level is in process.

Figure 90 Interrupt Request Sources shows the interrupt request sources.

External interrupts 0 and 1 can be activated by a low-level or a negative transition (selectable) at their corresponding input pin, external interrupts 2 and 3 can be programmed for triggering on a negative or a positive transition. The external interrupts 3 or 6 are combined with the corresponding alternate functions compare (output) and capture (input) on port 1.

For programming of the priority levels the interrupt vectors are combined to pairs. Each pair can be programmed individually to one of the four priority levels by setting or clearing one bit in special function register IP0 and one in IP1.

Figure 91 Interrupt Priority Level Structure shows the priority level structure.

Figure 90
Interrupt Request Sources

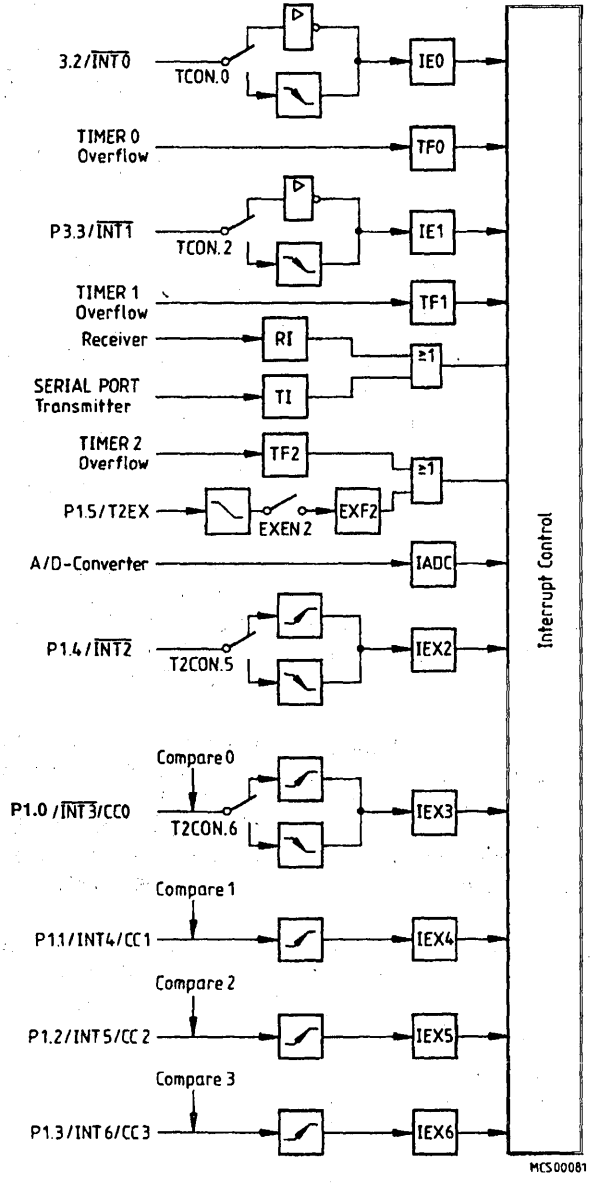
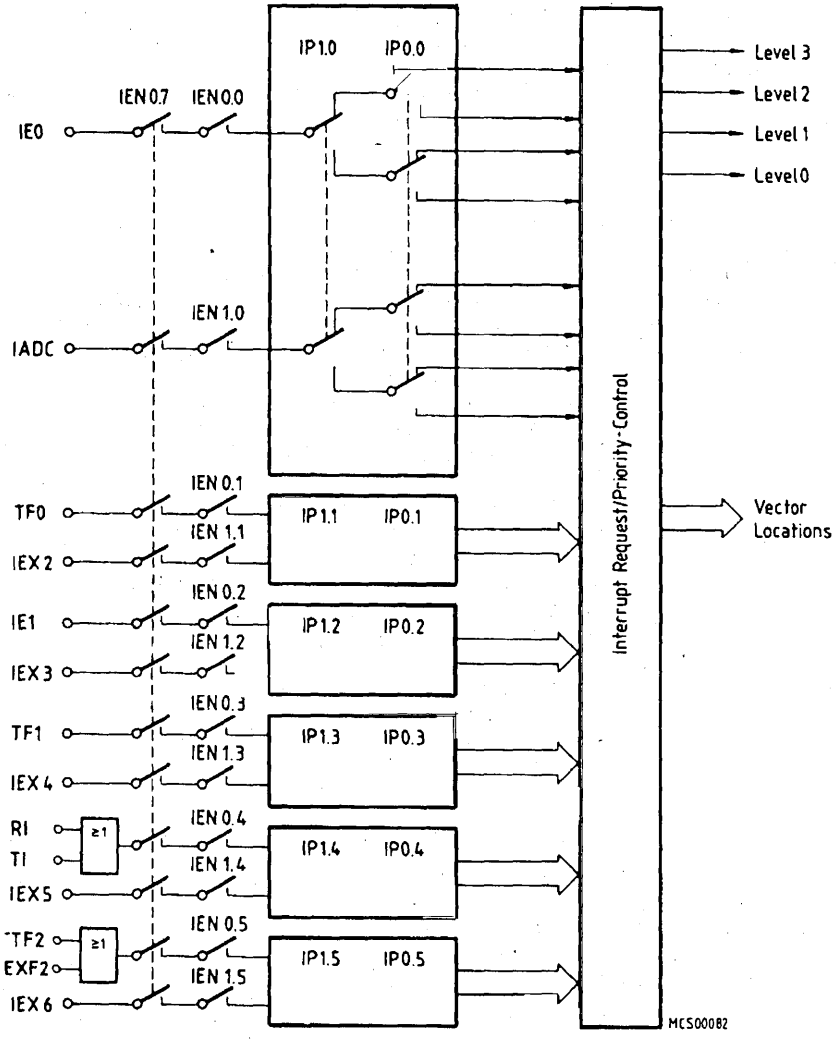


Figure 91
Interrupt Priority Level Structure



I/O Ports

The SAB 80C515A has six 8-bit I/O ports and one input port. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-directional I/O ports with internal pull-up resistors. That means, when configured as inputs, ports 1 to 5 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pull-up FET. Ports 1, 3 and 4 are provided for several alternate functions, as listed below:

Port	Symbol	Function
P1.0	$\overline{\text{INT3}}$/CC0	External interrupt 3 input, compare 0 output, capture 0 input
P1.1	INT4/CC1	External interrupt 4 input, compare 1 output, capture 1 input
P1.2	INT5/CC2	External interrupt 5 input, compare 2 output, capture 2 input
P1.3	$\overline{\text{INT6}}$/CC3	External interrupt 6 input, compare 3 output, capture 3 input
P1.4	$\overline{\text{INT2}}$	External interrupt 2 input
P1.5	T2EX	Timer 2 external reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external count or gate input
P3.0	RxD	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TxD	Serial port's transmitter data output (asynchronous) or data input/output (synchronous)
P3.2	$\overline{\text{INT0}}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe
P4.0	$\overline{\text{ADST}}$	A/D Converter, external start of conversion

The SAB 80C515A has one dual purpose input port. The ANx lines of port 6 in the SAB 80C515 can individually be used as analog or digital inputs. Reading the special function register P6 allows the user to input the digital values currently applied to the port pins. It is not necessary to select these modes by software; the voltages applied at port 6 pins can be converted to digital values using the A/D converter and at the same time the pins can be read via SFR P6. It must be noted, however, that the results in port P6 bits will be indeterminate if the levels at the corresponding pins are not within their V_{IL}/V_{IH} specifications. Furthermore, it is not possible to use port P6 as an output port. Special function register P6 is located at address 0DBH.

In Hardware Power Down Mode the port pins and several control lines enter a floating state. For more details see the section about Hardware Power Down Mode.

Power Saving Modes

The SAB 80C515A provides - due to Siemens ACMOS Technology - four modes in which power consumption can be significantly reduced.

- The **Slow Down Mode**
  The controller keeps up the full operating functionality, but is driven with one eighth of its normal operating frequency. Slowing down the frequency remarkable reduces power consumption.
- The **Idle Mode**
  The CPU is gated off from the oscillator, but all peripherals are still supplied with the clock and continue working.
- The **Software Power Down Mode**
  Operation of the SAB 80C515A is stopped, the on-chip oscillator and the RC-oscillator are turned off. This mode is used to save the contents of the internal RAM with a very low standby current and is fully compatible to the Power Down Mode of the SAB 80C515.
- The **Hardware Power Down Mode**
  Operation of the SAB 80C515A is stopped, the on-chip oscillator and the RC-Oscillator are turned off. The pin $\overline{\text{HWPD}}$ controls this mode. Port pins and several control lines enter a floating state. The Hardware Power Down Mode is new in the SAB 80C515A and is independent of the state of pin $\overline{\text{PE}}$/SWD (which enables only the software initiated power reduction mode).

Hardware Enable for Software controlled Power Saving Modes

A dedicated Pin $\overline{\text{PE}}$/SWD of the SAB 80C515A allows to block the Software controlled power saving modes. Since this pin is mostly used in noise-critical application it is combined with an automatic start of the Watchdog Timer.

$\overline{\text{PE}}$/SWD = V IH (logic high level):
Using of the power saving modes is not possible. The watchdog timer starts immediately after reset. The instruction sequences used for entering of power saving modes will not effect the normal operation of the device.

$\overline{\text{PE}}$/SWD = V IL (logic low level):
All power saving modes can be activated by software. The watchdog timer can be started by software at any time.

When left unconnected, Pin $\overline{\text{PE}}$/SWD is pulled high by a weak internal pull-up. This is done to provide system protection on default.

The logic-level applied to pin $\overline{\text{PE}}$/SWD can be changed during program execution to allow or to block the use of the power saving modes without any effect on the on-chip watchdog circuitry.

Requirements for Hardware Power Down Mode

There is no dedicated pin to enable the Hardware Power Down Mode. The control pin PE/SWD has no control function in this mode. It enables and disables only the use of software controlled power savings modes.

Software Controlled Power Savings Modes

All of these modes are entered by software. Special function register PCON (power control register, address is 87H) is used to select one of these modes.

Slow Down Mode

During slow down operation all signal frequencies that are derived from the oscillator clock, are divided by eight, also the clockout signal and the watchdog timer count.

The slow down mode is enabled by setting bit SD. The controller actually enters the slow down mode after a short synchronization period (max 2 machine cycles).

The slow down mode is disabled by clearing bit SD.

Idle Mode

During Idle Mode all peripherals of the SAB 80C515A (except for the watchdog timer) are still supplied by the oscillator clock. Thus the user has to take care which peripheral should continue to run and which has to be stopped during Idle.

The procedure to enter the Idle mode is similar to the one entering the power down mode. The two bits IDLE and IDLS must be set by two consecutive instructions to minimize the chance of unintentional activating of the idle mode.

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and the instruction to be executed following the RETI instruction will be the one following the instruction that set the bit IDLS.
- The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle mode if the assigned function is on. The control signals ALE and $\overline{\text{PSEN}}$ hold at logic level high (see Table 42).

Software Power Down Mode

The power down mode is entered by two consecutive instructions directly following each other. The first instruction has to set the flag PDE (power down enable) and must not set PDS (power down set). The following instruction has to set the start bit PDS. Bits PDE and PDS will automatically be cleared after having been set.

The instruction set that sets bit PDS is the last instruction executed before going into power down mode. The only exit from power down mode is a hardware reset.

The status of all output lines of the controller can be looked up in Table 42.

Hardware Controlled Power Down Mode

The pin $\overline{\text{HWPD}}$ controls this mode. If it is on logic high level (inactive) the part is running in the normal operating modes. If pin $\overline{\text{HWPD}}$ gets active (low level) the part enters the Hardware Power Down Mode; this is independent of the state of pin $\overline{\text{PE}}$/SWD.

$\overline{\text{HWPD}}$ is sampled once per machine cycle. If it is found active, the device starts a complete internal reset sequence. The watchdog timer is stopped and its status flag WDTS is cleared exactly the same effects as a hardware reset. In this phase the power consumption is not yet reduced. After completion of the internal reset both oscillators of the chip are disabled. At the same time the port pins and several control lines enter a floating state as shown in Table 42. In this state the power consumption is reduced to the power down current IPD. Also the supply voltage can be reduced. Table 42 also lists the voltages which may be applied at the pins during Hardware Power Down Mode without affecting the low power consumption.

Termination of the HWPD Mode

The power down state is maintained while pin $\overline{\text{HWPD}}$ is held active. If HWPD goes to high level (inactive state) an automatic start up procedure is performed:

- First the pins leave their floating condition and enter their default reset state (as they had immediately before going to float state).
- Both oscillators are enabled. The oscillator watchdog's RC oscillator starts up very fast (typ. less than 2 ms).
- Because the oscillator watchdog is active it detects a failure condition if the on-chip oscillator hasn't yet started. Hence, the watchdog keeps the part in reset and supplies the internal clock from the RC oscillator.
- Finally, when the on-chip oscillator has started, the oscillator watchdog releases the part from reset with oscillator watchdog status flag set. When automatic start of the watchdog was enabled ($\overline{\text{PE}}$/SWD connected to V_{CC}), the Watchdog Timer will start, too (with its default reload value for time-out period).
- The $\overline{\text{RESET}}$ pin overrides the Hardware Power Down function, i.e. if reset gets active during Hardware Power Down it is terminated and the device performs the normal reset function. (Thus, if pin $\overline{\text{RESET}}$ has to be inactive during Hardware Power Down Mode).

4

Table 42
Status of all pins during Idle Mode, Power Down Mode and Hardware Power Down Mode

Pins	Idle Mode Last instruction executed from		Power Down Mode Last instruction executed from		Hardware Power Down Status
	internal ROM	external ROM	internal ROM	external ROM	
P0	Data	float	Data	float[1]	
P1	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output	floating[1]
P2	Data	Address	Data	Data	
P3	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output	outputs
P4	Data /alt.output	Data/ alt.output	Data/ last output	Data/ last output	disabled
P5	Data alt.output	Data/ alt.output	Data/ last output	Data/ last output	input
P6	1)	1)	1)	1)	function
$\overline{EA}$					active input[2]
$\overline{PE}$/SWD					active input pull-up disabled[2]
XTAL1					active output
XTAL2					disabled input function[1]
$\overline{PSEN}$	high	high	low	low	
ALE	high	high	low	low	floating output[1]
*V*AREF, *V*AGND					active supply pins [3]
$\overline{Reset}$					active input must be high

[1] Applied voltage range at pins $V_{SS} \leq V_{IN} \leq V_{CC}$
[2] $V_{IN} = V_{SS}$ or $V_{IN} = V_{CC}$
[3] $V_{SS} \leq V_{CC}$; $V_{AREF} \geq V_{AGND}$

Serial Interface

The SAB 80C515A has a full duplex and receive buffered serial interface. It is functionally identical with the serial interface of the SAB 8051.

Table 43 shows possible configurations and the according baud rates.

Table 43
Baud Rate Generation

8-Bit synchronous channel	**Mode**		**Mode 0**		
	Baudrate	$f$osc = 12 MHz	1 MHz		
		$f$osc = 16 MHz	1.33 MHz		
		$f$osc = 18 MHz	1.5 MHz		
	derived from		fOSC		
8-Bit UART	**Mode**		**Mode 1**		
	Baudrate	$f$osc = 12 MHz	1 Baud - 62.5 kBaud	183 Baud - 375 kBaud	
		$f$osc = 16 MHz	1 Baud - 83 kBaud	244 Baud - 500 kBaud	
		$f$osc = 18 MHz	1 Baud - 93.7 kBaud	2375 Baud - 562.5 kBaud	
	derived from		Timer 1	10-Bit Baudrate Generator	
9-Bit UART	**Mode**		**Mode 2**		**Mode 3**
	Baudrate	$f$osc = 12 MHz	187.5 kBaud375 kBaud	1 Baud - 62.5 kBaud	183 Baud - 375 kBaud
		$f$osc = 16 MHz	250 kBaud/ 500 kBaud	1 Baud - 83.3 kBaud	244 Baud - 500 kBaud
		$f$osc = 18 MHz	281.2 kBaud/ 562.5 kBaud	1 Baud - 93.7 kBaud	275 Baud - 562.5 kBaud
	derived from		$f$osc/2	Timer 1	10-Bit Baudrate Generator

4

The Serial Interface can operate in 4 modes:

Mode 0: Shift register mode:
Serial data enters and exits through R x D. T x D outputs the shift clock 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, variable baud rate:
10-bit are transmitted (through T x D) or received (through R x D): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB80 in special function register SCON. The baud rate is variable.

Mode 2: 9-bit UART, fixed baud rate:
11-bit are transmitted (through T x D) or received (through R x D): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB80 in SCON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB80 or a second stop bit by setting TB80 to 1. On reception the 9th data bit goes into RB80 in special function register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-bit UART, variable baud rate:
11-bit are transmitted (through T x D) or received (through R x D): a start bit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

Variable Baud Rates for Serial Interface

Variable baud rates for modes 1 and 3 of serial interface can be derived from either timer 1 or a new dedicated Baudrate Generator.

The baud rate is generated by a free running 10-bit timer with programmable reload register.

$$\text{Mode 1.3 baud rate} = \frac{2^{\text{SMOD}} * f_{\text{OSC}}}{64 * (2^{10} - \text{SREL})}$$

The default value after reset in the reload registers SRELL and SRELH provides a baud rate of 4.8 kBaud (SMOD - 0) or 9.6 kBaud (SMOD - 1) at 12 MHz oscillator frequency. This guarantees full compatibility to the SAB 80C515.

Fail Safe Mechanisms

The SAB 80C515A offers enhanced fail safe mechanisms, which allow an automatic recovery from software upset or hardware failure:

- a programmable watchdog timer (WDT), with variable time-out period from 512 µs up to appr. 1.1 s @ 12 MHz. Upward compatible to SAB 80C515 watchdog timer.
- an oscillator watchdog (OWD) which monitors the on-chip oscillator and forces the microcontroller into reset state, in case the on-chip oscillator fails; it also controls the restart from the Hardware Power Down Mode and provides the clock for a fast internal reset after power-on.

Programmable Watchdog Timer

The WDT can be activated by hardware of software.

Hardware initialization is done when pin $\overline{PE}$/SWD (Pin 4) is held high during RESET. The SAB 80C515A then starts program execution with the WDT running. Since pin $\overline{PE}$/SWD is only sampled during Reset, the WDT cannot be started externally during normal operation.

Software initialization is done by setting bit SWDT in SFR IEN1.

A refresh of the watchdog timer is done by setting bits WDT (SFR IEN0) and SWDT consecutively. This double instruction sequence has been implemented to increase system security.

When a watchdog timer reset occurs, the watchdog timer keeps on running, but a status flag WDTS (SFR IP0) is set. This flag can also be cleared by software.

Figure 92 Block Diagram of the Programmable Watchdog Timer shows the block diagram of the programmable Watchdog Timer.

Oscillator Watchdog

The unit serves three functions:

- Monitoring of the on-chip oscillator's function.

  The watchdog monitors the on-chip oscillator's frequency; if it is lower than the frequency of the auxiliary RC oscillator in the watchdog unit, the internal clock is supplied by the RC oscillator and the device is forced into reset; if the failure condition disappears (i.e. the on-chip oscillator has again a higher frequency than the RC oscillator), the part executes a final reset phase of appr. 0.25 ms in order to allow the oscillator to stabilize; then the oscillator watchdog reset is released and the part starts program execution again.
- Restart from the Hardware Power Down Mode

  If the Hardware Power Down Mode is terminated the oscillator watchdog has to control the correct start-up of the on-chip oscillator and to restart the program. The oscillator watchdog function is only part of the complete Hardware Power Down sequence; however, the watchdog works identically to the monitoring function.
- Fast internal reset after power-on.
  In this function the oscillator watchdog unit provides a clock supply for the reset before the on-chip oscillator has started. In this case the oscillator watchdog unit also works identically to the monitoring function.

Figure 93 Functional Block Diagram of the Oscillator Watchdog shows the block diagram of the oscillator watchdog unit. It consists of an internal RC oscillator which provides the reference frequency for the frequency comparator.

Figure 92
Block Diagram of the Programmable Watchdog Timer

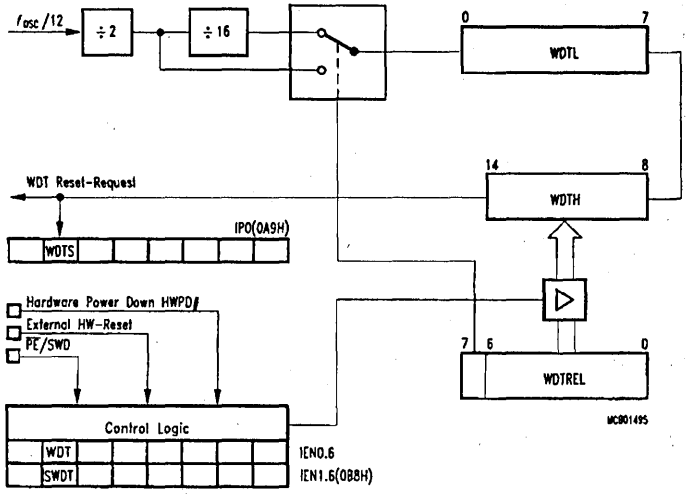
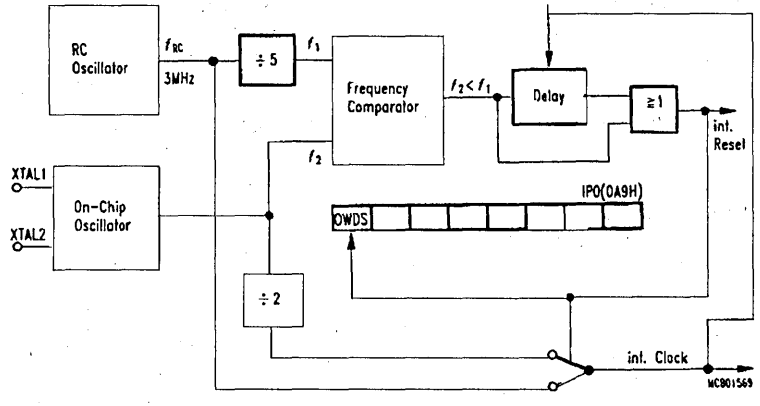


Figure 93
Functional Block Diagram of the Oscillator Watchdog



Fast internal reset after power-on.

The SAB 80C515A can use the oscillator watchdog unit for a fast internal reset procedure after power-on.

Normally members of the 8051 family (like the SAB 80C515) enter their default reset state not before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed in order to bring the device into the correct reset state. Especially if a crystal is used the start up time of the oscillator is relatively long (typ.1 ms). During this time period the pins have an undefined state which could have severe effects e.g. to actuators connected to port pins.

In the SAB 80C515A the oscillator watchdog unit avoids this situation. After power-on the oscillator watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 ms). In the following the watchdog circuitry detects a failure condition for the on-chip oscillator because this has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is valid the watchdog uses the RC oscillator output as clock source for the chip rather than the on-chip oscillator's output. This allows correct resetting of the part and brings also all ports to the defined state.

Delay time between power-on and correct reset state:

Typ.: 18 µs

Max.: 34 µs

Instruction Set

The SAB 80C515A / 83C515A-5 has the same instruction set as the industry standard 8051 microcontroller.

A pocket guide is available which contains the complete instruction set in functional and hexadecimal order. Furthermore, it provides helpful information about Special Function Registers, Interrupt Vectors and Assembler Directives.

Absolute Maximum Ratings

Ambient temperature under bias
SAB 80C515A/83C515A-5 .....0 to 70°C
SAB 80C515A-T3/
83C515A-5-T3 ..... -40 to 85°C
SAB 80C515A-T4/
83C515A-5-T4 ..... -40 to 110°C
Storage temperature..... -65 to 150°C
Voltage on V_{CC} pins with
respect to ground (V_{SS}) ..... -0.1 V to 6.5 V
Power dissipation .....1 W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of may affect device reliability.

DC Characteristics

V_{CC} = 5 V + 10%, - 15%; V_{CC} = 0 V

TA = 0 to 70 °C for the SAB 80C515A
TA = -40 to 85 °C for the SAB 80C515A-T3
TA = -40 to 110 °C for the SAB 80C515A-T4

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, $\overline{RESET}$, $\overline{HWPD}$)	V_{IL}	-0.5	0.2 V_{CC} -0.1	V	-
Input low voltage ($\overline{EA}$)	V_{IL1}	-0.5	0.2 V_{CC} -0.3	V	-
Input high voltage ($\overline{HWPD}$, $\overline{RESET}$)	V_{IH}	-0.5	0.2 V_{CC} + 0.1	V	-
Input high voltage (except $\overline{RESET}$, XTAL2 and $\overline{HWPD}$)	V_{IH}	0.2V_{CC} +0.9	V_{CC} + 0.5	V	-
Input high voltage to XTAL2	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	-
Input high voltage to $\overline{RESET}$ and $\overline{HWPD}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	-
Output low voltage (ports 1, 2, 3, 4, 5)	V_{OL}	-	0.45	V	I_{OL} = 1.6 mA[1)]
Output low voltage (port 0, ALE, $\overline{PSEN}$)	V_{OL1}	-	0.45	V	I_{OL} = 3.2 mA[1)]
Output high voltage (ports 1, 2, 3, 4, 5)	V_{OH}	2.4 0.9 V_{CC}	- -	V V	I_{OH} = -80 µA, I_{OH} = -10 µA

See Notes.

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Output high voltage (port 0 in external bus mode, ALE, PSEN)	V_{OH1}	2.4 0.9 V_{CC}	- -	V V	I_{OH} = -800 µA[2)] I_{OH} = -80 µA[2)]
Logic 0 input current (ports 1, 2, 3, 4, 5)	I_{IL}	-10	-50	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (ports 1, 2, 3, 4, 5)	I_{TL}	-65	-650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{EA}$, P6, $\overline{HWPD}$)	I_{LI}	- -	+100 +150	vA vA	0.45<V_{IN}<V_{CC} 0.45<V_{IN}<V_{CC} TA>100 °C
Input low current to RESET for reset	I_{IL2}	-10	-100	µA	V_{IN} = 0.45 V
Input low current (XTAL2)	I_{IL3}	--10		µA	V_{IN} = 0.45 V
Input low current (PE/SWD)	I_{IL4}	-	-15	µA	V_{IN} = 0.45 V
Pin capacitance	C_{IO}	-	10	pF	fC = 1 MHz, T_A = 25 °C
Power supply current: Active mode, 12 MHz[7)] Active mode, 18 MHz[7)] Idle mode, 12 MHz[7)] Idle mode, 18 MHz[7)] Slow down mode, 12 MHz Slow down mode, 18 MHz Power Down Mode	I_{CC} I_{CC} I_{CC} I_{CC} I_{CC} I_{PD}	- - - - - - -	35 TBD TBD TBD TBD TBD 50	mA mA mA mA mA mA µA	V_{CC} = 5 V,[4)] V_{CC} = 5 V,[4)] V_{CC} = 5 V,[5)] V_{CC} = 5 V,[5)] V_{CC} = 5 V,[6)] V_{CC} = 5 V,[6)] V_{CC} = 2...5.5V,[3)]

See Notes.

4

Notes

1) Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4 and 5. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

2) Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V CC specification when the address lines are stabilizing.

3) I_{PD} (Software Power Down Mode) is measured under the following conditions: $\overline{EA}$ = $\overline{RESET}$ = V_{CC}; Port0 = Port6 = V_{CC}; XTAL1 = N.C.; XTAL2 = V_{SS}; $\overline{PE}$/SWD = VSS; $\overline{HWPD}$ = V_{CC}; V_{AGND} = V_{SS}; V_{ARef} = V_{CC}; all other pins are disconnected.
I_{PD} (Hardware Power Down Mode): independent of any particular pin connection.

4) I_{CC} (active mode) is measured with: XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} - 0.5 V; XTAL1 = N.C.; $\overline{EA}$ = $\overline{PE}$/SWD = V_{CC}; Port0 = Port6 = V_{CC}; $\overline{HWPD}$ = V_{CC}; $\overline{RESET}$ = V_{SS}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

5) I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} - 0.5 V; XTAL1 = N.C.; $\overline{RESET}$ = V_{CC}; $\overline{HWPD}$ = V_{CC}; Port0 = Port6 = V_{CC}; $\overline{EA}$ = $\overline{PE}$/SWD = V_{SS}; all other pins are disconnected;

6) I_{CC} (slow down mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} - 0.5 V; XTAL 1 = N.C.; $\overline{RESET}$ = V_{CC}; $\overline{HWPD}$ = V_{CC}; Port6 = V_{CC}; $\overline{EA}$ = $\overline{PE}$/SWD = V_{SS}; all other pins are disconnected;

7) ICC Max at other frequencies is given by:
active mode: TBD
idle mode: TBD
Where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V.

A/D Converter Characteristics

VCC = 5 V + 10%, - 15%; VSS = 0 V
VAREF = VCC + 5%; VAGND = VSS + 0.2 V;
TA = 0 tp 70 °C for the SAB 80C515A/83C515A-5
TA = -40 to 85 °C for the SAB 80C515A-T3/83C515A-5-T3
TA = -40 to 110 °C for the SAB 80C515A-T4/83C515A-5-T4

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Analog input capacitance	C_I		25	70	pF	
Sample time (Inc. load time)	T_S			4 t_{CY}[1]	µs	[2]
Conversion time (inc. sample time)	T_C			14 t_{CY}[1]	µs	[3]
Total unadjusted error	TUE			+2	LSB	$V_{AREF} = V_{CC}$ $V_{AGND} = V_{SS}$
VAREF supply current			+20		µA	

[1] $t_{CY} = (8*2^{ADCL})/f_{OSC}$ ($t_{CY} = 1/f_{ADC}$; $fADC$; $f_{ADC} = f_{OSC}/(8*2^{ADCL})$

[2] This parameter specifies the time during the input capacitance C_I, can be charged/discharged by the external source. It must be guaranteed, that the input capacitance C_I, is fully loaded within this time. $4T_{CY}$ is 2 µs at the f_{OSC} = 16 MHz. After the end of the sample time TS, changes of the analog input voltage have no effect on the conversion result.

[3] This parameter includes the sample time TS. 14TCY is 7 µs at fOSC = 16 MHz.

4

AC Characteristics

V_{CC} = 5 V + 10%, - 15%; V_{SS} = 0 V

TA = 0 to 70 °C for the SAB 80C515A/83C515A-5

TA = -40 to 85 °C for the SAB 80C515A-T3/83C515A-5-T3

TA = -40 to 110°C for the SAB 80C515A-T4/83C515A-5-T4

(CL for port 0, ALE and PSEN outputs = 100 pF; CL for all other outputs = 80 pF)

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 to 18 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	71	-	$2\,t_{CLCL}$ - 40	-	ns
Address setup to ALE	t_{AVLL}	26	-	t_{CLCL} - 30	-	ns
Address hold after ALE	t_{LLAX}	26	-	t_{CLCL} - 30	-	ns
Address to valid instr in	t_{LLIV}	-	122	-	$4\,t_{CLCL}$-100	ns
ALE to $\overline{PSEN}$	t_{LLPL}	31	-	t_{CLCL} - 25	-	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	132	-	$3\,t_{CLCL}$ - 35	-	ns
$\overline{PSEN}$ to valid instr in	t_{PLIV}	-	92	-	$3\,t_{CLCL}$-75	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	-	0	-	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	-	46	-	t_{CLCL} - 10	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	48	-	t_{CLCL} - 8	-	ns
Address to valid instr in	t_{AVIV}	-	218	-	$5t_{CLCL}$ - 60	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	-	0	-	ns

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 to 18 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	233	-	$6t_{CLCL}$ - 100	-	ns
$\overline{WR}$ pulse width	t_{WLWH}	233	-	$6t_{CLCL}$ - 100	-	ns
Address hold after ALE	t_{LLAX2}	81	-	$2t_{CLCL}$ - 30	-	ns
$\overline{RD}$ to valid data in	t_{RLDV}	-	128	-	$5\ t_{CLCL}$ - 150	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	-	0	-	ns
Data float after $\overline{RD}$	t_{RHDZ}	0	51	-	$2t_{CLCL}$ - 60	ns
ALE to valid data in	t_{AVDV}	-	294	-	$8t_{CLCL}$ - 150	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	117	217	$3t_{CLCL}$ - 50	$3t_{CLCL}$ + 50	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	16	96	t_{CLCL} - 40	t_{CLCL} + 40	ns
Address valid to $\overline{WR}$	t_{AVWL}	92	-	$4t_{CLCL}$ - 130	-	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	11	-	t_{CLCL} - 45	-	ns
Data setup before $\overline{WR}$	t_{QVWH}	239	-	$7t_{CLCL}$ - 150	-	ns
Data hold after $\overline{WR}$	t_{WHQX}	16	-	t_{CLCL} - 150	-	ns
Address float after $\overline{RD}$	t_{RLAZ}	-	0	-	0	ns

4

Figure 94
Program Memory Read Cycle

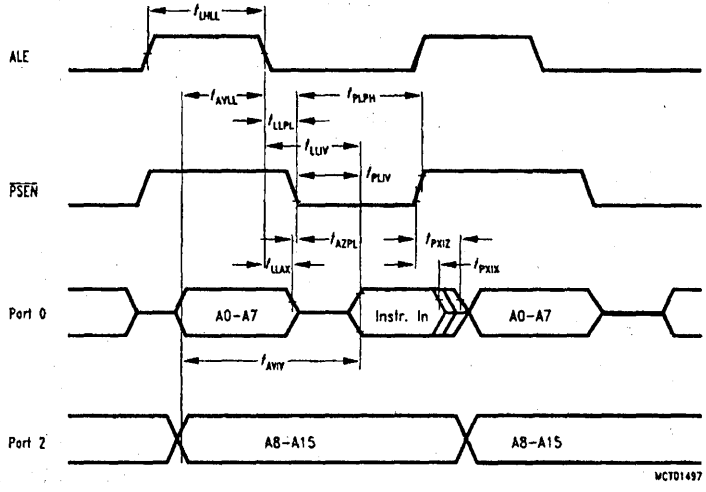


Figure 95
Data Memory Read Cycle

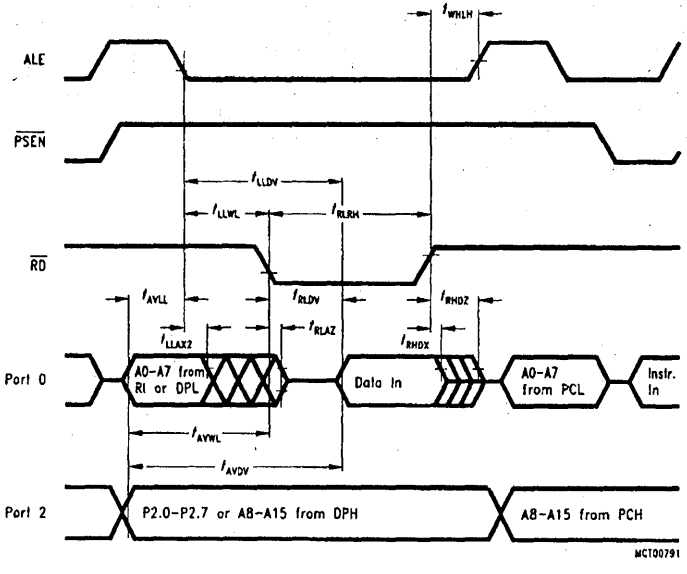
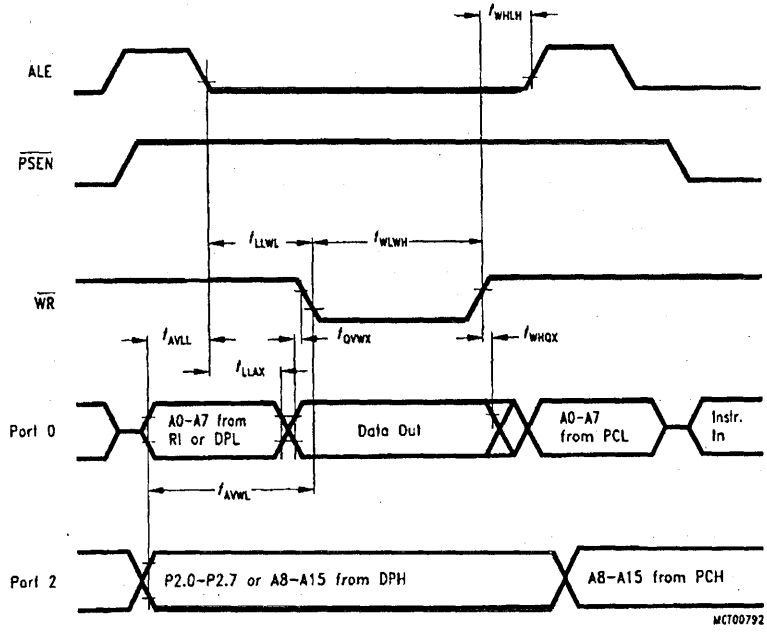


Figure 96
Data Memory Write Cycle

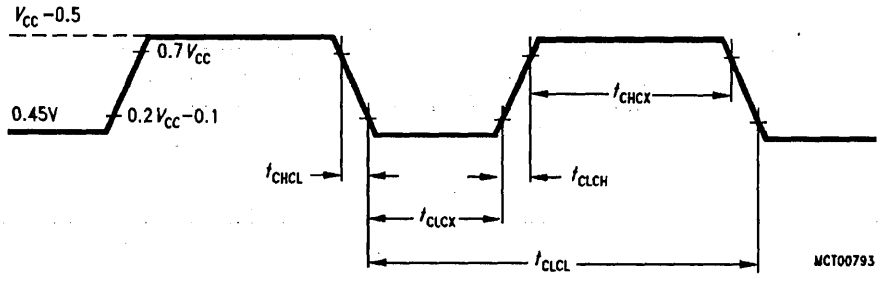


4

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		18 MHz Clock Variable Clock $1/t_{CLCL}$ = 1 to 18 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	55.6	1000	ns
High time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Rise time	t_{CLCH}	-	20	ns
Fall time	t_{CHCL}	-	20	ns
Oscillator frequency	$1/t_{CLCL}$	1	18	MHz

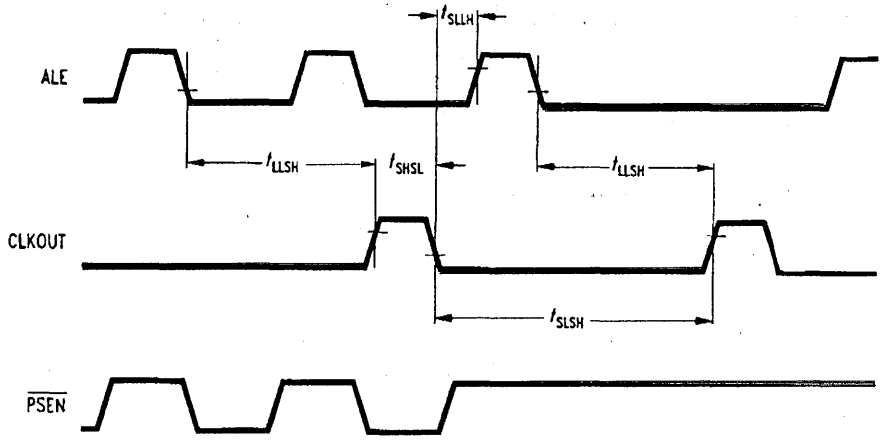
Figure 97
External Clock Cycle



System Clock Timing

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 1 to 18 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	349	-	$7t_{CLCL}$ - 40	-	ns
CLKOUT high time	t_{SHSL}	71	-	$2t_{CLCL}$ - 40	-	ns
CLKOUT low time	t_{SLSH}	516	-	$10t_{CLCL}$ - 40	-	ns
CLKOUT low to ALE high	t_{SLLH}	16	96	t_{CLCL} - 40	t_{CLCL} + 40	ns

Figure 98
System Clock Timing

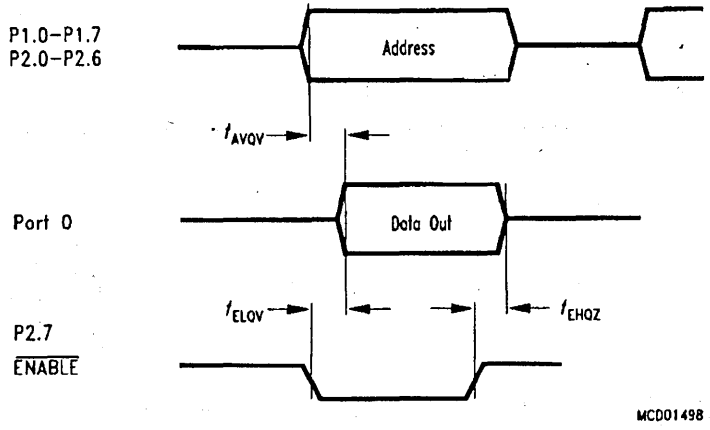


4

ROM Verification Characteristics for SAB 83C515A-5
ROM Verification Mode 1 (Standard Verify Mode for not Read Protected ROM)

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	-	$48t_{CLCL}$	ns
ENABLE to valid data	t_{ELQV}	-	$48t_{CLCL}$	ns
Data float after ENABLE	t_{EHQZ}	0	$48t_{CLCL}$	ns
Oscillator frequency	$1t_{CLCL}$	4	6	MHz

Figure 99
ROM Verification Mode 1

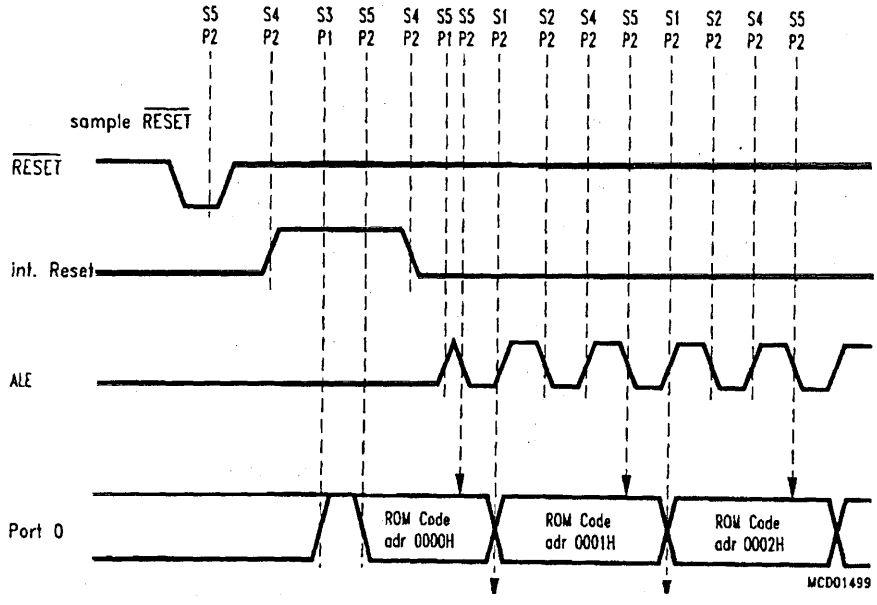


Address: P1.0–P1.7=A0–A7
P2.0–P2.6=A8–A14
Data: Port 0=D0–D7

Inputs $\overline{PSEN}$ = V_{SS}
ALE/$\overline{EA}$ = V_{IH}
$\overline{RESET}$ = V_{IL}

ROM Verification Mode 2 (New Verify Mode for Protected and not Protected ROM)

Figure 100
ROM Verification Mode 2



Inputs $\overline{EA}$=V_{IH}
ALE forced to low level by a weak pull down resistor (approx. 400 Ω), during $\overline{RESET}$ active ROM-Code: Port 0=D0-D7

Outputs Port 3.5 shows all 1024 cycles low level for one cycle when compared ROM-Code was not right

4

Application Example for Verifying the Internal ROM with ROM Verify Mode 2

Figure 101
Application Example for Verifying the Internal ROM with ROM Verify Mode 2

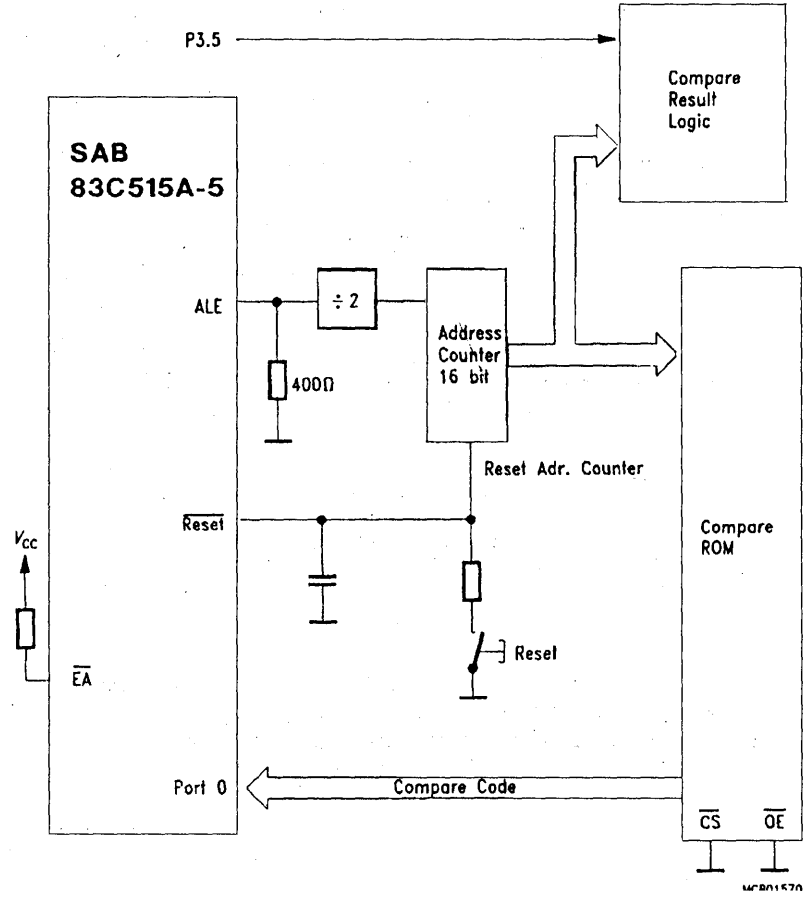
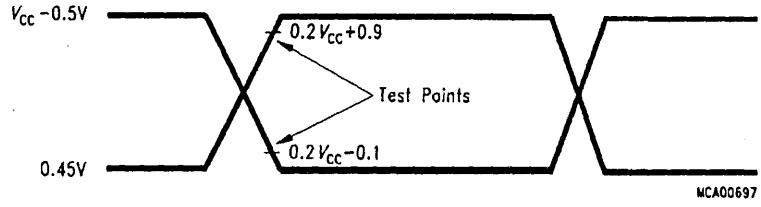
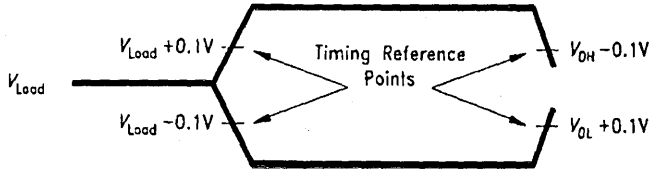


Figure 102
AC Testing: Input, Out Waveforms



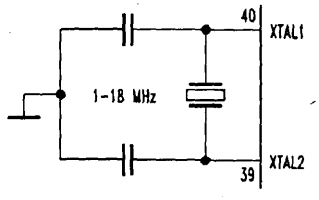
AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at V_{IHmin} for a logic '1' and V_{ILmax} for a logic '0'.

Figure 103
AC Testing: Float Waveforms



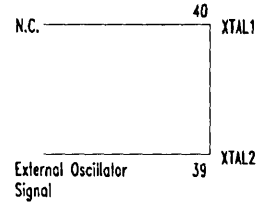
For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

Figure 104
Recommended Oscillator Circuits



C = 30pF ± 10pF
(incl. stray capacitance)

Crystal Oscillator Mode



Driving from External Source

High-Performance 8-Bit CMOS Single-Chip Microcontroller

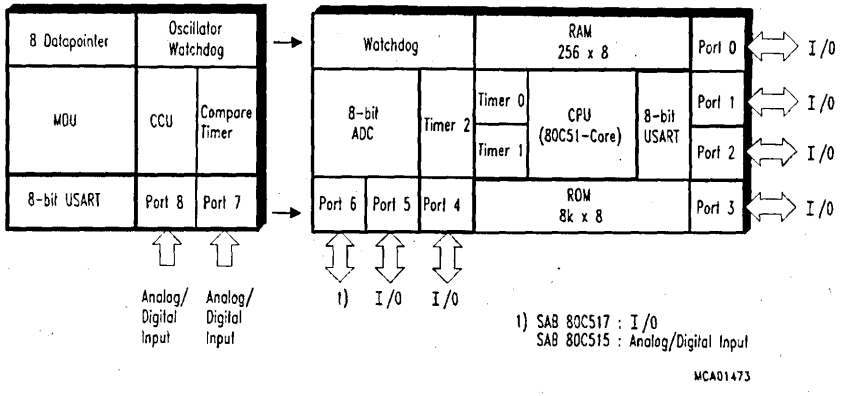
SAB 80C517 **Microcontroller with factory mask-programmable ROM**
SAB 80C537 **Microcontroller for external ROM**

- Versions for 12 MHz and 16 MHz operating frequency
- 8K x 8 ROM (SAB 80C517 only)
- 256 x 8 on-chip RAM
- Superset of SAB 80C51 architecture:
  - 1 µ s instruction cycle time at 12 MHz
  - 750 ns instruction cycle time art 16 MHz
  - 256 directly addressable bits
  - Boolean processor
  - 64 Kbyte external data and program memory addressing
- Four 16-bit timer/counters
- Powerful 16-bit compare/capture unit (CCU) with up to 21 high-speed or PWM output channels and 5 capture inputs
- Versatile "fail-safe" provisions
- Fast 32-bit division, 16-bit 2 multiplication, 32-bit normalize and shift by peripheral MUL/DIV unit (MDU)
- Eight data pointers for external memory addressing
- Fourteen interrupt vectors, four priority levels selectable
- 8-bit A/D converter with 12 multiplexed inputs and programmable ref. voltages
- Two full duplex serial interfaces
- Fully upward compatible with SAB 80C515
- Extended power saving modes
- Nine ports: 56 I/O lines, 12 input lines
- Three temperature ranges available:
  0 to 70°C
  -40 to 85°C
  -40 to 110°C
- Plastic packages: P-LCC-84/P-MRFP-100

The SAB 80C517/80C537 is a high-end member of the Siemens SAB 8051 family of microcontrollers. It is designed in Siemens ACMOS technology and based on the SAB 8051 architecture. ACMOS is a technology which combines high-speed and density characteristics with low-power consumption or dissipation.

While maintaining all the SAB 80C515 features and operating characteristics the SAB 80C517 is expanded in its arithmetic capabilities, "fail-safe" characteristics, analog signal processing and timer capabilities. The SAB 80C537 is identical with the SAB SAB 80C517 except that it lacks the on-chip program memory. The SAB 80C517/SAB 80C537 is supplied in a 84-pin plastic leaded chip carrier package (P-LCC-84) and in a 100-pin plastic quad metric flat package (P-MRFP-100).

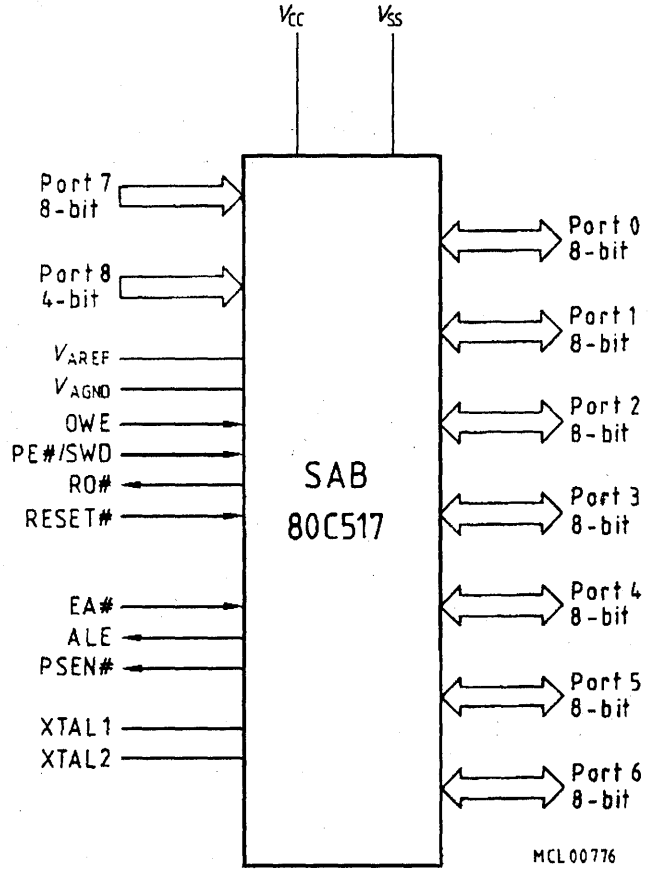
Figure 105
SAB 80C517/80C537



Ordering Information

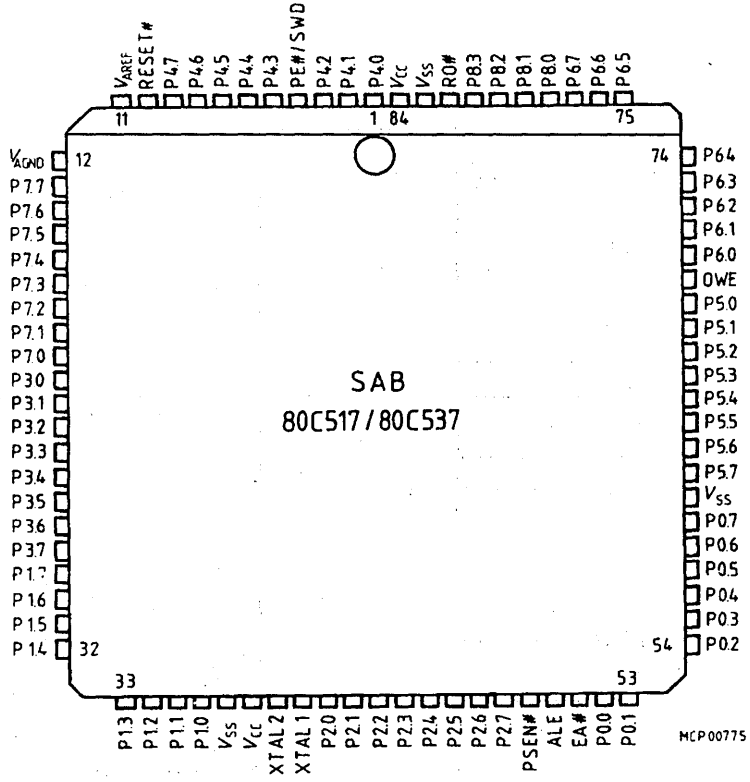
Type	Package	Description 8-bit CMOS microcontroller
SAB 80C517-N	P-LCC-84	with factory mask-programmable ROM, 12 MHz
SAB 80C517-M	P-MRFP-100	
SAB 80C537-N	P-LCC-84	for external memory, 12 MHz
SAB 80C537-M	P-MRFP-100	
SAB 80C517-N-T40/85	P-LCC-84	with factory mask-programmable ROM, 12 MHz, ext. temperature - 40 to 85°C
SAB 80C517-M-T40/85	P-MRFP-100	
SAB 80C537-N-T40/85	P-LCC-84	for external ROM, 12 MHz, ext. temperature - 40 to 85°C
SAB 80C537-M-T40/85	P-MRFP-100	
SAB 80C517-N16	P-LCC-84	with mask-programmable ROM, 16 MHz ext. temperature - 40 to 110°C
SAB 80C517-M16	P-MRFP-100	
SAB 80C537-N16	P-LCC-84	for external memory, 16 MHz
SAB 80C537-M16	P-MRFP-100	
SAB 80C517-N16-T40/85	P-LCC-84	with mask-programmable ROM, 16 MHZ ext. temperature - 40 to 85°C
SAB 80C517-16-N-T40/85	P-LCC-84	with factory mask-programmable ROM, 12 MHz

Figure 106
Logic Symbol



4

Figure 107
Pin Configuration (P-LCC-84)



Pin Definitions and Functions

Symbol	Pin Number		I/O*)	Function
	P-LCC-85	P-MFRP-100		
P4.0 - P4.7	1 - 3, 5 - 9	TBD	I/O	**Port 4** is a bidirectional I/O port with internal pull-up resistors. Port 4 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. This port also serves alternate compare functions. The secondary functions are assigned to the pins of port 4 as follows: - CM0 (P4.0): Compare Channel 0 - CM1 (P4.1): Compare Channel 1 - CM2 (P4.2): Compare Channel 2 - CM3 (P4.3): Compare Channel 3 - CM4 (P4.4): Compare Channel 4 - CM5 (P4.5): Compare Channel 5 - CM6 (P4.6): Compare Channel 6 - CM7 (P4.7): Compare Channel 7
$\overline{PE}$/ SWD	4	TBD	I	**$\overline{\text{Power saving modes enable}}$/ Start Watchdog Timer** A low level on this pin allows the software to enter the power down, idle and slow down mode. In case the low level is also seen during reset, the watchdog timer function is off on default. Use of the power saving modes is blocked, when this pin is held on high level. A high level during reset performs an automatic start of the watchdog timer immediately after reset. When left unconnected this pin is pulled high by a weak internal pull-up resistor.

*I=Input
O=Output

4

Pin Definitions and Functions

RESET	10	TBD	I	**$\overline{\text{RESET}}$** A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80C517. A small internal pull-up resistor permits power-on reset using only a capacitor connected to V_{SS}.
V_{AREF}	11	TBD		**Reference voltage** for the A/D converter.
V_{AREF}	12	TBD		**Reference ground** for the A/D converter.
P7.7 - P7.0	13 - 20	TBD	I	**Port 7** is an 8-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages, and for the lower 8-bit of the multiplexed analog inputs of the A/D converter, simultaneously.
P3.0 - P3.7	21 - 28	TBD	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – R x D0 (P3.0): receiver data input (asynchronous) or data input/output (synchronous) of serial interface – T x D0 (P3.1): transmitter data output (asynchronous) or clock output (synchronous) of serial interface 0 – $\overline{\text{INT0}}$ (P3.2): interrupt 0 input/timer 0 gate control

*I=Input
O=Output

Pin Definitions and Functions

				– $\overline{INT1}$ (P3.3): interrupt 1 input/timer 1 gate control – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – $\overline{WR}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory – $\overline{RD}$ (P3.7): the read control signal enables the external data memory to port 0
P1.7 - P1.0	29 - 36	TBD	I/O	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. It is used for the low order address byte during program verification. It also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: – $\overline{INT3}$/CC0 (P1.0): interrupt 3 input/ compare 0 output / capture 0 input – INT4/CC1 (P1.1): interrupt 4 input / compare 1 output / capture 1 input – INT5/CC2 (P1.2): interrupt 5 input / compare 2 output / capture 2 input

*I=Input
O=Output

4

Pin Definitions and Functions

				– INT6/CC3 (P1.3): interrupt 6 input / compare 3 output / capture 3 input – $\overline{INT2}$/CC4 (P1.4): interrupt 2 input / compare4 output / capture4 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input
XTAL2	39	TBD	-	**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL1	40	TBD	-	**XTAL1** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
P2.0 - P2.7	41 - 48	TBD	I/O	**Port 2** is a bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pull-up resistors. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.

*I=Input
O=Output

Pin Definitions and Functions

$\overline{\text{PSEN}}$	49	TBD	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
ALE	50	TBD	O	The **Address Latch Enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{\text{EA}}$	51	TBD	I	**External Access Enable** When held at high level, introductions fetched from the internal ROM when the PC is less than 8192. When held at low level, the SAB 80C517 fetches all instructions from external program memory. For the SAB 80C537 this pin must be tied low.
P0.0 - P0.7	52 - 59	TBD	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1 s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1 s. Port 0 also outputs the code bytes during program verification in the SAB 80C517. External pull-up resistors are required during program verification.

*I=Input
O=Output

4

Pin Definitions and Functions

P5.7 - P5.0	61 - 68	TBD	I/O	**Port 5** is a bidirectional I/O port with internal pull-up resistors. Port 5 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. This port also serves the alternate function "Concurrent Compare". The secondary functions are assigned to the port 5 pins as follows: - CCM0 (P5.0): concurrent compare 0 - CCM1 (P5.1): concurrent compare 1 - CCM2 (P5.2): concurrent compare 2 - CCM3 (P5.3): concurrent compare 3 - CCM4 (P5.4): concurrent compare 4 - CCM5 (P5.5): concurrent compare 5 - CCM6 (P5.6): concurrent compare 6 - CCM7 (P5.7): concurrent compare 7
OWE	69	TBD	I	**Oscillator Watchdog Enable** A high level on this pin enables the oscillator watchdog. When left unconnected this pin is pulled high by a weak internal pull-up resistor. When held at low level the oscillator watchdog function is off.
P6.0 - P6.7	70 - 77	TBD	I/O	**Port 6** is a bidirectional I/O port internal pull-up resistors. Port 6 pins that have 1 s written to them are pulled high by the internal pull-up resistors, and in that sate can be used as inputs. As inputs, port 6 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 6 also contains the external A/D converter control pin and the transmit and receive pins for serial channel 1. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate.

*I=Input
O=Output

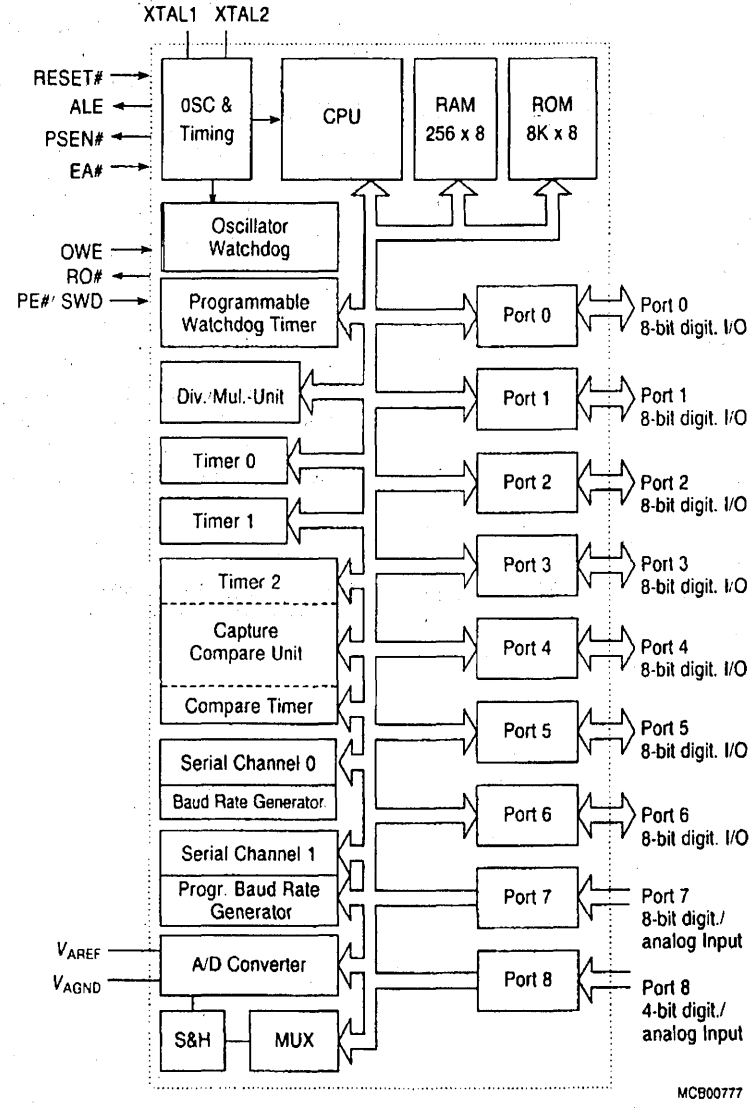
Pin Definitions and Functions

				The secondary functions are assigned to the pins of port 6, as follows: – $\overline{\text{ADST}}$ (P6.0): external A/D converter start pin – R x D1 (P6.1): receiver data input of serial interface 1 – T x D1 (P6.2): transmitter data output of serial interface 1
P8.0 - P8.3	78 - 81	TBD	I	**Port 8** is a 4-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages and for the higher 4-bit of the multiplexed analog inputs of the A/D converter simultaneously.
$\overline{\text{RO}}$	82	TBD	O	**$\overline{\text{Reset Output}}$** This pin outputs the internally synchronized reset request signal. This signal may be generated by an external hardware reset, a watchdog timer reset or an oscillator watchdog reset. The reset output is active low.
V_{ss}	37, 60, 83	TBD	-	**Circuit ground potential**
V_{cc}	38, 84	TBD	-	**Supply terminal** for all operating modes

*I=Input
O=Output

4

Figure 108
Block Diagram



Special Function Registers

All registers, except the program counter and the four general purpose register blanks, reside in the special function register area. The 81 special function registers include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripherals. There are also 128 directly addressable bits within the SFR area. The special function registers are listed in Table 44. In this table they are organized in groups which refer to the functional blocks of the SAB 80C517. Block names and symbols are listed in alphabetical order.

Table 44
Special Function Registers of the SAB 80C517

Address	Symbol	Name	Register	Contents after Reset
CPU	ACC	Accumulator	0E0H [1]	00H
	B	B-Register	0F0H [1]	00H
	DPH	Data-Pointer, High Byte	83H	
	DPL	Data Pointer, Low Byte	82H	
	DPSEL	Data Pointer Select Register	92H	XXXX.X000B [3]
	PSW	Program Status Word Register	0D0H [1]	00H
	SP	Stack Pointer	81H	07H
A/D-Converter	ADCON0	A/D Converter Control Register 0	0D8H [1]	00H
	ADCON1	A/D Converter Control Register 1	0DCH	XXXX.0000B [3]
	ADDAT	A/D Converter Data Register	0D9H	00H
	DAPR	D/A Converter Program Register	0DAH	00H
Interrupt System	IEN0	Interrupt Enable Register0	0A8H [1]	00H
	CTCON [2]	Com. Timer Control P Register	0E1H	0XXX.0000B [3]
	IEN1	Interrupt Enable Register 1	0B8H [1]	00H
	IEN2	Interrupt Enable Register 2	9AH	XXXX.0XX0B [3]
	IP0	Interrupt Priority Register 0	0A9H	00H
	IP1	Interrupt Priority Register 1	0B9H	XX00.0000B [3]
	IRCON	Interrupt Request Control Register	0C0H [1]	00H
	TCON [2]	Timer Control Register	88H [1]	00H
	T2CON [2]	Timer 2 Control Register	0C8H [1]	00H
MUL/DIV Unit	ARCON	Arithmetic Control Register	0EFH	0XXX.XXXXB [3]
	MD0	Multiplication/Division Register 0	0E9H	XXH [3]
	MD1	Multiplication/Division Register 1	0EAH	XXH [3]
	MD2	Multiplication/Division Register 2	0EBH	XXH [3]
	MD3	Multiplication/Division Register 3	0ECH	XXH [3]
	MD4	Multiplication/Division Register 4	0EDH	XXH [3]
	MD5	Multiplication/Division Register 5	0EEH	XXH [3]

[1] Bit-addressable special function registers.

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks

[3] X means that the value is indeterminate.

Table 44
Special Function Registers of the SAB 80C517

Address	Symbol	Name	Register	Contents after Reset
Compare/ Capture- Unit (CCU)	CCEN	Comp./Capture Enable Reg.	Ports	P0
	CC4EN	Comp./Capture 4 Enable Reg.		P1
	CCH1	Comp./Capture Reg. 1, High Byte		P2
	CCH2	Comp./Capture Reg. 2, High Byte		P3
	CCH3	Comp./Capture Reg. 3, High Byte		P4
	CCH4	Comp./Capture Reg. 4, High Byte		P5
	CCL1	Comp./Capture Reg. 1, Low Byte		P6
	CCL2	Comp./Capture Reg. 2, Low Byte		P7
	CCL3	Comp./Capture Reg. 3, Low Byte		P8
	CCL4	Comp./Capture Reg. 4, Low Byte		
	CMEN	Compare Enable Register		
	CMH0	Compare Reg. 0, High Byte		
	CMH1	Compare Reg. 1, High Byte		
	CMH2	Compare Reg. 2, High Byte		
	CMH3	Compare Reg. 3, High Byte		
	CMH4	Compare Reg. 4, High Byte		
	CMH5	Compare Reg. 5, High Byte		
	CMH6	Compare Reg. 6, High Byte		
	CMH7	Compare Reg. 7, High Byte		
	CML0	Compare Register 0, Low Byte		
	CML1	Compare Register 1, Low Byte		
	CML2	Compare Register 2, Low Byte		
	CML3	Compare Register 3, Low Byte		
	CML4	Compare Register 4, Low Byte		
	CML5	Compare Register 5, Low Byte		
	CML6	Compare Register 6, Low Byte		
	CML7	Compare Register 7, Low Byte		
	CMSEL	Compare Input Select		
	CRCH	Com./Rel./Capt. Reg. High Byte		
	CRCL	Com./Rel./Capt. Reg. Low Byte		
	CTCON	Com. Timer Control Reg.		
	CTRELH	Com. Timer Rel. Reg., High Byte		
	CTRELL	Com. Timer Rel. Reg. Low Byte		
	TH2	Timer 2, High Byte		
	TL2	Timer 2, Low Byte		
	T2CON	Timer 2 Control Register		

[1] Bit-addressable special function registers.

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks

[3] X means that the value is indeterminate.

Table 44
Special Function Registers of the SAB 80C517

Address	Symbol	Name	Register	Contents after Reset
Ports	P0	Port 0	80H[1]	FFH
	P1	Port 1	90H[1]	FFH
	P2	Port 2	0A0H[1]	FFH
	P3	Port 3	0B0H[1]	FFH
	P4	Port 4	0E8H[1]	FFH
	P5	Port 5	0F8H[1]	FFH
	P6	Port 6	0FAH	FFH
	P7	Port 7, Analog/Digital Input	0DBH	XXH[3]
	P8	Port 8, Analog/Digital Input, 4-bit	0DDH	XXH[3]
Pow.Sav. Mode	PCON	Power Control Register	87H	00H
Serial Channels	ADCON0[2]	A/D Converter Control Reg.	0D8H[1]	00H
	PCON[2]	Power Contrl Register	87H	00H
	S0BUF	Serial Channel 0 Buffer Reg.	99H	XXH[3]
	S0CON	Serial Channel 0 Control Reg.	98H[1]	00H
	S1BUF	Serial Channel 1 Buffer	9CJH	0XXH[3]
	S1CON	Serial Channel 1 Control Reg.	9BH	0X00.0000B[3]
	S1REL	Serial Channel 1 Reload Reg.	9DH	00H
Timer 0/ Timer 1	TCON	Timer Control Register	88H	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Watchdog	IEN0[2]	Interrupt Enable Register 0	0A8H	00H
	IEN1[2]	Interrupt Enable Register 1	0B8H	00H
	IP0[2]	Interrupt Priority Register 0	0A9H	00H
	IP1[2]	Interrupt Priority Register 1	0B9H	XX00.0000B[3]
	WDTREL	Watchdog Timer Reload Reg.	86H	00H

[1] Bit-addressable special function registers.

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks

[3] X means that the value is indeterminate.

4

Absolute Maximum Ratings

Ambient temperature under bias
SAB 80C517/80C537 ..... 0 to 70°C
SAB 80C517/80C537-
T40/85..... - 40 to 85°C
SAB 80C517/80C537-
T40/110..... - 40 to 110°C
Storage Temperature..... - 65 to 150°C
Voltage on V_{CC} pins with respect
to ground (V_{SS}) ..... -0.5 V to 6.5 V
Voltage on any pin with respect
to ground (V_{SS}) ..... - 0.5 to V_{CC} + 0.5 V
Input current on any pin during
overload condition..... -10mA to +10mA
Absolute sum of all input currents
during overload condition..... 100mA
Power dissipation..... 2 W

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.*

DC Characteristics

V_{CC} = 5V ± 10%; V_{SS} = 0 V;
T_A = 0 to 70°C; for the SAB 80C517/83C537
T_A = - 40 to 85°C; for the SAB 80C517-/83C537-T40/85
T_A = - 40 to 110°C; for the SAB 80C517-/83C537-T40/110

DC Characteristics

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$)	V_{IL}	- 0.5	0.2 V_{CC} - 0.1	V	-
Input low voltage ($\overline{EA}$)	V_{IL1}	- 0.5	0.2 V_{CC} 0.3	V	-
Input low voltage	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	-
Input high voltage to XTAL2	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	-
Input high voltage to $\overline{RESET}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	-
Output low voltage (ports 1, 2, 3, 4, 5, 6)	V_{OL}	-	0.45	V	I_{OL}= 1.6 mA[1]
Output low voltage (port 0, ALE, $\overline{PSEN}$, $\overline{RD}$)	V_{OL1}	-	0.45	V	I_{OL}= 3.2 mA[1]
Output high voltage (ports 1, 2, 3, 4, 5, 6)	V_{OH}	2.4 0.9 V_{CC}	- -	V V	I_{OH} = - 80 µA[2] I_{OH} = - 10 µA[2]

See "Notes" on page 244.

DC Characteristics

Output high voltage (ports 0 in external bus mode, ALE, PSEN, $\overline{RD}$)	V_{OH1}	2.4 0.9 V_{CC}	- -	V V	I_{OH} = - 800 µA I_{OH} = - 80 µA
Logic 0 input current (ports 1, 2, 3, 4, 5, 6)	I_{IL}	-	- 50	µA	V_{IN} = 0.45 V
Input low current to $\overline{RESET}$ for reset	I_{IL2}	-10	- 100	µA	V_{IN} = 0.45
Input low current (XTAL2)	I_{IL3}	-	- 15	µA	V_{IN}= 0.45 V
Input low current (OWE, $\overline{PE}$/SWD)	I_{IL4}	-	- 20	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (ports 1, 2, 3, 4, 5, and 6)	I_{TL}	-65	- 650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{EA}$, ports 7,8)	I_{LI}	-	+ 1	µA	0.45<V_{IN}<V_{CC}
Pin capacitance	C_{IO}	-	10	pF	f_C = 1 MHz, T_A = 25°C
Power-supply current					
- Active mode, 12 MHz[6)]	I_{CC}	-	40	mA	V_{CC} = 5 V,[4)]
- Idle mode, 12 MHz[6)]		-	15	mA	V_{CC} = 5 V,[5)]
- Slow down mode, 12 MHz[6)]		-	15	mA	V_{CC} = 5 V,[5)]
- Active mode, 16 MHz[6)]	I_{CC}	-	52.3	mA	V_{CC} = 5 V,[4)]
- Idle mode, 16 MHz[6)]		-	19	mA	V_{CC} = 5 V,[5)]
- Slow down mode, 16 MHz[6)]		-	19	mA	V_{CC} = 5 V,[5)]
- Power down mode	I_{PD}	-	50	µA	V_{CC} = 2...5.5 V,[3)]

See "Notes" on page 244.

4

A/D Converter Characteristics

V_{CC}= 5 V ± 10% V_{SS} = 0 V;
V_{AREF} = V_{CC} ± 5%; V_{AGND} = V_{SS} ± 0.2 V; $V_{INTAREF}$ - $V_{INTAGND}$ ≥ 1 V

T_A = 0 to 70°C for the SAB 80C517/83C537
T_A = -40 to 85°C for the SAB 80C517-83C537-T40/85
T_A = -40 to 110°C for the SAB 80C517/83C537-T40/110

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Analog input voltage	V_{AINPUT}	V_{AGND} - 0.2	–	V_{AREF} + 0.2	V	9)
Analog input capacitance	C_I	–	25	60	pF	7)
Load time	t_L	–	–	2 t_{CY}	µs	7)
Sample time (incl. load time)	t_S	–	–	7 t_{CY}	µs	7)
Conversion time (incl. sample time)	t_C	–	–	13 t_{CY}	µs	7)
Differential non-linearity	DNLE	–	± 1/2	± 1	LSB	$V_{INTAREF}$ =
Integral non-linearity	INLE	–	± 1/2	± 1	LSB	V_{AREF} =
Offset error			± 1/2	± 1	LSB	$V_{INTAGND}$ =
Gain error			± 1/2	± 1	LSB	V_{AGND} = V_{SS}
Total unadjusted error	TUE		± 1	± 2	LSB	7)
Internal reference error	$V_{INTRERERR}$	–	–	± 30	mV	8)
V_{AREF} supply current	I_{REF}	–	–	5	mA	8)

See "Notes".

Notes

1) Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4, 5, and 6. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation.
In the worst case (capacity loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

2) Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address lines are stabilizing.

3) Power down I_{PD} is measured with all output pins disconnected;
$\overline{EA}$ = $\overline{RESET}$ = V_{CC}; Port 0 = Port 7 = Port 8 =
V_{CC}; XTAL1 = N.C.; XTAL2 = V_{SS};
$\overline{PE}$/SWD = OWE = V_{SS}

4) I_{CC} (active mode) is measured with all output pins disconnected; XTAL2 driven with clock signal according to the figure below; XTAL1 = N.C.;
$\overline{EA}$ = OWE = $\overline{PE}$/SWD = V_{CC}; Port 0 = Port 7 = Port 8 = V_{CC};
$\overline{RESET}$ = V_{SS}. I_{CC} would be slightly higher if a crystal oscillator is used.

5) I_{CC} (idle mode,) is measured with all output pins disconnected and all peripherals disabled;
XTAL2 driven with clock signal according to the figure below; XTAL1 = N.C.;
$\overline{RESET}$ = OWE = V_{CC}; Port 0 = Port 7 = Port 8 =

V_{CC}; $\overline{EA}$ = $\overline{PE}$/SWD = V_{SS}
I_{CC} (slow down mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with clock signal according to the figure below; XTAL = N.C.; Port 7 = Port 8 = V_{CC}; $\overline{EA}$ = $\overline{PE}$/SWD = V_{SS}.

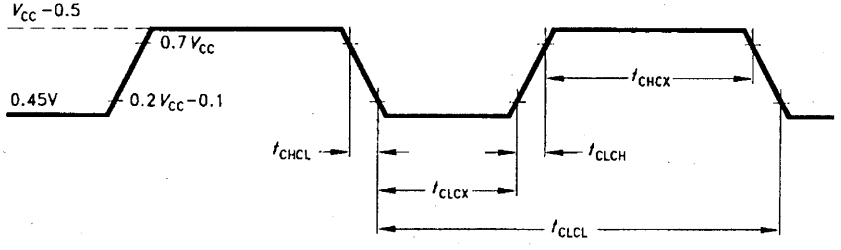
6) I_{CC} ($_{max}$) at other frequencies is given by:
active mode: $I_{CC\ max}$ = 3.1 x f_{OSC} + 3.0
idle mode: $I_{CC\ max}$ = 1.0 x f_{OSC} + 3.0
Where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V (see also notes 4 and 5)

7) The output impedance of the analog source must be low enough to assure full loading of the sample capacitance (C_I) during load time (T_L). After charging of the internal capacitance (C_I) in the load time (T_L) the analog input must be held constant for the rest of the sample time (T_S).

8) The differential impedance R_D of the analog reference voltage source must be less than 1 kΩ at reference supply voltage.

9) Exceeding the limit values at one or more input channels will cause additional current which is sinked sourced at these channels. This may also affect the accuracy of other channels which are operated within the specification.

Figure 109
Clock Signal Waveform for I cc Tests in Active, Idle Mode and Slow Down Mode



4

AC Characteristics for 12 and 16 MHz parts

V_{CC} = 5 V ± 10%; V_{SS} = 0 V
T_A = 0 to 70°C for SAB 80C517(-16)/83C537(-16)
T_A = - 40 to 85°C for SAB 80C517(-16)/83C537(-16)-T40/85
T_A = - 40 to 110°C for 80C517(-16)/83C537(-16)-T40/110

The AC Characteristics and Waveforms for both 12 and 16 MHz versions of the SAB 80C517/80C537 are identical to the SAB 80C515/80C535 except for the variable clock which varies from 1 MHz to 16 MHz in the SAB 80C517/80C537. Please refer to the SAB 80C515/80C535 for the AC Characteristics and Waveforms.

High-Performance 8-Bit CMOS Single-Chip Microcontroller

SAB 80C517A **Microcontroller for external ROM**
SAB 83C517A-5 **Microcontroller with factory mask-programmable ROM**

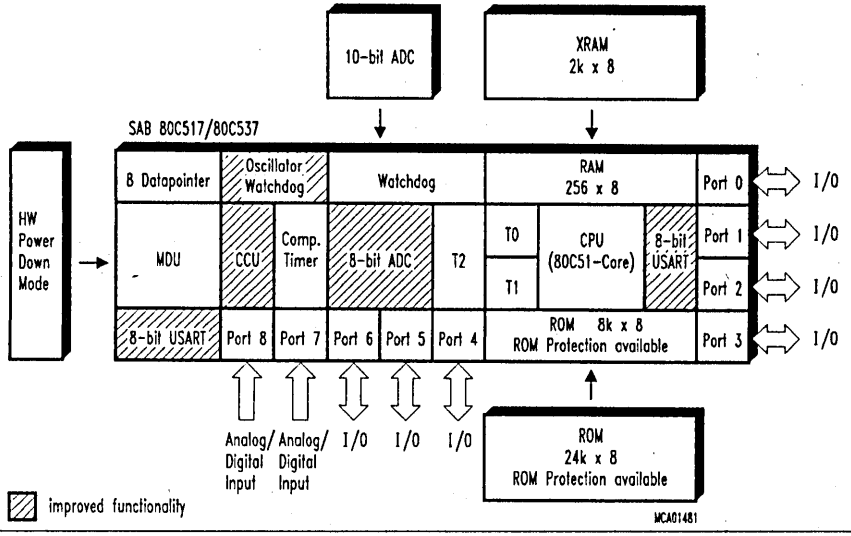
- SAB 80C517A / 83C517A-5, up to 18 MHz operation
- 32 K × 8 ROM (SAB 83C517A-5 only, ROM-Protection available)
- 256 × 8 on-chip RAM
- 2 K × 8 on-chip RAM (XRAM)
- Superset of SAB 80C51 architecture:
  - 1 µs instruction cycle time at 12 MHz
  - 666 ns instruction cycle time at 18 MHz
  - 256 directly addressable bits
  - Boolean processor
  - 64 Kbyte external data and program memory addressing
- Four 16-bit timer/counters
- Powerful 16-bit compare/capture unit (CCU) with up to 21 high-speed or PWM output channels and 5 capture inputs
- Versatile "fail-safe" provisions
- Fast 32-bit division, 16-bit multiplication, 32-bit normalize and shift by peripheral MUL/DIV unit (MDU)
- Eight data pointers for external memory addressing
- Seventeen interrupt vectors, four priority levels selectable
- 10-bit A/D converter with 12 multiplexed inputs
- Two full duplex serial interfaces with programmable Baudrate-Generators
- Fully upward compatible with SAB 80C515, SAB 80C517, SAB 80C515A
- Extended power saving mode
- Fast Power-On Reset
- Nine ports: 56 I/O lines, 12 input lines
- Three temperature ranges available:
  0 to 70 °C (T1)
  – 40 to 85 °C (T3)
  – 40 to 110 °C (T4)
- Plastic packages: P-LCC-84, P-MQFP 100

The SAB 80C517A/83C517A-5 is a high-end member of the Siemens SAB 8051 family of microcontrollers. It is designed in Siemens ACMOS technology and based on SAB 8051 architecture. ACMOS is a technology which combines high-speed and density characteristics with low-power consumption or dissipation.

While maintaining all the SAB 80C517 features and operating characteristics the SAB 80C517A is expanded in its "fail-safe" characteristics and timer capabilities. The SAB 80C517A is identical to the SAB 83C517A-5 except that it lacks the on-chip program memory. The SAB 80C517A / 83C517A-5 is supplied in a 84-pin plastic leaded chip carrier package (P-LCC-84) and in a 100-pin plastic quad flat package (P-MQFP-100).

4

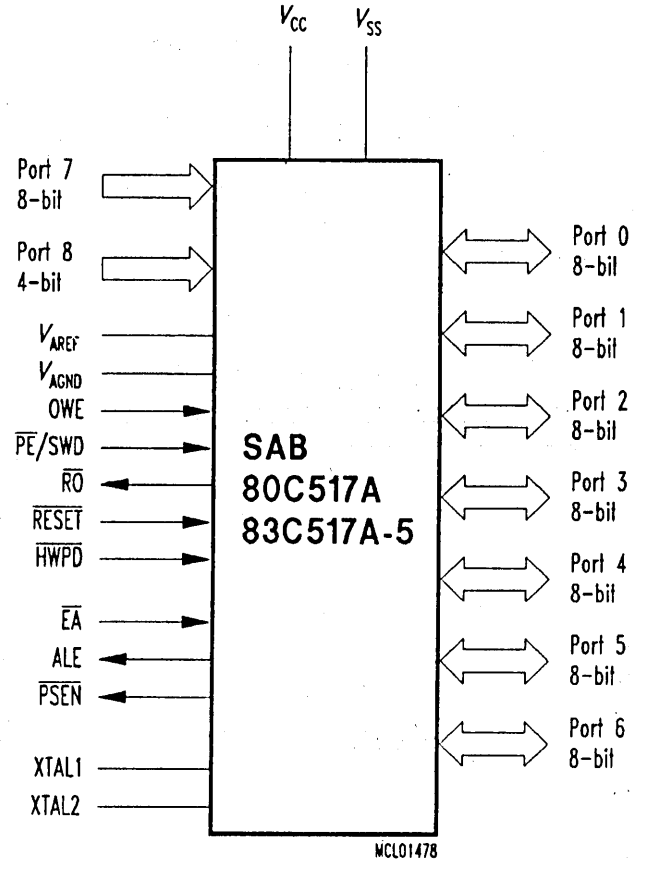
Figure 110
SAB 80C517A / 83C517A-5



Ordering Information

Type	Package	Description (8-bit single- microcontroller)
SAB 80C517A-N18	PL-CC-84	for external memory, 18 MHz
SAB 80C517A-S18	P-MQFP-100	
SAB 83C517A-5N18	PL-CC-84	with mask-programmable ROM, 18 MHz
SAB 83C517A-5S18	P-MQFP-100	
SAB 80C517A-N18-T3	PL-CC-84	for external memory, 18 MHz ext.temperature –40 to 85 °C
SAB 80C517A-S18-T3	P-MQFP-100	
SAB 83C517A-5N18-T3	PL-CC-84	with mask-programmable ROM, 18 MHz ext temperature –40 to 85 °C
SAB 83C517A-5S18-T3	P-MQFP-100	
SAB 80C517A-N18-T4	PL-CC-84	for external memory, 18 MHz ext.temperature –40 to 110 °C
SAB 80C517A-S18-T4	P-MQFP-100	
SAB 83C517A-5N18-T4	PL-CC-44	with mask-programmable ROM, 18 MHz ext temperature –40 to 110 °C
SAB 83C517A-5S18-T4	P-MQFP-100	

Figure 111
Logic Symbol



4

The pin functions of the SAB 80C517A are identical to those of the SAB 80C517 / 80C537 with one exception:

Typ	Symbol	PL-CC-84	P-QFP-100
SAB 80C517A	$\overline{\text{HWPD}}$	Pin 60	Pin 72
SAB 80C517 / 80C537	V_{SS}		

Figure 112
Pin Configuration (PL-CC-84)

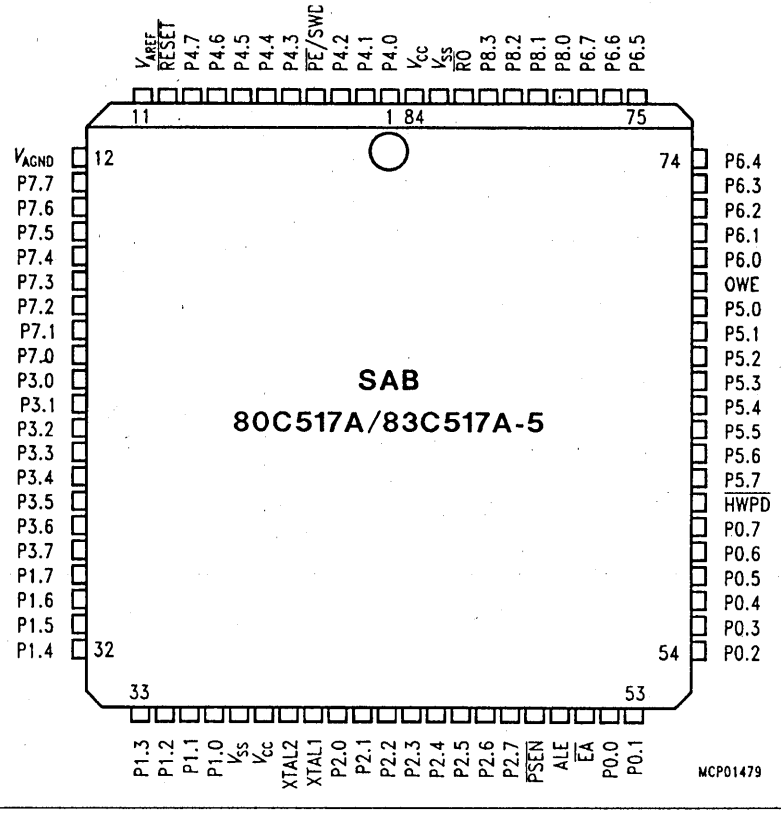
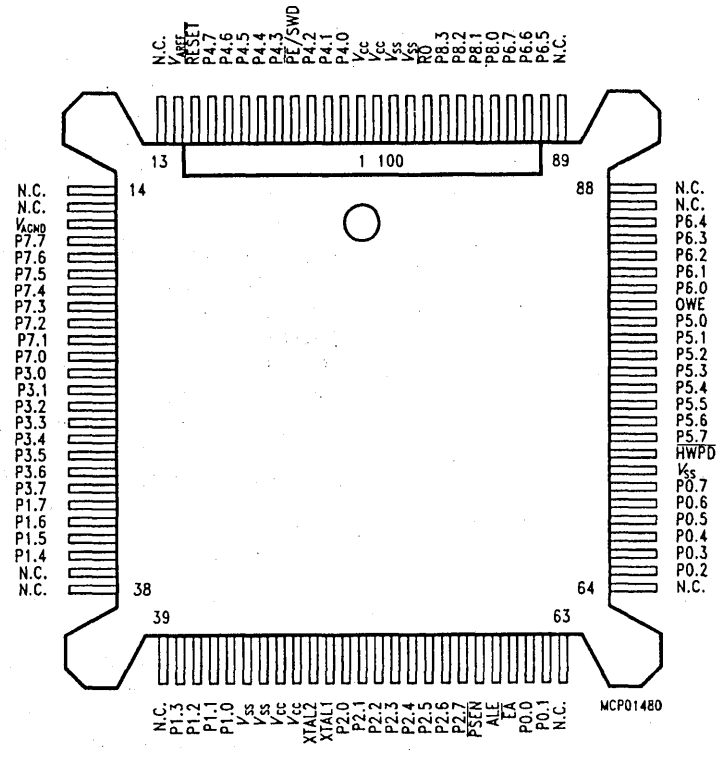


Figure 113
Pin Configuration (P-QFP-100)



Pin Definitions and Functions

Symbol	Pin Number		I/O *)	Function
	PL-CC-84	P-QFP-100		
P4.0 – P4.7	1– 3, 5 – 9	2 – 4, 6 – 10	I/O	**Port 4** is a bidirectional I/O port with internal pull-up resistors. Port 4 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. This port also serves alternate compare functions. The secondary functions are assigned to the pins of port 4 as follows: – CM0 (P4.0): Compare Channel 0 – CM1 (P4.1): Compare Channel 1 – CM2 (P4.2): Compare Channel 2 – CM3 (P4.3): Compare Channel 3 – CM4 (P4.4): Compare Channel 4 – CM5 (P4.5): Compare Channel 5 – CM6 (P4.6): Compare Channel 6 – CM7 (P4.7): Compare Channel 7
$\overline{PE}$ / SWD	4	5	I	**Power saving modes $\overline{enable}$/Start Watchdog Timer** A low level on this pin allows the software to enter the power down, idle and slow down mode. In case the low level is also seen during reset, the watchdog timer function is off on default. Use of the software controlled power saving modes is blocked, when this pin is held on high level. A high level during reset performs an automatic start of the watchdog timer immediately after reset. When left unconnected this pin is pulled high by a weak internal pull-up resistor.
$\overline{RESET}$	10	11	I	**$\overline{RESET}$** A low level on this pin for the duration of one machine cycle while the oscillator is running resets the SAB 80C517A. A small internal pull-up resistor permits power-on reset using only a capacitor connected to V_{SS}.

*) I = Input
O = Output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O *)	Function
	PL-CC-84	P-QFP-100		
V_{AREF}	11	12		**Reference voltage** for the A/D converter.
V_{AGND}	12	16		**Reference ground** for the A/D converter.
P7.7 – P7.0	13 – 20	17 – 24	I	**Port 7** is an 8-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages, and for the lower 8-bit of the multiplexed analog inputs of the A/D converter, simultaneously.
P3.0 – P3.7	21 – 28	25 – 32	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – R = D0 (P3.0): receiver data input (asynchronous) or data input/ output (synchronous) of serial interface – T = D0 (P3.1): transmitter data output (asynchronous) or clock output (synchronous) of serial interface 0 – $\overline{INT0}$ (P3.2): interrupt 0 input/timer 0 gate control

*) I = Input
O = Output

4

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O	Function
	PL-CC-84	P-QFP-100		
P3.0 – P3.7 (cont'd.)	21 – 28	25 – 32	I/O	– $\overline{INT1}$ (P3.3): interrupt 1 $\overline{input}$/timer 1 gate control – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – $\overline{WR}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory – $\overline{RD}$ (P3.7): the read control signal enables the external data memory to port 0
P1.7 – P1.0	29 – 36	33 – 36, 40 – 43	I/O	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. It is used for the low order address byte during program verification. It also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: – $\overline{INT3}$/CC0 (P1.0): interrupt 3 $\overline{input}$ / compare 0 output / capture 0 input – INT4/CC1 (P1.1): interrupt 4 input compare 1 output / capture 1 input – INT5/CC2 (P1.2): interrupt 5 input compare 2 output / capture 2 input

*) I = Input
O = Output

Definitions and Functions (cont'd)

Symbol	Pin Number		I/O	Function
	PL-CC-84	P-QFP-100		
P1.7–P1.0 (cont'd)	29-36	33-36, 40-43	I/O	– INT6/CC3 (P1.3): interrupt 6 input /compare 3 output /capture 3 input – $\overline{\text{INT2}}$/CC4 (P1.4): interrupt 2 input /compare 4 output /capture 4 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input
XTAL2	39	48	–	**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL1	40	49	–	**XTAL1** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
P2.0 – P2.7	41 – 48	50 – 57	I/O	**Port 2** is a bidirectional I/O port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pull-up resistors when issuing1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.

*) I = Input
O = Output

4

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O	Function
	PL-CC-84	P-QFP-100		
$\overline{PSEN}$	49	58	O	The **Program Store Enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. Remains high during internal program execution.
ALE	50	59	O	The **Address Latch Enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	51	60	I	**External Access Enable** When held at high level, instructions are fetched from the internal ROM (SAB 83C517A-5 only) when the PC is less than 8000H. When held at low level, the SAB 80C517A fetches all instructions from external program memory. For the SAB 80C517A this pin must be tied low.
P0.0 – P0.7	52– 59	61 – 62, 65 – 70	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-up resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB 83C517A if ROM-Protection was not enabled. External pull-up resistors are required during program verification.

*) I = Input
O = Output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O	Function
	PL-CC-84	P-QFP-100		
$\overline{\text{HWPD}}$	60	72	I	**$\overline{\text{Hardware Power Down}}$** A low level on this pin for the duration of one machine cycle while the oscillator is running resets the SAB 80C517A. A low level for a longer period will force the part to Power Down Mode with the pins floating (see Table 51).
P5.7 – P5.0	61 – 68	73 – 80	I/O I	**Port 5** is a bidirectional I/O port with internal pull-up resistors. Port 5 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. This port also serves the alternate function "Concurrent Compare" and "Set/Reset Compare". The secondary functions are assigned to the port 5 pins as follows: – CCM0 to CCM7 (P5.0 to P5.7): concurrent compare or Set/Reset
OWE	69	81	I/O	**Oscillator Watchdog Enable** A high level on this pin enables the oscillator watchdog. When left unconnected this pin is pulled high by a weak internal pull-up resistor. When held at low level the oscillator watchdog function is off.
P6.0 – P6.7	70 – 77	82 – 86, 90 – 92	I/O	**Port 6** is a bidirectional I/O port with internal pull-up resistors. Port 6 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, port 6 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. Port 6 also contains the external A/D converter control pin and the transmit and receive pins for serial channel 1.

*) I = Input
O = Output

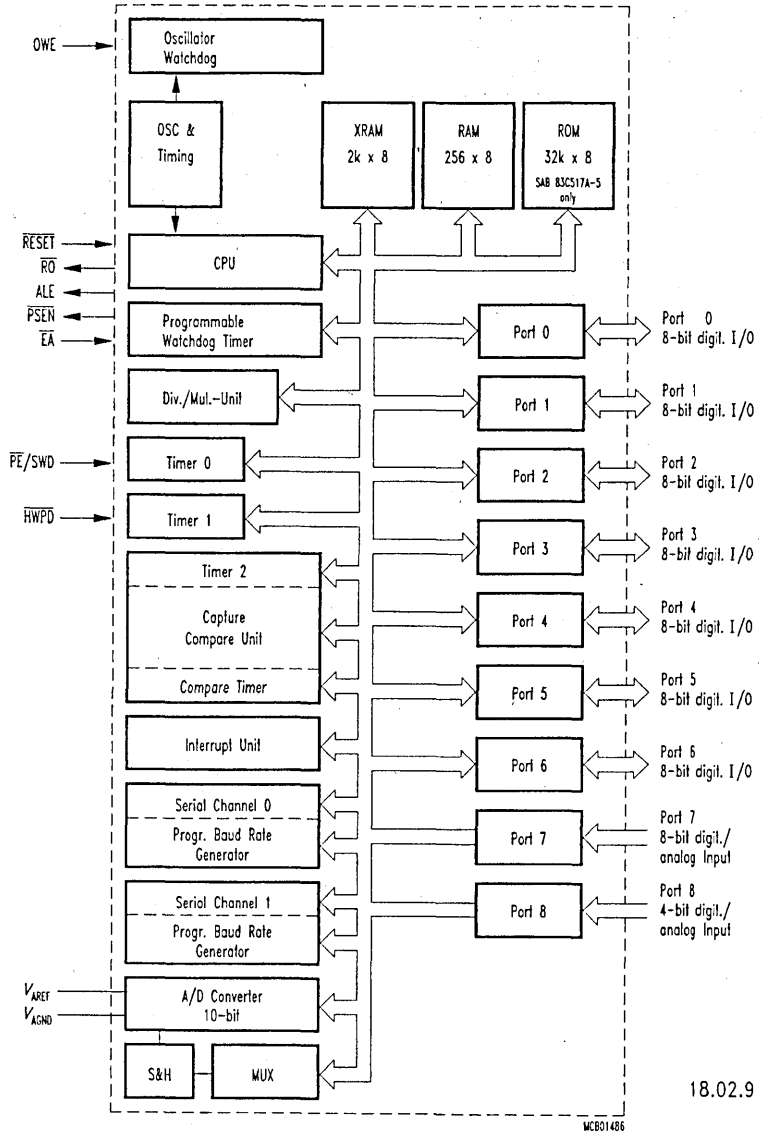
4

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O	Function
	PL-CC-84	P-QFP-100		
P6.0 – P6.7	70 – 77	82 – 86, 90 – 92	I/O	The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 6, as follows: – $\overline{ADST}$(P6.0): external A/D converter start pin – R × D1 (P6.1): receiver data input of serial interface 1 – T × D1 (P6.2): transmitter data output of serial interface 1
P8.0 – P8.3	78 – 81	93 – 96	I	**Port 8** is a 4-bit unidirectional input port. Port pins can be used for digital input, if voltage levels meet the specified input high/low voltages, and for the higher 4-bit of the multiplexed analog inputs of the A/D converter, simultaneously.
$\overline{RO}$	82	97	O	**$\overline{Reset\ Output}$** This pin outputs the internally synchronized reset request signal. This signal may be generated by an external hardware reset, a watchdog timer reset or an oscillator watchdog reset. The reset output is active low.
V_{SS}	37, 83	44, 45, 71,	–	**Circuit ground potential**
V_{CC}	38, 84	1, 46, 47,	–	**Supply terminal** for all operating modes

*) I = Input
O = Output

Figure 114
Block Diagram



Functional Description

The SAB 80C517A is based on 8051 architecture. It is a fully compatible member of the Siemens SAB 8051/80C51 microcontroller family being a significantly enhanced SAB 80C517. The SAB 80C517A is therefore compatible with code written for the SAB 80C517.

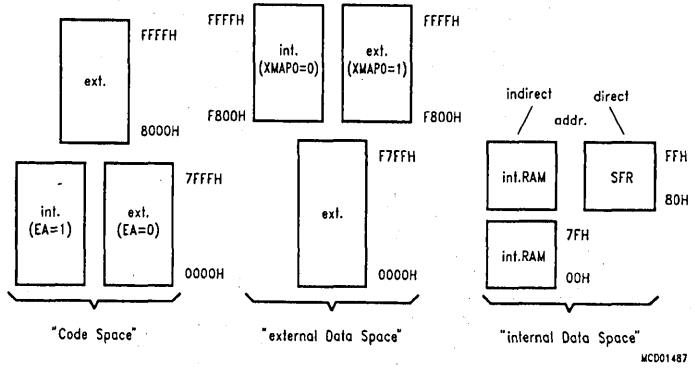
Having an 8-bit CPU with extensive facilities for bit-handling and binary BCD arithmetics the SAB 80C517A is optimized for control applications. With an 18 MHz crystal, 58% of the instructions are executed in 666.67 ns.

Being designed to close the performance gap to the 16-bit microcontroller world, the SAB 80C517A's CPU is supported by a powerful 32-/16-bit arithmetic unit and a more flexible addressing of external memory by eight 16-bit datapointers.

Memory Organization

According to the SAB 8051 architecture, the SAB 80C517A has separate address spaces for program and data memory. Figure 115 Memory Map illustrates the mapping of address spaces.

Figure 115
Memory Map



– **Program Memory** ('Code Space')

The SAB 83C517A-5 has 32 Kbyte of on-chip ROM, while the SAB 80C517A has no internal ROM. The program memory can externally be expanded up to 64 Kbyte. Pin $\overline{EA}$ controls whether program fetches below address 8000H are done from internal or external memory.

As a new feature the SAB 83C517A-5 offers the possibility of protecting the internal ROM against unauthorized access. This protection is implemented in the ROM-Mask. Therefore, the decision ROM-Protection 'yes' or 'no' has to be made when delivering the ROM-Code. Once enabled, there is no way of disabling the ROM-Protection.

Effect: The access to internal ROM done by an externally fetched MOVC instruction is disabled. Nevertheless, an access from internal ROM to external ROM is possible.

To verify the read protected ROM-Code a special ROM-Verify-Mode is implemented. This mode also can be used to verify unprotected internal ROM.

ROM-Protection	**ROM-Verification Mode (see 'AC Characteristics')**	**Restrictions**
no	ROM-Verification Mode 1 (standard 8051 Verification Mode) ROM-Verification Mode 2	–
yes	ROM-Verification Mode 2	– standard 8051 Verification Mode is disabled – externally applied MOVC accessing to internal ROM is disabled

4

– **Data Memory** ('Code Space')

The data memory space consists of an internal and an external memory space. The SAB 80C517A contains another 2 Kbyte on On-Chip RAM above the 256-bytes internal RAM of the base type SAB 80C517. This RAM is called XRAM in this document.

– **External Data Memory**

Up to 64 Kbyte external data memory can be addressed by instructions that use 8-bit or 16-bit indirect addressing. For 8-bit addressing MOVX instructions in combination with registers R0 and R1 can be used. A 16-bit external memory addressing is supported by eight 16-bit datapointers. Registers XPAGE and SYSCON control whether data fetches at addresses F800H to FFFFH are done from internal XRAM or from external data memory.

– **Internal Data Memory**

The internal data memory is divided into four physically distinct blocks:

- the lower 128 bytes of RAM including four banks containing eight registers each
- the upper 128 byte of RAM
- the 128 byte special function register area.
- a 2 K x 8 area which is accessed like external RAM (MOVX-instructions), implemented on chip at the address range from F800H to FFFFH. Special Function Register SYSCON controls whether data is read or written (to) XRAM or external RAM.

A mapping of the internal data memory is also shown in Figure 115 Memory Map . The overlapping address spaces are accessed by different addressing modes (see User's Manual SAB 80C517). The stack can be located anywhere in the internal data memory.

Architecture for the XRAM

The contents of the XRAM are not affected by a reset or HW Power Down. After power-up the contents are undefined, while remaining unchanged during and after a reset or HW Power Down if the power supply is not turned off.

The additional On-Chip RAM is logically located in the "external data memory" range at the upper end of the 64 Kbyte address range (F800H-FFFFH). It is possible to enable and disable (only by reset) the XRAM. If it is disabled the device shows the same behavior as the parts without XRAM, i.e. all MOVX accesses use the external bus to physically external data memory.

Accesses to XRAM

Because the XRAM is used in the same way as external data memory the same instruction types must be used for accessing the XRAM.

Note: If a reset occurs during a write operation to XRAM, the effect on XRAM depends on the cycle which the reset is detected at (MOVX is a 2-cycle instruction):

Reset detection at cycle 1:
The new value will not be written to XRAM. The old value is not affected.

Reset detection at cycle 2:
The old value in XRAM is overwritten by the new value.

– Accesses to XRAM using the DPTR

There are a Read and a Write instruction from and to XRAM which use one of the 16-bit DPTR for indirect addressing. The instructions are:

MOVX A, @DPTR (Read)
MOVX @DPTR, A (Write)

Normally the use of these instructions would use a physically external memory. However, in the SAB 80C517A the XRAM is accessed if it is enabled and if the DPTR points to the XRAM address space (DPTR ≥ F800H).

– Accesses to XRAM using the Registers R0/R1

The 8051 architecture also provides instructions for accesses to external data memory range which use only an 8-bit address (indirect addressing with registers R0 or R1). The instructions are:

MOVX A, @Ri (Read)
MOVX @Ri, A (Write)

In application systems, either a real 8-bit bus (with 8-bit address) is used or Port 2 serves as page register which selects pages of 256-byte. However, the distinction, whether Port 2 is used as general purpose I/O or as "page address" is made by the external system design. From the device's point of view it cannot be decided whether the Port 2 data is used externally as address or as I/O data!

Hence, a special page register is implemented into the SAB 80C517A to provide the possibility of accessing the XRAM also with the MOVX @Ri instructions, i.e. XPAGE serves the same function for the XRAM as Port 2 for external data memory.

Figure 116 Special Function Register XPAGE

The reset value of XPAGE is 00H.
XPAGE can be set and read by software.

The register XPAGE provides the upper address byte for accesses to XRAM with MOVX @Ri instructions. If the address formed from XPAGE and Ri is less than the XRAM address range, then an external access is performed. For the SAB 80C517A the contents of XPAGE must be greater or equal than F8H in order to use the XRAM. Of course, the XRAM must be enabled if it shall be used with MOVX @Ri instructions.

Thus, the register XPAGE is used for addressing of the XRAM; additionally its contents are used for generating the internal XRAM select. If the contents of XPAGE are less than the XRAM address range then an external bus access is performed where the upper address byte is provided by P2 and not by XPAGE!

Therefore, the software has to distinguish two cases, if the MOVX @Ri instructions with paging shall be used:

a) Access to XRAM:
The upper address byte must be written to XPAGE or P2; both writes selects the XRAM address range.

b) Access to external memory:
The upper address byte must be written to P2; XPAGE will be loaded with the same address in order to deselect the XRAM.

Figure 117 Special Function Register SYSCON

Addr. 0B1H	–	–	–	–	–	–	XMAP1	XMAP0	SYSCON

Bit	Function
XMAP0	Global enable/disable bit for XRAM memory. XMAP0 = 0: The access to XRAM (= On-Chip XDATA memory) is enabled. XMAP0 = 1: The access to XRAM is disabled. All MOVX accesses are performed by the external bus (reset state).
XMAP1	Control bit for $\overline{RD}$ /$\overline{WR}$ signals during accesses to XRAM; this bit has no effect if XRAM is disabled (XMAP0 = 1) or if addresses exceeding the XRAM address range are used for MOVX accesses. XMAP1 = 0: The signals $\overline{RD}$ and $\overline{WR}$ are not activated during accesses to XRAM. XMAP1 = 1: The signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM.

4

Control of XRAM in the SAB 80C517A

There are two control bits in register SYSCON which control the use and the bus operation during accesses to the additional On-Chip RAM (XRAM).

Reset value of SYSCON is xxxx xx01B.

The control bit XMAP0 is a global enable/ disable bit for the additional On-Chip RAM (XRAM). If this bit is set, the XRAM is disabled, all MOVX accesses use external memory via the external bus. In this case the SAB 80C517A does not use the additional On-Chip RAM and is compatible with the types without XRAM.

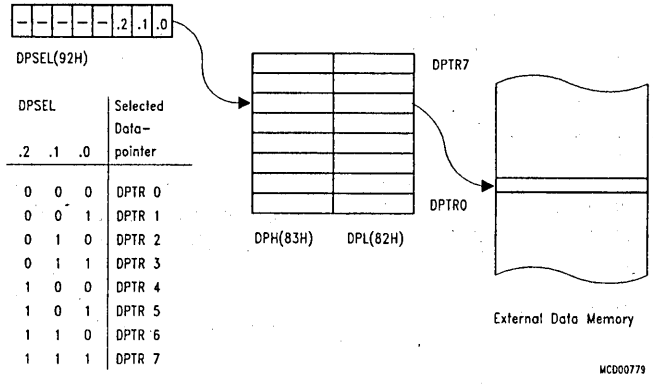
XMAP0 is hardware protected by an unsymmetric latch. An unintentional disabling of XRAM could be dangerous since indeterminate values would be read from the external bus. To avoid this the XMAP-bit is forced to '1' only by reset. Additionally, during reset an internal capacitor is loaded. So after reset state XRAM is disabled. Because of the load time of the capacitor XMAP0-bit once written to '0' (that is, discharging capacitor) cannot be set to '1' again by software. On the other hand any distortion (software hang up noise, ...) is not able to load this capacitor, too. That is, the stable status is XRAM enabled. The only way to disable XRAM after it was enabled is a reset.

The clear instruction for XMAP0 should be integrated in the program initialization routine before XRAM is used. In extremely noisy systems the user may have redundant clear instructions.

Multiple Datapointers

As a functional enhancement to standard 8051 controllers, the SAB 80C517A contains eight 16-bit datapointers. The instruction set uses just one of these datapointers at a time. The selection of the actual datapointer is done in special function register DPSEL (data pointer select, addr. 92H). Figure 118 Addressing of External Data Memory illustrates the addressing mechanism.

Figure 118
Addressing of External Data Memory



Special Function Registers

All registers, except the program counter and the four general purpose register banks, reside in the special function register area. The 81 special function registers include arithmetic registers, pointers, and registers that provide an interface between the CPU and the on-chip peripherals. There are also 128 directly addressable bits within the SFR area. All special function registers are listed in Table 45 and Table 46.

In Table 45 they are organized in numeric order of their addresses. In Table 46 they are organized in groups which refer to the functional blocks of the SAB 80C517A.

Table 45
Special Function Register

Address	Register	Contents after Reset	Address	Register	Contents after Reset
80H	**P0** [1]	**0FFH**	**A0H**	**P2** [1]	**0FFH**
81H	SP	07H	A1H	COMSETL	00H
82H	DPL	00H	A2H	COMSETH	00H
83H	DPH	00H	A3H	COMCLRL	00H
84H	(WDTL)		A4H	COMCLRH	00H
85H	(WDTH)		A5H	SETMSK	00H
86H	WDTREL	00H	A6H	CLRMSK	00H
87H	PCON	00H	A7H		
88H	**TCON** [1]	**00H**	**A8H**	**IEN0** [1]	**00H**
89H	TMOD	00H	A9H	IP0	00H
8AH	TL0	00H	AAH	S0RELL	0D9H
8BH	TL1	00H	ABH		
8CH	TH0	00H	ACH		
8DH	TH1	00H	ADH		
8EH			AEH		
8FH			AFH		
90H	**P1** [1]	**0FFH**	**B0H**	**P3** [1]	**0FFH**
91H	XPAGE	00H	B1H	SYSCON	XXXX XX01B
92H	DPSEL	XXXXX000B	B2H		
93H			B3H		
94H			B4H		
95H			B5H		
96H			B6H		
97H			B7H		

[1] Bit-addressable Special Function Register

Table 45
Special Function Register (cont'd)

Address	Register	Contents after Reset	Address	Register	Contents after Reset
98H	**S0CON** [1]	**00H**	**B8H**	**IEN1** [1]	**00H**
99H	S0BUF	0XXH	B9H	IP1	XX00 0000B
9AH	IEN2	XX00 00X0B	BAH	S0RELH	XXXX XX11B
9BH	S1CON	0X00 0000B	BBH	S1RELH	XXXX XX11B
9CH	S1BUF	0XXH	BCH		
9DH	S1RELL	00H	BDH		
9EH			BEH		
9FH			BFH		
C0H	**IRCON0** [1]	**00H**	**E0H**	**ACC** [1]	**00H**
C1H	CCEN	00H	E1H	CTCON	0X00 0000B
C2H	CCL1	00H	E2H	CML3	00H
C3H	CCH1	00H	E3H	CMH3	00H
C4H	CCL2	00H	E4H	CML4	00H
C5H	CCH2	00H	E5H	CMH4	00H
C6H	CCL3	00H	E6H	CML5	00H
C7H	CCH3	00H	E7H	CMH5	00H
C8H	**T2CON** [1]	**00H**	**E8H**	**P4** [1]	**0FFH**
C9H	CC4EN	00H	E9H	MD0	XXH
CAH	CRCL	00H	EAH	MD1	XXH
CBH	CRCH	00H	EBH	MD2	XXH
CCH	TL2	00H	ECH	MD3	XXH
CDH	TH2	00H	EDH	MD4	XXH
CEH	CCL4	00H	EEH	MD5	XXH
CFH	CCH4	00H	EFH	ARCON	0XXX XXXXB
D0H	**PSW** [1]	**00H**	**F0H**	**B** [1]	**00H**
D1H	IRCON1	00H	F1H		
D2H	CML0	00H	F2H	CML6	00H
D3H	CMH0	00H	F3H	CMH6	00H
D4H	CML1	00H	F4H	CML7	00H
D5H	CMH1	00H	F5H	CMH7	00H
D6H	CML2	00H	F6H	CMEN	00H
D7H	CMH2	00H	F7H	CMSEL	00H
D8H	**ADCON0** [1]	**00H**	**F8H**	**P5** [1]	**0FFH**
D9H	ADDATH	00H	F9H		
DAH	ADDATL	00H	FAH	P6	0FFH
DBH	P7	XXH	FBH		
DCH	ADCON1	XXXX 0000B	FCH		
DDH	P8	XXH	FDH	(IS0)	
DEH	CTRELL	00H	FEH	(IS1)	
DFH	CTRELH	00H	FFH	(EMREG)	

[1] Bit-addressable Special Function Register

Table 46
Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
Ports	P0	Port 0	80H [1]	0FFH
	P1	Port 1	90H [1]	0FFH
	P2	Port 2	0A0H [1]	0FFH
	P3	Port 3	0B0H [1]	0FFH
	P4	Port 4	0E8H [1]	0FFH
	P5	Port 5	0F8H [1]	0FFH
	P6	Port 6	0FAH	0FFH
	P7	Port 7, Analog/Digital Input	0DBH	
	P8	Port 8, Analog/Digital Input, 4-bit	0DDH	
Pow.Sav. Modes	PCON	Power Control Register	87H	00H
Serial Channels	ADCON0 [2]	A/D Converter Control Reg.	0D8H [1]	00H
	PCON [2]	Power Control Register	87H	00H
	S0BUF	Serial Channel 0 Buffer Reg.	99H	0XXH
	S0CON	Serial Channel 0 Control Reg.	98H [1]	00H
	S0RELL	Serial Channel 0 Reload Reg., low byte	AAH	D9H
	S0RELH	Serial Channel 0 Reload Reg., high byte	BAH	XXXX.XX11B [3]
	S1BUF	Serial Channel 1 Buffer Reg.	9CH	0XXH [3]
	S1CON	Serial Channel 1 Control Reg.	9BH	0X00 0000B [3]
	S1RELL	Serial Channel 1 Reload Reg., low byte	9DH	00H
	S1RELH	Serial Channel 1 Reload Reg., high byte	BBH	XXXX.XX11B [3]
Timer 0/ Timer 1	TCON	Timer Control Register	88H [1]	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Watchdog	IEN0 [2]	Interrupt Enable Register 0	0A8H [1]	00H
	IEN1 [2]	Interrupt Enable Register 1	0B8H [1]	00H
	IP0 [2]	Interrupt Priority Register 0	0A9H	00H
	IP1 [2]	Interrupt Priority Register 1	0B9H	XX00 0000B [3]
	WDTREL	Watchdog Timer Reload Reg.	86H	00H

[1] Bit-addressable special function registers
[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.
[3] X means that the value is indeterminate

A/D Converter

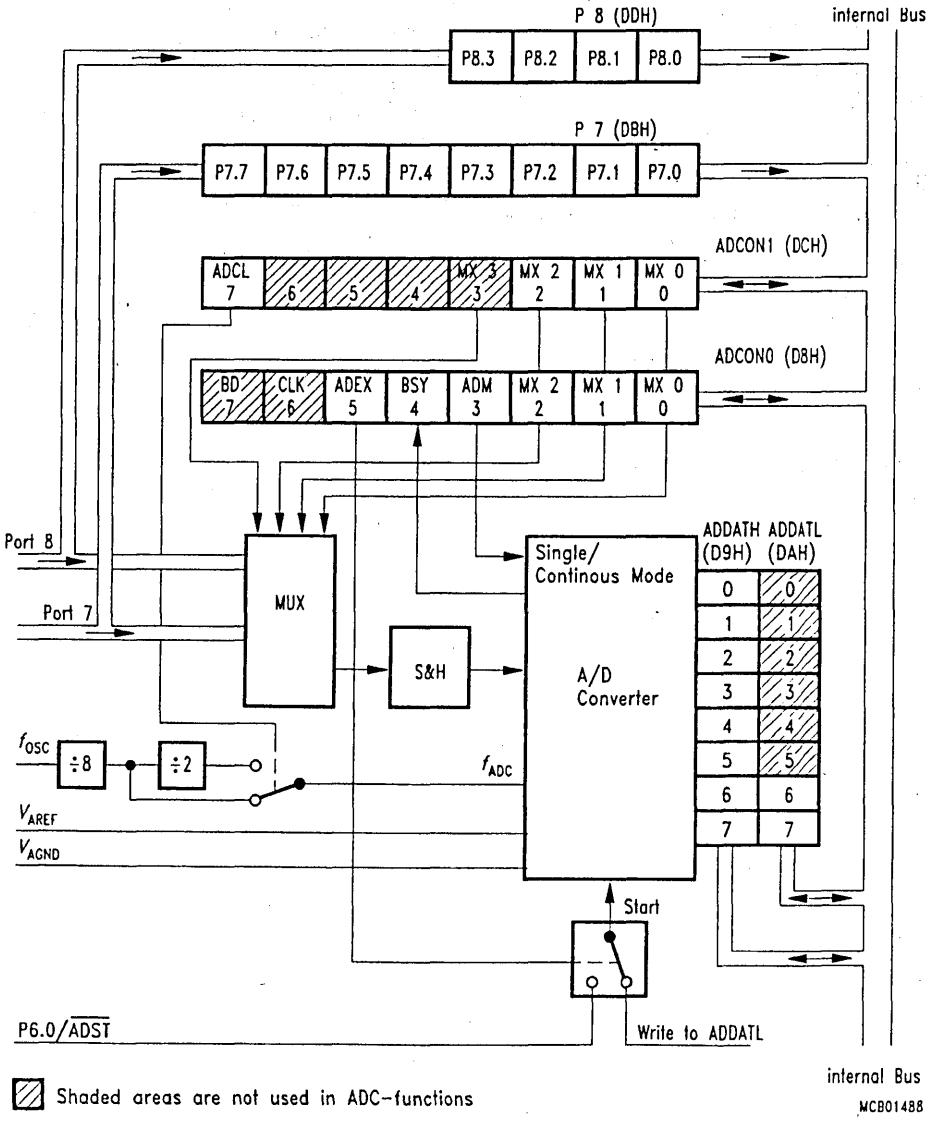
In the SAB 80C517A a new high performance / high-speed 12-channel 10-bit A/D-Converter is implemented. Its successive approximation technique provides 7 µs conversion time (f_{OSC} = 16 MHz). The conversion principle is upward compatible to the one used in the SAB 80C517. The main functional blocks are shown in Figure 119 Block Diagram A/D Converter .

The comparator is a fully differential comparator for a high power supply rejection ratio and very low offset voltages. The capacitor network is binary weighted providing genuine 10-bit resolution.

The table below shows the sample time T_S and the conversion time T_C, which are dependent on f_{OSC} and a new prescaler (see also Bit ADCL in SFR ADCON 1).

f_{OSC} [MHz]	prescaler	f_{ADC} [MHz]	Sample Time (incl. load time) T_S [µs]	Conv. Time (incl. sample time) T_C [µs]
12	÷ 8	1.5	2.67	9.33
	÷ 16	0.75	5.33	18.66
16	÷ 8	2.0	2.0	7.0
	÷ 16	1.0	4.0	14.0
18	÷ 8	–	–	–
	÷ 16	1.125	3.55	12.4

Figure 119
Block Diagram A/D Converter



Compare/Capture Unit (CCU)

The compare/capture unit is a complex timer/ register array for applications that require high speed I/O, pulse width modulation and more timer/counter capabilities.

The CCU contains

- one 16-bit timer/counter (**timer 2**) with 2-bit prescaler, reload capability and a max. clock frequency of $f_{OSC/12}$ (1 MHz with a 12 MHz crystal).
- one 16-bit timer (**compare timer**) with 8-bit prescaler, reload capability and a max. clock frequency of $f_{OSC/2}$ (6 MHz with a 12 MHz crystal).
- fifteen 16-bit compare registers.
- five of which can be used as 16-bit capture registers.
- up to 21 output lines controlled by the CCU.
- nine interrupts which can be generated by CCU-events.

Figure 120 Block Diagram of the Compare/ Capture Unit shows a block diagram of the CCU. Eight compare registers (CM0 to CM7) can individually be assigned to either timer 2 or the compare timer. Diagrams of the two timers are shown in Figures 121 and 122. The four compare/capture registers, the compare/reload/ capture register and the comset/comclr register are always connected to timer 2. Depending on the register type and the assigned timer three different compare modes can be selected.

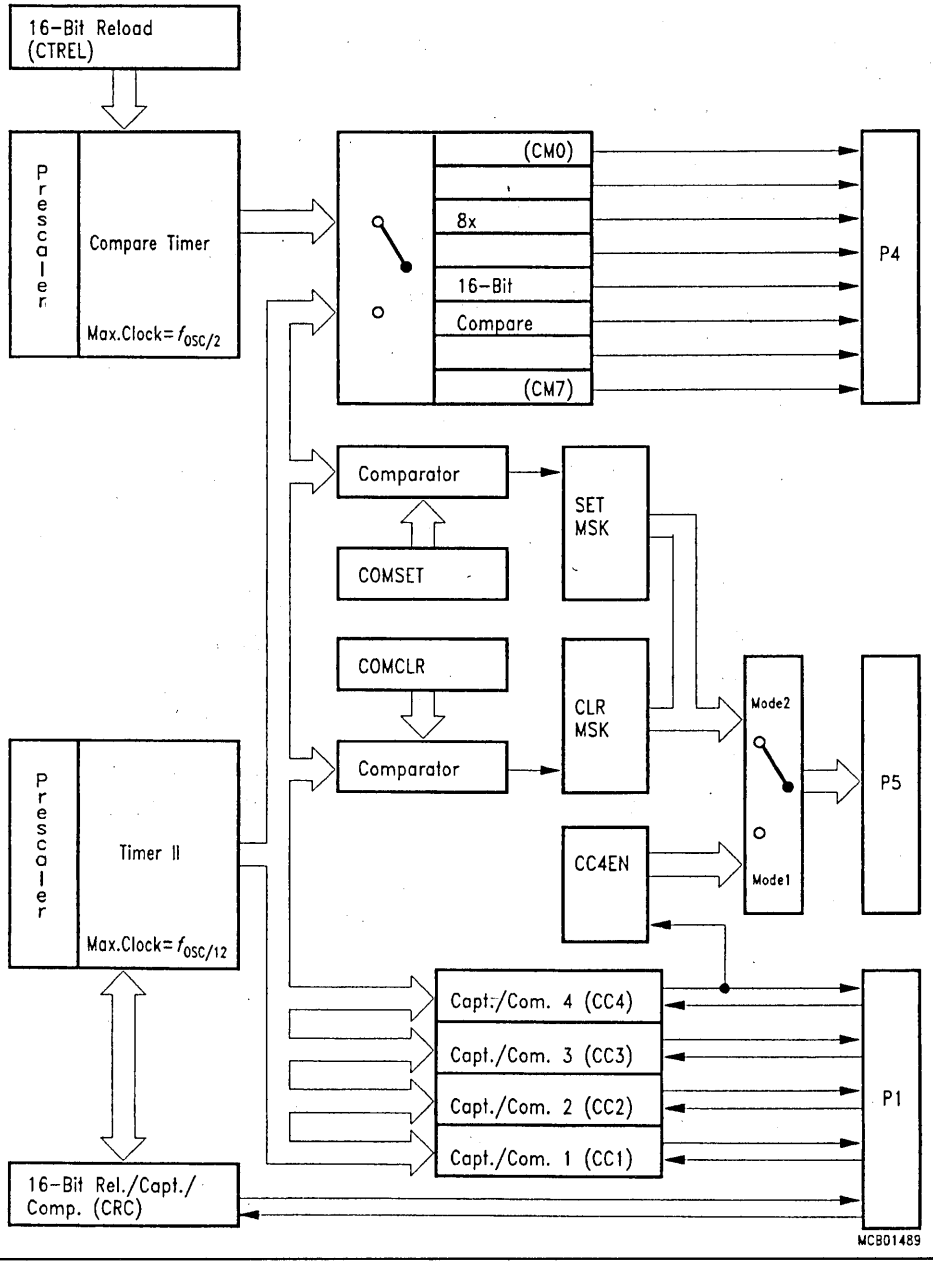
Table 47 illustrates possible combinations and the corresponding output lines.

Table 47
CCU Compare Configuration

Assigned Timer	Compare Register	Compare Output at	Possible Modes
Timer 2	CRCH/CRCL	P1.0/INT3/CC0	Comp. mode 0, 1 + Reload
	CC1H/CC1L	P1.1/INT4/CC1	Comp. mode 0, 1
	CC2H/CC2L	P1.2/INT5/CC2	Comp. mode 0, 1
	CC3H/CC3L	P1.3/$\overline{\text{INT6}}$/CC3	Comp. mode 0, 1
	CC4H/CC4L	P1.4/$\overline{\text{INT2}}$/CC4	Comp. mode 0, 1
	CC4H/CC4L	P5.0/CCM0	Comp. mode 1
	:	:	:
	CC4H/CC4L	P5.7/CCM7	Comp. mode 1
	COMSETL/COMSETH	P5.0/CCM0	Comp. mode 2
		:	:
		P5.7/CCM7	Comp. mode 2
	COMCLRL/COMCLRH	P5.0/CCM0	Comp. mode 2
		:	:
		P5.7/CCM7	Comp. mode 2
	CM0H/CM0L	P4.0/CM0	Comp. mode 1
	:	:	:
	CM7H/CM7L	P4.7/CM7	Comp. mode 1
Compare timer	CM0H/CM0L	P4.0/CM0	Comp. mode 0 (with shadow latches)
	:	:	:
	CM7H/CM7L	P4.7/CM7	Comp. mode 0 (with shadow latches)

4

Figure 120
Block Diagram of the Compare/Capture Unit



Compare

In compare mode, the 16-bit values stored in the dedicated compare registers are compared to the contents of the timer 2 register or the compare timer register. If the count value in the timer registers matches one of the stored value, an appropriate output signal is generated at the corresponding pin(s) and an interrupt is requested. Three compare modes are provided:

Mode 0: Upon a match the output signal changes from low to high. It returns to low level at timer overflow.

Mode 1: The transition of the output signal can be determined by software.
A timer overflow signal does not affect the compare-output.

Mode 2: In compare mode 2 the concurrent compare output pins on Port 5 are used as follows (see Figure 124 Compare-Mode 2 (Port 5 only))
– When a compare match occurs with register COMSET, a high level appears at the pins of port 5 whose corresponding bits in the mask register SETMSK (address 0A5H) are set.
– When a compare match occurs in register COMCLR, a low level appears at the pins of port 5 whose corresponding bits in the mask register CLRMSK (address 0A6H) are set.
Additionally the Port 5 pins used for compare mode 2 may also be directly written to by write instructions to SFR P5. Of course, the pins can also be read under program control.

Compare registers CM0 to CM7 use additional compare latches when operated in mode 0. Figure 123 Compare-Mode 0 with Registers CM0 to CM7 shows the function of these latches. The latches are implemented to prevent loss of compare matches which may occur when loading of the compare values is not correlated with the timer count. The compare latches are automatically loaded from the compare registers at every timer overflow.

Capture

This feature permits saving of the actual timer/counter contents into a selected register upon an external event or a software write operation. Two modes are provided to 'freeze' the current 16-bit value of timer 2 registers into a dedicated capture register.

Mode 0: Capture is performed in response to a transition at the corresponding port 1 pins CC0 to CC3.

Mode 1: Write operation into the low-order byte of the dedicated capture register causes the timer 2 contents to be latched into this register.

Reload of Timer 2

A 16-bit reload can be performed with the 16-bit CRC register, which is a concatenation of the 8-bit registers CRCL and CRCH. There are two modes from which to select:

Mode 0: Reload is caused by a timer overflow (auto-reload).

Mode 1: Reload is caused in response to a negative transition at pin T2EX (P1.5), which can also request an interrupt.

Timer/Counters 0 and 1

These timer/counters are fully compatible with timer/counter 0 or 1 of the SAB 8051 and can operate in four modes:

Mode 0: 8-bit timer/counter with 32:1 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto reload

Mode 3: Timer/counter 0 is configured as one8-bit timer;

External inputs $\overline{INT0}$ and $\overline{INT1}$ can be programmed to function as a gate for timer/ counters 0 and 1 to facilitate pulse width measurements.

Figure 121
Block Diagram of Timer 2

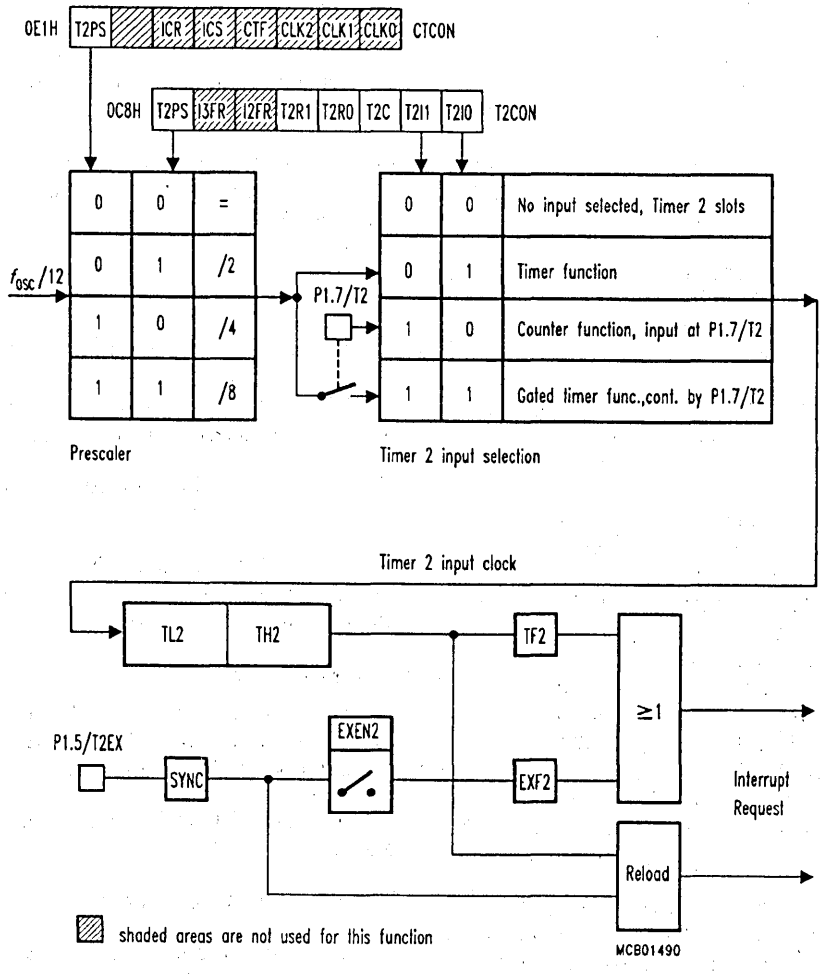


Figure 122
Block Diagram of the Compare Timer

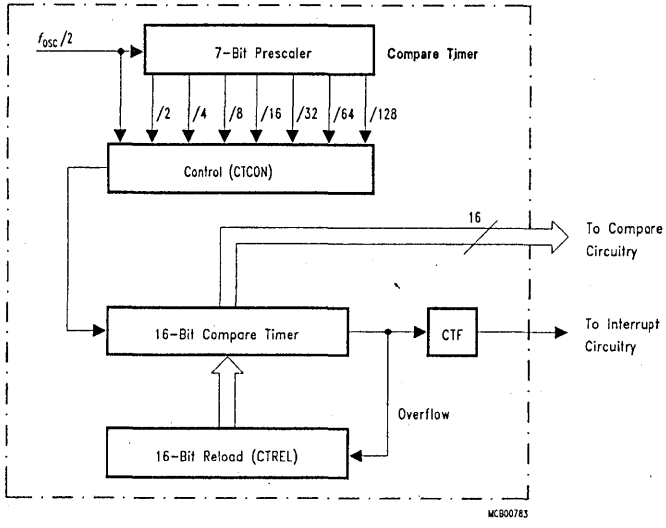


Figure 123
Compare-Mode 0 with Registers CM0 to CM7

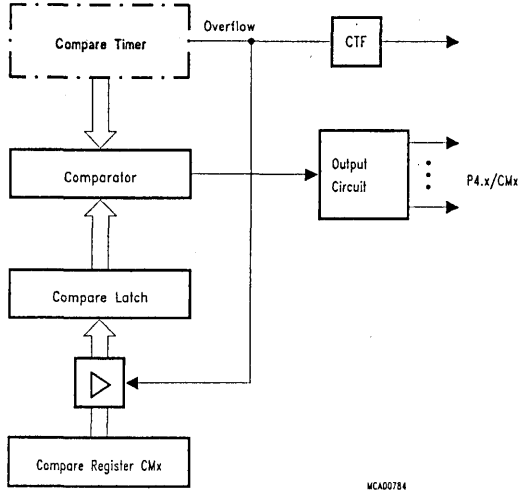
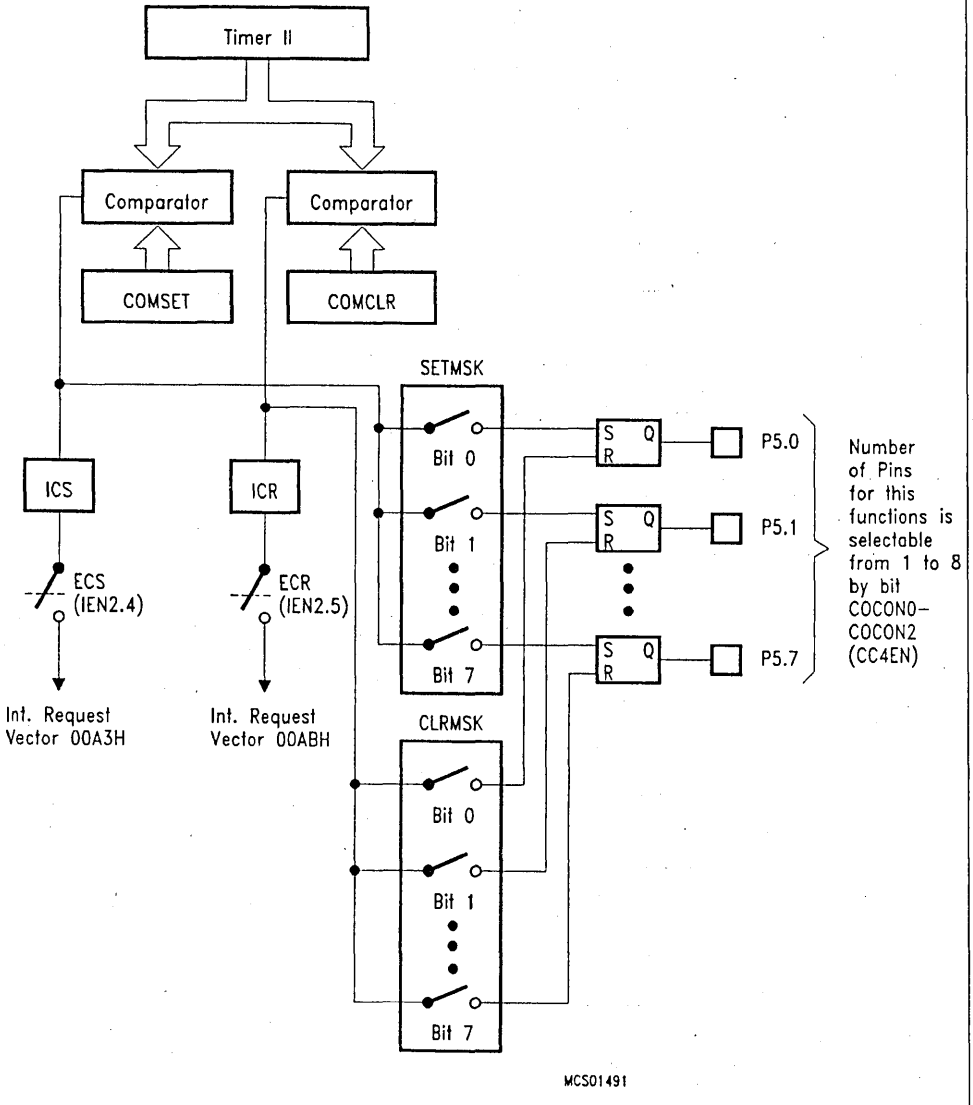


Figure 124
Compare-Mode 2 (Port 5 only)



Interrupt Structure

Table 48
Interrupt Sources and Vectors

Interrupt Request Flags	Interrupt Vector Address	Interrupt Source
IE0	0003H	External interrupt 0
TF0	000BH	Timer 0 overflow
IE1	0013H	External interrupt 1
TF1	001BH	Timer 1 overflow
RI0/TI0	0023H	Serial channel 0
TF2/EXF2	002BH	Timer 2 overflow/ext. reload
IADC	0043H	A/D converter
IEX2	004BH	External interrupt 2
IEX3	0053H	External interrupt 3
IEX4	005BH	External interrupt 4
IEX5	0063H	External interrupt 5
IEX6	006BH	External interrupt 6
RI1/TI1	0083H	Serial channel 1
ICMP0 to ICMP7	0093H	Compare match interrupt of Compare Registers CM0-CM7 assigned to Timer 2
CTF	009BH	Compare timer overflow
ICS	00A3H	Compare match interrupt of Compare Register COMSET
ICR	00ABH	Compare match interrupt of Compare Register COMCLR

The SAB 80C517A has 17 interrupt vectors with the above vector addresses and request flags. Each interrupt vector can be individually enabled/disabled. The response time to an interrupt request is more than 3 machine cycles and less than 9 machine cycles.

External interrupts 0 and 1 can be activated by a low-level or a negative transition (selectable) at their corresponding input pin, external interrupts 2 and 3 can be programmed for triggering on a negative or a positive transition. The external interrupts 2 to 6 are combined with the corresponding alternate functions compare (output) and capture (input) on port 1.

For programming of the priority levels the interrupt vectors are combined to pairs or triples. Each pair or triple can be programmed individually to one of four priority levels by setting or clearing one bit in special function register IP0 and one in IP1. Figure 125 Interrupt Structure of the SAB 80C517A shows the interrupt request sources, the enabling and the priority level structure.

Figure 125
Interrupt Structure of the SAB 80C517A

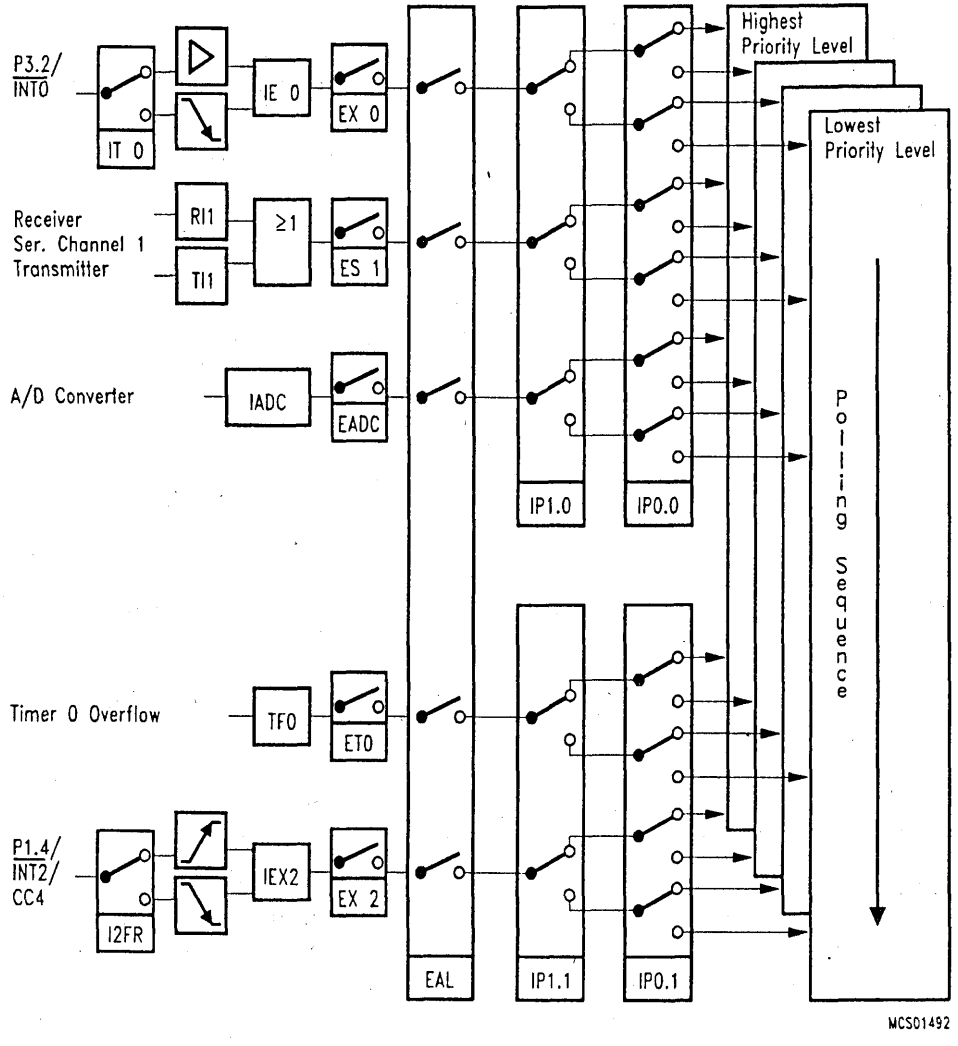
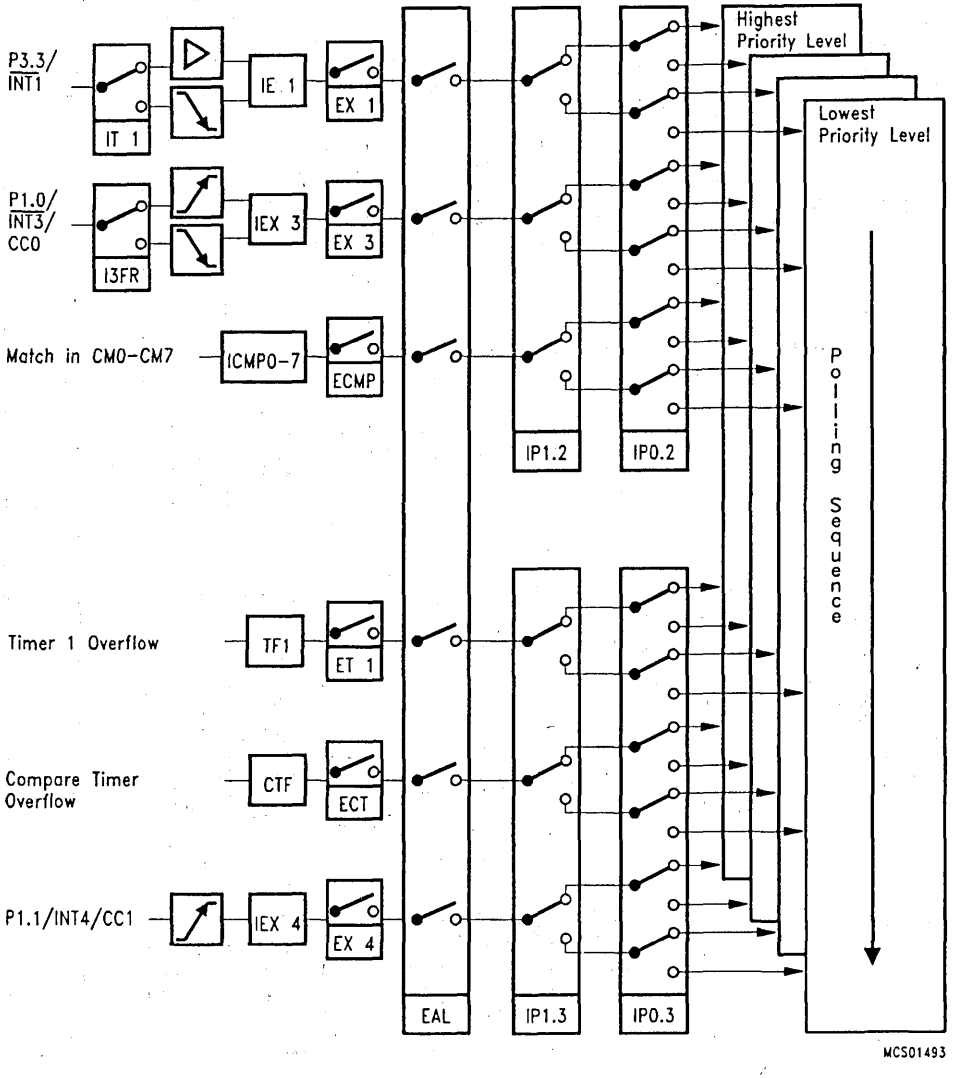
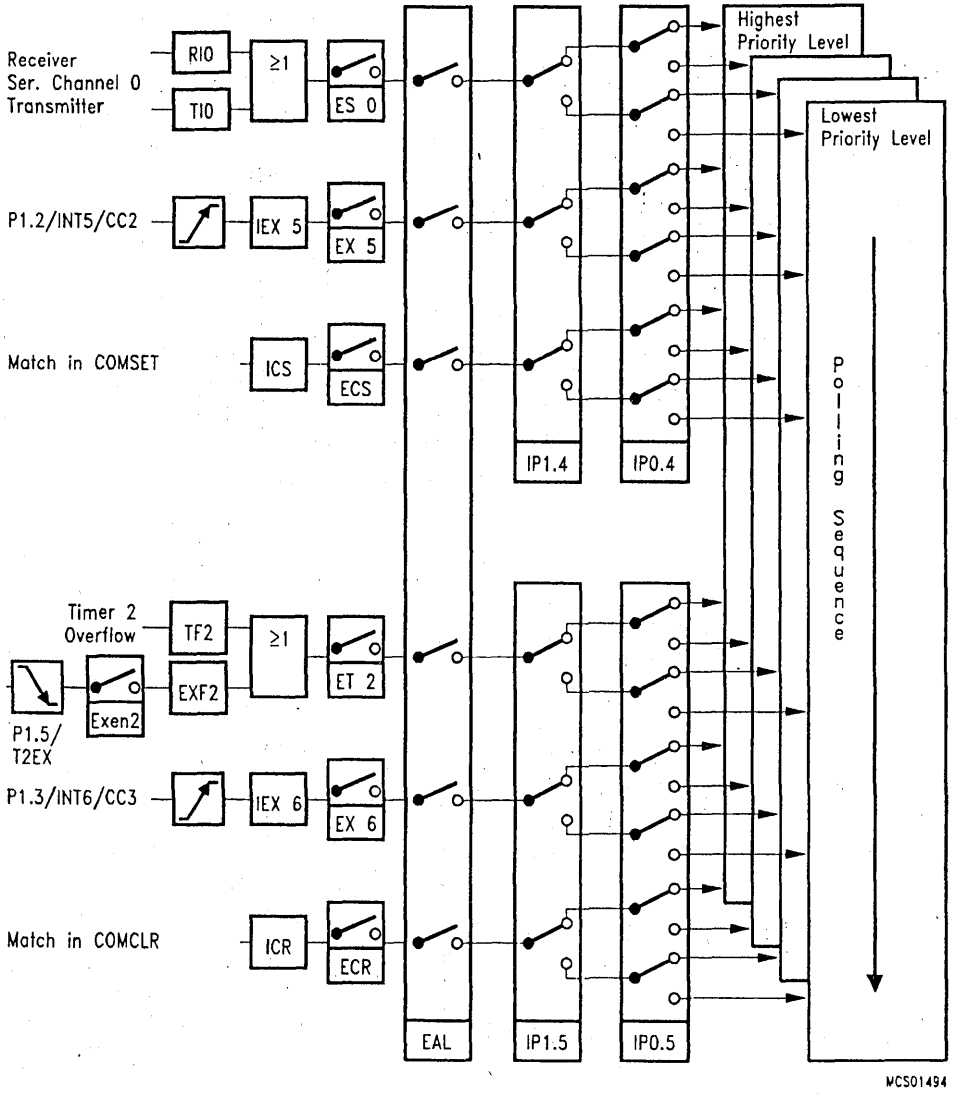


Figure 126
Interrupt Structure of the SAB 80C517A (cont'd)



4

Figure 126
Interrupt Structure of the SAB 80C517A (cont'd)



Multiplication/Division Unit

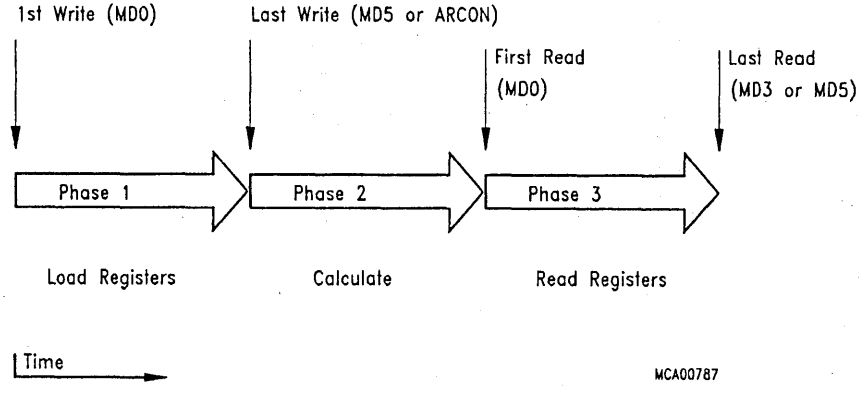
This on-chip arithmetic unit provides fast 32-bit division, 16-bit multiplication as well as shift and normalize features. All operations are integer operations.

Operation	Result	Remainder	Execution Time
32-Bit/16-Bit	32-Bit	16-Bit	6 t_{cy} [1]
16-Bit/16-Bit	16-Bit	16-Bit	4 t_{cy}
16-Bit *16-Bit	32-Bit	–	4 t_{cy}
32-Bit normalize	–	–	6 t_{cy} [2]
32-Bit shift left/right	–	–	6 t_{cy} [2]

[1] 1 t_{cy} = 1 µs @ 12 MHz oscillator frequency.
[2] The maximal shift speed is 6 shifts/cycle.

The MDU consists of six registers used for operands and results and one control register. Operation of the MDU can be divided in three phases:

Figure 127
Operation of the MDU



Operation of the MDU

To start an operation, register MD0 to MD5 (or ARCON) must be written to in a certain sequence according to Table 49 or 50. The order the registers are accessed determines the type of the operation. A shift operation is started by a final write operation to register ARCON (see also the register description).

Table 49
Performing a MDU-Calculation

Operation	32-Bit/16-Bit		16-Bit/16-Bit		16-Bit * 16-Bit	
First Write	MD0	D'endL	MD0	D'endL	MD0	M'andL
	MD1	D'end	MD1	D'endH	MD4	M'orL
	MD2	D'end	MD4	D'orL	MD1	M'andH
	MD3	D'endH	MD5	D'orH	MD5	M'orH
	MD4	D'orL				
Last Write	MD5	D'orH				
First Read	MD0	QuoL	MD0	QuoL	MD0	PrL
	MD1	Quo	MD1	QuoH	MD1	Pr
	MD2	Quo	MD4	RemL	MD2	Pr
	MD3	QuoH	MD5	RemH	MD3	PrH
	MD4	RemL				
Last Read	MD5	RemH				

Table 50
Shift Operation with the CCU

Operation	Normalize, Shift Left, Shift Right	
First Write	MD0	least significant byte
	MD1	
	MD2	
	MD3	most significant byte
Last Write	ARCON	start of conversion
First Read	MD0	least significant byte
	MD1	
	MD2	
Last Read	MD3	most significant byte

Abbreviations

D'end : Dividend, 1st operand of division
D'or : Divisor, 2nd operand of division
M'and : Multiplicand, 1st operand of multiplication
M'or : Multiplicator, 2nd operand of multiplication
Pr : Product, result of multiplication
Rem : Remainder
Quo : Quotient, result of division
...L : means, that this byte is the least significant of the 16-bit or 32-bit operand
...H : means, that this byte is the most significant of the 16-bit or 32-bit operand

I/O Ports

The SAB 80C517A has seven 8-bit I/O ports and two input ports (8-bit and 4-bit wide).

Port 0 is an open-drain bidirectional I/O port, while ports 1 to 6 are quasi-bidirectional I/O ports with internal pull-up resistors. That means, when configured as inputs, ports 1 to 6 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pull-up FET. Port 1, 3, 4, 5 and port 6 provide several alternate functions. Please see the "Pin Description" for details.

Port pins show the information written to the port latches, when used as general purpose port. When an alternate function is used, the port pin is controlled by the respective peripheral unit. Therefore the port latch must contain a "one" for that function to operate. The same applies when the port pins are used as inputs. Ports 1, 3, 4 and 5 are bit-addressable.

The SAB 80C517A has two dual-purpose input ports. The twelve port lines at port 7 and port 8 can be used as analog inputs for the A/D converter. If input voltages at P7 and P8 meet the specified digital input levels (V_{IL} and V_{IH}) the port can also be used as digital input port.

In Hardware Power Down Mode the port pins and several control lines enter a floating state. For more details see the section about Hardware Power Down Mode.

Power Saving Modes

The SAB 80C517A provides – due to Siemens ACMOS technology – four modes in which power consumption can be significantly reduced.

- The **Slow Down Mode**
  The controller keeps up the full operating functionality, but is driven with one eighth of its normal operating frequency. Slowing down the frequency remarkable reduces power consumption.
- The **Idle Mode**
  The CPU is gated off from the oscillator, but all peripherals are still supplied with the clock and continue working.
- The **Power Down Mode**
  Operation of the SAB 80C517A is stopped, the on-chip oscillator and the RC-oscillator are turned off. This mode is used to save the contents of the internal RAM with a very low standby current.
- The **Hardware Power Down Mode**
  Operation of the SAB 80C517A is stopped, the on-chip oscillator and the RC-Oscillator are turned off. The pin $\overline{\text{HWPD}}$ controls this mode. Port pins and several control lines enter a floating state. The Hardware Power Down Mode is independent of the state of pin $\overline{\text{PE}}$/SWD.

4

Hardware Enable for Software Controlled Power Saving Modes

A dedicated Pin ($\overline{PE}$/SWD) of the SAB 80C517A allows to block the Software controlled power saving modes. Since this pin is mostly used in noise-critical application it is combined with an automatic start of the Watchdog Timer.

$\overline{PE}$/SWD = V_{IH} (logic high level):
Use of the power saving modes is not possible. The watchdog timer starts immediately after reset. The instruction sequences used for entering of power saving modes will not affect the normal operation of the device.

$\overline{PE}$/SWD = V_{IL} (logic low level):
All power saving modes can be activated by software.

When left unconnected, Pin $\overline{PE}$/SWD is pulled high by a weak internal pull-up. This is done to provide system protection on default.

The logic-level applied to pin $\overline{PE}$/SWD can be changed during program execution to allow or to block the use of the power saving modes without any effect on the on-chip watchdog circuitry.

Requirements for Hardware Power Down Mode

There is no dedicated pin to enable the Hardware Power Down Mode. Nevertheless for a correct function of the Hardware Power Down Mode the oscillator watchdog unit including its internal RC oscillator is needed. Therefore this unit must be enabled by pin OWE (OWE = high). However, the control pin $\overline{PE}$/SWD has no control function in this mode. It enables and disables only the use of software controlled power saving modes.

Software Controlled Power Saving Modes

All of these modes are entered by software. Special function register PCON (power control register, address is 87H) is used to select one of these modes.

Slow Down Mode

During slow down operation all signal frequencies that are derived from the oscillator clock are divided by eight, also the clockout signal and the watchdog timer count.

The slow down mode is enabled by setting bit SD. The controller actually enters the slow down mode after a short synchronization period (max. 2 machine cycles).

The slow down mode is disabled by clearing bit SD.

Idle Mode

During idle mode all peripherals of the SAB 80C517A (except for the watchdog timer) are still supplied by the oscillator clock. Thus the user has to take care which peripheral should continue to run and which has to be stopped during Idle.

The procedure to enter the Idle mode is similar to the one entering the power down mode. The two bits IDLE and IDLS must be set by two consecutive instructions to minimize the chance of unintentional activating of the idle mode.

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and the instruction to be executed following the RETI instruction will be the one following the instruction that set the bit IDLS.
- The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to

output during idle mode if the assigned function is on. The control signals ALE and $\overline{PSEN}$ hold at logic high levels (see Table 51).

Power Down Mode

The power down mode is entered by two consecutive instructions directly following each other. The first instruction has to set the flag PDE (power down enable) and must not set PDS (power down set). The following instruction has to set the start bit PDS. Bits PDE and PDS will automatically be cleared after having been set.

The instruction that sets bit PDS is the last instruction executed before going into power down mode. The only exit from power down mode is a hardware reset.

The status of all output lines of the controller can be looked up in Table 51.

Hardware Controlled Power Down Mode

The pin $\overline{HWPD}$ controls this mode. If it is on logic high level (inactive) the part is running in the normal operating modes. If pin $\overline{HWPD}$ gets active (low level) the part enters the Hardware Power Down Mode; this is independent of the state of pin $\overline{PE}$/SWD.

$\overline{HWPD}$ is sampled once per machine cycle. If it is found active, the device starts a complete internal reset sequence. The watchdog timer is stopped and its status flag WDTS is cleared exactly the same effects as a hardware reset. In this phase the power consumption is not yet reduced. After completion of the internal reset both oscillators of the chip are disabled. At the same time the port pins and several control lines enter a floating state as shown in Table 51. In this state the power consumption is reduced to the power down current IPD. Also the supply voltage can be reduced. Table 51 also lists the voltages which may be applied at the pins during Hardware Power Down Mode without affecting the low power consumption.

Termination of HWPD Mode:

This power down state is maintained while pin $\overline{HWPD}$ is held active. If HWPD goes to high level (inactive state) an automatic start up procedure is performed:

- First the pins leave their floating condition and enter their default reset state (as they had immediately before going to float state).
- Both oscillators are enabled (only if OWE = high). The oscillator watchdog's RC oscillator starts up very fast (typ. less than 2 microseconds).
- Because the oscillator watchdog is active it detects a failure condition if the on-chip oscillator hasn't yet started. Hence, the watchdog keeps the part in reset and supplies the internal clock from the RC oscillator.
- Finally, when the on-chip oscillator has started, the oscillator watchdog releases the part from reset with oscillator watchdog status flag set.
  When automatic start of the watchdog was enabled ($\overline{PE}$/SWD connected to V_{CC}), the Watchdog Timer will start, too (with its default reload value for time-out period).
- The $\overline{Reset}$ pin overrides the Hardware Power Down function, i.e. if reset gets active during Hardware Power Down it is terminated and the device performs the normal reset function. (Thus, pin $\overline{Reset}$ has to be inactive during Hardware Power Down Mode).

4

Table 51
Status of all pins during Idle Mode, Power Down Mode and Hardware Power Down Mode

Pins	Idle Mode Last instruction executed from		Power Down Mode Last instruction executed from		Hardware Power Down	
					Status	Voltage range at pin
	internal ROM	external ROM	internal ROM	external ROM		
P0	Data	float	Data	float	floating outputs, disabled input function	$V_{SS} \le V_{IN} \le V_{CC}$
P1	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output		
P2	Data	Address	Data	Data		
P3	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output		
P4	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output		
P5	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output		
P6	Data/ alt.output	Data/ alt.output	Data/ last output	Data/ last output		
P7						
P8						
$\overline{EA}$					active input	$V_{IN} = V_{CC}$ or $V_{IN} = V_{SS}$
$\overline{PE}$/SWD					active input pull-up disabled	$V_{IN} = V_{CC}$ or $V_{IN} = V_{SS}$
XTAL1					active output	pin may not be driven
XTAL2					disabled input function	$V_{SS} \le V_{IN} \le V_{CC}$
$\overline{PSEN}$	high	high	low	low	floating outp., disabled input function	$V_{SS} \le V_{IN} \le V_{CC}$
ALE	high	high	low	low		
VAREF, VAGND					active supply pins	$V_{AGND} \le V_{IN} \le V_{CC}$

Table 51
Status of all pins during Idle Mode, Power Down Mode and Hardware Power Down Mode (cont'd)

Pins	Idle Mode Last instruction executed from		Power Down Mode Last instruction executed from		Hardware Power Down	
					Status	Voltage range at pin
	internal ROM	external ROM	internal ROM	external ROM		
OWE					active input, must be high pull-up disabl.	$V_{IN} = V_{CC}$
$\overline{\text{Reset}}$					active input must be high	$V_{IN} = V_{CC}$
$\overline{\text{RO}}$					floating output	$V_{SS} \leq V_{IN} \leq V_{CC}$

4

Serial Interfaces

The SAB 80C517A has two serial interfaces. Both interfaces are full duplex and receive buffered. They are functionally identical with the serial interface of the SAB 8051 when working as asynchronous channels. Serial interface 0 additionally has a synchronous mode. Table 52 shows possible configurations and the according baud rates.

Table 52
Baud Rate Generation

8-Bit synchronous channel	**Mode**		**Mode 0**			–
	Baud-rate	f_{OSC}= 12 MHz	1 MHz			–
		f_{OSC} = 16 MHz	1.33 MHz			
		f_{OSC} = 18 MHz	1.5 MHz			–
	derived from		f_{OSC}			–
8-Bit UART	**Mode**		**Mode 1**			**Mode B**
	Baud-rate	f_{OSC} = 12 MHz	1 Baud – 62.5 kBaud	183 Baud – 375 kBaud		366 Baud – 375 kBaud
		f_{OSC} = 16 MHz	1 Baud – 83 kBaud	244 Baud – 500 kBaud		244 Baud – 500 kBaud
		f_{OSC} = 18 MHz	1 Baud – 93.7 kBaud	2375 Baud – 562.5 kBaud		549 Baud – 562.5 kBaud
	derived from		Timer 1	10-Bit Baudrate Generator		10-Bit Baudrate Generator
9-Bit UART	**Mode**		**Mode 2**	**Mode 3**		**Mode A**
	Baud-rate	f_{OSC} = 12 MHz	187.5 kBaud/ 375 kBaud	1 Baud – 62.5 kBaud	183 Baud – 375 kBaud	366 Baud – 375 kBaud
		f_{OSC}= 16 MHz	250 Baud/ 500 kBaud	1 Baud – 83.3 kBaud	244 Baud – 500 kBaud	244 Baud – 500 kBaud
		f_{OSC} = 18 MHz	281.2 kBaud/ 562.5 kBaud	1 Baud – 93.7 kBaud	275 Baud – 562.5 kBaud	549 Baud – 562.5 kBaud
	derived from		f osc/2	Timer 1	10-Bit Baudrate Generator	10-Bit Baudrate Generator

Serial Interface 0

Serial Interface 0 can operate in 4 modes:

Mode 0: Shift register mode:

Serial data enters and exits through R × D0. T × D0 outputs the shift clock 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, variable baud rate:

10-bit are transmitted (through T × D0) or received (through R × D0): a startbit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB80 in special function register S0CON. The baud rate is variable.

Mode 2: 9-bit UART, fixed baud rate:

11-bit are transmitted (through T × D0) or received (through R × D0): a startbit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB80 in S0CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB80 or a second stop bit by setting TB80 to 1. On reception the 9th data bit goes into RB80 in special function register S0CON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-bit UART, variable baud rate:

11-bit are transmitted (through T × D0) or received (through R × D0): a startbit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

Variable Baud Rates for Serial Interface 0

Variable baud rates for modes 1 and 3 of serial interface 0 can be derived from either timer 1 or a dedicated Baudrate Generator.

The baud rate is generated by a free running 10-bit timer with programmable reload register.

$$\text{Mode 1.3 baud rate} = \frac{2^{SMOD} * f_{OSC}}{64*(2^{10}\text{-S0REL})}$$

The default value after reset in the reload registers S0RELL and S0RELH provide a baud rate of 4.8 kBaud (SMOD = 0) or 9.6 kBaud (SMOD = 1) at 12 MHz oscillator frequency. This guarantees full compatibility to the SAB 80C517.

4

Serial Interface 1

Serial interface 1 can operate in two asynchronous modes:

Mode A: 9-bit UART, variable baud rate.

11 bits are transmitted (through T × D1) or received (through R × D1): a startbit (0), 8 data bits (LSB first), a programmable 9th, and a stop bit (1). On transmission, the 9th data bit (TB81 in S1CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB81 or a second stop bit by setting TB81 to 1. On reception the 9th data bit goes into RB81 in special function register S1CON, while the stop bit is ignored.

Mode B: 8-bit UART, variable baud rate.

10 bits are transmitted (through T × D1) or received (through R × D1): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB81 in special function register S1CON.

Variable Baud Rates for Serial Interface 1.

Variable baud rates for modes A and B of serial interface 1 are derived from a dedicated baud rate generator.

The baud rate clock (baud rate = $\frac{\text{baud rate clock}}{16}$) is generated by a 10-bit free running timer with programmable reload register.

Mode A, B baudrate = $\frac{f_{OSC}}{32 * (2^{10} - \text{SREL})}$

Watchdog Units

The SAB 80C517A offers two enhanced fail safe mechanisms, which allow an automatic recovery from hardware failure or software upset:

- programmable watchdog timer (WDT), variable from 512 µs up to appr. 1.1s time-out period @12 MHz. Upward compatible to SAB 80515 watchdog.
- oscillator watchdog (OWD), monitors the on-chip oscillator and forces the micro-controller into reset state, in case the on-chip oscillator fails, controls the restart from the Hardware Power Down Mode and provides clock for a fast internal reset after power-on.

Programmable Watchdog Timer

The WDT can be activated by hardware or software.

Hardware initialization is done when pin $\overline{PE}$/SWD (Pin 4) is held high during RESET. The SAB 80C517A then starts program execution with the WDT running. Since Pin $\overline{PE}$/SWD is only sampled during Reset dynamic switching of the WDT is not possible.

Software initialization is done by setting bit SWDT.

A refresh of the watchdog timer is done by setting bits WDT and SWDT consecutively.

A block diagram of the watchdog timer is shown in Figure 128 Block Diagram of the Programmable Watchdog Timer .

When a watchdog timer reset occurs, the watchdog timer keeps on running, but a status flag WDTS is set. This flag can also be cleared by software.

Oscillator Watchdog

The unit serves three functions:

- Monitoring of the on-chip oscillator's function.
  The watchdog supervises the on-chip oscillator's frequency; if it is lower than the frequency of the auxiliary RC oscillator in the watchdog unit, the internal clock is supplied by the RC oscillator and the device is forced into reset; if the failure condition disappears (i.e. the on-chip oscillator has again a higher frequency than the RC oscillator), the part executes a final reset phase of appr. 0.25 ms in order to allow the oscillator to stabilize; then the oscillator watchdog reset is released and the part starts program execution again.
- Restart from the Hardware Power Down Mode.
  If the Hardware Power Down Mode is terminated the oscillator watchdog has to control the correct start-up of the on-chip oscillator and to restart the program. The oscillator watchdog function is only part of the complete Hardware Power Down sequence; however, the watchdog works identically to the monitoring function.
- Fast internal reset after power-on.
  In this function the oscillator watchdog unit provides a clock supply for the reset before the on-chip oscillator has started. In this case the oscillator watchdog unit also works identically to the monitoring function.

If the oscillator watchdog unit is to be used it must be enabled (this is done by applying high level to the control pin OWE).

Figure 129 Functional Block Diagram of the Oscillator Watchdog shows the block diagram of the oscillator watchdog unit. It consists of an internal RC oscillator which provides the reference frequency of the on-chip oscillator. The RC oscillator can be enabled/disabled by the control pin OWE. If it is disabled the complete unit has no function.

Figure 128
Block Diagram of the Programmable Watchdog Timer

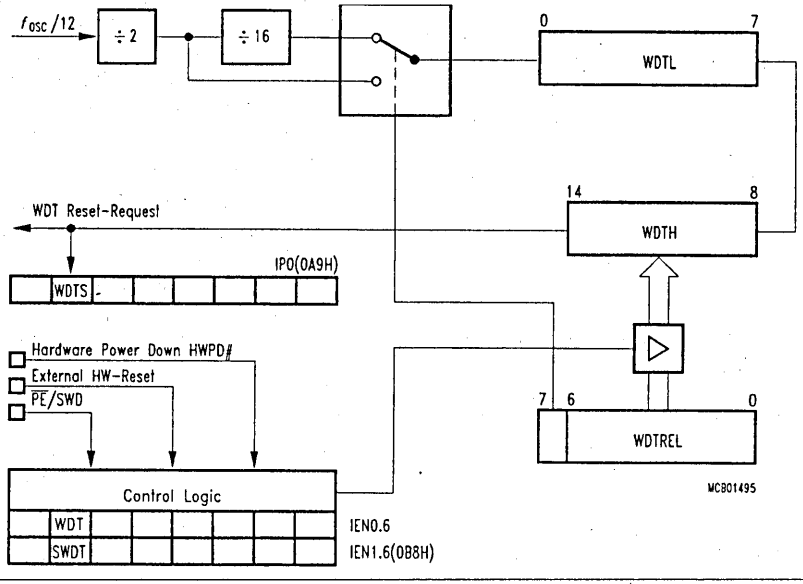
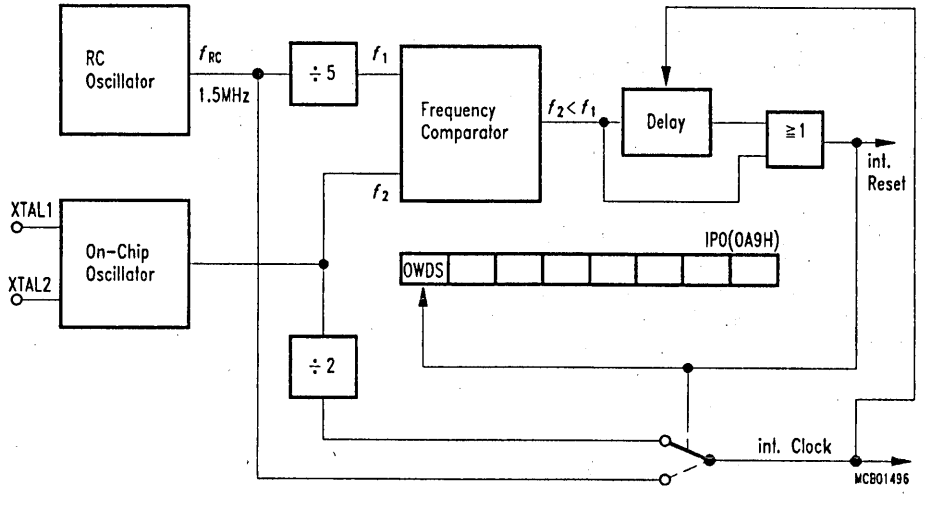


Figure 129
Functional Block Diagram of the Oscillator Watchdog



Fast internal reset after power-on

The SAB 80C517A can use the oscillator watchdog unit for a fast internal reset procedure after power-on.

Normally members of the 8051 family (like the SAB 80C517) enter their default reset state not before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed in order to bring the device into the correct reset state. Especially if a crystal is used the start up time of the oscillator is relatively long (typ. 1 ms). During this time period the pins have an undefined state which could have severe effects e.g. to actuators connected to port pins.

In the SAB 80C517A the oscillator watchdog unit avoids this situation. However, the oscillator watchdog must be enabled. In this case, after power-on the oscillator watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 micro-seconds). In the following the watchdog circuitry detects a failure condition for the on-chip oscillator because this has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is valid the watchdog uses the RC oscillator output as clock source for the chip rather than the on-chip oscillator's output. This allows correct resetting of the part and brings also all ports to the defined state.

Delay time between power-on and correct reset state:

Typ.: 18 µs
Max.: 34 µs

Instruction Set

The SAB 80C517A / 83C517A-5 has the same instruction set as the industry standard 8051 microcontroller.

4

Absolute Maximum Ratings

Ambient temperature under bias
SAB 80C517A/83C517A-5 ..... 0 to 70 °C
SAB 80C517A-T3/83C517
A-5-T3 ..... – 40 to 85 °C
SAB 80C517A-T4/83C517
A-5-T4 ..... – 40 to 110 °C
Storage temperature ..... – 65 to 150 °C
Voltage on V_{CC} pins with
respect to ground (V_{SS}) ..... – 0.5 V to 6.5 V
Power dissipation ..... 1 W

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

V_{CC} = 5 V + 10%, – 15%; V_{SS} = 0 V

T_A = 0 to 70 °C for the SAB 80C517A/83C517A-5
T_A = – 40 to 85 °C for the SAB 80C517A-T3/83C517A-5-T3
T_A = – 40 to 110 °C for the SAB 80C517A-T4/83C517A-5-T4

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, Reset, $\overline{HWPD}$)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage ($\overline{HWPD}$, Reset)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except Reset, XTAL2 and $\overline{HWPD}$)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL2	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	–
Input high voltage to Reset and $\overline{HWPD}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (ports 1, 2, 3, 4, 5, 6)	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA[1)]
Output low voltage (port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA[1)]

DC Characteristics (cont'd)

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Output high voltage (ports 1, 2, 3, 4, 5, 6)	V_{OH}	2.4 0.9 V_{CC}	– –	V V	$I_{OH} = -80\ \mu A$, $I_{OH} = -10\ \mu A$
Output high voltage (port 0 in external bus mode, ALE, $\overline{PSEN}$, $\overline{RD}$)	V_{OH1}	2.4 0.9 V_{CC}	– –	V V	$I_{OH} = -800\ \mu A$[2], $I_{OH} = -80\ \mu A$[2]
Logic Low input current (ports 1, 2, 3, 4, 5, 6)	I_{IL}	– 10	– 70	µA	$V_{IN} = 0.45$ V
Logical 1-to-0 transition current (ports 1, 2, 3, 4, 5, 6)	I_{TL}	– 65	– 650	µA	$V_{IN} = 2$ V
Input leakage current (port 0, $\overline{EA}$, ports 7, 8, $\overline{HWPD}$)	I_{LI}	–	± 100	nA	$0.45 < V_{IN} < V_{CC}$
		–	± 150	nA	$0.45 < V_{IN} < V_{CC}$ $T_A > 100$ °C
Input low current to $\overline{RESET}$ for reset	I_{IL2}	– 10	– 100	µA	$V_{IN} = 0.45$ V
Input low current (XTAL2)	I_{IL3}	–	– 15	µA	$V_{IN} = 0.45$ V
Input low current ($\overline{PE}$/SWD, OWE)	I_{IL4}	–	– 20	µA	$V_{IN} = 0.45$ V
Pin capacitance	C_{IO}	–	10	pF	$f_C = 1$ MHz, $T_A = 25$ °C
Power supply current:					
Active mode, 12 MHz[7]	I_{CC}	–	28	mA	$V_{CC} = 5$ V,[4]
Active mode, 18 MHz[7]	I_{CC}	–	37	mA	$V_{CC} = 5$ V,[4]
Idle mode, 12 MHz[7]	I_{CC}	–	24	mA	$V_{CC} = 5$ V,[5]
Idle mode, 18 MHz[7]	I_{CC}	–	31	mA	$V_{CC} = 5$ V,[5]
Slow down mode, 12 MHz	I_{CC}	–	12	mA	$V_{CC} = 5$ V,[6]
Slow down mode, 18MHz	I_{CC}	–	16	mA	$V_{CC} = 5$ V,[6]
Power Down Mode	I_{PD}	–	50	µA	$V_{CC} = 2...5.5$ V,[3]

See next page for notes.

Notes

1) Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and ports 1, 3, 4, 5 and 6. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

2) Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address lines are stabilizing.

3) I_{PD} (Power down mode) is measured with: EA = RESET = V_{CC}; Port0 = Port7 = Port8 = V_{CC}; XTAL1 = N.C.; XTAL2 = V_{SS}; PE/SWD = OWE = V_{SS}; HWPD = V_{CC} (Software Power Down mode); V_{AREF} = V_{CC}; V_{AGND} = V_{SS}; all other pins are disconnected.
Hardware Powerdown I_{PD} : OWE = V_{CC} or V_{SS}. No certain pin connection for the other pins.

4) I_{CC} (active mode) is measured with: XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.; EA = PE/SWD = V_{CC}; Port0 = Port7 = Port8 = V_{CC}; HWPD = V_{CC}; RESET = V_{SS} all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

5) I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.; RESET = V_{CC}; HWPD = V_{CC}; Port0 = Port7 = Port8 = V_{CC}; EA = PE/SWD = V_{SS}; all other pins are disconnected;

6) I_{CC} (slow down mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.; RESET = V_{CC}; HWPD = V_{CC}; Port7 = Port8 = V_{CC}; EA = PE/SWD = V_{SS}; all other pins are disconnected;

7) $I_{CC\,Max}$ at other frequencies is given by:

active mode: I_{CC} (max) = 1.50* f_{OSC} + 10
idle mode: I_{CC} (max) = 1.17* f_{OSC} + 10

where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V

A/D Converter Characteristics

V_{CC} = 5 V + 10%, – 15%; V_{SS} = 0 V
$V_{AREF} = V_{CC} \pm 5\%$; $V_{AGND} = V_{SS} \pm 0.2$ V;

T_A = 0 to 70 °C for the SAB 80C517A/83C517A-5
T_A = – 40 to 85 °C for the SAB 80C517A-T3/83C517A-5-T3
T_A = – 40 to 110 °C for the SAB 80C517A-T4/83C517A-5-T4

Parameter	Symbol	Limit Values			Units	Test Condition
		min.	typ.	max.		
Analog input capacitance	C_I		25	70	pF	
Sample time (Inc. load time)	T_S			4 t_{CY}[1]	µs	[2]
Conversion time (inc. sample time)	T_C			14 t_{CY}[1]	ms	[3]
Total unadjusted error	TUE			± 2	LSB	$V_{AREF} = V_{CC}$ $V_{AGND} = V_{SS}$
V_{AREF} supply current	I_{REF}		± 20		µA	[3]

[1] $t_{CY} = (8 * 2^{ADCL}) / f_{OSC}$; ($t_{CY} = 1/f_{ADC}$, $f_{ADC} = f_{OSC}/ (8 * 2^{ADCL})$)

[2] This parameter specifies the time during the input capacitance C_I, can be charged/discharged by the external source. It must be guaranteed, that the input capacitance C_I, is fully loaded within this time. $4t_{CY}$ is 2 µs at the f_{OSC} = 16 MHz. After the end of the sample time T_S, changes of the analog input voltage have no effect on the conversion result.

[3] This parameter includes the sample time T_S. $14t_{CY}$ is 7 µs at f_{OSC} = 16 MHz.

4

AC Characteristics

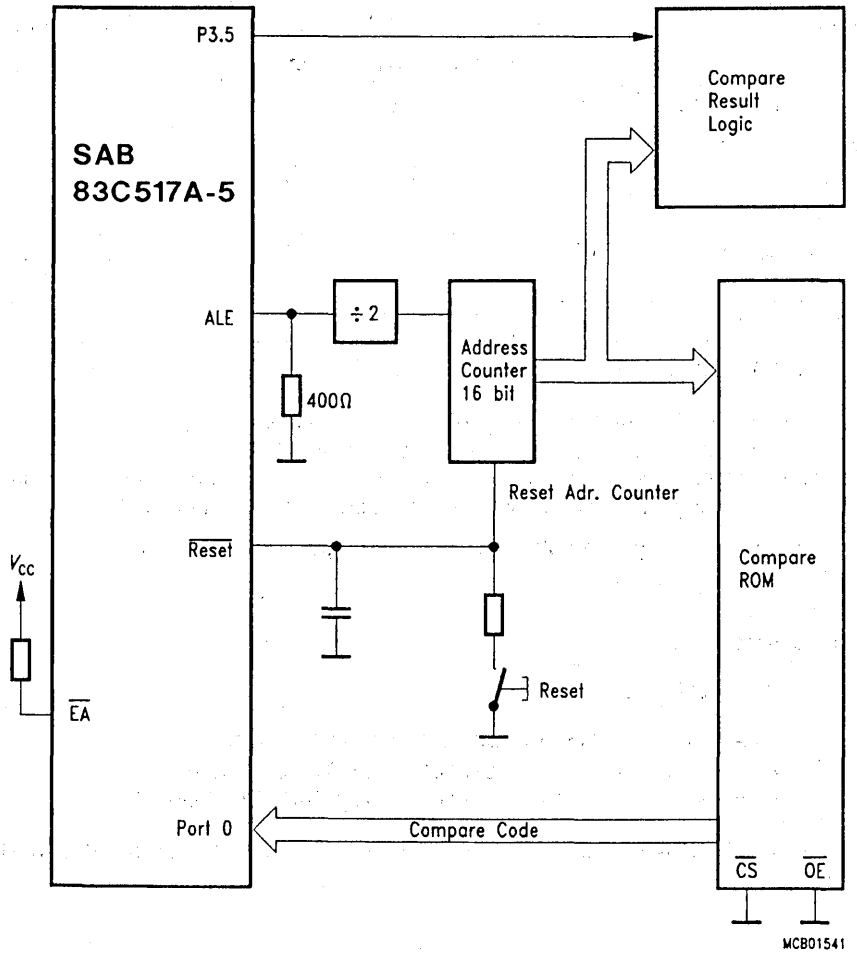
V_{CC} = 5 V + 10%, – 15%; V SS = 0 V

T_A = 0 to 70 of for the SAB 80C517A/83C517A-5
T_A = – 40 to 85 of for the SAB 80C517A-T3/83C517A-5-T3
T_A = – 40 to110 o C for the SAB 80C517A-T4/83C517A-5-T4

Please refer to the SAB 80C515A/83C515A-5 for the AC Characteristics and the Waveforms.

Please refer to the SAB 80C515A/83C515A-5 for the ROM verification characteristics.

Figure 130
Application Circuitry for Verifying the Internal ROM



Microcomputer Components

SAB 80515/SAB 80C515
SAB 80C515/SAB 80C535
SAB 80515/SAB 80535
8-Bit Single-Chip Microcontroller Family

User's Manual 12.92

Introduction

The SAB 80C515/80C535 is a new, powerful member of the Siemens SAB 8051 family of 8-bit microcontrollers. It is designed in Siemens ACMOS technology and is functionally compatible with the SAB 80515/ 80535 devices designed in MYMOS technology.

The ACMOS and the MYMOS versions [1,2] are stand-alone, high-performance single-chip microcontrollers based on the SAB 8051/80C51 architecture. While maintaining all the SAB 80(C)51 operating characteristics, the SAB 80(C)515/80(C)535[3] incorporate several enhancements which significantly increase design flexibility and overall system performance.

The low-power properties of Siemens ACMOS technology allow applications where power consumption and dissipation are critical. Furthermore, the SAB 80C515/ 80C535 has two software-selectable modes of reduced activity for further power reduction: idle and power-down mode.

The SAB 80(C)535 is identical to the SAB 80(C)515 except that it lacks the on-chip program memory. The SAB 80(C)515/ 80(C)535 is supplied in a 68-pin plastic leaded chip carrier package (P-LCC-68). In addition to the standard temperature range version (0 ° to + 70 °C) there are also versions for extended temperature ranges available (see data sheets).

Functional Description

The members of the SAB 80515 family of microcontrollers are:

SAB 80C515 Microcontroller, designed in Siemens ACMOS technology, with 8-Kbyte factory mask-programmable ROM

SAB 80C535 ROM-less version, identical to the SAB 80C515

SAB 80515 Microcontroller, designed in Siemens MYMOS technology, with 8-Kbyte factory mask-programmable ROM

SAB 80535 ROM-less version, identical to the SAB 80515

SAB 80515K Special ROM-less version of the SAB 80515 with an additional interface for program memory accesses. An external ROM that is accessed via the interface substitutes the SAB 80515's internal ROM.

The SAB 80(C)515 features are:

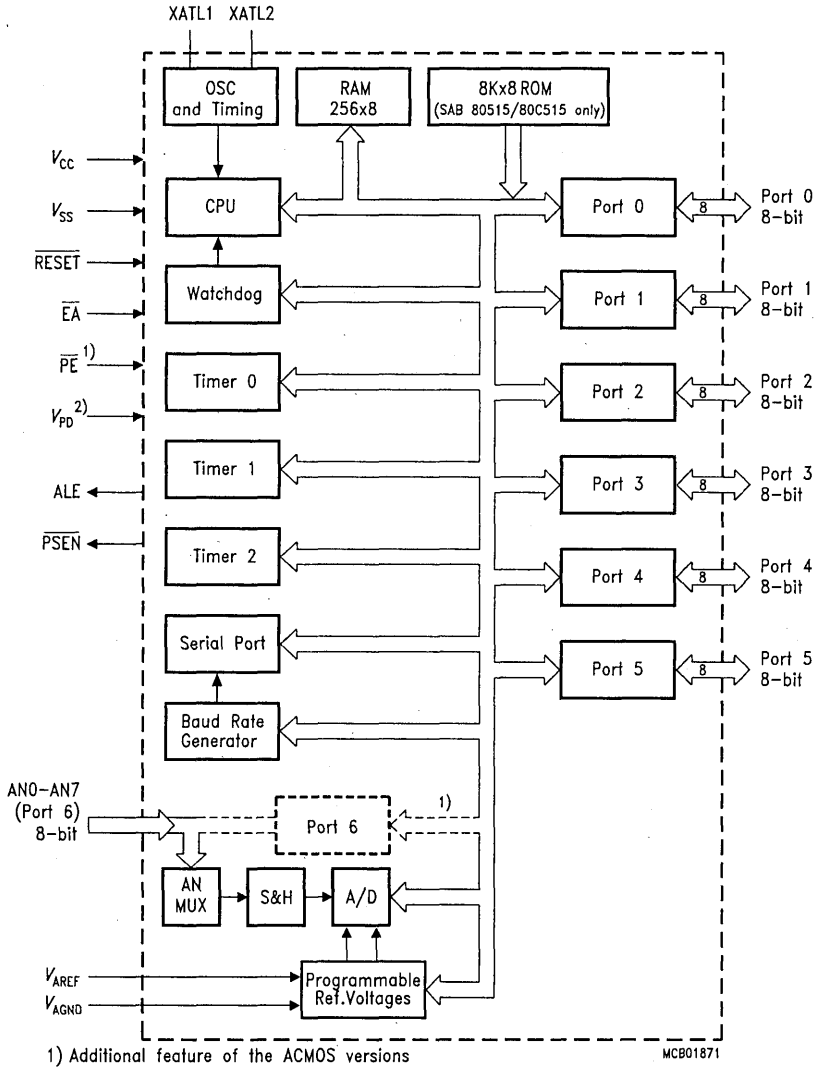
- 8 Kbyte on-chip program memory
- 256 byte on-chip RAM
- Six 8-bit parallel I/O ports
- One input port for digital input [4]
- Full-duplex serial port, 4 modes of operation, fixed or variabie baud rates
- Three 16-bit timer/counters
- 16-bit reload, compare, capture capability
- A/D converter, 8 multiplexed analog inputs, programmable reference voltages
- 16-bit watchdog timer
- Power-down supply for 40 byte of RAM
- Boolean processor
- 256 directly addressable bits
- 12 interrupt sources (7 external, 5 internal), 4 priority levels
- Stack depth up to 256 byte
- 1 µs instruction cycle at 12-MHz operation
- 4 µs multiply and divide
- External program and data memory expandable up to 64 Kbyte each
- Compatible with standard SAB 8080/8085 peripherals and memories
- Space-saving P-LCC-68 package

1. In this User's Manual the term "ACMOS versions" is used to refer to both the SAB 80C515 and SAB 80C535.
2. The term "MYMOS versions" stands for SAB 80535/80515.
3. The term "SAB 80(C)515" refers to the SAB 80515 and the SAB 80C515, unless otherwise noted.
4. Additional feature of the ACMOS versions

Introduction

For small-quantity applications and system development the SAB 80535 can be employed being the equivalent of an SAB 80515 without on-chip ROM.

Figure 2
Block Diagram



1) Additional feature of the ACMOS versions
2) Additional feature of the MYMOS versions

5

Fundamental Structure

The SAB 80(C)515/80(C)535 is a totally 8051-compatible microcontroller while its peripheral performance has been increased significantly.

Some of the various peripherals have been added to support the 8-bit core in case of stringent embedded control requirements without loosing compatibility to the 8051 architecture.

Furthermore, the SAB 80(C)515/80(C)535 contains e. g. an additional 8-bit A/D converter, two times as much ROM and RAM as the 80(C)51 and an additional timer with compare/capture/reload facilities for all kinds of digital signal processing.

"Figure 3 Detailed Block Diagram" shows a block diagram of the SAB 80(C)515/ 80(C)535.

The SAB 80C515/80C535 combines the powerful architecture of the industry standard controller SAB 80515/80535 with the advantages of the ACMOS technology (e. g. power-saving modes). The differences between MYMOS and ACMOS components are explained in Differences between MYMOS (SAB 80515/80535) and ACMOS (SAB 80C515/80C535) Versions on page 5-9.

Readers who are familiar with the SAB 8051 may concentrate on Differences between MYMOS (SAB 80515/80535) and ACMOS (SAB 80C515/80C535) Versions on page 5-9, System Reset on page 5-23, On-Chip Peripheral Components on page 5-25 and Interrupt System on page 5-77 where the differences between MYMOS and ACMOS components, the reset conditions, the peripherals and the interrupt system are described.

For newcomers to the 8051 family of microcontrollers, the following section gives a general view of the basic characteristics of the SAB 80515/80535. The details of operation are described later in Central Processing Unit on page 5-11 and Running H/F 1 on page 5-14.

Central Processing Unit

The CPU is designed to operate on bits and bytes. The instructions, which consist of up to 3 bytes, are performed in one, two or four machine cycles. One machine cycle requires twelve oscillator cycles. The instruction set has extensive facilities for data transfer, logic and arithmetic instructions. The Boolean processor has its own full-featured and bit-based instructions within the instruction set. The SAB 80(C)515/80(C)535 uses five addressing modes: direct access, immediate, register, register indirect access, and for accessing the external data or program memory portions a base register plus index-register indirect addressing.

Memory Organization

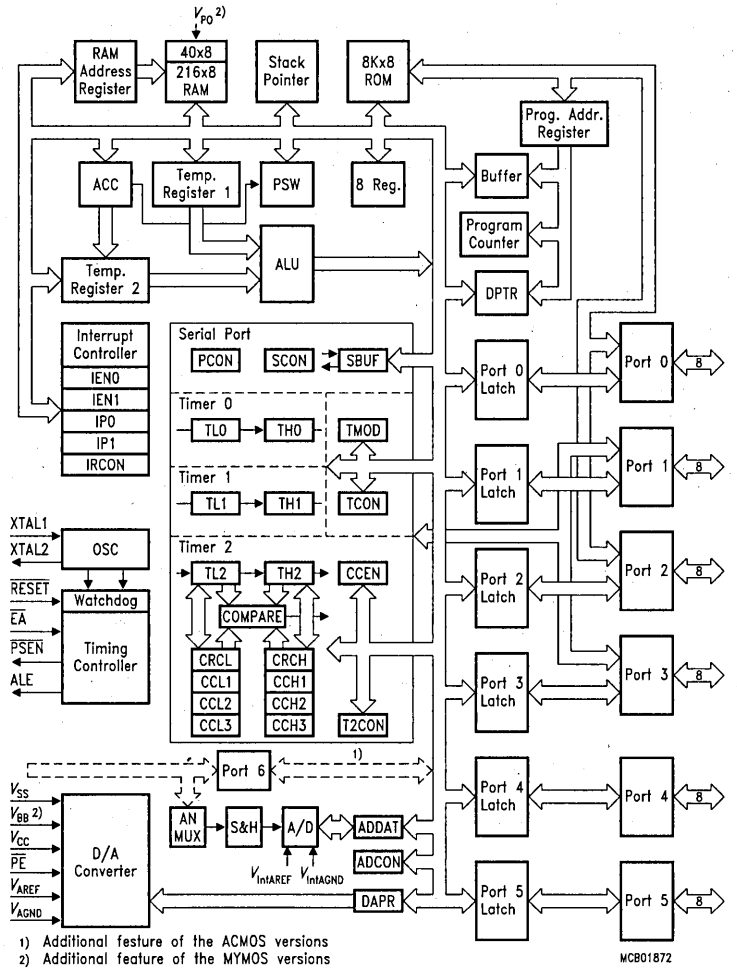
The SAB 80C515, 80515 have an internal ROM of 8 Kbyte. The program memory can externally be expanded up to 64 Kbyte (see bus expansion control). The internal RAM consists of 256 bytes. Within this address space there are 128 bit-addressable locations and four register banks, each with 8 general purpose registers. In addition to the internal RAM there is a further 128-byte address space for the special function registers, which are described in sections to follow.

Because of its Harvard architecture, the SAB 80(C)515/80(C)535 distinguishes between an external program memory portion (as mentioned above) and up to 64 Kbyte external data memory accessed by a set of special instructions.

Bus Expansion Control

The external bus interface of the SAB 80(C)515/80(C)535 consists of an 8-bit data bus (port 0), a 16-bit address bus (port 0 and port 2) and five control lines. The address latch enable signal (ALE) is used to demultiplex address and data of port 0. The program memory is accessed by the program store enable signal ($\overline{PSEN}$) twice a machine cycle. A separate external access line ($\overline{EA}$) is used to inform the controller while executing out of the lower 8 Kbyte of the program memory, whether to operate out of the internal or external program memory. The read or write strobe ($\overline{RD}$, $\overline{WR}$) is used for accessing the external data memory.

Figure 3
Detailed Block Diagram



Peripheral Control

All on-chip peripheral components - I/O ports, serial interface, timers, compare/capture registers, the interrupt controller and the A/D converter - are handled and controlled by the so-called special function registers. These registers constitute the easy-to-handle interface with the peripherals. This peripheral control concept, as implemented in the SAB 8051, provides the high flexibility for further expansion as done in the SAB 80(C)515/ 80(C)535.

Moreover some of the special function registers, like accumulator, B-register, program status word (PSW), stack pointer (SP) and the data pointer (DPTR) are used by the CPU and maintain the machine status.

Differences between MYMOS (SAB 80515/ 80535) and ACMOS (SAB 80C515/80C535) Versions

There are some differences between MYMOS and ACMOS versions concerning:

- Power Saving Modes
- Special Function Register PCON
- Port Driver Circuitry
- A/D Converter Input Ports
- A/D Converter Conversion Time
- Oscillator and Clock Circuit
- V_{BB} Pin

Power Saving Modes

The SAB 80515/80535 has just the power-down mode, which allows retention of the on-chip RAM contents through a backup supply connected to the V_{PD} pin.

The SAB 80C515/80C535 additionally has the following features:

- idle mode
- the same power supply pin V_{CC} for active, power-down and idle mode
- an extra pin ($\overline{PE}$) that allows enabling/ disabling the power saving modes
- starting of the power-saving modes by software via special function register PCON (Power Control Register)
- protection against unintentional starting of the power-saving modes

These items are described in detail in Power Saving Modes on page 5-68.

Special Function Register PCON

In the MYMOS version SAB 80515/80535 the SFR PCON (address 87H) contains only bit 7 (SMOD).

In the ACMOS version SAB 80C515/80C535 there are additional bits used (see "Figure 4 Special Function Register PCON (Address 87H)").

The bits PDE, PDS and IDLE, IDLS select the power-down mode or idle mode, respectively, when the power saving modes are enabled by pin $\overline{PE}$.

Furthermore, register PCON of the ACMOS version contains two general-purpose flags. For example, the flag bits GF0 and GF1 can be used to indicate whether an interrupt has occurred during normal operation or during idle. Then an instruction that activates idle can also set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can sample the flag bits.

Port Driver Circuitries

The port structures of the MYMOS and ACMOS versions are functionally compatible. For low power consumption the pullup arrangement is realized differently in both versions.

Port Structures on page 5-25 deals with the port structures in detail.

The A/D Converter Input Ports

The analog input ports (AN0 to AN7) of the SAB 80515/80535 can only be used as analog inputs for the A/D converter.

The analog input ports (P6.0 to P6.7) of the SAB 80C515/80C535 can be used either as input channels for the A/D converter or as digital inputs (see A/D Converter on page 5-50).

Figure 4 Special Function Register PCON (Address 87H)

87H	SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE	PCON
	7	6	5	4	3	2	1	0	

These bits are available in the MYMOS version

Symbol	Position	Function
SMOD	PCON.7	When set, the baud rate of the serial channel in mode 1, 2, 3 is doubled.
PDS	PCON.6	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	PCON.5	Idle start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
–	PCON.4	Reserved
GF1	PCON.3	General purpose flag
GF0	PCON.2	General purpose flag
PDE	PCON.1	Power-down enable bit. When set, starting of the power-down mode is enabled.
IDLE	PCON.0	Idle mode enable bit. When set, starting of the idle mode is enabled.

A/D Converter Timings

See the corresponding data sheets for the specification of t_L (load time), t_S (sample time), t_C (conversion time).

The Oscillator and Clock Circuits

There is no difference between the MYMOS and ACMOS versions if they are driven from a crystal or a ceramic resonator.

Please note that there is a difference between driving MYMOS and ACMOS components from external source. How to drive each device is described in Driving for External Source on page 5-74 and in each data sheet.

The V_{BB} Pin

The SAB 80515/80535 has an extra V_{BB} pin connected to the device's substrate. It must be connected to V_{SS} through a capacitor for proper operation of the A/D converter.

The SAB 80C515/80C535 has no V_{BB} pin. In ACMOS technology the substrate is directly connected to V_{CC}; therefore, the corresponding pin is used as an additional V_{CC} pin.

Central Processing Unit

General Description

The CPU (Central Processing Unit) of the SAB 80(C)515 consists of the instruction decoder, the arithmetic section and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers, and control the ALU processing.

The arithmetic section of the processor performs extensive data manipulation and is comprised of the Arithmetic/Logic Unit (ALU), an A register, B register and PSW register. The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under the control of the instruction decoder. The ALU performs the arithmetic operations add, subtract, multiply, divide, increment, decrement, BCD-decimal-add-adjust and compare, and the logic operations AND, OR, Exclusive OR, complement and rotate (right, left or swap nibble (left four)). Also included is a Boolean processor performing the bit operations of set, clear, complement, jump-if-not-set, jump-if-set-and-clear and move to/from carry. Between any addressable bit (or its complement) and the carry flag, it can perform the bit operations of logical AND or logical OR with the result returned to the carry flag. The A, B and PSW registers are described in Special Function Registers on page 5-15.

The program control section controls the sequence in which the instructions stored in program memory are executed. The 16-bit program counter (PC) holds the address of the next instruction to be executed. The PC is manipulated by the control transfer instructions. The conditional branch logic enables internal and external events to the processor to cause a change in the program execution sequence.

CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a phase 1 half, during which the phase 1 clock is active, and a phase 2 half, during which the phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts for two oscillator periods. Typically, arithmetic and logical operations take place during phase 1 and internal register-to-register transfers take place during phase 2.

The diagrams in Figure 5 show the fetch/execute timing related to the internal states and phases. Since these internal clock signals are not user-accessible, the XTAL2 oscillator signals and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

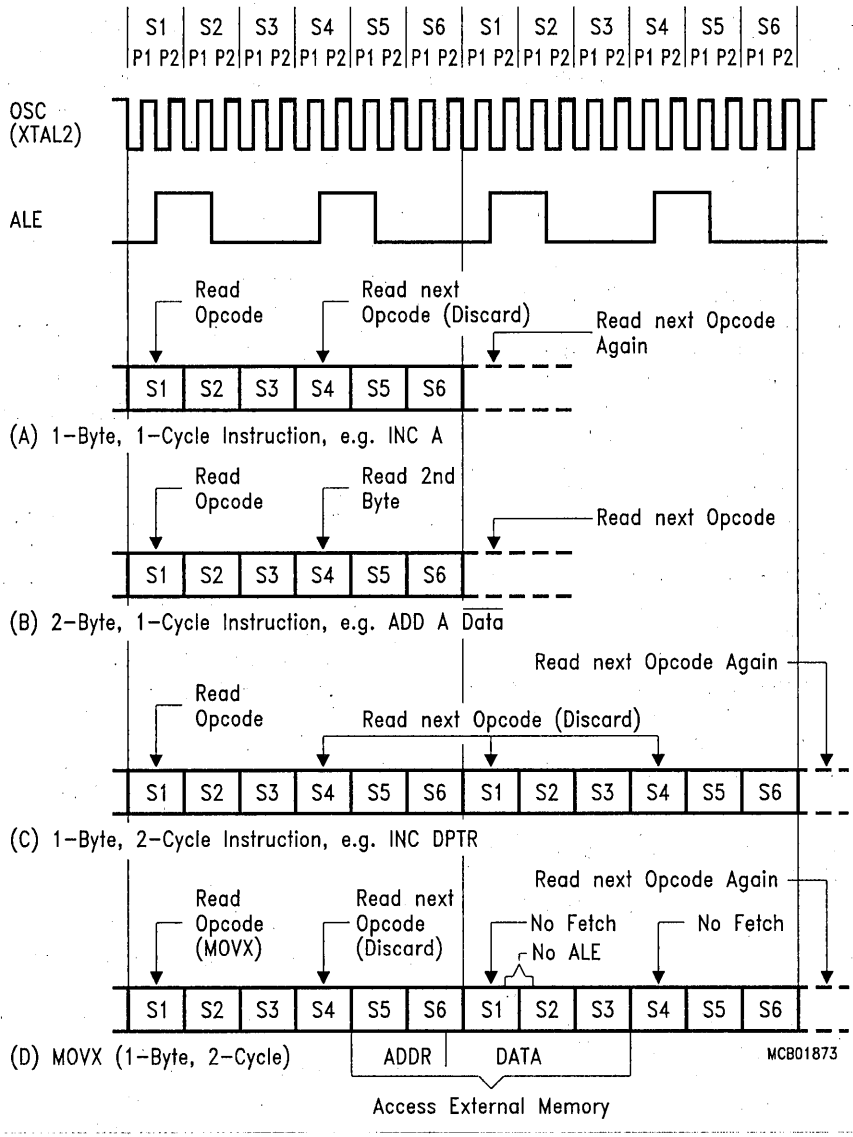
Execution of a one-cycle instruction begins at S1P2, when the op-code is latched into the instruction register. If it is a two-byte instruction, the second is read during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op-code) is ignored, and the program counter is not incremented. In any case, execution is completed at the end of S6P2.

Figures 5 A) and B) show the timing of a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most SAB 80(C)515 instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the

external data memory is being addressed and strobed. Figures 5 C) and D) show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

Figure 5
Fetch/Execute Sequence



Memory Organization

The SAB 80(C)515 CPU manipulates operands in the following four address spaces:

- up to 64 Kbyte of program memory
- up to 64 Kbyte of external data memory
- 256 bytes of internal data memory
- a 128-byte special function register area

Program Memory

The program memory of the SAB 80(C)515 consists of an internal and an external memory portion (see Figure 6 Program Memory Address Space). 8 Kbyte of program memory may reside on-chip (SAB 80C515/80515 only), while the SAB 80C535/80535 has no internal ROM. The program memory can be externally expanded up to 64 Kbyte. If the $\overline{EA}$ pin is held high, the SAB 80(C)515 executes out of the internal program memory unless the address exceeds 1FFFH. Locations 2000H through 0FFFFH are then fetched from the external memory. If the $\overline{EA}$ pin is held low, the SAB 80(C)515 fetches all instructions from the external program memory. Since the SAB 80C535/80535 has no internal program memory, pin $\overline{EA}$ must be tied low when using this device. In either case, the 16-bit program counter is the addressing mechanism.

Locations 03H through 93H in the program memory are used by interrupt service routines.

Data Memory

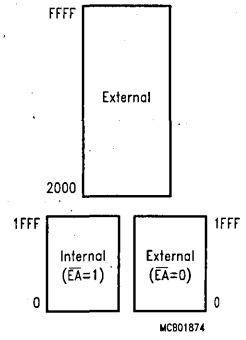
The data memory address space consists of an internal and an external memory portion.

Internal Data Memory

The internal data memory address space is divided into three physically separate and distinct blocks: the lower 128 bytes of RAM, the upper 128-byte RAM area, and the 128-byte special function register (SFR) area (see Figure 7 Data Memory/SFR Address Space). Since the latter SFR area and the upper RAM area share the same address locations, they must be accessed through different addressing modes. The map in Figure 7 Data Memory/SFR Address Space and the following table show the addressing modes used for the different RAM/SFR spaces.

Address Space	Locations	Addressing Mode
Lower 128 bytes of RAM	00H to 7FH	direct/indirect
Upper 128 bytes of RAM	80H to 0FFH	indirect
Special function registers	80H to 0FFH	direct

Figure 6
Program Memory Address Space



The lower 128 bytes of the internal RAM are again grouped in three address spaces (see Figure 8 Mapping of the Lower Portion of the Internal Data Memory):

1. list=1A general purpose register area occupies locations 0 through 1FH (see also General Purpose Registers on page 5-14).
2. The next 16 bytes, location 20H through 2FH, contain 128 directly addressable bits. Programming information: These bits can be referred to in two ways, both of which are acceptable for the ASM51. One way is to refer to their bit addresses, i.e. 0 to 7FH. The other way is by referencing to bytes 20H to 2FH. Thus bits 0 to 7 can also be referred to as bits 20.0 to 20.7, and bits 08H and 0FH are the same as 21.0 to 21.7 and so on. Each of the 16 bytes in this segment may also be addressed as a byte.)
3. Locations 30H to 7FH can be used as a scratch pad area.

Using the Stack Pointer (SP) - a special function register described in Special Function Registers on page 5-15 - the stack can be located anywhere in the whole internal data memory address space. The stack depth is limited only by the internal RAM available (256 byte maximum). However, the user has to take care that the stack is not overwritten by other data, and vice versa.

External Data Memory

Figure 7 Data Memory/SFR Address Space and Figure 8 Mapping of the Lower Portion of the Internal Data Memory contain memory maps which illustrate the internal/external data memory. To address data memory external to the chip, the "MOVX" instructions in combination with the 16-bit datapointer or an 8-bit general purpose register are used. Refer to chapter 9 (Instruction Set) or 5 (External Bus Interface) for detailed descriptions of these operations. A maximum of 64 Kbytes of external data memory can be accessed by instructions using 16-bit address.

Figure 7
Data Memory/SFR Address Space

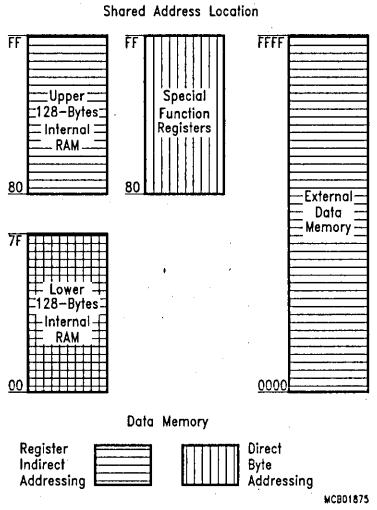
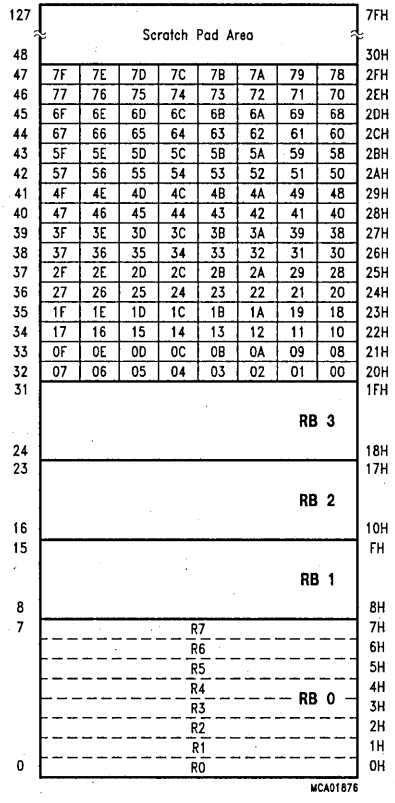


Figure 8
Mapping of the Lower Portion of the Internal Data Memory



General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general purpose register (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word, PSW.3 and PSW.4, select the active register bank (see description of the PSW). This allows fast context switching, which is useful when entering subroutines or interrupt service routines. ASM51 and the device SAB 80(C)515 default to register bank 0.

The 8 general purpose registers of the selected register bank may be accessed by register addressing. With register addressing the instruction op code indicates which register is to be used. For indirect accessing R0 and R1 are used as pointer or index register to address internal or external memory (e.g. MOV @R0).

Reset initializes the stack pointer to location 07H and increments it once to start from location 08H which is also the first register (R0) of register bank 1. Thus, if one is going to use more than one register bank, the SP should be initiated to a different location of the RAM which is not used for data storage.

Special Function Registers

The Special Function Register (SFR) area has two important functions. Firstly, all CPU register except the program counter and the four register banks reside here. The CPU registers are the arithmetic registers like A, B, PSW and pointers like SP, DPH and DPL.

Secondly, a number of registers constitute the interface between the CPU and all on-chip peripherals. That means, all control and data transfers from and to the peripherals use this register interface exclusively.

The special function register area is located in the address space above the internal RAM from addresses 80H to FFH. All 41 special function registers of the SAB 80(C)515 reside here.

Fifteen SFRs, that are located on addresses dividable by eight, are bit-addressable, thus allowing 128 bit-addressable locations within the SFR area.

Since the SFR area is memory mapped, access to the special function registers is as easy as with internal RAM, and they may be processed with most instructions. In addition, if the special functions are not used, some of them may be used as general scratch pad registers. Note, however, all SFRs can be accessed by direct addressing only.

The special function registers are listed in Table 1. Bit- and byte-addressable special function registers are marked with an asterisk at the symbol name.

Table 1
Special Function Registers

Symbol	Name	Address
* P0	Port 0	80H
SP	Stack pointer	81H
DPL	Data pointer, low byte	82H
DPH	Data pointer, high byte	83H
PCON	Power control register	87H
* TCON	Timer control register	88H
TMOD	Timer mode register	89H
TL0	Timer 0, low byte	8AH
TL1	Timer 1, low byte	8BH
TH0	Timer 0, high byte	8CH
TH1	Timer 1, high byte	8DH
* P1	Port 1	90H
* SCON	Serial channel control register	98H
SBUF	Serial channel buffer register	99H
* P2	Port 2	0A0H
* IEN0	Interrupt enable register 0	0A8H
IP0	Interrupt priority register 0	0A9H
* P3	Port 3	0B0H
* IEN1	Interrupt enable register 1	0B8H
IP1	Interrupt priority register 1	0B9H
* IRCON	Interrupt request control register	0C0H
CCEN	Compare/capture enable register	0C1H
CCL1	Compare/capture register 1, low byte	0C2H
CCH1	Compare/capture register 1, high byte	0C3H
CCL2	Compare/capture register 2, low byte	0C4H
CCH2	Compare/capture register 2, high byte	0C5H
CCL3	Compare/capture register 3, low byte	0C6H
CCH3	Compare/capture register 3, high byte	0C7H
* T2CON	Timer 2 control register	0C8H
CRCL	Compare/reload/capture register, low byte	0CAH 0CBH
CRCH	Compare/reload/capture register, high byte	0CCH 0CDH
TL2	Timer 2, low byte	0D0H
TH2	Timer 2, high byte	0D8H
* PSW	Program status word register	0D9H
* ADCON	A/D converter control register	0DAH
ADDAT	A/D converter data register	0DBH [1]
DAPR	D/A converter program register	0E0H
P6	Port 6	0E8H
* ACC	Accumulator	0F0H
* P4	Port 4	0F8H
* B	B register	
* P5	Port 5	

The SFR's marked with an asterisk (*) are bit- and byte-addressable.

[1] Additional feature of the ACMOS versions

The following paragraphs give a general overview of the special function register and refer to sections where a more detailed description can be found.

Accumulator, SFR Address 0E0H

ACC is the symbol for the accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

Figure 9
Program Status Word Register (PSW), SFR Address 0D0H

	0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H	
0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

The PSW register contains program status information.

Bit		Function
CY		Carry flag
AC		Auxiliary carry flag (for BCD operations)
F0		General purpose user flag 0
RS1	RS0	Register bank select control bits
0	0	Bank 0 selected, data address 00H - 07H
0	1	Bank 1 selected, data address 08H - 0FH
1	0	Bank 2 selected, data address 10H - 17H
1	1	Bank 3 selected, data address 18H - 1FH
OV		Overflow flag
F1		General purpose user flag
P		Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.

B Register, SFR Address 0F0H

The B register is used during multiply and divide and serves as both source and destination. For other instructions it can be treated as another scratch pad register.

Stack Pointer, SFR Address 081H

The Stack Pointer (SP) register is 8 bits wide. It is incriminated before data is stored during PUSH and CALL executions and decremented after data is popped during a POP and RET (RETI) execution, i.e. it always points to the last valid stack byte. While the stack may reside anywhere in on-chip RAM, the stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H above register bank zero. The SP can be read or written under software control.

Datapointer, SFR Address 082H and 083H

The 16-bit Datapointer (DPTR) register is a concatenation of registers DPH (data pointer's high order byte) and DPL (data pointer's low order byte). The data pointer is used in register-indirect addressing to move program memory constants and external data memory variables, as well as to branch within the 64 Kbyte program memory address space.

Ports 0 to 5

P0 to P5 are the SFR latches to port 0 to 5, respectively. The port SFRs 0 to 5 are bit-addressable. Ports 0 to 5 are 8-bit I/O ports (that is in total 48 I/O lines) which may be used as general purpose ports and which provide alternate output functions dedicated to the on-chip peripherals of the SAB 80(C)515.

Port 6 (AN0 to AN7)

In the MYMOS versions, the analog input lines AN0 to AN7 can only be used as inputs for the A/D converter.

In the ACMCS versions these lines may also be used as digital inputs. In this case they are addressed as an additional input port (port 6) via special function register P6 (0DBH). Since port 6 has no internal latch, the contents of SFR P6 only depends on the levels applied to the input lines.

For details about this port please refer to Parallel I/O on page 5-25.

Peripheral Control, Data and Status Registers

Most of the special function registers are used as control, status, and data registers to handle the on-chip peripherals.

In the special function register table the register names are organized in groups and each of these groups refer to one peripheral unit. More details on how to program these registers are given in the descriptions of the following peripheral units:

Unit	Symbol	Section
Ports	–	Parallel I/O on page 5-25
Serial Channel	–	Serial Interfaces on page 5-33
Timer 0/1	–	Timer 0 and Timer 1 on page 5-46
A/D-Converter	ADC	A/D Converter on page 5-50
Timer 2 with Comp/Capt/Reload	CCU	Timer 2 with Additional Compare/Capture/Reload on page 5-56
Power Saving Modes	–	Power Saving Modes on page 5-68
Watchdog Timer	WDT	Watchdog Timer on page 5-72
Interrupt System	–	Interrupt System on page 5-77

External Bus Interface

The SAB 80(C)515 allows for external memory expansion. To accomplish this, the external bus interface common to most 8051-based controllers is employed.

Accessing External Memory

It is possible to distinguish between accesses to external program memory and external data memory or other peripheral components respectively. This distinction is made by hardware: accesses to external program memory use the signal $\overline{PSEN}$ (program store enable) as a read strobe. Accesses to external data memory use $\overline{RD}$ and $\overline{WR}$ to strobe the memory (alternate functions of P3.7 and P3.6, see "Parallel I/O" on page 25). Port 0 and port 2 (with exceptions) are used to provide data and address signals. In this section only the port 0 and port 2 functions relevant to external memory accesses are described (for further details see chapter 7.1).

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Role of P0 and P2 as Data/Address Bus

When used for accessing external memory, port 0 provides the data byte time-multiplexed with the low byte of the address. In this state, port 0 is disconnected from its own port latch, and the address/data signal drives both FETs in the port 0 output buffers. Thus, in this application, the port 0 pins are not open-drain outputs and do not require external pullup resistors.

During any access to external memory, the CPU writes 0FFH to the port 0 latch (the special function register), thus obliterating whatever information the port 0 SFR may have been holding.

Whenever a 16-bit address is used, the high byte of the address comes out on port 2, where it is held for the duration of the read or write cycle. During this time, the port 2 lines are disconnected from the port 2 latch (the special function register).

Thus the port 2 latch does not have to contain 1 s, and the contents of the port 2 SFR are not modified.

If an 8-bit address is used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. It should be noted that, if a port 2 pin outputs an address bit that is a 1, strong pullups will be used for the entire read/write cycle and not only for two oscillator periods.

Timing

The timing of the external bus interface, in particular the relationship between the control signals ALE, $\overline{PSEN}$, $\overline{RD}$ and information on port 0 and port 2, is illustrated in "Figure 10 a) and b) External Program Memory Execution".

Data Memory	In a write cycle, the data byte to be written appears on port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at port 0 before the read strobe is deactivated.
Program Memory	Signal $\overline{PSEN}$ functions as a read strobe. For further information see "PSEN, Program Store Enable" on page 20.

External Program Memory Access

The external program memory is accessed under two conditions:

- whenever signal $\overline{EA}$ is active; or

- whenever the program counter (PC) contains a number that is larger than 01FFFH.

This requires the ROM-less versions SAB 80C535/80535 to have $\overline{EA}$ wired low to allow the lower 8 K program bytes to be fetched from external memory.

When the CPU is executing out of external program memory, all 8 bits of port 2 are dedicated to an output function and may not be used for general-purpose I/O. The contents of the port 2 SFR however is not affected. During external program memory fetches port 2 lines output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @DPTR or a MOVX @Ri).

Since the SAB 80C535/80535 has no internal program memory, accesses to program memory are always external, and port 2 is at all times dedicated to output the high-order address byte. This means that port 0 and port 2 of the SAB 80C535/80535 can never be used as general-purpose I/O. This also applies to the SAB 80C515/80515 when it is operated with only an external program memory.

$\overline{PSEN}$, Program Store Enable

The read strobe for external fetches is $\overline{PSEN}$. $\overline{PSEN}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{PSEN}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as for $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 12 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$ takes 6 oscillator periods. The execution sequence for these two types of read cycles is shown in "Figure 10 a) and b) External Program Memory Execution".

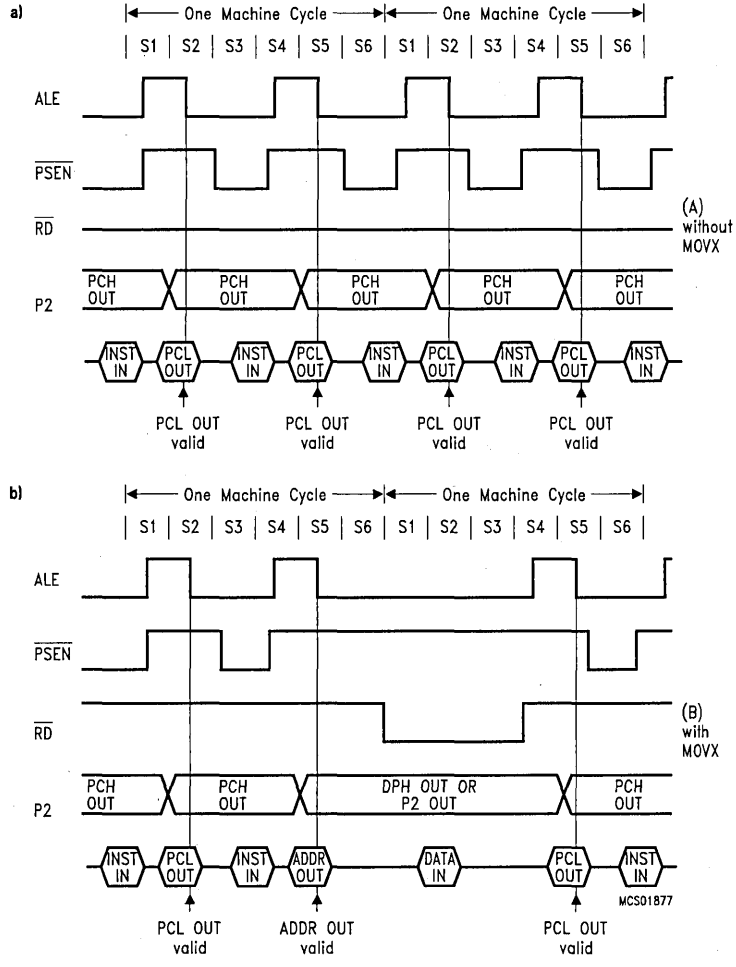
ALE, Address Latch Enable

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 into an external latch during fetches from external memory. The address byte is valid at the negative transition of ALE. For that purpose, ALE is activated twice every machine cycle. This activation takes place even it the cycle involves no external fetch. The only time no ALE pulse comes out is during an access to external data memory when $\overline{RD}$/$\overline{WR}$ signals are active. The first ALE of the second cycle of a MOVX instruction is missing (see "Figure 10 a) and b) External Program Memory Execution"). Consequently, in any system that does not use data memory, ALE is activated at a constant rate of 1/6 of the oscillator frequency and can be used for external clocking or timing purposes.

Overlapping External Data and Program Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is used for storing data. In the SAB 80(C)515, the external program and data memory spaces can be combined by AND-ing $\overline{PSEN}$ and $\overline{RD}$. A positive logic AND of these two signals produces an active low read strobe that can be used for the combined physical memory. Since the $\overline{PSEN}$ cycle is faster than the $\overline{RD}$ cycle, the external memory needs to be fast enough to adapt to the $\overline{PSEN}$ cycle.

Figure 10 a) and b)
External Program Memory Execution



System Reset

Hardware Reset and Power-Up Reset

Reset Function and Circuitries

The hardware reset function incorporated in the SAB 80(C)515 allows for an easy automatic start-up at a minimum of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation in order to restart the device. This is particularly done when the power-down mode (see "Power Saving Modes" on page 68) is to be terminated.

In addition to the hardware reset, which is applied externally to the SAB 80(C)515, there is also the possibility of an internal hardware reset. This internal reset will be initiated by the watchdog timer ("Watchdog Timer" on page 72).

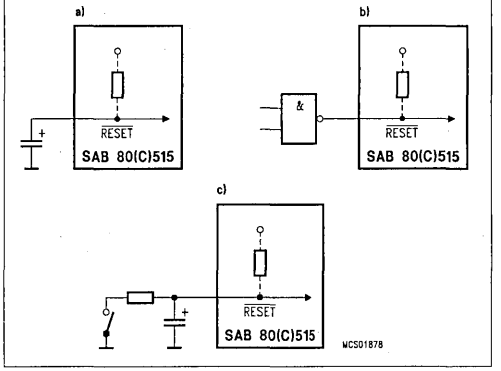
The reset input is an active low input at pin 10 ($\overline{\text{RESET}}$). An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the $\overline{\text{RESET}}$ pin must be held low for at least two machine cycles (24 oscillator periods) while the oscillator is running. With the oscillator running the internal reset is executed during the second machine cycle in which $\overline{\text{RESET}}$ is low and is repeated every cycle until $\overline{\text{RESET}}$ goes high again.

During reset, pins ALE and $\overline{\text{PSEN}}$ are configured as inputs and should not be stimulated externally. (An external stimulation at these lines during reset activates several test modes which are reserved for test purposes. This in turn may cause unpredictable output operations at several port pins).

A pullup resistor is internally connected to V_{CC} to allow a power-up reset with an external capacitor only. An automatic reset can be obtained when V_{CC} is applied by connecting the reset pin to V_{SS} via a capacitor as shown in "Table 2 Register Contents after Reset". After V_{CC} has been turned on the capacitor must hold the voltage level at the reset pin for a specified time below the upper threshold of the Schmitt trigger to effect a complete reset.

The time required is the oscillator start-up time plus 2 machine cycles, which, under normal conditions, must be at least 10 - 20 ms for a crystal oscillator. This requirement is usually met using a capacitor of 4.7 to 10 microfarad. The same considerations apply if the reset signal is generated externally ("Figure 11 a) - c) Reset Circuitries"). In each case it must be assured that the oscillator has started up properly and that at least two machine cycles have passed before the reset signal goes inactive.

Figure 11 a) - c)
Reset Circuitries



A correct reset leaves the processor in a defined state. The program execution starts at location 0000H. The default values of the special function registers (SFR) during and after reset are listed in "Table 2 Register Contents after Reset". After reset is internally accomplished the contents of the port latches of port 0 to 5 is 0FFH. This leaves port 0 floating, since it is an open drain port when not used as data/address bus. All other I/O port lines (ports 1 through 5) output a one (1).

In the MYMOS versions, the analog input lines AN0 to AN7 can only be used as inputs.

In the ACMOS versions these lines may also be used as digital inputs. In this case they are addressed as an additional input port (port 6) via special function register P6 (0DBH) Since port 6 has no internal latch, the contents of SFR P6 only depends on the levels applied to the input lines.

For details about this port please refer to "Parallel I/O" on page 25.

The contents of the internal RAM of the SAB 80(C)515 is not affected by a reset. After power-up the contents is undefined, while it remains unchanged during a reset if the power supply is not turned off.

Table 2
Register Contents after Reset

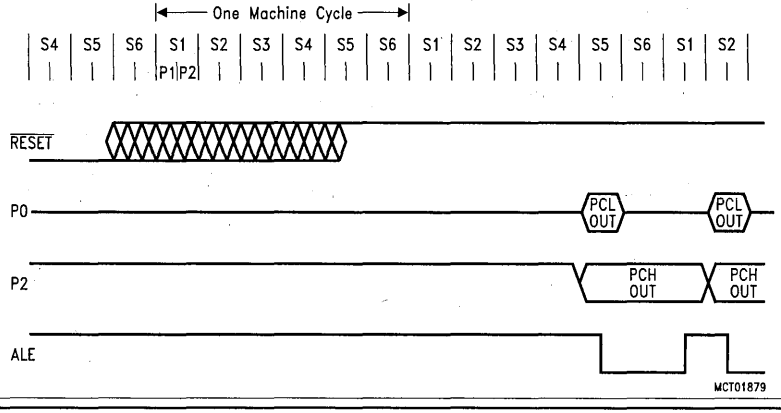
Register	Contents	Register	Contents
P0 - P5	0FFH	PCON	000X
DPTR	0000H	0000B	
TCON	00H	TMOD	00H
TL0, TH0	00H	TL1, TH1	00H
TL2, TH2	00H	SCON	00H
IEN0, IEN1	00H	SBUF	undefined
IRCON	00H	IP0	X000
		0000B	
CCL1, CCH1	00H	IP1	XX00
CCL3, CCH3	00H	0000B	
T2CON	00H	CCEN	00H
ADCON	00X0 0000B	CCL2, CCH2	00H
DAPR	00H	CRCL, CRCH	00H
B	00H	PSW	00H
PC	0000H	ADDAT	00H
SP	07H	ACC	00H
		Watchdog	0000H

Hardware Reset Timing

This section describes the timing of the hardware reset signal.

The input pin $\overline{\text{RESET}}$ is sampled once during each machine cycle. This happens in state 5 phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (low level at pin 10) the internal reset procedure is started. It needs two complete machine cycles to put the complete device to its correct reset state. i.e. all special function registers contain their default values, the port latches contain 1's etc. Note that this reset procedure is not performed if there is no clock available at the device. The $\overline{\text{RESET}}$ signal must be active for at least two machine cycles; after this time the SAB 80(C)515 remains in its reset state as long as the signal is active. When the signal goes inactive this transition is recognized in the following state 5 phase 2 of the machine cycle. Then the processor starts its address output (when configured for external ROM) in the following state 5 phase 1. One phase later (state 5 phase 2) the first falling edge at pin ALE occurs. "Figure 12 CPU Timing after RESET" shows this timing for a configuration with $\overline{\text{EA}}$ = 0 (external program memory). Thus, between the release of the $\overline{\text{RESET}}$ signal and the first falling edge at ALE there is a time period of at least one machine cycle but less than two machine cycles.

Figure 12
CPU Timing after RESET



On-Chip Peripheral Components

This chapter gives detailed information about all on-chip peripherals of the SAB 80(C)515 except for the integrated interrupt controller, which is described separately in Running H/F 1 on page 5-77. Parallel I/O on page 5-25 and Serial Interfaces on page 5-33 are associated with the general parallel and serial I/O facilities while the remaining sections describe the miscellaneous functions such as the timers, serial interface, A/D converter, power saving modes, watchdog timer, oscillator and clock circuitries, and system clock output.

Parallel I/O

Port Structures

Digital I/O

The SAB 80(C)515 allows for digital I/O on 48 lines grouped into 6 bidirectional 8-bit ports. Each port bit consists of a latch, an output driver and an input buffer. Read and write accesses to the I/O ports P0 through P5 are performed via their corresponding special function registers P0 to P5.

The output drivers of port 0 and 2 and the input buffers of port 0 are also used for accessing external memory. In this application, port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the port 2 pins continue emitting the P2 SFR contents (see also Port 0 and Port 2 Used as Address/ Data Bus on page 5-29 and Running H/F 1 on page 5-19 for more details about the external bus interface).

Digital/Analog Input Port

The analog input lines AN0 to AN7 of the MYMOS versions can only be used as analog inputs.

In the ACMOS versions these lines may also be used as digital inputs. In this case they are addressed as an additional input port (port 6) via special function register P6 (0DBH). Since port 6 has no internal latch, the contents of SFR P6 only depends on the levels applied to the input lines.

When used as analog input the required analog channel is selected by a three-bit field in SFR ADCON , as described in A/D Converter on page 5-50. Of course, it makes no sense to output a value to these input-only ports by writing to the SFR P6 or P8; this will have no effect.

If a digital value is to be read, the voltage levels are to be held within the input voltage specifications (V_{IL}/V_{IH}). Since P6 is not a bit-addressable register, all input lines of P6 are read at the same time by byte instructions.

Nevertheless, it is possible to use port 6 simultaneously for analog and digital input. However, care must be taken that all bits of P6 are masked which have an undetermined value caused by their analog function .

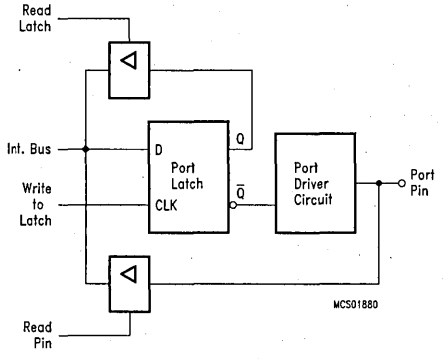
In order to guarantee a high-quality A/D conversion, digital input lines of port 6 should not toggle while a neighbouring port pin is executing an A/D conversion. This could produce crosstalk to the analog signal.

Digital I/O Port Circuitry (MYMOS/ACMOS)

Figure 13 Basic Structure of a Port Circuitry shows a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the 6 I/O-ports. The bit latch (one bit in the port's SFR) is represented as a type-D flip-flop, which will clock in a value from the internal bus in response to a "write-to-latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read-latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read-pin" signal from the CPU. Some instructions that read from a port (i.e. from the corresponding port

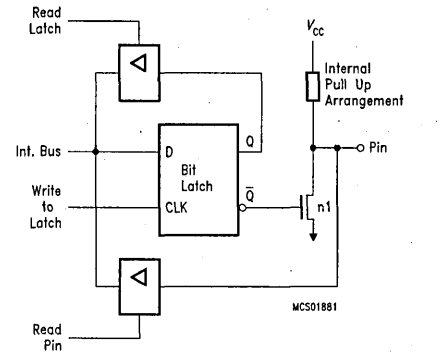
SFR P0 to P5) activate the "read-latch" signal, while others activate the "read-pin" signal (see Read-Modify-Write Feature of Ports 0 through 5 on page 5-32).

Figure 13
Basic Structure of a Port Circuitry



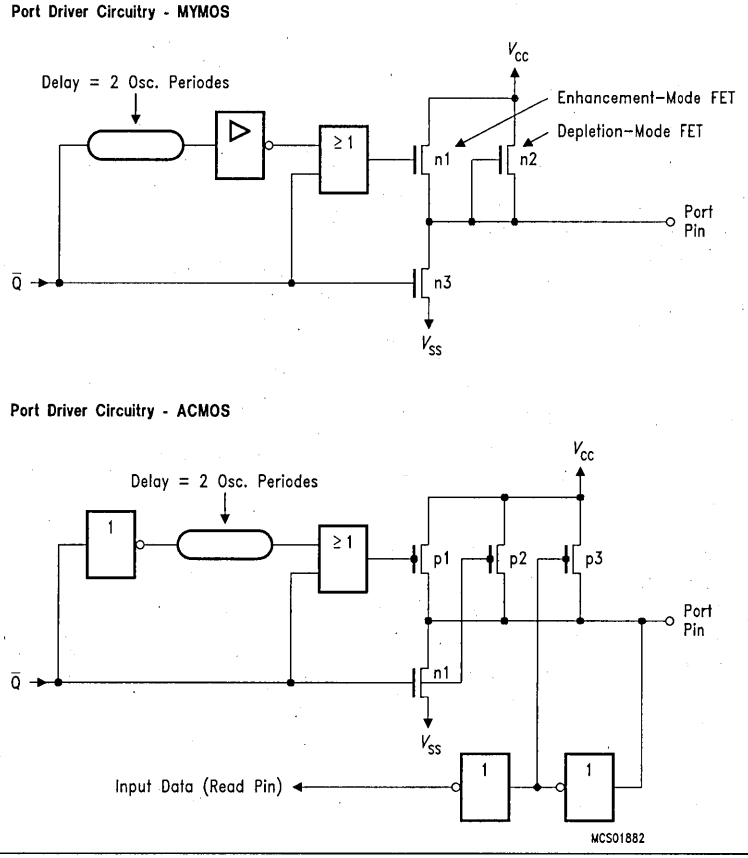
Port 1 through 5 output drivers have internal pullup FET's (see Figure 14 Basic Output Driver Circuit of Ports 1 through 5). Each I/O line can be used independently as an input or output. To be used as an input, the port bit must contain a one (1) (that means for Figure 14: Q = 0), which turns off the output driver FET n1. Then, for ports 1 through 5, the pin is pulled high by the internal pullups, but can be pulled low by an external source. When externally pulled low the port pins source current (I_{IL} or I_{TL}). For this reason these ports are sometimes called "quasi-bidirectional".

Figure 14
Basic Output Driver Circuit of Ports 1 through 5



In fact, the pullups mentioned before and included in Figure 14 Basic Output Driver Circuit of Ports 1 through 5 are pullup arrangements as shown in Figure 15 Output Driver Circuits of Ports 1 through 5. These pullup arrangements are realized differently in the MYMOS and ACMOS versions. In the next two sections both versions are discussed separately.

Figure 15
Output Driver Circuits of Ports 1 through 5



MYMOS Port Driver Circuitry

The output driver circuitry of the MYMOS version (Figure 15 Output Driver Circuits of Ports 1 through 5) consists of two pullup FETs (pullup arrangements) and one pulldown FET:

The **transistor n1** is a very strong pullup transistor which is only activated for two oscillator periods, if a 0-to-1 transition is executed by this port bit. Transistor n1 is capable of driving high currents.

The **transistor n2** is a weak pullup transistor, which is always switched on. When the pin is pulled down (e.g. when the port is used as input), it sources a low current. This value can be found as the parameter I_{IL} in the DC characteristics.

The **transistor n3** is a very strong pull-down transistor which is switched on when a "0" is programmed to the corresponding port latch. Transistor n3 is capable of sinking high currents (I_{OL} in the DC characteristics).
A short circuit to V_{CC} must be avoided if the transistor is turned on because the high current might destroy the FET.

ACMOS Port Driver Circuitry

The output driver circuitry of the ACMOS version (Figure 15 Output Driver Circuits of Ports 1 through 5) is realized by three pullup FETs (pullup arrangement) and one pulldown FET:

The **pulldown FET n1** is of n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a "0" is programmed to the port pin. A short circuit to V_{CC} must be avoided if the transistor is turned on, since the high current might destroy the FET.

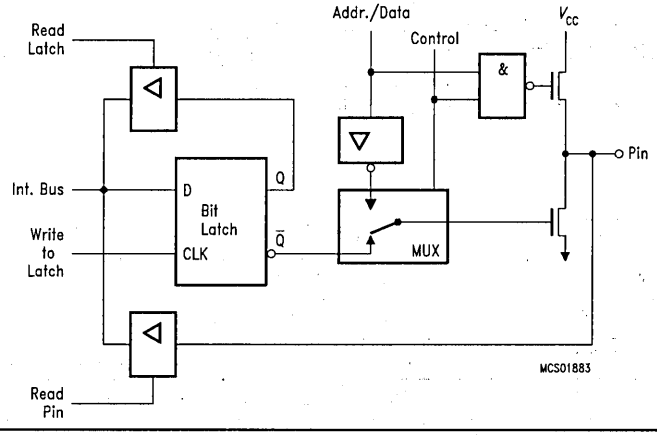
The **pullup FET p1** is of p-channel type. It is activated for two oscillator periods (S1P1 and S1P2) if a 0-to-1 transition is programmed to the port pin, i.e. a "1" is programmed to the port latch which contained a "0". The extra pullup can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.

The **pullup FET p2** is of p-channel type. It is always activated when a "1" is in the port latch, thus providing the logic high output level. This pullup FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.

The **pullup FET p3** is of p-channel type. It is only activated if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pullup current if a logic high level is to be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level, e.g. when used as input. In this configuration only the weak pullup FET p2 is active, which sources the current I_{IL}. If, in addition, the pullup FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, an additional power consumption can be avoided if port pins are used as inputs with a low level applied. However, the driving cabability is stronger if a logic high level is output.

Port 0, in contrast to ports 1 through 5, is considered as "true" bidirectional, because the port 0 pins float when configured as inputs. Thus, this port differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 16 a) Port 0 Circuitry) is used only when the port is emitting 1 s during the external memory accesses. Otherwise, the pullup is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a "1" to th port latch leaves both output FETs off and the pin floats. In that condition it can be used as high-impedance input. If port 0 is configured as general I/O port and has to emit logic high level (1), external pullups are required.

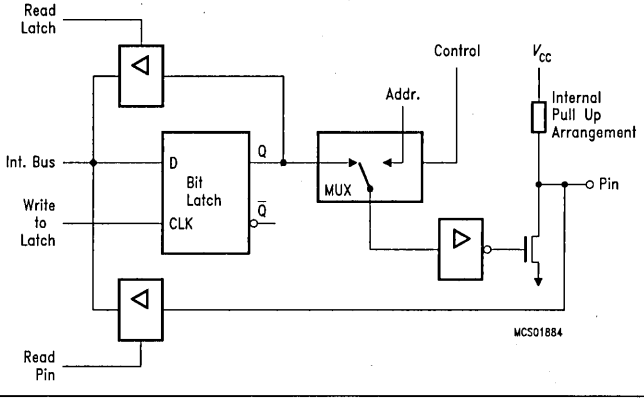
Figure 16 a)
Port 0 Circuitry



Port 0 and Port 2 Used as Address/Data Bus

As shown in Figure 16 a) Port 0 Circuitry and Figure 16 b) Port 2 Circuitry), the output drivers of ports 0 and 2 can be switched to an internal address or address/data bus for use in external memory accesses. In this application they cannot be used as general purpose I/O, even if not all address lines are used externally. The switching is done by an internal control signal dependent on the input level at the EA pin and/ or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P2 SFR remains unchanged while the P0 SFR has 1's written to it. Being an address/data bus, port 0 uses a pullup FET as shown in Figure 16 a) Port 0 Circuitry. When a 16-bit address is used, port 2 uses the additional strong pullups p1 to emit 1's for the entire external memory cycle instead of the weak ones (p2 and p3) used during normal port activity.

Figure 16 b)
Port 2 Circuitry

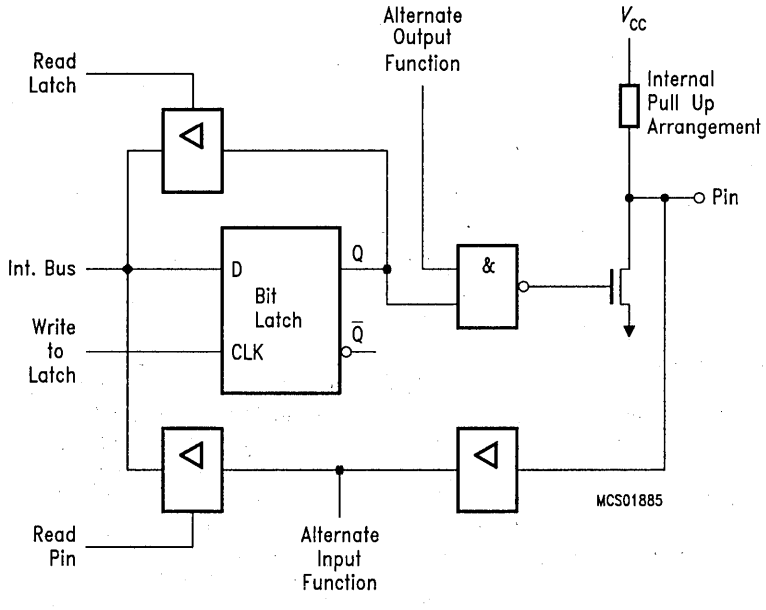


Alternate Functions

Figure 17 Port 1 and 3 shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise the pull-down FET is on and the port pin is stuck at 0. (This does not apply to ports 1.0 to 1.3 when operated in compare output mode; refer to Compare Function of Registers CRC, CC1 to CC3 on page 5-60 for details). After reset all port latches contain ones (1).

Port	Symbol	Alternate Function
P1.0	INT3#/CC0	External interrupt 3 input, compare 0 output, capture 0 input
P1.1	INT4/CC1	External interrupt 4 input, compare 1 output, capture 1 input
P1.2	INT5/CC2	External interrupt 5 input, compare 2 output, capture 2 input
P1.3	INT6/CC3	External interrupt 6 input, compare 3 output, capture 3 input
P1.4	INT2#	External interrupt 2 input
P1.5	T2EX	Timer 2 external reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external reload trigger input
P3.0	RXD	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TXD	Serial port's transmitter data output (asynchronous) or clock output (synchronous)
P3.2	INT0#	External interrupt 0 input, timer 0 gate control
P3.3	INT1#	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	WR#	External data memory write strobe
P3.7	RD#	External data memory read strobe

Figure 17
Port 1 and 3



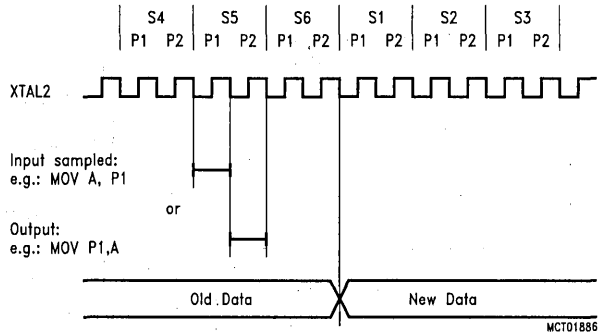
Port Handling

Port Timing

When executing an instruction that changes the value of a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are only sampled by their output buffers during phase 1 of any clock period (during phase 2 the output buffer holds the value it noticed during the previous phase 1). Consequently, the new value in the port latch will not appear at the output pin until the next phase 1, which will be at S1P1 of the next machine cycle.

When an instruction reads a value from a port pin (e.g. MOV A, P1) the port pin is actually sampled in state 5 phase 1 or phase 2 depending on port and alternate functions. Figure 18 Port Timing illustrates this port timing. It must be noted that this mechanism of sampling once per machine cycle is also used if a port pin is to detect an "edge", e.g. when used as counter input. In this case an "edge" is detected when the sampled value differs from the value that was sampled the cycle before. Therefore, there must be met certain requirements on the pulse length of signals in order to avoid signal "edges" not being detected. The minimum time period of high and low level is one machine cycle, which guarantees that this logic level is noticed by the port at least once.

Figure 18
Port Timing



Port Loading and Interfacing

The output buffers of ports 1 through 5 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can belooked up in the DC characteristics in the Data Sheet of the SAB 80(C)515. The corresponding parameters are V_{OL} and V_{OH}.

The same applies to port 0 output buffers. They do, however, require external pullups to drive floating inputs, except when being used as the address/data bus.

When used as inputs it must be noted that the ports 1 through 5 are not floating but have internal pullup transistors. The driving devices must be capable of sinking a sufficient current if a logic low level shall be applied to the port pin (the parameters I_{TL} and I_{IL} in the DC characteristics specify these currents). Port 0 as well as the input only ports 6 of the ACMOS versions have floating inputs when used for digital input.

Read-Modify-Write Feature of Ports 0 through 5

Some port-reading instructions read the latch and others read the pin (see Figure 13 Basic Structure of a Port Circuitry). The instructions reading the latch rather than the pin read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions, which are listed in Table 3 Read-Modify-Write Instructions. If the destination is a port or a port bit, these instructions read the latch rather than the pin. Note that all other instructions which can be used to read a port, exclusively read the port pin. In any case, reading from latch or pin, resp., is performed by reading the SFR P0 to P5; for example, "MOV A, P3" reads the value from port 3 pins, while "ANL P4, #0AAH" reads from the latch, modifies the value and writes it back to the latch.

Table 3
Read-Modify-Write Instructions

Instruction	Function
ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic exclusive OR; e.g. XRL P3, A
JBC	Jump if bit is set and clear bit; e.g. JBC P1.1, LABEL
CPL	Complement bit; e.g. CPL P3.0
INC	Increment byte; e.g. INC P4
DEC	Decrement byte; e.g. DEC P5
DJNZ	Decrement and jump if not zero; e.g. DJNZ P3, LABEL
MOV Px.y, C	Move carry bit to bit y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. The reason is that they read the port byte, all 8 bits, modify the addressed bit, then write the complete byte back to the latch.

The reason why read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a "1" is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor (approx. 0.7 V, i.e. a logic low level !) and interpret it as "0". For example, when modifying a port bit by a SETB or CLR instruction, another bit in this port with the above mentioned configuration might be changed if the value read from the pin were written back to the latch. However, reading the latch rather than the pin will return the correct value of "1".

Serial Interfaces

The serial port of the SAB 80(C)515S enables communication between microcontrollers or between the microcontroller and peripheral devices.

The serial port is full-duplex, meaning it can transmit and receive simultaneously. It is also receive buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register (however, if the first byte still has not been read by the time reception of the second byte is complete, the last received byte will be lost). The serial channel is completely compatible with the serial channel of the SAB 80(C)51.

Operating Modes of Serial Interface

The serial interface can operate in four modes (one synchronous mode, three asynchronous modes). The baud rate clock for this interface is derived from the oscillator frequency (mode 0, 2) or generated either by timer 1 or by a dedicated baud rate generator (mode 1, 3). A more detailed description of how to set the baud rate will follow in Baud Rates on page 5-35.

Mode 0: shift register (synchronous) mode:

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, variable baud rate:

10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB8 in special function register SCON. The baud rate is variable.

Mode 2: 9-bit UART, fixed baud rate:

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmission, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB8 or a second stop bit by setting TB8 to 1. On reception the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-bit UART, variable baud rate:

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmission, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB8 or a second stop bit by setting TB8 to 1. On reception, the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1. The serial interfaces also provide interrupt requests when a transmission or a reception of a frame has completed. The corresponding interrupt request flags for serial interface are TI or RI, resp. See Interrupt System on page 5-77 for more details about the interrupt structure. The interrupt request flags TI and RI can also be used for polling the serial interface if the serial interrupt is not to be used (i.e. serial interrupt not enabled).

The control and status bits of the serial channel 0 in special function register S0CON are illustrated in Figure 19 Special Function Register SCON (Address 98H). Figure 20 Special Function Register SBUF (Address 98H) shows the special function register S0BUF which is the data register for receive and transmit. The following table summarizes the operating modes of serial interface 0.

Figure 19 Special Function Register SCON (Address 98H)

	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

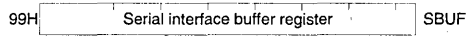
Bit		Symbol
SM0	SM1	
0	0	Serial mode 0: Shift register mode, fixed baud rate
0	1	Serial mode 1: 8-bit UART, variable baud rate
1	0	Serial mode 2: 9-bit UART, fixed baud rate
1	1	Serial mode 3: 9-bit UART, variable baud rate
SM2		Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3 and SM2 being set to 1, RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1 and SM2 = 1, RI will not be activated if a valid stop bit has not been received. In mode 0, SM2 should be 0.
REN		Receiver enable. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB8		Transmitter bit 8. Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8		Receiver bit 8. In modes 2 and 3 it is the 9th bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI		Transmitter interrupt. Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must by cleared by software.
RI		Receiver interrupt. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software.

The control and status bits of the serial channel in special function register SCON are illustrated in Figure 19 Special Function Register SCON (Address 98H). Figure 20 Special Function Register SBUF (Address 98H) shows the special function register SBUF which is the data register for receive and transmit. The following table summarizes the operating modes of the serial interface.

Table 4 Serial Interface, Mode Selection

SM0	SM1	Mode	Descriptions	Baud Rate
0	0	0	Shift register	$f_{OSC}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$f_{OSC}/64$ or $f_{OSC}/32$
1	1	3	9-bit UART	Variable

Figure 20 Special Function Register SBUF (Address 98H)



Receive and transmit buffer of serial interface. Writing to SBUF loads the transmit register and initiates transmission. Reading out SBUF accesses a physically separate receive register.

Multiprocessor Communication Feature

Modes 2 and 3 of the serial interface 0 have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB8. Then a stop bit follows. The port can be programmed such that when the stop bit is received, the serial port 0 interrupt will be activated (i.e. the request flag RI is set) only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor communications is as follows.

If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM2 again. The slaves that were not addressed leave their SM2 set and go on about their business, ignoring the incoming data bytes.

SM2 has no effect in mode 0. In mode 1 SM2 can be used to check the validity of the stop bit. If SM2 = 1 in mode 1, the receive interrupt will not be activated unless a valid stop bit is received.

Baud Rates

As already mentioned there are several possibilities to generate the baud rate clock for the serial interface depending on the mode in which it is operated.

To clarify the terminology, something should be said about the difference between "baud rate clock" and "baud rate". The serial interface requires a clock rate which is 16 times the baud rate for internal synchronization, as mentioned in the detailed description of the various operating modes in Detailed Description of the Operating Modes on page 5-38.

Therefore, the baud rate generator have to provide a "baud rate clock" to the serial interface which -there divided by 16 - results in the actual "baud rate". However, all formulas given in the following section already include the factor and calculate the final baud rate.

5

Mode 0

The baud rate in mode 0 is fixed:

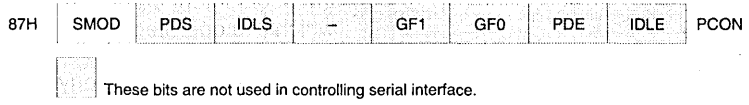
$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

Mode 2

The baud rate in mode 2 depends on the value of bit SMOD in special function register PCON (see Figure 21 Special Function Register PCON (Address 98H)). If SMOD = 0 (which is the value after reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is 1/32 of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{SMOD}}{64} * \text{oscillator frequency}$$

Figure 21 Special Function Register PCON (Address 98H)



Bit	Function
SMOD	When set, the baud rate of serial interface in modes 1, 2, 3 is doubled.

Modes 1 and 3

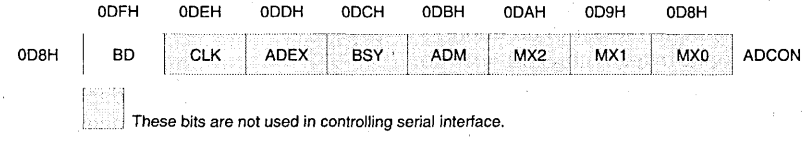
In these modes the baud rate is variable and can be generated alternatively by a dedicated baud rate generator or by timer 1.

Using the baud rate generator:

In modes 1 and 3, the SAB 80(C)515 can use the internal baud rate generator for the serial interface. To enable this feature, bit BD (bit 7 of special function register ADCON) must be set (see Figure 22 Special Function Register ADCON (Address 0D8H)). This baud rate generator divides the oscillator frequency by 2500. Bit SMOD (PCON.7) also can be used to enable a multiply by two prescaler (see Figure 21 Special Function Register PCON (Address 98H)). At 12-MHz oscillator frequency, the commonly used baud rates 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) are available. The baud rate is determined by SMOD and the oscillator frequency as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{2500} \times \text{oscillator frequency}$$

Figure 22 Special Function Register ADCON (Address 0D8H)



Bit	Function
BD	Baud rate enable. When set, the baud rate in modes 1 and 3 of serial interface is taken from a dedicated prescaler. Standard baud rates 4800 and 9600 baud at 12-MHz oscillator frequency can be achieved.

Using timer 1 to generate baud rates:

Timer 1 can be used for generating baud rates in mode 1 and 3 of the serial channel. Then the baud rate is determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{32} * (\text{Timer 1 OV-rate})$$

The timer 1 interrupt is usually disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its operating modes. In the most typical applications, it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In the case, the baud rate is given by the formula:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} * \text{oscillator frequency}}{32 * 12 \times (256 - (TH1))}$$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001B), and using the timer 1 interrupt for a 16-bit software reload.

Table 5 Timer 1 Generated Commonly Used Baud Rates lists various commonly used baud rates and shows how they can be obtained from timer 1.

Table 5 Timer 1 Generated Commonly Used Baud Rates

Baud Rate		f_{osc} (MHz)	SMOD	Timer 1		
				C/$\overline{T}$	Mode	Reload Value
Mode 1, 3:	62.5 Kbaud	12.0	1	0	2	FFH
	19.5 Kbaud	11.059	1	0	2	FDH
	9.6 Kbaud	11.059	0	0	2	FDH
	4.8 Kbaud	11.059	0	0	2	FAH
	2.4 Kbaud	11.059	0	0	2	F4H
	1.2 Kbaud	11.059	0	0	2	E8H
	110 Baud	6.0	0	0	2	72H
	110 Baud	12.0	0	0	1	FEEBH

5

Figure 23 Generation of Baud Rates for Serial Interface shows the mechanisms for baud rate generation of serial channel, while Table 6 Baud Rates of Serial Interface 0 summarizes the baud rate formulas for all usual configurations.

Figure 23
Generation of Baud Rates for Serial Interface

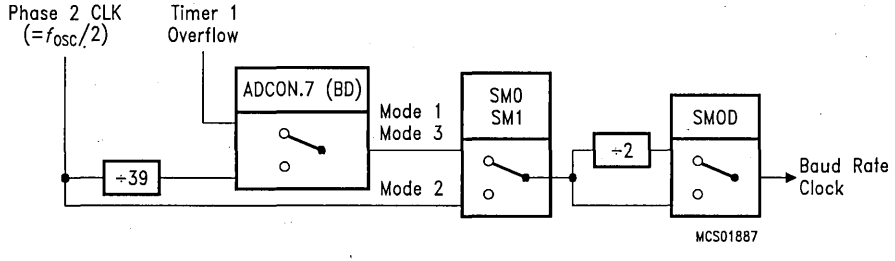


Table 6 Baud Rates of Serial Interface 0

Baud Rate Derived From	Interface Mode	Baud Rate
Timer 1 in mode 1	1, 3	$\frac{2^{SMOD}}{2} * \frac{1}{16}$ * (timer 1 overflow rate)
Timer 1 in mode 2	1, 3	$\frac{2^{SMOD}}{2} * \frac{1}{16} * \frac{f_{osc}}{12 \times (256 - (TH1))}$
Oscillator	2	$\frac{2^{SMOD}}{2} * \frac{1}{16} * \frac{f_{osc}}{2}$
Baud rate generator	1, 3	$\frac{2^{SMOD}}{2} * \frac{f_{osc}}{1250}$

Detailed Description of the Operating Modes

The following sections give a more detailed description of the several operating modes of the serial interface.

Mode 0, Synchronous Mode

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/ received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Figure 24 a) Functional Diagram - Serial Interface, Mode 0 and Figure 24 b) Timing Diagram - Serial Interface, Mode 0 show a simplified functional diagram of the serial port in mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write-to-SBUF" signal at S6P2 also loads a 1 into the 9th bit position of the transmit shift register and tells the TX control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write-to-SBUF" and activation of SEND.

SEND enables the output of the shift register to the alternate output function line P3.0, and also enables SHIFT CLOCK to the alternate output function line P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2, while the interface is transmitting. Before and after transmission SHIFT CLOCK remains high. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register is shifted one position to the right.

As data bits shift to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX control block to do one last shift and then deactivates SEND and sets TI. Both of these actions occur at S1P1 in the 10th machine cycle after "write-to-SBUF".

Reception is initiated by the condition REN = 1 and RI = 0. At S6P2 in the next machine cycle, the RX control unit writes the bits 1111 1110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 in every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted one position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 in the same machine cycle.

As data bits come in from the right, 1 s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX control block to do one last shift and load SBUF. At S1P1 in the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

Mode 1, 8-Bit UART

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception through RxD, the stop bit goes into RB8 (SCON).

The baud rate for serial interface 0 is determined by the timer 1 overflow rate or by the internal baud rate generator.

Figure 25 a) Functional Diagram - Serial Interface, Mode 1 and Figure 25 b) Timing Diagram - Serial Interface, Mode 1 show a simplified functional diagram of the serial channel in mode 1. The generation of the baud rate clock is described in Baud Rates on page 5-35.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write-to-SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX control block that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next roll-over in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the "write-to-SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit to TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just left of the MSB, and all positions to the left of that contain zero. This condition flags the TX control to do one last shift and then deactivates $\overline{\text{SEND}}$ and sets TI. This occurs at the 10th divide-by-16 rollover after "write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a reception is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollover with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16 counter states. At the 7th, 8th and 9th counter state of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come from the right, 1's shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the RX control block to do one last shift. The signal to load SBUF and RB8 and to set RI will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0

2) Either SM2 = 0 or the received stop bit = 1

If either of these two conditions is not met the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, no matter whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

Mode 2, 9-Bit UART

Mode 2 is functionally identical to mode 3 (see below). The only exception is, that in mode 2 the baud rate can be programmed to two fixed quantities: either 1/32 or 1/64 of the oscillator frequency. In mode 3 the baud rate clock is generated by timer 1, which is incremented by a rate of $f_{osc}/12$ or by the internal baud rate generator.

Mode 3, 9-Bit UART

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmission, the 9th data bit (TB8) can be assigned the value of 0 or 1. On reception the 9th data bit goes into RB8 in SCON.

The baud rate is generated by either using timer 1 or the internal baud rate generator (see Baud Rates on page 5-35).

Figure 26 a) Functional Diagram - Serial Interface, Modes 2 and 3 and Figure 26 b) Timing Diagram - Serial Interface, Modes 2 and 3 show a functional diagram of the serial interfaces in mode 2 and 3 and associated timing. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus the bit times are synchronized to the divide-by-16 counter, and not to the "write-to-SBUF" signal).

The transmission begins with the activation of $\overline{\text{SEND}}$, which puts the start bit to TxD. One bit time later, DATA is activated which enables the output bit of transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just left of the TB8, and all positions to the left of that contain zeros.

This condition flags the TX control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 11th divide-by-16 rollover after "write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled of a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FH is written to the input shift register.

At the 7th, 8th and 9th counter state of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come from the right, 1 s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which is a 9-bit register), it flags the RX control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and
2) Either SM2 = 0 or the received 9th data bit = 1

If either one of these two conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, the first 8 data bits go into SBUF. One bit time later, no matter whether the above conditions are met or not, the unit goes back to look for a 1-to-0 transition at the RxD.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

Figure 24 a)
Functional Diagram - Serial Interface, Mode 0

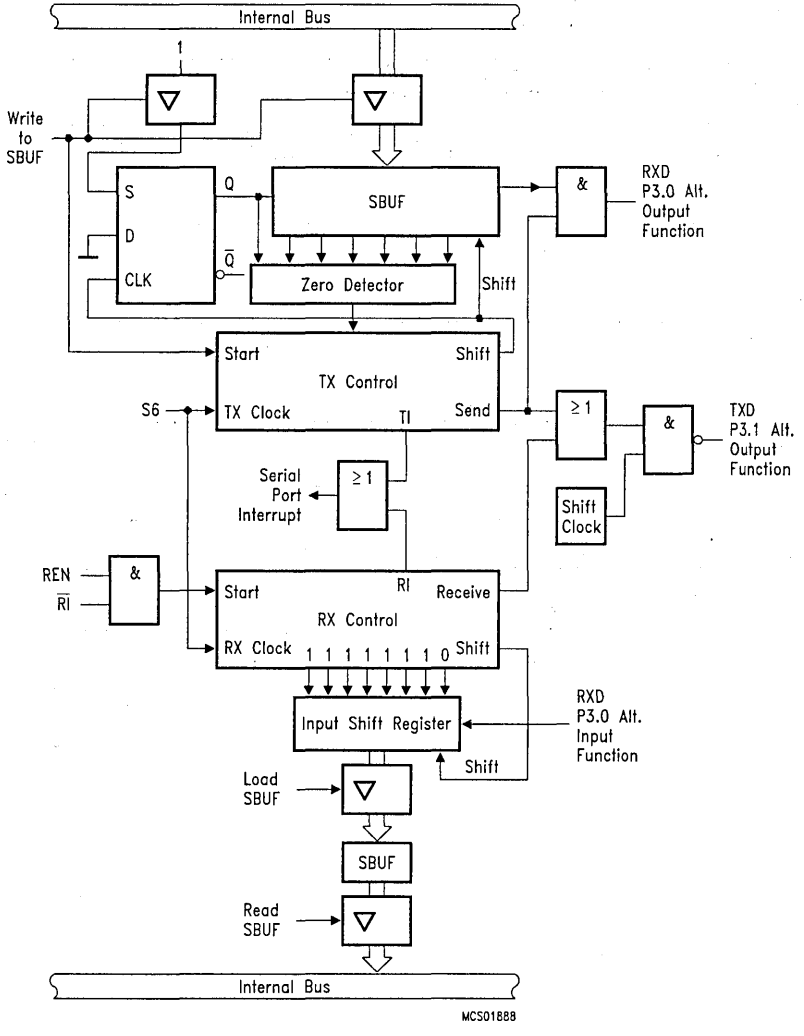


Figure 24 b)
Timing Diagram - Serial Interface, Mode 0

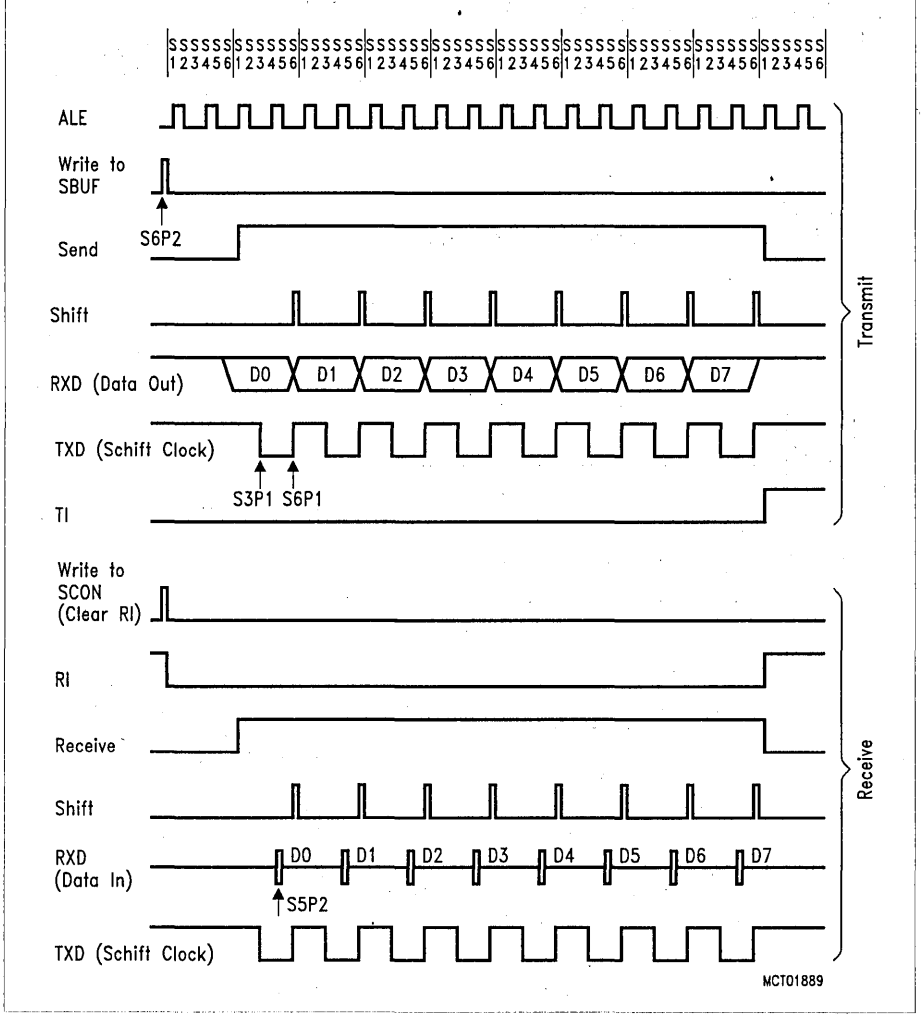


Figure 25 a)
Functional Diagram - Serial Interface, Mode 1

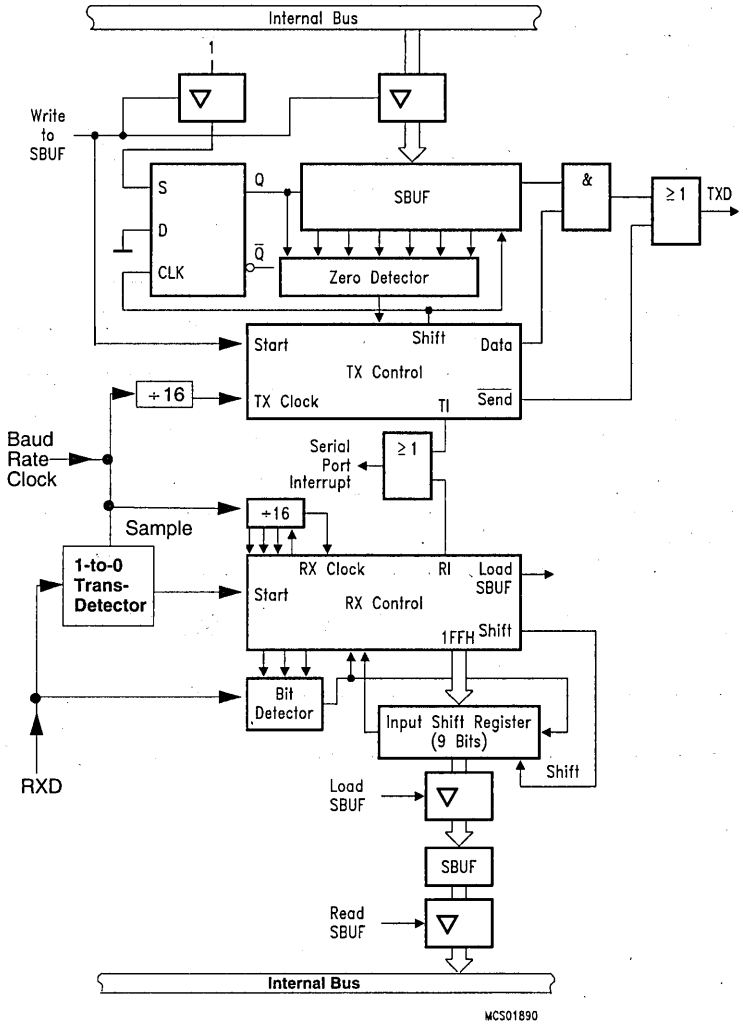


Figure 25 b)
Timing Diagram - Serial Interface, Mode 1

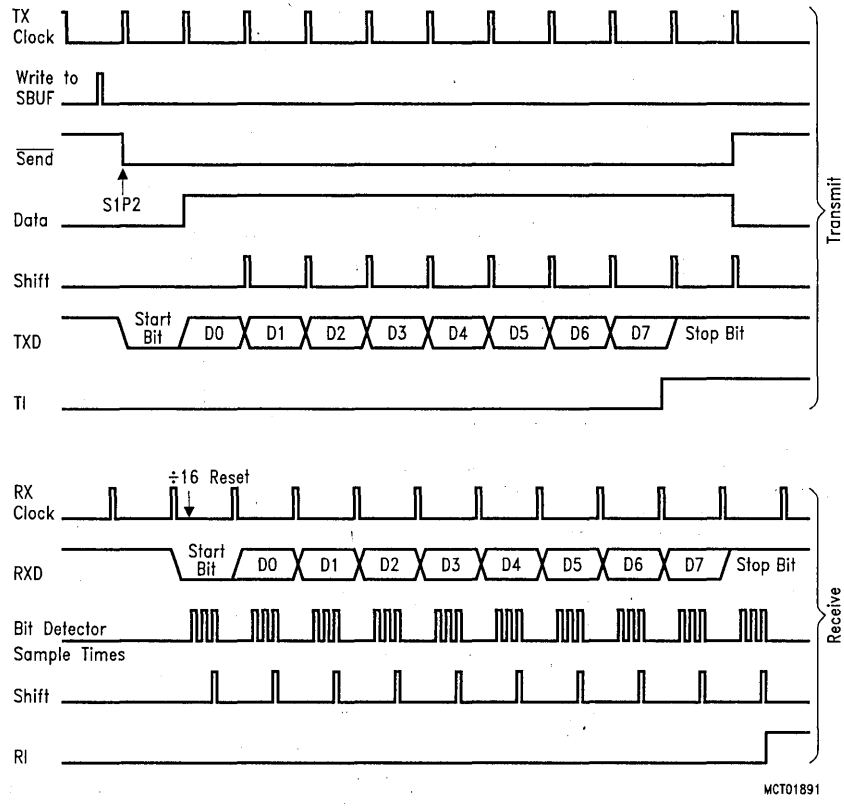


Figure 26 a)
Functional Diagram - Serial Interface, Modes 2 and 3

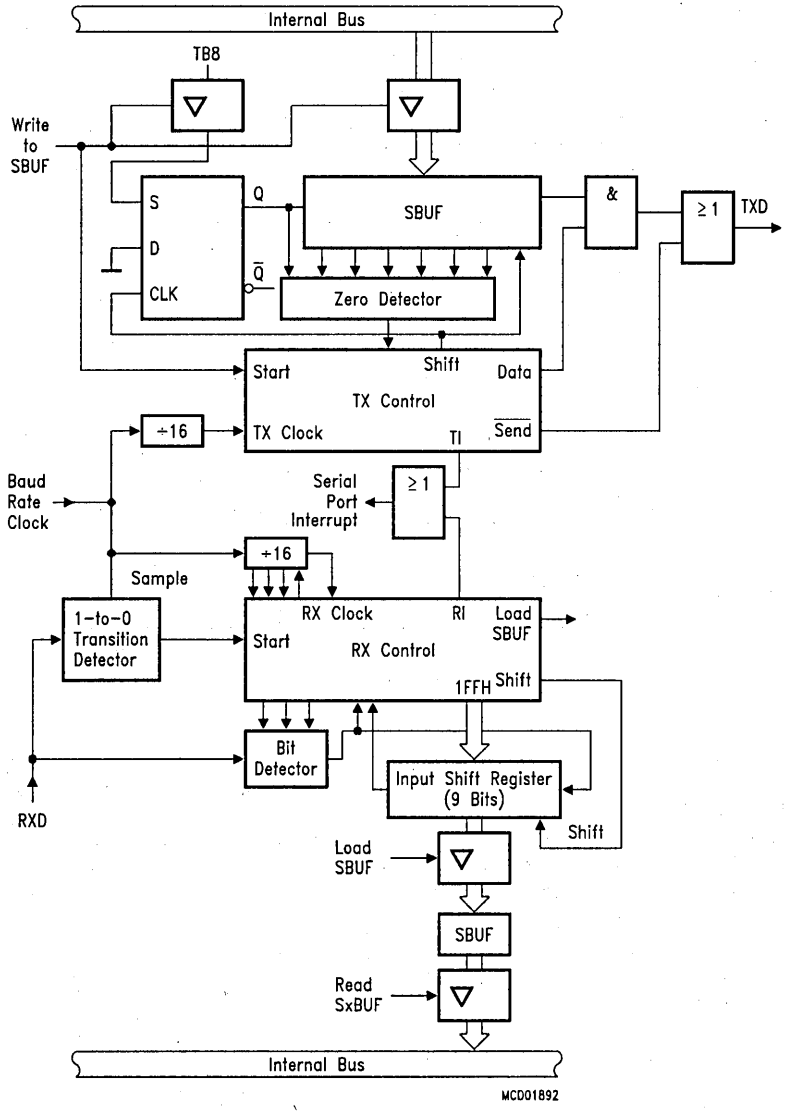
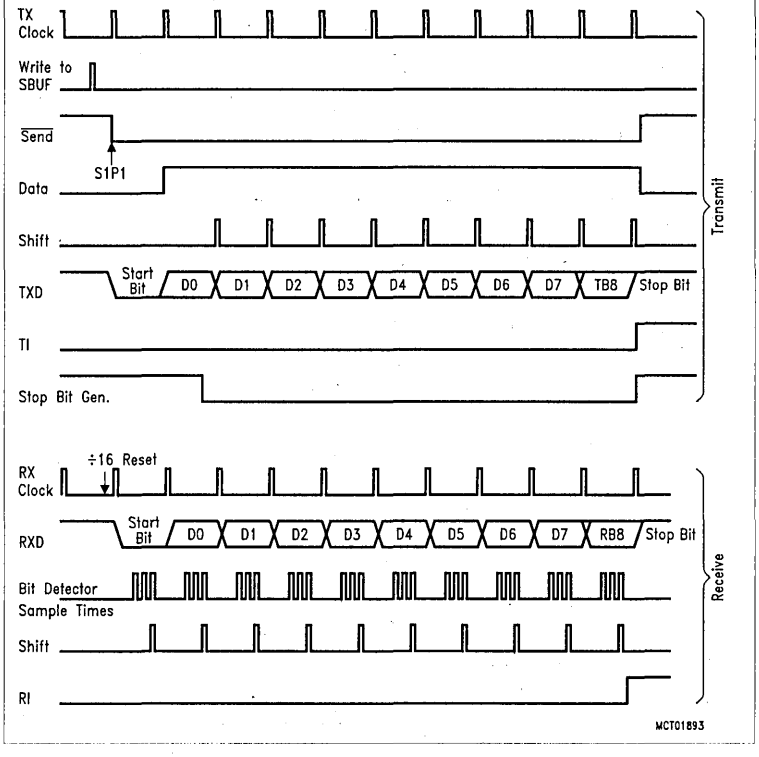


Figure 26 b)
Timing Diagram - Serial Interface, Modes 2 and 3



Timer 0 and Timer 1

The SAB 80(C)515 has three general purpose 16-bit timer/counters: timer 0, timer 1, timer 2 and the compare timer (timer 2 is discussed separately in Timer 2 with Additional Compare/ Capture/Reload on page 5-56). Timer/counter 0 and 1 are fully compatible with timer/counters 0 and 1 of the SAB 80(C)51 and can be used in the same operating modes.

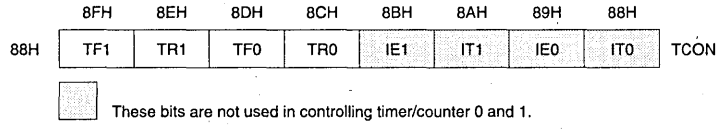
Timer/counter 0 and 1 which are discussed in this section can be configured to operate either as timers or event counters:

In "timer" function, the register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In "counter" function, the register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 and P3.5, resp.). In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

In addition to the "timer" and "counter" selection, timer/counters 0 and 1 have four operating modes from which to select.

Figure 27 Special Function Register TCON (Address 88H)



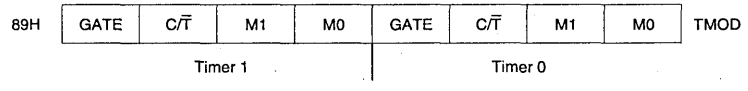
Bit	Function
TR0	Timer 0 run control bit. Set/cleared by software to turn timer/counter 0 ON/OFF.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 run control bit. Set/cleared by software to turn timer/counter 1 ON/OFF.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer/counter 0, TH1 and TL1 for timer/counter 1) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD, shown in Figure 27 Special Function Register TCON (Address 88H) and Figure 28 Special Function Register TMOD (Address 89H).

In the following descriptions the symbols TH0 and TL0 are used specify the high-byte and low-byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are described and shown for timer 0. If not explicitly noted, this applies also to timer 1.

5

Figure 28 Special Function Register TMOD (Address 89H)



Timer/counter 0/1 mode control register.

Bit		Symbol
Gate		Gating control. When set, timer/counter "x" is enabled only while "INTx" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
C/T		Counter or timer select bit. Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).
M1	M0	
0	0	8-bit timer/counter "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler.

Bit		Symbol
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH00 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops

Mode 0

Putting either timer/counter into mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. Figure 29 Timer/ Counter 0, Mode 0: 13-Bit Timer/Counter shows the mode 0 operation.

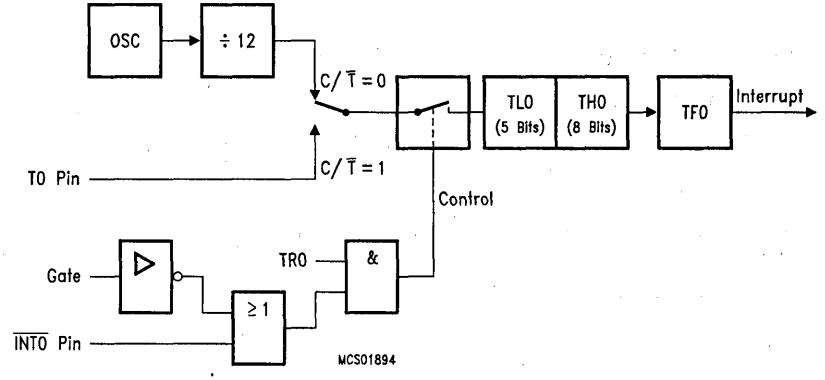
In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1's to all 0's, it sets the timer overflow flag TF0. The overflow flag TF0 then can be used to request an interrupt (see Interrupt System on page 5-77 for details about the interrupt structure).

The counted input is enabled to the timer when TR0 = 1 and either GATE = 0 or $\overline{INT0}$ = 1 (setting GATE = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements). TR0 is a control bit in the special function register TCON; GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR1, TF1, TH1, TL1, and $\overline{INT1}$ for the corresponding timer 1 signals in Figure 29 Timer/Counter 0, Mode 0: 13-Bit Timer/Counter. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

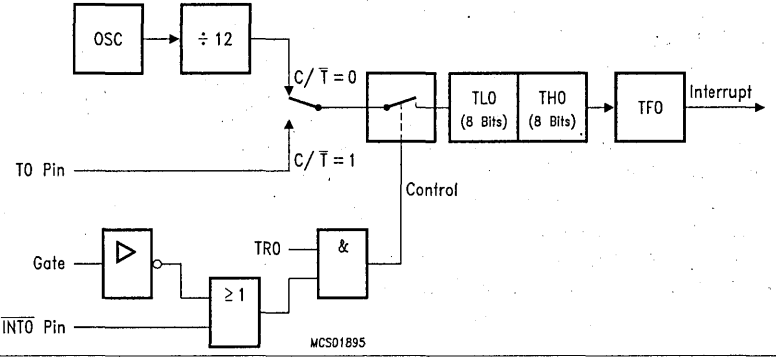
Figure 29
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter
The same applies to Timer/Counter 1



Mode 1

Mode 1 is the same as mode 0, except that the timer register is run with all 16 bits. Mode 1 is shown in Figure 30 Timer/Counter 0, Mode 0: 16-Bit Timer/Counter.

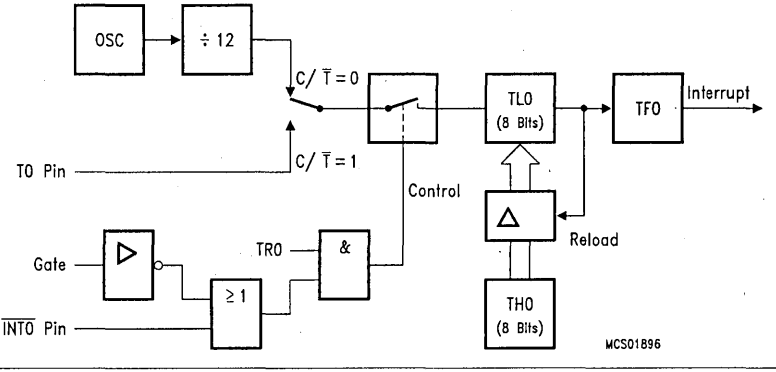
Figure 30
Timer/Counter 0, Mode 0: 16-Bit Timer/Counter
The same applies to Timer/Counter 1



Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 31 Timer/Counter 0, Mode 2: 8-Bit Timer/Counter with Auto-Reload. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

Figure 31
Timer/Counter 0, Mode 2: 8-Bit Timer/Counter with Auto-Reload
The same applies to timer/counter 1



Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on timer 0 is shown in Figure 32 Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters. TL0 uses the timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

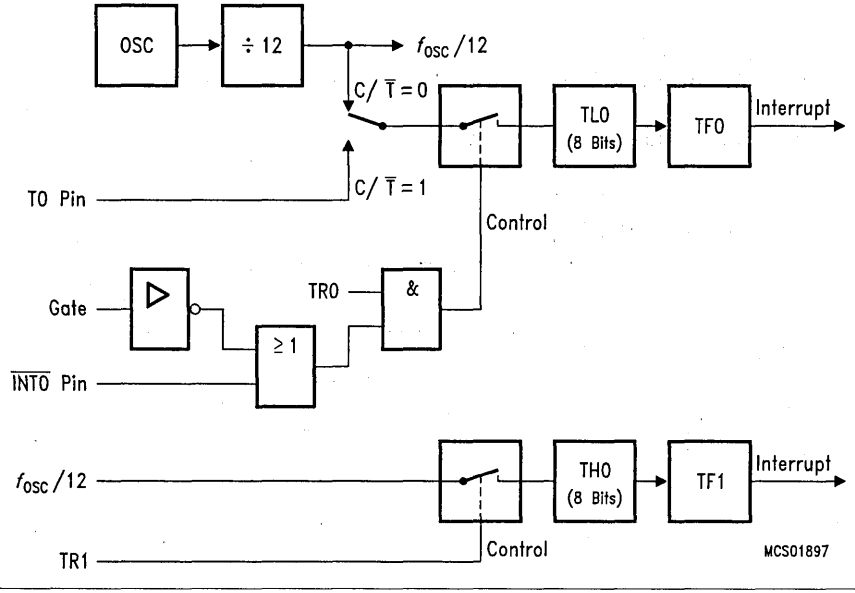
Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial channel as a baud rate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

A/D Converter

The SAB 80(C)515 provides an A/D converter with the following features:

- Eight multiplexed input channels
- The possibility of using the analog input channels (port 6) as digital inputs (ACMOS version only).
- Programmable internal reference voltages (16 steps each) via resistor array
- 8-bit resolution within the selected reference voltage range
- 13 machine cycles conversion time for ACMOS versions (including sample time)
- 15 machine cycles conversion time for MYMOS versions (including sample time)
- Internal start-of-conversion trigger
- Interrupt request generation after each conversion

Figure 32
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

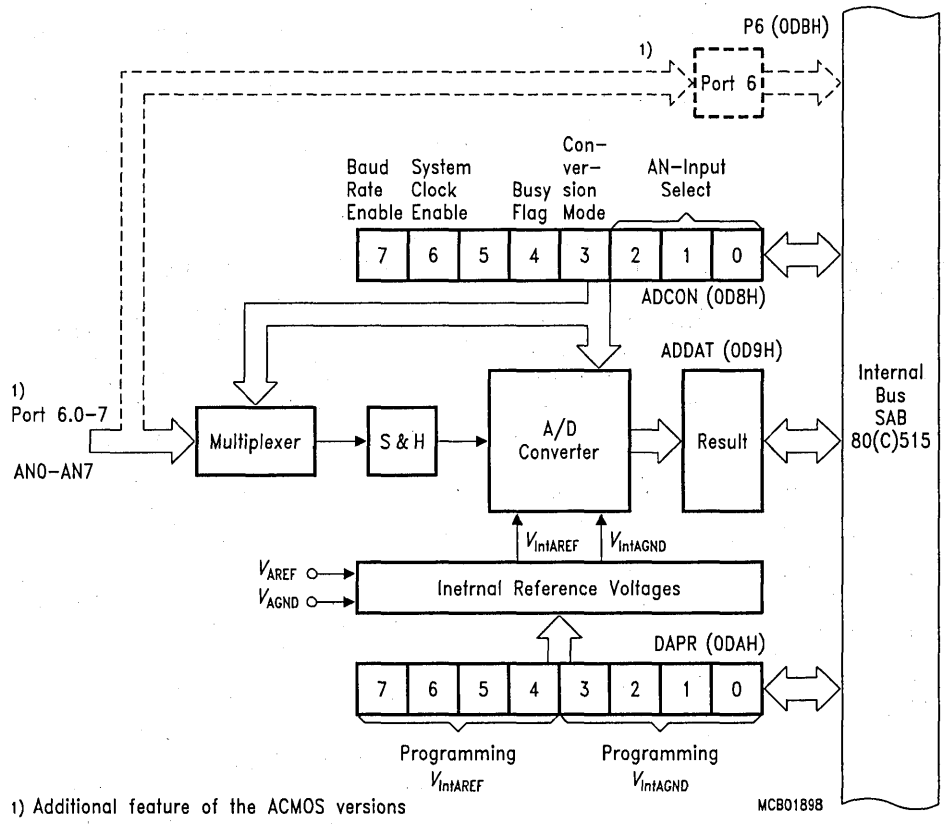


For the conversion, the method of successive approximation via capacitor array is used. The externally applied reference voltage range has to be held on a fixed value within the specifications (see section "A/D Converter Characteristics" in the data sheet). The internal reference voltages can be varied to reduce the reference voltage range of the A/D converter and thus to achieve a higher resolution.

Figure 33 A/D Converter Block Diagram shows a block diagram of the A/D converter. There are three user-accessible special function registers:

ADCON (A/D converter control register), ADDAT (A/D converter data register) and DAPR (D/A converter program register) for the programmable reference voltages.

Figure 33
A/D Converter Block Diagram

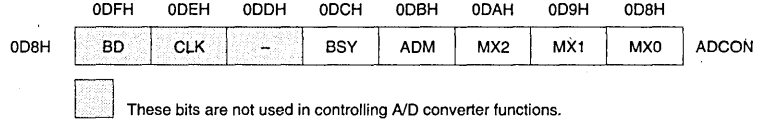


Function and Control

Initialization and Input Channel Selection

Special function register ADCON which is illustrated in Figure 34 Special Function Register ADCON (Address 0D8H) is used to set the operating modes, to check the status, and to select one of eight analog input channels.

Figure 34
Special Function Register ADCON (Address 0D8H)



Bit	Function
MX0 MX1 MX2	Select 8 input channels of the A/D converter, **see** Table 7 Selection of the Analog Input Channels
ADM	A/D conversion mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	Busy flag. This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is completed.

Register ADCON contains two mode bits. Bit ADM is used to choose the single or continuous conversion mode. In single conversion mode only one conversion is performed after starting, while in continuous conversion mode after the first start a new conversion is automatically started on completion of the previous one.

The busy flag BSY (ADCON.4) is automatically set when a conversion is in progress. After completion of the conversion it is reset by hardware. This flag can be read only, a write has no effect. There is also an interrupt request flag IADC (IRCON.0) that is set when a conversion is completed. See Interrupt System on page 5-77 for more details about the interrupt structure.

Table 7
Selection of the Analog Input Channels

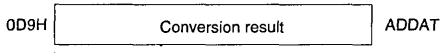
MX2	MX1	MX0	Selected Channel	Pin	
				MYMOS	ACMOS
0	0	0	Analog input 0	AN0	P6.0
0	0	1	Analog input 1	AN1	P6.1
0	1	0	Analog input 2	AN2	P6.2
0	1	1	Analog input 3	AN3	P6.3
1	0	0	Analog input 4	AN4	P6.4
1	0	1	Analog input 5	AN5	P6.5
1	1	0	Analog input 6	AN6	P6.6
1	1	1	Analog input 7	AN7	P6.7

The bits MX0 to MX2 in special function register ADCON are used for selection of the analog input channels.

Port 6 of the ACMOS versions is a dual purpose input port. If the input voltage meets the specified logic levels, it can be used as digital input as well regardless of whether the pin levels are sampled by the A/D converter at the same time.

The special function register ADDAT (Figure 35 Special Function Register ADDAT (Address 0D9H)) holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. ADDAT can be read or written under software control. If the A/D converter of the SAB 80(C)515 is not used, register ADDAT can be used as an additional general purpose register.

Figure 35
Special Function Register ADDAT (Address 0D9H)



This register contains the 8-bit conversion result.

Start of Conversion

An internal start of conversion is triggered by a write-to-DAPR instruction. The start procedure itself is independent of the value which is written to DAPR. However, the value in DAPR determines which internal reference voltages are used for the conversion (see Reference Voltages on page 5-53).

When single conversion mode is selected (ADM = 0) only one conversion is performed. In continuous mode after completion of a conversion a new conversion is triggered automatically, until bit ADM is reset.

Reference Voltages

The SAB 80(C)515 has two pins to which a reference voltage range for the on-chip A/D converter is applied (pin V_{AREF} for the upper voltage and pin V_{AGND} for the lower voltage). In contrast to conventional A/D converters it is now possible to use not only these externally applied reference voltages for the conversion but also internally generated reference voltages which are derived from the externally applied ones. For this purpose a resistor ladder provides 16 equidistant voltage levels between V_{AREF} and V_{AGND}. These steps can individually be assigned as upper and lower reference voltage for the converter itself. These internally generated reference voltages are called $V_{IntAREF}$ and $V_{IntAGND}$. The internal reference voltage programming can be thought of as a programmable "D/A converter" which provides the voltages $V_{IntARFF}$ and $V_{IntAGND}$ for the A/D converter itself.

The SFR DAPR (see Figure 36 Special Function Register DAPR) is provided for programming the internal reference voltages $V_{IntAREF}$ and $V_{IntAGND}$. For this purpose the internal reference voltages can be programmed in steps of 1/16 of the external reference voltages ($V_{AREF} - V_{AGND}$) by four bits each in register DAPR. Bits 0 to 3 specify $V_{IntAGND}$, while bits 4 to 7 specify $V_{IntAREF}$. A minimum of 1 V difference is required between the internal reference voltages V_{IntARF} and $V_{IntAGND}$ for proper operation of the A/D converter. This means, for example, in the case where V_{AREF} is 5 V and V_{AGND} is 0 V, there must be at least four steps difference between the internal reference voltages $V_{IntAREF}$ and $V_{IntAGND}$.

The values of $V_{IntAGND}$ and $V_{IntAREF}$ are given by the formulas:

$$V_{IntAGND} = V_{AGND} + \frac{DAPR\ (.3\text{-}.0)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < CH;

$$V_{IntAREF} = V_{AGND} + \frac{DAPR\ (.7\text{-}.4)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3H;

DAPR (.3-.0) is the contents of the low-order nibble, and DAPR (.7-.4) the contents of the high-order nibble of DAPR.

If DAPR (.3-.0) or DAPR (.7-.4) = 0, the internal reference voltages correspond to the external reference voltages V_{AGND} and V_{AREF}, respectively.

If $V_{AINPUT} > V_{IntAREF}$, the conversion result is 0FFH, if $V_{AINPUT} < V_{IntAGDN}$, the conversion result is 00H (V_{AINPUT} is the analog input voltage).

5

If the external reference voltages V_{AGND} = 0 V and V_{AREF} = +5 V (with respect to V_{SS} and V_{CC}) are applied, then the following internal reference voltages $V_{IntAGND}$ and $V_{IntAREF}$ shown in Table 8 Adjustable Internal Reference Voltages can be adjusted via the special function register DAPR.

Figure 36
Special Function Register DAPR
(Address DAH)

0DAH	Programming of $V_{IntAREF}$	Programming of $V_{IntAGND}$	DAPR

D/A converter program register. Each 4-bit nibble is used to program the internal reference voltages. Write-access to DAPR starts conversion.

$$V_{IntAGND} = V_{AGND} + \frac{DAPR\ (.3\text{-}.0)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < 13;

$$V_{IntAGND} = V_{AGND} + \frac{DAPR\ (.7\text{-}.4)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3;

Table 8
Adjustable Internal Reference Voltages

Step	DAPR (.3-.0) DAPR (.7-.4)	$V_{IntAGND}$	$V_{IntAREF}$
0	0000	0.0	5.0
1	0001	0.3125	–
2	0010	0.625	–
3	0011	0.9375	–
4	0100	1.25	1.25
5	0101	1.5625	1.5625
6	0110	1.875	1.875
7	0111	2.1875	2.1875
8	1000	2.5	2.5
9	1001	2.8125	2.8125
10	1010	3.125	3.125
11	1011	3.4375	3.4375
12	1100	3.75	3.75
13	1101	–	4.0625
14	1110	–	4.375
15	1111	–	4.68754

The programmability of the internal reference voltages allows adjusting the internal voltage range to the range of the external analog input voltage. It may be used to increase the resolution of the converted analog input voltage by starting a second conversion with a compressed internal reference voltage range close to the previously measured analog value. Figure 37 Adjusting the Internal Reference Voltages within Range of the External Analog Input Voltages and Figure 38 Increasing the Resolution by a Second Conversion illustrate these applications.

Figure 37
Adjusting the Internal Reference Voltages within Range of the External Analog Input Voltages

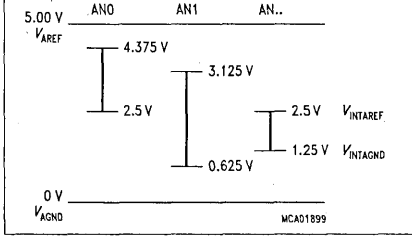
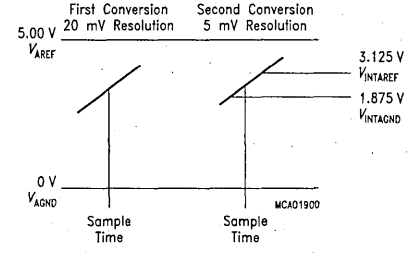


Figure 38
Increasing the Resolution by a Second Conversion



The external reference voltage supply need only be applied when the A/D converter is used, otherwise the pins V_{AREF} and V_{AGND} may be left unconnected. The reference voltage supply has to meet some requirements

concerning the level of V_{AGND} and V_{AREF} and the output impedance of the supply voltage (see also "A/D Converter Characteristics" in the data sheet).

The voltage V_{AREF} must meet the following specification:
$V_{AREF} = V_{CC}$ +/– 5%

The voltage V_{AGND} must meet a similar specification:
$V_{AGND} = V_{SS}$ + /– 0.2 V

The differential output impedance of the analog reference supply voltage should be less than 5 kOhm

If the above mentioned operating conditions are not met the accuracy of the converter may be decreased.

Furthermore, the analog input voltage V_{AINPUT} must not exceed the range from (V_{AGND} – 0.2 V) to (V_{AREF} + 0.2 V). Otherwise, a static input current might result at the corresponding analog input which will also affect the accuracy of the other input channels.

A/D Converter Timing

A conversion is started by writing into special function register DAPR. A write-to-DAPR will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle and the busy flag BSY will be set.

The conversion procedure is divided into three parts:

Load time (t_L):

During this time the analog input capacitance C_I (see data sheet) must be loaded to the analog input voltage level. The external analog source needs to be strong enough to source the current to load the analog input capacitance during the load time. This causes some restrictions for the impedance of the analog source. For a typical application the value of the impedance should be less than approx. 5 kOhms.

Sample time (t_S):

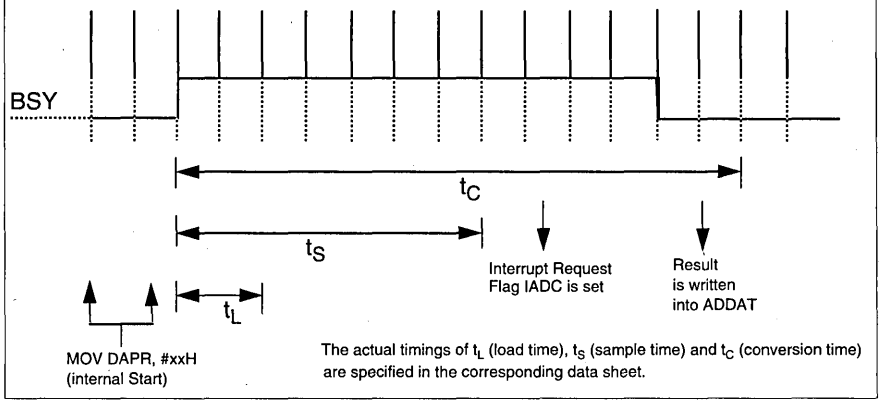
During this time the internal capacitor array is connected to the selected analog input channel. The sample time includes the load time which is described above. After the load time has passed the selected analog input must be held constant for the rest of the sample time. Otherwise the internal calibration of the comparator circuitry could be affected which might result in a reduced accuracy of the converter. However, in typical applications a voltage change of approx. 200 - 300 mV at the inputs during this time has no effect.

Conversion time (t_C):

The conversion time t_C includes the sample and load time. Thus, t_C is the total time required for one conversion. After the load time and sample time have elapsed, the conversion itself is performed during the rest of t_C. In the last machine cycle the converted result is moved to ADDAT; the busy flag (BSY) is cleared before. The A/D converter interrupt is generated by bit IADC in register IRCON. IADC is already set some cycles before the result is written to ADDAT. The flag IADC is set before the result is available in ADDAT because the shortest possible interrupt latency time is taken into account in order to ensure optimal performance. Thus, the converted result appears at the same time in ADDAT when the first instruction of the interrupt service routine is executed. Similar considerations apply to the timing of the flag BSY where usually a "JB BSY,$" instruction is used for polling.

If a continuous conversion is established, the next conversion is automatically started in the machine cycle following the last cycle of the previous conversion.

Figure 39
Timing Diagram of an A/D Converter



Timer 2 with Additional Compare/Capture/ Reload

The timer 2 with additional compare/capture/ reload features is one of the most powerful peripheral units of the SAB 80(C)515. It is used for all kinds of digital signal generation and event capturing like pulse generation, pulse width modulation, pulse width measuring etc.

This allows various automotive control applications (ignition/injection-control, anti-lock-brake ...) as well as industrial applications (DC-, three-phase AC- and stepper-motor control, frequency generation, digital-to-analog conversion, process control ...).

Please note that this timer is not equivalent to timer 2 of the SAB 80(C)52 (see Timer 2 on page 5-58).

Timer 2 in combination with the compare/ capture/reload registers allows the following modes:

Compare:	Up to 4 PWM signals with 65535 steps at maximum, and 1 µsec resolution
Capture:	Up to 4 high speed inputs with 1 µsec resolution
Reload:	Modulation of timer 2 cycle time

The block diagram in Figure 40 a) Timer 2 Block Diagram shows the general configuration of timer 2 with the additional compare/capture/reload registers.

The corresponding port functions are listed in Table 9 Alternate Port Functions of Timer 2.

Table 10 Additional Special Function Registers of Timer 2 shows the additional special function registers of timer 2.

Figure 40 a)
Timer 2 Block Diagram

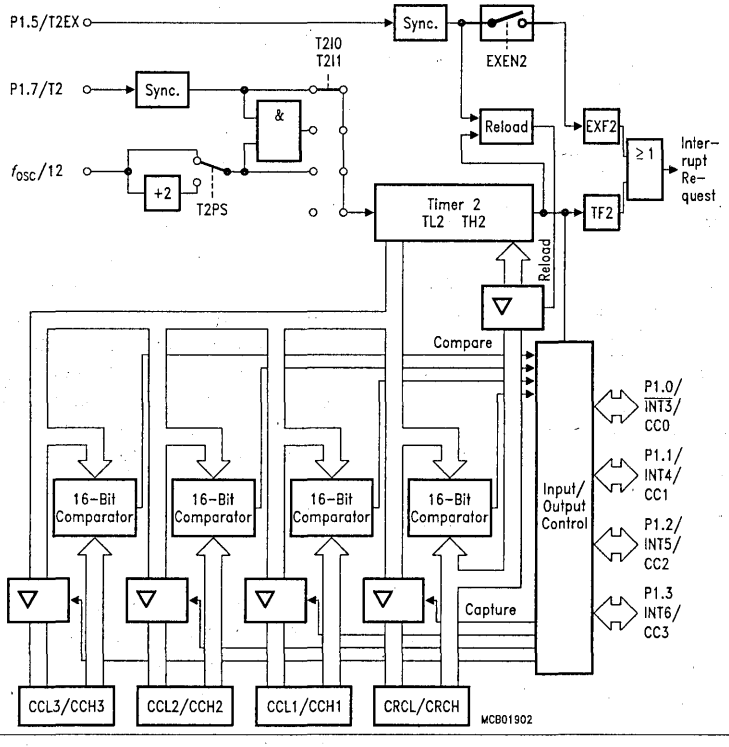


Figure 40 b)
Timer 2 in Reload Mode

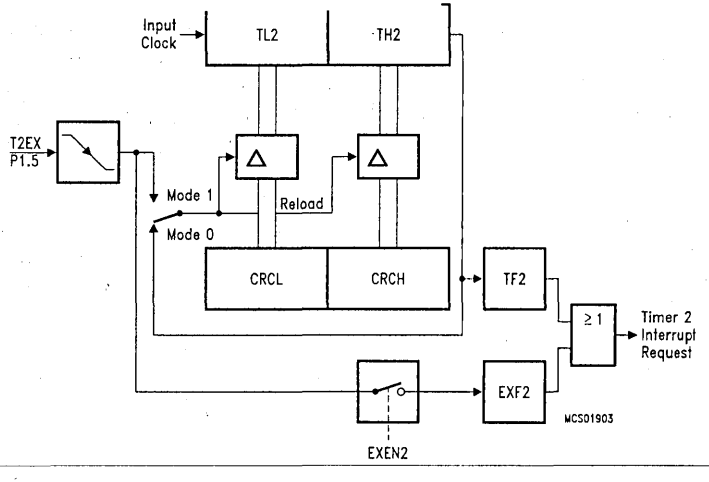


Table 9
Alternate Port Functions of Timer 2

Pin Symbol	Input (I) Output (O)	Function
P1.7/T2	I/O	External count or gate input to timer 2
P1.5/T2EX	I/O	External reload trigger input
P1.3/INT6/CC3	I/O	Comp. output/capture input for CC register 3
P1.2/INT5/CC2	I/O	Comp. output/capture input for CC register 2
P1.1/INT4/CC1	I/O	Comp. output/capture input for CC register 1
P1.0/$\overline{\text{INT3}}$/CC0	I/O	Comp. output/capture input for CRC register

Table 10
Additional Special Function Registers of Timer 2

Symbol	Description	Address
CCEN	Comp./capture enable reg.	0C1H
CCH1	Comp./capture reg. 1, high byte	0C3H
CCH2	Comp./capture reg. 2, high byte	0C5H
CCH3	Comp./capture reg. 3, high byte	0C7H
CCL1	Comp./capture reg. 1, low byte	0C2H
CCL2	Comp./capture reg. 2, low byte	0C4H
CCL3	Comp./capture reg. 3, low byte	0C6H
CRCH	Com./rel./capt. reg., high byte	0CBH
CRCL	Com./rel./capt. reg., low byte	0CAH
IRCON	Interrupt control register	0C0H
TH2	Timer 2, high byte	0CDH
TL2	Timer 2, low byte	0CCH
T2CON	Timer 2 control register	0C8H

Timer 2

The timer 2, which is a 16-bit-wide register, can operate as timer, event counter, or gated timer.

Timer Mode

In timer function, the count rate is derived from the oscillator frequency. A 2:1 prescaler offers the possibility of selecting a count rate of 1/12 or 1/24 of the oscillator frequency. Thus, the 16-bit timer register (consisting of TH2 and TL2) is either incremented in every machine cycle or in every second machine cycle. The prescaler is selected by bit T2PS in special function register T2CON (see Figure 41 Special Function Register T2CON). If T2PS is cleared, the input frequency is 1/12 of the oscillator frequency; if T2PS is set, the 2:1 prescaler gates 1/24 of the oscillator frequency to the timer.

Gated Timer Mode

In gated timer function, the external input pin T2 (P1.7) functions as a gate to the input of timer 2. If T2 is high, the internal clock input is gated to the timer. T2 = 0 stops the counting procedure. This will facilitate pulse width measurements. The external gate signal is sampled once every machine cycle (for the exact port timing, please refer to Parallel I/O on page 5-25.

Event Counter Mode

In the counter function, the timer 2 is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7). In this function, the external input is sampled every machine cycle. When the sampled inputs show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the timer register in the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle (see also Parallel I/O on page 5-25 for the exact sample time at the port pin P1.7).

Note: The prescaler must be off for proper counter operation of timer 2, i.e. T2PS must be 0.

In either case, no matter whether timer 2 is configured as timer, event counter, or gated timer, a rolling-over of the count from all 1's to all 0's sets the timer overflow flag TF2 (bit 6 in SFR IRCON, interrupt request control) which can generate an interrupt.

If TF2 is used to generate a timer overflow interrupt, the request flag must be cleared by the interrupt service routine as it could be necessary to check whether it was the TF2 flag or the external reload request flag EXF2 which requested the interrupt (for EXF2 see below). Both request flags cause the program to branch to the same vector address.

The input clock to timer 2 is selected by bits T2I0, T2I1, and T2PS as listed in Figure 43 Timer 2 with Registers CCx in Compare Mode 0 (CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6).

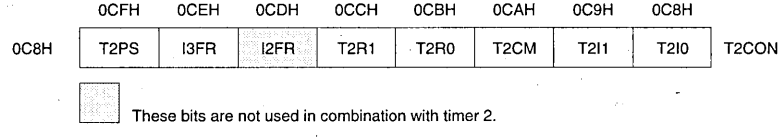
Reload of Timer 2 (see Figure 42 Port Latch in Compare Mode 0)

The reload mode for timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON as listed in Figure 43 Timer 2 with Registers CCx in Compare Mode 0 (CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6). Two reload modes are selectable:

In mode 0, when timer 2 rolls over from all 1's to all 0's, it not only sets TF2 but also causes the timer 2 registers to be loaded with the 16-bit value in the CRC register, which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H.

In mode 1, a 16-bit reload from the CRC register is caused by a negative transition at the corresponding input pin T2EX/P1.5. In addition, this transition will set flag EXF2, if bit EXEN2 in SFR IEN1 is set. If the timer 2 interrupt is enabled, setting EXF2 will generate an interrupt (more about interrupts in Interrupt System on page 5-77). The external input pin T2EX is sampled in every machine cycle. When the sampling shows a high in one cycle and a low in the next cycle, a transition will be recognized. The reload of timer 2 registers will then take place in the cycle following the one in which the transition was detected.

Figure 41 Special Function Register T2CON



Timer 2 control register. Bit-addressable register which controls timer 2 function and compare mode of registers CRC, CC1 to CC3.

Bit		Symbol
T2I1	T2I0	Timer 2 input selection
0	0	No input selected, timer 2 stops
0	1	Timer function input frequency = f_{OSC}/12 (T2PS = 0) or f_{OSC}/24 (T2PS = 1)
1	0	Counter function, external input signal at pin T2/P1.7
1	1	Gated timer function, input controlled by pin T2/P1.7
T2R1	T2R0	Timer 2 reload mode selection
0	X	Reload disabled
1	0	Mode 0: auto-reload upon timer 2 overflow (TF2)
1	1	Mode 1: reload upon falling edge at pin T2EX/P1.5

Bit	Symbol
T2CM	Compare mode bit for registers CRC, CC1 through CC3. When set, compare mode 1 is selected. T2CM = 0 selects compare mode 0.
I3FR	External interrupt 3 falling/rising edge flag, also used for capture function in combination with register CRC (see Capture Function on page 5-66). If set, capture to register to CRC (if enabled) will occur on a positive transition at pin P1.0/$\overline{INT3}$/CC0. If I3FR is cleared, capture will occur on a negative transition.
T2PS	Prescaler select bit. When set, timer 2 is clocked in the "timer" or "gated timer" function with 1/24 of the oscillator frequency. T2PS = 0 gates $f_{OSC}/12$ to timer 2. T2PS must be 0 for the counter operation of timer 2.

Compare Function of Registers CRC, CC1 to CC3

The compare function of a timer/register combination can be described as follows. The 16-bit value stored in a compare/capture register is compared with the contents of the timer register. If the count value in the timer register matches the stored value, an appropriate output signal is generated at a corresponding port pin, and an interrupt is requested.

The contents of a compare register can be regarded as 'time stamp' at which a dedicated output reacts in a predefined way (either with a positive or negative transition). Variation of this 'time stamp' somehow changes the wave of a rectangular output signal at a port pin. This may - as a variation of the duty cycle of a periodic signal - be used for pulse width modulation as well as for a continually controlled generation of any kind of square wave forms. In the case of the SAB 80(C)515, two compare modes are implemented to cover a wide range of possible applications.

The compare modes 0 and 1 are selected by bit T2CM in special function register T2CON (see Figure 42 Port Latch in Compare Mode 0). In both compare modes, the new value arrives at the port pin 1 within the same machine cycle in which the internal compare signal is activated.

The four registers CRC, CC1 to CC3 are multifunctional as they additionally provide a capture, compare or reload capability (the CRC register only, see Timer 2 on page 5-58). A general selection of the function is done in register CCEN (see Figure 48 Capture with Register CRC). Please note that the compare interrupt CC0 can be programmed to be negative or positive transition activated. The internal compare signal (not the output signal at the port pin!) is active as long as the timer 2 contents is equal to the one of the appropriate compare registers, and it has a rising and a falling edge. Thus, when using the CRC register, it can be selected whether an interrupt should be caused when the compare signal goes active or inactive, depending on bit I3FR in T2CON. For the CC registers 1 to 3 an interrupt is always requested when the compare signal goes active (see Figure 44 Function of Compare Mode 0).

Compare Mode 0

In mode 0, upon matching the timer and compare register contents, the output signal changes from low to high. It goes back to a low level on timer overflow. As long as compare mode 0 is enabled, the appropriate output pin is controlled by the timer circuit only, and not by the user. Writing to the port will have no effect. Figure 43 Timer 2 with Registers CCx in Compare Mode 0 (CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6) shows a functional diagram of a port latch in compare mode 0. The port latch is directly controlled by the two signals timer overflow and compare. The input line from the internal bus and the write-to-latch line are disconnected when compare mode 0 is enabled.

Compare mode 0 is ideal for generating pulse width modulated output signals, which in turn can be used for digital-to-analog conversion via a filter network or by the controlled device itself (e.g. the inductance of a DC or AC motor). Mode 0 may also be used for providing output clocks with initially defined period and

duty cycle. This is the mode which needs the least CPU time. Once set up, the output goes on oscillating without any CPU intervention. Figure 44 Function of Compare Mode 0 and Figure 45 Modulation Range of a PWM Signal, Generated with a Timer 2/CCx Register Combination in Compare Mode 0 illustrate the function of compare mode 0.

Figure 42
Port Latch in Compare Mode 0

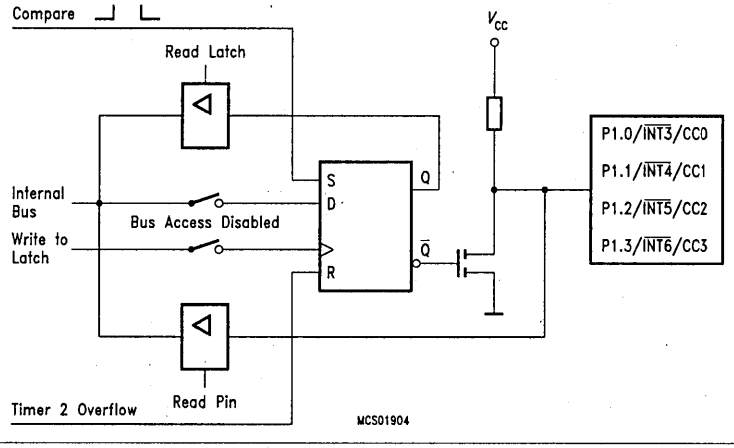


Figure 43
Timer 2 with Registers CCx in Compare Mode 0
(CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6)

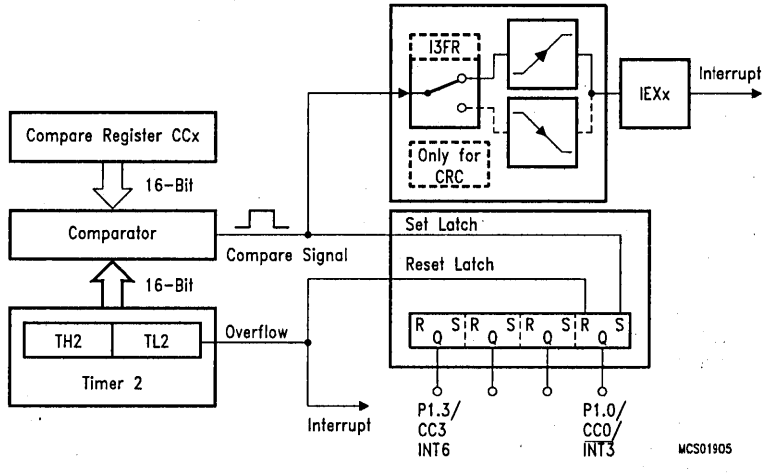
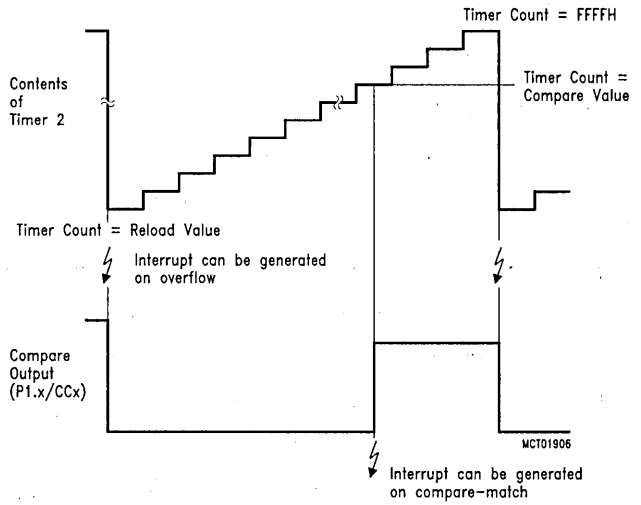


Figure 44
Function of Compare Mode 0



Modulation Range in Compare Mode 0

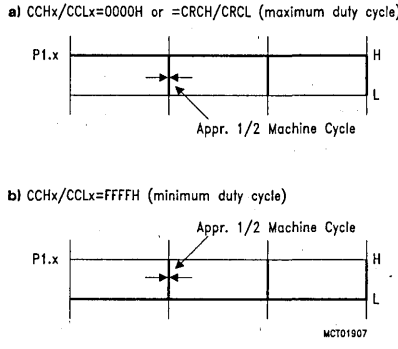
Generally it can be said that for every PWM generation in compare mode 0 with n-bit wide compare registers there are 2^n different settings for the duty cycle. Starting with a constant low level (0% duty cycle) as the first setting, the maximum possible duty cycle then would be

$(1 - 1/2^n) \times 100\%$

This means that a variation of the duty cycle from 0% to real 100% can never be reached if the compare register and timer register have the same length. There is always a spike which is as long as the timer clock period.

This "spike" may either appear when the compare register is set to the reload value (limiting the lower end of the modulation range) or it may occur at the end of a timer period. In a timer 2/CCx register configuration in compare mode 0 this spike is divided into two halves: one at the beginning when the contents of the compare register is equal to the reload value of the timer; the other half when the compare register is equal to the maximum value of the timer register (here: 0FFFFH). Please refer to Figure 46 Timer 2 with Registers CCx in Compare Mode 1 (CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6) where the maximum and minimum duty cycle of a compare output signal is illustrated. Timer 2 is incremented with the machine clock ($f_{osc}/12$), thus at 12-MHz operational frequency, these spikes are both approx. 500 ns long.

Figure 45
Modulation Range of a PWM Signal, Generated with a Timer 2/CCx Register Combination in Compare Mode 0



The following example shows how to calculate the modulation range for a PWM signal. To calculate with reasonable numbers, a reduction of the resolution to 8-bit is used. Otherwise (for the maximum resolution of 16-bit) the modulation range would be so severely limited that it would be negligible.

Example:

Timer 2 in auto-reload mode; contents of reload register CRC = 0FF00H

$$\text{Restriction of modulation range} = \frac{1}{256 \times 2} \times 100\% = 0.195\%$$

This leads to a variation of the duty cycle from 0.195% to 99.805% for a timer 2/CCx register configuration when 8 of 16 bits are used.

Compare Mode 1

In compare mode 1, the software adaptively determines the transition of the output signal. It is commonly used when output signals are not related to a constant signal period (as in a standard PWM generation) but must be controlled very precisely with high resolution and without jitter. In compare mode 1, both transitions of a signal can be controlled. Compare outputs in this mode can be regarded as high speed outputs which are independent of the CPU activity.

If mode 1 is enabled, and the software writes to the appropriate output latch at the port, the new value will not appear at the output pin until the next compare match occurs. Thus, one can choose whether the output signal is to make a new transition (1-to-0 or 0-to-1, depending on the actual pin-level) or should keep its old value at the time the timer 2 count matches the stored compare value.

Figure 47 Special Function Register CCEN shows a functional diagram of a timer/compare register/port latch configuration in compare mode 1. In this function, the port latch consists of two separate latches. The upper latch (which acts as a "shadow latch") can be written under software control, but its value will only be transferred to the output latch (and thus to the port pin) in response to a compare match.

Note that the double latch structure is transparent as long as the internal compare signal is active. While the compare signal is active, a write operation to the port will then change both latches. This may become important when driving timer 2 with a slow external clock. In this case the compare signal could be active for many machine cycles in which the CPU could unintentionally change the contents of the port latch. For details see also Using Interrupts in Combination with the Compare Function on page 5-65.

A read-modify-write instruction (see Parallel I/O on page 5-25) will read the user-controlled "shadow latch" and write the modified value back to this "shadow-latch". A standard read instruction will - as usual - read the pin of the corresponding compare output.

Figure 46
Timer 2 with Registers CCx in Compare Mode 1
(CCx stands for CRC, CC1 to CC3; IEXx stands for IEX3 to IEX6)

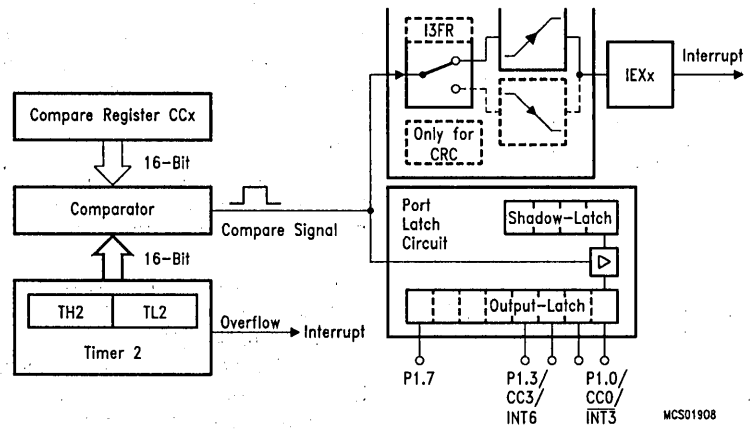


Figure 47 Special Function Register CCEN

0C1H	COCAH3	COCAL3	COCAH2	COCAI2	COCAH1	COCAL1	COCAH0	COCAL0	CCEN

Compare/capture enable register selects compare or capture function for register CRC, CC1 to CC3.

Bit		Function
COCAH0	COCAL0	Compare/capture mode for CRC register
0	0	Compare/capture disabled
0	1	Capture on falling/rising edge at pin P1.0/$\overline{INT3}$/CC0
1	0	Compare enabled
1	1	Capture on write operation into register CRCL
COCAH1	COCAL1	Compare/capture mode for CC register 1
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.1/INT4/CC1
1	0	Compare enabled
1	1	Capture on write operation into register CCL1
COCAH2	COCAL2	Compare/capture mode for CC register 2
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.2/INT5/CC2
1	0	Compare enabled
1	1	Capture on write operation into register CCL2
COCAH3	COCAL3	Compare/capture mode for CC register 3
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.3/INT6/CC3
1	0	Compare enabled
1	1	Capture on write operation into register CCL3

Using Interrupts in Combination with the Compare Function

The compare service of registers CRC, CC1, CC2 and CC3 is assigned to alternate output functions at port pins P1.0 to P1.3. Another option of these pins is that they can be used as external interrupt inputs. However, when using the port lines as compare outputs then the input line from the port pin to the interrupt system is disconnected (but the pin's level can still be read under software control). Thus, a change of the pin's level will not cause a setting of the corresponding interrupt flag. In this case, the interrupt input is directly connected to the (internal) compare signal thus providing a **compare interrupt**.

The compare interrupt can be used very effectively to change the contents of the compare registers or to determine the level of the port outputs for the next "compare match". The principle is, that the internal compare signal (generated at a match between timer count and register contents) not only manipulates the compare output but also sets the corresponding interrupt request flag. Thus, the current task of the CPU is interrupted - of course provided the priority of the compare interrupt is higher than the present task priority - and the corresponding interrupt service routine is called. This service routine then sets up all the necessary parameters for the next compare event.

Some **advantages** in using compare interrupts:

Firstly, there is no danger of unintentional overwriting a compare register before a match has been reached. This could happen when the CPU writes to the compare register without knowing about the actual timer 2 count.

Secondly, and this is the most interesting advantage of the compare feature, the output pin is exclusively controlled by hardware therefore completely independent from any service delay which in real time applications could be disastrous. The compare interrupt in turn is not sensitive to such delays since it loads the parameters for the next event. This in turn is supposed to happen after a sufficient space of time.

Please note two special cases where a program using compare interrupts could show a "surprising" behavior:

The first configuration has already been mentioned in the description of compare mode 1. The fact that the compare interrupts are transition activated becomes important when driving timer 2 with a slow external clock. In this case it should be carefully considered that the compare signal is active as long as the timer 2 count is equal to the contents of the corresponding compare register, and that the compare signal has a rising and a falling edge. Furthermore, the "shadow latches" used in compare mode 1 are transparent while the compare signal is active.

Thus, with a slow input clock for timer 2, the comparator signal is active for a long time (= high number of machine cycles) and therefore a fast interrupt controlled reload of the compare register could not only change the "shadow latch" - as probably intended - but also the output buffer.

When using the CRC , you can select whether an interrupt should be generated when the compare signal goes active or inactive, depending on the status of bit I3FR in T2CON (see Figure 66 Special Function Register TCON (Address 88H) on page 5-81).

Initializing the interrupt to be negative transition triggered is advisive in the above case. Then the compare signal is already inactive and any write access to the port latch just changes the contents of the "shadow-latch".

Please note that for CC registers 1 to 3 an interrupt is always requested when the compare signal goes active.

The second configuration which should be noted is when compare function is combined with negative transition activated interrupts. If the port latch of port P1.0 contains a 1, the interrupt request flags IEX2 will immediately be set after enabling the compare mode for the CRC register. The reason is that first the external interrupt input is controlled by the pin's level. When the compare option is enabled the interrupt logic input is switched to the internal compare signal, which carries a low level when no true comparison is detected. So the interrupt logic sees a 1-to-0 edge and sets the interrupt request flag.

An unintentional generation of an interrupt during compare initialization can be prevented if the request flag is cleared by software after the compare is activated and before the external interrupt is enabled.

Capture Function

Each of the three compare/capture registers CC1 to CC3 and the CRC register can be used to latch the current 16-bit value of the timer 2 registers TL2 and TH2. Two different modes are provided for this function. In mode 0, an external event latches the timer 2 contents to a dedicated capture register. In mode 1, a capture will occur upon writing to the low order byte of the dedicated 16-bit capture register. This mode is provided to allow the software to read the timer 2 contents "on-the-fly".

In mode 0, the external event causing a capture is for CC registers 1 to 3:

- a positive transition at pins CC1 to CC3 of port 1

For the CRC register:

- a positive or negative transition at the corresponding pin, depending on the status of the bit I3FR in SFR T2CON.

If the edge flag is cleared, a capture occurs in response to a negative transition; if the edge flag is set a capture occurs in response to a positive transition at pin P1.0/INT3/ CC0.

In both cases the appropriate port 1 pin is used as input and the port latch must be programmed to contain a one (1). The external input is sampled in every machine cycle. When the sampled input shows a low (high) level in one cycle and a high (low) in the next cycle, a transition is recognized. The timer 2 contents is latched to the appropriate capture register in the cycle following the one in which the transition was identified.

In mode 0 a transition at the external capture inputs of registers CC0 to CC3 will also set the corresponding external interrupt request flags IEX3 to IEX6. If the interrupts are enabled, an external capture signal will cause the CPU to vector to the appropriate interrupt service routine.

In mode 1 a capture occurs in response to a write instruction to the low order byte of a capture register. The write-to-register signal (e.g. write-to-CRCL) is used to initiate a capture. The value written to the dedicated capture register is irrelevant for this function. The timer 2 contents will be latched into the appropriate capture register in the cycle following the write instruction. In this mode no interrupt request will be generated.

Figure 49 Compare with Registers CC1 to CC3 and Figure 50 Reset and RAM Backup Power Timing of the SAB 80515/80535 show functional diagrams of the capture function of timer 2. Figure 49 Compare with Registers CC1 to CC3 illustrates the operation of the CRC register, while Figure Figure 50 Reset and RAM Backup Power Timing of the SAB 80515/80535 shows the operation of the compare/capture registers 1 to 3.

The two capture modes can be established individually for each capture register by bits in SFR CCEN (compare/capture enable register). That means, in contrast to the compare modes, it is possible to simultaneously select mode 0 for one capture register and mode 1 for another register. The bit positions and functions of CCEN are listed in Figure 48 Capture with Register CRC.

Figure 48
Capture with Register CRC

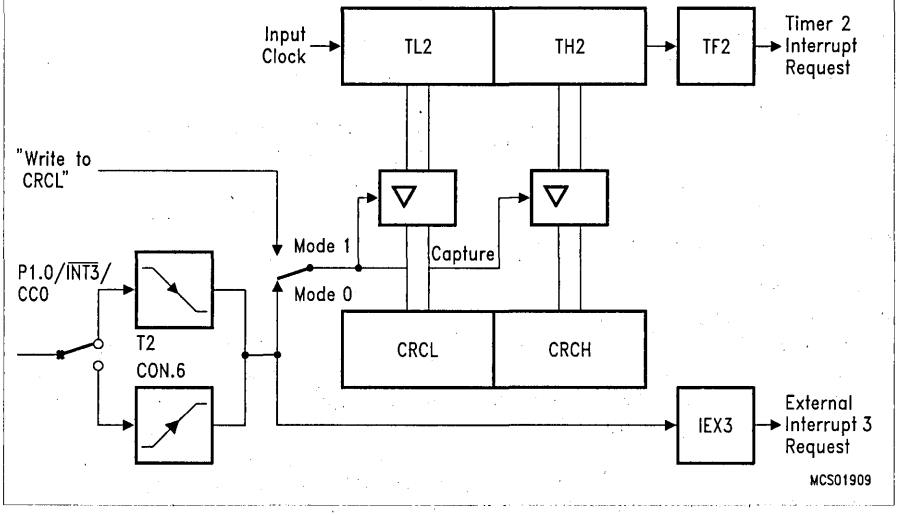
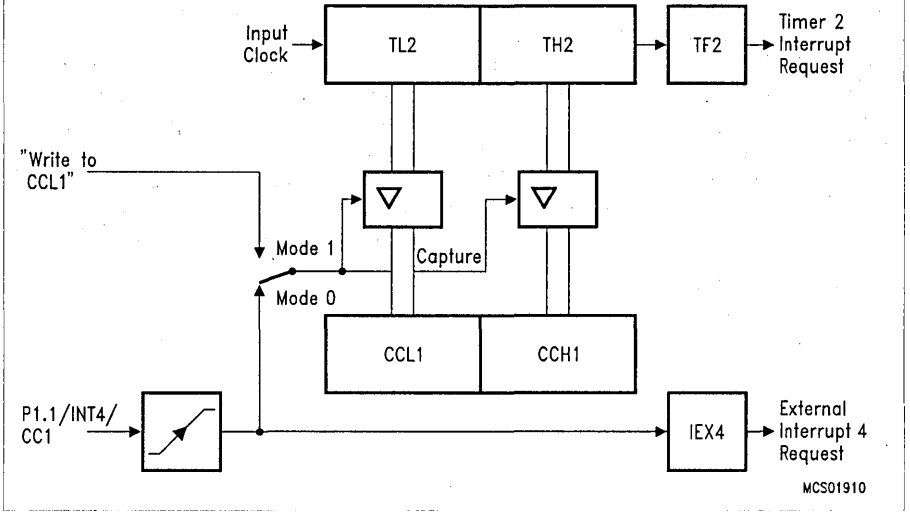


Figure 49
Compare with Registers CC1 to CC3



Power Saving Modes

For significantly reducing power consumption, the SAB 80(C)515/80(C)535 provides two Power Saving Modes:

The Power-Down Mode

Operation of the component stops completely, the oscillator is turned off. Only the internal RAM is supplied with a very low standby current.

The Idle Mode(SAB 80C515/80C535 only)

The CPU is gated off from the oscillator. All peripherals are further supplied by the oscillator clock and are able to do their jobs.

These modes are described separatly for each component in the following sections.

There are numerous applications which require high system security and at the same time reliability in electrically noisy environments. In such applications unintentional entering of the power saving modes must be absolutely avoided. A power saving mode would reduce the controller's perfomance (in case of idle mode) or even stop each operation (in case of power-down mode). This situation might be fatal for the system. Such critical applications often use the watchdog timer to prevent the system from program upsets. Then accidental entering of the power saving modes would even stop the watchdog timer and would circumvent the task of system reliability.

The SAB 80C515/80C535 provides software and hardware protection against accidental entering as described above (see Power Saving Modes of the SAB 80515/80535 on page 5-68).

Power Saving Modes of the SAB 80515/80535

The SAB 80515/80535 allows a reduction of the power consumption using the power-down mode.

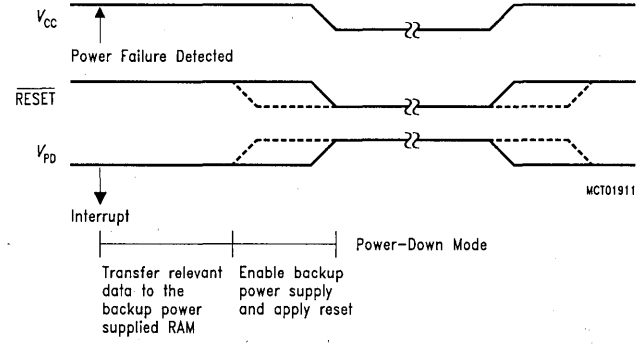
Power-Down Mode of the SAB 80515/80535

The power-down mode in the SAB 80515/80535 allows a reduction of V_{CC}, to zero while saving 40 bytes of the on-chip RAM through a backup supply connected to the V_{PD} pin. In the following, the terms V_{CC} and V_{PD} are used to specify the voltages on pin V_{CC} and pin V_{PD}, respectively.

If $V_{CC} > V_{PD}$, the 40 bytes are supplied from V_{CC}. V_{PD} may then below. If $V_{CC} < V_{PD}$, the current for the 40 bytes is drawn from V_{PD}. The addresses of these backup-powered RAM locations range from 88 to 127 (58H to 7FH). The current drawn from the backup power supply is typically 1 mA, max. 3 mA.

To utilize this feature, the user's system - upon detecting an imminent power failure - would interrupt the processor in some manner to transfer relevant data to the 40-byte on-chip RAM and enable the backup power supply to the V_{PD} pin. Then a reset should be accomplished before V_{CC} falls below its operating limit. When power returns, a power-on reset should be accomplished, and the backup supply needs to stay on long enough to resume normal operation. Figure 51 Special Function Register PCON (Address 87H) of the SAB 80C515/80C535 illustrates the timing upon a power failure.

Figure 50
Reset and RAM Backup Power Timing of the SAB 80515/80535



Power Saving Modes of the SAB 80C515/ 80C535

Differences between the Power-Down Modes of the SAB 80C515/80C535 and the SAB 80515/80535

The power-down mode of the SAB 80515/ 80535 allows retention of 40 bytes on-chip RAM through a backup supply connected to the V_{PD} pin.

The ACMOS versions SAB 80C515/80C535 have the following additional features:

- The same power supply (pin V_{CC}) for active, power-down (retention of the whole int. RAM area), and idle mode.
- An extra pin ($\overline{PE}$) that allows enabling/ disabling of the power-saving modes.
- A software protection that enables the power saving modes via special function register PCON (Power Control Register).

Hardware Enable for the Use of the Power Saving Modes

To provide power saving modes together with effective protection against unintentional entering of these modes, the SAB 80C515/ 80C535 has an extra pin for disabling the use of the power saving modes. This pin is called $\overline{PE}$ (power saving enable), and its function is as follows:

$\overline{PE}$ = 1 (logic high level)
Use of the power saving modes is not possible. The instruction sequences used for entering these modes will not affect the normal operation of the device.

$\overline{PE}$ = 0 (logic low level)
All power saving modes can be activated as described in the following sections.

When left unconnected, the pin $\overline{PE}$ is pulled to high level by a weak internal pullup. This is done to provide system protection by default.

In addition to the hardware enable/disable of the power saving modes, a double-instruction sequence which is described in the corresponding sections is necessary to enter power-down and idle mode. The combination of all these safety precautions provides a maximum of system protection.

Application Example for Switching Pin $\overline{PE}$

For most applications in noisy environments, certain components external to the chip are used to give warning of a power failure or a turn off of the power supply.

These circuits could be used to control the $\overline{PE}$ pin. The possible steps to go into power-down mode could then be as follows:

A power-fail signal forces the controller to go into a high priority interrupt routine. This interrupt routine saves the actual program status. At the same time pin $\overline{PE}$ is pulled low by the power-fail signal.

Finally the controller enters power-down mode by executing the relevant double-instruction sequence.

Power-Down Mode of the SAB 80C515/ 80C535

In the power-down mode, the on-chip oscillator is stopped. Therefore, all functions are stopped, only the contents of the on-chip RAM and the SFR's are held. The port pins controlled by their port latches output the values that are held by their SFR'S. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode; when enabled, the clockout signal (P1.6/CLKOUT) will stop at low level. ALE and $\overline{PSEN}$ are held at logic low level (see Table 11 Status of External Pins During Idle and Power-Down Mode).

If the power-down mode is to be used, the pin $\overline{PE}$ must be held low. Entering the power-down mode is done by two consecutive instructions immediately following each other. The first instruction has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6). The following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, will not initiate the power-down mode. Bit PDE and PDS will automatically be cleared after having been set and the value shown when reading one of these bits is always zero (0). Figure 51 Special Function Register PCON (Address 87H) of the SAB 80C515/80C535 shows the special function register PCON. This double- instruction sequence is implemented to minimize the chance of unintentionall entering the power-down mode, which could possibly "freeze" the chip's activity in an undesired status.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the power-down mode is composed of byte handling instructions.

The following instruction sequence may serve as an example:

```
ORL   PCON,#00000010B   ;Set bit PDE,
                        ;bit PDS must not be set
ORL   PCON,#01000000B   ;Set bit PDS,
                        ;bit PDE must not be set
```

The instruction that sets bit PDS is the last instruction executed before going into power-down mode. If idle mode and power-down mode are invoked simultaneously, the power-down mode takes precedence.

The only exit from power-down mode is a hardware reset. Reset will redefine all SFR'S, but will not change the contents of the internal RAM.

In the power-down mode, V_{CC} can be reduced to minimize power consumption. Care must be taken, however, to ensure that V_{CC} is not reduced before the power-down mode is invoked, and that V_{CC} is restored to its normal operating level before the power-down mode is terminated. The reset signal that terminates the power-down mode also frees the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

Idle Mode of the SAB 80C515/80C535

In idle mode the oscillator of the SAB 80C515/ 80C535 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial channel, the A/D converter and all timers, except for the

watchdog timer, are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption, which can be achieved by this feature, depends on the number of peripherals running. If all timers are stopped and the A/D converter and the serial interface are not running, maximum power reduction can be achieved. This state is also the test condition for the idle I_{CC} (see the DC characteristics in the data sheet).

Thus, the user has to make sure that the right peripheral continues to run or is stopped, respectively, during idle. Also, the state of all port pins - either the pins controlled by their latches or controlled by their secondary functions - depends on the status of the controller when entering idle.

Normally the port pins hold the logical state they had at the time idle was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle if the assigned function is on. This applies for the compare outputs as well as for the system clock output signal and the serial interface in case the latter could not finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ are held at logic high levels (see Table 11).

During idle, as in normal operating mode, the ports can be used as inputs. Thus, a capture or reload operation as well as an A/D conversion can be triggered, the timers can be used to count external events and external interrupts can be detected.

Table 11
Status of External Pins During Idle and Power-Down Mode

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	Power-down	Idle	Power-down
ALE	High	Low	High	Low
PSEN	High	Low	High	Low
Port 0	Data	Data	Float	Float
Port 1	Data/ alternate outputs	Data/ last output	Data/ alternate outputs	Data/ last output
Port 2	Data	Data	Address	Data
Port 3	Data/ alternate outputs	Data/ last output	Data/ alternate outputs	Data/ last output
Port 4	Data	Data	Data	Data
Port 5	Data	Data	Data	Data

The watchdog timer is the only peripheral which is automatically stopped during idle. The idle mode makes it possible to "freeze" the processor's status for a certain time or until an external event causes the controller to go back into normal operating mode. Since the watchdog timer is stopped during idle mode, this useful feature of the SAB 80C515/80C535 is provided even if the watchdog function is used simultaneously.

If the idle mode is to be used the pin $\overline{\text{PE}}$ must be held low. Entering the idle mode is to be done by two consecutive instructions immediately following each other. The first instruction has to set the flag bit IDLE (PCON.0) and must not set bit IDLS (PCON.5), the following instruction has to set the start bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). The hardware ensures that a concurrent setting of both bits, IDLE and IDLS will not initiate the idle mode. Bits IDLE and IDLS will automatically be cleared after having been set. It one of these register bits is read the value shown is zero (0). Figure 51 Special Function Register PCON (Address 87H) of the SAB 80C515/80C535 shows special function register PCON. This double-instruction sequence is implemented to minimize the chance of unintentionally entering the idle mode.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the idle mode is to be done by byte handling instructions.

The following instruction sequence may serve as an example:

```
ORL   PCON,#00000001B  ;Set bit IDLE,
                       ;bit IDLE must not be set
ORL   PCON,#00100000B  ;Set bit IDLS,
                       ;bit IDLS must not be set
```

The instruction that sets bit IDLS is the last instruction executed before going into idle mode.

Terminating the Idle Mode

The idle mode can be terminated by activation of any enabled interrupt. The CPU operation is resumed, the interrupt will be serviced and the next instruction to be executed after the RETI instruction will be the one following the instruction that set the bit IDLS.

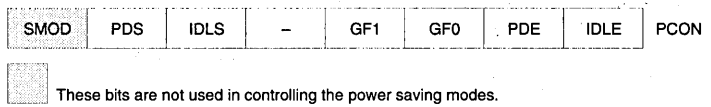
The other possibility of terminating the idle mode is a hardware reset. Since the oscillator is still running, the hardware reset is held active for only two machine cycles for a complete reset.

Watchdog Timer

As a means of graceful recovery from software or hardware upset a watchdog timer is provided in the SAB 80(C)515/80(C)535. If the software fails to clear the watchdog timer at least every 65532 µs, an internal hardware reset will be initiated. The software can be designed such that the watchdog times out if the program does not progress properly. The watchdog will also time out if the software error was due to hardware-related problems. This prevents the controller from malfunctioning for longer than 65 ms if a 12-MHz oscillator is used.

The watchdog timer is a 16-bit counter which is incremented once every machine cycle. After an external reset the watchdog timer is disabled and cleared to 0000H. The counter is started by setting bit SWDT (bit 6 in SFR IEN1). After having been started, the bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. Note that the watchdog timer cannot be stopped by software. It can only be cleared to 0000H by first setting bit WDT (IEN0.6) and with the next instruction setting SWDT. Bit WDT will automatically be cleared

Figure 51
Special Function Register PCON (Address 87H) of the SAB 80C515/80C535

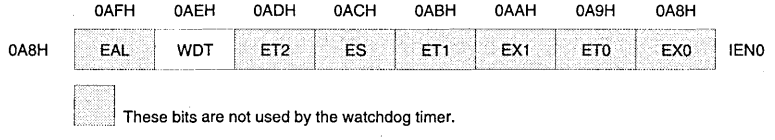


Bit	Function
PDS	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	IDLE start bit. The instruction that sets the IDSL flag bit is the last instruction before entering the idle mode.
GF1	General purpose flag
GF0	General purpose flag
PDE	Power-down enable bit. When set, starting the power-down mode is enabled.
IDLE	Idle mode enable bit. When set, starting the idle mode is enabled.

during the second machine cycle after having been set. For this reason, setting SWDT bit has to be a one cycle instruction (e.g. SETB SWDT). This double instruction clearing of the watchdog timer was implemented to minimize the chance of unintentionally clearing the watchdog. To prevent the watchdog from overflowing, it must be cleared periodically.

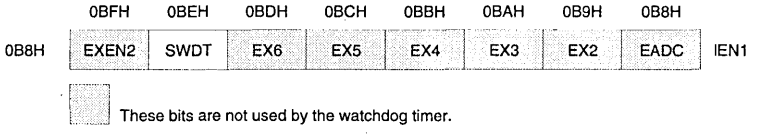
If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state FFFCH, which lasts four machine cycles. This internal reset differs from an external reset only to the extent that the watchdog timer is not disabled. Bit WDTS (was set by starting WDT) allows the software to examine from which source the reset was initiated. If it is set, the reset was caused by a watchdog timer overflow.

Figure 52 Special Function Register IEN0



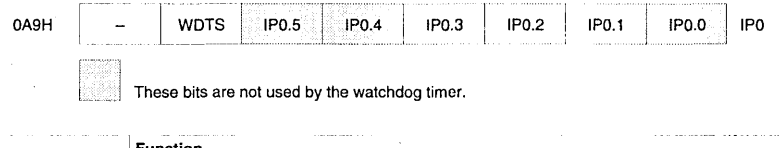
Bit	Function
WDT	Watchdog timer refresh flag. Set to initiate a refresh of the watchdog timer. Must be set directly before SWDT is set to prevent an unintentional refresh of the watchdog timer.

Figure 53 Special Function Register IEN1



Bit	Function
SWDT	Watchdog timer start/refresh flag. Set to activate/refresh the watchdog timer. When directly set after setting WDT, a watchdog timer refresh is performed.

Figure 54 Special Function Register IP0



Bit	Function
WDTS	Watchdog timer status flag. Set by hardware when the watchdog timer was started. Can be read by software.

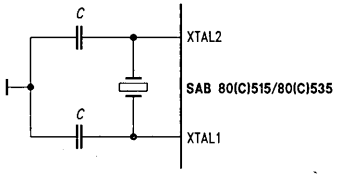
Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components as a Pierce oscillator. The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clock signals to the chip at half the oscillator frequency. These signals define the internal phases, states and machine cycles, as described in Running H/F 1 on page 5-11.

Crystal Oscillator Mode

Figure 55 Recommended Oscillator Circuit for the SAB 80(C)515/80(C)535 shows the recommended oscillator circuit.

Figure 55
Recommended Oscillator Circuit for the SAB 80(C)515/80(C)535



C = 30 pF ± 10 pF for Quartz Crystal

MCS01912

In this application the on-chip oscillator is used as a crystal-controlled, positive-reactance oscillator (a more detailed schematic is given in Figure 57 External Clock Source and Figure 59 External Clock Source). It is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. The crystal specifications and capacitances are non-critical. In this circuit 30 pF can be used as single capacitance at any frequency together with a good quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. If a ceramic resonator is used, C_1 and C_2 are normally selected to be different values. We recommend consulting the manufacturer of the ceramic resonator for value specifications of these capacitors.

Driving for External Source

The SAB 80(C)515/80(C)535 can be driven from an external oscillator.

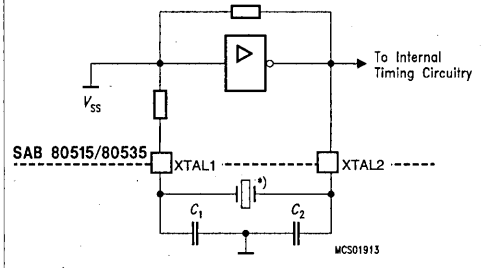
Please note that there is a difference between driving MYMOS and ACMOS devices from an external clock source.

Driving the SAB 80515/80535 from External Source

For driving the SAB 80515/80535 from an external clock source, the external clock signal is to be applied to XTAL2. A pullup resistor is recommended to increase the noise margin, but is optional if the output high level of the driving gate meets the V_{IH1} specification of XTAL2.

XTAL1 has to be connected to ground (see Figure 56 On-Chip Oscillator Circuitry).

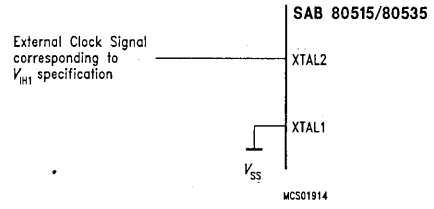
Figure 56
On-Chip Oscillator Circuitry



MCS01913

*) Quartz Crystal or Ceramic Resonator

Figure 57
External Clock Source



Driving the SAB 80C515/80C535 from External Source

For driving the SAB 80C515/80C535 from an external clock source, the external clock signal is to be applied to XTAL2, as shown in Figure 60 Special Function Register ADCON (Address 0D8H). A pullup resistor is recommended, but is optional if the output high level of the driving gate corresponds to the V_{IH1} specification of XTAL2.

XTAL1 has to be left unconnected.

Figure 58
On-Chip Oscillator Circuitry

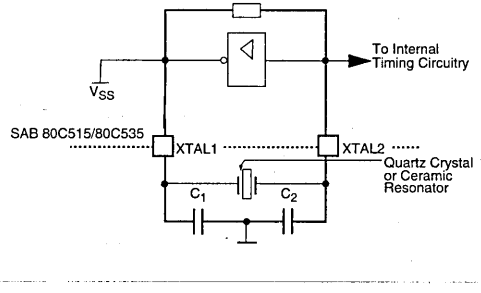
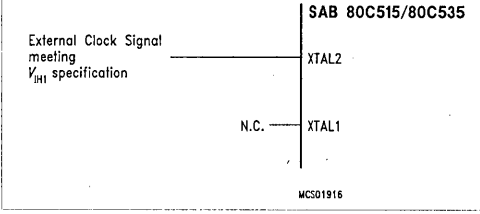


Figure 59
External Clock Source



System Clock Output

For peripheral devices requiring a system clock, the SAB 80(C)515/80(C)535 provides a clock output signal derived from the oscillator frequency as an alternate output function on pin P1.6/CLKOUT. If bit CLK is set (bit 6 of special function register ADCON, seeFigure 60 Special Function Register ADCON (Address 0D8H)), a clock signal with 1/12 of the oscillator frequency is gated to pin P1.6/ CLKOUT. To use this function the port pin must be programmed to a one (1), which is also the default after reset.

5

Figure 60
Special Function Register ADCON (Address 0D8H)

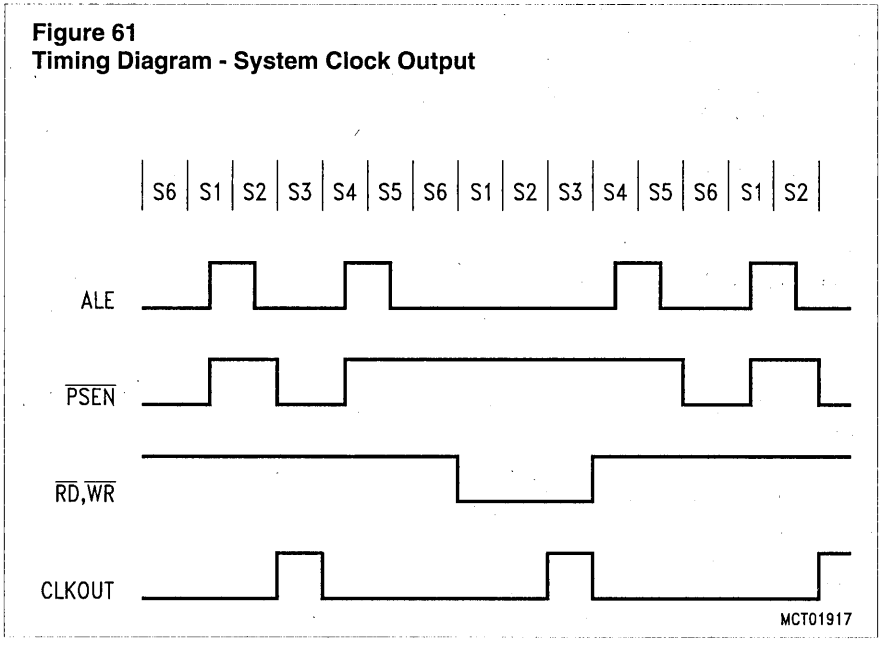
	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	ADCON

These bits are not used in controlling the clock out functions.

Bit	Function
CLK	Clockout enable bit. When set, pin P1.6/CLKOUT outputs the system clock which is 1/12 of the oscillator frequency.

The system clock is high during S3P1 and S3P2 of every machine cycle and low during all other states. Thus, the duty cycle of the clock signal is 1:6. Associated with a MOVX instruction the system clock coincides with the last state (S3) in which a $\overline{RD}$ or $\overline{WR}$ signal is active. A timing diagram of the system clock output is shown in Figure 61 Timing Diagram - System Clock Output.

Figure 61
Timing Diagram - System Clock Output



Interrupt System

The SAB 80C515/80C535 provides 12 interrupt sources with four priority levels.

Five interrupts can be generated by the on-chip peripherals (i.e. timer 0, timer 1, timer 2, compare timer, serial interface and A/D converter), and seven interrupts may be triggered externally (see Figure 64).

Interrupt Structure

A common mechanism is used to generate the various interrupts, each source having its own request flag(s) located in a special function register (e.g. TCON, IRCON, SCON). Provided that the peripheral or external source meets the condition for an interrupt, the dedicated request flag is set, whether an interrupt is enabled or not. For example, each timer 0 overflow sets the corresponding request flag TF0. If it is already set, it retains a one (1). But the interrupt is not necessarily serviced.

Now each interrupt requested by the corresponding flag can individually be enabled or disabled by the enable bits in SFR's IEN0, IEN1 (see Figures 64 and 65). This determines whether the interrupt will actually be performed. In addition, there is a global enable bit for all interrupts which, when cleared, disables all interrupts independent of their individual enable bits.

Figure 62
Interrupt Structure of the SAB 80(C)515/80(C)535

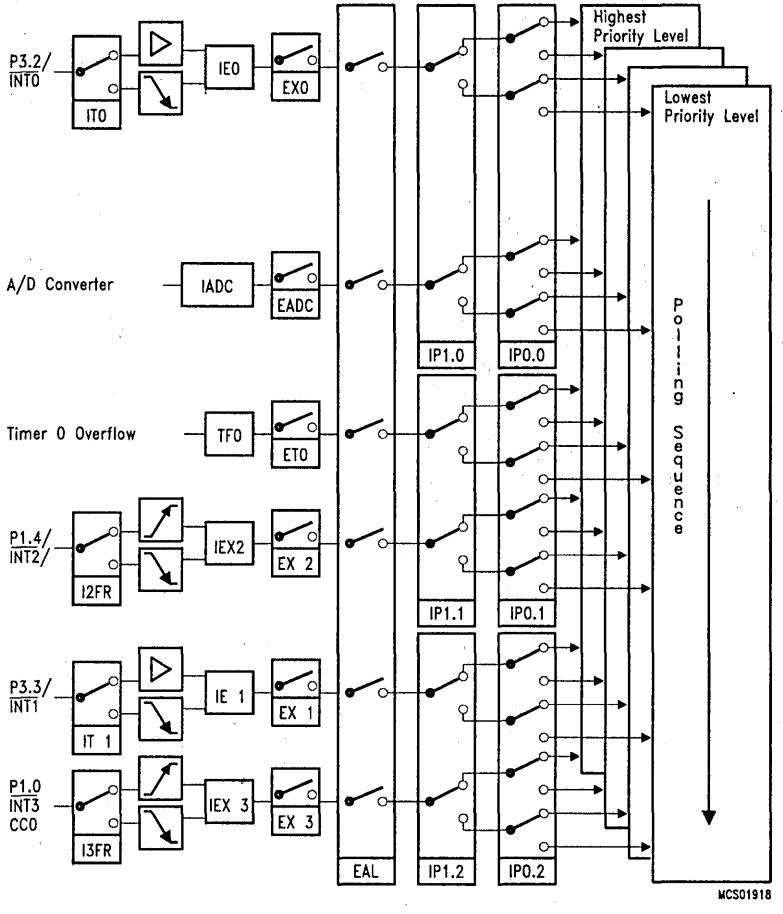
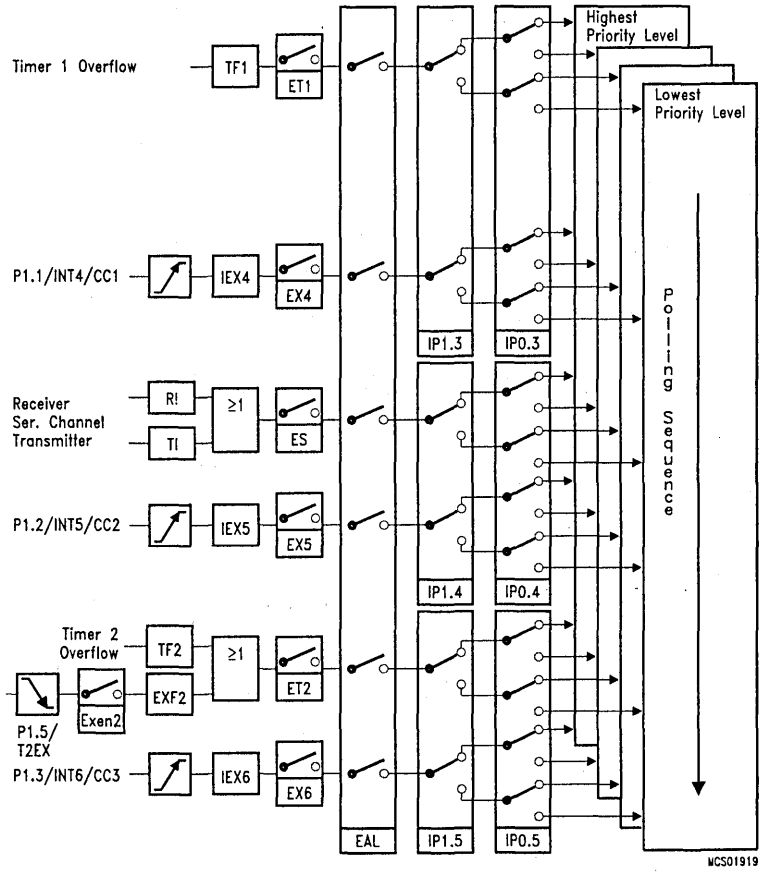
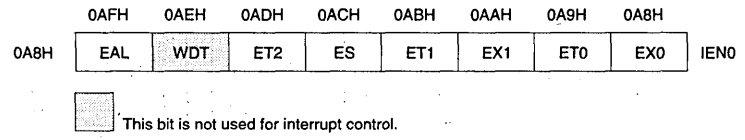


Figure 63
Interrupt Structure of the SAB 80(C)515/80(C)535 (continued)



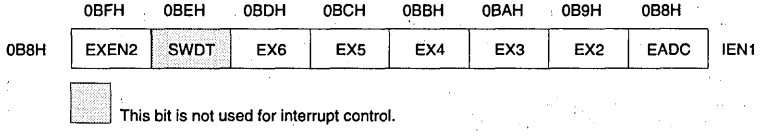
5

Figure 64
Special Function Register IEN0 (Address 0A8H)



Bit	Function
EX0	Enables or disables external interrupt 0. If EX0 = 0, external interrupt 0 is disabled.
ET0	Enables or disables the timer 0 overflow interrupt. If ET0 = 0, the timer 0 interrupt is disabled.
EX1	Enables or disables external interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ET1	Enables or disables the timer 1 overflow interrupt. If ET1 = 0, the timer 1 interrupt is disabled.
ES	Enables or disables the serial channel interrupt. If ES = 0, the serial channel interrupt is disabled.
ET2	Enables or disables the timer 2 overflow or external reload interrupt. If ET2 = 0, the timer 2 interrupt is disabled.
EAL	Enables or disables all interrupts. If EAL = 0, no interrupt will be acknowledged. If EAL = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

Figure 65
Special Function Register IEN1 (Address 0B8H)



Bit	Function
EADC	Enables or disables the A/D converter interrupt. If EADC = 0, the A/D converter interrupt is disabled.
EX2	Enables or disables external interrupt 2/capture/compare interrupt 4. If EX2 = 0, external interrupt 2 is disabled.
EX3	Enables or disables external interrupt 3/capture/compare interrupt 0. If EX3 = 0, external interrupt 3 is disabled.
EX4	Enables or disables external interrupt 4/capture/compare interrupt 0. If EX4 = 0, external interrupt 4 is disabled.
EX5	Enables or disables external interrupt 5/capture/compare interrupt 0. If EX5 = 0, external interrupt 5 is disabled.
EX6	Enables or disables external interrupt 6/capture/compare interrupt 0 If EX6 = 0, external interrupt 6 is disabled.
EXEN2	Enables or disables the timer 2 external reload interrupt. EXEN2 = 0 disables the timer 2 external reload interrupt. The external reload function is not affected by EXEN2.

In the following the interrupt sources are discussed individually.

The external interrupts 0 and 1 ($\overline{INT0}$ and $\overline{INT1}$) can each be either level-activated or negative transition-activated, depending on bits IT0 and IT1 in register TCON (see Figure 8-4). The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated this interrupt is cleared by the hardware when the service routine is vectored too, but only if the interrupt was transition-activated. If the interrupt was level-activated, then the requesting external source directly controls the request flag, rather than the on-chip hardware.

The timer 0 and timer 1 interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective timer/counter registers (exception see section 7.3.4 for timer 0 in mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored too.

The serial port interrupt is generated by a logical OR of flag RI and TI in SFR SCON (see Figure 7-7). Neither of these flags is cleared by hardware when the service routine is vectored too. In fact, the service routine will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, and the bit will have to be cleared by software.

The timer 2 interrupt is generated by the logical OR of bit TF2 in register T2CON and bit EXF2 in register IRCON. Figures 8-5 and 8-6 show SFR's T2CON and IRCON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared by software.

5

Figure 66
Special Function Register TCON (Address 88H)

	8FH	8EH	C5H	8CH	8BH	8AH	89H	88H	
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

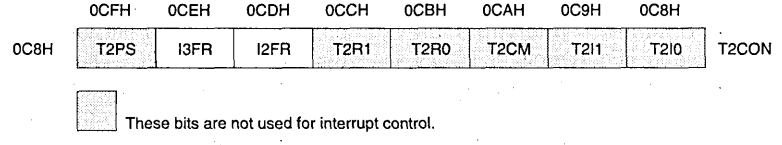
These bits are not used for interrupt control.

Bit	Function
IT0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE0	Interrupt 0 edge flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE1	Interrupt 1 edge flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

The A/D converter interrupt is generated by IADC in register IRCON (see Figure 8-6). It is set some cycles before the result is available. That is, if an interrupt is generated, in any case the converted result in ADDAT is valid on the first instruction of the interrupt service routine (with respect to the minimal interrupt response time). If continuous conversions are established, IADC is set once during each conversion. If an A/D converter interrupt is generated, flag IADC will have to be cleared by software.

The external interrupt 2 ($\overline{INT2}$/) can be either positive or negative transition-activated depending on bit I2FR in register T2CON (see Figure 8-5). The flag that actually generates this interrupt is bit IEX2 in register IRCON. If an interrupt 2 is generated, flag IEX2 is cleared by hardware when the service routine is vectored too.

Figure 67
Special Function Register TCON (Address 0C8H)



Bit	Function
I2FR	External interrupt 2 falling/rising edge flag. When set, the interrupt 2 request flag IEX2 will be set on a positive transition at pin P1.4/$\overline{INT2}$. I2FR = 0 specifies external interrupt 2 to be negative-transition activated.
I3FR	External interrupt 3 falling/rising edge flag. When set, the interrupt 3 request flag IEX3 will be set on a positive transition at pin P1.0/$\overline{INT3}$. I3FR = 0 specifies external interrupt 3 to be negative-transition active.

Like the external interrupt 2, **the external interrupt 3 ($\overline{INT3}$)** can be either positive or negative transition-activated, depending on bit I3FR in register T2CON. The flag that actually generates this interrupt is bit IEX3 in register IRCON. In addition, this flag will be set if a compare event occurs at pin P1.0/$\overline{INT3}$/CC0, regardless of the compare mode established and the transition at the respective pin. The flag IEX3 is cleared by hardware when the service routine is vectored too.

The external interrupts 4 (INT4), 5 (INT5), 6 (INT6) are positive transition-activated. The flags that actually generate these interrupts are bits IEX4, IEX5, and IEX6 in register IRCON (see Figure 8-6). In addition, these flags will be set if a compare event occurs at the corresponding output pin P1.1/INT4/CC1, P1.2/INT5/CC2, and P1.3/INT6/CC3, regardless of the compare mode established and the transition at the respective pin. When an interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored too.

Figure 68
Special Function Register IRCON (Address 0C0H)

	0C7H	0C6H	0C5H	0C4H	0C3H	0C2H	0C1H	0C0H	
0C0H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

Bit	Function
IADC	A/D converter interrupt request flag. Set by hardware at the end of a conversion. Must be cleared by software.
IEX2	External interrupt 2 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.4/$\overline{INT2}$/CC4. Cleared when interrupt processed.
IEX3	External interrupt 3 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.0/$\overline{INT3}$/CC0. Cleared when interrupt processed.
IEX4	External interrupt 4 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.1/INT4/CC1. Cleared when interrupt processed.
IEX5	External interrupt 5 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.2/INT5/CC2. Cleared when interrupt processed.
IEX6	External interrupt 6 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.3/INT6/CC3. Cleared when interrupt processed.
TF2	Timer 2 overflow flag. Set by timer 2 overflow. Must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
EXF2	Timer 2 external reload flag. Set when a reload is caused by a negative transition on pin T2EX while EXEN2 = 1. When the timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector the timer 2 interrupt routine. Can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.

All of these bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The only exceptions are the request flags IE0 and IE1. If the external interrupts 0 and 1 are programmed to be level-activated, IE0 and IE1 are controlled by the external source via pin $\overline{INT0}$ and $\overline{INT1}$, respectively. Thus, writing a one to these bits will not set the request flag IE0 and/or IE1. In this mode, interrupts 0 and 1 can only be generated by software and by writing a 0 to the corresponding pins $\overline{INT0}$(P3.2) and $\overline{INT1}$(P3.3), provided that this will not affect any peripheral circuit connected to the pins.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the special function registers IEN0 and IEN1 (Figures 8-2 and 8-3). Note that IEN0 contains also a global disable bit, EAL, which disables all interrupts at once. Also note that in the SAB 8051 the interrupt priority register IP is located at address 0B8H; in the SAB 80(C)515/80(C)535 this location is occupied by register IEN1.

Priority Level Structure

As already mentioned above, all interrupt sources are combined as pairs. Table 12 Pairs of Interrupt Sources lists the structure of the interrupt sources.

Table 12 Pairs of Interrupt Sources

External Interrupt 0	A/D Converter Interrupt
Timer 0 interrupt	External interrupt 2
External interrupt 1	External interrupt 3
Timer 1 interrupt	External interrupt 4
Serial channel 0 interrupt	External interrupt 5
Timer 2 interrupt	External interrupt 6

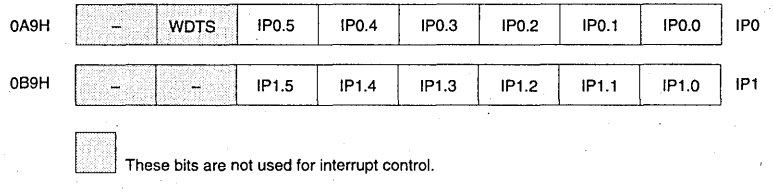
Each pair of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register IP0 and one in IP1 (Figure 69 Special Function Registers IP0 and IP1 (Address 0A9H and 0B9H)). A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another interrupt of the same or a lower priority. An interrupt of the highest priority level cannot be interrupted by another interrupt source.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is to be serviced first. Thus, within each priority level there is a second priority structure determined by the polling sequence, as follows (see Figure 70 Priority-Within-Level Structure):

Within one pair the "left" interrupt is serviced first

The pairs are serviced from top to bottom of the table.

Figure 69
Special Function Registers IP0 and IP1 (Address 0A9H and 0B9H)



Corresponding bit locations in both registers are used to set the interrupt priority level of an interrupt pair.

Bit		Function
IP1.x	IP0.x	–
0	0	Set priority level 0 (lowest)
0	1	Set priority level 1
1	0	Set priority level 2
1	1	Set priority level 3 (highest)

Bit	Function
IP1.0/IP0.0	IE0/IADC
IP1.1/IP0.1	TF0/IEX2
IP1.2/IP0.2	IE1/IEX3
IP1.3/IP0.3	TF1/IEX4
IP1.4/IP0.4	RI + TI/IEX5
IP1.5/IP0.5	TF2 + EXF2/IEX6

Figure 70
Priority-Within-Level Structure

High → Low		Priority
Interrupt source		
IE0	IADC	High
TF0	IEX2	
IE1	IEX3	↓
TF1	IEX4	
RI + TI	IEX5	
TF2 + EXF2	IEX6	Low

Note: This "priority-within-level" structure is only used to resolve simultaneous requests of the same priority level.

How Interrupts are Handled

The interrupt flags are sampled at S5P2 in each machine cycle. The sampled flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate a LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1) An interrupt of equal or higher priority is already in progress.

2) The current (polling) cycle is not in the final cycle of the instruction in progress.

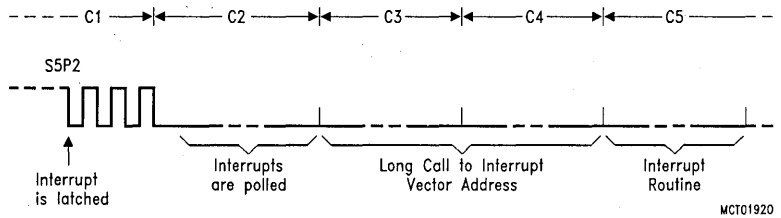
3)The instruction in progress is RETI or any write access to registers IEN0, IEN1, IEN2 or IP0 and IP1.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress is completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write access to registers IEN0, IEN1 or IP0 and IP1, then at least one more instruction will be executed before any interrupt is vectored too; this delay guarantees that changes of the interrupt status can be observed by the CPU.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if any interrupt flag is active but not being responded to for one of the conditions already mentioned, or if the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle interrogates only the pending interrupt requests.

The polling cycle/LCALL sequence is illustrated in Figure 71 Interrupt Response Timing Diagram.

Figure 71
Interrupt Response Timing Diagram



Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in Figure 71 Interrupt Response Timing Diagram, then, in accordance with the above rules, it will be vectored too during C5 and C6 without any instruction for the lower priority routine to be executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not; then this has to be done by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored too, as shown in the following (Table 13 Interrupt Sources and Vectors).

Table 13
Interrupt Sources and Vectors

Interrupt Request Flags	Interrupt Vector Address	Interrupt Source
IE0	0003H	External interrupt 0
TF0	000BH	Timer overflow
IE1	0013H	External interrupt 1
TF1	001BH	Timer 1 overflow
RI/TI	0023H	Serial channel
TF2/EXF2	002BH	Timer 2 overflow/ext. reload
IADC	0043H	A/D converter
IEX2	004BH	External interrupt 2
IEX3	0053H	External interrupt 3
IEX4	005BH	External interrupt 4
IEX5	0063H	External interrupt 5
IEX6	006BH	External interrupt 6

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the two top bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt priority level. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress. In this case no interrupt of the same or lower priority level would be acknowledged.

External Interrupts

The external interrupts 0 and 1 can be programmed to be level-activated or negative-transition activated by setting or clearing bit IT0 or IT1, respectively, in register TCON (see Figure 67 Special Function Register TCON (Address 0C8H)). If ITx = 0 (x = 0 or 1), external interrupt x is triggered by a detected low level at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is negative edge-triggered. In this mode, if successive samples of the $\overline{INTx}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

If the external interrupt 0 or 1 is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

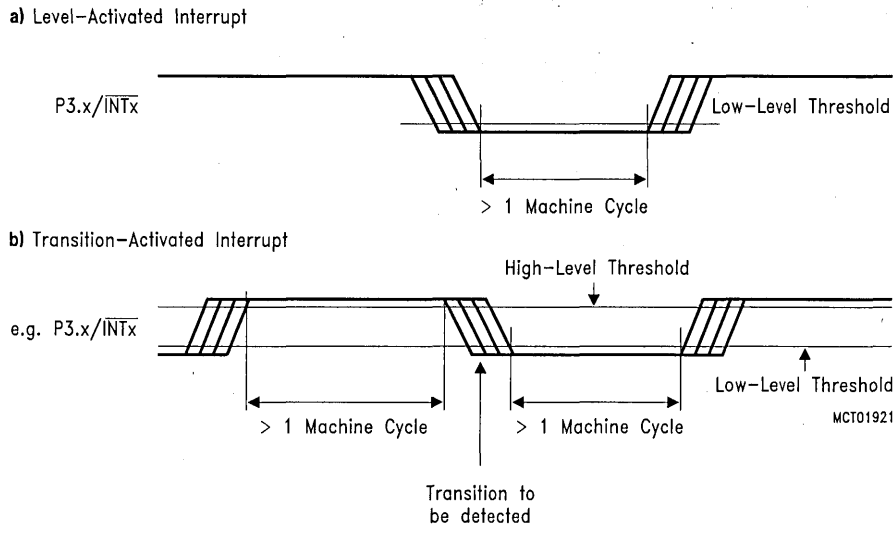
The external interrupts 2 and 3 can be programmed to be negative or positive transition-activated by setting or clearing bit I2FR or I3FR in register T2CON (see Figure 68 Special Function Register IRCON (Address 0C0H)). If IxFR = 0 (x = 2 or 3), external interrupt x is negative transition-activated. If IxFR = 1, external interrupt is triggered by a positive transition.

The external interrupts 4, 5, and 6 are activated by a positive transition. The external timer 2 reload trigger interrupt request flag EXF2 will be activated by a negative transition at pin P1.5/T2EX but only if bit EXEN2 is set.

Since the external interrupt pins ($\overline{INT2}$ to INT6) are sampled once in each machine cycle, an input high or low should be held for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin low (high for $\overline{INT2}$ and $\overline{INT3}$, if it is programmed to be negative transition-active) for at least one cycle, and then hold it high (low) for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set (see Figure 72 External Interrupt Detection). The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

5

Figure 72
External Interrupt Detection



Response Time

If an external interrupt is recognized, its corresponding request flag is set at S5P2 in every machine cycle. The value is not polled by the circuitry until the next machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus a minimum of three complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

A longer response time would be obtained if the request was blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of 'the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or a write access to registers IEN0, IEN1 or IP0, IP1, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction, if the instruction is MUL or DIV).

Thus, in a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

Microcomputer Components

SAB 80C517/80C537
8-Bit CMOS Single-Chip Microcontroller

User's Manual 11.92

Introduction

The SAB 80C517/80C537 is a high-end microcontroller in the Siemens SAB 8051 8-bit microcontroller family. It is based on the well-known industry standard 8051 architecture; a great number of enhancements and new peripheral features extend its capabilities to meet the extensive requirements of new applications. Nevertheless, the SAB 80C517 maintains compatibility within the Siemens SAB 8051 family; in fact, the SAB 80C517 is a superset of the Siemens SAB 80C515/80C535 microcontroller thus offering an easy upgrade path for SAB 80(C)515/80(C)535 users.

In addition to all features of the SAB 80C515, there are several enhancements for higher performance. The SAB 80C517 has been expanded e.g. in its arithmetic characteristics, fail save mechanisms, analog signal processing facilities and timer capabilities.

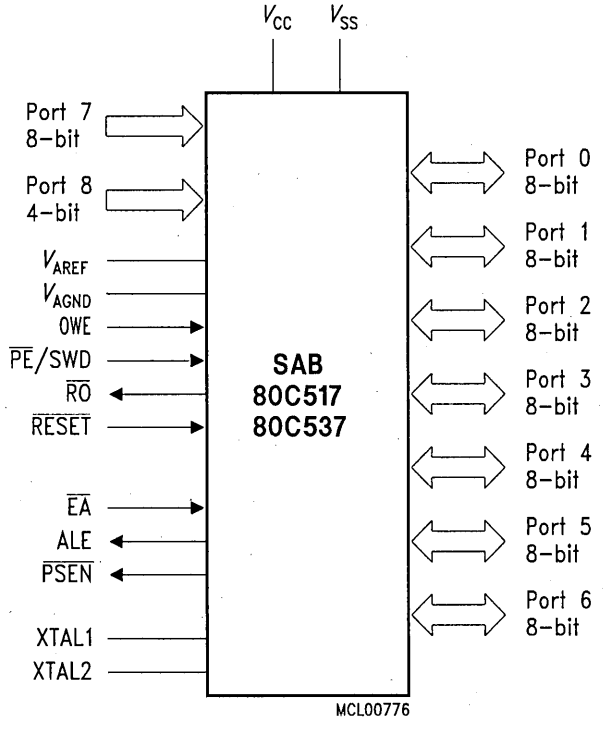
Listed below is a summary of the main features of the SAB 80C517/80C537:

- 8 Kbyte on-chip program memory (SAB 80C517 only)
- ROMless version also available (SAB 80C537)
- Full compatibility with SAB 80C515/ 80C535
- 256 byte on-chip RAM
- 256 directly addressable bits
- 1 microsecond instruction cycle at 12-MHz oscillator frequency
- 64 of 111 instructions are executed in one instruction cycle
- External program and data memory expandable up to 64 Kbyte each
- 8-bit A/D converter
  - –2 multiplexed inputs
  - –Programmable reference voltages
  - –External/internal start of conversion
- Two 16-bit timers/counters (8051 compatible)
- Powerful compare/capture unit (CCU) based on a 16-bit timer/counter and a high-speed 16-bit timer for fast compare functions
  - –One 16-bit reload/compare/capture register
  - –Four 16-bit compare/capture registers, one of which serves up to nine compare channels (concurrent compare)
- Eight fast 16-bit compare registers
- Arithmetic unit for division, multiplication, shift and normalize operations
- Eight datapointers instead of one for indirect addressing of program and external data memory
- Extended watchdog facilities
  - –16-bit programmable watchdog timer
  - –Oscillator watchdog
- Nine ports
  - –Seven bidirectional 8-bit ports
  - –One 8-bit and one 4-bit input port for analog and digital input signals
- Two full-duplex serial interfaces with own baud rate generators
- Four priority level interrupt systems, 14 interrupt vectors
- Three power saving modes
  - –Slow-down mode
  - –Idle mode
  - –Power-down mode
- Siemens high-performance ACMOS technology
- P-LCC-84 package

The ROMless version SAB 80C537 is identical with the SAB 80C517 except for the fact that it lacks the on-chip program memory; the SAB 80C537 is designed for applications with external program memory.

In this manual, any reference made to the SAB 80C517 applies to both versions, the SAB 80C517 and the SAB 80C537, unless otherwise noted.

Figure 1
Logic Symbol



Fundamental Structure

The SAB 80C517 is a totally 8051-compatible microcontroller while its peripheral performance has been increased significantly. It includes the complete SAB 80(C)515, providing 100% upward compatibility. This means that all existing 80515 programs or user's program libraries can be used further on without restriction and may be easily extended to the new SAB 80C517.

The SAB 80C517 is in the Siemens line of highly integrated microcontrollers for control applications. Some of the various on-chip peripherals have been added to support the 8-bit core in case of stringent real-time requirements. The 32-bit/16-bit arithmetic unit, the improved 4-level interrupt structure and the increased number of eight 16-bit datapointers are meant to give such a CPU support. But strict compatibility to the 8051 architecture is a principle of the SAB 80C517's design.

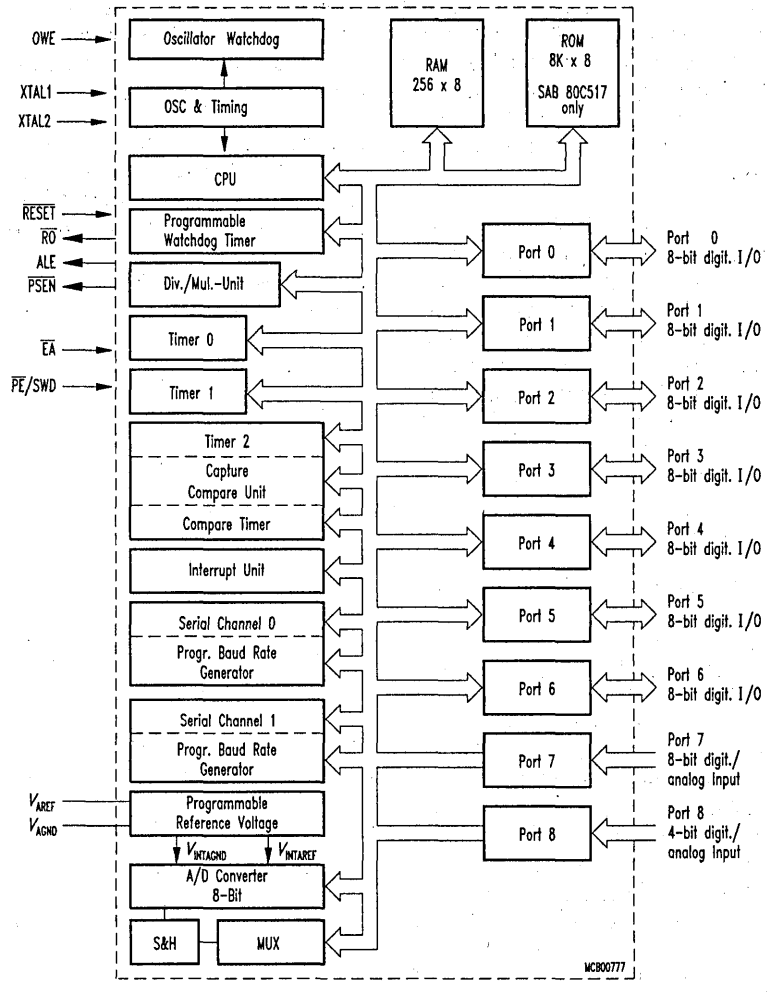
Furthermore, the SAB 80C517 contains three additional 8-bit I/O ports and twelve general input lines. The additional serial channel is compatible to an 8051-UART and provided with an independent and freely programmable baud rate generator. An 8-bit resolution A/D-converter with software-adjustable reference voltages has been integrated to allow analog signal processing. As a counterpart to the A/D converter, the SAB 80C517 includes a powerful compare/capture unit with two 16-bit timers for all kinds of digital signal processing. The controller has been completed with well considered provisions for "fail-safe" reaction in critical applications and offers all CMOS features like low power consumption as well as an idle, power-down and slow-down mode.

Readers who are familiar with the SAB 8051 or SAB 80515 may concentrate on chapters 6 and 7 where the reset conditions and the new peripheral components are described. The chapter on Interrupt System has a special section for 80515 professionals where enhancements of the interrupt structure compared to the SAB 80515 are summarized.

For readers, however, who are newcomers to the 8051 family of microcontrollers, the following section may give a general view of the basic characteristics of the SAB 80C517.

The details of operation are described in later chapters.

Figure 2
Functional Block Diagram



Central Processing Unit

The CPU is designed to operate on bits and bytes. The instructions, which consist of up to 3 bytes, are performed in one, two or four machine cycles. One machine cycle requires twelve oscillator cycles. The instruction set has extensive facilities for data transfer, logic and arithmetic instructions. The Boolean processor has its own full-featured and bit-based instructions within the instruction set. The SAB 80C517 uses five addressing modes: direct access, immediate, register, register indirect access, and for accessing the external data or program memory portions a base register plus index-register indirect addressing.

Memory Organization

The SAB 80C517 has an internal ROM of 8 Kbyte. The program memory can externally be expanded up to 64 Kbyte (see Bus Expansion Control). The internal RAM consists of 256 bytes. Within this address space there are 128 bit-addressable locations and four register banks, each with 8 general purpose registers. In addition to the internal RAM there is a further 128-byte address space for the special function registers, which are described in sections to follow.

Because of its Harvard architecture, the SAB 80C517 distinguishes between an external program memory portion (as mentioned above) and up to 64 Kbyte external data memory accessed by a set of special instructions. As an important improvement of the 8051 architecture, the SAB 80C517 contains eight datapointers (instead of one in the 8051) which speed up external data access.

Bus Expansion Control

The external bus interface of the SAB 80C517 consists of an 8-bit data bus (port 0), a 16-bit address bus (port 0 and port 2) and five control lines. The address latch enable signal (ALE) is used to demultiplex address and data of port 0. The program memory is accessed by the program store enable signal ($\overline{PSEN}$) twice a machine cycle. A separate external access line ($\overline{EA}$) is used to inform the controller while executing out of the lower 8 Kbyte of the program memory, whether to operate out of the internal or external program memory. The read or write strobe ($\overline{RD}$, $\overline{WR}$) is used for accessing the external data memory.

Peripheral Control

All on-chip peripheral components - I/O ports, serial interfaces, timers, compare/capture registers, the interrupt controller and the A/D converter - are handled and controlled by the so-called special function registers. These registers constitute the easy-to-handle interface with the peripherals. This peripheral control concept, as implemented in the SAB 8051, provides the high flexibility for further expansion as done in the SAB 80C517.

Moreover some of the special function registers, like accumulator, B register, program status word (PSW), stack pointer (SP) and the data pointers (DPTR) are used by the CPU and maintain the machine status.

Memory Organization

The SAB 80C517 CPU manipulates operands in the following four address spaces:

- up to 64 Kbyte of program memory
- up to 64 Kbyte of external data memory
- 256 bytes of internal data memory
- a 128-byte special function register area

Program Memory

The program memory of the SAB 80C517 consists of an internal and an external memory portion (see Figure 3 Program Memory Address Phase Address Space). 8 Kbytes of program memory may reside on-chip (SAB 80C517 only), while the SAB 80C537 has no internal ROM. The program memory can be externally expanded up to 64 Kbyte. If the $\overline{EA}$ pin is held high, the SAB 80C517 executes out of the internal program memory unless the address exceeds 1 FFFH. Locations 2000H through 0FFFFH are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the SAB 80C517 fetches all instructions from the external program memory. Since the SAB 80C537 has no internal program memory, pin $\overline{EA}$ must be tied low when using this device. In either case, the 16-bit program counter is the addressing mechanism.

Locations 03H through 93H in the program memory are used by interrupt service routines.

Data Memory

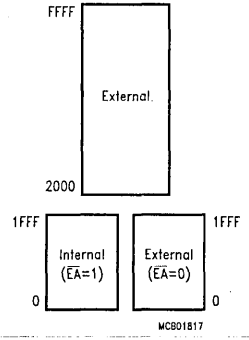
The data memory address space consists of an internal and an external memory portion.

Internal Data Memory

The internal data memory address space is divided into three physically separate and distinct blocks: the lower 128 byte of RAM, the upper RAM area, and the 128-byte special function register (SFR) area (see Figure 4 Data Memory/SFR Address Spaces). While the latter SFR area and the upper RAM area share the same address locations, they must be accessed through different addressing modes. The map in Figure 4 Data Memory/ SFR Address Spaces and the following table show the addressing modes used for the different RAM/SFR spaces.

Address Space	Locations	Addressing Mode
Lower 128 bytes of RAM	00H to 7FH	direct/indirect
Upper 128 bytes of RAM	80H to 0FFH	indirect
Special function registers	80H to 0FFH	direct

Figure 3
Program Memory Address Phase Address Space



The lower 128 bytes of the internal RAM are again grouped in three address spaces (see Figure 5 Mapping of the Lower Portion of the Internal Data Memory):

1. A general purpose register area occupies locations 0 trough 1FH (see also General Purpose Registers on page 6-11).
2. The next 16 bytes, locations 20H through 2FH, contain 128 directly addressable bits. (Programming information: These bits can be referred to in two ways, both of which are acceptable for the ASM51. One way is to refer to their addresses, i.e. 0 to 7FH. The other way is with reference to bytes 20H to 2FH. Thus bits 0 to 7 can also be referred to as bits 20.0-20.7, and bits 8-0FH are the same as 21.0-21.7 and so on. Each of the 16 bytes in this segment may also be addressed as a byte.)
3. Locations 30H to 7FH can be used as a scratch pad area.

Using the stack pointer (SP) - a special function register described in Special Function Registers on page 6-11 - the stack can be located anywhere in the whole internal data memory address space. The stack depth is limited only by the internal RAM available (256 byte maximum). However, pay attention to the fact that the stack is not overwritten by other data, and vice versa.

External Data Memory

Figure 4 Data Memory/SFR Address Spaces and Figure 5 Mapping of the Lower Portion of the Internal Data Memory contain memory maps which illustrate the internal/external data memory. To address data memory external to the chip, the "MOVX" instructions in combination with a 16-bit datapointer or an 8-bit general purpose register are used. Refer to chapter 9 (Instruction Set) or 5 (External Bus Interface) for detailed descriptions of these operations. A maximum of 64 Kbytes of external data memory can be accessed by instructions using a 16-bit address.

The datapointer structure in the SAB 80C517 deserves special attention, since it consists of eight 16-bit registers which can be alternatively selected as datapointers. See Special Function Registers on page 6-11 and chapter 5 for further details.

Figure 4
Data Memory/SFR Address Spaces

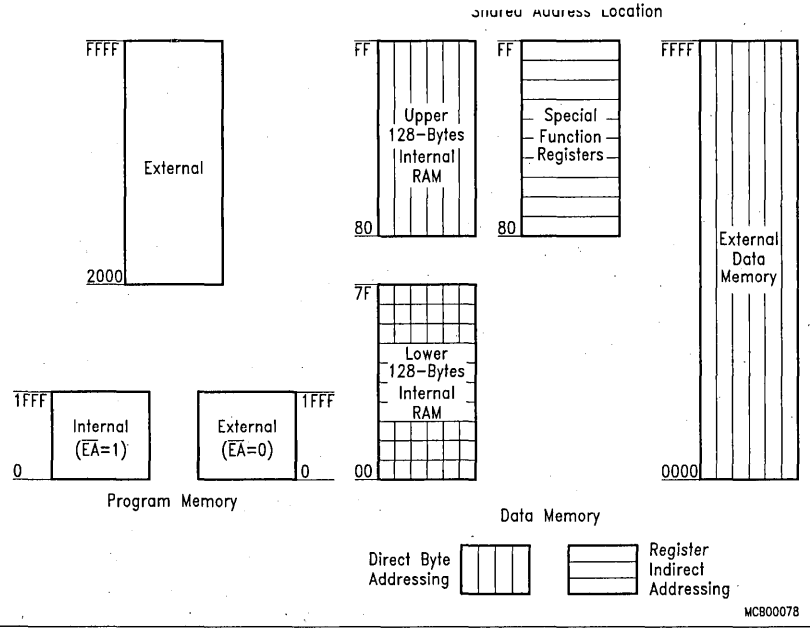
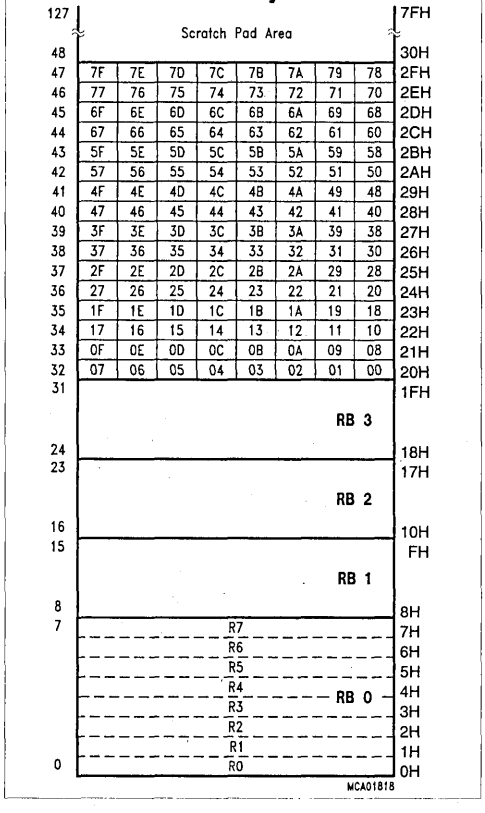


Figure 5
Mapping of the Lower Portion of the Internal Data Memory



General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general purpose registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word, PSW.3 and PSW.4, select the active register bank (see description of the PSW). This allows fast context switching, which is useful when entering subroutines or interrupt service routines. ASM51 and the device SAB 80C517 default to register bank 0.

The 8 general purpose registers of the selected register bank may be accessed by register addressing. With register addressing the instruction of code indicates which register is to be used. For indirect addressing R0 and R1 are used as pointer or index register to address internal or external memory (e.g. MOV @R0).

Reset initializes the stack pointer to location 07H and increments it once to start from location 08H which is also the first register (R0) of register bank 1. Thus, if one is going to use more than one register bank, the SP should be initialized to a different location of the RAM which is not used for data storage.

Special Function Registers

The special function register (SFR) area has two important functions. Firstly, all CPU registers except the program counter and the four register banks reside here. The CPU registers are the arithmetic registers like A, B, PSW and pointers like SP, DPHx and DPLx.

Secondly, a number of registers constitute the interface between the CPU and all on-chip peripherals. That means, all control and data transfers from and to the peripherals use this register interface exclusively.

The special function register area is located in the address space above the internal RAM from addresses 80H to FFH. All 81 special function registers of the SAB 80C517 reside here.

Sixteen SFRs, that are located on addresses dividable by eight, are bit-addressable, thus allowing 128 bit-addressable locations within the SFR area.

Since the SFR area is memory mapped, access to the special function registers is as easy as with the internal RAM, and they may be processed with most instructions. In addition, if the special functions are not used, some of them may be used as general scratch pad registers. Note, however, all SFRs can be accessed by direct addressing only.

The special function registers are listed in the following tables where they are organized in functional groups which refer to the functional blocks of the SAB 80C517. Block names and

addressable special function registers are marked with a dot in the fifth column. Special function registers with bits belonging to more then one functional block are marked with an asterisk at the symbol name.

Special Function Registers of the SAB 80C517

Block	Symbol	Name	Address	Contents after Reset
CPU	**ACC**	Accumulator	**0E0H** 1)	00H
	B	B-Register	**0F0H** 1)	00H
	DPH	Data Pointer, High Byte	83H	–
	DPL	Data Pointer, Low Byte	82H	–
	DPSEL	Data Pointer Select Register	92H	–
	PSW	Program Status Word Register	**0D0H** 1)	00H
	SP	Stack Pointer	81H	07H
A/D-Converter	**ADCON0**	A/D Converter Control Register 0	**0D8H** 1)	**00H**
	ADCON1	A/D Converter Control Register 1	0DCH	XXXX.0000B 3)
	ADDAT	A/D Converter Data Register	0D9H	00H
	DAPR	D/A Converter Program Register	0DAH	00H
Compare/ Capture Unit (CCU)	CCEN	Compare/Capture Enable Register	0C1H	00H
	CC4EN	Compare/Capture 4 Enable Register	0C9H	X000.0000B 3)
	CCH1	Compare/Capture Register 1, High Byte	0C3H	00H
	CCH2	Compare/Capture Register 2, High Byte	0C5H	00H
	CCH3	Compare/Capture Register 3, High Byte	0C7H	00H
	CCH4	Compare/Capture Register 4, High Byte	0CFH	00H
	CCL1	Compare/Capture Register 1, Low Byte	0C2H	00H
	CCL2	Compare/Capture Register 2, Low Byte	0C4H	00H
	CCL3	Compare/Capture Register 3, Low Byte	0C6H	00H
	CCL4	Compare/Capture Register 4, Low Byte	0CEH	00H
	CMEN	Compare Enable Register	0F6H	00H
	CMH0	Compare Register 0, High Byte	0D3H	00H
	CMH1	Compare Register 1, High Byte	0D5H	00H
	CMH2	Compare Register 2, High Byte	0D7H	00H
	CMH3	Compare Register 3, High Byte	0E3H	00H
	CMH4	Compare Register 4, High Byte	0E5H	00H
	CMH5	Compare Register 5, High Byte	0E7H	00H
	CMH6	Compare Register 6, High Byte	0F3H	00H
	CMH7	Compare Register 7, High Byte	0F5H	00H
	CML0	Compare Register 0, Low Byte	0D2H	00H
	CML1	Compare Register 1, Low Byte	0D4H	00H
	CML2	Compare Register 2, Low Byte	0D6H	00H
	CML3	Compare Register 3, Low Byte	0E2H	00H
	CML4	Compare Register 4, Low Byte	0E4H	00H
	CML5	Compare Register 5, Low Byte	0E6H	00H
	CML6	Compare Register 6, Low Byte	0F2H	00H
	CML7	Compare Register 7, Low Byte	0F4H	00H
	CMSEL	Compare Input Select	0F7H	00H
	CRCH	Com./Rel./Capt. Register, High Byte	0CBH	00H
	CRCL	Com./Rel./Capt. Register, Low Byte	0CAH	00H
	CTCON	Com. Timer Control Register	0E1H	00H
	CTRELH	Com. Timer Rel. Register, High Byte	0DFH	00H
	CTRELL	Com. Timer Rel. Register, Low Byte	0DEH	00H
	TH2	Timer 2, High Byte	0CDH	00H
	TL2	Timer 2, Low Byte	0CCH	00H
	T2CON	Timer 2 Control Register	0C8H 1)	00H

Block	Symbol	Name	Address	Contents after Reset
Interrupt System	IEN0	Interrupt Enable Register 0	0A8H 1)	00H
	CTCON 2)	Com. Timer Control Register	0E1H	0XXX.0000B 3)
	IEN1	Interrupt Enable Register 1	0B8H 1)	00H
	IEN2	Interrupt Enable Register 2	9AH	XXXX.0XX0B 3)
	IP0	Interrupt Priority Register 0	0A9H	00H
	IP1	Interrupt Priority Register 1	0B9H	XX00.0000B 3)
	IRCON	Interrupt Request Control Register	0C0H 1)	00H
	TCON 2)	Timer Control Register	88H 1)	00H
	T2CON 2)	Timer 2 Control Register	0C8H 1)	00H
MUL/DIV Unit	ARCON	Arithmetic Control Register	0EFH	0XXX.XXXXB 3)
	MD0	Multiplication/Division Register 0	0E9H	XXH 3)
	MD1	Multiplication/Division Register 1	0EAH	XXH 3)
	MD2	Multiplication/Division Register 2	0EBH	XXH 3)
	MD3	Multiplication/Division Register 3	0ECH	XXH 3)
	MD4	Multiplication/Division Register 4	0EDH	XXH 3)
	MD5	Multiplication/Division Register 5	0EEH	XXH 3)
Ports	P0	Port 0	80H 1)	FFH
	P1	Port 1	90H 1)	FFH
	P2	Port 2	0A0H 1)	FFH
	P3	Port 3	0B0H 1)	FFH
	P4	Port 4	0E8H 1)	FFH
	P5	Port 5	0F8H 1)	FFH
	P6	Port 6	0FAH	FFH
	P7	Port 7, Analog/Digital Input	0DBH	XXH 3)
	P8	Port 8, Analog/Digital Input, 4Bit	0DDH	XXH 3)
Pow. Sav.M	PCON	Power Control Register	87H	00H
Serial Channels	ADCON0 2)	A/D Converter Control Register	0D8H 1)	00H
	PCON 2)	Power Control Register	87H	00H
	S0BUF	Serial Channel 0 Buffer Register	99H	XXH 3)
	S0CON	Serial Channel 0 Control Register	98H1)	00H
	S1BUF	Serial Channel 1 Buffer Register	9CH	XXH 3)
	S1CON	Serial Channel 1 Control Register	9BH	0X00.0000B3)
	S1REL	Serial Channel 1 Reload Register	9DH	00H
Timer0/ Timer1	TCON	Timer Control Register	88H 1)	00H
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Watchdog	IEN0 2)	Interrupt Enable Register 0	0A8H 1)	00H
	IEN1 2)	Interrupt Enable Register 1	0B8H 1)	00H
	IP0 2)	Interrupt Priority Register 0	0A9H	00H
	WDTREL	Watchdog Timer Reload Register	86H	00H

6

The following paragraphs give a general overview of the special function registers and refer to sections where a more detailed description can be found.

Accumulator, SFR Address 0E0H

ACC is the symbol for the accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

Figure 6
Program Status Word Register (PSW), SFR Address 0D0H

	0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H	
0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

The PSW register contains program status information.

Bit	Function
CY	Carry Flag
AC	Auxiliary carry flag (for BCD operations)
F0	General purpose user flag 0
RS1 RS0	Register bank select control bits
0 0	Bank 0 selected, data address 00H-07H
0 1	Bank 1 selected, data address 08H-0FH
1 0	Bank 2 selected, data address 10H-17H
1 1	Bank 3 selected, data address 18H-1FH
OV	Overflow flag
F1	General purpose user flag
P	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.

B Register, SFR Address 0F0H

The B register is used during multiply and divide and serves as both source and destination. For other instructions it can be treated as another scratch pad register.

Stack Pointer, SFR Address 081H

The stack pointer (SP) register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions and decremented after data is popped during a POP and RET (RETI) execution, i.e. it always points to the last valid stack byte. While the stack may reside anywhere in on-chip RAM, the stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H above register bank zero. The SP can be read or written under software control.

Datapointer, SFR Address 082H and 083H Datapointer Select Register, SFR Address 092H

As a functional enhancement to standard 8051 controllers, the SAB 80C517 contains eight 16-bit registers which can be used as datapointers. To be compatible with 8051 architecture, the instruction set uses just one of these datapointers at a time. The selection of the actual datapointer is done in special function register DPSEL (datapointer select register, address 92H).

Each 16-bit datapointer (DPTRx) register is a concatenation of registers DPHx (data pointer's high order byte) and DPLx (data pointer's low order byte). These pointers are used in register-indirect addressing to move program memory constants and external data memory variables, and to branch within the 64-Kbyte program memory address space.

Since the datapointers are mainly used to access the external world, they are described in more detail in Eight Datapointers for Faster External Bus Access on page 6-18.

Ports 0 to 8

P0 to P8 are the SFR latches to port 0 to 8, respectively. The port SFRs 0 to 5 are bit-addressable. Ports 0 to 6 are 8-bit I/O ports (that is in total 56 I/O lines) which may be used as general purpose ports and which provide alternate output functions dedicated to the on-chip peripherals of the SAB 80C517.

Port 7 (8-bit) and port 8 (4-bit) are general purpose input ports and have no internal latch. That means, these port lines are used for the 12 multiplexed input lines of the A/D converter but can also be used as digital inputs. P7/P8 are the associated SFRs when the digital

value is to be read by the CPU. Both ports can be read only. You can find more about the ports in Parallel I/O on page 6-27.

Peripheral Control, Data and Status Registers

Most of the special function registers are used as control, status and data registers to handle the on-chip peripherals.

In the special function register table the register names are organized in groups and each of these groups refer to one peripheral unit. More details on how to program these registers are given in the descriptions of the following peripheral units:

Unit	Symbol	Section
Ports	–	Parallel I/O on page 6-27
Serial channels	–	Serial Interfaces on page 6-33
Timer 0/1	–	Timer 0 and Timer 1 on page 6-52
A/D converter	ADC	A/D Converter on page 6-56
Compare/capture unit	CCU	The Compare/Capture Unit (CCU) on page 6-62
Arithmetic unit (MUL/DIV unit)	MDU	Arithmetic Unit on page 6-90
Power saving control unit	–	Power Saving Modes on page 6-95
Watchdog unit	WDT/OWD	Fail Save Mechanisms on page 6-99
Interrupt system	–	Interrupt System on page 6-109

External Bus Interface

The SAB 80C517 allows for external memory expansion. To accomplish this, the external bus interface common to most 8051-based controllers is employed.

To speed up external bus accesses, the SAB 80C517 contains eight 16-bit registers used as datapointers. This enhancement to the 8051 architecture is described in Eight Datapointers for Faster External Bus Access on page 6-18.

Accessing External Memory

It is possible to distinguish between accesses to external program memory and external data memory or other peripheral components respectively. This distinction is made by hardware: Accesses to external program memory use the signal PSEN (program store enable) as a read strobe. Accesses to external data memory use RD and WR to strobe the memory (alternate functions of P3.7 and P3.6, see Parallel I/O on page 6-27.). Port 0 and port 2 (with exceptions) are used to provide data and address signals. In this section only the port 0 and port 2 functions relevant to external memory accesses are described (for further details see chapter 7.1).

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Role of P0 and P2 as Data/Address Bus

When used for accessing external memory, port 0 provides the data byte time-multiplexed with the low byte of the address. In this state, port 0 is disconnected from its own port latch, and the address/data signal drives both FETs in the port 0 output buffers. Thus, in this application, the port 0 pins are not open-drain outputs and do not require external pullup resistors.

During any access to external memory, the CPU writes 0FFH to the port 0 latch (the special function register), thus obliterating whatever information the port 0 SFR may have been holding.

Whenever a 16-bit address is used, the high byte of the address comes out on port 2, where it is held for the duration of the read or write cycle. During this time, the port 2 lines are disconnected from the port 2 latch (the special function register).

Thus the port 2 latch does not have to contain 1s, and the contents of the port 2 SFR are not modified.

If an 8-bit address is used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. It should be noted that, if a port 2 pin outputs an address bit that is a 1, strong pullups will be used for the entire read/write cycle and not only for two oscillator periods.

Timing

The timing of the external bus interface, in particular the relationship between the control signals ALE, PSEN, RD/WR and information on port 0 and port 2, is illustrated in "Figure 8 a) and b) External Program Memory Execution".

Data memory: In a write cycle, the data byte to be written appears on port 0 just before WR is activated, and remains there until after WR is deactivated. In a read cycle, the incoming byte is accepted at port 0 before the read strobe is deactivated.

Program memory: Signal PSEN functions as a read strobe. For further information see PSEN, Program Store Enable on page 6-21.

External Program Memory Access

The external program memory is accessed under two conditions:

- whenever signal EA is active; or

- whenever the program counter (PC) contains a number that is larger than 01FFFH

This requires the ROMless version SAB 80C537 to have $\overline{EA}$ wired low to allow the lower 8 K program bytes to be fetched from external memory.

When the CPU is executing out of external program memory, all 8 bits of port 2 are dedicated to an output function and may not be used for general-purpose I/O. The contents of the port 2 SFR however is not affected. During external program memory fetches port 2 lines output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @DPTR or a MOVX @Ri).

Since the SAB 80C537 has no internal program memory, accesses to program memory are always external, and port 2 is at all times dedicated to output the high-order address byte. This means that port 0 and port 2 of the SAB 80C537 can never be used as general-purpose I/O. This also applies to the SAB 80C517 when it is operated with only an external program memory.

Eight Datapointers for Faster External Bus Access

The Importance of Additional Datapointers

The standard 8051 architecture provides just one 16-bit pointer for indirect addressing of external devices (memories, peripherals, latches, etc.). Except for a 16-bit "move immediate" to this datapointer and an increment instruction, any other pointer handling is to be done byte by byte. For complex applications with numerous external peripherals or extended data storage capacity this turned out to be a "bottle neck" for the 8051's communication to the external world. Especially programming in high-level languages (PLM51, "C", PASCAL51) requires extended RAM capacity and at the same time a fast access to this additional RAM because of the reduced code efficiency of these languages.

How the Eight Datapointers of the SAB 80C517 are Realized

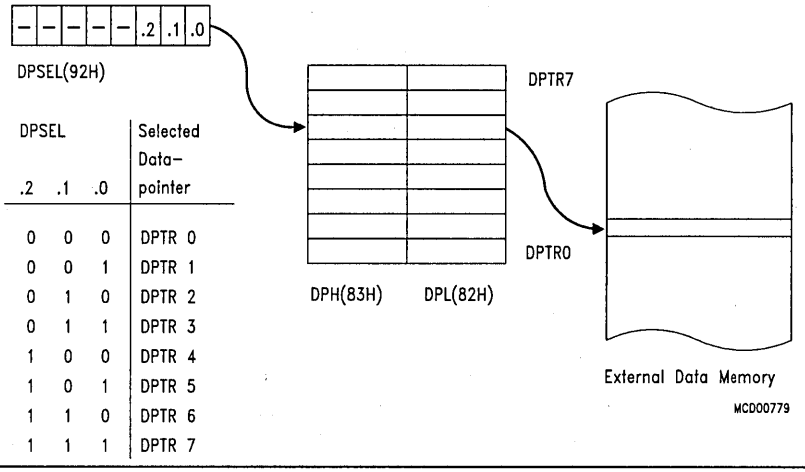
Simply adding more datapointers is not suitable because of the need to keep up 100% compatibility to the 8051 instruction set. This instruction set, however, allows the handling of only one single 16-bit datapointer (DPTR, consisting of the two 8-bit SFRs DPH and DPL).

To meet both of the above requirements (speed up external accesses, 100% compatibility to 8051 architecture) the SAB 80C517 contains a set of eight 16-bit registers from which the actual datapointer can be selected.

This means that the user's program may keep up to eight 16-bit addresses resident in these registers, but only one register at a time is selected to be the datapointer. Thus the datapointer in turn is accessed (or selected) via indirect addressing. This indirect addressing is done through a special function register called DPSEL (data pointer select register). All instructions of the SAB 80C517 which handle the datapointer therefore affect only one of the eight pointers which is addressed by DPSEL at that very moment.

"Figure 7 Accessing of External Data Memory via Multiple Datapointers" illustrates the addressing mechanism: a 3-bit field in register DPSEL points to the currently used DPTRx. Any standard 8051 instruction (e.g. MOVX @DPTR, A - transfer a byte from accumulator to an external location addressed by DPTR) now uses this activated DPTRx.

Figure 7
Accessing of External Data Memory via Multiple Datapointers



DPSEL .2	.1	.0	Selected Data-pointer
0	0	0	DPTR 0
0	0	1	DPTR 1
0	1	0	DPTR 2
0	1	1	DPTR 3
1	0	0	DPTR 4
1	0	1	DPTR 5
1	1	0	DPTR 6
1	1	1	DPTR 7

Advantages of Multiple Datapointers

Using the above addressing mechanism for external data memory results in less code and faster execution of external accesses. Whenever the contents of the datapointer must be altered between two or more 16-bit addresses, one single instruction, which selects a new datapointer, does this job. If the program uses just one datapointer, then it has to save the old value (with two 8-bit instructions) and load the new address, byte by byte. This not only takes more time, it also requires additional space in the internal RAM.

Application Example and Performance Analysis

The following example shall demonstrate the involvement of multiple data pointers in a table transfer from the code memory to external data memory.

Start address of ROM source table: 1FFFH
Start address of table in external RAM: 2FA0H

1) Using only One Datapointer (Code for an 8051)

Initialization Routine

Action	Code
Initialize shadow_variables with source_pointer	MOV LOW(SRC_PTR), #0FFH MOV HIGH(SRC_PTR), #1FH
Initialize shadow_variables with destination_pointer	MOV LOW(DES_PTR), #0A0H MOV HIGH(DES_PTR), #2FH

Table Look-up Routine under Real Time Conditions

Action	Code	Machine Cycles
Save old datapointer	PUSH DPL PUSH DPH	2 2
Load Source Pointer	MOV DPL, LOW(SRC_PTR) MOV DPH, HIGH(SRC_PTR)	2 2
Increment and check for end of table (execution time not relevant for this consideration)	INC DPTR CJNE... ...	– – –
Fetch source data byte from ROM table	MOVC A,@DPTR	2

Save source_pointer and load destination_pointer	MOV LOW(SRC_PTR), DPL
	MOV HIGH(SRC_PTR), DPH
	MOV DPL, LOW(DES_PTR)
	MOV DPH, HIGH(DES_PTR)
		2
		2
		2
		2

Increment destination_pointer (ex. time not relevant)	INC DPTR	–
Transfer byte to destination address	MOVX @DPTR, A	2
Save destination_pointer	MOV LOW(DES_PTR), DPL MOV HIGH(DES_PTR),DPH	2 2
Restore old datapointer	POP DPH POP DPL	2 2
Total execution time (machine cycles)	–	28

2) Using Two Datapointers (Code for an SAB 80C517)

Initialization Routine

Action	Code
Initialize DPTR6 with source pointer	MOV DPSEL, #06H MOV DPTR, #1FFFH
Initialize DPTR7 with destination pointer	MOV DPSEL, #07H MOV DPTR, #2FA0H

Table Look-up Routine under Real Time Conditions

Action	Code	Machine Cycles
Save old source pointer	PUSH DPSEL	2
Load source pointer	MOV DPSEL, #06H	2
Increment and check for end of table (execution time not relevant for this consideration)	INC DPTR CJNE... ...	– – –
Fetch source data byte from ROM table	MOVC A,@DPTR	2
Save source_pointer and load destination_pointer	MOV DPSEL, #07H	2
Transfer byte to destination address	MOVX @DPTR, A	2
Save destination pointer and restore old datapointer	POP DPSEL	2
Total execution time (machine cycles)	–	12

The above example shows that utilization of the SAB 80C517's multiple datapointers can make external bus accesses two times as fast as with a standard 8051 or 8051 derivative. Here, four data variables in the internal RAM and two additional stack bytes were spared, too. This means for some applications where all eight datapointers are employed that an SAB 80C517 program has up to 24 byte (16 variables and 8 stack bytes) of the internal RAM free for other use.

$\overline{\text{PSEN}}$, Program Store Enable

The read strobe for external fetches is $\overline{\text{PSEN}}$. $\overline{\text{PSEN}}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{\text{PSEN}}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{\text{PSEN}}$ is activated its timing is not the same as for $\overline{\text{RD}}$. A complete $\overline{\text{RD}}$ cycle, including activation and deactivation of ALE and $\overline{\text{RD}}$, takes 12 osillator periods. A complete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$ takes 6 oscillator periods. The execution sequence for these two types of read cycles is shown in Figure 5-2 a) and b).

ALE, Address Latch Enable

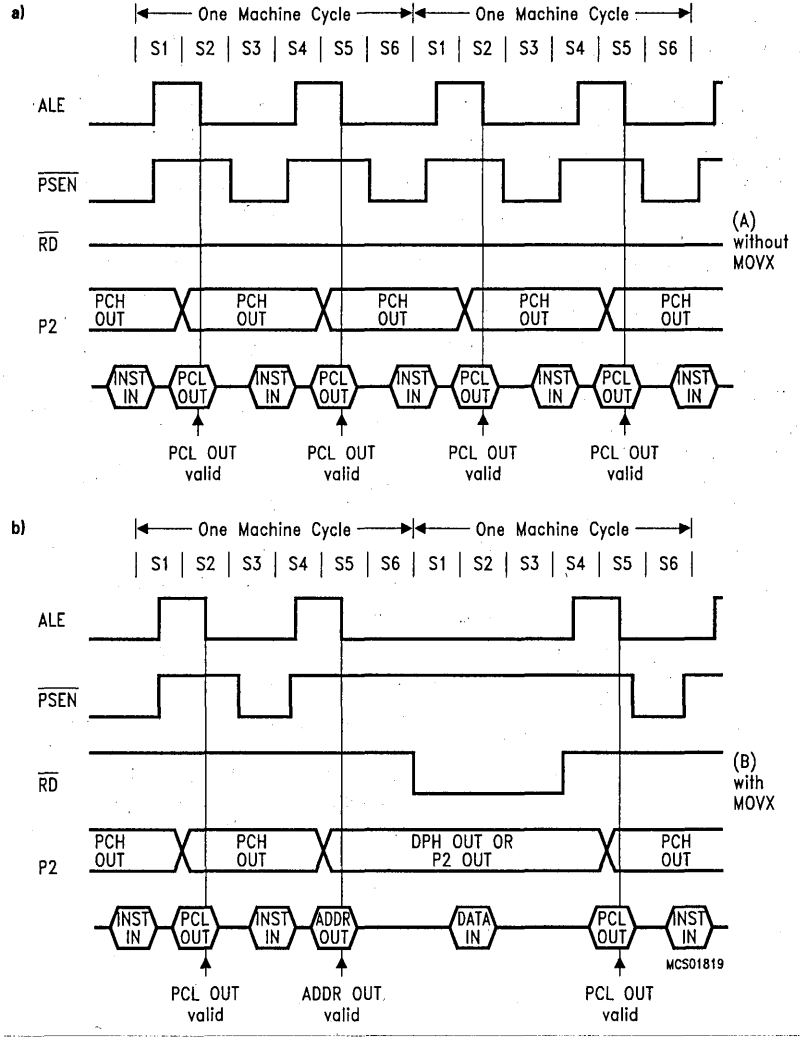
The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 into an external latch during fetches from external memory. The address byte is valid at the negative transition of ALE. For that purpose, ALE is activated twice every machine cycle. This activation takes place even if the cycle involves no external fetch. The only time no ALE pulse comes out is during an access to external data memory when $\overline{\text{RD}}$/$\overline{\text{WR}}$ signals are active. The first ALE of the second cycle of a MOVX instruction is missing (see Figure 5-2 b)). Consequently, in any system that does not use data memory, ALE is activated at a constant rate of 1/6 of the oscillator frequency and can be used for external clocking or timing purposes.

Overlapping External Data and Program Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is used for storing data. In the SAB 80C517, the external program and data memory spaces can be combined by AND-ing $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive logic AND of these two signals produces an active low read strobe that can be used for the combined physical memory. Since the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory needs to be fast enough to adapt to the $\overline{\text{PSEN}}$ cycle.

6

Figure 8 a) and b)
External Program Memory Execution



System Reset

Reset Function and Circuitries

The hardware reset function incorporated in the SAB 80C517 allows for an easy automatic start-up at a minimum of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation in order to restart the device. This is particularly done when the power-down mode (see Power Saving Modes on page 6-95) is to be terminated.

Additionally to the hardware reset, which is applied externally to the SAB 80C517, there are two internal reset sources, the watchdog timer and the oscillator watchdog. They are described in detail in Fail Save Mechanisms on page 6-99. The chapter at hand only deals with the external hardware reset.

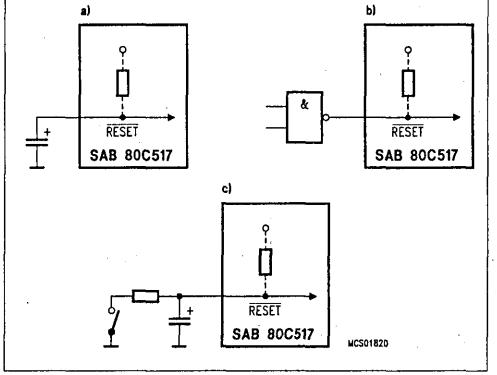
The reset input is an active low input at pin 10 ($\overline{\text{RESET}}$). An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the $\overline{\text{RESET}}$ pin must be held low for at least two machine cycles (24 oscillator periods) while the oscillator is running. With the oscillator running the internal reset is executed during the second machine cycle in which $\overline{\text{RESET}}$ is low and is repeated every cycle until $\overline{\text{RESET}}$ goes high again.

During reset, pins ALE and $\overline{\text{PSEN}}$ are configured as inputs and should not be stimulated externally. (An external stimulation at these lines during reset activates several test modes which are reserved for test purposes. This in turn may cause unpredictable output operations at several port pins).

A pullup resistor is internally connected to V_{CC} to allow a power-up reset with an external capacitor only. An automatic reset can be obtained when V_{CC} is applied by connecting the reset pin to V_{SS} via a capacitor as shown in "Figure 9 Reset Circuitries". After V_{CC} has been turned on, the capacitor must hold the voltage level at the reset pin for a specified time below the upper threshold of the Schmitt trigger to effect a complete reset.

The time required is the oscillator start-up time plus 2 machine cycles, which, under normal conditions, must be at least 10 - 20 ms for a crystal oscillator. This requirement is usually met using a capacitor of 4.7 to 10 microfarad. The same considerations apply if the reset signal is generated externally (Figure 9 b). In each case it must be assured that the oscillator has started up properly and that at least two machine cycles have passed before the reset signal goes inactive.

Figure 9
Reset Circuitries



A correct reset leaves the processor in a defined state. The program execution starts at location 0000H. The default values of the special function registers (SFR) to which they are forced during reset are listed in "Table 1". After reset is internally accomplished the port latches of ports 0 to 6 default in 0FFH. This leaves port 0 floating, since it is an open drain port when not used as data/address bus. All other I/O port lines (ports 1 through 6) output a one (1). Ports 7 and 8, which are input-only ports, have no internal latch and therefore the contents of the special function registers P7 and P8 depend on the levels applied to ports 7 and 8.

The contents of the internal RAM of the SAB 80C517 is not affected by a reset. After power-up the contents is undefined, while it remains unchanged during a reset it the power supply is not turned off.

Table 1

Register	Contents	Register	Contents
PC	0000_H	IEN0, IEN1	00_H
ACC	00_H	IEN2	XXXX 0XX0B
ADCON0	00_H	IP0, IP1	00_H
ADCON1	XXXX 0000B	IRCON	00_H
ADDAT	00_H	MD0-5	XX_H
ARCON	0XXX XXXXB	P0-P6	$0FF_H$
B	00_H	PCON	00_H
CCL1-4	00_H	PSW	00_H
CCH1-4	00_H	S0BUF, S1BUF	$0XX_H$
CCEN	00_H	S0CON	00_H
CC4EN	00_H	S1CON	0X00 0000B
CMEN	00_H	S1REL	00_H
CML0-7	00_H	SP	07_H
CMH0-7	00_H	TCON	00_H
CMSEL	00_H	TL0, TH0	00_H
CRCL, CRCH	00_H	TL1, TH1	00_H
CTCON	XXXX 0000B	TL2, TH2	00_H
CTRELL, CTRELH	00_H	TMOD	00_H
DAPR	00_H	T2CON	00_H
DPSEL	XXXX X000B	WDTREL	00_H
DPTR0-7	0000_H	–	–

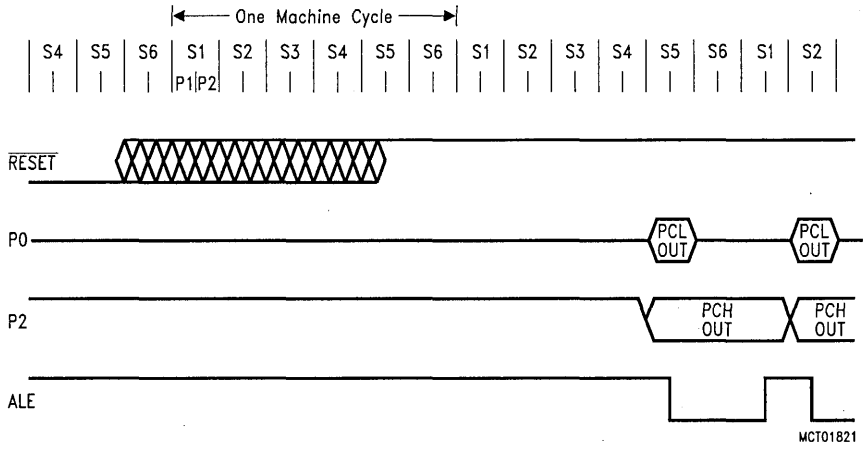
Hardware Reset Timing

This section describes the timing of the hardware reset signal.

The input pin $\overline{\text{RESET}}$ is sampled once during each machine cycle. This happens in state 5 phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (low level at pin 10) the internal reset procedure is started. It needs two complete machine cycles to put the complete device to its correct reset state, i.e. all special function registers contain their default values, the port latches contain 1's etc. Note that this reset procedure is not performed if there is no clock available at the device (This can be avoided using the oscillator watchdog, which provides an auxiliary clock for performing a correct reset without clock at the XTAL1 and XTAL2 pins. See Fail Save Mechanisms on page 6-99 for further details). The $\overline{\text{RESET}}$ signal must be active for at least two machine cycles; after this time the SAB 80C517 remains in its reset state as long as the signal is active. When the signal goes inactive this transition is recognized in the following state 5 phase 2 of the machine cycle. Then the processor starts its address output (when configured for external ROM) in the following state 5 phase 1. One phase later (state 5 phase 2) the first falling edge at pin ALE occurs.

"Figure 10 CPU Timing after Reset" shows this timing for a configuration with $\overline{\text{EA}} = 0$ (external program memory). Thus, between the release of the $\overline{\text{RESET}}$ signal and the first falling edge at ALE there is a time period of at least one machine cycle but less than two machine cycles.

Figure 10
CPU Timing after Reset



Reset Output Pin ($\overline{RO}$)

As mentioned before the SAB 80C517 internally synchronizes an external reset signal at pin $\overline{RESET}$ in order to perform a reset procedure. Additionally, the SAB 80C517 provides several "fail-save" mechanisms, e.g. watchdog timer and oscillator watchdog, which can internally generate a reset, too. Thus, it is often important to inform also the peripherals external to the chip that a reset is being performed and that the controller will soon start its program again.

For that purpose, the SAB 80C517 has a pin dedicated to output the internal reset request. This reset output ($\overline{RO}$) at pin 82 shows the internal (and already synchronized) reset signal requested by any of the three possible sources in the SAB 80C517: external hardware reset, watchdog timer reset, oscillator watchdog reset. The duration of the active low signal of the reset output depends on the source which requests it. In the case of the external hardware reset it is the synchronized external reset signal at pin $\overline{RESET}$. In the case of a watchdog timer reset or oscillator watchdog reset the RESET OUT signal takes at least two machine cycles, which is the minimal duration for a reset request allowed. For details - how the reset requests are OR-ed together and how long they last - see also chapter 7.8 "Fail-Save Mechanisms".

6

On-Chip Peripheral Components

This chapter gives detailed information about all on-chip peripherals of the SAB 80C517 except for the integrated interrupt controller, which is described separately. Parallel I/O on page 6-27 and Serial Interfaces on page 6-33 are associated with the general parallel and serial I/O facilities while the remaining sections describe the miscellaneous functions such as the timers, A/D converter, compare/capture unit, multiplication/division unit, power saving modes, "fail-save" mechanisms, oscillator and clock circuitries and system clock output.

Parallel I/O

Port Structures

Digital I/O

The SAB 80C517 allows for digital I/O on 56 lines grouped into 7 bidirectional 8-bit ports. Each port bit consists of a latch, an output driver and an input buffer. Read and write accesses to the I/O ports P0 through P6 are performed via their corresponding special function registers P0 to P6.

The output drivers of port 0 and 2 and the input buffers of port 0 are also used for accessing external memory. In this application, port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the port 2 pins continue emitting the P2 SFR contents (see also chapter 7.1.2 and chapter 5 for more details about the external bus interface).

Digital/Analog Input Ports

Ports 7 and 8 are available as input ports only and provide for two functions. When used as digital inputs, the corresponding SFR's P7 and P8 contain the digital value applied to port 7 and port 8 lines. When used for analog inputs the desired analog channel is selected by a three-bit field in SFR ADCON0 or a four-bit field in SFR ADCON1, as described in A/D Converter on page 6-56. Of course, it makes no sense to output a value to these input-only ports by writing to the SFR's P7 or P8; this will have no effect.

If a digital value is to be read, the voltage levels are to be held within the input voltage specifications (V_{IL}/V_{IH}). Since P7 and P8 are not bit-addressable registers, all input lines of P7 or P8 are read at the same time by byte instructions.

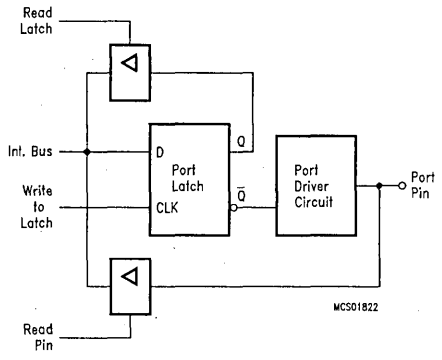
Nevertheless, it is possible to use ports 7 and 8 simultaneously for analog and digital input. However, care must be taken that all bits of P7 or P8 that have an undetermined value caused by their analog function are masked.

In order to guarantee a high-quality A/D conversion, digital input lines of port 7 and port 8 should not toggle while a neighboring port pin is executing an A/D conversion. This could produce crosstalk to the analog signal.

Digital I/O Port Circuitry

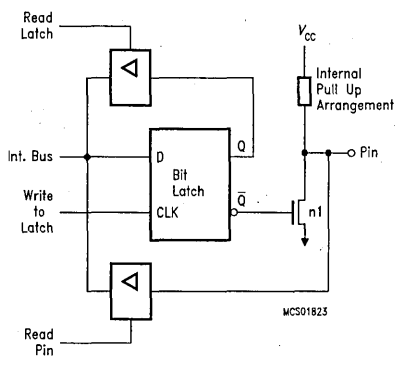
Figure 11 Basic Structure of a Port Circuitry shows a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the 7 I/O-ports. The bit latch (one bit in the port's SFR) is represented as a type-D flip-flop, which will clock in a value from the internal bus in response to a "write-to-latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read-latch" signal from the CPU. The level of the port pin self is placed on the internal bus in response to a "read-pin" signal from the CPU. Some instructions that read from a port (i.e. from the corresponding port SFR P0 to P6) activate the "read-latch" signal, while others activate the "read-pin" signal (see Read-Modify-Write Feature of Ports 0 through 6 on page 6-33).

Figure 11
Basic Structure of a Port Circuitry



Port 1 through 6 output drivers have internal pullup FET's (see Figure 12 Basic Output Driver Circuit of Ports 1 through 6). Each I/O line can be used independently as an input or output. To be used as an input, the port bit must contain a one (1) (that means for Figure 12: $\overline{Q}$ = 0), which turns off the output driver FET n1. Then, for ports 1 through 6, the pin is pulled high by the internal pullups, but can be pulled low by an external source. When externally pulled low the port pins source current (I_{IL} or I_{TL}). For this reason these ports are sometimes called "quasi-bidirectional".

Figure 12
Basic Output Driver Circuit of Ports 1 through 6



In fact, the pullups mentioned before and included in Figure 12 Basic Output Driver Circuit of Ports 1 through 6 are pullup arrangements as shown in Figure 13 Output Driver Circuit of Ports 1 through 6. One n-channel pulldown FET and three pullup FETs are used:

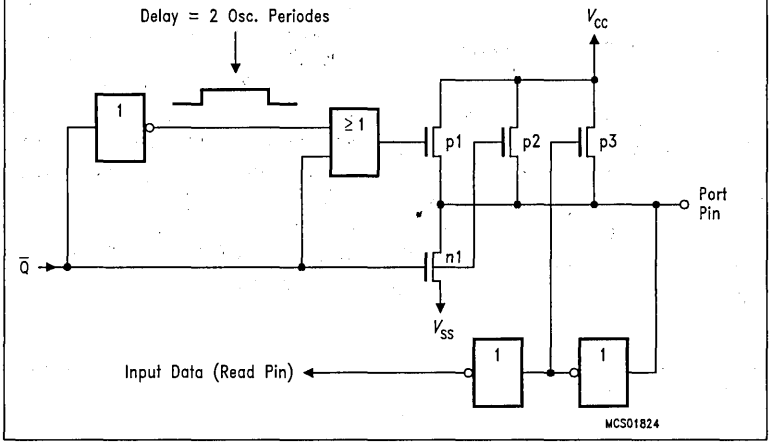
The **pulldown FET n1** is of n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a "0" is programmed to the port pin. A short circuit to V_{CC} must be avoided if the transistor is turned on, since the high current might destroy the FET.

The **pullup FET p1** is of p-channel type. It is activated for two oscillator periods (S1P1 and S1P2) if a 0-to-1 transition is programmed to the port pin, i.e. a "1" is programmed to the port latch which contained a "0". The extra pullup can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.

The **pullup FET p2** is of p-channel type. It is always activated when a "1" is in the port latch, thus providing the logic high output level. This pullup FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.

The **pullup FET p3** is of p-channel type. It is only activated if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pullup current if a logic high level shall be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level, e.g. when used as input. In this configuration only the weak pullup FET p2 is active, which sources the current I_{IL}. If, in addition, the pullup FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, an additional power consumption can be avoided if port pins are used as inputs with a low level applied. However, the driving capability is stronger if a logic high level is output.

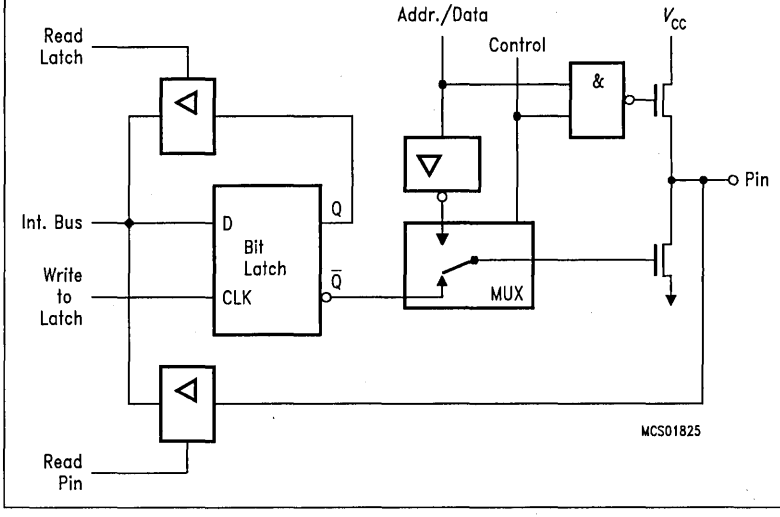
Figure 13
Output Driver Circuit of Ports 1 through 6



Port 0, in contrast to ports 1 through 6, is considered as "true" bidirectional, because the port 0 pins float when configured as inputs. Thus, this port differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 14 Port 0 Circuitry) is used only when the port is emitting 1's during the external memory accesses. Otherwise, the pullup is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a "1" to the port latch leaves both output FETs off and the pin floats.

6

Figure 14
Port 0 Circuitry



In that condition it can be used as high-impedance input. If port 0 is configured as general I/O port and has to emit logic high level (1), external pullups are required.

Port 0 and Port 2 used as Address/Data Bus

As shown in Figure 14 Port 0 Circuitry and Figure 15 Port 2 Circuitry, the output drivers of ports 0 and 2 can be switched to an internal address or address/data bus for use in external memory accesses. In this application they cannot be used as general purpose I/O, even if not all address lines are used externally. The switching is done by an internal control signal dependent on the input level at the $\overline{EA}$ pin and/or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P2 SFR remains unchanged while the P0 SFR has 1's written to it. Being an address/data bus, port 0 uses a pullup FET as shown in Figure 14 Port 0 Circuitry). When a 16-bit address is used, port 2 uses the additional strong pullups p1 to emit 1's for the entire external memory cycle instead of the weak ones (p2 and p3) used during normal port activity.

Figure 15
Port 2 Circuitry

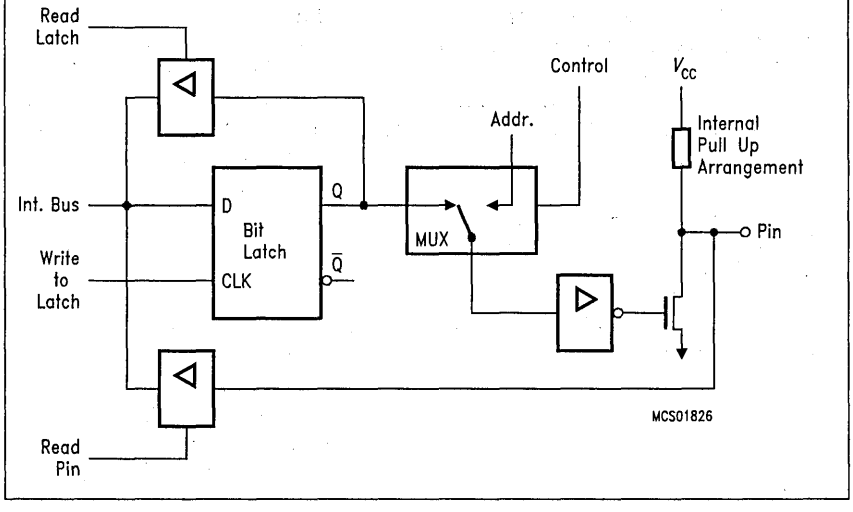
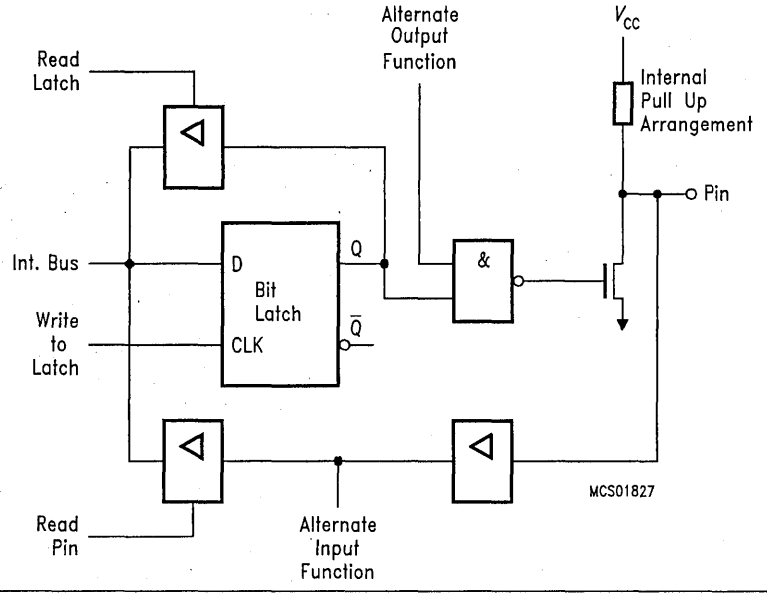


Figure 16
Circuitry of Ports 1, 3, 4, 5, and 6.0 through 6.2



Alternate Functions

Several pins of ports 1, 3, 4, 5 and 6 are multifunctional. They are port pins and also serve to implement special features as listed in Table 2 Alternate Functions of Port Pins.

Figure 16 Circuitry of Ports 1, 3, 4, 5, and 6.0 through 6.2 shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise the pull-down FET is on and the port pin is stuck at 0. (This does not apply to ports 1.0 to 1.4 and ports 5.0 to 5.7 when operated in compare output mode; refer to Compare Function in the CCU on page 6-71 for details). After reset all port latches contain ones (1).

Ports 6.3 through 6.7 have no alternate functions as described above. Therefore, the port circuitry can do without the switching capability between alternate function and normal I/O operation. This more simple circuitry is shown as basic port structure in Figure 11 Basic Structure of a Port Circuitry and Figure 12 Basic Output Driver Circuit of Ports 1 through 6.

Table 2
Alternate Functions of Port Pins.

Port	Pin	Alternate Function
P1.0	$\overline{INT3}$/CC0	Ext. interrupt 3/capture 0/compare 0
P1.1	INT4/CC1	Ext. interrupt 4/capture 1/compare 1
P1.2	INT5/CC2	Ext. interrupt 5/capture 2/compare 2
P1.3	INT6/CC3	Ext. interrupt 6/capture 3/compare 3
P1.4	$\overline{INT2}$/CC4	Ext. interrupt 2/capture 4/compare 4
P1.5	T2EX	Timer 2 ext. reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external count input
P3.0	RXD	Serial input channel 0
P3.1	TXD	Serial output channel 0
P3.2	$\overline{INT0}$	Ext. interrupt 0
P3.3	$\overline{INT1}$	Ext. interrupt 1
P3.4	T0	Timer 0 external count input
P3.5	T1	Timer 1 external count input
P3.6	$\overline{WR}$	External data memory write strobe
P3.7	$\overline{RD}$	External data memory read strobe
P4.0	CM0	Compare 0 of compare unit CM0-7
P4.1	CM1	Compare 1 of compare unit CM0-7
P4.2	CM2	Compare 2 of compare unit CM0-7
P4.3	CM3	Compare 3 of compare unit CM0-7
P4.4	CM4	Compare 4 of compare unit CM0-7
P4.5	CM5	Compare 5 of compare unit CM0-7
P4.6	CM6	Compare 6 of compare unit CM0-7
P4.7	CM7	Compare 7 of compare unit CM0-7
P5.0	CCM0	Concurrent compare 0
P5.1	CCM1	Concurrent compare 1
P5.2	CCM2	Concurrent compare 2
P5.3	CCM3	Concurrent compare 3
P5.4	CCM4	Concurrent compare 4
P5.5	CCM5	Concurrent compare 5
P5.6	CCM6	Concurrent compare 6
P5.7	CCM7	Concurrent compare 7
P6.0	$\overline{ADST}$	Ext. A/D converter start
P6.1	RXD1	Serial input channel 1
P6.2	TXD1	Serial output channel 1

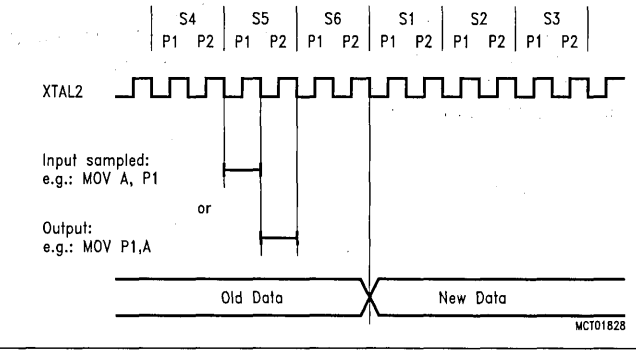
Port Handling

Port Timing

When executing an instruction that changes the value of a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are only sampled by their output buffers during phase 1 of any clock period (during phase 2 the output buffer holds the value it noticed during the previous phase 1). Consequently, the new value in the port latch will not appear at the output pin until the next phase 1, which will be at S1P1 of the next machine cycle.

When an instruction reads a value from a port pin (e.g. MOV A, P1) the port pin is actually sampled in state 5 phase 1 or phase 2 depending on port and alternate functions. Figure 17 Port Timing illustrates this port timing. It must be noted that this mechanism of sampling once per machine cycle is also used if a port pin is to detect an "edge", e.g. when used as counter input. In this case an "edge" is detected when the sampled value differs from the value that was sampled the cycle before. Therefore, there must be met certain requirements on the pulse length of signals in order to avoid signal "edges" not being detected. The minimum time period of high and low level is one machine cycle, which guarantees that this logic level is noticed by the port at least once.

Figure 17
Port Timing



Port Loading and Interfacing

The output buffers of ports 1 through 6 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can be looked up in the DC characteristics in the Data Sheet of the SAB 80C517. The corresponding parameters are V_{OL} and V_{OH}.The same applies to port 0 output buffers. They do, however, require external pullups to drive floating inputs, except when being used as the address/data bus.

When used as inputs it must be noted that the ports 1 through 6 are not floating but have internal pullup transistors. The driving devices must be capable of sinking a sufficient current if a logic low level shall be applied to the port pin (the parameters I_{TL} and I_{IL} in the DC characteristics specify these currents). Port 0 as well as the input only ports 7 and 8, however, have floating inputs when used for digital input.

Read-Modify-Write Feature of Ports 0 through 6

Some port-reading instructions read the latch and others read the pin (see Figure 11 Basic Structure of a Port Circuitry). The instructions reading the latch rather than the pin read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions, which are listed in Table 3 Read-Modify-Write Instructions. If the destination is a port or a port bit, these instructions read the latch rather than the pin. Note that all other instructions which can be used to read a port, exclusively read the port pin. In any case, reading from latch or pin, resp., is performed by reading the SFR P0 to P6; for example, "MOV A, P3" reads the value from port 3 pins, while "ANL P4, #0AAH" reads from the latch, modifies the value and writes it back to the latch.

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. The reason is that they read the port byte, all 8 bits, modify the addressed bit, then write the complete byte back to the latch.

Table 3 Read-Modify-Write Instructions

Instruction	Function
ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic exclusive OR; e.g. XRL P3, A
JBC	Jump if bit is set and clear bit; e.g. JBC P1.1, LABEL
CPL	Complement bit; e.g. CPL P3.0
INC	Increment byte; e.g. INC P4
DEC	Decrement byte; e.g. DEC P5
DJNZ	Decrement and jump if not zero; e.g. DJNZ P3, LABEL
MOV Px.y, C	Move carry bit to bit y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

The reason why read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a "1" is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor (approx. 0.7 V, i.e. a logic low level!) and interpret it as "0". For example, when modifying a port bit by a SETB or CLR instruction, another bit in this port with the above mentioned configuration might be changed if the value read from the pin were written back to the latch. However, reading the latch rather than the pin will return the correct value of "1".

Serial Interfaces

The SAB 80C517 has two serial interfaces which are functionally nearly identical concerning the asynchronous modes of operation. The two channels are full-duplex, meaning they can transmit and receive simultaneously. They are also receive buffered, meaning they can commence reception of a second byte before a previously received byte has been read from the receive register (however, if the first byte still has not been read by the time reception of the second byte is complete, the last received byte will be lost). The serial channel 0 is completely compatible with the serial channel of the SAB 80(C)51. Serial channel 1 has the same

functionality in its asynchronous modes, but the synchronous mode is lacking.

Serial Interface 0

Operating Modes of Serial Interface 0

The serial interface 0 can operate in four modes (one synchronous mode, three asynchronous modes). The baud rate clock for this interface is derived from the oscillator frequency (mode 0, 2) or generated either by timer 1 or by a dedicated baud rate generator (mode 1, 3). A more detailed description of how to set the baud rate will follow in Baud Rates of Serial Channel 0 on page 6-36.

Mode 0: Shift register (synchronous) mode:

Serial data enters and exits through RXD0. TxD0 outputs the shift clock. 8 data bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 8-bit UART, variable baud rate:

10 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB80 in special function register S0CON. The baud rate is variable.

Mode 2: 9-bit UART, fixed baud rate:

11 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmission, the 9th data bit (TB80 in S0CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB80 or a second stop bit by setting TB80 to 1. On reception the 9th data bit goes into RB80 in special function register S0CON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 9-bit UART, variable baud rate:

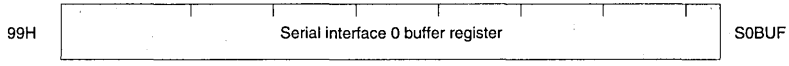
11 bits are transmitted (through TxD0) or received (through RxD0): a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmission, the 9th data bit (TB80 in S0CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB80 or a second stop bit by setting TB80 to 1. On reception, the 9th data bit goes into RB80 in special function register S0CON, while the stop bit is ignored. In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. Reception is initiated in mode 0 by the condition RI0 = 0 and REN0 = 1. Reception is initiated in the other modes by the incoming start bit if REN0 = 1. The serial interfaces also provide interrupt requests when a transmission or a reception of a frame has completed. The corresponding interrupt request flags for serial interface 0 are TI0 or RI0, resp. See Interrupt System on page 6-109 for more details about the interrupt structure. The interrupt request flags TI0 and RI0 can also be used for polling the serial interface 0 if the serial interrupt is not to be used (i.e. serial interrupt 0 not enabled).

The control and status bits of the serial channel 0 in special function register S0CON are illustrated in Figure 19 Special Function Register S0CON (Address 98H). Figure 18 Special Function Register S0BUF (Address 99H) shows the special function register S0BUF which is the data register for receive and transmit. The following table summarizes the operating modes of serial interface 0.

SM0	SM1	Mode	Descriptions	Baud Rate
0	0	0	Shift register	$f_{OSC}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$f_{OSC}/64$ or $f_{OSC}/32$
1	1	3	9-bit UART	Variable

Figure 18 Special Function Register S0BUF (Address 99H)



Receive and transmit buffer of serial interface 0. Writing to S0BUF loads the transmit register and initiates transmission. Reading out S0BUF accesses a physically separate receive register.

Figure 19 Special Function Register S0CON (Address 98H)

	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
98H	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	S0CON

Bit		Symbol
SM0	SM1	
0	0	Serial mode 0: Shift register mode, fixed baud rate
0	1	Serial mode 1: 8-bit UART, variable baud rate
1	0	Serial mode 2: 9-bit UART, fixed baud rate
1	1	Serial mode 3: 9-bit UART, variable baud rate
SM20		Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3 and SM20 being set to 1, RI0 will not be activated if the received 9th data bit (RB80) is 0. In mode 1 and SM20 = 1, RI0 will not be activated if a valid stop bit has not been received. In mode 0, SM20 should be 0.
REN0		Receiver enable. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB80		Transmitter bit 8. Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB80		Receiver bit 8. In modes 2 and 3 it is the 9th bit that was received. In mode 1, if SM20 = 0, RB80 is the stop bit that was received. In mode 0, RB80 is not used.
TI0		Transmitter interrupt. Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must by cleared by software.
RI0		Receiver interrupt. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software

Multiprocessor Communication Feature

Modes 2 and 3 of the serial interface 0 have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB80. Then a stop bit follows. The port can be programmed such that when the stop bit is received, the serial port 0 interrupt will be activated (i.e. the request flag RI0 is set) only if RB80 = 1. This feature is enabled by setting bit SM20 in S0CON. A way to use this feature in multiprocessor communications is as follows.

If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM20 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM20 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM20 again. The slaves that were not addressed leave their SM20 set and go on about their business, ignoring the incoming data bytes.

SM20 has no effect in mode 0. In mode 1 SM20 can be used to check the validity of the stop bit. If SM20 = 1 in mode 1, the receive interrupt will not be activated unless a valid stop bit is received.

Baud Rates of Serial Channel 0

As already mentioned there are several possibilities to generate the baud rate clock for the serial interface 0 depending on the mode in which it is operated.

To clarify the terminology, something should be said about the difference between "baud rate clock" and "baud rate". The serial interface requires a clock rate which is 16 times the baud rate for internal synchronization, as mentioned in the detailed description of the various operating modes in Detailed Description of the Operating Modes on page 6-43.

Therefore, the baud rate generators have to provide a "baud rate clock" to the serial interface which - there divided by 16 - results in the actual "baud rate". However, all formulas given in the following section already include the factor and calculate the final baud rate.

Mode 0

The baud rate in mode 0 is fixed:

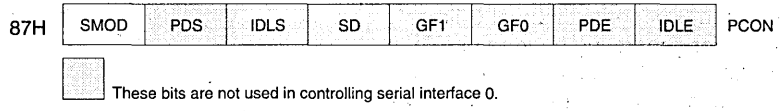
$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

Mode 2

The baud rate in mode 2 depends on the value of bit SMOD in special function register PCON (see Figure 20 Special Function Register PCON (Address 87H)). If SMOD = 0 (which is the value after reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is1/32 of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{SMOD}}{64} \times \text{oscillator frequency}$$

Figure 20 Special Function Register PCON (Address 87H)



Bit	Function
SMOD	When set, the baud rate of serial interface 0 in modes 1, 2, 3 is doubled.

Modes 1 and 3

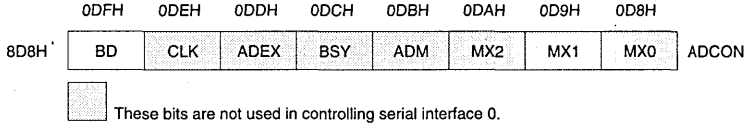
In these modes the baud rate is variable and can be generated alternatively by a dedicated baud rate generator or by timer 1.

Using the baud rate generator:

In modes 1 and 3, the SAB 80C517 can use the internal baud rate generator for serial interface 0. To enable this feature, bit BD (bit 7 of special function register ADCON0) must be set (see Figure 21 Special Function Register ADCON0 (Address 0D8H)). This baud rate generator divides the oscillator frequency by 2500. Bit SMOD (PCON.7) also can be used to enable a multiply-by-two prescaler (see Figure 20 Special Function Register PCON (Address 87H)). At 12-MHz oscillator frequency, the commonly used baud rates 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) are available. The baud rate is determined by SMOD and the oscillator frequency as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{2469} \times \text{oscillator frequency}$$

Figure 21 Special Function Register ADCON0 (Address 0D8H)



Bit	Function
BD	Baud rate enable. When set, the baud rate in modes 1 and 3 of serial interface 0 is taken from a dedicated prescaler. Standard baud rates 4800 and 9600 baud at 12-MHz oscillator frequency can be achieved.

Using timer 1 to generate baud rates:

In mode 1 and 3 of serial channel 0 timer 1 can be used for generating baud rates. Then the baud rate is determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{32} * \text{(timer 1 overflow rate)}$$

The timer 1 interrupt is usually disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its operating modes. In the most typical applications, it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In the case, the baud rate is given by the formula:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} * \text{oscillator frequency}}{32 \times 12 \times (256 - (TH1))}$$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001B), and using the timer 1 interrupt for a 16-bit software reload.

Table 4 Timer 1 Generated Commonly Used Baud Rates lists various commonly used baud rates and shows how they can be obtained from timer 1.

Table 4 Timer 1 Generated Commonly Used Baud Rates

Baud Rate		f_{osc} (MHz)	SMOD	Timer 1		
				C/T	Mode	Reload Value
Mode 1, 3:	62.5 Kbaud	12.0	1	0	2	FFH
	19.2 Kbaud	11.059	1	0	2	FDH
	9.6 Kbaud	11.059	0	0	2	FDH
	4.8 Kbaud	11.059	0	0	2	FAH
	2.4 Kbaud	11.059	0	0	2	F4H
	1.2 Kbaud	11.059	0	0	2	E8H
	110 Baud	6.0	0	0	2	72H
	110 Baud	12.0	0	0	1	FEEBH

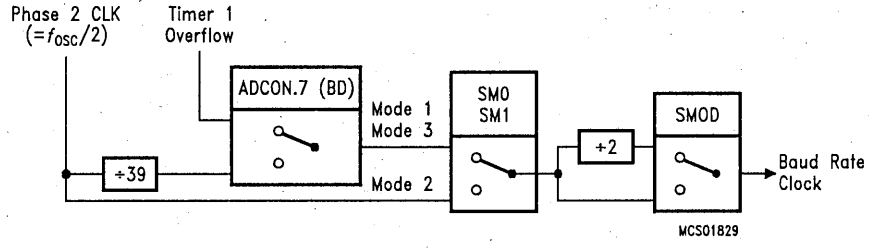
Figure 22 Generation of Baud Rates for Serial Channel 0 shows the mechanisms for baud rate generation of serial channel 0, while Table 5 Baud Rates of Serial Interface 0 summarizes the baud rate formulas for all usual configurations.

Table 5 Baud Rates of Serial Interface 0

Baud Rate Derived from	Interface Mode	Baud Rate
Timer 1 in mode 1 (see Table 4 Timer 1 Generated Commonly Used Baud Rates)	1, 3	$\frac{2^{SMOD}}{2} * \frac{1}{16}$ * (timer 1 overflow rate)
Timer 1 in mode 2 (Table 4 Timer 1 Generated Commonly Used Baud Rates)	1, 3	$\frac{2^{SMOD}}{2} * \frac{1}{16} * \frac{f_{osc}}{12 \times (256 - (TH1))}$
Oscillator	2	$\frac{2^{SMOD}}{2} * \frac{1}{16} * \frac{f_{osc}}{2}$
BD	1, 3	$\frac{2^{SMOD}}{2} * \frac{1}{16} * \frac{f_{osc}}{1250}$

6

Figure 22
Generation of Baud Rates for Serial Channel 0



Serial Interface 1

Operating Modes of Serial Interface 1

The serial interface 1 is an asynchronous channel only and is able to operate in two modes, as an 8-bit or 9-bit UART. These modes, however, correspond to the above mentioned modes 1, 2 and 3 of serial interface 0. The multiprocessor communication feature is identical with this feature in serial interface 0. The serial interface 1 has its own interrupt request flags RI1 and TI1 which have a dedicated interrupt vector location (see Interrupt System on page 6-109 for more details about the interrupts). The baud rate clock for this interface is generated by a dedicated baud rate generator. A more detailed description how to set the baud rate follows in Baud Rates of Serial Channel 0 on page 6-36.

Mode A: 9-bit UART, variable baud rate:

11 bits are transmitted (through TxD1) or received (through RxD1): a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmission, the 9th data bit (TB81 in S1CON) can be assigned to the value of 0 or 1. For example, the parity bit (P in the PSW) could be moved into TB81 or a second stop bit by setting TB81 to 1. On reception the 9th data bit goes into RB81 in special function register S0CON, while the stop bit is ignored. In fact, mode A of serial interface 1 is identical with mode 2 or 3 of serial interface 0 in all respects except the baud rate generation (see Baud Rates of Serial Channel 0 on page 6-36).

Mode B: 8-bit UART, variable baud rate:

10 bits are transmitted (through TxD1) or received (through RxD1): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB81 in special function register S1CON. In fact, mode B of serial interface 1 is identical with mode 1 of serial interface 0 in all respects except for the baud rate generation (see Baud Rates of Serial Channel 0 on page 6-36).

In both modes, transmission is initiated by any instruction that uses S1BUF as a destination register. Reception is initiated by the incoming start bit if REN1 = 1. The serial interfaces also provide interrupt requests when a transmission or a reception of a frame has completed. The corresponding interrupt request flags for serial interface 1 are TI1 or RI1, resp. See Interrupt System on page 6-109 for more details about the interrupt structure. The interrupt request flags TI1 and RI1 can also be used for polling the serial interface 1 if the serial interrupt shall not be used (i.e. serial interrupt 1 not enabled).

The control and status bits of the serial channel 1 in special function register S1CON are illustrated in Figure 23 Special Function Register S1CON (Address 9BH). Figure 24 Special Function Register S1BUF (Address 9CH) shows the special function register S1BUF which is the data register for receive and transmit. Note that these special function registers are not bit-addressable. Due to this fact bit instructions cannot be used for manipulating these registers. This is important especially for S1CON where a polling and resetting of the RI1 or TI1 request flag cannot be performed by JNB and CLR instructions but must be done by a sequence of byte instructions, e.g.:

LOOP:	MOV	A,S1CON	
	JNB	ACC.0,LOOP	;Testing of RI1
	ANL	S1CON,#0FEH	;Resetting of RI1

Figure 23 Special Function Register S1CON (Address $9B_H$)

$9B_H$	SM	–	SM21	REN1	TB81	RB81	TI1	RI1	S1CON

Bit	Function
SM	SM = 0: serial mode A; 9-bit UART SM = 1: serial mode B; 8-bit UART
SM21	Enables the multiprocessor communication feature in mode A. If SM21 is set to 1, RI1 will not be activated if the received 9th data bit (RB81) is 0. In mode B, if SM21 = 1, RI1 will not be activated if a valid stop bit was not received.
REN1	Receiver enable of interface 1. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB81	Transmitter bit 8 of interface 1. Is the 9th data bit that will be transmitted in mode A. Set or cleared by software as desired.
RB81	Receiver bit 8 of interface 1. Is the 9th data bit that was received in mode A. In mode B, if SM21 = 0, RB81 is the stop bit that was received.
TI1	Transmitter interrupt of interface 1. Is the transmit interrupt flag. Set by hardware at the beginning of the stop bit in any serial transmission. Must be cleared by software.
RI1	Receiver interrupt of interface 1. Is the receive interrupt flag. Set by hardware at the halfway through the stop bit time in any serial reception. Must be cleared by software.

Figure 24 Special Function Register S1BUF (Address 9CH)

9CH	Serial interface 1 buffer register	S1BUF

Receive and transmit buffer of serial interface 1. Writing to S1BUF loads the transmit register and initiates transmission. Reading out S1BUF accesses a physically separate receive register.

Multiprocessor Communication Feature

Mode A of the serial interface 1 has a special provision for multiprocessor communication. In this mode, 9 data bits are received. The 9th bit goes into RB81. Then follows a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt (i.e. the request flag RI1 is set) will be activated only if RB81 = 1. This feature is enabled by setting bit SM21 in S1CON. A way to use this feature in multiprocessor communications is as follows.

If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM21 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM21 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave is setting SM21 again. The slaves that were not addressed leave their SM21 set and go on about their business, ignoring the incoming data bytes.

In mode B SM21 can be used to check the validity of the stop bit. If SM21 = 1 in mode B, the receive interrupt will not be activated unless a valid stop bit is received.

Baud Rates of Serial Channel 1

As already mentioned serial interface 1 uses its own dedicated baud rate generator for baud rate generation in both operating modes (see Figure 25 Baud Rate Generator for Serial Interface 1).

This baud rate generator consists of a free running 8-bit timer with $f_{osc}/2$ input frequency. The timer is automatically reloaded at overflow by the contents of register S1REL (see Figure 26 Special Function Register S1REL (Address 9DH)). The timer must be started by writing the desired reload value to register S1REL. The baud rate in operating modes A and B can be determined by following formula:

$$\text{Mode A, B baud rate} = \frac{\text{oscillator frequency}}{32 \times (256 - \text{S1REL})}$$

At 12-MHz oscillator frequency a baud rate range from about 1.5 kbaud up to 375 kbaud is covered. Using the fast baud rates offers the same functionality as the operating mode 2 in serial interface 0 with its fixed baud rates.

Figure 25
Baud Rate Generator for Serial Interface 1

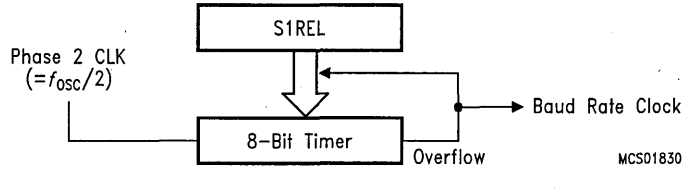


Figure 26 Special Function Register S1REL (Address 9DH)

8-bit reload register for baud rate generator of serial interface 1.

Detailed Description of the Operating Modes

The following sections give a more detailed description of the several operating modes of the two serial interfaces.

The Serial Interfaces on page 6-33 and A/D Converter Timing on page 6-61. apply to both of the serial interfaces. The description of the synchronous mode 0 and the asynchronous mode 2 refers only to serial interface 0.

Mode 0, Synchronous Mode (Serial Interface 0)

Serial data enters and exits through RxD0. TxD0 outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Figures 27 a) and b) show a simplified functional diagram of the serial port in mode 0, and associated timing.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The "write-to-S0BUF" signal at S6P2 also loads a 1 into the 9th bit position of the transmit shift register and tells the TX control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write-to-S0BUF" and activation of SEND.

SEND enables the output of the shift register to the alternate output function line P3.0, and also enables SHIFT CLOCK to the alternate output function line P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2, while the interface is transmitting. Before and after transmission SHIFT CLOCK remains high. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register is shifted one position to the right.

As data bits shift to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX control block to do one last shift and then deactivates SEND and sets TI0. Both of these actions occur at S1P1 in the 10th machine cycle after "write-to-S0BUF".

Reception is initiated by the condition REN0 = 1 and RI0 = 0. At S6P2 in the next machine cycle, the RX control unit writes the bits 1111 1110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 in every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted one position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 in the same machine cycle.

As data bits come in from the right, 1 s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX control block to do one last shift and load S0BUF. At S1P1 in the 10th machine cycle after the write to S0CON that cleared RI0, RECEIVE is cleared and RI0 is set.

Mode 1/Mode B, 8-Bit UART (Serial Interfaces 0 and 1)

Ten bits are transmitted (through TxD0 or TxD1), or received (through RxD0 or RxD1): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception through RxD0, the stop bit goes into RB80 (S0CON), on reception through RxD1, RB81 (S1C0N) stores the stop bit.

The baud rate for serial interface 0 is determined by the timer 1 overflow rate or by the internal baud rate generator of serial interface 0. Serial interface 1 receives the baud rate clock from its own baud rate generator.

Figures 28 a) and b) show a simplified functional diagram of both serial channels in mode 1 or mode B, resp. The generation of the baud rate clock by the various timers is described in Serial Interfaces on page 6-33.

Transmission is initiated by any instruction that uses S0BUF/S1BUF as a destination register. The "write-to-S0BUF/S1BUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX control block that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next roll-over in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the "write-to-S0BUF/S1BUF" signal).

The transmission begins with activation of SEND, which puts the start bit to TxD0/TxD1. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD0/TxD1. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just left of the MSB, and all positions to the left of that contain zero. This condition flags the TX control to do one last shift and then deactivate SEND and set TI0/TI1. This occurs at the 10th divide-by-16 rollover after "write-to-S0BUF/S1BUF".

Reception is initiated by a detected 1-to-0 transition at RxD0/RxD1. For this purpose RxD0/RxD1 is sampled at a rate of 16 times whatever baud rate has been established. When a reception is detected, the divide-by-16 counter is immediately reset, and 1 FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollover with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16 counter states. At the 7th, 8th and 9th counter state of each bit time, the bit detector samples the value of RxD0/RxD1. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come from the right, 1's shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1/B is a 9-bit register), it flags the RX control block to do one last shift. The signal to load S0BUF/S1BUF and RB80/RB81, and to set RI0/RI1 will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI0/RI1 = 0, and

2) either SM20/SM21 = 0 or the received stop bit = 1

If either of these two conditions is not met the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB80/RB81, the 8 data bits go into S0BUF/S1BUF, and RI0/RI1 is activated. At this time, no matter whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD0/RxD1.

Mode 2, 9-Bit UART (Serial Interface 0)

Mode 2 is functionally identical to mode 3 (see below). The only exception is, that in mode 2 the baud rate can be programmed to two fixed quantities: either 1/32 or 1/64 of the oscillator frequency. Note that serial interface 0 cannot achieve this baud rate in mode 3. Its baud rate clock is generated by timer 1, which is incremented by a rate of $f_{osc}/12$. The dedicated baud rate generator of serial interface 1 however is clocked by a $f_{osc}/2$-signal and so its maximum baud rate is $f_{osc}/32$.

Mode 3 / Mode A, 9-Bit UART (Serial Interfaces 0 and 1)

Eleven bits are transmitted (through TxD0/ TxD1), or received (through RxD0/RxD1): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmission, the 9th data bit (TB80/TB81) can be assigned the value of 0 or 1. On reception the 9th data bit goes into RB80/ RB81 in S0CON/S1CON.

Figures 28 a) and b) show a functional diagram of the serial interfaces in mode 2 and 3 or mode A, resp. and associated timing. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses S0BUF/S1BUF as a destination register. The "write to S0BUF/S1BUF" signal also loads TB80/TB81 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus the bit times are synchronized to the divide-by-16 counter, and not to the "write-to-S0BUF/S1BUF" signal).

The transmission begins with the activation of SEND, which puts the start bit to TxD0/TxD1. One bit time later, DATA is activated which enables the output bit of transmit shift register to TxD0/TxD1. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data shift out to the right, zeros are clocked in from the left. When TB80/TB81 is at the output position of the shift register, then the stop bit is just left of the TB80/TB81, and all positions to the left of that contain zeros.

This condition flags the TX control unit to do one last shift and then deactivate SEND and set TI0/TI1. This occurs at the 11th divide-by-16 rollover after "write-to-S0BUF/S1BUF".

Reception is initiated by a detected 1-to-0 transition at RxD0/RxD1. For this purpose RxD0/RxD1 is sampled of a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FH is written to the input shift register.

At the 7th, 8th and 9th counter state of each bit time, the bit detector samples the value of RxD0/RxD1. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come from the right, 1's shift out to the left. When the start bit arrives at the leftmost position in the shift register (which is a 9-bit register), it flags the RX control block to do one last shift, load S0BUF/S1BUF and RB80/ RB81, and set RI0/RI1. The signal to load S0BUF/S1BUF and RB80/RB81, and to set RI0/RI1, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

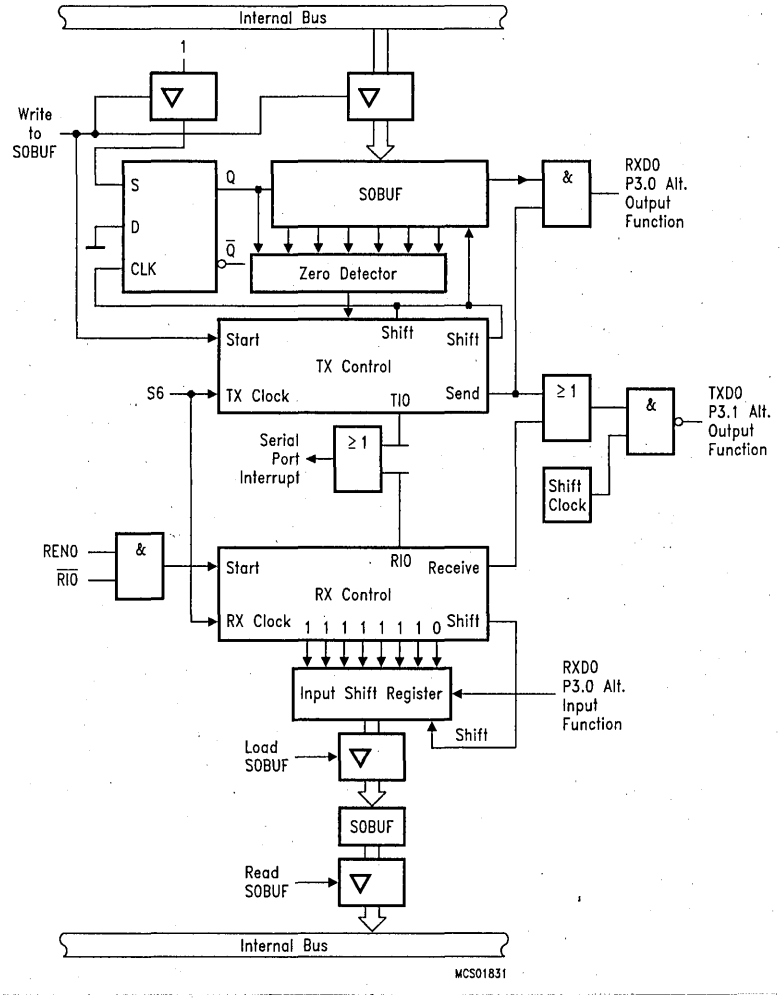
1) RI0/RI1 = 0, and

2) either SM20/SM21 = 0 or the received 9th data bit = 1

If either one of these two conditions is not met, the received frame is irretrievably lost, and RI0/RI1 is not set. If both conditions are met, the received 9th data bit goes into RB80/ RB81, the first 8 data bits go into S0BUF/

S1BUF. One bit time later, no matter whether the above conditions are met or not, the unit goes back to look for a 1-to-0 transition at the RxD0/RxD1 input.

Figure 27 a)
Functional Diagram - Serial Interface 0, Mode 0



Note that the value of the received stop bit is irrelevant to S0BUF/S1BUF, RB80/RB81, or RI0/RI1.

Figure 27 b)
Timing Diagram - Serial Interface 0, Mode 0

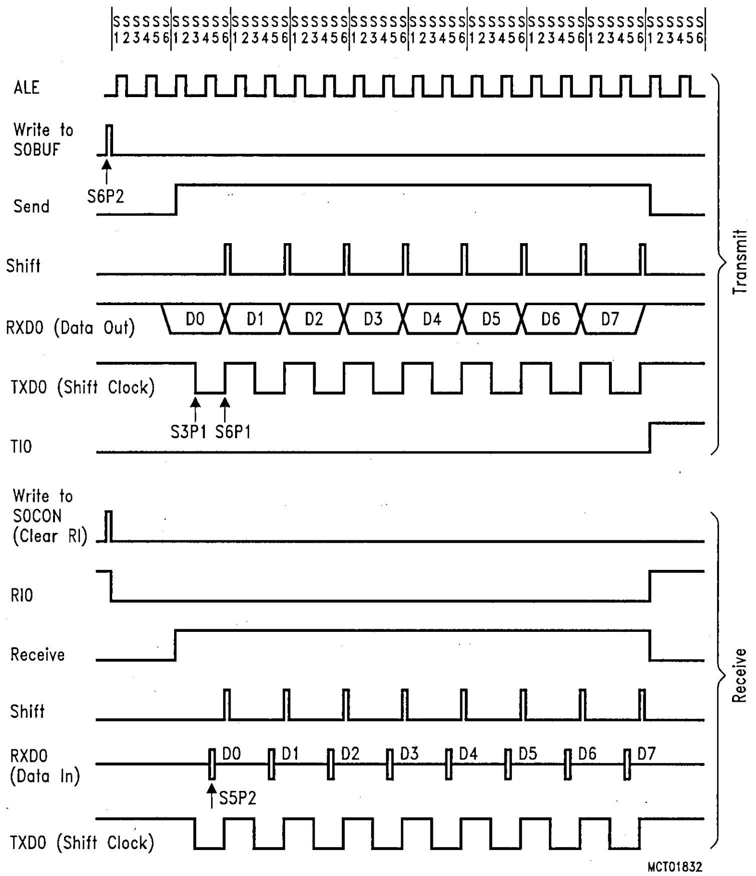
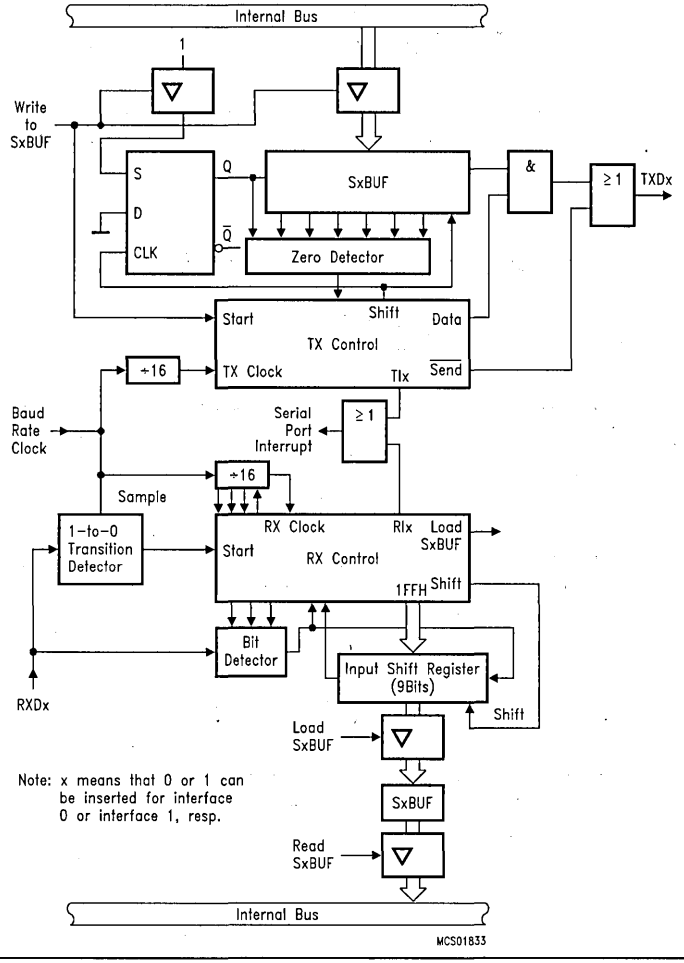
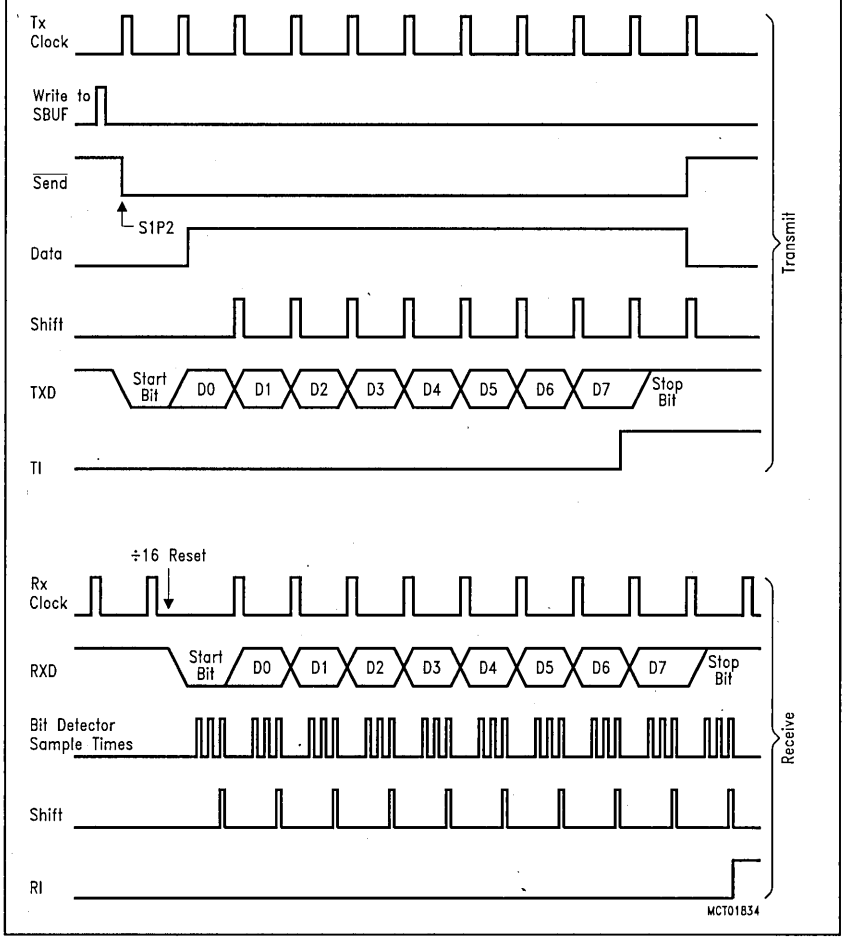


Figure 28 a)
Functional Diagram - Serial Interfaces 0 and 1, Mode 1/Mode B



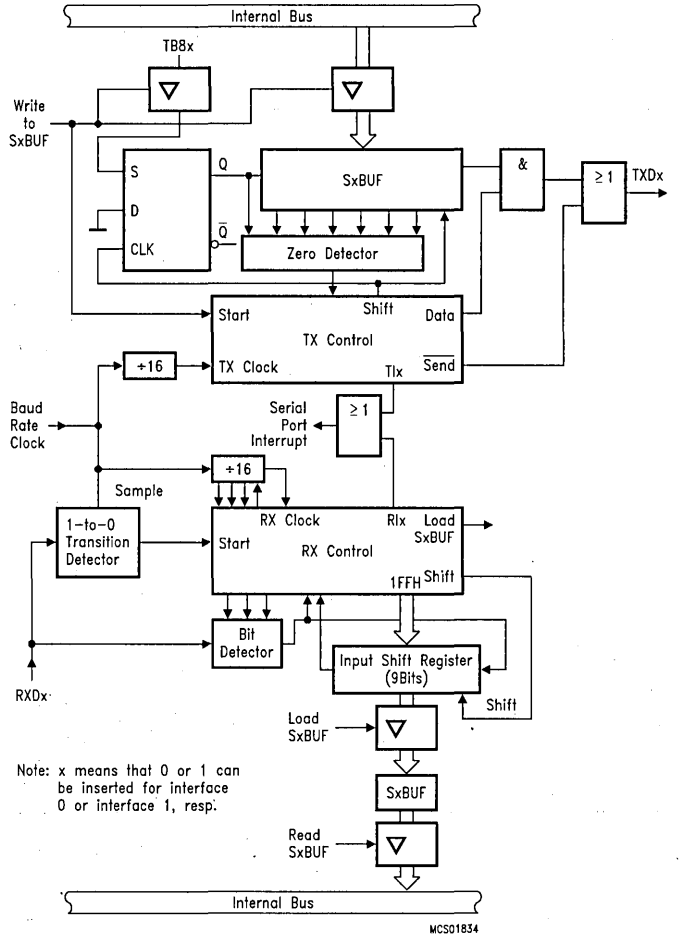
Note: x means that 0 or 1 can be inserted for interface 0 or interface 1, resp.

Figure 28 b)
Timing Diagram - Serial Interfaces 0 and 1, Mode 1/Mode B



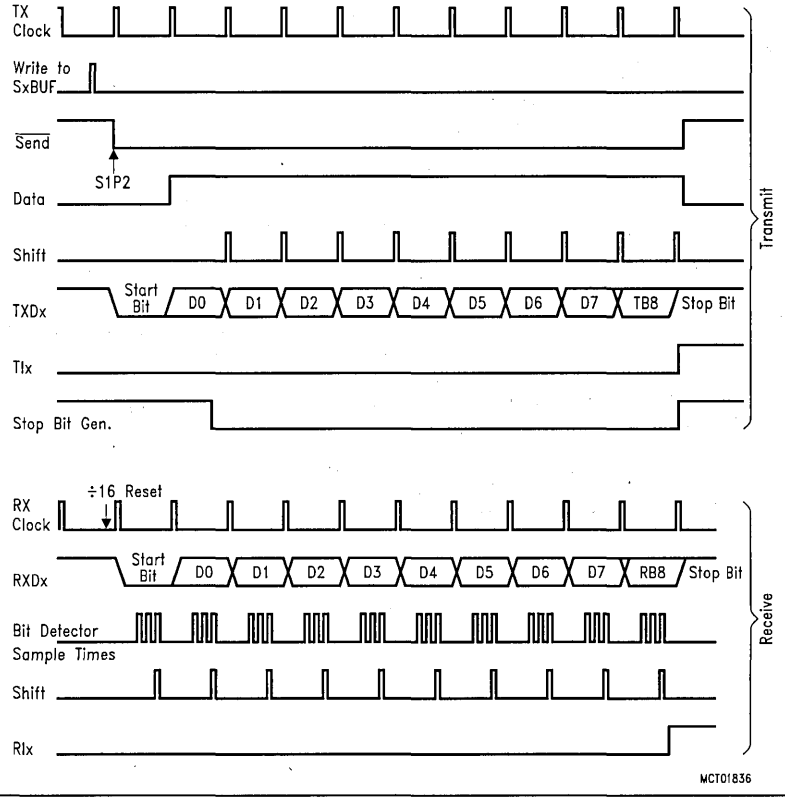
6

Figure 29 a)
Functional Diagram - Serial Interface 0 and 1, Modes 2 and 3/ Mode A



Note: x means that 0 or 1 can be inserted for interface 0 or interface 1, resp.

Figure 29 b)
Timing Diagram - Serial Interfaces 0 and 1, Modes 2 and 3/ Mode A



Timer 0 and Timer 1

The SAB 80C517 has a number of general purpose 16-bit timer/counters: timer 0, timer 1, timer 2 and the compare timer (timer 2 and the compare timer are discussed separately in The Compare/Capture Unit (CCU) on page 6-62). Timer/counter 0 and 1 are fully compatible with timer/counters 0 and 1 of the SAB 8051 and can be used in the same operating modes.

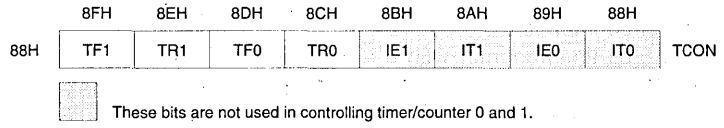
Timer/counter 0 and 1 which are discussed in this section can be configured to operate either as timers or event counters:

In "timer" function, the register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In "counter" function, the register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 and P3.5, resp.). In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

In addition to the "timer" and "counter" selection, timer/counters 0 and 1 have four operating modes from which to select.

Figure 30 Special Function Register TCON (Address 88H)



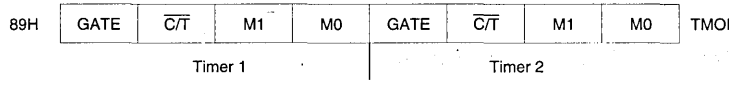
Bit	Function
TR0	Timer 0 run control bit. Set/cleared by software to turn timer/counter 0 On/Off.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 run control bit. Set/cleared by software to turn timer/counter 1 On/Off.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer/counter 0, TH1 and TL1 for timer/counter 1) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD, shown in Figure 30 Special Function Register TCON (Address 88H) and Figure 31 Special Function Register TMOD (Address 89H).

In the following descriptions the symbols TH0 and TL0 are used specify the high-byte and low-byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are described and shown for timer 0. If not explicitly noted, this applies also to timer 1.

6

Figure 31 Special Function Register TMOD (Address 89H)



Timer/counter 0/1 mode control register

Bit		Symbol
Gate		Gating control. When set, timer/counter "x" is enabled only while "INTx" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
C/T#		Counter or timer select bit. Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).
M1	M0	
1	0	8-bit timer/counter "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH00 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops

Mode 0

Putting either timer/counter into mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. Figure 32 Timer/Counter 0/1, Mode 0: 13-Bit Timer/Counter shows the mode 0 operation.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1's to all 0's, it sets the timer overflow flag TF0. The overflow flag TF0 then can be used to request an interrupt (see Interrupt System on page 6-109 for details about the interrupt structure). The counted input is enabled to the timer when TR0 = 1 and either GATE = 0 or $\overline{INT0}$ = 1 (setting GATE = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements). TR0 is a control bit in the special function register TCON; GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR1, TF1, TH1, TL1, and $\overline{INT1}$ for the corresponding timer 1 signals in Figure 32 Timer/Counter 0/1, Mode 0: 13-Bit Timer/Counter. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as mode 0, except that the timer register is run with all 16 bits. Mode 1 is shown in Figure 33 Timer/Counter 0/1, Mode 1: 16-Bit Timer/Counter.

Figure 32
Timer/Counter 0/1, Mode 0: 13-Bit Timer/Counter

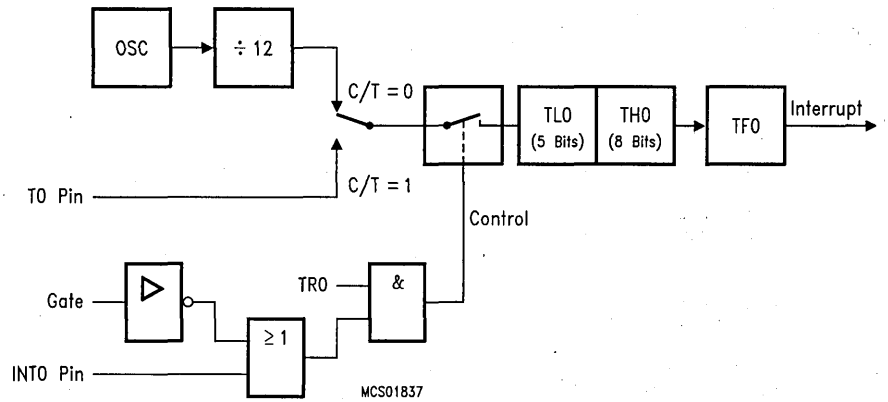
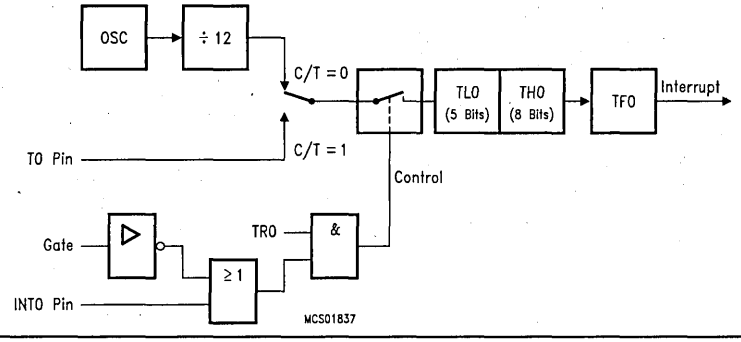


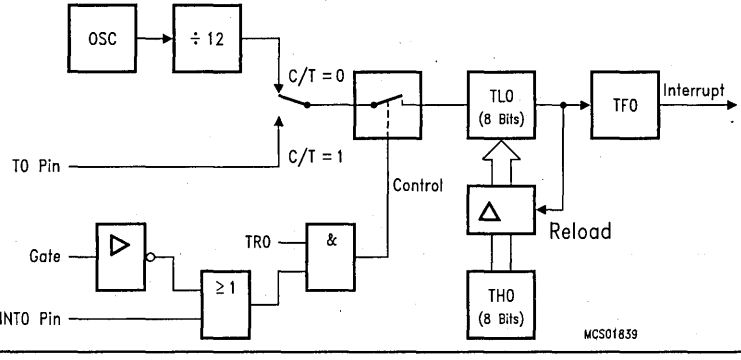
Figure 33
Timer/Counter 0/1, Mode 1: 16-Bit Timer/Counter



Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 34 Timer/Counter 0/1, Mode 2: 8-Bit Timer/Counter with Auto Reload. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

Figure 34
Timer/Counter 0/1, Mode 2: 8-Bit Timer/Counter with Auto Reload



Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on timer 0 is shown in Figure 35 Timer/Counter 0, Mode 3: Two 8-Bit Timer/Counter. TL0 uses the timer 0 control bits: $\overline{C/T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial channel as a baud rate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

A/D Converter

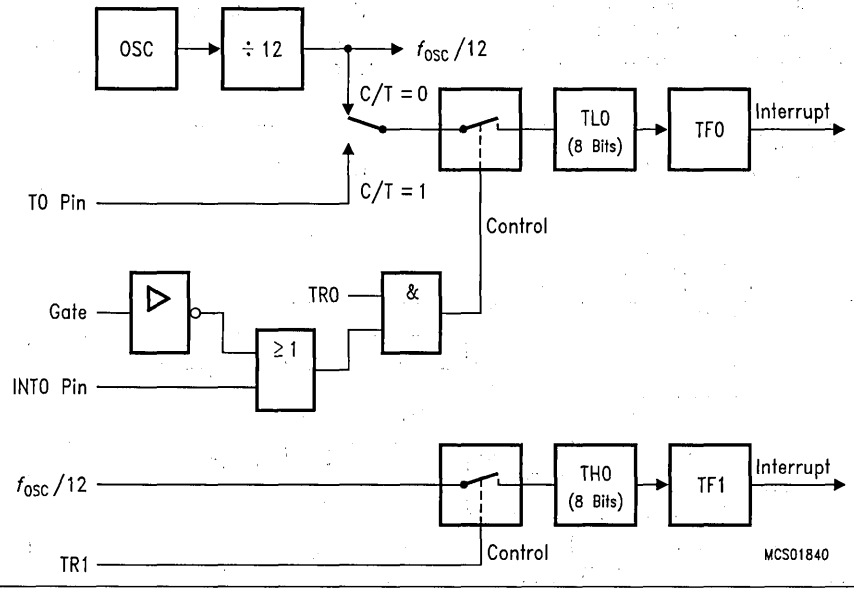
The SAB 80C517 provides an A/D converter with the following features:

- 12 multiplexed input channels, which can also be used as digital inputs (port 7, port 8)
- Programmable internal reference voltages (16 steps each) via resistor array
- 8-bit resolution within the selected reference voltage range
- 13 microseconds conversion time (including sample time) at 12-MHz oscillator frequency
- Selectable external or internal start-of-conversion trigger
- Interrupt request generation after each conversion

For the conversion, the method of successive approximation via capacitor array is used. The externally applied reference voltage range has to be held on a fixed value within the specifications (see section "A/D Converter Characteristics" in the data sheet). The internal reference voltages can be varied to reduce the reference voltage range of the A/D converter and thus to achieve a higher resolution.

Figure 36 Block Diagram of the A/D Converter shows a block diagram of the A/D converter. There are four user-accessible special function registers: ADCON0, ADCON1 (A/D converter control registers), ADDAT (A/D converter data register) and DAPR (D/A converter program register) for the programmable reference voltages. The analog input channels (port 7 and port 8) can also be used for digital input; refer also to Parallel I/O on page 6-27.

Figure 35
Timer/Counter 0, Mode 3: Two 8-Bit Timer/Counter



Function and Control

Initialization and Input Channel Selection

Special function register ADCON0 which is illustrated in Figure 37 Special Function Register ADCON0 (Address 0D8H) is used to set the operating modes, to check the status, and to select one of eight analog input channels. Special function register ADCON1 (Figure 38 Special Function RegisterADCON1 (Address 0DCH)) controls the selection of all twelve input channels.

Register ADCON0 contains two mode bits. Bit ADM is used to choose the single or continuous conversion mode. In single conversion mode only one conversion is performed after starting, while in continuous conversion mode after the first start a new conversion is automatically started on completion of the previous one.

An externally controlled conversion can be achieved by setting the bit ADEX. In this mode on single conversion is triggered by a 1-to-0 transition at pin P6.0/ADST, if ADM is 0. P6.0/ADST is sampled assuring S5P2 of every machine cycle. When the samples show a logic high in one cycle and a logic low in the next cycle the transition is detected and the conversion is started. When ADM and ADEX is set, a continuous conversion is started when pin P6.0/ADST sees a low level; the conversion is stopped when the pin P6.0/ADST goes back to high. The last commenced conversion during low level will be completed.

The busy flag BSY (ADCON0.4) is automatically set when a conversion is in progress. After completion of the conversion it is reset by hardware. This flag can be read only, a write has no effect.There is also an interrupt request flag IADC (IRCON.0) that is set when a conversion is completed. See Interrupt System on page 6-109 for more details about the interrupt structure.

Figure 36
Block Diagram of the A/D Converter

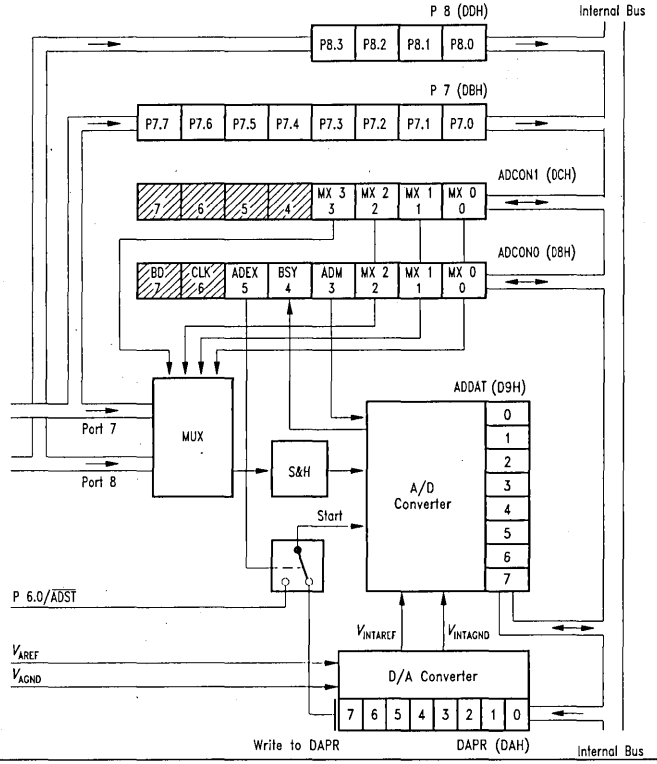


Figure 37
Special Function Register ADCON0 (Address 0D8H)

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0

☐ These bits are not used in controlling A/D converter functions

Bit	Function
MX0 MX1 MX2 MX3	Select 12 input channels of the A/D converter.
ADM	A/D conversion mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	Busy flag. This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is completed.
ADEX	Internal/external start of conversion. When set, the external start of conversion by P6.0/$\overline{ADST}$ is enabled.

Figure 38
Special Function RegisterADCON1 (Address 0DCH)

0DCH	–	–	–	–	MX3	MX2	MX1	MX0	ACON1

A/D converter control register 1. It contains channel selection bits MX0 to MX3. Bits MX0 to MX2 can be written or read either in ADCON0 or in ADCON1.

Table 6
Selection of the Analog Input Channels

MX3	MX2	MX1	MX0	Selected Channel	Pin
0	0	0	0	Analog input 0	P7.0
0	0	0	1	Analog input 1	P7.1
0	0	1	0	Analog input 2	P7.2
0	0	1	1	Analog input 3	P7.3
0	1	0	0	Analog input 4	P7.4
0	1	0	1	Analog input 5	P7.5
0	1	1	0	Analog input 6	P7.6
0	1	1	1	Analog input 7	P7.7
1	X*)	0	0	Analog input 8	P8.0
1	X	0	1	Analog input 9	P8.1
1	X	1	0	Analog input 10	P8.2
1	X	1	1	Analog input 11	P8.3

*) X means that the value may be 1 or 0.

The bits MX0 to MX2 in special function register ADCON0 and the bits MX0 to MX3 in ADCON1 are used for selection of the analog input channel. Table 6 Selection of the Analog Input Channels lists the selected input channels. The bits MX0 to MX2 are represented in both the registers ADCON0 and ADCON1; however, these bits are present only once; it has the same effect irrespective of whether they are accessed via ADCON0 or ADCON1. This is done in order to maintain software compatibility to the SAB 80(C)515. In this device there are only eight input channels which are selected by MX0 to MX2 in ADCON0. Thus, a program written for the SAB 80(C)515 selects one of the lower eight input channels (port 7) if the bit MX3 is reset which is the default value after reset. (For clarity: In the SAB 80(C)515 the analog input channel is called port 6 or AN0 to AN7, resp. However, it is found on the same address (0DBH) as the SAB 80C517's port 7.)

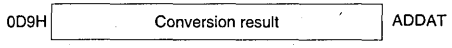
If all 12 multiplexed input channels are required register ADCON1 is to be used. It contains a four-bit field to select one of all 12 input channels, the eight inputs at port 7 and the four inputs at port 8. Thus, there are two methods of selecting a channel of port 7 and it does not matter which is used: if a new channel is selected in ADCON1 the change is automatically done in the corresponding bits MX0 to MX2 in ADCON0 and vice versa. If bit MX3 is set, the additional analog inputs at port 8 are used. MX0 and MX1 then determine which channel of port 8 is being selected (see Table 6 Selection of the Analog Input Channels).

Ports P7 and P8 are dual purpose input ports. If the input voltage meets the specified logic levels, they can be used as digital inputs as well regardless of whether the pin levels are sampled by the A/D converter at the same time.

The special function register ADDAT (Figure 39 Special Function Register ADDAT (Address 0D9H)) holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. ADDAT can be read or written under

software control. If the A/D converter of the SAB 80C517 is not used, register ADDAT can be used as an additional general purpose register.

Figure 39
Special Function Register ADDAT (Address 0D9H)



This register contains the 8-bit conversion result.

Start of Conversion

An internal start of conversion (ADEX = 0) is triggered by a write-to-DAPR instruction. The start procedure itself is independent of the value which is written to DAPR. However, the value in DAPR determines which internal reference voltages are used for the conversion (see Reference Voltages on page 6-59). When single conversion mode is selected (ADM = 0) only one conversion is performed. In continuous mode after completion of a conversion a new conversion is triggered automatically, until bit ADM is reset.

When external start of conversion is selected a write-to-DAPR will not start the conversion; in this case, conversion starts when a falling edge at pin P6.0/$\overline{\text{ADST}}$ is detected. In single conversion mode one conversion is performed until the next falling edge at P6.0/$\overline{\text{ADST}}$ is recognized. In continuous mode new conversions are started automatically as long as pin P6.0/$\overline{\text{ADST}}$ is on low level. This is done until P6.0/$\overline{\text{ADST}}$ goes to logic high level; in this case the last commenced conversion is completed.

Reference Voltages

The SAB 80C517 has two pins to which a reference voltage range for the on-chip A/D converter is applied (pin V_{AREF} for the upper voltage and pin V_{AGND} for the lower voltage). In contrast to conventional A/D converters it is now possible to use not only these externally applied reference voltages for the conversion but also internally generated reference voltages which are derived from the externally applied ones. For this purpose a resistor ladder provides 16 equidistant voltage levels between V_{AREF} and V_{AGND}. These steps can individually be assigned as upper and lower reference voltage for the converter itself. These internally generated reference voltages are called $V_{INTAREF}$ and $V_{INTAGND}$. The internal reference voltage programming can be thought of as a programmable "D/A converter" which provides the voltages $V_{INTAREF}$ and $V_{INTAGND}$ for the A/D converter itself.

The SFR DAPR (see Figure 40 Special Function Register DAPR (Address DAH)) is provided for programming the internal reference voltages $V_{INTAREF}$ and $V_{INTAGND}$. For this purpose the internal reference voltages can be programmed in steps of 1/16 of the external reference voltages ($V_{AREF} - V_{AGND}$) by four bits each in register DAPR. Bits 0 to 3 specify $V_{INTAGND}$, while bits 4 to 7 specify $V_{INTAREF}$. A minimum of 1 V difference is required between the internal reference voltages $V_{INTAREF}$ and $V_{INTAGND}$ for proper operation of the A/D converter. This means, for example, in the case where V_{AREF} is 5 V and V_{AGND} is 0 V, there must be at least four steps difference between the internal reference voltages $V_{INTAREF}$ and $V_{INTAGND}$.

The values of $V_{IntAGND}$ and $V_{IntAREF}$ are given by the formulas:

$$V_{INTAGND} = V_{AGND} + \frac{\text{DAPR (.3-.0)}}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < CH;

$$V_{INTAREF} = V_{AGND} + \frac{\text{DAPR (.7-.4)}}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3H;

DAPR (.3-.0) is the contents of the low-order nibble, and DAPR (.7-.4) the contents of the high-order nibble of DAPR.

Figure 40
Special Function Register DAPR (Address DAH)

0DAH	Programming of $V_{INTAREF}$	Programming of $V_{INTAGND}$	DAPR

D/A converter program register. Each 4-bit nibble is used to program the internal reference voltages. Write-access to DAPR starts conversion.

$$V_{INTAGND} = V_{AGND} + \frac{DAPR\ (.3\text{-}.0)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < 13;

$$V_{INTAREF} = V_{AGND} + \frac{DAPR\ (.7\text{-}.4)}{16}\ (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3;

If DAPR (.3-.0) or DAPR (.7-.4) = 0, the internal reference voltages correspond to the external reference voltages V_{AGND} and V_{AREF}, respectively.

If $V_{AINPUT} > V_{INTAREF}$, the conversion result is 0FFH, if $V_{AINPUT} < V_{INTAGND}$, the conversion result is 00H (V_{AINPUT} is the analog input voltage).

If the external reference voltages V_{AGND} = 0 V and V_{AREF} = +5 V (with respect to V_{SS} and V_{CC}) are applied, then the following internal reference voltages $V_{INTAGND}$ and $V_{INTAREF}$ shown in Table 7 Adjustable Internal Reference Voltages can be adjusted via the special function register DAPR.

Table 7
Adjustable Internal Reference Voltages

Step	DAPR (.3-.0) DAPR (.7-.4)	$V_{INTAGND}$	$V_{INTAREF}$
0	0000	0.0	5.0
1	0001	0.3125	–
2	0010	0.625	–
3	0011	0.9375	–
4	0100	1.25	1.25
5	0101	1.5625	1.5625
6	0110	1.875	1.875
7	0111	2.1875	2.1875
8	1000	2.5	2.5
9	1001	2.8125	2.8125
10	1010	3.125	3.125
11	1011	3.4375	3.4375
12	1100	3.75	3.75
13	1101	–	4.0625
14	1110	–	4.375
15	1111	–	4.68754

The programmability of the internal reference voltages allows adjusting the internal voltage range to the range of the external analog input voltage or it may be used to increase the resolution of the converted analog input voltage by starting a second conversion with a compressed internal reference voltage range close to the previously measured analog value. Figure 41 Adjusting the Internal Reference Voltages to the Range of the External Analog Input Voltages and Figure 42 Increasing the Resolution by a Second Conversion illustrate these applications.

Figure 41
Adjusting the Internal Reference Voltages to the Range of the External Analog Input Voltages

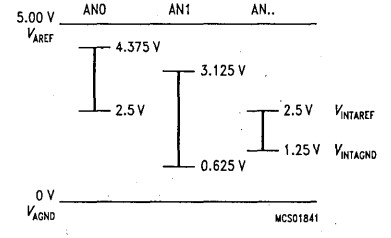
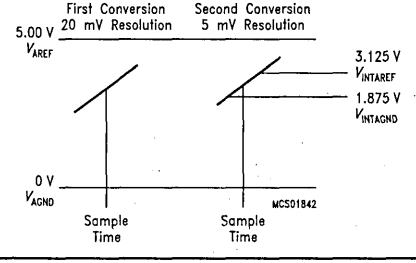


Figure 42
Increasing the Resolution by a Second Conversion



The external reference voltage supply need only be applied when the A/D converter is used, otherwise the pins V_{AREF} and V_{AGND} may be left unconnected. The reference voltage supply has to meet some requirements concerning the level of V_{AGND} and V_{AREF} and the output impedance of the supply voltage (see also "A/D Converter Characteristics" in the data sheet).

The voltage V_{AREF} must meet the following specification:
$V_{AREF} = V_{CC}$ +/– 5 %

The voltage V_{AGND} must meet a similar specification:
$V_{AGND} = V_{SS}$ + /– 0.2 V

The differential output impedance of the analog reference supply voltage should be less than 1 kOhm.

If the above mentioned operating conditions are not met the accuracy of the converter may be decreased.

Furthermore, the analog input voltage V_{AINPUT} must not exceed the range from (V_{AGND} – 0.2 V) to (V_{AREF} + 0.2 V). Otherwise, a static input current might result at the corresponding analog input which will also affect the accuracy of the other input channels.

A/D Converter Timing

A conversion is internally started by writing into special function register DAPR (ADEX = 0). A write-to-DAPR will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle and the busy flag BSY will be set. When external start is selected (ADEX = 1) the conversion starts in the machine cycle following the one where the low level was detected at P6.0/$\overline{\text{ADST}}$.

The conversion procedure is divided into three parts:

Load time (t_L):

During this time the analog input capacitance C_I (see data sheet) must be loaded to the analog input voltage level. The external analog source needs to be strong enough to source the current to load the analog input capacitance during the load time. This causes some restrictions for the impedance of the analog source.

Sample time (t_S):

During this time the internal capacitor array is connected to the selected analog input channel. The sample time includes the load time which is described above. After the load time has passed the selected analog input must be held constant for the rest of the sample time. Otherwise the internal calibration of the comparator circuitry could be affected which might result in a reduced accuracy of the converter. However, in typical applications a voltage change of approx. 200 - 300 mV at the inputs during this time has no effect.

Conversion time (t_C):

The conversion time t_C includes the sample and load time. Thus, t_C is the total time required for one conversion. After the load time and sample time have elapsed, the conversion itself is performed during the rest of t_C. In the last machine cycle the converted result is moved to ADDAT; the busy flag (BSY) is cleared before. The A/D converter interrupt is generated by bit IADC in register IRCON. IADC is already set some cycles before the result is written to ADDAT. The flag IADC is set before the result is available in ADDAT because the shortest possible interrupt latency time is taken into account in order to ensure optimal performance. Thus, the converted result appears at the same time in ADDAT when the first instruction of the interrupt service routine is executed. Similar considerations apply to the timing of the flag BSY where usually a "JB BSY,$" instruction is used for polling.

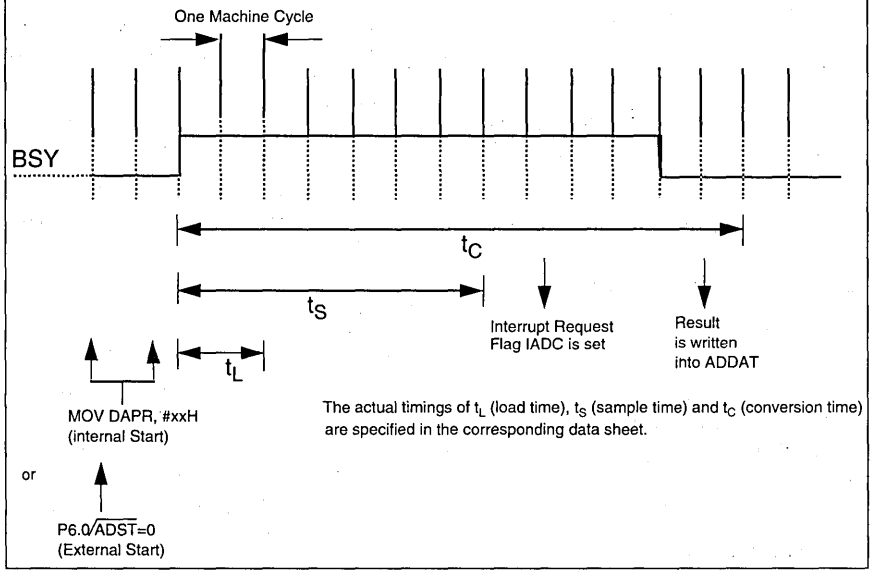
If a continuous conversion is established, the next conversion is automatically started in the machine cycle following the last cycle of the previous conversion.

The Compare/Capture Unit (CCU)

The compare/capture unit is one of the SAB 80C517's most powerful peripheral units for use in all kinds of digital signal generation and event capturing like pulse generation, pulse width modulation, pulse width measuring etc.

The CCU consists of two 16-bit timer/counters with automatic reload feature and an array of 13 compare or compare/capture registers. A set of six control registers is used for flexible adapting of the CCU to a wide variety of user's applications.

Figure 43
Timing Diagram of an A/D Converter



The CCU is the ideal peripheral for various automotive control applications (ignition/ injection control, anti-lock brakes, etc.) as well as for industrial applications (DC, three-phase AC, and stepper motor control, frequency generation, digital-to-analog conversion, process control, etc.)

The detailed description in the following sections refers to the CCU's functional blocks as listed below:

Timer 2 with $f_{osc}/12$ input clock, 2-bit prescaler, (4-bit prescaler, in SAB 80C517 identification mark "BB" or later), 16-bit reload, counter/ gated timer mode and overflow interrupt request.

Compare timer with $f_{osc}/2$ input clock, 8-bit prescaler, 16-bit reload and overflow interrupt request.

Compare/(reload/)capture register array consisting of four different kinds of registers:
one 16-bit compare/reload/capture register,
three 16-bit compare/capture registers,
one 16-bit compare/capture register with additional "concurrent compare" feature,
eight 16-bit compare registers with timer-overflow controlled loading.

Altogether the register array may control up to 21 output lines and can request up to 7 independent interrupts.

For brevity, in the following text all double-byte compare, compare/capture or compare/reload/capture registers are called CMx (x = 0 ... 7), CCx (x = 0 ... 4) or CRC register, respectively.

The block diagram in Figure 44 Block Diagram of the CCU shows the general configuration of the CCU. All CCx registers and the CRC register are exclusively assigned to timer 2. Each of the eight compare registers CM0 through CM7 can either be assigned to timer 2 or to the faster compare timer, e.g. to provide up to 8 PWM channels. The assignment of the CMx registers - which can be done individually for every single register - is combined with an automatic selection of one of the two possible compare modes.

Port 5, port 4 and seven lines of port 1 have alternate functions dedicated to the CCU. These functions are listed in Table 8 Alternate Port Functions of the CCU. Normally each register controls one dedicated output line at the ports. Register CC4 is an exception as it can manipulate up to nine output lines (one at port 1.4 and the other eight at port 5) concurrently. This feature, the "concurrent compare", is described in Timer/Compare Register Configurations in the CCU on page 6-75.

Note that for an alternate input function the port-bit latch has to be programmed with a '1'. For bit latches of port pins that are used as compare outputs, the value to be written to the bit latches depends on the compare mode established.

A list of all special function registers concerned with the CCU is given in Table 9 Special Function Registers of the CCU.

Figure 44
Block Diagram of the CCU

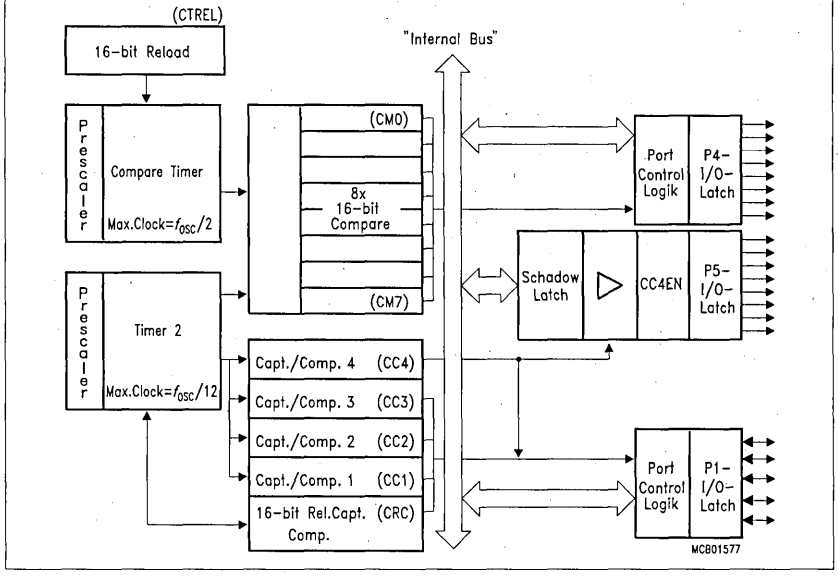


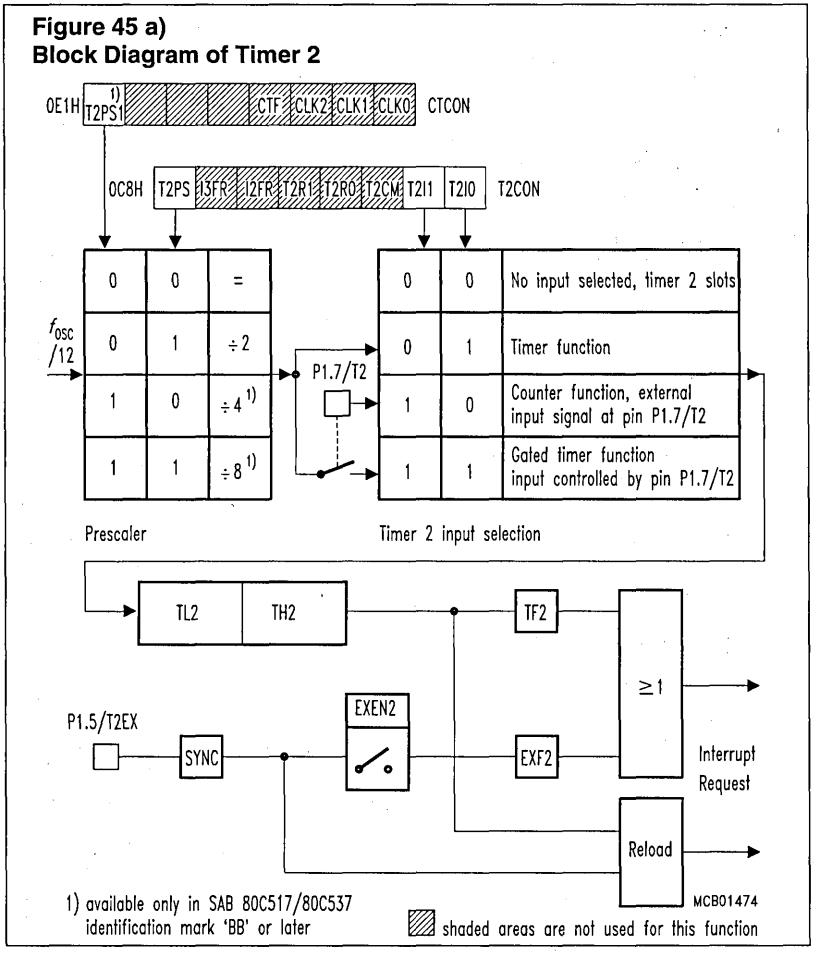
Table 8 Alternate Port Functions of the CCU

Pin Symbol	Pin No.	Input (I) Output (O)	Function
P5.0/CCM0	68	I/O	Concurrent compare 0
P5.1/CCM1	67	I/O	Concurrent compare 1
P5.2/CCM2	66	I/O	Concurrent compare 2
P5.3/CCM3	65	I/O	Concurrent compare 3
P5.4/CCM4	64	I/O	Concurrent compare 4
P5.5/CCM5	63	I/O	Concurrent compare 5
P5.6/CCM6	62	I/O	Concurrent compare 6
P5.7/CCM7	61	I/O	Concurrent compare 7
P4.7/CM7	9	I/O	Comp. output for the CM7 reg.
P4.6/CM6	8	I/O	Comp. output for the CM6 reg.
P4.5/CM5	7	I/O	Comp. output for the CM5 reg.
P4.4/CM4	6	I/O	Comp. output for the CM4 reg.
P4.3/CM3	5	I/O	Comp. output for the CM3 reg.
P4.2/CM2	3	I/O	Comp. output for the CM2 reg.
P4.1/CM1	2	I/O	Comp. output for the CM1 reg.
P4.0/CM0	1	I/O	Comp. output for the CM0 reg.
P1.7/T2	29	I/O	External count or gate input to timer 2
P1.5/T2EX	31	I/O	External reload trigger input
P1.4/$\overline{\text{INT2}}$/CC4	32	I/O	Comp. output/capture input for CC register 4
P1.3/INT6/CC3	33	I/O	Comp. output/capture input for CC register 3
P1.2/INT5/CC2	34	I/O	Comp. output/capture input for CC register 2
P1.1/INT4/CC1	35	I/O	Comp. output/capture input for CC register 1
P1.0/$\overline{\text{INT3}}$/CC0	36	I/O	Comp. output/capture input for CRC register

Table 9 Special Function Registers of the CCU

Symbol	Description	Address
CCEN	Comp./capture enable reg.	0C1H
CC4EN	Comp./capture 4 enable reg.	0C9H
CCH1	Comp./capture reg. 1, high byte	0C3H
CCH2	Comp./capture reg. 2, high byte	0C5H
CCH3	Comp./capture reg. 3, high byte	0C7H
CCH4	Comp./capture reg. 4, high byte	0CFH
CCL1	Comp./capture reg. 1, low byte	0C2H
CCL2	Comp./capture reg. 2, low byte	0C4H
CCL3	Comp./capture reg. 3, low byte	0C6H
CCL4	Comp./capture reg. 4, low byte	0CEH
CMEN	Compare enable register	0F6H
CMH0	Compare reg. 0, high byte	0D3H
CMH1	Compare reg. 1, high byte	0D5H
CMH2	Compare reg. 2, high byte	0D7H
CMH3	Compare reg. 3, high byte	0E3H
CMH4	Compare reg. 4, high byte	0E5H
CMH5	Compare reg. 5, high byte	0E7H
CMH6	Compare reg. 6, high byte	0F3H
CMH7	Compare reg. 7, high byte	0F5H
CML0	Compare reg. 0, low byte	0D2H
CML1	Compare reg. 1, low byte	0D4H
CML2	Compare reg. 2, low byte	0D6H
CML3	Compare reg. 3, low byte	0E2H
CML4	Compare reg. 4, low byte	0E4H
CML5	Compare reg. 5, low byte	0E6H
CML6	Compare reg. 6, low byte	0F2H
CML7	Compare reg. 7, low byte	0F4H
CMSEL	Compare input select	0F7H
CRCH	Com./rel./capt. reg., high byte	0CBH
CRCL	Com./rel./capt. reg., low byte	0CAH
CTCON	Com. timer control reg.	0E1H
CTRELH	Com. timer rel. reg., high byte	0DFH
CTRELL	Com. timer rel. reg., low byte	0DEH
IRCON	Interrupt control register	0C0H
TH2	Timer 2, high byte	0CDH
TL2	Timer 2, low byte	0CCH
T2CON	Timer 2 control register	0C8H

Figure 45 a)
Block Diagram of Timer 2



Timer 2

Timer 2 is one of the two 16-bit time bases of the compare/capture unit. It can operate as timer, event counter, or gated timer. The block diagram in Figure 45 a) Block Diagram of Timer 2 shows the general configuration of the timer 2.

Timer Mode

In timer function, the count rate is derived from the oscillator frequency. A 2:1 prescaler offers the possibility of selecting a count rate of 1/12 or 1/24 of the oscillator frequency. Thus, the 16-bit timer register (consisting of TH2 and TL2) is either incremented in every machine cycle or in every second machine cycle. The prescaler is selected by bit T2PS in special function register T2CON (see Figure 46 Special Function Register T2CON). If T2PS is cleared, the input frequency is 1/12 of the oscillator frequency; if T2PS is set, the 2:1 prescaler gates 1/24 of the oscillator frequency to the timer.

Gated Timer Mode

In gated timer function, the external input pin T2 (P1.7) functions as a gate to the input of timer 2. If T2 is high, the internal clock input is gated to the timer. T2 = 0 stops the counting

procedure. This will facilitate pulse width measurements. The external gate signal is sampled once every machine cycle (for the exact port timing, please refer to Parallel I/O on page 6-27).

Event Counter Mode

In the counter function, the timer 2 is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7). In this function, the external input is sampled every machine cycle. When the sampled inputs show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the timer register in the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle (see also Parallel I/O on page 6-27 for the exact sample time at the port pin P1.7).

Note: The prescaler must be off for proper counter operation of timer 2, i.e. T2PS must be 0.

In either case, no matter whether timer 2 is configured as timer, event counter, or gated timer, a rolling-over of the count from all 1's to all 0's sets the timer overflow flag TF2 (bit 6 in SFR IRCON, interrupt request control) which can generate an interrupt.

If TF2 is used to generate a timer overflow interrupt, the request flag must be cleared by the interrupt service routine as it could be necessary to check whether it was the TF2 flag or the external reload request flag EXF2 which requested the interrupt (for EXF2 see below). Both request flags cause the program to branch to the same vector address.

The input clock to timer 2 is selected by bits T2I0, T2I1, and T2PS as listed in Figure 46 Special Function Register T2CON.

Reload of Timer 2

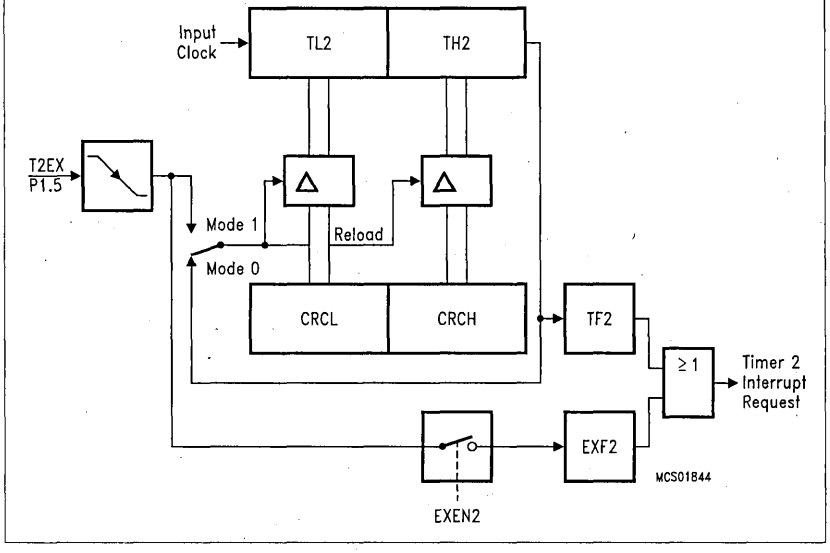
The reload mode for timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON as listed in Figure 45 b) Timer 2 in Reload Mode. Two reload modes are selectable:

In mode 0, when timer 2 rolls over from all 1's to all 0's, it not only sets TF2 but also causes the timer 2 registers to be loaded with the 16-bit value in the CRC register, which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H.

In mode 1, a 16-bit reload from the CRC register is caused by a negative transition at the corresponding input pin T2EX/P1.5. In addition, this transition will set flag EXF2, if bit EXEN2 in SFR IEN1 is set.

If the timer 2 interrupt is enabled, setting EXF2 will generate an interrupt. The external input pin T2EX is sampled in every machine cycle. When the sampling shows a high in one cycle and a low in the next cycle, a transition will be recognized. The reload of timer 2 registers will then take place in the cycle following the one in which the transition was detected.

Figure 45 b)
Timer 2 in Reload Mode



The Compare Timer

This timer - the fourth timer in the SAB 80C517 - is implemented to function as a fast 16-bit time base for the compare registers CM0 to CM7. The compare timer combine with the CMx registers can be employed as high-speed output unit or as a fast 16-bit pulse-width modulator unit. For this case, every CMx register assigned to the compare timer automatically operates in compare mode 0: a compare timer overflow sets the corresponding output line at port 4 to low level, a compare match pulls the pin high again (see also Compare Mode 0 on page 6-71).

The minimum resolution attainable at the port 4 outputs is $t_{CYCLE}/6$ (appr. 166.6 ns at f_{OSC} = 12 MHz). The compare timer is provided with a 16-bit auto-reload and an 8-bit prescaler for a very high flexibility concerning timer period length and input clock frequency. A block diagram of the compare timer is shown in Figure 47 Compare Timer Block Diagram.

Figure 46 Special Function Register T2CON

	0CFH	0CEH	0CDH	0CCH	0CBH	0CAH	0C9H	0C8H	
0C8H	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

These bits are not used in controlling the CCU.

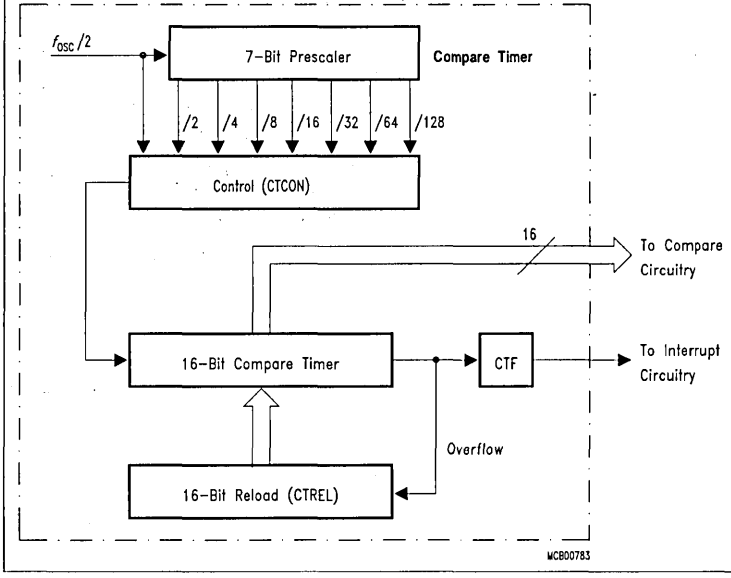
Timer 2 control register. Bit-addressable register which controls timer 2 function and compare mode of registers CRC, CC1 to CC3.

Bit		Symbol
T2I1	T2I0	Timer 2 input selection
0	0	No input selected, timer 2 stops
0	1	Timer function input frequency = $f_{OSC}/12$ (T2PS = 0) or $f_{OSC}/24$ (T2PS = 1)
1	0	Counter function, external input controlled by pin T2/P1.7.
1	1	Gated timer function, input controlled by pin T2/P1.7
T2R1	T2R0	Timer 2 reload mode selection
0	X	Reload disabled
1	0	Mode 0: auto-reload upon timer 2 overflow (TF2)
1	1	Mode 1: reload upon falling edge at pin T2EX/P1.5.
T2CM		Compare mode bit for registers CRC, CC1 through CC3. When set, compare mode 1 is selected. T2CM = 0 selects compare mode 0.
T2PS		Prescaler select bit. When set, timer 2 is clocked in the "timer" or "gated timer" function with 1/24 of the oscillator frequency. T2PS = 0 gates $f_{OSC}/12$ to timer 2. T2PS must be 0 for the counter operation of timer 2.

Input Clock Selection

The compare timer receives its input clock from a programmable prescaler which provides eight different input frequencies: $f_{osc}/2$, $f_{osc}/4$, $f_{osc}/8$, $f_{osc}/16$, $f_{osc}/32$, $f_{osc}/64$, $f_{osc}/128$, $f_{osc}/256$. The selection can be done in a three-bit field (binary coded) in special function register CTCON (see Figure 48 Compare Timer Control Register CTCON). Register CTCON can be written to at any time, its default value after reset is 00H (that is $f_{osc}/2$ input frequency).

Figure 47
Compare Timer Block Diagram

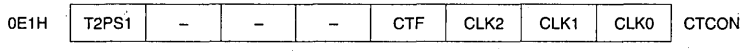


Programming the Compare Timer in Auto-Reload Operation

The compare timer is, once started, a free-running 16-bit timer, which upon overflow is automatically reloaded by the contents of the special function register CTRELL (compare timer reload register, low byte) and CTRELH (compare timer reload register, high byte). An initial writing to the reload register CTRELL (the low byte) starts the timer. If the compare timer is already running, a write-to-CTRELL again triggers an instant reload of the timer, in other words restarts the timer in the cycle following the write instruction with the count being loaded to the reload registers CTRELH/CTRELL.

Compare timer control register. Contains clock selection bits for the compare timer, the compare timer overflow flag and the control bit for the timer 2 prescaler.

Figure 48 Compare Timer Control Register CTCON



Bit	Function
CLK2 CLK1 CLK0	Compare timer input clock selection. See table below.
CTF	Compare timer overflow flag. Must be cleared by software. If the compare timer interrupt is enabled, CTF = 1 will cause an interrupt.
T2PS1	Prescaler select bit for timer 2 T2PS1 must be 0 for the counter operation of timer 2.

CLK2	CLK1	CLK0	Function
0	0	0	Compare timer input clock is $f_{OSC}/2$
0	0	1	Compare timer input clock is $f_{OSC}/4$
0	1	0	Compare timer input clock is $f_{OSC}/8$
0	1	1	Compare timer input clock is $f_{OSC}/16$
1	0	0	Compare timer input clock is $f_{OSC}/32$
1	0	1	Compare timer input clock is $f_{OSC}/64$
1	1	0	Compare timer input clock is $f_{OSC}/128$
1	1	1	Compare timer input clock is $f_{OSC}/256$

When the reload register is to be loaded with a 16-bit value, the high byte of CTREL must be written first to ensure a determined start or restart position. Writing to the low byte then triggers the actual reload procedure mentioned above. The 16-bit reload value can be overwritten at any time.

Overflow Interrupt of the Compare Timer

The compare timer has - as any other timer in the SAB 80C517 - its own interrupt request flag, which is in this case called CTF. This flag is located in register CTCON.CTF and is set when the timer count rolls over from all ones to the reload value.

The overflow interrupt eases e.g. software control of pulse width modulated output signals. A periodic interrupt service routine caused by an overflow of the compare timer can be used to load new values in the assigned compare registers and thus change the corresponding PWM output accordingly.

Please refer to Interrupt System on page 6-109 for details about the overflow interrupt (enabling, vector address, priority, etc.).

Compare Function in the CCU

The compare function of a timer/register combination can be described as follows. The 16-bit value stored in a compare or compare/capture register is compared with the contents of the timer register. If the count value in the timer register matches the stored value, an appropriate output signal is generated at a corresponding port pin.

The contents of a compare register can be regarded as 'time stamp' at which a dedicated output reacts in a predefined way (either with a positive or negative transition). Variation of this 'time stamp' somehow changes the wave of a rectangular output signal at a port pin. This may - as a variation of the duty cycle of a periodic signal - be used for pulse width modulation as well as for a continually controlled generation of any kind of square wave forms. In the case of the SAB 80C517, two compare modes are implemented to cover a wide range of possible applications (see Compare Modes of the CCU on page 6-71

In the SAB 80C517 - thanks to the high number of 13 compare registers and two associated timers - several timer/compare register combinations are selectable. In some of these configurations one of the two compare modes may be freely selected, others, however, automatically establish a compare mode. In the following the two possible modes are generally discussed. This description will be referred to in later sections where the compare registers are described.

Compare Modes of the CCU

As already mentioned, there are only a few compare registers with their corresponding port circuitry which are able to serve both compare modes. In most cases the mode is automatically set depending on the timer which is used as time base or depending on the port which outputs the compare signal.

Compare Mode 0

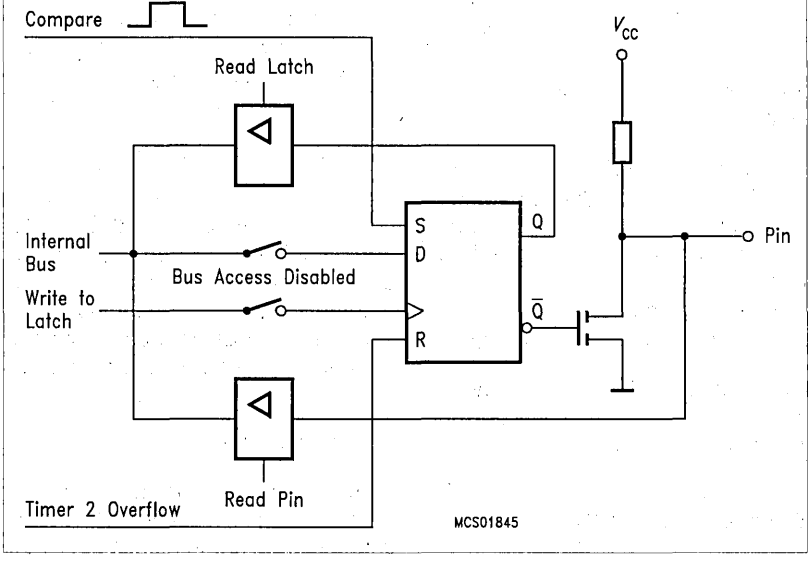
In mode 0, upon matching the timer and compare register contents, the output signal changes from low to high. It goes back to a low level on timer overflow. As long as compare mode 0 is enabled, the appropriate output pin is controlled by the timer circuit only, and not by the user. Writing to the port will have no effect. Figure 49 Port Latch in Compare Mode 0 shows a functional diagram of a port latch in compare mode 0. The port latch is directly controlled by the two signals timer overflow and compare. The input line from the internal bus and the write-to-latch line are disconnected when compare mode 0 is enabled.

Compare mode 0 is ideal for generating pulse width modulated output signals, which in turn can be used for digital-to-analog conversion via a filter network or by the controlled device itself (e.g. the inductance of a DC or AC motor). Mode 0 may also be used for providing output clocks with initially defined period and

duty cycle. This is the mode which needs the least CPU time. Once set up, the output goes on oscillating without any CPU intervention. Figure 50 Function of Compare Mode 0 illustrates the function of compare mode 0.

For some information on how to operate a timer/compare register configuration to generate PWM signals (e.g. by using a compare interrupt).

Figure 49
Port Latch in Compare Mode 0



Modulation Range of a PWM Signal and Differences between the Two Timer/ Compare Register Configurations in the CCU

There are two timer/compare register configurations in the CCU which can operate in compare mode 0 (either timer 2 with a CCx (CRC and CC1 to CC4) register or the compare timer with a CMx register). They basically operate in the same way, but show some differences concerning their modulation range when used for PWM.

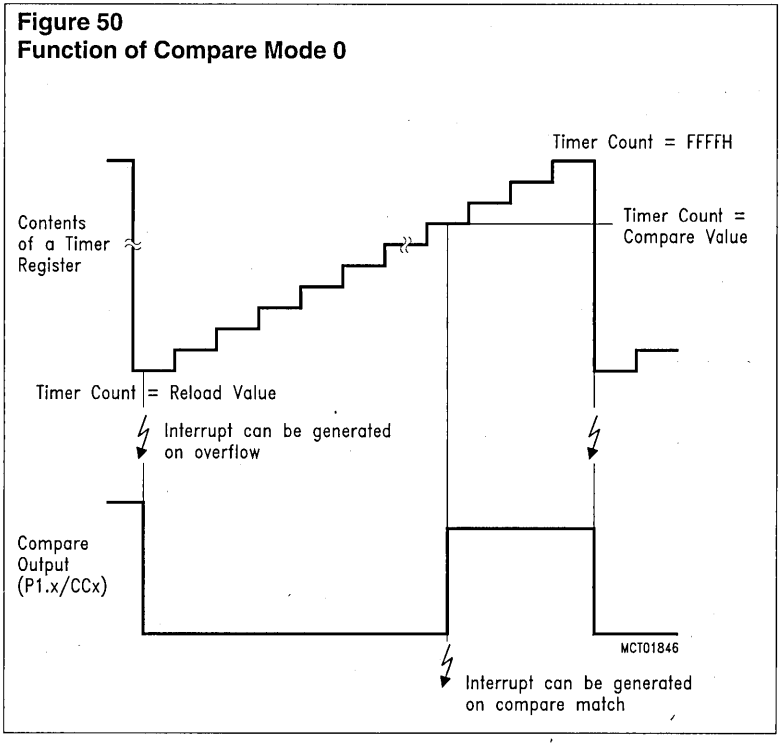
Generally it can be said that for every PWM generation with n-bit wide compare registers there are 2^n different settings for the duty cycle. Starting with a constant low level (0% duty cycle) as the first setting, the maximum possible duty cycle then would be

$$(1 - 1/2^n) \times 100\ \%$$

This means that a variation of the duty cycle from 0% to real 100% can never be reached if the compare register and timer register have the same length. There is always a spike which is as long as the timer clock period.

In the SAB 80C517 there are two different modulation ranges for the above mentioned two timer/compare register combinations. The difference is the location of the above spike within the timer period: at the end of a timer period or at the beginning plus the end of a timer period. Please refer to the description of the relevant timer/register combination in Timer/Compare Register Configurations in the CCU on page 6-75 for details.

Figure 50
Function of Compare Mode 0



Compare Mode 1

In compare mode 1, the software adaptively determines the transition of the output signal. This mode can only be selected for compare registers assigned to timer 2. It is commonly used when output signals are not related to a constant signal period (as in a standard PWM generation) but must be controlled very precisely with high resolution and without jitter. In compare mode 1, both transitions of a signal can be controlled. Compare outputs in this mode can be regarded as high speed outputs which are independent of the CPU activity.

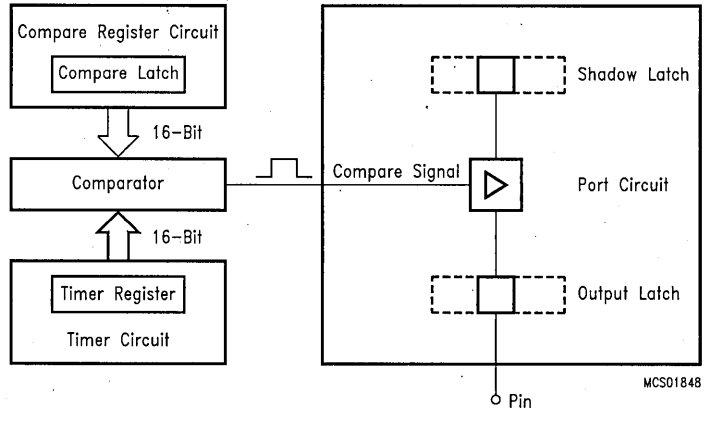
If mode 1 is enabled, and the software writes to the appropriate output latch at the port, the new value will not appear at the output pin until the next compare match occurs. Thus, one can choose whether the output signal is to make a new transition (1-to-0 or 0-to-1, depending on the actual pin-level) or should keep its old value at the time the timer 2 count matches the stored compare value.

Figure 51 Compare Function of Compare Mode 1 shows a functional diagram of a timer/compare register/port latch configuration in compare mode 1. In this function, the port latch consists of two separate latches. The upper latch (which acts as a "shadow latch") can be written under software control, but its value will only be transferred to the output latch (and thus to the port pin) in response to a compare match.

Note that the double latch structure is transparent as long as the internal compare signal is active. While the compare signal is active, a write operation to the port will then change both latches. This may become important when driving timer 2 with a slow external clock. In this case the compare signal could be active for many machine cycles in which the CPU could unintentionally change the contents of the port latch. For details see also Using Interrupts in Combination with the Compare Function on page 6-80.

A read-modify-write instruction (see Parallel I/O on page 6-27) will read the user-controlled "shadow latch" and write the modified value back to this "shadow-latch". A standard read instruction will - as usual - read the pin of the corresponding compare output.

Figure 51
Compare Function of Compare Mode 1



Timer/Compare Register Configurations in the CCU

The compare function and the reaction of the corresponding outputs depend on the timer/compare register combination. Basically, all compare functions implemented in the SAB 80(C)515 can also be used in the SAB 80C517. Furthermore, the SAB 80C517 has nine further compare registers and an additional 16-bit timer, thus providing a high flexibility in assigning compare registers to timers and output lines.

Table 10 CCU Configurations shows possible configurations of the CCU and the corresponding compare modes which can be selected. The following sections describe the function of these configurations.

Table 10 CCU Configurations

Assigned Timer	Compare Register	Compare Output at	Possible Modes
Timer 2	CRCH/CRCL	P1.0/INT3/CC0	Comp. mode 0, 1 + Reload
	CCH1/CCL1	P1.0/INT4/CC1	Comp. mode 0, 1
	CCH2/CCL2	P1.0/INT5/CC2	Comp. mode 0, 1
	CCH3/CCL3	P1.0/INT6/CC3	Comp. mode 0, 1
	CCH4/CCL4	P1.0/INT2/CC4	Comp. mode 0, 1
	CCH4/CCL4	P5.0/CCM0	Comp. mode 1
	:	:	:
	CCH4/CCL4	P5.7/CCM7	Comp. mode 1
	CMH0/CML0	P4.0/CM0	Comp. mode 1
	:	:	:
	CMH7/CML7	P4.7/CM7	Comp. mode 1
Compare timer	CMH0/CML0	P4.0/CM0	Comp. mode 0 (with shadow latches)
	:	:	:
	CMH7/CML7	P4.7/CM7	Comp. mode 0 (with shadow latches)

Compare Function of Timer 2 with Registers CRC, CC1 to CC4

Compare Function of Registers CRC, CC1 to CC3

The compare function of registers CRC, CC1 to CC3 is completely compatible with the corresponding function of the SAB 80(C)515. Registers CRC, CC1 to CC3 are permanently connected to timer 2.

All four registers are multifunctional as they additionally provide a capture (see Capture Function in the CCU on page 6-88) or a reload capability (the CRC register only, see Timer 2 on page 6-66). A general selection of the function is done in register CCEN (see Figure 52 Special Function Register CCEN). For compare function they can be used in compare mode 0 or 1, respectively. The compare mode is selected by setting or clearing bit T2CM in special function register T2CON.

Figure 52 Special Function Register CCEN

0C1H	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0	CCEN

Compare/capture enable register selects compare or capture function for register CRC, CC1 to CC3.

Bit		Function
COCAH0	**COCAL0**	**Compare/capture mode for CRC register**
0	0	Compare/capture disabled
0	1	Capture on falling/rising edge at pin P1.0/$\overline{\text{INT3}}$/CC0
1	0	Compare enabled
1	1	Capture on write operation into register CRCL
COCAH1	**COCAL1**	**Compare/capture mode for CC register 1**
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.1/INT4/CC1
1	0	Compare enabled
1	1	Capture on write operation into register CCL1
COCAH2	**COCAL2**	**Compare/capture mode for CC register 2**
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.2/INT5/CC2
1	0	Compare enabled
1	1	Capture on write operation into register CCL2
COCAH3	**COCAL3**	**Compare/capture mode for CC register 3**
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.3/INT6/CC3
1	0	Compare enabled
1	1	Capture on write operation into register CCL3

Figure 53 Timer 2 with Registers CCx (= CRC and CC1 to CC4) in Compare Mode 0 and Figure 54 Timer 2 with Registers CCx (= CRC and CC1 to CC4) in Compare Mode 1 show the general timer/compare register/port latch configuration for registers CRC and CC1 to CC4 in compare mode 0 and compare mode 1.

Please note that the compare interrupts of registers CRC and CC4 can be programmed to be negative or positive transition activated. Compare interrupts for the CC1 to CC3 registers are always positive transition activated.

Figure 53
Timer 2 with Registers CCx (= CRC and CC1 to CC4) in Compare Mode 0

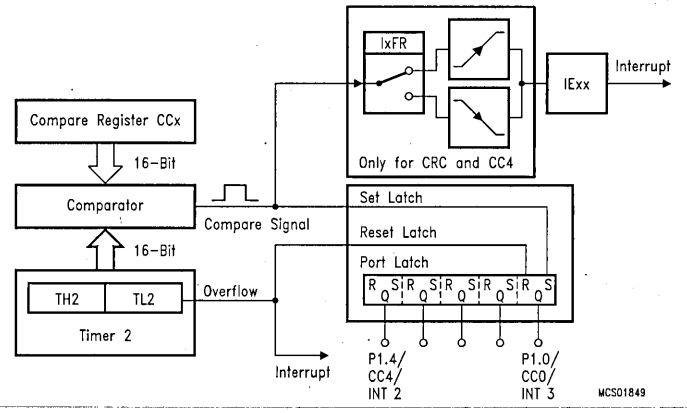
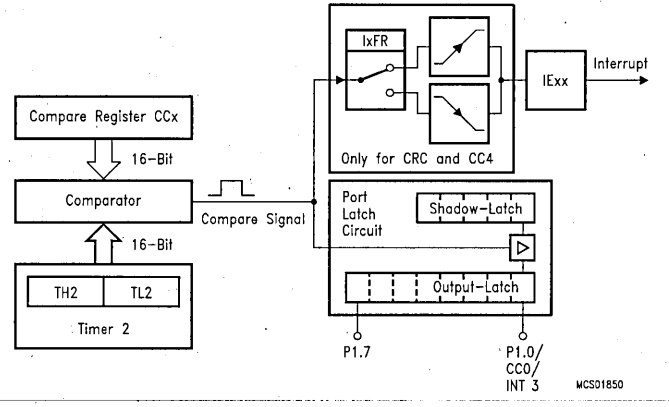


Figure 54
Timer 2 with Registers CCx (= CRC and CC1 to CC4) in Compare Mode 1

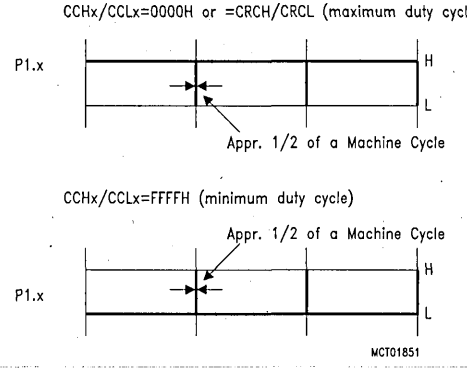


Modulation Range in Compare Mode 0

As already mentioned in the general description of compare mode 0 (Compare Modes of the CCU on page 6-71), a 100% variation of the duty cycle of a PWM signal cannot be reached. A time portion of $1/(2^n)$ of an n-bit timer period is always left over. This "spike" may either appear when the compare register is set to the reload value (limiting the lower end of the modulation range) or it may occur at the end of a timer period.

In a timer 2/CCx register configuration in compare mode 0 this spike is divided into two halves: one at the beginning when the contents of the compare register is equal to the reload value of the timer; the other half when the compare register is equal to the maximum value of the timer register (here: 0FFFFH). Please refer to Figure 55 Modulation Range of a PMW Signal Generated with a Timer 2/CCx Register Combination in Mode 0 Compare where the maximum and minimum duty cycle of a compare output signal is illustrated. Timer 2 is incremented with the machine clock ($f_{osc}/12$), thus at 12-MHz operational frequency, these spikes are both approx. 500 ns long.

Figure 55
Modulation Range of a PMW Signal Generated with a Timer 2/CCx Register Combination in Mode 0 Compare



The following example shows how to calculate the modulation range for a PWM signal. To calculate with reasonable numbers, a reduction of the resolution to 8-bit is used. Otherwise (for the maximum resolution of 16-bit) the modulation range would be so severely limited that it would be negligible.

Example:

Timer 2 in auto-reload mode; contents of reload register CRC = 0FF00H

$$\text{Restriction of module. Range} = \frac{1}{256 \times 2} \times 100\% = 0.195\%$$

This leads to a variation of the duty cycle from 0.195% to 99.805% for a timer 2/CCx register configuration when 8 of 16 bits are used.

Compare Function of Register CC4; "Concurrent Compare"

Compare register CC4 is new in the SAB 80C517 and permanently assigned to timer 2. It has its own compare/capture enable register CC4EN (see Figure 56 "Concurrent Compare" Function of Register CC4). Register CC4 can be set to operate as any of the other CC registers (see also Figure 53 Timer 2 with Registers CCx (= CRC and CC1 to CC4) in Compare Mode 0 and 7-43). Its output pin is P1.4/CC4/$\overline{\text{INT2}}$ and it has a dedicated compare mode select bit COMO located in register CC4EN.

In addition to the standard operation in compare mode 0 or 1, there is another feature called 'concurrent compare' which is just an application of compare mode 1 to more than one output pin. Concurrent compare means that the comparison of CC4 and timer 2 can manipulate up to nine port pins concurrently. A standard compare register in compare mode 1 normally transfers a preprogrammed signal level to a single output line. Register CC4, however, is able to put a 9-bit pattern to nine output lines. The nine output lines consist of one line at port P1.4 (which is the standard output for register CC4) and an additional eight lines at port 5 (see Figure 56 "Concurrent Compare" Function of Register CC4).

Concurrent compare is an ideal and effective option where more than one synchronous output signal is to be generated. Applications including this requirement could among others be a complex multiple-phase stepper motor control as well as the control of ignition coils of a car engine. All these applications have in common that predefined bit-patterns must be put to an output port at a precisely predefined moment. This moment refers to a special count of timer 2, which was loaded to compare register CC4.

Figure 57 Example for a "Concurrent Compare" at Port 5 gives an example of how

to generate eight different rectangular wave forms at port 5 using a pattern table and a time schedule for these patterns. The patterns are moved into port 5 before the corresponding timer count is reached. The (future) timer count at which the pattern shall appear at the port must be loaded to register CC4. Thus the user can mask each port bit differently depending on whether he wants the output to be changed or not.

Concurrent compare is enabled by setting bit COCOEN in special function register CC4EN. A '1' in this bit automatically sets compare mode 1 for register CC4, too. A 3-bit field in special function register CC4EN determines the additional number of output pins at port 5. Port P1.4/CC4/$\overline{\text{INT2}}$ is used as a standard output pin in any compare mode for register CC4.

Figure 56
"Concurrent Compare" Function of Register CC4

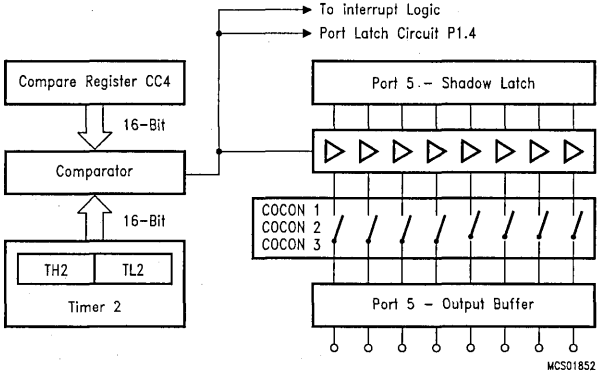


Figure 57
Example for a "Concurrent Compare" at Port 5

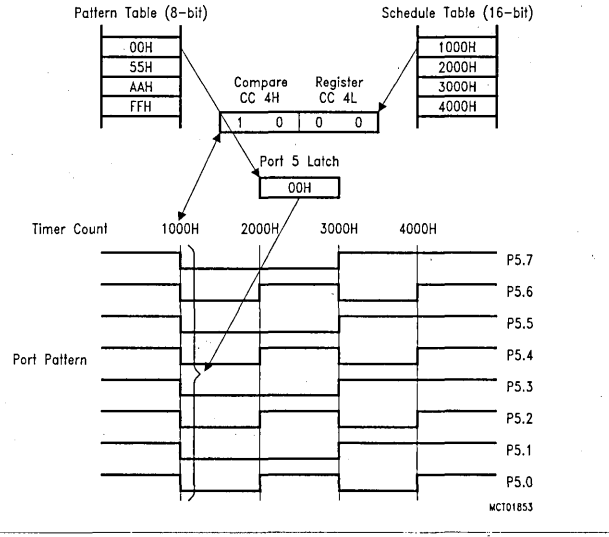


Figure 58 Compare/Capture Enable Register CC4EN

0C9H	–	COCON2	COCON1	COCON0	COCOEN	COCAH4	COCAL4	COMO	CC4EN

Selects compare or capture function, number of concurrent compares and compare mode of register CC4.

Bit		Function
COCAH4	COCAL4	Compare/capture mode for CC4 register
0	0	Compare/capture disabled
0	1	Capture on falling/rising edge at pin P1.0/$\overline{INT2}$/CC4
1	0	Compare enabled
1	1	Capture on write operation into register CC4L.
COMO		Compare mode bit. When set compare mode 1 is selected for CC4. COMO = 0 selects compare mode 0.
COCOEN		Enables the compare mode 1 and the concurrent compare output for CC4. Setting of this bit automatically sets bit COMO.
COCON2 COCON1 COCON0		Selects additional concurrent compare outputs at port 5. See table below.

COCON2	COCON1	COCON0	Function
0	0	0	One additional output of CC4 at P5.0
0	0	1	Additional outputs of CC4 at P5.0 to P5.1
0	1	0	Additional outputs of CC4 at P5.0 to P5.2
0	1	1	Additional outputs of CC4 at P5.0 to P5.3
1	0	0	Additional outputs of CC4 at P5.0 to P5.4
1	0	1	Additional outputs of CC4 at P5.0 to P5.5
1	1	0	Additional outputs of CC4 at P5.0 to P5.6
1	1	1	Additional outputs of CC4 at P5.0 to P5.7

Using Interrupts in Combination with the Compare Function

The compare service of registers CRC, CC1, CC2, CC3 and CC4 is assigned to alternate output functions at port pins P1.0 to P1.4. Another option of these pins is that they can be used as external interrupt inputs. However, when using the port lines as compare outputs then the input line from the port pin to the interrupt system is disconnected (but the pin's level can still be read under software control). Thus, a change of the pin's level will not cause a setting of the corresponding interrupt flag. In this case, the interrupt input is directly connected to the (internal) compare signal thus providing a compare interrupt.

The compare interrupt can be used very effectively to change the contents of the compare registers or to determine the level of the port outputs for the next "compare match". The principle is, that the internal compare signal (generated at a match between timer count and register contents) not only manipulates the compare output but also sets the corresponding interrupt request flag. Thus, the current task of the CPU is interrupted - of course provided the priority of the compare interrupt is higher than the present task priority - and the corresponding interrupt service routine is called. This service routine then sets up all the necessary parameters for the next compare event.

Some advantages in using compare interrupts:

Firstly, there is no danger of unintentional overwriting a compare register before a match has been reached. This could happen when the CPU writes to the compare register without knowing about the actual timer 2 count.

Secondly, and this is the most interesting advantage of the compare feature, the output pin is exclusively controlled by hardware therefore completely independent from any service delay which in real time applications could be disastrous. The compare interrupt in turn is not sensitive to such delays since it loads the parameters for the next event. This in turn is supposed to happen after a sufficient space of time.

Please note two special cases where a program using compare interrupts could show a "surprising" behavior:

The first configuration has already been mentioned in the description of compare mode 1. The fact that the compare interrupts are transition activated becomes important when driving timer 2 with a slow external clock. In this case it should be carefully considered that the compare signal is active as long as the timer 2 count is equal to the contents of the corresponding compare register, and that the compare signal has a rising and a falling edge. Furthermore, the "shadow latches" used in compare mode 1 are transparent while the compare signal is active.

Thus, with a slow input clock for timer 2, the comparator signal is active for a long time (= high number of machine cycles) and therefore a fast interrupt controlled reload of the compare register could not only change the "shadow latch" - as probably intended - but also the output buffer.

When using the CRC or CC4 register, you can select whether an interrupt should be generated when the compare signal goes active or inactive, depending on the status of bits I3FR or I2FR in T2CON, respectively.

Initializing the interrupt to be negative transition triggered is advised in the above case. Then the compare signal is already inactive and any write access to the port latch just changes the contents of the "shadow-latch".

Please note that for CC registers 1 to 3 an interrupt is always requested when the compare signal goes active.

The second configuration which should be noted is when compare functions are combined with negative transition activated interrupts. If the port latch of port P1.0 or P.1.4 contains a 1, the interrupt request flags IEX3 or IEX2 will immediately be set after enabling the compare mode for the CRC or CC4 register. The reason is that first the external interrupt input is controlled by the pin's level. When the compare option is enabled the interrupt logic input is switched to the internal compare signal, which carries a low level when no true comparison is detected. So the interrupt logic sees a 1-to-0 edge and sets the interrupt request flag.

An unintentional generation of an interrupt during compare initialization can be prevented if the request flag is cleared by software after the compare is activated and before the external interrupt is enabled.

Compare Function of Registers CM0 to CM7

The CCU of the SAB 80C517 contains another set of eight compare registers, an additional timer (the compare timer) and some control SFR in the CCU which have not been described yet. These compare registers and the compare timer are mainly dedicated to PWM applications.

The additional compare registers CM0 to CM7, however, are not permanently assigned to the compare timer, each register may individually be configured to work either with timer 2 or the compare timer as shown in Table 10 CCU Configurations.

The flexible assignment of the CMx registers allows an independent use of two time bases where by different application requirements can be met. Any CMx register connected to the compare timer automatically works in compare mode 0 e.g. to provide fast PWM with low CPU intervention. Together with timer 2, CMx registers operate in compare mode 1; the latter configuration, which is described in the next section, allows the CPU to control the compare output transitions directly.

The assignment of the eight registers CM0 to CM7 to either timer 2 or to the compare timer is done by an 8-channel 2:1 multiplexer (shown in the general block diagram in Figure 44 Block Diagram of the CCU. The multiplexer can be programmed by the corresponding bits in special function register CMSEL (see Figure 59 Special Function Register CMSEL). The compare function itself can individually be enabled in the SFR CMEN (see Figure 60 Special Function Register CMEN).

Note however that these register are not bit-addressable, which means that the value of single bits can only be changed by AND-ing or OR-ing the register with a certain mask.

Contains select bits for registers CM0 to CM7. When set, CMLx/CMHx are assigned to the compare timer and compare mode 0 is enabled. The compare registers are assigned to timer 2 if CMSELx = 0. In this case compare mode 1 is selected.

Figure 59 Special Function Register CMSEL

0F7H	CMSEL.7	CMSEL.6	CMSEL.5	CMSEL.4	CMSEL.3	CMSEL.2	CMSEL.1	CMSEL.0	CMSEL

Figure 60 Special Function Register CMEN

0F6H	CMEN.7	CMEN.6	CMEN.5	CMEN.4	CMEN.3	CMEN.2	CMEN.1	CMEN.0	CMEN

Contains enable bits for compare registers CM0 to CM7. When set, compare function is enabled and led to the output lines.

Bit	Function
CMEN.7	Compare enable bit for CM7
CMEN.6	Compare enable bit for CM6
CMEN.5	Compare enable bit for CM5
CMEN.4	Compare enable bit for CM4
CMEN.3	Compare enable bit for CM3
CMEN.2	Compare enable bit for CM2
CMEN.1	Compare enable bit for CM1
CMEN.0	Compare enable bit for CM0

Bit	Function
CMSEL7	Select bit for CM7
CMSEL6	Select bit for CM6
CMSEL5	Select bit for CM5
CMSEL4	Select bit for CM4
CMSEL3	Select bit for CM3
CMSEL2	Select bit for CM2
CMSEL1	Select bit for CM1
CMSEL0	Select bit for CM0

First Configuration: CMx Registers Assigned to the Compare Timer

Every CMx register switched to the compare timer as a time base operates in compare mode 0 and uses a port 4 pin as an alternate output function (see Table 8 Alternate Port Functions of the CCU).

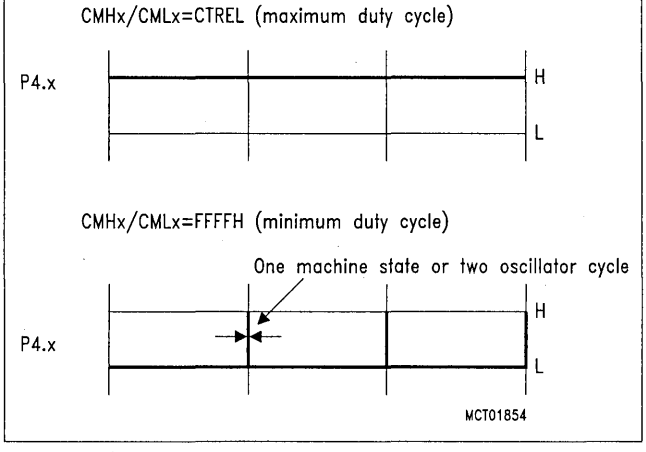
Modulation Range in Compare Mode 0

In the general description of compare mode 0 (Compare Modes of the CCU on page 6-71) and in the description of the timer 2/CCx register configuration (Compare Function of Timer 2 with Registers CRC, CC1 to CC4 on page 6-75) it was mentioned that a compare output is restricted in its maximum or minimum duty cycle. There is always a time portion of $1/2^n$ (at n-bit timer length) which is left over. This "spike" may either appear when the compare register is set to the reload value (limiting the lower end of the modulation range) or it may occur at the end of a timer period as realized in this configuration. In a compare timer/CMx register configuration, the compare output is set to a constant high level if the contents of the compare registers are equal to the reload register (CTREL). The compare output shows a high level for one timer clock period when a CMx register is set to 0FFFFH. Thus, the duty cycle can be varied from 0.xx% to 100% depending on the resolution selected (see calculation example in Compare Function of Timer 2 with Registers CRC, CC1 to CC4 on page 6-75). Please refer to Figure 61 Modulation Range of a PWM Signal Generated with a Compare Timer/CMx Register Combination where the maximum and minimum duty cycle of a compare output signal is illustrated. One clock period of the compare timer is equal to one machine state (= 2 oscillator periods) if the prescaler is off. Thus, at 12-MHz operational frequency the spike is approx. 166.6 ns long.

The "Timer Overflow Controlled" Loading

There is one great difference between a CMx register and the other previously described compare registers: compare outputs controlled by CMx registers have no dedicated interrupt function. They use a "timer overflow controlled loading" (further on called "TOC loading") to reach the same performance as an interrupt controlled compare. To show what this "TOC loading" is for, it will be explained more detailed in the following:

Figure 61
Modulation Range of a PWM Signal Generated with a Compare Timer/CMx Register Combination



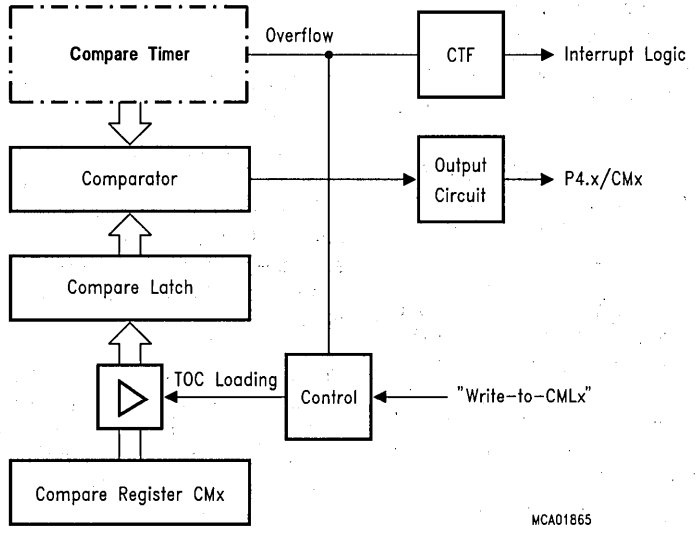
The main advantage of the compare function in general is that the controller's outputs are precisely timed by hardware, no matter which task is running on the CPU. This in turn means that the CPU normally does not know about the timer count. So, if the CPU writes to a compare register only in relation to the program flow, then it could easily be that a compare register is overwritten before the timer had the chance to reach the previously loaded compare value. Hence, there must be something to "synchronize" the loading of the compare registers to the running timer circuitry. This could either be an interrupt caused by the timer circuitry (as described before) or a special hardware circuitry.

Thus "TOC-loading" means that there is dedicated hardware in the CCU which synchronizes the loading of the compare registers CMx in such a way that there is no loss of compare events. It also relieves the CPU of interrupt load.

What does this hardware look like:

A CMx compare register in compare mode 0 consists of two latches. When the CPU tries to access a CMx register it only addresses a register latch and not the actual compare latch which is connected to the comparator circuit. The contents of the register latch may be changed by the CPU at any time because this change would never affect the compare event for the current timer period. The compare latch (the "actual" latch) holds the compare value for the present timer period. Thus the CPU only changes the compare event for the next timer period since the loading of the latch is performed by the timer overflow signal of the compare timer

Figure 62
Compare Function of a CMx Register Assigned to the Compare Timer



This means for an application which uses several PWM outputs that the CPU does not have to serve every single compare line by an individual interrupt. It only has to watch the timer overflow of the compare timer and may then set up the compare events of all compares for the next timer period. This job may take the whole current timer period since the TOC loading prevents unintentional overwriting of the actual (and prepared) value in the compare latch.

Figure 62 Compare Function of a CMx Register Assigned to the Compare Timer shows a more detailed block diagram of a CMx register connected to the compare timer. It illustrates that the CPU can only access the special function register CMx; the actual compare latch is, however, loaded at timer overflow. The timer overflow signal also sets an interrupt request flag (CTF in register CTCON) which may be used to inform the CPU by an interrupt that a new timer cycle has started and that the compare values for the next cycle may be programmed from now on.

The activation of the TOC loading depends on a few conditions described in the following. A TOC loading is performed only if the CMLx register has been changed by the CPU. A write instruction to the low byte of the CMx register is used to enable the loading.

The 8-bit architecture of the SAB 80C517 requires such a defined enable mechanism because 16-bit values are to be transferred in two portions (= two instructions).

Imagine the following situation: one instruction (e.g. loading the low byte of the compare register) is executed just before timer overflow and the other instruction (loading the high byte) after the overflow. If there were no "rule", the TOC loading would just load the new low byte into the compare latch. The high byte - written after timer overflow - would have to wait till the next timer overflow.

The mentioned condition for TOC loading prevents such undesired behavior. If the user writes the high byte first then no TOC loading will happen before the low byte has been written - even if there is a timer overflow in between. If the user just intends to change the low byte of the compare latch then the high byte may be left unaffected.

Summary of the above description of the TOC loading:

- The CMx registers are - when switched to the compare timer - protected from direct loading by the CPU. A register latch couple provides a defined load time at timer overflow.
- Thus, the CPU has a full timer period to load a new compare value: there is no danger of overwriting compare values which are still needed in the current timer period.
- When writing a 16-bit compare value, the high byte should be written first since the write-to-low-byte instruction enables a 16-bit wide TOC loading at next timer overflow.
- If there was no write access to a CMx low byte then no TOC loading will take place.
- Because of the TOC loading, all compare values written to CMx registers are only activated in the next timer period.

Initializing the Compare Register/Compare Latch Circuit

Normally when the compare function is desired the initialization program would just write to the compare register (called 'register latch'). The compare latch itself cannot be accessed directly by a move instruction, it is exclusively loaded by the timer overflow signal.

In some very special cases, however, an initial loading of the compare latch could be desirable. If the following sequence is observed during initialization then latches, the register and the compare latch, can be loaded before the compare mode is enabled.

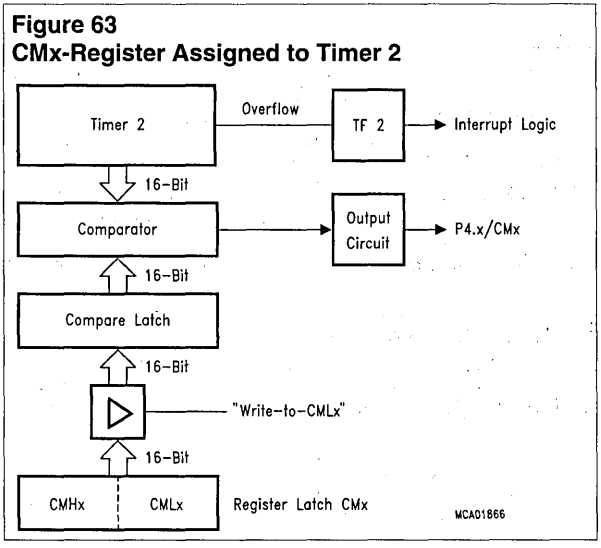
Action:	Comment:
Select compare mode 1 (CMSEL.x = 0).	This is also the default value after reset.
Move the compare value for the first timer period to the compare register CMx (high byte first).	In compare mode 1 the compare latch is loaded directly after a write-to-CMLx. Thus the value slips directly into the compare latch.
Switch on compare mode 0 (CMSEL.x = 1).	Now select the right compare mode.
Move the compare value for the second timer period to the compare register.	The register latch is loaded. This value is used after the first timer overflow.
Enable the compare function (CMEN.x = 1)	The compare output is switched to low level.
Set up the prescaler for the compare timer.	
Set specific compare output to low level (CLRP4.x)	
Start the compare timer with a desired value (write-to-CTREL)	Compare function is initialized. The output will oscillate.

Second Configuration
CMx Registers Assigned to Timer 2

Any CMx register switched to timer 2 as a time base operates in compare mode 1. In this case CMx registers behave like any other compare register connected to timer 2 (e.g. the CRC or CCx registers). Please refer to the above description of compare mode 1 for further details.

Since there are no dedicated interrupts for the CMx compare outputs, again a buffered compare register structure is used to determine an exact 16-bit wide loading of the compare value: the compare value is transferred to the actual compare latches at a write-to-CMLx instruction (low byte of CMx). Thus, the CMx register is to be written in a fixed order, too: high byte first, low byte second. If the high byte may remain unchanged it is sufficient to load only the low byte. See Figure 63 CMx-Register Assigned to Timer 2, block diagram of a CMx register connected to timer 2.

Figure 63
CMx-Register Assigned to Timer 2



Capture Function in the CCU

Each of the four compare/capture registers CC1 to CC4 and the CRC register can be used to latch the current 16-bit value of the timer 2 registers TL2 and TH2. Two different modes are provided for this function. In mode 0, an external event latches the timer 2 contents to a dedicated capture register. In mode 1, a capture will occur upon writing to the low order byte of the dedicated 16-bit capture register. This mode is provided to allow the software to read the timer 2 contents "on-the-fly".

In mode 0, the external event causing a capture is

- for CC registers 1 to 3: a positive transition at pins CC1 to CC3 of port 1
- for the CRC and CC4 register: a positive or negative transition at the

corresponding pins, depending on the status of the bits I3FR and I2FR in SFR T2CON. If the edge flags are cleared, a capture occurs in response to a negative transition; if the edge flags are set a capture occurs in response to a positive transition at pins P1.0/ $\overline{\text{INT3}}$/ CC0 and P1.4/ $\overline{\text{INT2}}$/ CC4.

In both cases the appropriate port 1 pin is used as input and the port latch must be programmed to contain a one (1). The external input is sampled in every machine cycle. When the sampled input shows a low (high) level in one cycle and a high (low) in the next cycle, a transition is recognized. The timer 2 contents is latched to the appropriate capture register in the cycle following the one in which the transition was identified.

In mode 0 a transition at the external capture inputs of registers CC0 to CC4 will also set the corresponding external interrupt request flags IEX2 to IEX6. If the interrupts are enabled, an external capture signal will cause the CPU to vector to the appropriate interrupt service routine.

In mode 1 a capture occurs in response to a write instruction to the low order byte of a capture register. The write-to-register signal (e.g. write-to-CRCL) is used to initiate a capture. The value written to the dedicated capture register is irrelevant for this function. The timer 2 contents will be latched into the appropriate capture register in the cycle following the write instruction. In this mode no interrupt request will be generated.

Figure 64 Capture with Registers CRC, CC4 and Figure 65 Capture with Registers CC1 to CC3 show functional diagrams of the capture function of timer 2. Figure 64 Capture with

Registers CRC, CC4 illustrates the operation of the CRC or CC4 register, while Figure 65 Capture with Registers CC1 to CC3 shows the operation of the compare/capture registers 1 to 3.

The two capture modes can be established individually for each capture register by bits in SFR CCEN (compare/capture enable register) and CC4EN (compare/capture 4 enable register). That means, in contrast to the compare modes, it is possible to simultaneously select mode 0 for one capture register and mode 1 for another register. The bit positions and functions of CCEN are listed in Figure 52 Special Function Register CCEN, the one for CC4EN in Figure 58 Compare/ Capture Enable Register CC4EN.

Figure 64
Capture with Registers CRC, CC4

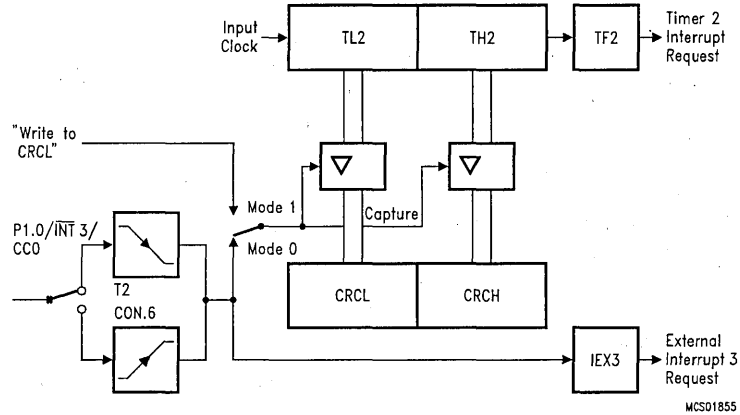
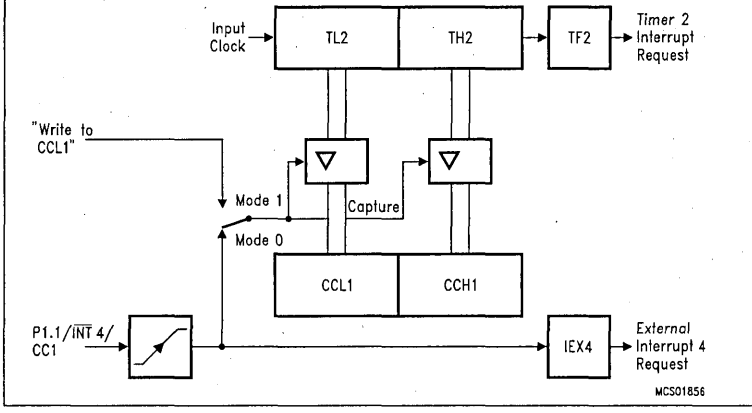


Figure 65
Capture with Registers CC1 to CC3



Arithmetic Unit

This on-chip arithmetic unit of the SAB 80C517 provides fast 32-bit division, 16-bit multiplication as well as shift and normalize features. All operations are unsigned integer operations.

The arithmetic unit (further on also called MDU for "Multiplication/Division Unit") has been integrated to support the 8051 core of the SAB 80C517 in real-time control applications. It can increase the execution speed of math-intensive software routines by factor 5 to 10.

The MDU is handled by seven registers, which are memory mapped as special function registers like any other registers for peripheral control. Therefore, the arithmetic unit allows operations concurrently to and independent of the CPU's activity.

The following table describes the four general operations the MDU is able to perform:

Operation	Result	Remainder	Execution Time
32bit/16bit	32bit	16bit	6 t_{CY} 1)
16bit/16bit	16bit	16bit	4 t_{CY} 1)
16bit x 16bit	32bit	–	4 t_{CY} 1)
32-bit normalize	–	–	6 t_{CY} 2)
32-bit shift L/R	–	–	6 t_{CY} 2)

1) 1 t_{CY} = 1 microsecond at 12-MHz oscillator frequency

2) The maximal shift speed is 6 shifts per machine cycle

Programming the MDU

Operating Registers of the MDU

The seven SFR of the MDU consist of registers MD0 to MD5, which contain the operands and the result (or the remainder, resp.) and one control register called ARCON.

Thus MD0 to MD5 are used twofold:

- for the operands before a calculation has been started and
- for storage of the result or remainder after a calculation.

This means that any calculation of the MDU overwrites its operands. If a program needs the original operands for further use, they should be stored in general purpose registers in the internal RAM.

Operation of the MDU

The MDU can be regarded as a special coprocessor for multiplication, division and shift. Its operations can be divided into three phases (see also Figure 66 Operating Phases of the MDU):

1) Loading the MDx registers

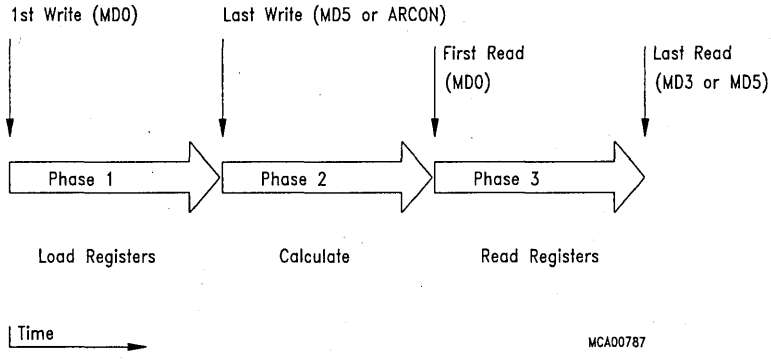
2) Executing the calculation

3) Reading the result from the MDx registers

During phase two, the MDU works on its own parallelly to the CPU. Execution times of the above table refer to this phase. Because of the fast operation and the determined execution time for SAB 80C517's instructions, there is no need for a busy flag. The CPU may execute a determined number of instructions before the result is fetched. The result and the remainder of an operation may also be stored in the MDx registers for later use.

Phase one and phase three require CPU activity. In these phases the CPU has to transfer the operands and fetch the results.

Figure 66
Operating Phases of the MDU



How to Select an Operation

The MDU has no dedicated instruction register (only for shift and normalize operations, register ARCON is used in such a way). The type of calculation the MDU has to perform is selected following the order in which the MDx registers are written to (see Table 11 Programming the MDU for Multiplication and Division. This mechanism also reduces execution time spent for controlling the MDU. Hence, a special write sequence selects an operation.

The MDU monitors the whole write and read-out sequence to ensure that the CPU has fetched the result correctly and was not interrupted by another calculation task. (See The Error Flag on page 6-94).

Thus, a complete operation lasts from writing the first byte of the operand in phase 1 until reading the last byte of the result in phase 3.

Multiplication/Division

The general mechanism to start an MDU activity has been described above. The following description of the write and read sequences adds to the information given in the table below where the write and read operations necessary for a multiplication or division are listed.

Table 11 Programming the MDU for Multiplication and Division

Operation	32Bit/16Bit	16Bit/16Bit	16Bit x 16Bit
First Write	MD0 D'endL	MD0 D'endL	MD0 M'andL
	MD1 D'end	MD1 D'endH	MD4 M'orL
	MD2 D'end		
	MD3 D'endH	MD4 D'orL	MD1 M'andH
	MD4 D'orL		
Last Write	MD5 D'orH	MD5 D'orH	MD5 M'orH
First Read	MD0 QuoL	MD0 QuoL	MD0 PrL
	MD1 Quo	MD1 QuoH	MD1
	MD2 Quo		
	MD3 QuoH	MD4 RemL	MD2
	MD4 RemL		
Last Read	MD5 RemH	MD5 RemH	MD3 PrH

Write Sequence

The first and the last write operation in phase one are fixed for every calculation of the MDU. All write operations in-between determine the type of MDU calculation.

A write-to-MD0 is the first transfer to be done in any case. This write resets the MDU and triggers the error flag mechanism (see below).

The next two or three write operations select the calculation type (32bit/16bit, 16bit/16bit, 16bit *16bit).
The last write-to-MD5 finally starts the selected MUL/DIV operation.

Read Sequence

Any read-out of the MDx registers should begin with MD0

The last read from MD5 (division) or MD3 (multiplication) determines the end of a whole calculation and releases the error flag mechanism.

There is no restriction on the time within which a calculation must be completed. The CPU is allowed to continue the program simultaneously to phase 2 and to fetch the result bytes at any time.

If the user's program takes care that interrupting a calculation is not possible, monitoring of the calculation process is probably not needed. In this case, only the write sequence must be observed.

Any new write access to MD0 starts a new calculation, no matter whether the read-out of the former result has been completed or not.

Normalize and Shift

Register ARCON controls an up to 32-bit wide normalize and shift operation in registers MD0 to MD3. It also contains the overflow flag and the error flag which are described in the next two sections. Figure 67 Register ARCON illustrates special function register ARCON.

Write Sequence

A write-to-MD0 is also the first transfer to be done for normalize and shift. This write resets the MDU and triggers the error flag mechanism (see below).

To start a shift or normalize operation the last write must access register ARCON.

Read Sequence

The order in which the first three registers MD0 to MD2 are read is not critical

The last read from MD3 determines the end of a whole shift or normalize procedure and releases the error flag mechanism.

Note: Any write access to ARCON triggers a shift or normalize operation and therefore changes the contents of registers MD0 to MD3!

Figure 67 Register ARCON

0EFH	MDEF	MDOV	SLR	SC.4	SC.3	SC.2	SC.1	SC.0	ARCON

Arithmetic control register. Contains control flags and the shift counter of the MDU. Triggers a shift or a normalize operation in register MD0 to MD3 when being written to.

Bit	Function
MDEF	Error flag. Indicates an improperly performed operation. MDEF is set by hardware when an operation is retriggered by a write access to MDx before the first operation has been completed. MDEF is automatically cleared after being read.
MDOV	Overflow flag. Exclusively controlled by hardware. MDOV is set by following events: – division by zero – *multiplication with a result greater than 0FFFFH.*
SLR	Shift direction bit. When set, shift right is performed. SLR = 0 selects shift left operation.
SC.4 SC.3 SC.2 SC.1 SC.0	Shift counter. *When preset with 00000B, normalizing is selected. After operation* SC.0 to SC.4 contain the number of normalizing shifts performed. When set with a value ≠ 0, shift operation is started. The number of shifts performed is determined by the count written to SC.0 to SC.4.

Normalizing

Normalizing is done on an integer variable stored in MD0 (least significant byte) to MD3 (most significant byte). This feature is mainly meant to support applications where floating point arithmetic is used. "To normalize" means, that all reading zeroes of an integer variable in registers MD0 to MD3 are removed by shift left operations. The whole operation is completed when the MSB (most significant bit) contains a '1'.

To select a normalize operation, the five bit field ARCON.0 to ARCON.4 must be cleared. That means, a write-to-ARCON instruction with the value XXX0 0000B starts the operation.

After normalizing, bits ARCON.0 to ARCON.4 contain the number of shift left operations which were done. This number may further on be used as an exponent. The maximum number of shifts in a normalize operation is 31 (= 2^5 – 1). The operation takes six machine cycles at most, that means 6 microseconds at 12 MHz.

Shifting

In the same way - by a write-to-ARCON instruction - a shift left/right operation can be started. In this case register bit SLR (ARCON.5) has to contain the shift direction, and ARCON.0 to ARCON.4 the shift count (which must not be 0, otherwise a normalize operation would be executed). During shift, zeroes come into the left or right end of the registers MD0 or MD3, respectively.

The first machine cycle of a shift left/right operation executes four shifts, while all following cycles perform 6 shifts. Hence, a 31-bit shift takes 6 microseconds at 12 MHz.

Completion of both operations, normalize and shift, can also be controlled by the error flag mechanism. The error flag is set if one of the relevant registers (MD0 through MD3) is accessed before the previously commenced operation has been completed.

For proper operation of the error flag mechanism, it is necessary to take care that the right write or read sequence to or from registers MD0 to MD3 (see Table 12 Programming a Shift or Normalize Operation) is maintained.

Table 12 Programming a Shift or Normalize Operation

Operation	Normalize, Shift Left, Shift Right	
First write	MD0	least significant byte
	MD1	
	MD2	
	MD3	most significant byte
Last write	ARCON	start of conversion
First read	MD0	least significant byte
	MD1	
	MD2	
Last read	MD3	most significant byte

The Overflow Flag

An overflow flag is provided for some exceptions during MDU calculations. There are three cases where flag MDOV ARCON.6 is set by hardware:

- Division by zero
- Multiplication with a result greater then 0000 FFFF H
  (= auxiliary carry of the lower 16bit)
- Start of normalizing if the most significant bit of MD3 is set (MD3.7 = 1).

Any operation of the MDU which does not match the above conditions clears the overflow flag. Note that the overflow flag is exclusively controlled by hardware. It cannot be written to.

The Error Flag

An error flag, bit MDEF in register ARCON (Figure 67 Register ARCON), is provided to indicate whether one of the arithmetic operations of the MDU (multiplication, division, normalize, shift left/right) has been restarted or interrupted by a new operation.

This can possibly happen e.g. when an interrupt service routine interrupts the writing or reading sequence of the arithmetic operation in the main program and starts a new operation. Then the contents of the corresponding registers are indeterminate (they would normally show the result of the last operation executed).

In this case the error flag can be used to indicate whether the values in the registers MD0 to MD5 are the expected ones or whether the operation must be repeated. For a multiplication/division, the error flag mechanism is automatically enabled with the first write instruction to MD0 (phase 1). According to the above described programming sequences, this is the first action for every type of calculation. The mechanism is disabled with the final read instruction from MD3 or MD5 (phase 3). Every instruction which rewrites MD0 (and therefore tries to start a new calculation) in phases 1 through 3 of the same process sets the error flag.

The same applies for any shift operation (normalize, shift left/right). The error flag is set if the user's program reads one of the relevant registers (MD0 to MD3) or if it writes to MD0 again before the shift operation has been completed.

Please note that the error flag mechanism is just an option to monitor the MDU operation. If the user's program is designed such that an MDU operation cannot be interrupted by other calculations, then there is no need to pay attention to the error flag. In this case it is also possible to change the order in which the MDx registers are read, or even to skip some register read instructions. Concerning the shift or normalize instructions, it is possible to read the result before the complete execution time of six machine cycles has passed (e.g. when a small number of shifts has been programmed). All of the above "illegal" actions would set the error flag, but on the other hand do not affect a correct MDU operation. The user has just to make sure that everything goes right.

The error flag (MDEF) is located in ARCON and can be read only. It is automatically cleared after being read.

Power Saving Modes

The SAB 80C517 provides - due to Siemens ACMOS technology - three modes in which power consumption can be significantly reduced.

Idle mode
The CPU is gated off from the oscillator. All peripherals are still provided with the clock and are able to work.

Power-down mode
Operation of the SAB 80C517 is completely stopped, the oscillator is turned off. This mode is used to save the contents of the internal RAM with a very low standby current.

Slow-down mode
The controller keeps up the full operating functionality, but its normal clock frequency is internally divided by eight. This slows down all parts of the controller, the CPU and all peripherals, to 1/8th of their normal operating frequency. Slowing down the frequency greatly reduces power consumption.

All of these modes - a detailed description of each is given in the following sections - are entered by software. Special function register PCON (power control register, see Figure 68 Special Function Register PCON (Address 87H)) is used to select one of these modes.

These power saving modes, especially the power-down mode, replace the hardware power-down supply for the internal RAM via a dedicated pin, as it is common with NMOS microcontrollers. During the power saving modes, the power supply for the SAB 80C517 is again via all V_{CC} pins. There is no further dedicated pin for power-down supply.

For the SAB 80C517 several provisions have been made to quality it for both electrically noisy environments and applications requiring high system security. In such applications unintentional entering of the power saving modes must be absolutely avoided. A power saving mode would reduce the controller's performance (in the case of slow-down mode) or even stop any operation (in the case of power-down mode). This situation might be fatal for the system, which is controlled by the microcontroller. Such critical applications often use the watchdog timer to prevent the system from program upsets. Then, an accidental entering of the power saving modes would even stop the watchdog timer and would circumvent the watchdog timer's task of system protection.

Hardware Enable for the Use of the Power Saving Modes

To provide power saving modes together with effective protection against unintentional entering of these modes, the SAB 80C517 has an extra pin disabling the use of the power saving modes. As this pin will most likely be used only in critical applications it is combined with an automatic start of the watchdog timer (see the description in Fail Save Mechanisms on page 6-99). This pin is called $\overline{PE}$/SWD (powers saving enable/start watchdog timer) and its function is as follows:

$\overline{PE}$/SWD = 1 (logic high level)
- Use of the power saving modes is not possible. The instruction sequences used for entering these modes will not affect the normal operation of the device.
- If and only if $\overline{PE}$/SWD is held at high level during reset, the watchdog timer is started immediately after reset is released.

$\overline{PE}$/SWD = 0 (logic low level)
- All power saving modes can be activated as described in the following sections
- The watchdog timer has to be started by software if system protection is desired.

When left unconnected, the pin $\overline{PE}$/SWD is pulled to high level by a weak internal pullup. This is done to provide system protection by default.

The logic level applied to pin $\overline{PE}$/SWD can be changed during program execution in order to allow or block the use of the power saving modes without any effect on the on-chip watchdog circuitry; (the watchdog timer is started only if $\overline{PE}$/SWD is on high level at the moment when reset is released; a change at $\overline{PE}$/SWD during program execution has no effect on the watchdog timer; this only enables or disables the use of the power saving modes.). A change of the pin's level is detected in state 3, phase 1. A Schmitt trigger is used at the input to reduce susceptibility to noise.

In addition to the hardware enable/disable of the power saving modes, a double-instruction sequence which is described in the corresponding sections is necessary to enter power-down and idle mode. The combination of all these safety precautions provide a maximum of system protection.

Application Example for Switching Pin $\overline{PE}$/SWD

For most applications in noisy environments, components external to the chip are used to give warning of a power failure or a turn off of the power supply. These circuits could be used to control the $\overline{PE}$/SWD pin. The possible steps to go into power-down mode could then be as follows:

A power-fail signal forces the controller to go into a high priority interrupt routine. This interrupt routine saves the actual program status. At the same time pin $\overline{PE}$/SWD is pulled low by the power-fail signal.

Finally the controller enters power-down mode by executing the relevant double-instruction sequence.

Idle Mode

In idle mode the oscillator of the SAB 80C517 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial channels, the A/D converter, the oscillator watchdog, the division/multiplication unit and all timers, except for the watchdog timer, are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption, which can be achieved by this feature, depends on the number of peripherals running. If all timers are stopped and the A/D converter and the division/multiplication unit are not running, maximum power reduction can be achieved. This state is also the test condition for the idle I_{CC} (see the DC characteristics in the data sheet).

Thus, the user has to take into account that the right peripheral continues to run or is stopped, respectively, during idle. Also, the state of all port pins - either the pins controlled by their latches or controlled by their secondary functions - depends on the status of the controller when entering idle.

Normally the port pins hold the logical state they had at the time idle was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle if the assigned function is on. This applies for the compare outputs as well as for the system clock output signal and the serial interface in case the latter could not finish reception or transmission during normal operation. The control signals ALE and $\overline{PSEN}$ are held at logic high levels (see Table 13 Status of External Pins During Idle and Power-Down Mode).

During idle, as in normal operating mode, the ports can be used as inputs. Thus, a capture or reload operation as well as an A/D conversion can be triggered, the timers can be used to count external events and external interrupts can be detected.

Table 13 Status of External Pins During Idle and Power-Down Mode

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	Power-down	Idle	Power-down
ALE	High	Low	High	Low
PSEN	High	Low	High	Low
Port 0	Data	Data	Float	Float
Port 1	Data/alternate outputs	Data/ last output	Data/alternate outputs	Data/ last output
Port 2	Data	Data	Address	Data
Port 3	Data/alternate outputs	Data/ last output	Data/alternate outputs	Data/ last output
Port 4	Data/alternate outputs	Data last output	Data/alternate outputs	Data/ last output
Port 5	Data/alternate outputs	Data/ last output	Data/alternate outputs	Data/ last output
Port 6	Data/alternate outputs	Data/ last output	Data/alternate outputs	Data/ last output

The watchdog timer is the only peripheral which is automatically stopped during idle. The idle mode makes it possible to "freeze" the processor's status for a certain time or until an external event causes the controller to go back into normal operating mode. Since the watchdog timer is stopped during idle mode, this useful feature of the SAB 80C517 is provided even if the watchdog function is used simultaneously.

If the idle mode is to be used the pin $\overline{PE}$/SWD must be held low. Entering the idle mode is to be done by two consecutive instructions immediately following each other. The first instruction has to set the flag bit IDLE (PCON.0) and must not set bit IDLS (PCON.5), the following instruction has to set the start bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). The hardware ensures that a concurrent setting of both bits, IDLE and IDLS will not initiate the idle mode. Bits IDLE and IDLS will automatically be cleared after having been set. If one of these register bits is read the value shown is zero (0). Figure 68 Special Function Register PCON (Address 87H) shows special function register PCON. This double-instruction sequence is implemented to minimize the chance of unintentionally entering the idle mode.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the idle mode is to be done by byte handling instructions.

The following instruction sequence may serve as an example:

```
ORL   PCON,#00000001B   ;Set bit IDLE,
                        ;bit IDLS must not be set
ORL   PCON,#00100000B   ;Set bit IDLS,
                        ;bit IDLE must not be set
```

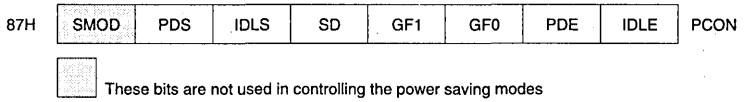
The instruction that sets bit IDLS is the last instruction executed before going into idle mode.

Terminating the Idle Mode

The idle mode can be terminated by activation of any enabled interrupt. The CPU operation is resumed, the interrupt will be serviced and the next instruction to be executed after the RETI instruction will be the one following the instruction that set the bit IDLS.

The other possibility of terminating the idle mode is a hardware reset. Since the oscillator is still running, the hardware reset is held active for only two machine cycles for a complete reset.

Figure 68 Special Function Register PCON (Address 87H)



Bit	Function
PDS	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	IDLE start bit. The instruction that sets the IDSL flag bit is the last instruction before entering the idle mode.
SD	When set, the slow-down mode is enabled.
GF1	General purpose flag
GF0	General purpose flag
PDE	Power-down enable bit. When set, starting the power-down mode is enabled.
IDLE	Idle mode enable bit. When set, starting the idle mode is enabled.

Power-Down Mode

In the power-down mode, the on-chip oscillator is stopped. Therefore, all functions are stopped, only the contents of the on-chip RAM and the SFR's are held. The port pins controlled by their port latches output the values that are held by their SFR'S. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode; when enabled, the clockout signal (P1.6/CLKOUT) will stop at low level. ALE and PSEN are held at logic low level (see Figure 13 Output Driver Circuit of Ports 1 through 6).

If the power-down mode is to be used, the pin PE/SWD must be held low. Entering the power-down mode is done by two consecutive instructions immediately following each other. The first instruction has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6). The following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, will not initiate the power-down mode. Bit PDE and PDS will automatically be cleared after having been set and the value shown when reading one of these bits is always zero (0).

Figure 68 Special Function Register PCON (Address 87H) shows the special function register PCON. This double- instruction sequence is implemented to minimize the chance of unintentional entering the power-down mode, which could possibly "freeze" the chip's activity in an undesired status.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the power-down mode is composed of byte handling instructions.

The following instruction sequence may serve as an example:

The instruction that sets bit PDS is the last

```
ORL   PCON,#00000010B  ;Set bit PDE,
                       ;bit PDS must not be set

ORL   PCON,#01000000B  ;Set bit PDS,
                       ;bit PDE must not be set
```

instruction executed before going into power-down mode. If idle mode and power-down mode are invoked simultaneously, the power-down mode takes precedence.

The only exit from power-down mode is a hardware reset. Reset redefines all SFR'S, but will not change the contents of the internal RAM.

In the power-down mode, V_{CC} can be reduced to minimize power consumption. Care must be taken, however, to ensure that V_{CC} is not reduced before the power-down mode is invoked, and that V_{CC} is restored to its normal operating level before the power-down mode is terminated. The reset signal that terminates the power-down mode also frees the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

Slow-Down Mode

In some applications, where power consumption and dissipation is critical, the controller might run for a certain time at reduced speed (e.g. if the controller is waiting for an input signal). Since in CMOS devices there is an almost linear interdependence of the operating frequency and the power supply current, a reduction of the operating frequency results in reduced power consumption.

In the slow-down mode all signal frequencies that are derived from the oscillator clock are divided by eight. This also includes the clockout signal at pin P1.6/CLKOUT.

If the slow-down mode is to be used the pin $\overline{PE}$/SWD must be held low.

The slow-down mode is entered by setting bit SD (PCON.4), see Figure 68 Special Function Register PCON (Address 87H). The controller actually enters the slow-down mode after a short synchronization period (max. two machine cycles). The slow-down mode can be used together with idle and power-down mode.

The slow-down mode is disabled by clearing bit SD.

Fail Save Mechanisms

The SAB 80C517 offers two on-chip peripherals which monitor the program flow and ensure an automatic "fail-safe" reaction for cases where the controller's hardware fails or the software hangs up:

A programmable watchdog timer (WDT) with variable time-out period from 512 microseconds up to approx. 1.1 seconds at 12 MHz.
The SAB 80C517's WDT is a superset of the SAB 80515 watchdog.

An oscillator watchdog (OWD) which monitors the on-chip oscillator and forces the microcontroller into the reset state if the on-chip oscillator fails.

Programmable Watchdog Timer

To protect the system against software upset, the user's program has to clear this watchdog within a previously programmed time period. If the software fails to do this periodical refresh of the watchdog timer, an internal hardware reset will be initiated. The software can be designed so that the watchdog times out if the program does not work properly. It also times out if a software error is based on hardware-related problems.

The watchdog timer in the SAB 80C517 is a 15-bit timer, which is incremented by a count rate of either $f_{CYCLE}/2$ or $f_{CYCLE}/32$ ($f_{CYCLE} = f_{OSC}/12$). That is, the machine clock is divided by a series arrangement of two prescalers, a divide-by-two and a divide-by-16 prescaler (see Figure 69 Block Diagram of the Programmable Watchdog Timer). The latter is enabled by setting bit WDTREL.7.

Immediately after start (see next section for the start procedure), the watchdog timer is initialized to the reload value programmed to WDTREL.0 - WDTREL.6. After an external HW reset (or power-on reset), register WDTREL is cleared to 00H. The lower seven bits of WDTREL can be loaded by software at any time.

Examples (given for a 12-MHz oscillator frequency):

WDTREL =	Time-Out Period	Comments
00H	65.535 ms	This is the default value and coincides with the watchdog period of the SAB 80515
80H	1.1 s	Maximum time period
7FH	512 µs	Minimum time period

Starting the Watchdog Timer

There are two ways to start the watchdog timer depending on the level applied to pin $\overline{PE}$/SWD (pin 4). This pin serves two functions, because it is also used for blocking the power saving modes. For details see chapter 7.7.

The First Possibility of Starting the Watchdog Timer

The automatic start of the watchdog timer directly after an external HW reset is a hardware start initialized by strapping pin 4 ($\overline{PE}$/SWD) to V_{CC}. In this case the power-saving modes (power-down mode, idle mode and slow-down mode) are also disabled and cannot be started by software.

The self-start of the watchdog timer by a pin option has been implemented to provide high system security in electrically very noisy environments.

Note:

The automatic start of the watchdog timer is only performed if $\overline{PE}$/SWD (power-save enable/start watchdog timer) is held at high level while reset is active. A positive transition at this pin during normal program execution will not start the watchdog timer.

Furthermore, when using the hardware start, the watchdog timer starts running with its default time- out period. The value in the reload register WDTREL, however, can be overwritten at any time to set any time-out period desired.

The Second Possibility of Starting the Watchdog Timer

The watchdog timer can also be started by software. This method is compatible to the start procedure in the SAB 80(C)515. Setting of bit SWDT in special function register IEN1 (Figure 72 Special Function Register IEN1) starts the watchdog timer. Using the software start, the time-out period can be programmed before the watchdog timer starts running.

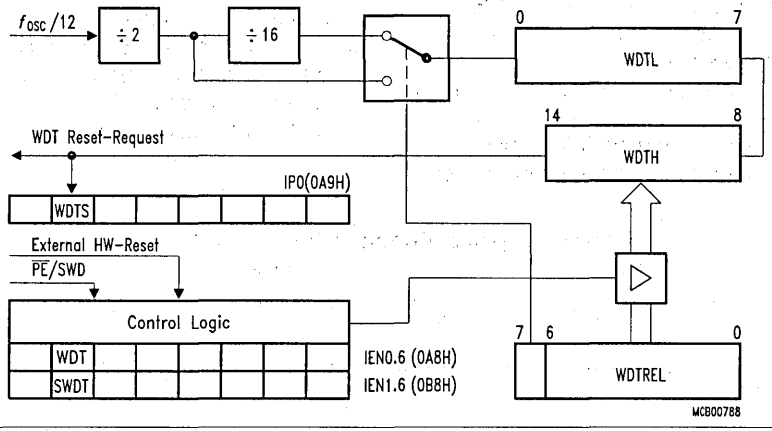
Note that once the watchdog timer has been started it cannot be stopped by anything but an external hardware reset through pin 10 with a low level applied to pin $\overline{PE}$/SWD.

Refreshing the Watchdog Timer

At the same time the watchdog timer is started, the 7-bit register WDTH is preset by the contents of WDTREL.0 to WDTREL.6. Once started the watchdog cannot be stopped by software but can only be refreshed to the reload value by first setting bit WDT (IEN0.6) and by the next instruction setting SWDT (IEN1.6). Bit WDT will automatically be cleared during the second machine cycle after having been set. For this reason, setting SWDT bit has to be a one cycle instruction (e.g. SETB SWDT). This double-instruction refresh of the watchdog timer is implemented to minimize the chance of an unintentional reset of the watchdog.

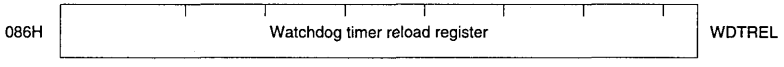
The reload register WDTREL can be written to at any time, as already mentioned. Therefore, a periodical refresh of WDTREL can be added to the above mentioned starting procedure of the watchdog timer. Thus a wrong reload value caused by a possible distortion during the write operation to the WDTREL can be corrected by software.

Figure 69
Block Diagram of the Programmable Watchdog Timer



6

Figure 70 Special Function Register WDTREL

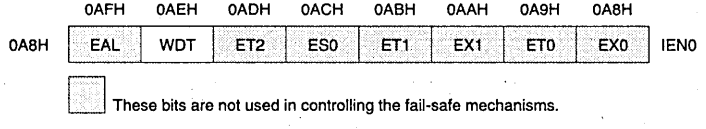


Bit	Function
WDTREL.7	Prescaler select bit. When set, the watchdog is clocked through an additional divide-by-16 prescaler (see Figure 69 Block Diagram of the Programmable Watchdog Timer).
WDTREL.6 to WDTREL.0	Seven bit reload value for the high-byte of the watchdog timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

Watchdog Reset and Watchdog Status Flag

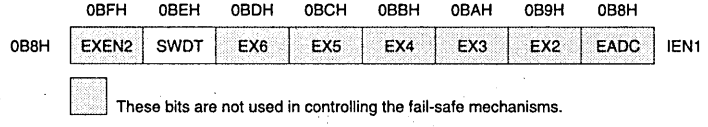
If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state 7FFCH. The duration of the reset signal then depends on the prescaler selection (either 8 cycles or 128 cycles). This internal reset differs from an external one only in so far as the watchdog timer is not disabled and bit WDTS (watchdog timer status, bit 6 in special function register IP0) is set. Figure 73 Watchdog Status Flags and Reset Requests shows a block diagram of all reset requests in the SAB 80C517 and the function of the watchdog status flags. The WDTS flag is a flip-flop, which is set by a watchdog timer reset and cleared by an external HW reset. Bit WDTS allows the software to examine from which source the reset was activated. The watchdog timer status flag can also be cleared by software.

Figure 71 Special Function Register IEN0



Bit	Function
WDT	Watchdog timer refresh flag. Set to initiate a refresh of the watchdog timer. Must be set directly before SWDT is set to prevent an unintentional refresh of the watchdog timer.

Figure 72 Special Function Register IEN1



Bit	Function
SWDT	Watchdog timer start flag. Set to activate the watchdog timer. When directly set after setting WDT, a watchdog timer refresh is performed.

Figure 73
Watchdog Status Flags and Reset Requests

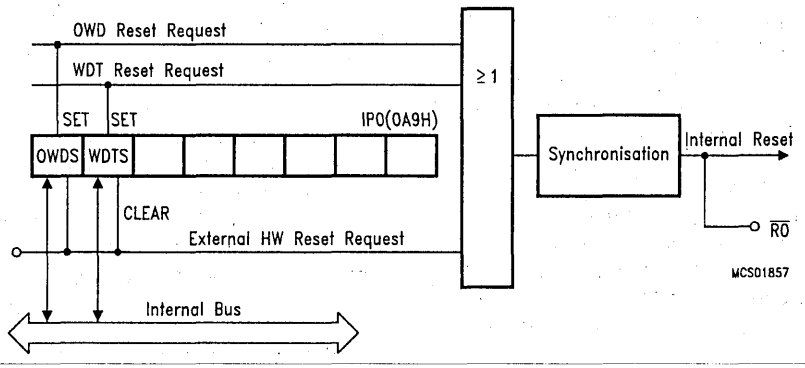


Figure 74 Special Function Register IP0

0A9H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0

These bits are not used in controlling the fail-safe mechanisms.

Bit	Function
OWDS	Oscillator watchdog timer status flag. Set by hardware when an oscillator watchdog reset occurred. Can be cleared or set by software
WDTS	Watchdog timer status flag. Set by hardware when a watchdog timer reset occurred. Can be cleared or set by software

Oscillator Watchdog

What happens in a microcontroller system if the controller's on-chip oscillator stops working? This failure e.g. caused by a broken crystal, an open connection to the crystal, or a long-term disturbance normally leaves the system in a random, undetermined state. The SAB 80C517 provides a "fail-safe" reaction upon an oscillator failure. If the on-chip oscillator frequency falls below a certain limit due to a hardware defect, the oscillator watchdog initiates an internal reset. This reset state is maintained until the on-chip oscillator is working again. This ensures a maximum of system protection with a minimum of susceptibility to distortion or to operating errors.

In the reset state all port pins of the SAB 80C517 show a '1'.

The oscillator watchdog consists of an integrated RC oscillator combined with a frequency comparator. If the on-chip oscillator's frequency falls below the frequency of the RC oscillator, the comparator generates a signal which initiates a reset.

The RC oscillator runs with a frequency of typically 300 kHz and works without any external components. It also determines, as long as it is used, the lower limit of the SAB 80C517's operating frequency, which is therefore specified at 1 MHz.

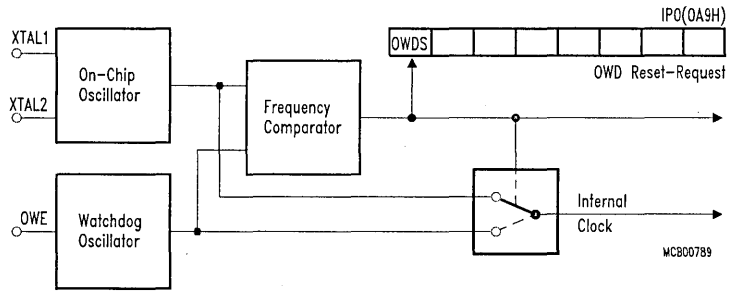
Since the frequency comparator of the oscillator watchdog takes its inputs directly from the on-chip oscillator, the minimum frequency of 1 MHz does not restrict the use of the slow-down mode. In this mode the CPU runs with one eighth of the normal clock rate (see Power Saving Modes on page 6-95).

The oscillator watchdog circuitry can be enabled externally. If the OWE pin (oscillator watchdog enable) is pulled low, the oscillator watchdog function is off. If the pin is left unconnected or has a logic high level, the watchdog oscillator is activated. Thus, the watchdog is enabled even if the pin or the path to the pin is broken.

Like the watchdog timer circuitry, the oscillator watchdog circuitry contains a status flip-flop. This flip-flop is set when an oscillator failure is detected and it is cleared by an external HW reset or by software (see Figure 73 Watchdog Status Flags and Reset Requests).

The block diagram in Figure 75 Functional Block Diagram of the Oscillator Watchdog illustrates the function of the oscillator watchdog. Note that the OWD reset request is held for at least three additional cycles after the on-chip oscillator returns to normal operation. This is done to ensure a proper oscillator startup.

Figure 75
Functional Block Diagram of the Oscillator Watchdog



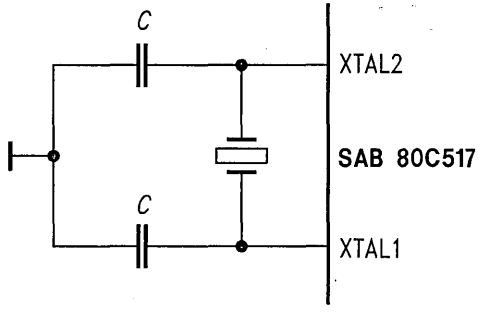
Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components as a Pierce oscillator. The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clock signals to the chip at half the oscillator frequency. These signals define the internal phases, states and machine cycles, as described in chapter 3.

Figure 76 Recommended Oscillator Circuit shows the recommended oscillator circuit.

6

Figure 76
Recommended Oscillator Circuit



$C = 30\ pF \pm 10\ pF$ for Quartz Crystal

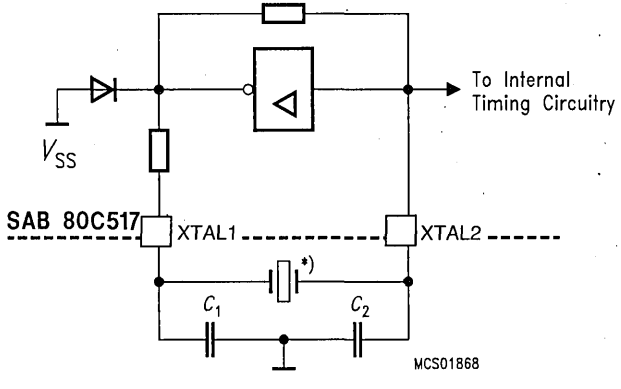
MCS01867

In this application the on-chip oscillator is used as a crystal-controlled, positive-reactance oscillator (a more detailed schematic is given in Figure 77 On-Chip Oscillator Circuitry). It is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. The crystal specifications and capacitances are non-critical. In this circuit 30 pF can be used as single

capacitance at any frequency together with a good quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. It a ceramic resonator is used, C_1 and C_2 are normally selected to be of somewhat higher values, typically 47 pF. We recommend consulting the manufacturer of the ceramic resonator for value specifications of these capacitors.

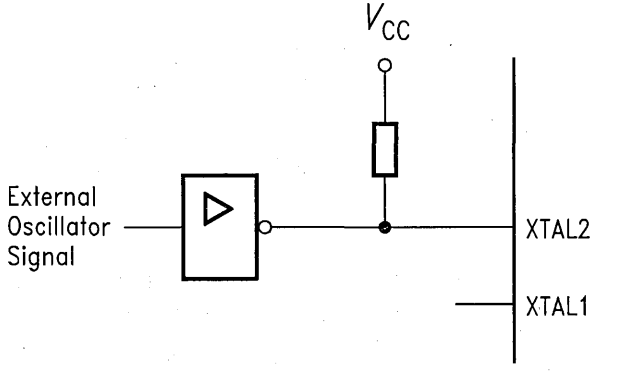
To drive the SAB 80 C517 with an external clock source, the external clock signal is to be applied to XTAL2, as shown in Figure 78 External Clock Source. XTAL1 has to be left unconnected. A pullup resistor is suggested (to increase the noise margin), but is optional if V_{OH} of the driving gate corresponds to the V_{IH2} specification of XTAL2.

Figure 77
On-Chip Oscillator Circuitry



*) Quartz Crystal or Ceramic Resonator

Figure 78
External Clock Source



System Clock Output

For peripheral devices requiring a system clock, the SAB 80C517 provides a clock output signal derived from the oscillator frequency as an alternate output function on pin P1.6/CLKOUT. If bit CLK is set (bit 6 of special function register ADCON0, see Figure 79 Special Function Register ADCON0 (Address 0D8H)), a clock signal with 1/12 of the oscillator frequency is gated to pin P1.6/CLKOUT. To use this function the port pin must be programmed to a one (1), which is also the default after reset.

Figure 79 Special Function Register ADCON0 (Address 0D8H)

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0

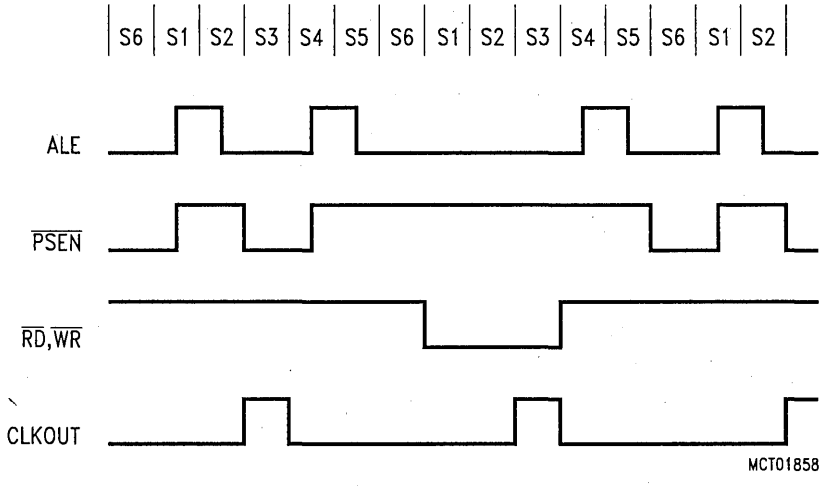
These bits are not used in controlling the clock out functions.

Bit	Function
CLK	Clockout enable bit. When set, pin P1.6/CLKOUT outputs the system clock which is 1/12 of the oscillator frequency.

The system clock is high during S3P1 and S3P2 of every machine cycle and low during all other states. Thus, the duty cycle of the clock signal is 1:6. Associated with a MOVX instruction the system clock coincides with the last state (S3) in which a $\overline{RD}$ or $\overline{WR}$ signal is active. A timing diagram of the system clock output is shown in Figure 80 Timing Diagram - System Clock Output.

Note: During slow-down operation (see Power Saving Modes on page 6-95) the frequency of the clockout signal is divided by eight.

Figure 80
Timing Diagram - System Clock Output



Interrupt System

The SAB 80C517 provides 14 interrupt sources with four priority levels. Seven interrupts can be generated by the on-chip peripherals (i.e. timer 0, timer 1, timer 2, compare timer, serial interfaces 0 and 1 and A/D converter), and seven interrupts may be triggered externally.

Short Description of the Interrupt Structure for Advanced SAB 80(C)515 Users

The interrupt structure of the SAB 80C517 has been mainly adapted from the SAB 80(C)515. Thus, each interrupt source has its dedicated interrupt vector and can be enabled/disabled individually; there are also four priority levels available.

In the SAB 80C517 two interrupt sources have been added:

- Compare timer overflow interrupt
- Receive and transmit interrupt of serial interface 1

In the SAB 80(C)515 the 12 interrupt sources are combined to six pairs; each pair can be programmed to one of the four interrupt priority levels. In the SAB 80C517 the new interrupt sources were added to two of these pairs, thus forming triplets; therefore, the 14 interrupt sources are combined to six pairs or triplets; each pair or triplet can be programmed to one of the four interrupt priority levels (see "Priority Level Structure" on page 116)

Figure 81 a) Interrupt Structure of the SAB 80C517 gives a general overview of the interrupt sources and illustrates the request and control flags described in the next sections. The priority structure and the corresponding control bits are listed in "Priority Level Structure" on page 116.

Interrupt Structure

A common mechanism is used to generate the various interrupts, each source having its own request flag(s) located in a special function register (e.g. TCON, IRCON, S0CON, S1CON). Provided the peripheral or external source meets the condition for an interrupt, the dedicated request flag is set, whether an interrupt is enabled or not. For example, each timer 0 overflow sets the corresponding request flag TF0. If it is already set, it retains a one (1). But the interrupt is not necessarily serviced.

Now each interrupt requested by the corresponding flag can individually be enabled or disabled by the enable bits in SFR's IEN0, IEN1, IEN2 (see Figures 82, 83, and 84). This determines whether the interrupt will actually be performed. In addition, there is a global enable bit for all interrupts which, when cleared, disables all interrupts independent of their individual enable bits.

Figure 81 a)
Interrupt Structure of the SAB 80C517

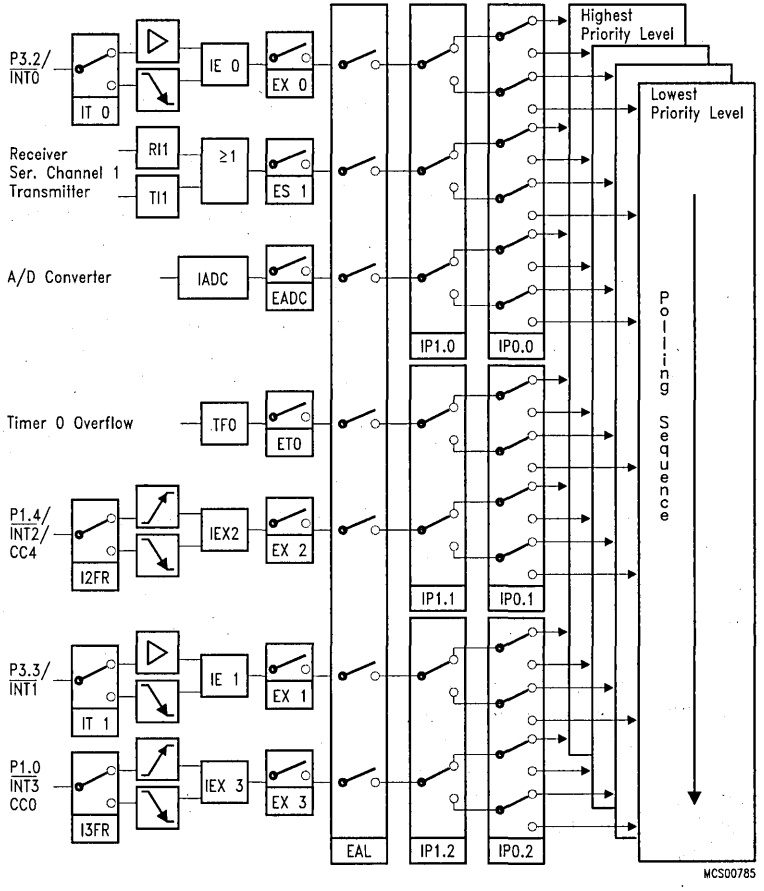


Figure 81 b)
Interrupt Structure of the SAB 80C517 (cont'd)

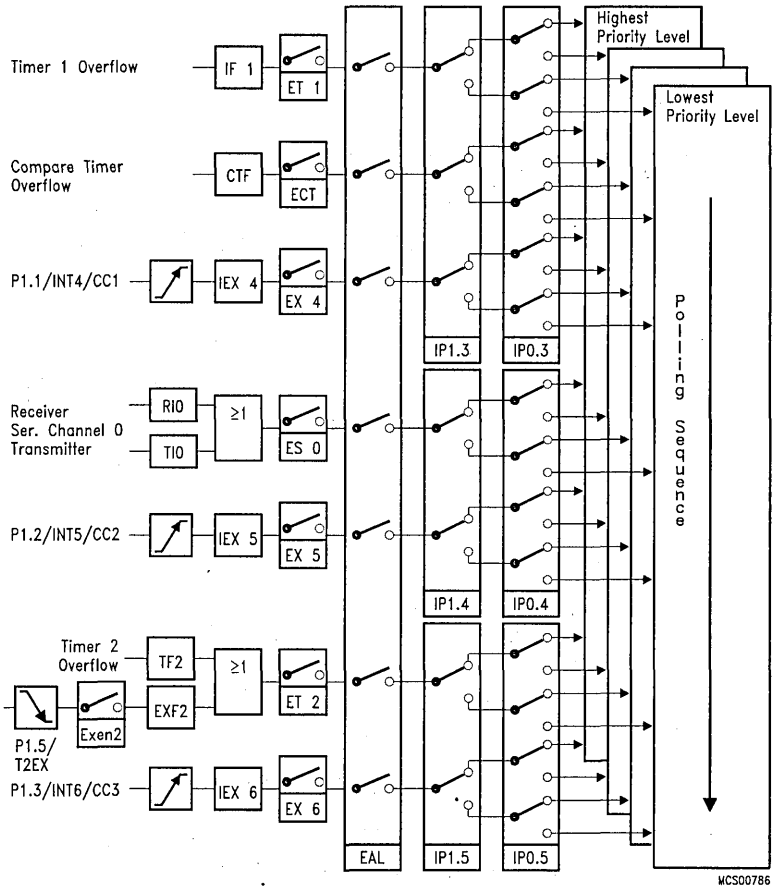


Figure 82 Special Function Register IEN0 (Address 0A8H)

	0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H	
0A8H	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0	IEN0

This bit is not used for interrupt control.

Bit	Function
EX0	Enables or disables external interrupt 0. If EX0 = 0, external interrupt 0 is disabled.
ET0	Enables or disables the timer 0 overflow interrupt. If ET0 = 0, the timer 0 interrupt is disabled.
EX1	Enables or disables external interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ET1	Enables or disables the timer 1 overflow interrupt. If ET1 = 0, the timer 1 interrupt is disabled.
ES0	Enables or disables the serial channel 0 interrupt. If ES0 = 0, the serial channel 0 interrupt is disabled.
ET2	Enables or disables the timer 2 overflow or external reload interrupt. If ET2 = 0, the timer 2 interrupt is disabled.
EAL	Enables or disables all interrupts. If EAL = 0, no interrupt will be acknowledged. If EAL = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

Figure 83 Special Function Register IEN1 (Address 0B8H)

	0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	
0B8H	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

bit is not used for interrupt control.

Bit	Function
EADC	Enables or disables the A/D converter interrupt. If EADC = 0, the A/D converter interrupt is disabled.
EX2	Enables or disables external interrupt 2/capture/compare interrupt 4. If EX2 = 0, external interrupt 2 is disabled.
EX3	Enables or disables external interrupt 3/capture/compare interrupt 0. If EX3 = 0, external interrupt 3 is disabled.
EX4	Enables or disables external interrupt 4/capture/compare interrupt 1. If EX4 = 0, external interrupt 4 is disabled.
EX5	Enables or disables external interrupt 5/capture/compare interrupt 2. If EX5 = 0, external interrupt 5 is disabled.
EX6	Enables or disables external interrupt 6/capture/compare interrupt 3. If EX6 = 0, external interrupt 6 is disabled.
EXEN2	Enables or disables the timer 2 external reload interrupt. EXEN2 = 0 disables the timer 2 external reload interrupt. The external reload function is not affected by EXEN2.

Figure 84 Special Function Register IEN2 (Address 09AH)

09AH	–	–	–	–	ECT	–	–	ES1	IEN2

Bit	Function
ES1	Enable serial interrupt of interface 1. Enables or disables the interrupt of serial interface 1. If ES1 = 0, the interrupt is disabled.
ECT	Enable compare timer interrupt. Enables or disables the interrupt at compare timer overflow. If ECT = 0, the interrupt is disabled.

In the following the interrupt sources are discussed individually.

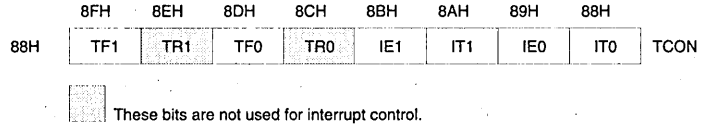
The external interrupts 0 and 1 ($\overline{INT0}$ and $\overline{INT1}$) can each be either level-activated or negative transition-activated, depending on bits IT0 and IT1 in register TCON (see Figure 85 Special Function Register TCON (Address 88H)). The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated this interrupt is cleared by the hardware when the service routine is vectored to, but only if the interrupt was transition-activated. If the interrupt was level-activated, then the requesting external source directly controls the request flag, rather than the on-chip hardware.

The timer 0 and timer 1 interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective timer/counter registers (exception see "Mode 3" on page 55 for timer 0 in mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored too.

The two interrupts of the serial interfaces are generated by the request flags RI0 and TI0 (in register S0CON) or RI1 and TI1 (in register S1CON), respectively. That is, the two request flags of each serial interface are logically OR-ed together. Neither of these flags is cleared by hardware when the service routine is vectored too. In fact, the service routine of each interface will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, and the bit will have to be cleared by software.

The timer 2 interrupt is generated by the logical OR of bit TF2 in register T2CON and bit EXF2 in register IRCON. Figure 86 Special Function Register T2CON (Address 0C8H) and Figure 87 Special Function Register IRCON (Address 0C0H) show SFR's T2CON and IRCON. Neither of these flags is cleared by hardware when the service routine is vectored too. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared by software.

Figure 85 Special Function Register TCON (Address 88H)

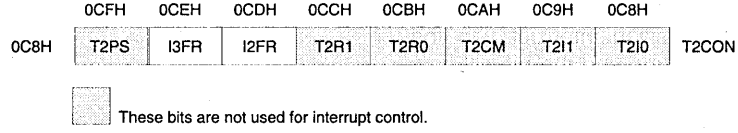


Bit	Function
IT0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE0	Interrupt 0 edge flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low-level triggered external interrupts.
IE1	Interrupt 1 edge flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

The A/D converter interrupt is generated by IADC in register IRCON (see Figure 87 Special Function Register IRCON (Address 0C0H)). It is set some cycles before the result is available. That is, if an interrupt is generated, in any case the converted result in ADDAT is valid on the first instruction of the interrupt service routine (with respect to the minimal interrupt response time). If continuous conversions are established, IADC is set once during each conversion. If an A/D converter interrupt is generated, flag IADC will have to be cleared by software.

The external interrupt 2 ($\overline{INT2}$/CC4) can be either positive or negative transition-activated depending on bit I2FR in register T2CON (see Figure 86 Special Function Register T2CON (Address 0C8H)). The flag that actually generates this interrupt is bit IEX2 in register IRCON. In addition, this flag will be set if a compare event occurs at the corresponding output pin P1.4/$\overline{INT2}$/CC4, regardless of the compare mode established and the transition at the respective pin. If an interrupt 2 is generated, flag IEX2 is cleared by hardware when the service routine is vectored too.

Figure 86 Special Function Register T2CON (Address 0C8H)



Bit	Function
I2FR	External interrupt 2 falling/rising edge flag. When set, the interrupt 2 request flag IEX2 will be set on a positive transition at pin P1.4/$\overline{INT2}$. I2FR = 0 specifies external interrupt 2 to be negative-transition activated.
I3FR	External interrupt 3 falling/rising edge flag. When set, the interrupt 3 request flag IEX3 will be set on a positive transition at pin P1.0/$\overline{INT3}$. I3FR = 0 specifies external interrupt 3 to be negative-transition active.

Like the external interrupt 2, **the external interrupt 3** can be either positive or negative transition-activated, depending on bit I3FR in register T2CON. The flag that actually generates this interrupt is bit IEX3 in register IRCON. In addition, this flag will be set if a compare event occurs at pin P1.0/INT3/CC0, regardless of the compare mode established and the transition at the respective pin. The flag IEX3 is cleared by hardware when the service routine is vectored too.

The external interrupts 4 (INT4), 5 (INT5), 6 (INT6) are positive transition-activated. The flags that actually generate these interrupts are bits IEX4, IEX5, and IEX6 in register IRCON (see Figure 87 Special Function Register IRCON (Address 0C0H)). In addition, these flags will be set if a compare event occurs at the corresponding output pin P1.1/INT4/CC1, P1.2/INT5/CC2, and P1.3/INT6/CC3, regardless of the compare mode established and the transition at the respective pin. When an interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored too.

The compare timer interrupt is generated by bit CTF in register CTCON (see Figure 88 Special Function Register CTCON (Address 0E1H)), which is set by a rollover in the compare timer. If a compare timer interrupt is generated, flag CTF will have to be cleared by software.

Figure 87 Special Function Register IRCON (Address 0C0H)

	0C7H	0C6H	0C5H	0C4H	0C3H	0C2H	0C1H	0C0H	
0C0H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

Bit	Function
IADC	A/D converter interrupt request flag. Set by hardware at the end of a conversion. Must be cleared by software.
IEX2	External interrupt 2 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.4/INT2/CC4. Cleared when interrupt processed.
IEX3	External interrupt 3 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.0/INT3/CC0. Cleared when interrupt processed.
IEX4	External interrupt 4 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.1/INT4/CC1. Cleared when interrupt processed.
IEX5	External interrupt 5 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.2/INT5/CC2. Cleared when interrupt processed.
IEX6	External interrupt 6 edge flag. Set by hardware when external interrupt edge was detected or when a compare event occurred at pin 1.3/INT6/CC3. Cleared when interrupt processed.
TF2	Timer 2 overflow flag. Set by timer 2 overflow. Must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
EXF2	Timer 2 external reload flag. Set when a reload is caused by a negative transition on pin T2EX while EXEN2 = 1. When the timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector the timer 2 interrupt routine. Can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.

Figure 88 Special Function Register CTCON (Address 0E1H)

0E1H	T2PS1 [1]	–	–	–	CTF	CLK2	CLK1	CLK0	CTCON

These bits are not used for interrupt control.

Bit	Function
CTF	Compare timer overflow. Set by hardware at a rollover of the compare timer. Must be cleared by software. If the compare timer interrupt is enabled. CTF = 1 will cause an interrupt.

All of these bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The only exceptions are the request flags IE0 and IE1. If the external interrupts 0 and 1 are programmed to be level-activated, IE0 and IE1 are controlled by the external source via pin $\overline{INT0}$ and $\overline{INT1}$, respectively. Thus, writing a one to these bits will not set the request flag IE0 and/or IE1. In this mode, interrupts 0 and 1 can only be generated by software and by writing a 0 to the corresponding pins $\overline{INT0}$ (P3.2) and $\overline{INT1}$ (P3.3), provided that this will not affect any peripheral circuit connected to the pins.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the special function registers IEN0, IEN1 and IEN2 (Figures 82, 83, and 84). Note that IEN0 contains also a global disable bit, EAL, which disables all interrupts at once. Also note that in the SAB 8051 the interrupt priority register IP is located at address 0B8H; in the SAB 80C517 this location is occupied by register IEN1.

1)Only available in SAB 80C517 identification mark 'BB' or later.

Priority Level Structure

As already mentioned above, all interrupt sources are combined as pairs or triplets; Table 14 Pairs and Triplets of Interrupt Sources lists the structure of the interrupt sources.

Table 14 Pairs and Triplets of Interrupt Sources

External interrupt 0	Serial channel 1 interrupt	A/D converter interrupt
Timer 0 interrupt	–	External interrupt 2
External interrupt 1	–	External interrupt 3
Timer 1 interrupt	Compare timer interrupt	External interrupt 4
Serial channel 0 interrupt	–	External interrupt 5
Timer 2 interrupt	–	External interrupt 6

Each pair or triplet of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register IP0 and one in IP1 (Figure 89 Special Function Registers IP0 and IP1 (Address 0A9H and 0B9H)). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another interrupt of the same or a lower priority. An interrupt of the highest priority level cannot be interrupted by another interrupt source.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is to be serviced first. Thus, within each priority level there is a second priority structure determined by the polling sequence, as follows (see Figure 90 Priority-Within-Level Structure.):

- Within one pair or triplet the leftmost interrupt is serviced first, then the second and third, when available.
- The pairs or triplets are serviced from top to bottom of the table.

Figure 89 Special Function Registers IP0 and IP1 (Address 0A9H and 0B9H)

0A9H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0
0B9H	–	–	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	IP1

These bits are not used for interrupt control.

Corresponding bit locations in both registers are used to set the interrupt priority level of an interrupt pair or triplet.

Bit		Function
IP1.x	IP0.x	–
0	0	Set priority level 0 (lowest)
0	1	Set priority level 1
1	0	Set priority level 2
1	1	Set priority level 3 (highest)

Bit	Function
IP1.0/IP0.0	IE0/RI1 + TI1/IADC
IP1.1/IP0.1	TF0/IEX2
IP1.2/IP0.2	IE1/IEX3
IP1.3/IP0.3	TF1/CTF/IEX4
IP1.4/IP0.4	RI0 + TI0/IEX5
IP1.5/IP0.5	TF2 + EXF2/IEX6

Figure 90 Priority-Within-Level Structure.

High	→	Low	Priority
	Interrupt Source		
IE0	RI1+TI1	IADC	High
TF0		IEX2	
IE1		IEX3	↓
TF1	CTF	IEX4	
RI0 + TI0	–	IEX5	
TF2 + EXF2	–	IEX6	Low

Note: This "priority-within-level" structure is only used to resolve simultaneous requests of the same priority level.

How Interrupts are Handled

The interrupt flags are sampled at S5P2 in each machine cycle. The sampled flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate a LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1) An interrupt of equal or higher priority is already in progress.

2) The current (polling) cycle is not in the final cycle of the instruction in progress.

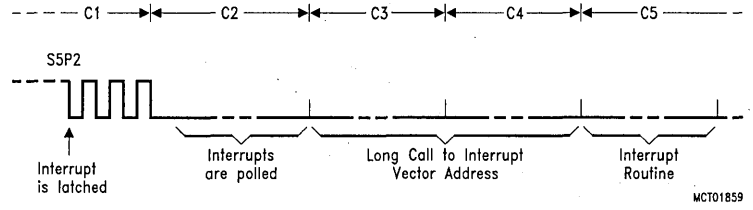
3) The instruction in progress is RETI or any write access to registers IEN0, IEN1, IEN2 or IP0 and IP1.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress is completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write access to registers IEN0, IEN1, IEN2 or IP0 and IP1, then at least one more instruction will be executed before any interrupt is vectored too; this delay guarantees that changes of the interrupt status can be observed by the CPU.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if any interrupt flag is active but not being responded to for one of the conditions already mentioned, or if the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle interrogates only the pending interrupt requests.

The polling cycle/LCALL sequence is illustrated in Figure 91 Interrupt Response Timing Diagram .

Figure 91
Interrupt Response Timing Diagram



Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in Figure 91 Interrupt Response Timing Diagram , then, in accordance with the above rules, it will be vectored to during C5 and C6 without any instruction for the lower priority routine to be executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not; then this has to be done by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored too, as shown in the following table.

Table 15 Interrupt Source and Vectors

Interrupt Request Flags	Interrupt Vector Address	Interrupt Source
IE0	0003H	External interrupt 0
TF0	000BH	Timer 0 overflow
IE1	0013H	External interrupt 1
TF1	001BH	Timer 1 overflow
RI0/TI0	0023H	Serial channel 0
TF2/EXF2	002BH	Timer 2 overflow/ext. reload
IADC	0043H	A/D converter
IEX2	004BH	External interrupt 2
IEX3	0053H	External interrupt 3
IEX4	005BH	External interrupt 4
IEX5	0063H	External interrupt 5
IEX6	006BH	External interrupt 6
RI1/TI1	0083H	Serial channel 1
CTF	009BH	Compare timer overflow

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the two top bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt priority level. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress. In this case no interrupt of the same or lower priority level would be acknowledged.

External Interrupts

The external interrupts 0 and 1 can be programmed to be level-activated or negative-transition activated by setting or clearing bit IT0 or IT1, respectively, in register TCON (see Figure 85 Special Function Register TCON (Address 88H)). If ITx = 0 (x = 0 or 1), external interrupt x is triggered by a detected low level at the $\overline{\text{INTx}}$ pin. If ITx = 1, external interrupt x is negative edge-triggered. In this mode, if successive samples of the $\overline{\text{INTx}}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

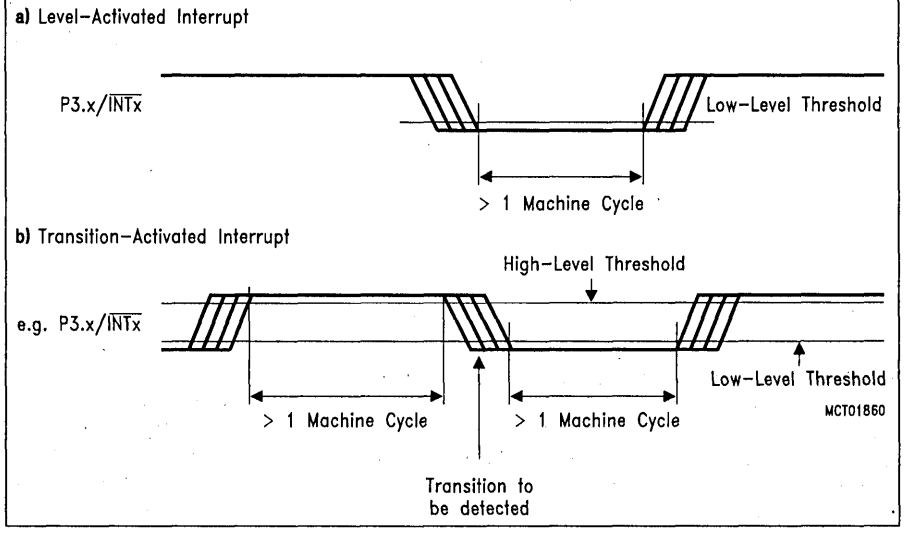
If the external interrupt 0 or 1 is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

The external interrupts 2 and 3 can be programmed to be negative or positive transition-activated by setting or clearing bit I2FR or I3FR in register T2CON (see Figure 86 Special Function Register T2CON (Address 0C8H)). If IxFR = 0 (x = 2 or 3), external interrupt x is negative transition-activated. If IxFR = 1, external interrupt is triggered by a positive transition.

The external interrupts 4, 5, and 6 are activated by a positive transition. The external timer 2 reload trigger interrupt request flag EXF2 will be activated by a negative transition at pin P1.5/T2EX but only if bit EXEN2 is set.

Since the external interrupt pins ($\overline{\text{INT2}}$ to $\overline{\text{INT6}}$) are sampled once in each machine cycle, an input high or low should be held for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin low (high for $\overline{\text{INT2}}$ and $\overline{\text{INT3}}$, if it is programmed to be negative transition-active) for at least one cycle, and then hold it high (low) for at least one cycle to ensure that the transition is recognized so that the

Figure 92
External Interrupt Detection



corresponding interrupt request flag will be set (see Figure 92 External Interrupt Detection). The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

Response Time

If an external interrupt is recognized, its corresponding request flag is set at S5P2 in every machine cycle. The value is not polled by the circuitry until the next machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus a minimum of three complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

A longer response time would be obtained if the request was blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or a write access to registers IEN0, IEN1, IEN2 or IP0, IP1, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction, if the instruction is MUL or DIV).

Thus, in a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

Instruction Set

7

Instruction Set

The instruction set of the SAB 8051 family of microcontrollers includes 111 instructions, 49 of which are single-byte, 45 two-byte, and 17 three-byte instructions. The instruction opcode format consists of a function mnemonic followed by a "destination, source" operand field. This field specifies the data type and addressing method(s) to be used.

All members of the 8051 family can be programmed with the same instruction set common to the basic member, the SAB 8051.

All microcontrollers are 100% software compatible with the SAB 8051 and may be programmed with 8051 assembler or high-level language, ASM51 or PLM51 respectively.

Addressing Modes

The SAB 8051 family of microcontrollers uses five addressing modes:

- register
- direct
- register indirect
- immediate
- base register plus index register indirect

Table summarizes which memory spaces may be accessed by each of the addressing modes.

Addressing Modes	Associated Memory Spaces
Register Addressing	R0 through R7 of Selected Register Bank ACC,B, CY (Bit), DPTR
Direct Addressing	Lower 128 Bytes of Internal RAM Special-Function Register
Register Indirect Addressing	Internal RAM (@R1, @R0, SP) External Data Memory (@R1, @R0, @DPTR)
Immediate Addressing	Program Memory
Base Register plus Register Addressing	Program Memory (@DPTR + A, @PC + A)

Register Addressing

Register addressing accesses the eight working registers (R0-R7) of the selected register bank. The least significant bits of the instruction opcode indicate which register is to be used. ACC, B, DPTR and CY, the Boolean processor accumulator can also be addressed as registers.

Direct Addressing

Direct addressing is the only method of accessing the special-function registers. The 128 bytes of internal RAM are also directly addressable.

Register Indirect Addressing

Register indirect addressing uses the contents of either R0 or R1 (in the selected register bank) as a pointer to locations in a 256-byte block: the 128 bytes of internal RAM or the lower 256 bytes of external data memory. Note that the special-function registers are not accessible by this method. Access to the full 64 Kbyte of external data memory address space is accomplished by using the 16-bit data pointer. Execution of PUSH and POP instructions also uses register indirect addressing. The stack may reside anywhere in the internal RAM.

Immediate Addressing

Immediate addressing allows constants to be part of the instruction in program memory.

Base Register plus Index Register Addressing

Base register plus index register addressing allows a byte to be accessed from program memory via an indirect move from the location whose address is the sum of a base register (DPTR or PC) and index register ACC. This mode facilitates look-up-table accesses.

Boolean Processor

The Boolean processor is a bit processor integrated within the SAB 8051 family of microcontrollers. It has its own instruction set, accumulator (the carry flag), and bit-addressable RAM and I/O.

- set bit
- clear bit
- complement bit
- jump if bit is set
- jump if bit is not set
- jump if bit is set and clear bit
- move bit from/to carry

Addressable bits, or their complements, may be logically ANDed or ORed with the contents of the carry flag. The result is returned to the carry register.

Introduction to the Instruction Set

The instruction set is divided into four functional groups:

- data transfer
- arithmetic
- logic
- control transfer

Data Transfer

Data operations are divided into three classes:

- general-purpose
- accumulator-specific
- address-object

None of these operations affects the PSW flag settings except a POP or MOV directly to the PSW.

General-Purpose Transfers

- MOV performs a bit or byte transfer from the source operand to the destination operand.
- PUSH increments the SP register and then transfers a byte from the source operand to the stack location currently addressed by SP.
  POP transfers a byte operand from the stack location addressed by the SP to the destination operand and then decrements SP.

Accumulator-Specific Transfers

- XCH exchanges the byte source operand with register A (accumulator)
- XCHD exchanges the low-order nibble of the source operand byte with the low-order nibble of A.
- MOVX performs a byte move between the external data memory and the accumulator. The external address can be specified by the DPTR register (16 bits) or the R1 or R0 register (8 bits).
- MOVC moves a byte from program memory to the accumulator. The operand in A is used as an index into a 256-byte table pointed to by the base register (DPTR or PC). The byte operand accessed is transferred to the accumulator.

Address-Object Transfer

- Mov DPTR, # data loads 16 bits of immediate data into a pair of destination registers, DPH and DPL.

Arithmetic

The SAB 8051 family of microcontrollers has four basic mathematical operations. Only 8-bit operations using unsigned arithmetic are supported directly. Refer to the technical description of the SAB 80C517 for 16-bit and 32-bit arithmetic operations. The overflow flag, however, permits the addition and subtraction operation to serve for both unsigned and signed binary integers. Arithmetic can also be performed directly on packed BCD representations.

Addition

- INC (increment) adds one to the source operand and puts the result in the operand.
- ADD adds A to the source operand and returns the result to A.
- ADDC (add with carry) adds A and the source operand, then adds one (1) if CY is set, and puts the result in A.
- DA (decimal-add-adjust for BCD addition) corrects the sum which results from the binary addition of two-digit decimal operands. The packed decimal sum formed by DA is returned to A. CY is set if the BCD result is greater than 99; otherwise it is cleared.

Subtraction

- SUBB (subtract with borrow) subtracts the second source operand from the first operand (the accumulator), subtracts one (1) if CY is set and returns the result to A.
- DEC (decrement) subtracts one (1) from the source operand and returns the result to the operand.

Multiplication

- MUL performs an unsigned multiplication of the A register, returning a double-byte result. A receives the low-order byte, B receives the high-order byte. OV is cleared if the top half of the result is zero and is set if it is not zero. CY is cleared. AC is unaffected.

Division

- DIV performs an unsigned division of the A register by the B register, and returns the integer quotient to A and returns the fractional remainder to the B register. Division by zero leaves indeterminate data in registers A and B and sets OV; otherwise OV is cleared. CY is cleared. AC is unaffected.

Flags

Unless otherwise stated in the above descriptions, the flags of PSW are affected as follows:

- CY is set if the operation causes a carry to or a borrow from the resulting high-order bit. Otherwise CY is cleared.
- AC is set if the operation results in a carry from the low-order four bits of the result (during addition), or a borrow from the high-order bits to the low-order bits (during subtraction); otherwise AC is cleared.
- OV is set if the operation results in a carry to the high-order bit of the result but not a carry from the bit, or vice versa; otherwise OV is cleared. OV is used in two's-complement arithmetic, because it is set when the signal result cannot be represented in 8 bits.
- P is set if the modulo 2 sum of the eight bits in the accumulator is 1 (odd parity); otherwise P is cleared (even parity). When a value is written to the PSW register, the P bit remains unchanged, as it always reflects the parity of A.

Logic

The SAB 8051 family of microcontrollers perform basic logic operations on both bit and byte operands.

Single-Operand Operations

- CLR sets A or any directly addressable bit to zero (0).
- SETB sets any directly bit-addressable bit to one (1).
- CPL is used to complement the contents of

the A register without affecting any flag, or any directly addressable bit location.

- RL, RLC, RR, RRC, SWAP are the five operations that can be performed on A. RL rotate left RR, rotate right, RLC rotate left through carry, RRC rotate right through carry, and SWAP rotate left four. For RLC and RRC the CY flag becomes equal to the last bit rotated out. SWAP rotates A left four places to exchange bits 3 through 0 with bits 7 through 4.

Two-Operand Operations

- ANL performs bitwise logical ANDing of two operands (for both bit and byte operands) and returns the result to the location of the first operand.
- ORL performs bitwise logical ORing of two source operands (for both bit and byte operands) and returns the result to the location of the first operand.
- XRL performs bitwise logical exclusive ORing of two source operands (byte operands) and returns the result to the location of the first operand.

Control Transfer

There are three classes of control transfer operations: unconditional calls, returns and jumps, conditional jumps, and interrupts. All control transfer operations cause, some upon a specific condition, the program execution to continue a non-sequential location in program memory.

Unconditional Calls, Returns and Jumps

Unconditional calls, returns and jumps transfer control from the current value of the program counter to the target address. Both direct and indirect transfers are supported.

- ACALL and LCALL push the address of the next instruction onto the stack and then transfer control to the target address. ACALL is a 2-byte instruction used when the target address is in the current 2K page. LCALL is a 3-byte instruction that addresses the full 64K program space. In ACALL, immediate data (i.e. an 11-bit address field) is concatenated to the five most significant bits of the PC (which is pointing to the next instruction). If ACALL is in the last 2 bytes of a 2K page then the call will be made to the next page since the PC will have been incremented to the next instruction prior to execution.
- RET transfers control to the return address saved on the stack by a previous call operation and decrements the SP register by two (2) to adjust the SP for the popped address.
- AJMP, LJMP and SJMP transfer control to the target operand. The operation of AJMP and LJMP are analogous to ACALL and LCALL. The SJMP (short jump) instruction provides for transfers within a 256-byte range centered about the starting address of the next instruction (-128 to + 127).
- JMP@A + DPTR performs a jump relative to the DPTR register. The operand in A is used as the offset (0-255) to the address in the DPTR register. Thus the effective destination for a jump can be anywhere in the program memory space.

Conditional Jumps

Conditional jumps perform a jump contingent upon a specific condition. The destination will be within a 256-byte range centered about the starting address of the next instruction (-128 to +127).

- JZ performs a jump if the accumulator is zero.
- JNZ performs a jump if the accumulator is not zero.
- JC performs a jump if the carry flag is set.
- JNC performs a jump if the carry flag is not set.
- JB performs a jump if the directly addressed bit is set.
- JNB performs a jump if the directly addressed bit is not set.
- JBC performs a jump if the directly addressed bit is set and then clears the directly addressed bit.
- CJNE compares the first operand to the

second operand and performs a jump if they are not equal. CY is set if the first operand is less than the second operand; otherwise it is cleared. Comparisons can be made between A and directly addressable bytes in internal data memory or an immediate value and either A, a register in the selected register bank, or a register indirect addressed byte of the internal RAM.

- DJNZ decrements the source operand and returns the result to the operand. A jump is performed if the result is not zero. The source operand of the DJNZ instruction may be any directly addressable byte in the internal data memory. Either direct or register addressing may be used to address the source operand.

Interrupt Returns

- RETI transfers control as RET does, but additionally enables interrupts of the current priority level.

Instruction Definitions

All 111 instructions of the SAB 8051 family of microcontrollers can essentially be condensed to 54 basic operations, in the following ordered alphabetically according to the operation mnemonic section.

A brief example of how the instruction might be used is given as well as its effect on the PSW flags. The number of bytes and machine cycles required, the binary machine language encoding, and a symbolic description or restatement of the function is also provided.

Note: Only the carry, auxiliary carry, and overflow flags are discussed. The parity bit is computed after every instruction cycle that alters the accumulator. Similarly, instructions which alter directly addressed registers could affect the other status flags if the instruction is applied to the PSW. Status flags can also be modified by bit manipulation.

7

Instruction	Flag			Instruction	Flag		
	CY	OV	AC		CY	OV	AC
ADD	X	X	X	SETB C	1		
ADDC	X	X	X	CLR C	0		
SUBB	X	X	X	CPL C	X		
MUL	0	X		ANL C, bit	X		
DIV	0	X		ANL C,/bit	X		
DA	X			ORL C, bit	X		
RRC	X			ORL C, /bit	X		
RLC	X			MOV C, bit	X		
CJNE	X						

Notes on Data Addressing Modes

Rn	Working register R0-R7
direct	128 internal RAM locations, any I/O port, control or status register
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data 16	16-bit constant included as bytes 2 and 3 of instruction
bit	128 software flags, any I/O pin, control or status bit
A	Accumulator

Notes on Program Addressing Modes

addr 16	Destination address for LCALL and LJMP may be anywhere within the 64-Kbyte program memory address space.
addr 11	Destination address for ACALL and AJMP will be within the same 2-Kbyte page of program memory as the first byte of the following instruction.
rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127/ -128 bytes relative to the first byte of the following instruction.

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the stack pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, op code bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2k block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345H. After executing the instruction,

ACALL SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Operation: ACALL
(PC)←(PC) + 2
(SP)←(SP) + 1
(SP)←(PC7-0)
(SP)←(SP) + 1
(SP)←(PC15-8)
(PC10-0)←page address

Encoding:

a10 a9 a8 1	0 0 0 1	a7 a6 a5 a4	a3 a2 a1 a0

Bytes: 2

Cycles: 2

ADD A, <src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the accumulator, leaving the result in the accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction,

ADD A,R0

will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A,Rn

Operation: ADD
(A)←(A) + (Rn)

Encoding:

0 0 1 0	1 r r r

Bytes: 1

Cycles: 1

ADD A,direct

Operation: ADD
(A)←(A) + (direct)

Encoding:

0 0 1 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ADD A,@Ri

Operation: ADDC
(A)←(A) + (C) + ((Ri))

Encoding:

0 0 1 0	0 1 1 i

Bytes: 1

Cycles: 1

ADD A,#data

Operation: ADD
(A)←(A) + #data

Encoding:

0 0 1 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ADDC A, <src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the accumulator contents, leaving the result in the accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 13, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

ADDC A,R0

will leave 6EH (01101110B) in the accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn

Operation: ADDC
(A)←(A) + (C) + (Rn)

Encoding:

0 0 1 1	1 r r r

Bytes: 1

Cycles: 1

ADDC A,direct

Operation: ADDC
(A)←(A) + (C) + (direct)

Encoding:

0 0 1 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ADDC A,@Ri

Operation: ADDC
(A)←(A) + (C) + ((Ri))

Encoding:

0 0 1 1	0 1 1 i

Bytes: 1

Cycles: 1

ADDC A,#data

Operation: ADDC
(A)←(A) + (C) + #data

Encoding:

0 0 1 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (after incrementing the PC twice), op code bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2k block of program memory as the first byte of the instruction following AJMP.

Example: The label "JMPADR" is at program memory location 0123H. The instruction,

AJMP JMPADR

is at location 0345H and will load the PC with 0123H.

Operation: AJMP
(PC)←(PC) + (2)
(PC10-0)←page address

Encoding:

a10 a9 a8 0	0 0 0 1	a7 a6 a5 a4	a3 a2 a1 a0

Bytes: 2

Cycles: 2

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

ANL A,R0

will leave 81H (10000001B) in the accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the accumulator at run-time. The instruction,

ANL P1,#01110011B

will clear bits 7,3, and 2 output port 1.

ANL A,Rn

Operation: ANL
(A)←(A) ∧ (Rn)

Encoding:

0 1 0 1	1 r r r

Bytes: 1

Cycles: 1

ANL A,direct

Operation: ANL
(A)←(A) ∧ (direct)

Encoding:

0 1 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ANL A,@Ri

Operation: ANL
(A)←(A) ∧ ((Ri))

Encoding:

0 1 0 1	0 1 1 i

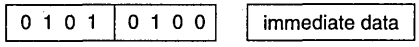
Bytes: 1

Cycles: 1

ANL A,#data

Operation: ANL
(A)←(A) ∧ #data

Encoding:



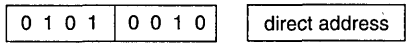
Bytes: 2

Cycles: 1

ANL direct,A

Operation: ANL
(direct)←(direct) ∧ (A)

Encoding:



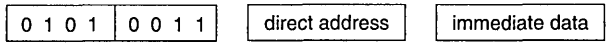
Bytes: 2

Cycles: 1

ANL direct,#data

Operation: ANL
(direct)←(direct) ∧ #data

Encoding:



Bytes: 3

Cycles: 2

ANL C, <src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC. 7 = 1, and OV = 0:

```
MOV   C,P1.0     ;Load carry with input pin state
ANL   C,ACC.7    ;AND carry with accumulator bit 7
ANL   C,/OV      ;AND with inverse of overflow flag
```

ANL C,bit

Operation: ANL
(C)←(C) ∧(bit)

Encoding:

1 0 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 2

ANL C,/bit

Operation: ANL
(C)←(C) ∧ ¬ (bit)

Encoding:

1 0 1 1	0 0 0 0	bit address

Bytes: 2

Cycles: 2

CJNE <dest-byte>, <src-byte>, rel

Function: Compare and Jump if Not Equal

Description: CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The accumulator contains 34H, Register 7 contains 56H. The first instruction in the sequence,

```
                CJNE    R7,#60H,NOT_EQ
;               ...     .....           ; R7 = 60H
NOT-EQ:         JC      REQ_LOW         ; IF R7 < 60H
;               ...     .....           ; R7 > 60H
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

```
WAIT:  CJNE     A,P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

CJNE A, direct,rel

Operation: (PC)←(PC) + 3
if (A) <> (direct)
then (PC)←(PC) + relative offset
if (A) < (direct)
then (C)←1
else (C)←0

Encoding:

1 0 1 1	0 1 0 1	direct address	rel. address

Bytes: 3

Cycles: 2

CJNE A,#data,rel

Operation: (PC)←(PC) + 3
if (A) <> data
then (PC)←(PC) + relative offset
if (A) < data
then (C)←1
else (C)←0

Encoding:

1 0 1 1	0 1 0 0	immediate data	rel. address

Bytes: 3

Cycles: 2

CJNE Rn,#data,rel

Operation: (PC)←(PC) + 3
if (Rn) <> data
then (PC)←(PC) + relative offset
if (Rn) < data
then (C)←1
else (C)←0

Encoding:

1 0 1 1	1 r r r	immediate data	rel. address

Bytes: 3

Cycles: 2

CJNE @Ri,#data,rel

Operation: (PC)←(PC) + 3
if ((Ri)) <> data
then (PC)←(PC) + relative offset
if ((Ri)) < data
then (C)←1
else (C)←0

Encoding:

1 0 1 1	0 1 1 i	immediate data	rel. address

Bytes: 3

Cycles: 2

CLR A

Function: Clear accumulator

Description: The accumulator is cleared (all bits set to zero). No flags are affected.

Example: The accumulator contains 5CH (01011100B). The instruction,

CLR A

will leave the accumulator set to 00H (00000000B).

Operation: CLR
(A)←0

Encoding:

1 1 1 0	0 1 0 0

Bytes: 1

Cycles: 1

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction,

CLR P1.2

will leave the port set to 59H (01011001B).

CLR C

Operation: CLR
(C)←0

Encoding:

1 1 0 0	0 0 1 1

Bytes: 1

Cycles: 1

CLR bit

Operation: CLR
(bit)←0

Encoding:

1 1 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 1

CPL A

Function: Complement accumulator

Description: Each bit of the accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to zero and vice versa. No flags are affected.

Example: The accumulator contains 5CH (01011100B). The instruction,

CPL A

will leave the accumulator set to 0A3H (10100011B).

Operation: CPL

(A)← ¬ (A)

Encoding:

1 1 1 1	0 1 0 0

Bytes: 1

Cycles: 1

CPL bit

Function: Complement bit

Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.

Note:

When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction sequence,

CPL P1.1

CPL P1.2

will leave the port set to 5BH (01011011B).

CPL C

Operation: CPL
(C)← ¬ (C)

Encoding:

1 0 1 1	0 0 1 1

Bytes: 1

Cycles: 1

CPL bit

Operation: CPL
(bit)← ¬ (bit)

Encoding:

1 0 1 1	0 0 1 0	bit address

Bytes: 2

Cycles: 1

DA A

Function: Decimal adjust accumulator for addition

Description: DA A adjusts the eight-bit value in the accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If accumulator bits 3-0 are greater than nine (xxxx1010-xxxx1111), or if the AC flag is one, six is added to the accumulator producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx-1111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the accumulator, depending on initial accumulator and PSW conditions.

Note:

DA A cannot simply convert a hexadecimal number in the accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence,

```
ADDC  A,R3
DA    A
```

will first perform a standard two's-complement binary addition, resulting in the value 0BEH (10111110) in the accumulator. The carry and auxiliary carry flags will be cleared.

The decimal adjust instruction will then alter the accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56,67, and the carry-in. The carry flag will be set by the decimal adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), then the instruction sequence,

```
ADD   A,#99H
DA    A
```

will leave the carry set and 29H in the accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30-1=29.

Operation: DA

contents of accumulator are BCD
if [[(A3-0) > 9] ∨ [(AC) = 1]]
then (A3-0) ← (A3-0) + 6
and
if [[(A7-4) > 9] ∨ [(C) = 1]]
then (A7-4) ← (A7-4) + 6

Encoding:

1 1 0 1	0 1 0 0

Bytes: 1

Cycles: 1

DEC byte

Function: Decrement

Description: The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence,

```
DEC   @R0
DEC   R0
DEC   @R0
```

will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

DEC A

Operation: DEC
(A) ← (A) - 1

Encoding:

0 0 0 1	0 1 0 0

Bytes: 1

Cycles: 1

DEC Rn

Operation: DEC
(Rn) ← (Rn) - 1

Encoding:

0 0 0 1	1 r r r

Bytes: 1

Cycles: 1

DEC direct

Operation: DEC
(direct) ← (direct) - 1

Encoding:

0 0 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

DEC @Ri

Operation: DEC
((Ri)) ← ((Ri)) - 1

Encoding:

0 0 0 1	0 1 1 i

Bytes: 1

Cycles: 1

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the accumulator by the unsigned eight-bit integer in register B. The accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: If B had originally contained 00H, the values returned in the accumulator and B register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 x 18) + 17. Carry and OV will both be cleared.

Operation: DIV
(A15-8)
(B7-0) ← (A)/(B)

Encoding:

1 0 0 0	0 1 0 0

Bytes: 1

Cycles: 4

DJNZ <byte>, <rel-addr>

Function: Decrement and Jump if not zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note:

When this instruction is used to modify an output port, the value used as the

original port data will be read from the output data latch, not the input pins.

Example: Internal RAM locations 40H, 50H, and 60H contain the values, 01H 70H, and 15H, respectively. The instruction sequence,

```
DJNZ   40H,LABEL_1
DJNZ   50H,LABEL_2
DJNZ   60H,LABEL_3
```

will cause a jump to the instruction at label LABEL_2 with values 00H, 6FH, and 15H in the three RAM locations. The first jump was not taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
        MOV     R2,#8
TOGGLE: CPL     P1.7
        DJNZ    R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output port 1. Each pulse will last three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Operation: DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) - 1
if (Rn) > 0 or (Rn) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	1 r r r	rel. address

Bytes: 2

Cycles: 2

DJNZ direct,rel

Operation: DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) - 1
if (direct) > 0 or (direct) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	0 1 0 1	direct address	rel. address

Bytes: 3

Cycles: 2

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Register 0 contains 7EH (01111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequences.

```
INC     @R0
INC     R0
INC     @R0
```

will leave register 0 set to 7 FH and internal RAM locations 7 EH and 7FH holding (respectively) 00H and 41H.

INC A

Operation: INC

(A) ← (A) + 1

Encoding:

0 0 0 0	0 1 0 0

Bytes: 1

Cycles: 1

INC Rn

Operation: INC

(Rn) ← (Rn) + 1

Encoding:

0 0 0 0	1 r r r

Bytes: 1

Cycles: 1

INC direct

Operation: INC

(direct) ← (direct) + 1

Encoding:

0 0 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

INC @Ri

Operation: INC
((Ri)) ← ((Ri)) + 1

Encoding:

0 0 0 0	0 1 1 i

Bytes: 1

Cycles: 1

INC DPTR

Function: Increment data pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed; an overflow of the low-order byte of the data pointer (DPL from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

```
INC     DPTR
INC     DPTR
INC     DPTR
```

will change DPH and DPL to 13H AND 01H.

Operation: INC

(DPTR) ← (DPTR) + 1

Encoding:

1 0 1 0	0 0 1 1

Bytes: 1

Cycles: 2

JB bit,rel

Function: Jump if bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010B. The accumulator holds 56 (01010110B). The instruction sequence,

```
JB      P1.2,LABEL1
JB      ACC.2,LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Operation: JB
(PC) ← (PC) + 3
if (bit) = 1
then (PC) ← (PC) + rel

Encoding:

0 0 1 0	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JBC bit,rel

Function: Jump if bit is set and clear bit

Description: If the indicated bit is one, branch to the address indicated; otherwise proceed with the next instruction. In either case, clear the designated bit. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note:

When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example: The accumulator holds 56H (01010110B). The instruction sequence,

```
JBC     ACC.3,LABEL1
JBC     ACC.2,LABEL2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the accumulator modified to 52H (01010010B).

Operation: JBC
(PC) ← (PC) + 3
if (bit) = 1
then (bit) ← 0
(then (PC) ← (PC)+ rel

Encoding:

0 0 0 1	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JC rel

Function: Jump if carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence,

```
JC      LABEL1
CPL     C
JC      LABEL2
```

will set the carry and cause program executive to continue at the instruction identified by the label LABEL2.

Operation: JC
(PC) ← (PC) + 2
if (C) = 1
then (PC) ← (PC) + rel

Encoding:

0 1 0 0	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

JMP @A+DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the accumulator nor the data pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
            MOV     DPTR, #JMP_TBL
            JMP     @A+DPTR
JMP_TBL:    AJMP    LABEL0
            AJMP    LABEL1
            AJMP    LABEL2
            AJMP    LABEL3
```

If the accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Operation: JMP
(PC) ← (A) + (DPTR)

Encoding:

0 1 1 1	0 0 1 1

Bytes: 1

Cycles: 2

JNB bit,rel

Function: Jump if bit not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is a computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

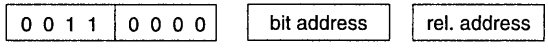
Example: The data present at input port 1 is 11001010B. The accumulator holds 56H (01010110B). The instruction sequence,

```
JNB     P1.3,LABEL1
JNB     ACC.3,LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Operation: JNB
(PC) ← (PC) + 3
if (bit) = 0
then (PC) ← (PC) + rel

Encoding:



Bytes: 3

Cycles: 2

JNC rel

Function: Jump if carry is not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

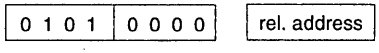
Example: The carry flag is set. The instruction sequence,

```
JNC     LABEL1
CPL     C
JNC     LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Operation: JNC
(PC) ← (PC) + 2
if (C) = 0
then (PC) ← (PC) + rel

Encoding:



Bytes: 2

Cycles: 2

JNZ rel

Function: Jump if accumulator not zero

Description: If any bit of the accumulator is one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally holds 00H. The instruction sequence,

```
JNZ   LABEL1
INC   A
JNZ   LABEL2
```

will set the accumulator to 01H and continue at label LABEL2.

Operation: JNZ
(PC) ← (PC) + 2
if (A) ≠ 0
then (PC) ← (PC) + rel

Encoding:

0 1 1 1	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

7

JZ rel

Function: Jump if accumulator zero

Description: If all bits of the accumulator are zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally contains 01H. The instruction sequence,

```
JZ    LABEL1
DEC   A
JZ    LABEL2
```

will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.

JZ
(PC) ← (PC) + 2
if (A) = 0
then (PC) ← (PC) + rel

Encoding:

0 1 1 0	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the stack pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64 kbyte program memory address space. No flags are affected.

Example: Initially the stack pointer equals 07H. The label "SUBRTN" is assigned to program memory location 1234H. After executing the instruction,

LCALL SUBRTN

at location 0123H, the stack pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1234H.

Operation: LCALL

(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC7-0)
(SP) ← (SP) + 1
((SP)) ← (PC15-8)
(PC) ← addr15-0

Encoding:

0 0 0 1	0 0 1 0	addr15...addr8	addr7...addr0

Bytes: 3

Cycles: 2

LJMP addr16

Function: Long jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64K program memory address space. No flags are affected.

Example: The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction,

LJMP JMPADR

at location 0123H will load the program counter with 1234H.

Operation: LJMP

(PC) ← addr15-0

Encoding:

0 0 0 0	0 0 1 0	addr15...addr8	addr7...addr0

Bytes: 3

Cycles: 2

MOV <dest-byte>,<src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH).

```
MOV   R0,#30H   ;R0 < = 30H
MOV   A,@R0     ;A < = 40H
MOV   R1,A      ;R1 < = 40H
MOV   R,@R1     ;B < = 10H
MOV   @R1,P1    ;RAM(40H) < = 0CAH
MOV   2,P1      ;P2 < 0CAH
```

leaves the value 30H in register 0, 40H in both the accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A,Rn

Operation: MOV
(A) ← (Rn)

Encoding:

1 1 1 0	1 r r r

Bytes: 1

Cycles: 1

MOV A,direct*

Operation: MOV
(A) ← (direct)

Encoding:

1 1 1 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

*MOV A,ACC is not a valid instruction.

MOV A,@Ri

Operation: MOV
(A) ← ((Ri))

Encoding:

1 1 1 0	0 1 1 i

Bytes: 1

Cycles: 1

MOV A,#data

Operation: MOV
(A) ← #data

Encoding:

0 1 1 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

MOV Rn,A

Operation: MOV
(Rn) ← (A)

Encoding:

1 1 1 1	1 r r r

Bytes: 1

Cycles: 1

MOV Rn,direct

Operation: MOV
(Rn) ← (direct)

Encoding:

1 0 1 0	1 r r r	direct address

Bytes: 2

Cycles: 2

MOV Rn,#data

Operation: MOV
(Rn) ← #data

Encoding:

0 1 1 1	1 r r r	immediate data

Bytes: 2

Cycles: 1

MOV direct,A

Operation: MOV
(direct) ← (A)

Encoding:

1 1 1 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

MOV direct,Rn

Operation: MOV
(direct) ← (Rn)

Encoding:

1 0 0 0	1 r r r	direct address

Bytes: 2

Cycles: 2

MOV direct,direct

Operation: MOV
(direct) ← (direct)

Encoding:

1 0 0 0	0 1 0 1	dir. addr. (src)	dir. addr. (dest)

Bytes: 3

Cycles: 2

MOV direct,@Ri

Operation: MOV
(direct) ← ((Ri))

Encoding:

1 0 0 0	0 1 1 i	direct address

Bytes: 2

Cycles: 2

MOV direct,#data

Operation: MOV
(direct) ← # data

Encoding:

0 1 1 1	0 1 0 1	direct address	immediate data

Bytes: 3

Cycles: 2

MOV @Ri,A

Operation: MOV
((Ri)) ← (A)

Encoding:

1 1 1 1	0 1 1 i

Bytes: 1

Cycles: 1

MOV @Ri,direct

Operation: MOV
((Ri)) ← (direct)

Encoding:

1 0 1 0	0 1 1 i	direct addr.

Bytes: 2

Cycles: 2

MOV @Ri,#data

Operation: MOV
((Ri)) ← #data

Encoding:

0 1 1 1	0 1 1 i	immediate data

Bytes: 2

Cycles: 1

MOV <dest-bit>, <src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input port 3 is 11000101B. The data previously written to output port 1 is 35H (00110101B).

```
MOV   P1.3,C
MOV   C,P3.3
MOV   P1.2,C
```

will leave the carry cleared and change port 1 to 39H (00111001B).

MOV C,bit

Operation: MOV
(C) ← (bit)

Encoding:

1 0 1 0	0 0 1 0	bit address

Byte: 2

Cycles: 1

MOV bit,C

Operation: MOV
(bit) ← (C)

Encoding:

1 0 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 2

MOV DPTR, #data16

Function: Load data pointer with a 16-bit constant

Description: The data pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,
MOV DPTR,#1234H

will load the value 1234H into the data pointer: DPH will hold 12H and DPL will hold 34H.

Operation: MOV
(DPTR) ← #data15-0
DPH □ DPL ← #data15-8 □ #data7-0

Encoding:

1 0 0 1	0 0 0 0	immed. data 15-8	immed. data 7-0

Bytes: 3

Cycles: 2

MOVC A,@A + <base-reg>

Function: Move code byte

Description: The MOVC instructions load the accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit accumulator contents and the contents of a 16-bit base register, which may be either the data pointer or the PC. In the latter case, the PC

is incremented to the address of the following instruction before being added to the accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the accumulator. The following instructions will translate the value in the accumulator to one of four values defined by the DB (define byte) directive.

```
REL_PC:   INC     A
          MOVC    A,@A + PC
          RET
          DB      66H
          DB      77H
          DB      88H
          DB      99H
```

If the subroutine is called with the accumulator equal to 01H, it will return with 77H in the accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the accumulator instead.

MOVC A,@A + DPTR

Operation: MOVC
(A) ← ((A) + (DPTR))

Encoding:

1 0 0 1	0 0 1 1

Bytes: 1

Cycles: 2

MOVC A,@A + PC

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

Encoding:

1 0 0 0	0 0 1 1

Bytes: 1

Cycles: 2

MOVX <dest-byte>, <src-byte>

Function: Move external

Description: The MOVX instructions transfer data between the accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or 16-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight -bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, the data pointer generates a 16-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 special function register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64 kbytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the data pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines (e.g., an SAB 8155 RAM/I/O/Timer) is connected to the SAB 80515 port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H AND 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

```
MOVX  A,@R1
MOVX  @R0,A
```

copies the value 56H into both the accumulator and external RAM location 12H.

MOVX A,@Ri

Operation: MOVX
(A) ← ((Ri))

Encoding:

1 1 1 0	0 0 1 i

Bytes: 1

Cycles: 2

MOVX A,@DPTR

Operation: MOVX
(A) ← ((DPTR))

Encoding:

1 1 1 0	0 0 0 0

Bytes: 1

Cycles: 2

MOVX @Ri,A

Operation: MOVX
((Ri)) ← (A)

Encoding:

1 1 1 1	0 0 1 i

Bytes: 1

Cycles: 2

MOVX @DPTR,A

Operation: MOVX
((DPTR))← (A)

Encoding:

1 1 1 1	0 0 0 0

Bytes: 1

Cycles: 2

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the accumulator and register B. The low-order byte of the 16-bit product is left in the accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,

MUL AB

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the accumulator is cleared. The overflow flag is set, carry is cleared.

Operation: MUL
(A7-0) ← (A) X (B)
(B15-8)

Encoding:

1 0 1 0	0 1 0 0

Bytes: 1

Cycles: 4

NOP

Function: No operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of port 2 lasting exactly

5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the instruction sequence,

```
CLR    P2.7
NOP
NOP
NOP
NOP
SETB   P2.7
```

Operation: NOP

Encoding:

0 0 0 0	0 0 0 0

Bytes: 1

Cycles: 1

ORL <dest-byte> <src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

```
ORL    A,R0
```

will leave the accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the accumulator at run-time. The instruction,

```
ORL    P1,#00110010B
```

will set bits 5, 4, and 1 of output port 1.

ORL A,Rn

Operation: ORL
(A) ← (A) ∨ (Rn)

Encoding:

0 1 0 0	1 r r r

Bytes: 1

Cycles: 1

ORL A,direct

Operation: ORL
(A) ← (A) ∨ (direct)

Encoding:

0 1 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ORL A,@Ri

Operation: ORL
(A) ← (A) ∨ ((Ri))

Encoding:

0 1 0 0	0 1 1 i

Bytes: 1

Cycles: 1

ORL A,#data

Operation: ORL
(A) ← (A) ∨ #data

Encoding:

0 1 0 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ORL direct,A

Operation: ORL
(direct) ← (direct) ∨ (A)

Encoding:

0 1 0 0	0 0 1 0	direct address

Bytes: 2

Cycles: 1

ORL direct,#data

Operation: ORL

(direct) ← (direct) ∨ #data

Encoding:

0 1 0 0	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

ORL C,<src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC.7 = 1, or OV = 0:

```
MOV   C,P1.0    ;Load carry with input pin p10
ORL   C,ACC.7   ;OR carry with the accumulator bit 7
ORL   C,/OV     ;OR carry with the inverse of OV.
```

ORL C,bit

Operation: ORL
(C) ← (C) ∨ (bit)

Encoding:

0 1 1 1	0 0 1 0	bit address

Bytes: 2

Cycles: 2

ORL C,/bit

Operation: ORL
(C) ← (C) ∨ ¬ (bit)

Encoding:

1 0 1 0	0 0 0 0	bit address

Bytes: 2

Cycles: 2

POP direct

Function: Pop from stack

Description: The contents of the internal RAM location addressed by the stack pointer is read, and the stack pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The stack pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP   DPH
POP   DPL
```

will leave the stack pointer equal to the value 30H and the data pointer set to 0123H. At this point the instruction,

```
POP   SP
```

will leave the stack pointer set to 20H. Note that in this special case the stack pointer was decremented to 2FH before being loaded with the value popped (20H).

Operation: POP
(direct) ← ((SP))
(SP) ← (SP) - 1

Encoding:

1 1 0 1	0 0 0 0	direct address

Bytes: 2

Cycles: 2

PUSH direct

Function: Push onto stack

Description: The stack pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the stack pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine the stack pointer contains 09H. The data pointer holds the value 0123H. The instruction sequence,

```
PUSH   DPL
PUSH   DPH
```

will leave the stack pointer set to 0BH and store 23H AND 01H in internal RAM locations 0AH and 0BH, respectively.

Operation: Push
(SP) ← (SP) + 1
((SP)) ← (direct)

Encoding:

1 1 0 0	0 0 0 0	direct address

Bytes: 2

Cycles: 2

RET

Function: Return from subroutine

Description: RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the stack pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The stack pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

```
RET
```

will leave the stack pointer equal to the value 09H. Program execution will continue at location 0123H.

Operation: RET
(PC15-8) ← ((SP))
(SP) ← (SP) - 1
(PC7-0) ← ((SP))
(SP) ← (SP) - 1

Encoding:

0 0 1 0	0 0 1 0

Bytes: 1

Cycles: 2

RETI

Function: Return from interrupt

Description: RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The stack pointer is left decremented by two. No other registers are affected; the PSW is not automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is

generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt had been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example: The stack pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RETI

will leave the stack pointer equal to 09H and return program execution to location 0123H.

Operation: RETI
(PC15-8) ← ((SP))
(SP) ← (SP) - 1
(PC7-0) ← ((SP))
(SP) ← (SP) - 1

Encoding:

0 0 1 1	0 0 1 0

Bytes: 1

Cycles: 2

RL A

Function: Rotate accumulator left

Description: The eight bits in the accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

RL A

leaves the accumulator holding the value 8BH (10001011B) with the carry unaffected.

Operation: RL
(An + 1) ← (An) n = 0 - 6
(A0) ← (A7)

Encoding:

0 0 1 0	0 0 1 1

Bytes: 1

Cycles: 1

RLC A

Function: Rotate accumulator Left through carry flag

Description: The eight bits in the accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The accumulator holds the value 0C5H (11000101B, and the carry is zero. The instruction,

RLC A

leaves the accumulator holding the value 8AH (10001010B) with the carry set.

Operation: RLC
(An + 1) ← (An) n = 0 - 6
(A0) ← (C)
(C ← (A7)

Encoding:

0 0 1 1	0 0 1 1

Bytes: 1

Cycles: 1

RR A

Function: Rotate accumulator right

Description: The eight bits in the accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

RR A

leaves the accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Operation: RR
(An)← (An + 1) n = 0 - 6
(A7) ← (A0)

Encoding:

0 0 0 0	0 0 1 1

Bytes: 1

Cycles: 1

RRC A

Function: Rotate accumulator right through carry flag

Description: The eight bits in the accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The accumulator holds the value 0C5H (11000101B), the carry is zero. The instruction,

RRC A

leaves the accumulator holding the value 62 (01100010B) with the carry set.

Operation: RRC
(An) ← (An + 1) n = 0 - 6
(A7) ← (C)
(C) ← (A0)

Encoding:

0 0 0 1	0 0 1 1

Bytes: 1

Cycles: 1

SETB <bit>

Function: Set bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output port 1 has been written with the value 34H (00110100B).

The instructions,

```
SETB   C
SETB   P1.0
```

will leave the carry flag set to 1 and change the data output on port 1 to 35H (00110101B).

SETB C

Operation: SETB
(C) ← 1

Encoding:

1 1 0 1	0 0 1 1

Bytes: 1

Cycles: 1

SETB bit

Operation: SETB
(bit) ← 1

Encoding:

1 1 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 1

SJMP rel

Function: Short jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction,

SJMP RELADR

will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H.

Note:

Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

Encoding:

1 0 0 0	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

SUBB A,<src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the accumulator, leaving the result in the accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set before executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction.

SUBB A,Rn

Operation: SUBB
(A) ← (A) - (C) - (Rn)

Encoding:

1 0 0 1	1 r r r

Bytes: 1

Cycles: 1

SUBB A,direct

Operation: SUBB
(A) ← (A) - (C) - (direct)

Encoding:

1 0 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

SUBB A,@Ri

Operation: SUBB
(A) ← (A) - (C) - ((Ri))

Encoding:

1 0 0 1	0 1 1 i

Bytes: 1

Cycles: 1

SUBB A,#data

Operation: SUBB
(A) ← (A) - (C) - #data

Encoding:

1 0 0 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

SWAP A

Function: Swap nibbles within the accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

Example: The accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the accumulator holding the value 5CH (01011100B).

Operation: SWAP
(A3-0) ↔ (A7-4)

Encoding:

1 1 0 0	0 1 0 0

Bytes: 1

Cycles: 1

XCH A,<byte>

Function: Exchange accumulator with byte variable

Description: XCH loads the accumulator with the contents of the indicated variable, at the same time writing the original accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20H. The accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCH A,@R0

will leave RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the accumulator.

XCH A,Rn

Operation: XCH
(A) ⇄ (Rn)

Encoding:

1 1 0 0	1 r r r

Bytes: 1

Cycles: 1

XCH A,direct

Operation: XCH
(A) ◄► (direct)

Encoding:

1 1 0 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

XCH A,@Ri

Operation: XCH
(A) ◄► ((Ri))

Encoding:

1 1 0 0	0 1 1 i

Bytes: 1

Cycles: 1

XCHD A,@Ri

Function: Exchange digit

Description: XCHD exchanges the low-order nibble of the accumulator (bits 3-0, generally representing a hexadecimal or BCD digit), with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCHD A,@R0

will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the accumulator.

Operation: XCHD
(A3-0) ◄► ((RI)3-0)

Encoding:

1 1 0 1	0 1 1 i

Bytes: 1

Cycles: 1

XRL <dest-byte>, <src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When the instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

XRL A,R0

will leave the accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the accumulator at run-time. The instruction,

XRL P1,#00110001B

will complement bits 5, 4, and 0 of output port 1.

7

XRL A,Rn

Operation: XRL

(A) ← (A) ⊻ (Rn)

Encoding:

0 1 1 0	1 r r r

Bytes: 1

Cycles: 1

XRL A,direct

Operation: XRL

(A) ← (A) ⊻ (direct)

Encoding:

0 1 1 0	0 1 0 1	direct address

XRL A,@Ri

Operation: XRL
(A) ← (A) ⊻ ((Ri))

Encoding:

0 1 1 0	0 1 1 i

Bytes: 1

Cycles: 1

XRL A,#data

Operation: XRL
(A) ← (A) ⊻ #data

Encoding:

0 1 1 0	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

XRL direct,A

Operation: XRL
(direct) ← (direct) ⊻ (A)

Encoding:

0 1 1 0	0 0 1 0	direct address

Bytes: 2

Cycles: 1

XRL direct,#data

Operation: XRL
(direct) ← (direct) ⊻ #data

Encoding:

0 1 1 0	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

Mnemonic		Description	Byte	Cycle
Arithmetic Operations				
ADD	A,Rn	Add Register to Accumulator	1	1
ADD	A,Direct	Add Direct Byte to Accumulator	2	1
ADD	A,@Ri	Add Indirect RAM to Accumulator	1	1
ADD	A,#Data	Add Immediate Data to Accumulator	2	1
ADDC	A,Rn	Add Register to Accumulator with Carry Flag	1	1
ADDC	A,Direct	Add Direct Byte to A with Carry Flag	2	1
ADDC	A,@Ri	Add Indirect RAM to A with Carry Flag	1	1
ADDC	A,#Data	Add Immediate Data to A with Carry Flag	2	1
SUBB	A,Rn	Subtract Register from A with Borrow	1	1
SUBB	A, Direct	Subtract Direct Byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract Indirect RAM from A with Borrow	1	1
SUBB	A,#Data	Subtract Immediate Data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment Register	1	1
INC	Direct	Increment Direct Byte	2	1
INC	@Ri	Increment Indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement Register	1	1
DEC	Direct	Decrement Direct Byte	2	1
DEC	@Ri	Decrement Indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

Mnemonic		Description	Byte	Cycle
Logical Operations				
ANL	A,Rn	AND Register to Accumulator	1	1

Mnemonic		Description	Byte	Cycle
Logical Operations				
ANL	A,Direct	AND Direct Byte to Accumulator	2	1
ANL	A,@Ri	AND Indirect RAM to Accumulator	1	1
ANL	A,#data	AND Immediate Data to Accumulator	2	1
ANL	Direct,A	AND Accumulator to Direct Byte	2	1
ANL	Direct,#Data	AND Immediate Data to Direct Byte	3	2
ORL	A,Rn	OR Register to Accumulator	1	1
ORL	A,Direct	OR Direct Byte to Accumulator	2	1
ORL	A,@Ri	OR Indirect RAM to Accumulator	1	1
ORL	A,#Data	OR Immediate Data to Accumulator	2	1
ORL	Direct,A	OR Accumulator to Direct Byte	2	1
ORL	Direct,#Data	OR Immediate Data to Direct Byte	3	2
XRL	A,Rn	Exclusive OR Register to Accumulator	1	1
XRL	A,Direct	Exclusive OR Direct Byte to Accumulator	2	1
XRL	A,@Ri	Exclusive OR Indirect RAM to Accumulator	1	1
XRL	A,#Data	Exclusive OR Immediate Data to Accumulator	2	1
XRL	Direct,A	Exclusive OR Accumulator to Direct Byte	2	1
XRL	Direct,#Data	Exclusive OR Immediate Data to Direct Byte	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through Carry Flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry Flag	1	1
SWAP	A	Swap Nibbles within the Accumulator	1	1

Mnemonic		Description	Byte	Cycle
Data Transfer				
MOV	A,RN	Move Register to Accumulator	1	1

Mnemonic		Description	Byte	Cycle
Data Transfer				
MOV	A,Direct*	Move Direct Byte to Accumulator	2	1
MOV	A,@Ri	Move Indirect RAM to Accumulator	1	1
MOV	A,#Data	Move Immediate Data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to Register	1	1
MOV	Rn, Direct	Move Direct Byte to Register	2	2
MOV	Rn,#Data	Move Immediate Data to Register	2	1
MOV	Direct,A	Move Accumulator to Direct Byte	2	1
MOV	Direct,Rn	Move Register to Direct Byte	2	2
MOV	Direct,Direct	Move Direct Byte to Direct Byte	3	2
MOV	Direct,@Ri	Move Indirect RAM to Direct Byte	2	2
MOV	Direct,#Data	Move Immediate Data to Direct Byte	3	2
MOV	@Ri,A	Move Accumulator to Indirect RAM	1	1
MOV	@Ri,Direct	Move Direct Byte to Indirect RAM	2	2
MOV	@Ri,#Data	Move Immediate Data to Indirect RAM	2	1
MOV	DPTR,#Data16	Load Data Pointer with a 16-Bit Constant	3	2
MOVC	A,@A + DPTR	Move Code Byte Relative to DPTR to Accumulator	1	2
MOVC	A,@A + PC	Move Code Byte Relative to PC to Accumulator	1	2
MOVX	A,@Ri	Move External RAM (8-Bit Addr.) to Accumulator	1	2
MOVX	A,@DPTR	Move External RAM (16-bit Addr.) to Accumulator	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit Addr.)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-Bit Addr.)	1	2
PUSH	Direct	Push Direct Byte onto Stack	2	2
POP	Direct	Pop Direct Byte from Stack	2	2
XCH	A,Rn	Exchange Register with Accumulator	1	1
XCH	A,Direct	Exchange Direct Byte with Accumulator	2	1
XCH	A,@Ri	Exchange Indirect RAM with Accumulator	1	1

Mnemonic		Description	Byte	Cycle
Data Transfer				
XCHD	A,@Ri	Exchange Low-Order Digit Indirect RAM with A	1	1

Mnemonic		Description	Byte	Cycle
Boolean Variable Manipulation				
CLR	C	Clear Carry Flag	1	1
CLR	Bit	Clear Direct Bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	Bit	Set Direct Bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	Bit	Complement Direct Bit	2	1
ANL	C,Bit	AND Direct Bit to Carry Flag	2	2
ANL	C,/Bit	AND Complement of Direct Bit to Carry	2	2
ORL	C,Bit	OR Direct Bit to Carry Flag	2	2
ORL	C,/Bit	OR Complement of Direct Bit to Carry	2	2
MOV	C,Bit	Move Direct Bit to Carry Flag	2	1
MOV	Bit,C	Move Carry Flag to Direct Bit	2	2
ACALL	Addr 11	Absolute Subroutine Call	2	2
LCALL	Addr 16	Long Subroutine Call	3	2
RET		Return from Subroutine	1	2
RETI		Return from Interrupt	1	2
AJMP	Addr 11	Absolute Jump	2	2
LJMP	Addr 16	Long Jump	3	2
SJMP	Rel	Short Jump (Relative Addr.)	2	2
JMP	@A + DPTR	Jump Indirect Relative to the DPTR	1	2
JZ	Rel	Jump if Accumulator is Zero	2	2
JNZ	Rel	Jump if Accumulator is Not Zero	2	2
JC	Rel	Jump if Carry Flag is Set	2	2
JNC	Rel	Jump if Carry Flag is Not Set	2	2
JB	Bit,Rel	Jump if Direct Bit is Set	3	2
JNB	Bit,Rel	Jump if Direct Bit is Not Set	3	2

Mnemonic		Description	Byte	Cycle
Boolean Variable Manipulation				
JBC	Bit, Rel	Jump if Direct Bit is Set and Clear Bit	3	2
CJNE	A,Direct,Rel	Compare Direct Byte to A and Jump if Not Equal	3	2
CJNE	A,#Data,Rel	Comp. Immed. to A and Jump if Not Equal	3	2
CJNE	Rn,#Data,Rel	Comp. Immed. to Reg. and Jump if Not Equal	3	2
CJNE	@Ri,#Data,Rel	Comp. Immed. to Ind. and Jump if Not Equal	3	2
DJNZ	Rn,Rel	Decrement Register and Jump if Not Zero	2	2
DJNZ	Direct,Rel	Decrement Direct and Jump if Not Zero	3	2
NOP		No Operation	1	1

Notes on Data Addressing Modes:

Rn —Working register R0-R7

Direct —128 internal RAM locations, any I/O port, control or status register

@Ri —Indirect internal or external RAM location addressed by register R0 or R1

#Data —8-bit constant included in instruction

#Data16 —8-bit constant included as bytes 2 and 3 of instruction

Bit —128 software flags, any I/O pin, control or status bit

A —Accumulator

Notes on Program Addressing Modes:

Addr 16 —Destination address for LCALL and LJMP may be anywhere within the 64 kbyte program memory address space.

Addr 11 —Destination address for ACALL and AJMP will be within the same 2 kbyte page of program memory as the first byte of the following instruction.

Rel —SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127/-128 bytes relative to first byte of the following instruction.

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
00	1	NOP	
01	2	AJMP	Code Addr
02	3	LJMP	Code Addr
03	1	RR	A

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
04	1	INC	A
05	2	INC	Data Addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	Bit Addr, Code Addr
11	2	ACALL	Code Addr
12	3	LCALL	Code Addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	Data Addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	Bit Addr, Code Addr
21	2	AJMP	Code Addr

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
22	1	RET	
23	1	RL	A
24	2	ADD	A, # Data
25	2	ADD	A, Data Addr
26	1	ADD	A, @R0
27	1	ADD	A, @R1
28	1	ADD	A, R0
29	1	ADD	A,R1
2A	1	ADD	A, R2
2B	1	ADD	A, R3
2C	1	ADD	A, R4
2D	1	ADD	A, R5
2E	1	ADD	A, R6
2F	1	ADD	A, R7
30	3	JNB	Bit Addr, Code Addr
31	2	ACALL	Code Addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A, #Data
35	2	ADDC	A, Data Addr
36	1	ADDC	A, @R0
37	1	ADDC	A, @R1
38	1	ADDC	A, R0
39	1	ADDC	A, R1
3A	1	ADDC	A, R2
3B	1	ADDC	A, R3
3C	1	ADDC	A, R4
3D	1	ADDC	A, R5
3E	1	ADDC	A, R6
3F	1	ADDC	A, R7

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
40	2	JC	Code Addr
41	2	AJMP	Code Addr
42	2	ORL	Data Addr, A
43	3	ORL	Data Addr, # Data
44	2	ORL	A, # Data
45	2	ORL	A, Data Addr
46	1	ORL	A, @ R0
47	1	ORL	A, @ R1
48	1	ORL	A, R0
49	1	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E	1	ORL	A, R6
4F	1	ORL	A, R7
50	2	JNC	Code Addr
51	2	ACALL	Code Addr
52	2	ANL	Data Addr, A
53	3	ANL	Data Addr, # Data
54	2	ANL	A, # Data
55	2	ANL	A, Data Addr
56	1	ANL	A, @ R0
57	1	ANL	A, @ R1
58	1	ANL	A, R0
59	1	ANL	A, R1
5A	1	ANL	A, R2
5B	1	ANL	A, R3
5C	1	ANL	A, R4
5D	1	ANL	A, R5

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
5E	1	ANL	A, R6
5F	1	ANL	A, R7
60	2	JZ	Code Addr
61	2	AJMP	Code Addr
62	2	XRL	Data Addr, A
63	3	XRL	Data Addr, #Data
64	2	XRL	A, #Data
652	2	XRL	A, Data Addr
66	1	XRL	A, @ R0
67	1	XRL	A, @ R1
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E	1	XRL	A, R6
6F	1	XRL	A, R7
70	2	JNZ	Code Addr
71	2	ACALL	Code Addr
72	2	ORL	C, Bit Addr
73	1	JMP	@A + DPTR
74	2	MOV	A, #Data
75	3	MOV	Data Addr, #Data
76	2	MOV	@R0, #Data
77	2	MOV	@R1, #Data
78	2	MOV	R0, #Data
79	2	MOV	R1, #Data
7A	2	MOV	R2, #Data
7B	2	MOV	R3, #Data

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
7C	2	MOV	R4, #Data
7D	2	MOV	R5, #Data
7E	2	MOV	R6, #Data
7F	2	MOV	R7, #Data
80	2	SJMP	Code Addr
81	2	AJMP	Code Addr
82	2	ANL	C, Bit Addr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	Data Addr, Data Addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
8A	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C	2	MOV	data addr, R4
8D	2	MOV	data addr, R5
8E	2	MOV	data addr, R6
8F	2	MOV	data addr, R7
90	3	MOV	DPTR, #Data
91	2	ACALL	Code Addr
92	2	MOV	Bit Addr, C
93	1	MOVC	A, @A + DPTR
94	2	SUBB	A, #Data
95	2	SUBB	A, Data Addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, R0
99	1	SUBB	A, R1

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, /Bit Addr
A1	2	AJMP	Code Addr
A2	2	MOV	C, Bit Addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		Reserved	
A6	2	MOV	@R0, Data Addr
A7	2	MOV	@R1, Data Addr
A8	2	MOV	R0, Data Addr
A9	2	MOV	R1, Data Addr
AA	2	MOV	R2, Data Addr
AB	2	MOV	R3, Data Addr
AC	2	MOV	R4, Data Addr
AD	2	MOV	R5, Data Addr
AE	2	MOV	R6, Data Addr
AF	2	MOV	R7, Data Addr
B0	2	ANL	C, /Bit Addr
B1	2	ACALL	Code Addr
B2	2	CPL	Bit Addr
B3	1	CPL	C
B4	3	CJNE	A, #Data, Code Addr
B5	3	CJNE	A, Data Addr, Code Addr
B6	3	CJNE	@R0, #Data, Code Addr

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
B7	3	CJNE	@R1, #Data, Code Addr
B8	3	CJNE	R0, #Data, Code Addr
B9	3	CJNE	R1, #Data, Code Addr
BA	3	CJNE	R2, #Data, Code Addr
BB	3	CJNE	R3, #Data, Code Addr
BC	3	CJNE	R4, #Data, Code Addr
BD	3	CJNE	R5, #Data, Code Addr
BE	3	CJNE	R6, #Data, Code Addr
BF	3	CJNE	R7, #Data, Code Addr
C0	2	PUSH	Data Addr
C1	2	AJMP	Code Addr
C2	2	CLR	Bit Addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A, Data Addr
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
CF	1	XCH	A, R7
D0	2	POP	Data Addr
D1	2	ACALL	Code Addr
D2	2	SETB	Bit Addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	Data Addr, Code Addr
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1
D8	2	DJNZ	R0, Code Addr
D9	2	DJNZ	R1, Code Addr
DA	2	DJNZ	R2, Code Addr
DB	2	DJNZ	R3, Code Addr
DC	2	DJNZ	R4, Code Addr
DD	2	DJNZ	R5, Code Addr
DE	2	DJNZ	R6, Code Addr
DF	2	DJNZ	R7, Code Addr
E0	1	MOVX	A@DPTR
E1	2	AJMP	Code Addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A@R1
E4	1	CLR	A
E5	2	MOV	A,Data Addr*
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4

Instruction Op Codes in Hexadecimal Order			
Hex Code	**Number of Bytes**	**Mnemonic**	**Operands**
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	Code Addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	Data Addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

*MOV A,ACC is not a valid instruction

Microcontroller Application Notes
Article Reprints/Application Briefs

8

Applications Using Operation of Timer 2 in the SAB 80515/80535–Generating Pulse Width Modulated Signals

SAB 80515/80535

8

Application Note

Introduction

This application introduces the user to the features of the Timer 2 on the SAB 80515/ 80535. It gives basic application hints for programming and using the functions of the Timer 2 structure. An example on generating pulse-width-modulated signals with minimum software is also given.

Timer 2 Structure

The Timer 2 of the SAB 80515 with its 16-bit compare/reload/capture register and three 16-bit compare/capture registers is capable of generating pulse width modulated output signals with very little software effort. This unit is also referred to as Programmable Timer/Counter Register Array (PTRA).

The SAB 80515 offers two different compare modes, explained in more detail below, which are useful for the generation of these output signals.

In either compare mode, the values stored in the selected compare registers are compared continuously to the count value of Timer 2. For this purpose, each compare register has its own comparator circuit. Within one machine cycle, all comparisons are done simultaneously.

Each of the four registers has a fixed relation to a port 1 pin, illustrated in the following table:

Register	Port Latch	Alternate Function Symbols
CRC Register	P1.0	$\overline{\text{INT3}}$/CC0
CC1 Register	P1.1	INT4/CC1
CC2 Register	P1.2	INT5/CC2
CC3 Register	P1.3	INT6/CC3

The alternate functions for which port pins P1.0 to P1.3 can be used are shown in the following table:

Port Pin		Alternate Function
P1.0	INT3	External Interrupt 3 input; can be selected to be active on a negative or positive transition.
	CC0	Compare output from or capture input to CRC register.
P1.1	INT4	External Interrupt 4 input; active on a positive transition.
	CC1	Compare output from or capture input to CC1 register.
P1.2	INT5	External Interrupt 5 input; active on a positive transition.
	CC2	Compare output from or capture input to CC2 register.
P1.3	INT6	External Interrupt 6 input; active on a positive transition.
	CC3	Compare output from or capture input to CC3 register.

The selected modes of the compare/capture registers determine which of the possible alternate functions is actually used. For brevity, the description of the use of alternate functions is given for the CRC register and port pin P1.0 only.

If no compare or capture modes are enabled for the CRC register, pin P1.0 can be used for general purpose I/O or as an external interrupt input. In the latter case, the port latch must contain a one (1) to allow the external source to control the pin. It is, however, possible to cause an interrupt by toggling the port latch by software. Bit I3FR in SFR T2CON is used to select the active edge for $\overline{\text{INT3}}$ (refer to the SAB 80515 User's Manual for details). If the external capture mode is enabled for the CRC register (SFR CCEN = XXXXXX01B), a transition at P1.0 will not only cause the request flag IEX3

(SFR IRCON) to be set, but will also latch the current Timer 2 contents into the CRC register. If the pin is not controlled by an external source, software may toggle the port latch to do a capture of the Timer 2 content. This facilitates reading the timer "on-the-fly", since reading the timer directly requires a certain read procedure to be observed in order to obtain the correct value (to read the 16-bit register, only 8-bit MOV instructions are available). This feature is in addition to the special capture mode provided in the SAB 80515 for a software capture.

If either of the two possible compare modes is enabled for the CRC register, the port pin P1.0 is used as an output. The functions of the two compare modes are described separately in the following sections.

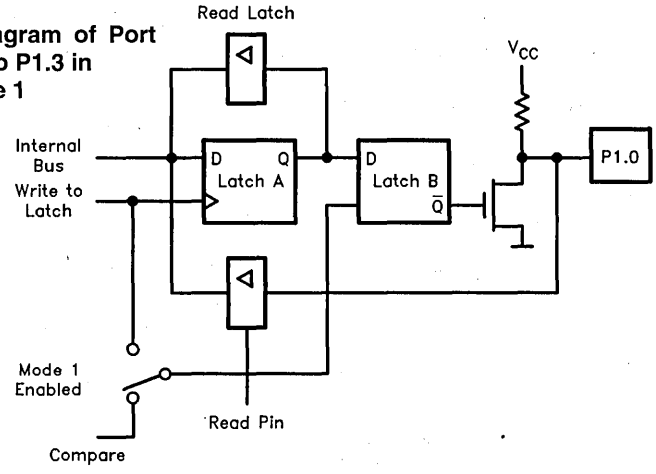
In compare mode (T2CM = 0), the port latch is controlled only by the Timer 2 overflow and the match signal of the comparator related to the CRC register. The user has no access to the port latch as long as compare mode 0 is enabled (SFR CCEN = XXXXXX10B). The input line from pin P1.0 to the interrupt request flag IEX3 is disconnected and IEX3 is controlled by the output of the comparator. If Timer 2 overflows, the port latch is set to zero. If a match is detected between the value stored in the CRC register and the content of Timer 2, then the port latch is set to one. The setting of the request flag IEX3 depends on the selected active edge. If I3FR = 0, IEX3 is set with the positive transition of the comparator output signal. If I3FR = 0, IEX3 will be set when the comparator output goes inactive, i.e. shows a negative transition.

In compare mode 1 (T2CM = 1), the port latch P1.0 is separated into two latches: one is connected to the internal bus and can be read or written to under software control. This is referred to as latch A. The other latch, called latch B, is a transparent latch with the input connected to latch A. Its output connected to the port pin and its clock input controlled by the comparator output. Figure 1 Functional Diagram of Port Latches P1.0 to P1.3 in Compare Mode 1 illustrates these two port latches. The function of compare mode 1 is illustrated by the following example:

Assume that port P1.0 contains a one, register CRC contains a value xxxx, IEX3 is programmed to be positive transition active, and Timer 2 is running. When compare mode 1 is enabled (SFR CCEN = XXXXXX10B), the port latch is separated, and both latch A and latch B contain a one. The user may now write a zero into latch A. Latch B is unchanged, since the clock input is low. When a match is detected between the value xxxx stored in the CRC register and the contents of Timer 2, the comparator output signal goes active. This causes the value of latch A to be transferred to latch B and pin P1.0 is set to zero. Simultaneously, the interrupt request flag IEX3 is set, informing the user of the successful match.

User response depends on the application. Writing a one to latch A causes a positive transition at the next compare event within the following Timer 2 period. Changing both the port latch and the compare value in register CRC causes a new match either within the current Timer 2 period if the new value is higher than the actual Timer 2 content, or within the next Timer 2 period otherwise. If the user changes neither the port latch nor the CRC register, the port pin will remain in its state regardless of further compare events. However, each further compare event (one during each Timer 2 period, if compare mode 1 is enabled) will set the interrupt request flag IEX3. This enables the user to count the compare events until a certain number is reached. Servicing the port latch at that point will cause a new transition of the pin at the next match.

Figure 1
Functional Diagram of Port Latches P1.0 to P1.3 in Compare Mode 1



NOTE:

Since the transition detection circuit is installed directly in front of the interrupt request flag, IEX3 may be set when a compare mode is enabled or disabled. For an explanation of this, consider the following conditions:

- IEX3 is programmed to be set on a negative transition (I3FR = 0)
- Port latch P1.0 contains a one, so a logic high level is applied to the input of the detection circuit at IEX3
- The CRC register contains 0000H and Timer 2 contains a value > 0000H.

  If a compare mode is now enabled for the CRC register, the input line to the transition detector is switched from the port pin to the output of the comparator, which is zero (no match). The transition detector recognizes a negative transition at its input and causes IEX3 to be set. The same procedure applies to the other condition: I3FR = 1/P1.0 = 0/ comparator output = 1.

For registers CC1 to CC3 and their appropriate port pins, the compare and capture features function in a similar way, except that the setting of the interrupt flags and the capture are caused only by a positive transition at the respective port pins.

Using Compare Mode 0

The following section shows a concrete example for generating pulse width modulated output signals using the compare mode 0. The CRC register is used to do a 16-bit reload of Timer 2 in order to vary the period of the signals, while registers CC1 to CC3 are selected to operate in compare mode 0. The appendix contains a listing of the example program.

First an initialization routine is executed. It selects the input frequency for Timer 2, the compare and reload mode, loads the reload value into the CRC register, and starts the timer. Then the first loading of the compare registers CC1 to CC3 is done in the Timer 2 overflow interrupt service routine. The main program, which is not detailed in this example, computes the compare values and stores them in registers R2 to R7, from which the values are loaded into registers CC1 to CC3 during the interrupt procedure. If the compare values were left unchanged over several Timer 2 periods, the following output signals would result.

Figure 2
Pulse Width Modulation in Compare Mode 0

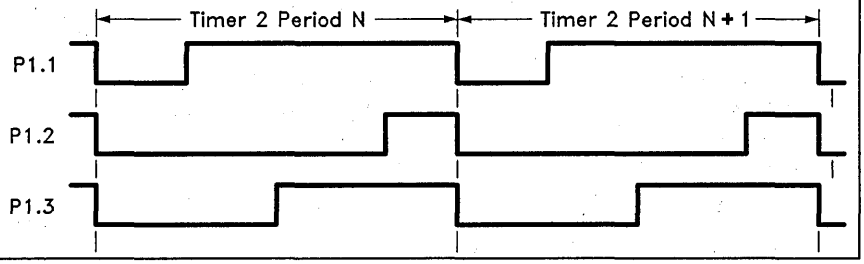
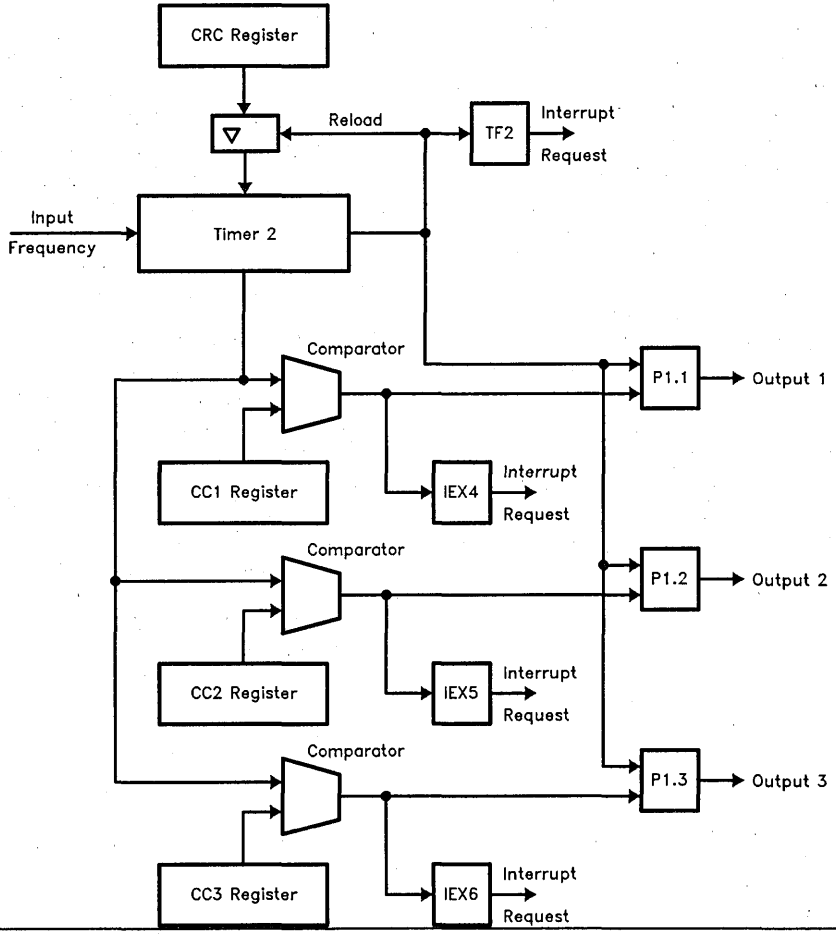


Figure 3
PTRA Configuration for the Generation of PWM Signals



8

Figure 3 PTRA Configuration for the Generation of PWM Signals illustrates the configuration of the PTRA structure after the initialization. The CRC register serves as a 16-bit reload register, while registers CC1 to CC3 are selected as 16-bit compare registers. The comparator output signal of each compare register is connected to the appropriate interrupt request flag (IEX4 to IEX6). Since the port pin P1.0 is not used in this application, it can be used as general purpose I/O or as an external interrupt input. It cannot, however, be used as a capture input to the CRC register, since selecting the reload mode for the CRC register disables the capture and compare modes for this register.

In this example, all three output signals have the same period as determined by the reload value stored in the CRC register. On an overflow of Timer 2, the port latches P1.1 to P1.3 are reset and the value of the CRC register is loaded into Timer 2. The timer continues incrementing from this start value up to the next overflow. The time period results from the formula

TP = (12/fosc) * (65536 - <reload value>)

where <reload value> is the content of the CRC register. The frequency of the output signals is 1/TP. The assembler ASM51 provides an easy way to compute the reload value for a given time period. If the time period is given in s (at 12 MHz in this example), the reload value can be programmed in the following way:

```
TIME_PERIOD EQU 1234;1234 µs time
period at 12MHz

MOV       CRCL,#LOW(-TIME_PERIOD)
MOV       CRCH,#HIGH(-TIME_PERIOD)
```

Note the minus sign, which states that the value stored is 65536 - TIME_PERIOD.

The time during which an output signal is high is determined by the value stored in the associated compare register. The same mathematical relation applies here as for the reload value:

HT = (12/fosc) * (65536 - <compare value>)

where <compare value> is the contents of the respective compare register CC1 to CC3. Note that if a compare value is smaller than the reload value, the associated output pin will remain low, since there will never be a match.

If the compare values are not changed by the main program from one Timer 2 period to the next, the compare registers need not be reloaded in the Timer 2 interrupt service routine. A software flag called READY is used to indicate whether or not the compare registers have to be reloaded. If the main program has computed new compare values, it sets flag READY after storing these values in registers R2 to R7. The first instruction in the Timer 2 interrupt service routine checks the READY flag and returns to the main program if READY = 0. Otherwise the compare registers are loaded with the contents of R2 to R7. This procedure shortens the time needed for servicing the Timer 2 interrupt if the compare values are not changed.

The READY flag also has a second function. The instruction sequence which loads registers R2 to R7 can be interrupted by the Timer 2 interrupt. To avoid loading indeterminate values into the compare registers, the reload procedure in the Timer 2 interrupt service routine must not be executed until all new compare values are stored in R2 to R7. Since the READY flag will be reset in this case, the routine which loads the compare registers will be skipped in the interrupt service routine.

Before loading the compare registers with new values in the Timer 2 interrupt service routine, all compare modes are disabled (CCEN = 00000000B). This is done for a reason similar to that described above. When the 16-bit compare registers are reloaded with 8-bit MOV instructions, indeterminate compare values may occur in the registers. These values may cause

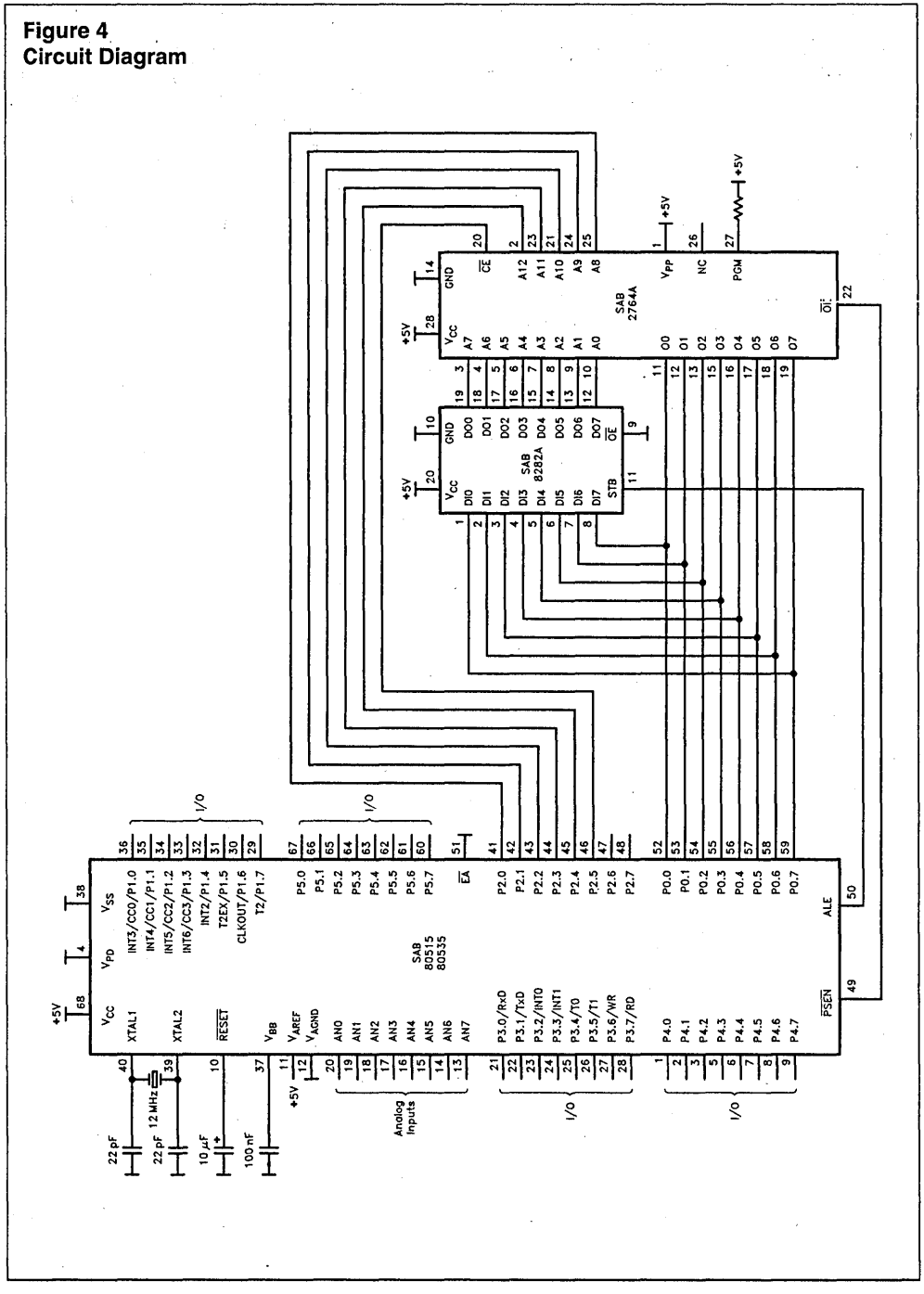
unexpected match conditions. To avoid this, first compare is disabled. The registers are then loaded with new values, and finally the compare is again enabled. At the end the READY flag is reset.

Because vectoring to the Timer 2 interrupt service routine, disabling compare mode, and reloading the compare registers all take time, there is a limit to the smallest low pulse at the output pins. Vectoring to the interrupt routine in a single interrupt routine takes no more than 8 and no fewer than 3 cycles. Checking the READY flag, disabling compare, reloading the 3 16-bit registers, and enabling compare again will take 18 cycles. Therefore, the smallest low pulse should not be less than 0026D (26 cycles), otherwise a match is not recognized.

Figure 4 Circuit Diagram shows a detailed circuit diagram of an SAB 80535 with an external program memory represented by a 8 Kbyte SAB 2764A EPROM. This diagram also shows the basic power supply connections when using the SAB 80535.

8

Figure 4
Circuit Diagram



```
MCS-51  MACRO    ASSEMBLER   T2COMP

ISIS-II MCS-51 MACRO ASSMEBLER V2.1
OBJECT MODULE PLACED IN :F1:T2COMP.OBJ
ASSEMBLER INVOKED BY:  ASM51 :F1:T2COMP.SRC NO SYMBOLS DATE(04.03.86)
LOC   OBJ      LINE   SOURCE
                  1   $NOMOD51
                  2   $INCLUDE (REG515.PDF)
      = 1         3   +1$NOLIST
                160
      1234      161   TIME PERIOD     EQU   1234H    ; TIME PERIOD COMPARE SIGNALS
      0000      162   READY           BIT   0        ; READY FLAG DEFINITION
      0021      163   COMPARE_1_LOW   DATA  21H      ; TEMPORARY STORAGE LOCATIONS
      0022      164   COMPARE_1_HIGH  DATA  22H      ;  FOR COMPARE VALUES
      0026      165   COMPARE_3_HIGH  DATA  26H
                166
                167   ;*****************************************  *
                168   ;*
                169   ;*      INITIALIZATION ROUTINE
                170   ;*
                171   ;*****************************************  *
                172
                173   INIT:
0000  75CACC    174   MOV  CRCL,#LOW(-TIME_PERIOD);; LOAD RELOAD VALUE
0003  75CBED    175   MOV  CRCH,#HIGH(-TIME_PERIOD);;
0006  75CB10    176   MOV  T2CON,#00010000B;        ; RELOAD ON  TIMER OVERFLOW
                177                                 ; NO PRESCALER; COMPARE MODE 0
0009  C200      178   CLR  READY                     ; CLEAR FLAG READY
000B  D2C8      179   SETBT2I0                       ; START TIMER 2, FOSC/12
                180
000D  020045    181   JMP  MAIN_PROG                 ; JUMP TO MAIN PROGRAM
                182
                183
MCS-51  MACRO    ASSEMBLER   T2COMP

LOC   OBJ      LINE   SOURCE
                184   ;*********************************************  *
                185   ;*
                186   ;* TIMER 2 OVERFLOW INTERRUPT SERVICE ROUTINE.
                187   ;* LOADS NEW COMPARE VALUES INTO THE COMPARE REGISTERS.
                188   ;*
                189   ;*********************************************  *
                190
----------      191      CSEG   AT TIMER2
                192
                193   TF2_INT:
002B  300014    194      JNB   READY,INT_END    ; CHECK FLAG READY
002E  75C100    195      MOV   CCEN,$0          ; DISABLE COMPARE
0031  8AC2      196      MOV   CCL1,R2          ; LOAD NEW COMPARE VALUES
0033  8BC3      197      MOV   CCH1,R3          ;
0035  8CC3      198      MOV   CCL2,R4          ;
0037  8DC3      199      MOV   CCH2,R5          ;
0039  8EC6      200      MOV   CCL3,R6          ;
003B  8FC7      201      MOV   CCH3,R7          ;
003D  75C1A8    202      MOV   CCEN,#10101000B  ; ENABLE COMPARE 1 TO 3
0040  C200      203      CLR   READY            ; CLEAR FLAG READY
                204   INT_END:
0042  C2C6      205      CLR   TF2              ; CLEAR TIMER 2 OVERFLOW FLAG
0044  32        206      RETI
                207
```

8

```
                208   ;*******************************************************  *
                209   ;*
                210   ;* MAIN PROGRAM.
                211   ;* COMPUTES THE NEW COMPARE VALUES AND STORES THEM INTO
                212   ;* REGISTERS R2 TO T7 IN CURRENT REGISTER BANK.
                213   ;*
                214   ;* SINCE IT DEPENDS ON THE APPLICATION, THE MAIN PROGRAM
                215   ;* IS NOT DETAILED HERE.
                216   ;*
                217   ;*******************************************************  *
MCS-51 MACRO ASSEMBLER   T2COMP (Continued)
LOC   OBJ      LINE   SOURCE
                218
                219   MAIN_PROG:
                220   ;             *
                221   ;             *
                222   ;             *
0045  C200      223                 CLR READY               ; CLEAR FLAG READY
0047  AA21      224                 MOV R2,COMPARE_1_LOW    ; STORE NEW COMPARE VALUES
0049  AB22      225                 MOV R3,COMPARE_1_HIGH   ;
                226   ;             *
                227   ;             *
                228   ;             *
004B  AF26      229                 MOV R7,COMPARE_3_HIGH   ;
004D  D200      230                 SETBREADY               ; SET FLAG READY
                231   ;             *
                232   ;             *
                233   ;             *
                234   ;             *
                235                 END
REGISTER BANK(S) USED: 0
ASSEMBLY COMPLETE, NO ERRORS FOUND
```

Operation of the A/D Converter in the SAB 80515/80535

SAB 80515/80535

8

Application Note

Introduction

This application note discusses the Analog to Digital converter in the SAB 80515/80535 single chip microcontroller. It provides the SAB 80515/80535 user with detailed information on the features and characteristics of the converter. Included in this application note are design tips for using the converter, as well as an application example with program listing which illustrates how an 8-bit digital result obtained as a result of conversion of an analog input can be displayed on a terminal over the serial channel of the SAB 80515/80535.

The A/D Converter of the SAB 80515/80535

The SAB 80515/80535 is an 8-bit single chip microcontroller, which contains an on-chip A/D converter. It provides a simple interface between analog and digital circuitry. It permits the replacement of discrete A/D components with this on-board circuitry.

This application note illustrates the use of the SAB 80515/80535's A/D converter. The electrical characteristics and operating conditions are discussed in more detail than in the User's Manual.

The following topics are covered:

- Fundamentals and principles of A/D conversion with the SAB 80515/80535
- Transfer characteristics and error definition
- Electrical characteristics of the SAB 80515/80535's A/D converter
- Design considerations of the device
- Sample application of the device

Principles of A/D Conversion with the SAB 80515/80535

An Analog to Digital converter converts analog input signals (voltages) applied at the "analog inputs" into the corresponding digital value. The range of the analog inputs which result in digital values between 00H and FFH in the device's 8-bit converter is defined by the reference voltages (reference ground VAGND and reference voltage VAREF). These voltages are applied externally to the device.

Different principles may be used in A/D conversion. They differ mainly in accuracy, resolution, conversion time, costs, etc. The A/D converter of the SAB 80515/80535 uses the principle of successive-approximation. This technique is much faster than the more common dual slope conversion and allows tracking of signals at higher frequencies. Successive-approximation method uses binary fractions (1/2, 1/4, 1/8, etc.). The unknown input value is compared first to the 1/2 of the reference value, determining the most significant bit of the result. Depending on this MSB, the unknown value is compared with 1/4 or (1/4 + 1/2) ref. value thus determining the next bit of the result. Following this procedure eight times produces an eight-bit result. The conversion time is independent of the input value.

To ensure meaningful data, the input value to the comparator cannot change during conversion. To ensure this, the SAB 80515/80535 samples the input and latches this value at the comparator input during the conversion. This avoids fluctuations in the analog input to the comparator. Suppression of noise in the analog input value as is normal with dual slope converters is not required, since there is no input signal integration.

For the above reasons, the reference voltage must be held at a constant level during the whole conversion time. Unlike the analog inputs, there is no sample-and-hold circuit at the reference input. To avoid unpredictable results, the user must provide a constant

noise-free reference voltage to the device.

The successive approximation converter does not use an R-2R network because the variations in resistance values affect the accuracy of the A/D converter. Instead of the R-2R approach, a network of 256 capacitors is used. These capacitors are organized in binary weighted groups. Switching these groups generates the binary weighted reference values. Using this principle improves the converter accuracy and also the immunity against temperature and frequency changes.

Note that this technique affects the electrical characteristics at the analog input pins (see next section for further information).

A special feature of the SAB 80515/80535's A/D converter is the ability to program internal reference voltages. This feature allows reference ground (IVAGND) and reference voltage (IVAREF) to be defined as fractions of externally applied references. Device software makes it possible to select for each among 16 values in equal steps, one for IVAGND and one for IVAREF. It is also possible to program IVAGND = VAGND and IVAREF = VAREF.

Figure 5
A/D Converter Block Diagram

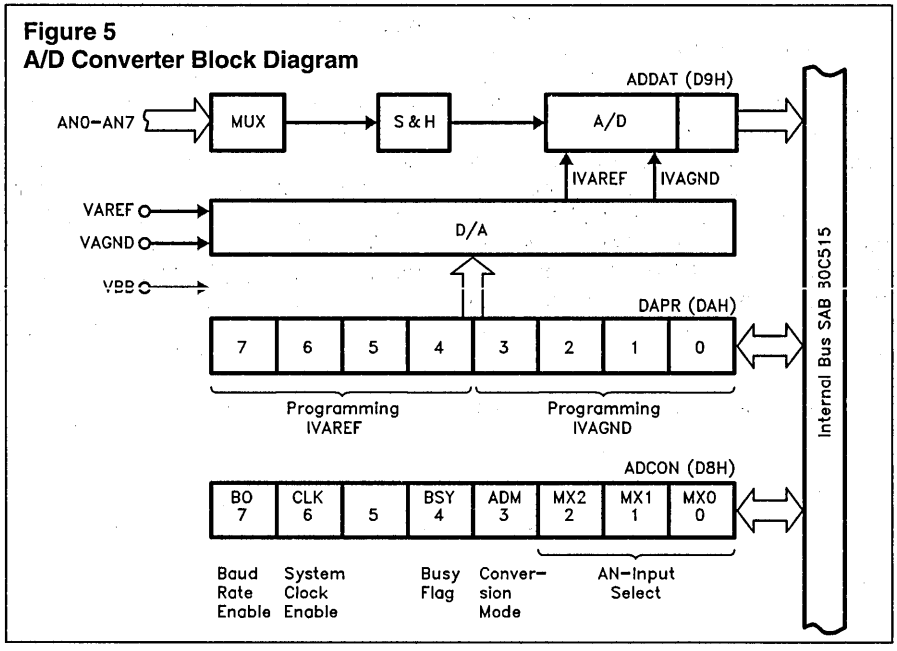


Figure 5 A/D Converter Block Diagram shows the block diagram of the SAB 80515/80535's A/D converter. It shows the following main features:

- Eight analog inputs, selectable via a multiplexer
- Sample & hold function for analog inputs
- External references available to the converter after internal division
- Special Function Registers ADCON, ADDAT, DAPR, used for converter control, status and results

The Special Function Registers (SFR) are described in detail in the SAB 80515 User's Manual. Some points to consider when programming the SFRs include:

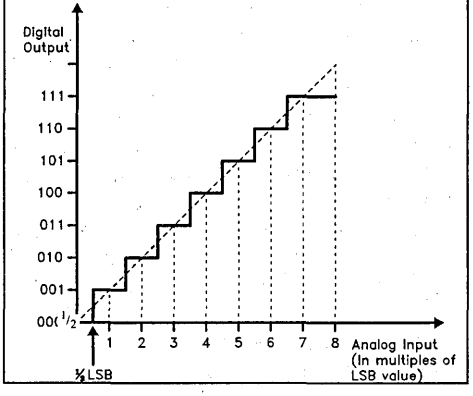
- The A/D conversion is initiated by writing the reference selection value into the SFR DAPR (addr.: DAH). Writing a "00H" starts the conversion using the externally applied references (VAGND and VAREF) directly.
  Writing into DAPR during a conversion causes an interrupt to the conversion followed by a restart.
- Altering the value of SFR ADCON (addr.: D8H) during a conversion causes unpredictable results.
- Once a conversion is complete, the SFR ADDAT (addr.: D9H) contains the result of the conversion. If the A/D converter isn't used, the SFR ADDAT is available for general purpose read/write storage.
- The conversion time is 15 machine cycles. Using a 12 MHz oscillator, gives a conversion time of 15 µs. This includes the sample time.

Transfer Characteristics and Error Definition

Figure 6 Ideal Transfer Characteristic shows the ideal transfer characteristic for a 3-bit A/D converter.

Since possible analog input values are a continuum, they must be quantized by partitioning the continuum into 2^n discrete digital values. The number n equals the number of bits in the converter. All analog values within a given range are represented by the same digital value, which corresponds to the nominal mid-range value. Each converter has inherent quantization uncertainty of 1/2 LSB (Least Significant Bit). The ideal transfer characteristic shown in Figure 6 Ideal Transfer Characteristic gives the first digital transition (from "00H" to "01H") at the analog value of 1/2 LSB.

Figure 6
Ideal Transfer Characteristic



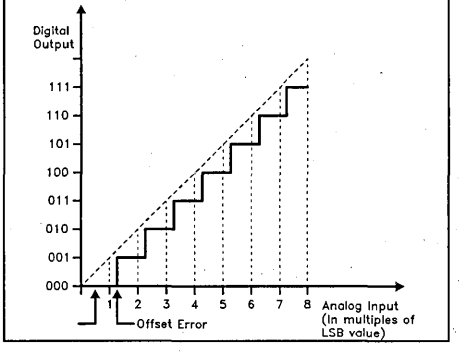
Based on the ideal transfer characteristic, four different error types are defined:

- Offset error
- Integral non-linearity
- Differential non-linearity
- Gain error

Each of these is described in detail below.

The offset error (Figure 7 Offset Error) is the mean adjustment in input voltage required to bring the digital output to the first digital transition (from "00H" to "01H") of the converter. The deviation from the ideal value of this transition (at 1/2 LSB) is the offset of the converter.

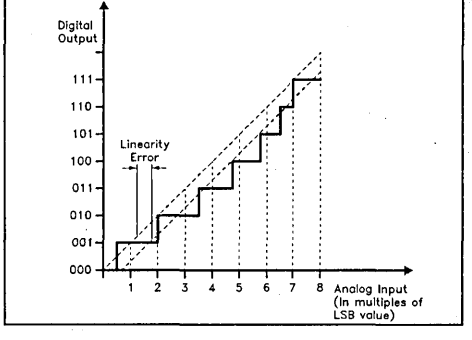
Figure 7
Offset Error



This error may vary over the full temperature range. An adjustment is therefore exact only for a fixed temperature.

The integral nonlinearity is also known as the linearity error. This is defined as the maximum deviation of the actual transfer function from the ideal straight line at any point along the function. It can be expressed as a percentage of full scale or, as shown in Figure 8 Integral Nonlinearity, in multiples of the LSB value. The value of integral nonlinearity assumes that other errors, such as offset and gain, have been adjusted to zero. Linearity error cannot be adjusted and is an inherent characteristic of the converter.

Figure 8
Integral Nonlinearity

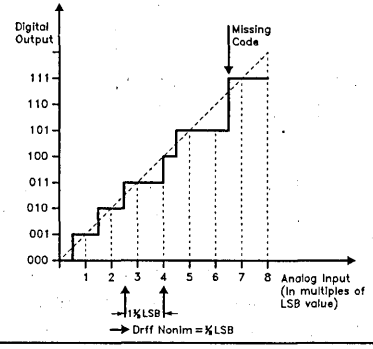


8

The differential nonlinearity (Figure 9 Differential Nonlinearity) is the maximum deviation of any quantum from its ideal analog input value between any adjacent pairs of digital numbers, over the full range of the digital output. If each transition is exactly 1 LSB, the differential nonlinearity is zero. If the transitions are 1 LSB ±1 LSB, then there is the possibility of missing code, i.e. digital value misses, e.g. the output might jump from 011 to 101, missing out 100.

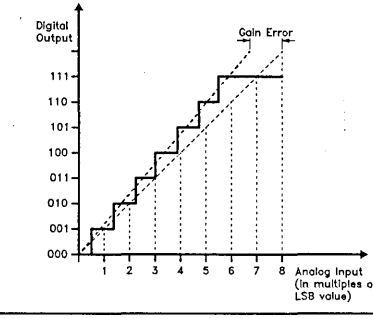
Conversely, differential nonlinearity of less than 1 LSB, automatically excludes missing codes.

Figure 9
Differential Nonlinearity



The gain error (Figure 10 Gain Error) is also known as the scale factor error. It is the difference in slope between the real and the ideal transfer characteristic. It may be expressed in LSB's or as a percentage of analog magnitude.

Figure 10
Gain Error



This error can be adjusted by changing the reference voltage or adjusting the input voltage division. The temperature drift of the gain error is generally smaller than that of the offset error.

All these errors are specified for the SAB 80515/80535 in the A/D characteristics section of the data sheet. The LSB unit refers to an 8-bit resolution. Therefore, 1 LSB is (VAREF - VAGND)/256, giving 20 mV for a reference voltage of 5.12V.

The specified errors are valid over the total operating temperature range specified for the device (0 °C to +70 °C for standard parts, -40 °C to +85 °C for extended temperature range parts).

Using the internal programming ability for the reference voltages IVAGND and IVAREF does not affect the errors and the accuracy of the A/D converter itself. The absolute values of these errors will be the same over the full reference range, as well as in a smaller internally programmed reference range. The resolution is increased by using narrower references. As an example, if the internally programmed references are IVAGND = 2.5V and IVAREF = 3.75V, the resolution is ≈5 mV. This allows a more exact measurement of differences between several successive analog voltages; the differences can be determined in steps of 5 mV, rather than steps of 20 mV (full reference range). This feature is useful for the measurement of analog differences in closed control loops. Once an initial value is established using the full range, a second conversion with changed internal reference voltages gains higher resolution.

Electrical Characteristics of the SAB 80515/80535's A/D Converter

The electrical operating conditions and maximum ratings on the pins of the SAB 80515/80535's A/D converter are given in the datasheet.

The Analog Input Pins AN0-AN7

Maximum Ratings

For the analog inputs the same maximum ratings are valid as for all other pins. These maximum ratings are specified in the SAB 80515/80535's datasheet. Thus, the maximum voltage at the pins is restricted to -0.5V up to +7V with respect to ground (V_{SS}). The device may never be exposed to voltages above these values, otherwise the chip might be damaged.

Normal Operating Characteristics

As mentioned above, the A/D converter of the SAB 80515/80535 uses a capacitor network instead of a R-2R network. This affects the input impedance of the analog inputs.

The input impedance of the analog inputs is mainly capacitive with a negligible resistive component. This capacitance is relevant at pins AN0-AN7 only during the load time, which is part of the conversion time. At other times, the inputs have very high impedance typical leakage currents of nA's, as shown in Figure 11 Analog Interface, an RC Network.

The load time is the period which is available to charge the inputs before they are sampled during the remainder of the sample time when the comparator is adjusted. The rest of the conversion time is taken up by the actual conversion process.

The input impedance requires that the analog source from which the analog value is generated must be capable of fully charging all the analog input pins AN0-7 to the value to be measured within the load time. In the worst case the capacitance must be charged to the full input voltage from the completely discharged state.

The input capacitance CI of the converter and the internal resistance of the analog source form a RC lowpass filter, which has a charging function shown in Figure 12 Ri, CI Change Time.

Where:

t: Time

V(t): Voltage after time t

Ri: Internal resistance of the analog source

CI: Input capacitance of the A/D converter. This formula determines the change over a given time period.

t	V/Vmax [%]
1 RC	63%
2 RC	87%
5 RC	99.3%
8 RC	99.97%
10 RC	99.995%

The maximum value of Ri can be calculated for a charging time of at least 10 Ri X CI (error less than 0.005%) as:

$$T_L \geq 10 \times (Ri \times CI)$$

$$RI \leq \frac{T_L}{10 \times CI}$$

Assuming the worst-case conditions of:

Load time (FOSC = 12 MHz): 2 µs
Max. Input Capacity: 50 pF (typ. 25 pF)

We Get:

$$RI \leq \frac{2\mu S}{10 \times 50pf} = 4k\Omega$$

8

Figure 11
Analog Interface, an RC Network

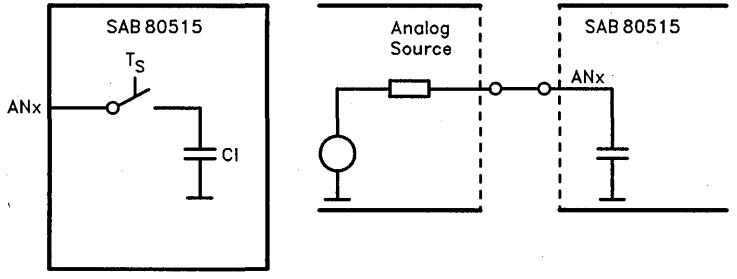
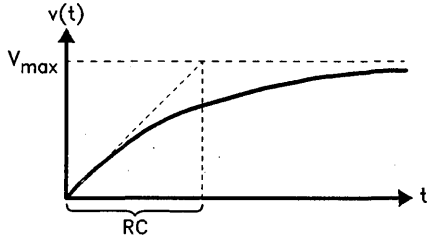


Figure 12
Ri, CI Change Time



V(t) = Vmax(1 - e-t/Ri - CT)

Ri should therefore be less than 4 Ω (see also datasheet SAB 80515/80535).

These conditions are reached only under worst-case conditions.

The Reference Voltage Pins (VAGND and VAREF)

Absolute Maximum Ratings

As with other pins, the maximum voltage range at these pins is -0.5V to +7V, as given in the datasheet.

Normal Operating Characteristics

The normal operating conditions for VAGND and VAREF are also specified in the SAB 80515/80535 datasheet, under A/D characteristics. The operating conditions under which conversion is valid are:

VAGND = V_{SS} 0.2V

VAREF = V_{CC} 5%

In other words, externally applied reference voltages must not deviate from the digital power supply voltage for more than these values. Although adjustment of the external reference is not possible internal programming of the references can be done under software control.

Exceeding the values specified will not damage the part if the maximum ratings are not exceeded. However, the specified accuracy is no longer guaranteed.

The allowed operating voltages of the analog inputs AN0-AN7 are different from the maximum ratings. They are defined by the voltages at VAGND and

VAREF:

VAGND - 0.2V ≤ VAINPUT ≤VAREF + 0.2V

An exact conversion requires that the reference voltages be held constant during the whole conversion time. The normal digital power supply for V_{CC} is therefore appropriate for VAREF because of noise and glitches on this voltage. It is recommended

that the digital power supply and the analog reference supply be separate. The VAREF current is 5 mA max. and the differential internal resistance of the reference supply must be less than 1 k Ω. This is a result of similar charging processes to those at the analog inputs and the reference power supply must take this into account.

The V_{BB} Pin

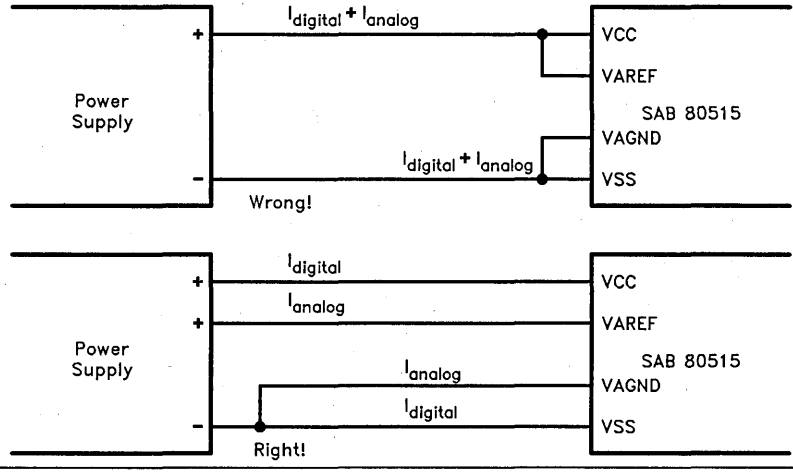
This pin is connected to the substrate of the chip. A back-bias generator generates a negative voltage (with respect to ground) in the substrate. This generator has noise too. Noise from this could affect the A/D converter operation. It must therefore be eliminated by a capacitor between V_{BB} and V_{SS} (ground). The capacitor should be between 47 nF and 100 nF (not 500 pF-1000 pF, as described in an older release of the datasheet).

Design Considerations

Reference Voltage Supply

A design with the SAB 80515/80535's A/D converter must follow rules similar to other A/D converter designs. Generation of the reference voltage supply is critical. It is recommended that analog and digital grounds are connected together. Care must be taken to avoid the digital current at V_{SS} injecting noise into the analog ground potential. The ground lines to V_{SS} and VAGND should be isolated from each other as much as possible. Figure 13 Power Supply Connections illustrates the recommended supply connections.

Figure 13
Power Supply Connections



8

For best reference voltage stability on VAREF, a separate power supply should be provided. If, for cost reasons, only one supply voltage is used for both digital and analogy supply, the voltage at VAREF should be stabilized with a lowpass filter. Note that the differential internal resistance of the reference supply must not exceed 1 k Ω, as described above.

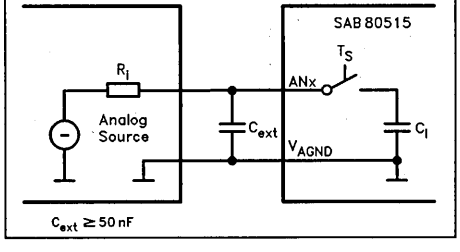
The supply should be grounded across a storage capacitor (tantalum) and a smaller HF-capacitor (ceramic) as near the device as possible.

Analog Input Supply

The use of analog power sources with higher internal resistances than 4 k Ω is possible under certain conditions. The internal resistances requirement is a result of the charging current necessary to overcome the analog input capacitance. This current is only needed during the sample time. If the analog value changes only slowly with time, a large external capacitor at the analog input is able to supply the charging current during the sample time. This is shown in Figure 14 Analog Input Supply.

The external capacitor should be at least 1000 times the value of the internal capacitance (50 nF = 1000 X 50 pF). The error induced by the external capacitor is therefore kept less than 0.1%.

Figure 14
Analog Input Supply



Using an external capacitor, the analog source must replace only that charge lost by the external capacitor. The time available for this is the time between two consecutive conversions on this channel. This depends on user software, but is at least three times the sample time if continuous conversion on the same channel is selected.

The equation for Ri can be applied here:

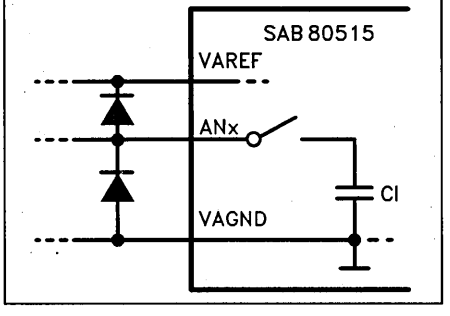
$$Ri \leq \frac{T}{10 \times CI}$$

Ri ≤ T/10 X CI

The difference is the time "T" between two consecutive conversions on the channel. This time is longer than the sample time, which permits higher internal resistance in the analog power source.

An analog input overload protection can be provided by two diodes, as shown in Figure 15 Protection Diodes for Analog Inputs. A Schottky diode satisfies the rating for the connection to VAGND because of its lower forward voltage characteristic.

Figure 15
Protection Diodes for Analog Inputs



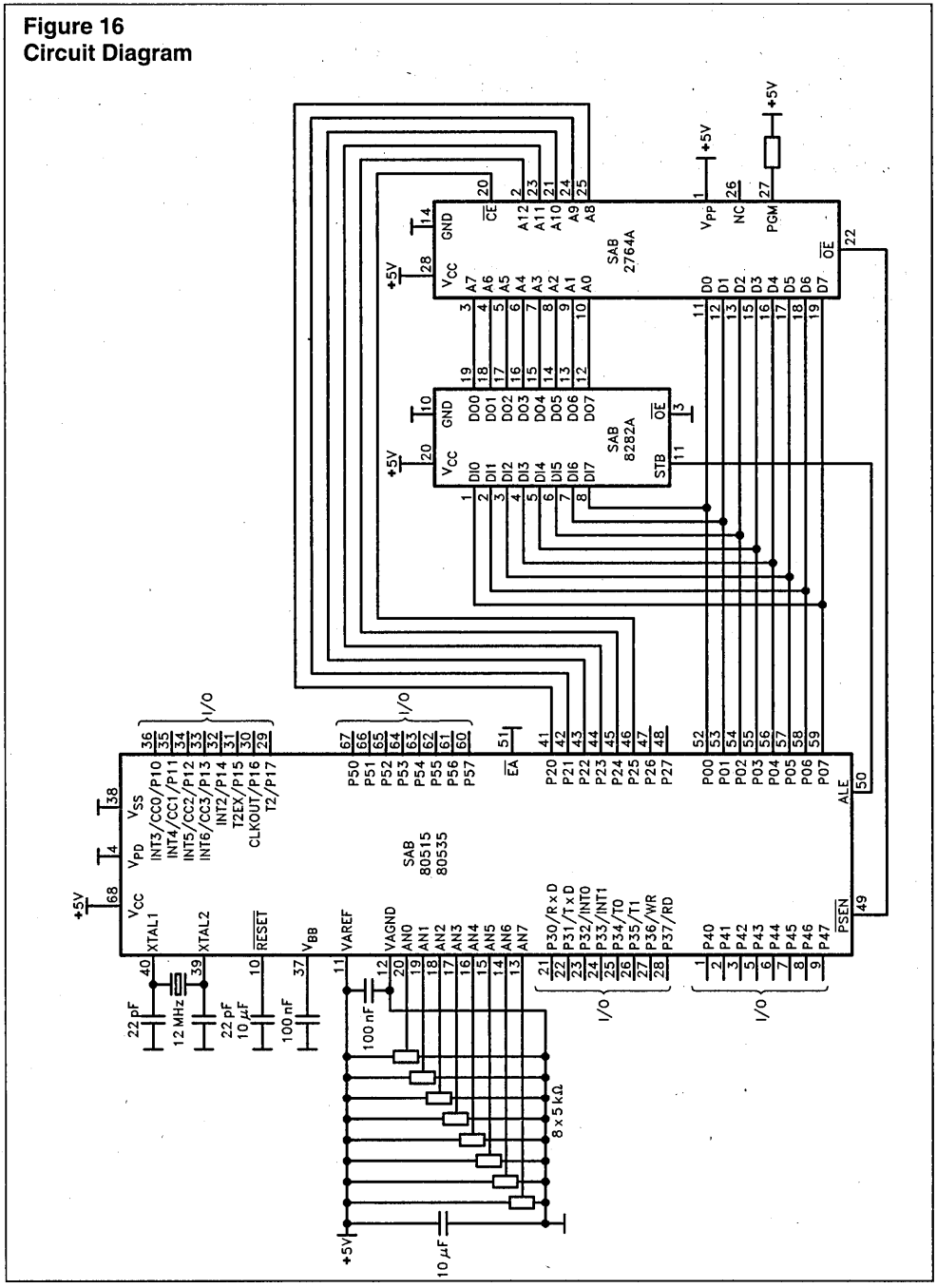
Application Example

The following application shows how to use the A/D converter for 8 different analog signals applied at the pins AN0-AN7. The analog values are generated by eight potentiometers. A terminal can be used as an output device through the SAB 80515/ 80535's serial channel.

The "single conversion" operating mode is used. Operation begins by writing a "00H" to DAPR, which deselects the internal reference voltage programming. The program polls the BSY flag for the termination of the conversion. The result is converted into BCD format and sent to the terminal. This procedure is followed for each of the eight analog channels.

Programming details are given in the program listings.

Figure 16
Circuit Diagram



```
MCS-51 MACRO ASSEMBLER                APPLICATION NOTE ON SAB 80515 A/D CONVERTER                19.02.86

DOS 3.20 (038-N) MCS-51 MACRO ASSEMBLER, V2.3
OBJECT MODULE PLACED IN ADNOTE.OBJ
ASSEMBLER INVOKED BY:  C:\INTEL\ASM51.EXE ADNOTE.A51


LOC  OBJ            LINE     SOURCE

                       1     $TITLE(               APPLICATION NOTE ON SAB 80515 A/D CONVERTER)
                       2     $DATE(19.02.86)
                       3     $NOSYMBOLS
                       4     $DEBUG
                       5     $NOMOD51
                       6     $INCLUDE(REG515.PDF)
                 =1    7 +1  $NOLIST
                     165
                     166     ;*******************************************************************
                     167     ;*                                                                 *
                     168     ;*              APPLICATION NOTE FOR THE SAB 80515                 *
                     169     ;*                   A / D   C O N V E R T E R                     *
                     170     ;*                                                                 *
                     171     ;*******************************************************************
                     172
----                 173             CSEG    AT      00H
                     174
0000 020003          175             LJMP    INITIALISATION
                     176
                     177     INITIALISATION:
                     178
0003 758180          179             MOV     SP,#80H                 ;INITIALIZE STACKPOINTER
0006 53877F          180             ANL     PCON,#7FH
0009 75D880          181             MOV     ADCON,#80H              ;4800 BAUD FOR SERIAL INTERFACE
                     182                                             ;WITH F OSC = 12 MHZ
                     183
000C 7598CA          184             MOV     SCON,#11001010B         ;9-BIT UART(MODE 3), NO RECEPTION
                     185
                     186
                     187     FIRST_MESSAGE:
                     188
000F 900097          189             MOV     DPTR,#MESSAGE_1
0012 12003B          190             CALL    MESSAGE_OUT             ;SEND FIRST MESSAGE TO TERMINAL
                     191
                     192
                     193     ;=======================================
                     194
                     195     CONVERSION:
                     196
0015 7A08            197             MOV     R2,#8D                  ;LOOP COUNTER
0017 7B00            198             MOV     R3,#00H                 ;ANALOG INPUT COUNTER
                     199
                     200     CONV_LOOP:
                     201
0019 EB              202             MOV     A,R3
001A 53D8C0          203             ANL     ADCON,#11000000B        ;MODE FOR A/D CONVERSION: SINGLE
001D 42D8            204             ORL     ADCON,A                 ;SELECT ANALOG CHANNEL
001F 75DA00          205             MOV     DAPR,#00H               ;START CONVERSION, NO INTERNAL
                     206                                             ;REFERENCE VOLTAGES USED
                     207
0022 120046          208             CALL    SPACE                   ;SPACES (20H) TO TERMINAL
                     209
0025 20DCFD          210             JB      BSY,$                   ;WAITING FOR TERMINATION OF CONVERSION
                     211
0028 A8D9            212             MOV     R0,ADDAT
002A 120050          213             CALL    BCD_OUT                 ;RESULT TO TERMINAL
                     214
002D 08              215             INC     R3                      ;NEXT CHANNEL
002E DAE9            216             DJNZ    R2,CONV_LOOP            ;END OF LOOP
                     217
0030 740D            218             MOV     A,#0DH
0032 120086          219             CALL    DISPLAY                 ;'CARRIAGE RETURN' TO TERMINAL
                     220
0035 12008E          221             CALL    DELAY                   ;WAIT A MOMENT!
0038 020015          222             LJMP    CONVERSION              ;AGAIN !
                     223
                     224     ;=======================================
                     225
                     226
                     227
                     228
```

8

```
MCS-51 MACRO ASSEMBLER          APPLICATION NOTE ON SAB 80515 A/D CONVERTER          19.02.86

LOC  OBJ            LINE     SOURCE

                     229     ;=========================================================================
                     230     ;    SUBROUTINES!!!!
                     231     ;
                     232     ;---------------------------------------
                     233     MESSAGE_OUT:
                     234
0038 E4              235             CLR     A
003C 93              236             MOVC    A,@A+DPTR               ;FETCH CHARACTER
                     237
003D 6006            238             JZ      END_MESSAGE             ;END OF MESSAGE ?
                     239
003F 120086          240             CALL    DISPLAY                 ;CHARACTER TO TERMINAL
                     241
0042 A3              242             INC     DPTR
0043 80F6            243             SJMP    MESSAGE_OUT             ;NEXT CHARACTER
                     244
                     245     END_MESSAGE:
0045 22              246             RET
                     247     ;---------------------------------------
                     248
                     249     ;---------------------------------------
                     250     SPACE:
                     251
0046 7420            252             MOV     A,#' '
0048 7906            253             MOV     R1,#6
                     254
                     255     SPACE_LOOP:
004A 120086          256             CALL    DISPLAY
004D D9FB            257             DJNZ    R1,SPACE_LOOP           ;6 BLANKS TO TERMINAL
                     258
004F 22              259             RET
                     260     ;---------------------------------------
                     261
                     262     ;---------------------------------------
                     263     BCD_OUT:
                     264
0050 E8              265             MOV     A,R0                    ;HEX NUMBER COMES IN R0!
                     266
                     267     HUNDREDS:
0051 75F064          268             MOV     B,#100D
0054 84              269             DIV     AB
0055 6009            270             JZ      HUND1                   ;REPLACE '0' WITH ' '
0057 C2D5            271             CLR     F0                      ;FLAG FOR '0'
0059 2430            272             ADD     A,#30H                  ;CONVERSION TO ASCII
005B 120086          273             CALL    DISPLAY                 ;TO TERMINAL
005E 8007            274             SJMP    TENS
0060 7420            275     HUND1:  MOV     A,#' '
0062 120086          276             CALL    DISPLAY                 ;BLANK TO TERMINAL
0065 D2D5            277             SETB    F0
                     278     TENS:
0067 E5F0            279             MOV     A,B                     ;REMAINDER TO ACCU
0069 75F00A          280             MOV     B,#10D
006C 84              281             DIV     AB
006D 7003            282             JNZ     TEN3                    ;REPLACE '0' WITH ' '
006F 20D507          283     TEN1:   JB      F0,TEN2
0072 2430            284     TEN3:   ADD     A,#30H                  ;CONVERSION TO ASCII
0074 120086          285             CALL    DISPLAY                 ;TO TERMINAL
0077 8005            286             SJMP    ONES
0079 7420            287     TEN2:   MOV     A,#' '
007B 120086          288             CALL    DISPLAY                 ;BLANK TO TERMINAL
                     289
                     290     ONES:
007E E5F0            291             MOV     A,B                     ;REMAINDER TO ACCU
0080 2430            292             ADD     A,#30H                  ;CONVERSION TO ASCII
0082 120086          293             CALL    DISPLAY                 ;TO TERMINAL
                     294
0085 22              295             RET
                     296     ;---------------------------------------
                     297
                     298     ;---------------------------------------
                     299     DISPLAY:
                     300
0086 3099FD          301             JNB     TI,$                    ;WAITING FOR END OF LAST TRANSMIT
0089 C299            302             CLR     TI
008B F599            303             MOV     SBUF,A                  ;SEND CHARACTER
008D 22              304             RET
```

```
MCS-51 MACRO ASSEMBLER              APPLICATION NOTE ON SAB 80515 A/D CONVERTER          19.02.86

LOC  OBJ            LINE     SOURCE

                     305     ;--------------------------------------
                     306
                     307     ;--------------------------------------
                     308     DELAY:
                     309
008E 7F06            310             MOV     R7,#6
                     311     DELAY_LOOP:
0090 DDFE            312             DJNZ    R5,$
0092 DEFC            313             DJNZ    R6,DELAY_LOOP
0094 DFFA            314             DJNZ    R7,DELAY_LOOP
0096 22              315             RET
                     316     ;--------------------------------------
                     317
                     318
                     319     MESSAGE_1:
0097 1B              320             DB      1BH,'E'                 ;CLEAR SCREEN
0098 45
0099 0D              321             DB      0DH,0AH                 ;CR, LF
009A 0A
009B 0D              322             DB      0DH,0AH
009C 0A
009D 0D              323             DB      0DH,0AH
009E 0A
009F 0D              324             DB      0DH,0AH
00A0 0A
00A1 20202020        325             DB      '               ***   A / D  C O N V E R T E R   D E M O   ***'
00A5 20202020
00A9 20202020
00AD 20202A2A
00B1 2A202020
00B5 2041202F
00B9 20442020
00BD 43204F20
00C1 4E205620
00C5 45205220
00C9 54204520
00CD 52202020
00D1 44204520
00D5 4D204F20
00D9 2020202A
00DD 2A2A
00DF 0D              326             DB      0DH,0AH
00E0 0A
00E1 0D              327             DB      0DH,0AH
00E2 0A
00E3 20202020        328             DB      '               ***         S A B   8 0 5 1 5           ***'
00E7 20202020
00EB 20202020
00EF 20202A2A
00F3 2A202020
00F7 20202020
00FB 20202053
00FF 20412042
0103 20202038
0107 20302035
010B 20312035
010F 20202020
0113 20202020
0117 20202020
011B 2020202A
011F 2A2A
0121 0D              329             DB      0DH,0AH
0122 0A
0123 0D              330             DB      0DH,0AH
0124 0A
0125 0D              331             DB      0DH,0AH
0126 0A
0127 0D              332             DB      0DH,0AH
```

8

On-Chip A/D Converters in Siemens SAB 8051-Based Microcontrollers

SAB-51 Family of Microcontrollers

8

Application Note

Introduction

This application note discusses technical details of the on-chip A/D converter integrated into several members of the SIEMENS SAB 8051 Microcontroller Family. The information is a continuation of an earlier application note on the A/D converter in the SAB 80515/80535.

Operation of the A/D Converter in the SAB 80515

The technical details given are background information on this on-chip peripheral, intended to help in designing these microcontrollers into difficult and electrically noisy environments.

This application note includes discussion of the A/D Converter specification, as well as recent updates to this specification.

SAB 80515's A/D Converter Architecture

The following description of the on-chip A/D converter unit (ADCU) concentrates on hardware and specification details of this particular microcontroller peripheral.

Conversion Principle

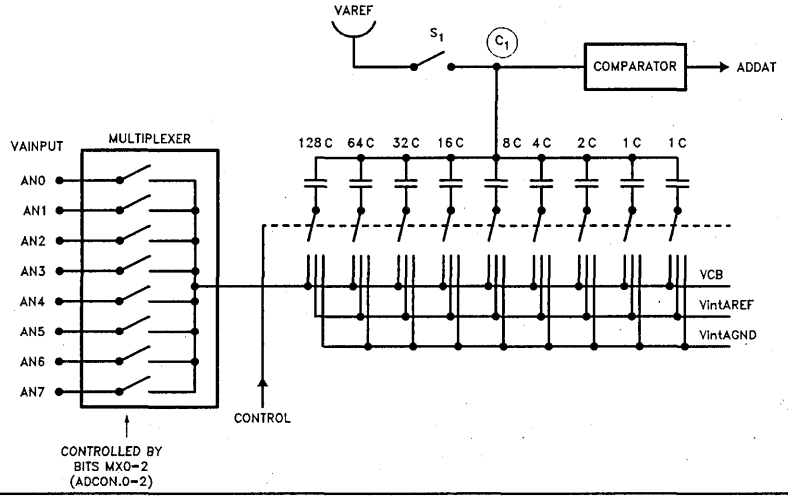
The ADCU conversion uses the successive approximation principle. Instead of an R-2R resistor ladder with which the analog input is compared, the ADCU uses an array of 256 small capacitors which are charged through the analog input. These capacitors not only perform the sample-and-hold function, but are also grouped into the binary weights used in the conversion process itself.

The "top ends" of all capacitors are connected to the comparator and, through the switch S1, to the positive reference voltage VAREF. The comparator itself is described in the next section. The connection to the "bottom" of each group of binary weighted capacitors can be connected to either the selected analog input or to one of the two internal reference voltages.

For the sake of explanation, it is assumed that the internal reference voltages are connected directly to the reference voltages applied to pins VAGND and VAREF.

Figure 17 Detailed Block Diagram shows a block diagram of the ADCU. The conversion process itself is described in the following three steps. The timings used in the explanation are based on the use of a crystal oscillator of 12 MHz.

Figure 17
Detailed Block Diagram



Step 1 (1.0 µs-2.0 µs)

At the start of the conversion process, switch S1 is closed. This switch connects the top of all capacitors to the reference voltage VAREF. The bottom of each group of capacitors is connected to one analog input by selecting one of eight multiplexer channels with bits ADCON0-ADCON2.

Interpretation of Step 1

This step is the first part of the sample phase and is known as the load time (TL) in the specification. During these 2 µs, the complete capacitor-array has to be precharged in the analog input through the VAREF-pin. At the end of this period, the voltage VCB at the bottom of the capacitor network must be the same as the analog input being measured (= VAINPUT).

Step 2 (3.0 µs-5.0 µs)

Switch S1 is opened. The top of the capacitor-array remains connected to the high impedance comparator input and the bottom to the analog input. During the next 3 µs, the comparator is adjusted to the voltage of node C1 (for details of the comparator see next section). This potential still equals to the voltage at VAREF since node C1 was precharged to that voltage during step 1.

Interpretation of Step 2

This second part of the sample phase follows the load phase. The analog input is still connected to the bottom of the capacitor-array. Any change in the voltage at the analog input pin such as a spike or transition will be capacitively coupled to node C1. This will disturb the adjustment of the comparator. During this time the analog input must therefore be held strictly constant.

Step 3 (Remaining Conversion Time)

During this time the actual conversion process (successive approximation) takes place:

The bottom of each group of capacitors is disconnected from the analog input and is pulled to VINTAGND. Node C1 is currently at

8

a potential VC1 = (VAREF - VAINPUT).

The first group of capacitor-cells (128C, corresponding to the Most Significant Bit (MSB)), is then switched to VINTAREF. This causes a charge transfer in the capacitor array and thereby a superimposition of VC1 with VAREF/2, i.e. the potential of node C1 alters to become VC1 = (VAREF - VAINPUT + VAREF/2). The new VC1 is now compared to VAREF (adjustment-voltage of the comparator). Depending on the result of the comparison, this 128C-group remains at VINTAREF (for VC1 VINTAREF) or is switched back to VINTAGND (for VC1 > VINTAREF).

The same process is repeated with the next group of capacitor-cells (in this case the 64C capacitor). The conversion is complete when the last capacitor-group (1C) of one capacitor cell has been compared.

Upon completion of the comparison, the voltage VC1 applied to the comparator can be calculated by the formula:

$$VAREF - VAINPUT + \sum_{i=0}^{7} bi \times \frac{VINTAREF}{2^{8-i}}$$

All bits bi (i = 0 to 7) are latched in the position of the switches at the bottom of each capacitor group, thereby providing the result of the conversion process.

Interpretation of Step 3

The actual conversion is done in step 3 when the analog input voltage is measured against the weighted capacitor network. Since the analog input is disconnected from the capacitor network, changes at the analog input no longer affect the conversion result. Note that during this time, charge transfer in the capacitor-array still causes dynamic current spikes drawn from the analog reference. However, from the user's point of view, step 3 is the least critical phase of the whole conversion process.

The Comparator

The accuracy of an A/D-converter is very dependent on the comparator used. The previous section, especially step 2, stated that the ADCU comparator is susceptible to noise or spikes through the analog input pin during certain time windows. A standard comparator (such as is used in a common OP-AMP application) compares the input value with a reference, which is always present. The ADCU comparator can be viewed as an inverting three-stage amplifier with some coupling capacitors. It is precharged initially with a reference voltage which is then used as a threshold point.

The principle of this three-stage comparator can be explained through the function of one stage. It consists of a coupling capacitor in series with an inverting amplifier as shown in Figure 18a. A feedback switch allows the inverter's output to be short-circuited with its input.

The feedback-switch is closed during the load phase of the comparator-stage, as discussed in step 1. This causes the coupling capacitor to buffer the differential voltage VD between the input stage and the operating point of the inverting amplifier. Opening the feedback switch activates the comparator at the previously adjusted potential (Figure 18b). The amplifier responds to small variations from this potential with an amplified output. A concatenation of three of these stages results in an amplification which digitizes even the smallest variation at the comparator input.

Additionally, the three stages of the adjustment process can be serialized by opening the three feedback switches one after another. This compensates for unintentional noise in the coupling capacitors which might be generated by opening the switches during the 3 µs duration of step 2 mentioned above. This explains why the

comparator is susceptible to noise during this period. Any spikes or noise capacitively coupled to its input during the last phase of the sample time might shift the comparator threshold.

A smooth analog input signal should be supplied during the whole sample phase (TS). A transition on the analog input signal during the critical time of the sample phase may give unexpected results. On the other hand, this comparator concept, combined with the conversion principle described above, provides high immunity against temperature changes and results in a narrow variation of device parameters during production.

Expansion and Update of the ADCU Specification

The ADCU specifications in the datasheet give generally condensed information on A/D converter characteristics. This section provides the reader with a more detailed interpretation of the main points in the specification. It also updates the specification with more recent information.

VAREF/VAGND Voltages

The limits of the device reference voltage inputs are a function of the microcontroller type and operational supply voltage. Some devices have a standard ratiometric ADCU on board, which allows an adjustment of the reference over a wide range of external voltages. Other devices have internally adjustable reference voltages, the voltage range being controlled by software.

The latter type cannot function correctly with any variation in the reference voltages. The voltages applied to VAREF and VAGND are specified within a very narrow range of the supply voltage for such devices.

Whichever type of microcontroller is used, the reference voltage, once adjusted to the desired value within specifications, must be held steady during the entire conversion process. This, in turn, requires a low impedance to the reference source. Reference voltage impedance and analog source impedance requirements are discussed in the following sections.

For reference voltage specifications refer to either the A/D Converter Characteristics given in the respective datasheets or to the list given below in Table 1.

For all parts VAREF - VAGND must be greater than 1V or four steps of the internally programmable reference voltages, whichever is less.

8

Figure 18
One Stage of the Comparator

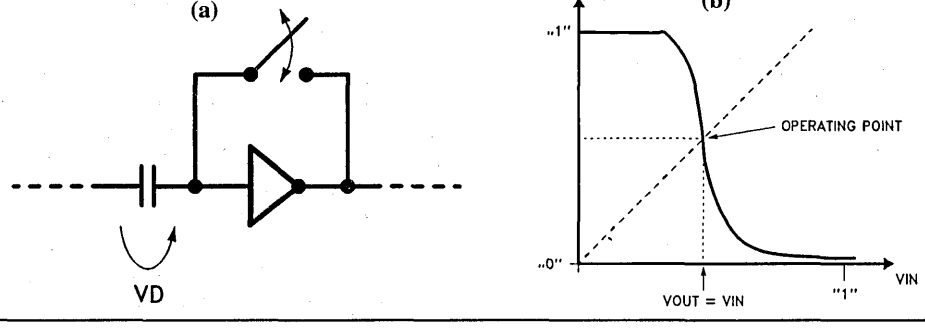


Table 1 Voltage Limits on VAREF and VAGND

Device	VSS	VCC	VAGNDmin<	VAGNDmax	VAREFmin	VAREFmax
SAB 80515	0V	5V ±10%	0V	0V	5V	5V
SAB 80C515	0V	5V ±10%	0V	0V	5V	5V

Timing

Timing requirements have already been discussed in the description of the conversion principle.

The load time TL is an important ADCU characteristic and has therefore been added to the specification.

The load time TL is specified as the period during which the ADCU internal capacitance must be charged by the analog source. TL is the first part of the sample time TS. The ADCU uses the remaining part to adjust its comparator. The entire conversion time TC consists of TS (which includes TL) and a part during which the conversion is performed. TC is specified in the datasheet and differs slightly among device types.

Analog and Reference Voltage Source Impedance Requirements

A/D Converter Characteristics in the datasheets contains absolute values required for impedances of the reference and analog source only. This section gives pointers on determining the ADCU's requirements on the analog part of an application as well.

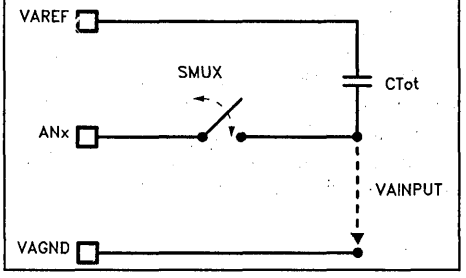
Impedance of the Analog Source

Two parameters from the ADCU's specification are necessary to determine the maximum impedance of a signal source such as a sensor:

- the load time TL
- the maximum capacitance of the on-chip capacitor array (CTot).

Figure 19 Equivalent Circuit for the ADCU during Load Time shows a simple equivalent circuit for the ADCU during load time TL. At the moment a conversion is started, a multiplexer switch to the selected channel is closed and the analog input is connected to the internal capacitors of the ADCU. This is shown as SMUX in Figure 19.

Figure 19
Equivalent Circuit for the ADCU during Load Time



The ADCU's entire internal capacitance consists of the capacitor array, which is typically 25 pF, and stray capacitance from the substrate equivalent to another 20 pF-40 pF to substrate. The actual amount depends on the microcontroller itself. In any case, the internal capacitance total must be charged by the analog source. This implies that a constant input impedance cannot be specified for the ADCU.

The switch SMUX and the internal traces to the capacitors array have very low resistance. During TL, the ADCU can be regarded as a capacitor with a capacitance of CTot. The internal resistance of the whole analog circuitry connected to the analog inputs of the ADCU can be defined as RI. RI and CTot form an RC-element, with time constant t determined by "t = RI * CTot".

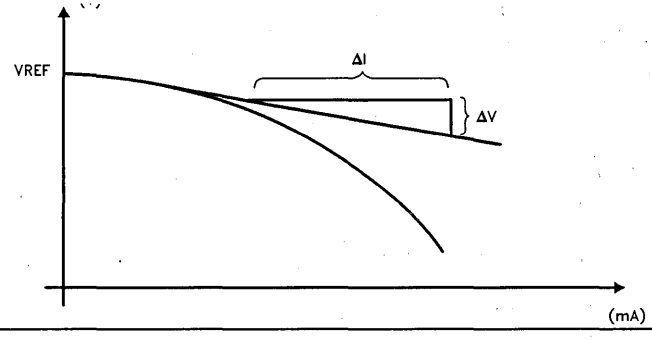
Given that an accuracy better than 0.05% is required, the formula

$VC = (1 - e^{-TL/\tau}) * VAINPUT$

gives $\tau = 0.13 * TL$

As an example, assume that the load time of the ADCU is TL = 2 µs and its maximum internal capacitance CTot = 65 pF. The above formulae result in a maximum impedance for the analog source of:
RI = 0.13 TL/CTot = 4 k Ω

Figure 20
Definition of the Reference Supply Impedance



Observing the above rules for the analog circuit impedance ensures that at least 99.95% of VAINPUT is sampled during load time TL. Sampling actually continues during the whole time TS to improve the match between VAINPUT and the voltage at CTot.

The value for RI given above is meant as an example. For a given application, the calculation should be done using the values for CTot and the accuracy desired.

Impedance of the Reference Source

During the conversion process, each group of capacitor cells is individually switched to either VAREF or VAGND. Due to this switching and associated charge transfers in the capacitor array, the reference source must supply additional current over and above the current some parts use to generate internal reference voltages.

The reference output impedance must be low enough to supply this additional current and is therefore specified as a differential impedance. The diagram in Figure 20 Definition of the Reference Supply Impedance illustrates the differential impedance, rD, which is given by the formula:

$$rD = \frac{dV}{dI}$$

where V = VREF

The actual value for rD can be referenced under A/D converter characteristics in the datasheet.

3.4 Specification of Errors

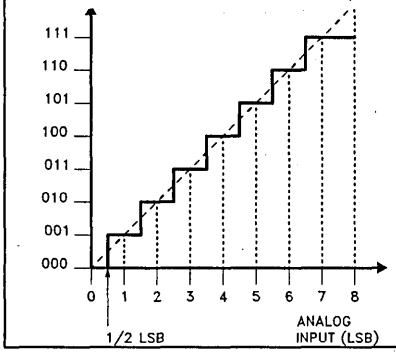
Several error sources which modify the ideal transfer characteristic of an A/D converter were defined and discussed in the application note titled "Operation of the A/D converter in the SAB 80515/80535". The error sources included:

- Offset Error
- Integral Non-Linearity
- Differential Non-Linearity
- Gain Error

8

Figure 21a shows a diagram of the ideal transfer characteristic of any A/D converter. A 3-bit converter is shown here for clarity. All analog values within a given quantum are represented by the same digital value, which corresponds to the mid value of this quantum. Connecting all of these mid values leads to a straight line, which, for an ideal A/D converter, is the diagonal in the 1st quadrant. This also implies that the first step to the digital value 01 occurs when an analog input voltage is applied which corresponds to a 1/2 LSB. All error definitions stated in the datasheet refers to this ideal transfer characteristic.

Figure 21a
Ideal Transfer Characteristic



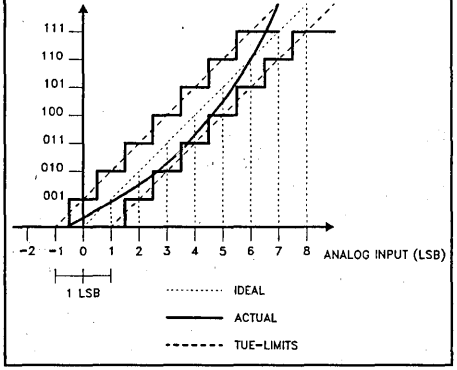
All the above errors affect the A/D converter ideal transfer characteristic. Their effect is defined as the Total Unadjusted Error (TUE), which is now an A/D converter characteristic.

The TUE of the ADCU is not simply the sum of several individually measured errors. Since some ADC errors can cancel each other, e.g. the offset and the gain error, the Total Unadjusted Error can be less than the sum of individual errors.

Some ADCU errors can be compensated for. A negative offset error can be adjusted in software by subtracting known offset from the actually measured value. The TUE, however is an "unadjusted" error, which means that no correction method has been taken into account. It therefore gives a true image of the converter accuracy.

Figure 21b shows the TUE as a maximum deviation of the actual transfer characteristic (bold line) from the ideal transfer characteristic corresponding to the full scale range. The TUE defines the range of the transfer characteristic where the real value must lie.

Figure 21b
The Total Unadjusted Error



Accuracy/Resolution in a Reduced Reference Voltage Range

General Considerations

All A/D converter units of the SAB 8051 family are ratiometric A/D converters. Some (e.g the SAB 80512) allow a reduction of the reference voltage range external to the chip, others (e.g., the SAB 80515) provide software-adjustable internal reference voltages. This section deals with the effect of reducing the reference voltage range on ADCU accuracy.

Generally, the absolute **accuracy** of an 8-bit A/D converter cannot be improved by using a smaller reference voltage range. The errors in an ADCU originate in physical

phenomena like temperature substrate noise, etc., and these cannot be reduced by reducing the reference voltage.

However, this is not the point in ratiometric A/D converters. The main advantage of these converters is that they are able to provide higher resolution. Since the distinction between accuracy and resolution can become a source of confusion, it is worth discussing further.

Accuracy versus Resolution

The advantage of ratiometric A/D converters is that they increase the resolution of the analog input. Increasing the resolution just means that the quantum of analog values which corresponds to one digital value is reduced.

In case of the Siemens Microcontrollers, this is performed by selecting a smaller reference voltage range, i.e. by varying the lower or higher reference voltage. All ADCUs in the Siemens SAB 8051 family provide the equivalent of a 10-bit resolution by a reduction of the reference voltage range to as little as 1V. An ADCU such as that in the SAB 80515 allows for a dynamic (software controlled) variation of the reference voltage. Therefore this converter, which is actually an 8-bit converter, can look like a 10-bit converter over the full conversion-range from 0V to 5V using appropriate software. The absolute accuracy of this "10-bit converter", is the absolute deviation between the analog input and the digital result. This is not necessarily any better than that of the "8-bit converter" (VAGND = 0V and VAREF = 5V), although it can be improved by implementing correction methods in the control program. The resolution, on the other hand, is improved by using a smaller reference voltage range. An analog interval which gave one digital step with a 5V reference voltage gives four digital steps in the same interval, and therefore an additional two bits of resolution.

The adjustable reference voltage feature, in combination with software error correction algorithms make precise A-to-D conversions with low-cost on-chip converters possible.

Errors in a Reduced Reference Voltage Range

All error definitions in the datasheet A/D converter characteristics refer to the standard conversion range of 0V to 5V, (the "8-bit range"). For the purpose of the following discussion, the term "LSB8" is used to describe one LSB (least significant bit) over this range. Similarly, "LSB10" describes one LSB of a conversion made within a range of a quarter of the full reference voltage range.

As discussed above, the four different errors which affect the accuracy of an A/D converter are:

Offset Error
Gain Error
DNLE (Differential Nonlinearity Error)
INLE (Integral Nonlinearity Error)

This section discusses the effect of a variation in reference voltage range on these errors.

Offset and Gain Error, Integral Nonlinearity

These errors are systematic errors. This means that they occur independent of application, disregarding temperature drift. Reducing of the reference voltage range will not reduce these errors.

Take as an example an ADCU with a reference voltage range of VAGND = 0V and VAREF = 5.0V. Assume a negative offset error of 1/2 LSB8 and no other errors. This means that the first transition of the transfer characteristic (the change from 00H to 01H) occurs at an input voltage of VAINPUT = 0.00V. According to the ideal transfer characteristic, the absolute offset error is approximately -10 mV. If we now reduce the upper internal reference voltage to 1.25V,

then 1 LSB (now an "LSB10") of the digital result now refers to a quantum of approximately 5 mV. For VAINPUT = 0.00V, with the same Offset error the conversion result is 2 LSB10 (ADDAT = 02H). Thus, an absolute offset error is independent of the reference voltage selected. In other words, the error is the same regardless which "window" within the full scale range is selected..

A similar analysis shows that the same applies to both the Integral Nonlinearity Error and to the Gain Error if these are considered separately.

Both the Offset Error, and the Integral Nonlinearity Error can be disregarded for our ADCUs, since they can be compensated for in software. The Gain Error is more significant. Since it is given as the difference in slope measured at its biggest extent, it is by definition a scaled error comparison of an actual A-to-D converter transfer characteristic and comparing with an ideal characteristic shows a combination of errors. Some of these compensate for each other and can be determined only at certain points on the transfer characteristic.

To determine the actual "absolute error" at a given input voltage, both Gain and Offset Errors must be summed. Differential Nonlinearity is not considered yet. It will be discussed in detail later. The following example demonstrates Gain and Offset errors for different reference voltage ranges.

To simplify matters, again consider the 3-bit converter. This has an Offset error of -2 LSB and a Gain error of +1 LSB over full range. Figure 22 Offset and Gain Error in a Reduced Reference Voltage Range shows the effect of both errors on the transfer characteristic of the converter. Reducing the reference voltage range has no influence on the absolute accuracy of the conversion. The dotted window in Figure 22 Offset and Gain Error in a Reduced Reference Voltage Range shows the reduced reference voltage range. In this case, it is half the size of the full scale range. Using a terminology similar to that above, LSBs are therefore LSB3 in the full range and LSB4 in the reduced.

The dotted window in Figure 22 Offset and Gain Error in a Reduced Reference Voltage Range illustrates that changing the reference voltages just extracts a window out of the full scale range and increases resolution. The actual Offset and Gain Error line crosses the y-axis of the small diagram at the binary value 011 three LSB4 units. This three LSB4 error is a combined error of four LSB4 Offset Error and -1 LSB4 Gain Error. The Gain Error in the reduced range is scaled down to one LSB4 (1/2 LSB3) because this error is measured in the middle of the transfer characteristic. The lower reference voltage of the reduced range is at 2 LSB3 and the offset error also 2 LSB3, which together gives 4 LSB3 on the x-axis. This is the middle of the full range.

Summarizing, it can be said that Offset-, Gain-Error and Integral Nonlinearity of the ratiometric A-to-D converters in the SIEMENS 8051-family can be regarded as absolute errors. Absolute errors are the deviation of the actual from the ideal value at a given input voltage. Changing the reference voltages has no effect on these errors. The number of LSBs in which the errors are expressed depends on the reference range and therefore on the "LSB-Unit" (LSB8 or LSB10).

Differential Nonlinearity Error (DNLE)

The Offset and Gain Error, described above are, in most cases, bigger than Nonlinear Errors. On the other hand, they are stable and can be compensated for by software.

The Differential Nonlinearity Error is a statistical error with more than one source. A major part is due to the noise inherent in any mixed analog/digital system. This noise derives from the controller's environment on the PC-board and from the device itself. Even good A/D Converter test-boards carry some noise on the analog lines. Distinguishing which part of the DNLE is induced by the test equipment and which by the chip itself is difficult. There is also a small DNLE in the A/D-converter which can't be avoided. This "self-induced" DNLE comes from on-chip noise and noise on the chip's substrate. More complex origins are inherent in conversion method principles. Empirical measurements show that the DNLE component is not a constant error and therefore varies with reference voltage range. The DNLE due to the converter itself in fact shows a dependence to the reference voltage selected. This DNL component scales with the size of the window selected.

10-Bit Resolution?

The datasheet error definition disagrees with the 10-bit resolution claimed above. 10-bit resolution is possible, even with a differential nonlinearity error of 1 eight-bit-LSB (LSB8) maximum. This corresponds to approximately 20 mV.

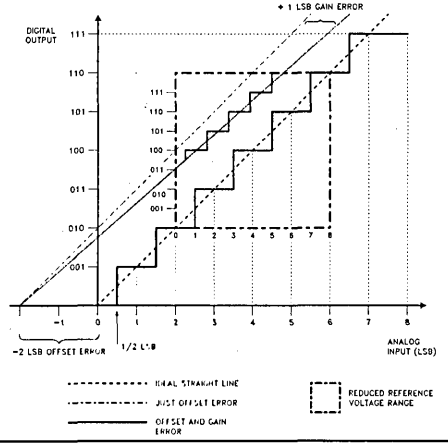
A DNLE of 1 LSB8 would result in missing codes in the transfer characteristics of the 10-bit range, which means that 10-bit resolution appears to be "inappropriate".

However, the DNLE discussed in the above paragraph is an exception to the principle of "absolute errors". The "self-induced" DNLE of the chip presents no problem for 10-bit resolution. The problem is the chip's susceptibility to external noise. Keeping this noise as small as possible is a major task for the design engineer. Further application notes will give some design hints for device connections necessary for high resolution with minimal errors.

In summary, the typical DNLE is generally under 1 LSB8 and depends heavily on the PC-board environment in which the microcontroller is operated. Observing the design rules for dealing with weak analog signals leads to a typical DNLE of less than 1 LSB10 (10-bit LSB) with no missing code.

8

Figure 22
Offset and Gain Error in a Reduced Reference Voltage Range



All errors other than DNLE are systematic errors. They affect the absolute accuracy of A/D conversion, but cannot lead to missing codes and therefore don't affect resolution.

Errors in the Internal Reference Voltages

For microcontrollers which provide on-board ADCUs with programmable reference voltages, another relevant parameter is the accuracy of the internal reference voltages. The absolute accuracy of a conversion result measured in a reduced reference voltage range also depends on the accuracy with which the internal reference voltages can be adjusted.

The internal reference voltages are generated by a simple digital-to-analog converter. It is essentially a tapped resistor ladder consisting of 16 equal resistors.

A write-to-DAPR instruction connects the internal reference voltage nodes to the appropriate taps. The lower and upper nibble of the SFR DAPR determines the tap position for the lower and higher reference voltage respectively. The accuracy of the internal reference voltages therefore depends on the accuracy of the internal resistor network. In the ideal case where all resistors of the network have exactly the same value, the differential voltage between two taps of the resistor ladder is exactly 1/16 of the reference voltage applied externally.

Slight variations in the ratio of the resistance values are inherent in the device and are unavoidable. These variations result in a so-called internal reference error which is now a part of the A/D Converter Characteristics specification.

This VINTREFERR defines the maximum deviation of the actual internal reference voltage from the ideal value. The ideal value depends on the reference voltage applied to the chip. The formulae given in the user's manual of the SAB 80515/80535 can be used to calculate the internal reference voltages.

Example 1:

VAGND = 0.00V DAPR = C4H

VAREF = 5.12V

$$\begin{pmatrix} VINTAGND = VAGND + \frac{DAPR\,(0-3)}{16} \bullet (VAREF - VAGND) = 1.280V \\ VINTAREF = VAGND + \frac{DAPR\,(4-7)}{16} \bullet (VAREF - VAGND) = 3.840V \end{pmatrix}$$

The above values for VINTAGND and VINTAFER are the ideal values. With VINTREFERR = ±15 mV, we get the following:

1.265V < VINTAGND < 1.295V

3.825V < VINTAREF < 3.855V

The resolution of the conversion in this example is 10 mV (9-bit). That means that in a 9-bit conversion with a reference voltage range of DAPR = C4H, the above deviation of the internal reference voltage must be taken into account.

Consequences of VINTREFERR

The importance of this error depends on the application.

The following two examples show the effect of the VINTREFERR in typical applications:

A. Measuring Absolute Voltages

This is a standard application for any A/D-Converter.

For the calculation of the Total Unadjusted Error in a **reduced reference voltage range**, VINTREFERR must be taken into account. If both internal reference voltages are set to taps of the resistor ladder, then the VINTREFERR or VINTAGND can be considered as an additional offset error. VINTREFERR of VINTAREF is additional gain error in the conversion. The following example shows the effect of the VINTREFERR on the accuracy of the conversion:

VAGND = 0.00V DAPR = 84H

VAREF = 5.12V VAINPUT = 1.500V

$$\begin{cases} \text{VINTAGND} = 1.280\text{V (ideal)} \\ \text{VINTAREF} = 2.56\text{V (ideal)} \\ \text{StepWidth} = \dfrac{\text{VAINPUT} - \text{VINTAGND}}{256} = 5\text{mV} \end{cases}$$

The correct result of the conversion would be:

$$\text{ADDAT} = \frac{\text{VAINPUT} - \text{VINTAGND}}{\text{StepWidth}} = 2\text{CH}$$

Let's assume that a hypothetical VINTREFERR leads to the following internal reference voltages.

VINTAGND = 1.28V - 5 mV

VINTAREF = 2.56V - 10 mV

This gives a new step width of 4.98 mV and an absolute offset error caused by the new VINTAGND of one LSB10. The actual result is now

$$\text{ADDAT} = \frac{1.500\text{V} - 1.275\text{V}}{4.98\text{mV}} = 45\text{D} = 2\text{DH}$$

Therefore, the error caused by the internal references in this example is one LSB10.

B. Measuring Differential Voltages

The feature of software-adjustable reference voltages is ideal for a tracking converter application.

The requirements for a tracking converter are high resolution and monotonic behavior. Both the requirements are accomplished by the ADCU of the SAB 80(C)515. In this case, the internal reference error is not significant. The maximum VINTREFERR merely changes the analog quantum for one digital value by

$$\pm\frac{\text{VINTREFERR}}{256}$$

As an example, taking the same parameters as in example 1:

VAREF = 5.12V VAGND = 0.00V

The smallest possible reference voltage range is therefore:

$$\frac{\text{VINTAREF} - \text{VINTAGND}}{4} = 1.28\text{V}$$

The resolution of the conversion is in this case 5.00 mV. That is, the analog quantum corresponding to each LSB is 5.00 mV. If we assume the worst case for both internal reference (-15 mV for VINTAGND and +15 mV for VINTAREF or vice versa), the analog equivalent to one LSB would expand or compress by 0.117 mV.

In other words this results in a step width error in the worst case of ~2.3%. The absolute amount of the error therefore increases with the differential voltage of the

signal between two sample points.

Summarizing, it may be said that the VINTREFERR, the error of the internal reference, cannot be disregarded in applications which use the internal reference voltages since this error affects the absolute accuracy of a conversion.

For clarity, the above examples deal with rather large errors. Typical values for the VINTREFERR are less than 5 mV.

Further, it should be mentioned that specifications are constantly being updated. Values are given in this application note to illustrate errors specified in the datasheet. The current datasheet should always be consulted for the latest error definitions and values.

References:

1. Application Note—Operation of the A/D Converter in the SAB 80515

2. H. Kessler, R. Rossgotterer
   Ein 8-Bit-Analog-Digital-Wandler in MOS-Silizium-Gate-Technologie nach dem Ladungsverteilungsverfahren
   Siemens Forshc.-u. Entwickl.-Ber. Bd. 8 (1979) Nr. 5

3. SAB 80515/80535, 8-Bit Single-Chip Microcontroller, Datasheet
   SAB 80512/80532, 8-Bit Single-Chip Microcontroller, Datasheet
   SAB 80C515/80C535 8-Bit Single-Chip Microcontroller, Datasheet
   SAB 80512K, 8-Bit Single-Chip Microcontroller, ROM-less Version, Datasheet
   SAB 80515K, 8-Bit Single-Chip Microcontroller, ROM-less Version, Datasheet

4. Microcomputer Components
   SAB 80515/80535 8-Bit Single-Chip Microcontroller User's Manual 7.85

5. Microcomputer Components
   SAB 80512/80532 8-Bit-Single-Chip Microcontroller User's Manual 2.88

10 Bits of Resolution with the 8-Bit On-Chip A/D Converter of the SAB 80515/80535

SAB 80515/80535

8

Application Note

Introduction

This application note describes a software routine to achieve 10 bits of resolution with the A/D converter of the SAB 80515/535.

Analog-to-digital (A/D) and digital-to-analog (D/A) converters are devices which interface physical parameters, which are analog, to digital computation and control. Some applications in which A/D converters are used include processing systems, sampled-data control systems, data telemetry systems and automatic test systems. Besides A/D converters, these systems usually employ transducers to interface with physical/analog quantities such as temperature, pressure, flow, acceleration and position, as well as microcontrollers or microprocessors to process the acquired data.

Siemens SAB 80515/535 microcontroller facilitates the design of a data conversion system by providing an on-chip A/D converter. The inherent 8031 based architecture of the SAB 80515/535 allows direct handling of 8-bit quantities only. However, by combining the on-chip special features of the A/D converter and the processing capability of the SAB 80515/535, a 10-bit result can be achieved and stored in two 8-bit registers.

Quantization and Resolution of an A/D Converter

Quantizing is the process of transforming a continuous analog signal into a set of discrete output states. Resolution of an A/D converter is the number of bits required to describe the output states. The number of output states for a binary coded A/D converter is 2^n where n is the number of bits. Thus an 8-bit A/D converter has an eight bit resolution and 256 output states. A 10-bit A/D converter has a 10-bit resolution and 1024 output states.

In any part of the input range of the A/D converter, there is a small range of analog values within which the same digital output is produced. This small range is known as the analog quantization size or quantum Q. The quantum is found by dividing the full scale analog range by the number of output states.

Therefore, $Q = FSR/2^n$ where FSR is the Full Scale Range and n is the number of bits.

The quantum Q, introduces an error and for a given analog input value to an A/D converter, the output error will vary anywhere from 0 to plus or minus Q/2. This error is called quantization noise. It can be reduced only by increasing the resolution of the converter, thereby making the quantization finer.

With 1024 possible quantized output states from a 10-bit A/D converter, a quantum for a full scale range of 5.12V will be equal to 5.12V/210 = 5 mV. The same level of quantum can be achieved with the 8-bit A/D converter on the SAB 80515/535. By programming the internal reference voltage sources, the entire analog input range of 5.12V may be divided into four ranges of 1.28V each. For a full scale range of 1.28V, the quantum is then equal to 1.28/28 = 5 mV. Therefore, the four ranges, each with 256 quantized output states will give a total of 1024 digital output states with the same quantum level as achieved by using a 10-bit A/D converter.

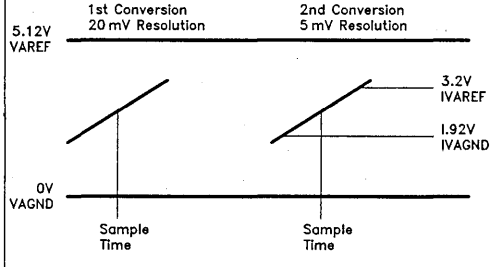
A/D Converter with Programmable Reference Voltages

The 8-bit A/D converter of the SAB 80515/535 has 8 multiplexed analog inputs and its operation is based on the method of successive approximation by using a capacitive load distribution. The use of capacitors in place of a resistive network ensures a better immunity against temperature and frequency changes, thereby providing a better accuracy of the A/D converter. The analog signal at the selected input channel is sampled for 5 machine cycles (5 µs at an oscillator frequency of 12 MHz), which will then be

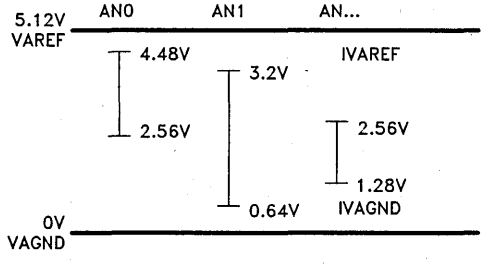
held constant at the sampled level for the rest of the conversion time of 10 µs at an oscillator frequency of 12 MHz. One-time or continuous conversions may be performed. The end of a conversion may cause an interrupt.

Moreover, the two internal reference voltages IVAREF and IVAGND can be programmed for a 4-bit resolution (16 steps), referred to the externally applied reference voltage VAREF. Each 4-bit value, one for IVAGND and another for IVAREF, is put in a D/A converter program register called DAPR. DAPR is an 8-bit register in which the low-order nibble holds the digital value of IVAGND, while the high-order nibble holds the digital value of IVAREF. As soon as a digital value is written to DAPR, the corresponding analog levels for the IVAREF and IVAGND are computed and the A/D conversion is then performed in this range. By reducing the range of conversion, the resolution of the A/D converter can be increased up to 10 bits ("Figure 23 Increase in A/D Converter Resolution"). Different internal analog voltage ranges may be set for each input by means of software.

Figure 23
Increase in A/D Converter Resolution



1. Improved resolution by second measurement



2. Matching various external voltage ranges to the A/D converter

8

Software

A simple way to achieve 10 bits of resolution with A/D converter of the SAB 80515/535 is to divide the entire 5.12V range (VAREF = 5.12V and VAGND = 0V) into four equal parts of 1.28V each. This would give four ranges starting from 0V to 1.28V, 1.28V to 2.56V, 2.56V to 3.84 and 3.84V to 5.12V. First an 8-bit conversion is performed by programming the IVAREF and IVAGND to 5.12V and 0V respectively. The 8-bit result obtained from this conversion has unique value for the two Most Significant Bits (MSBs) for each of the four ranges, i.e. 00XX XXXXB for range from 0V to 1.28V, 01XX XXXXB for range from 1.28V to 2.56, etc. These two MSBs constitute the two MSBs of the final 10-bit result. Based on this first conversion a decision is made as to which of the four ranges the analog signal belongs. The IVAREF and IVAGND are then programmed to have the respective values in the narrow range and another conversion is performed. The 8-bit result obtained, gives the next 8 bits of the final 10-bit result.

This straight-forward method of achieving 10-bit resolution introduces an error when the analog signal lies at the boundary of two

adjacent ranges. The irreducible error which results from the quantizing process may produce the digital result in one range, while the actual analog signal may belong to the adjacent range. This indeterminateness of the correct range introduces an additional error of 1/2 LSB.

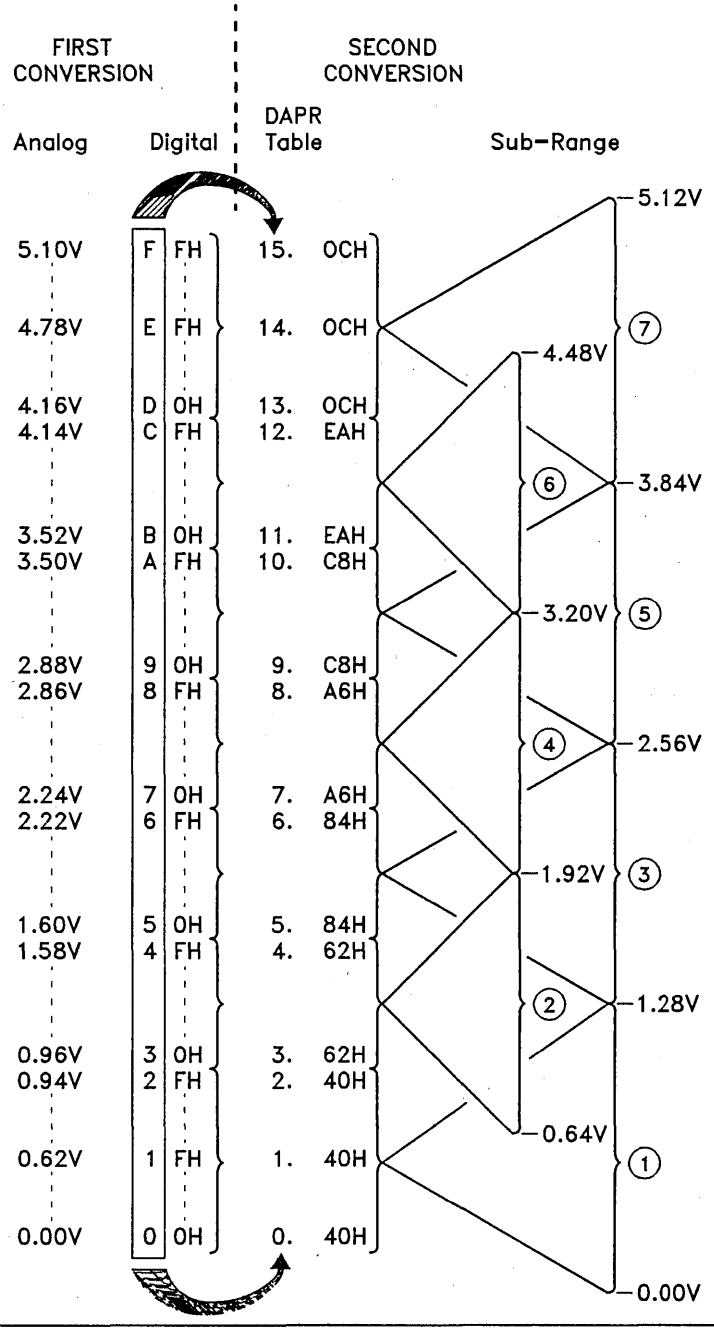
To circumvent this problem, another approach is adopted, in which the entire range of 5.12V is divided into several sub-ranges of 1.28V such that each sub-range overlaps the other at its mid-point, as shown in "Figure 24 A/D Converter Sub-Ranges". The sub-ranges are numbered from 1 through 7. The even-numbered sub-range is offset by 0.64V from its preceding odd-numbered sub-range, thereby overlapping the upper half of the preceding sub-range. The algorithm to select a sub-range for a given analog signal is discussed later but the basic principle to achieve 10-bit resolution is the same as before.

First, an 8-bit result of the A/D conversion is performed in the 5.12V range and then a decision is made as to which of the seven sub-ranges the signal belongs. The second A/D conversion in the selected sub-range then gives the lower 8 bits of the final 10-bit result in one of the two following ways. If an odd-numbered sub-range is selected, then the 8-bit result directly gives the lower 8 bits of the 10-bit result. However, if an even-numbered sub-range is selected, which is offset by 0.64V from the odd-numbered sub-range, an 80H (digital equivalent of FSR/2 = 1.28V/2 = 0.64V) is added to the digital result to compensate for this offset. Any carry from this addition will also modify the value of the two MSBs of the final 10-bit result.

Algorithm

"Figure 24 A/D Converter Sub-Ranges" graphically represents the assignment of the DAPR register in the narrow range. The left nibble of the 8-bit result obtained from the A/D converter in the full range is used as a pointer in the DAPR look-up table to get a value for DAPR register for the second conversion in the narrow range. The second bit of the selected DAPR value for the narrow range decides whether an even or odd-numbered sub-range is selected. When this bit is '0' (for an odd-numbered sub-range), the third and fourth bits of the DAPR value become the two MSBs of the final 10-bit result. When an even-numbered sub-range is selected, an 80H is added to the A/D conversion result obtained in the narrow range. Again, the third and fourth bits of the selected DAPR value corresponds to the two MSBs of the final 10-bit result. The carry generated by the addition of 80H to the digital result is added to these bits to determine the two MSBs of the final 10-bit result. These two bits concatenated with the 8-bit result obtained by programming the DAPR register in the narrow range gives the final 10-bit result.

Figure 24
A/D Converter Sub-Ranges



8

System Design Hints

The recommended design for use of the SAB 80515/535 is shown in "Figure 25 Capacitors for the SAB 80515/535". Standard design rules such as low impedance wire and minimized connector lengths will ensure low system resistance and inductance. In addition, there are several key points to consider for the most accurate and stable performance of the A/D converter.

First, a dedicated reference voltage for the A/D converter is highly desirable. Since V_{CC} is typically noisy and unstable, the V_{CC} line is unsuitable as an analog reference. In addition, many standard voltage regulators are not stable enough to meet the requirements of the SAB 80515, if maximum accuracy is desired. Strict adherence to the VAREF specification is recommended.

Additionally, separate digital and analog ground lines (tied near the source) are recommended to provide a smooth analog GND. However, if there is a large voltage differential between the ground lines, it is desirable to tie the ground lines close to the chip. By doing so, the noise in the digital ground will be reflected in the analog ground, resulting in a loss of accuracy.

Lastly, "Figure 25 Capacitors for the SAB 80515/535" is a schematic for the NMOS implementation of the SAB 80515/535. In the CMOS implementation of this device, the V_{BB} pin becomes another V_{CC} pin. Therefore, when designing for both NMOS and CMOS devices, it will be necessary to bring a V_{CC} line close to the V_{BB} pin, to be connected via a jumper pin, as required by the CMOS device. In addition, when using CMOS devices, the long capacitor connected between V_{SS} and V_{CC} (pin 68) may be removed so that only one short 100 nF capacitor remains between V_{SS} and V_{CC} (pins 37 and 38).

Software Hints

A typical software routine to achieve 10 bit resolution is given in Appendix A. To reduce noise, the address lines should remain as quiescent as possible to minimize the power draw from the VCC. Therefore, it is recommended that other tasks which change the address lines are not initiated while waiting for the end of conversion.

In addition, the Jump if Busy (JB BSY,addr) instruction should be placed at the beginning of a page in the program memory where the upper 14 address lines don't change. An example is shown:

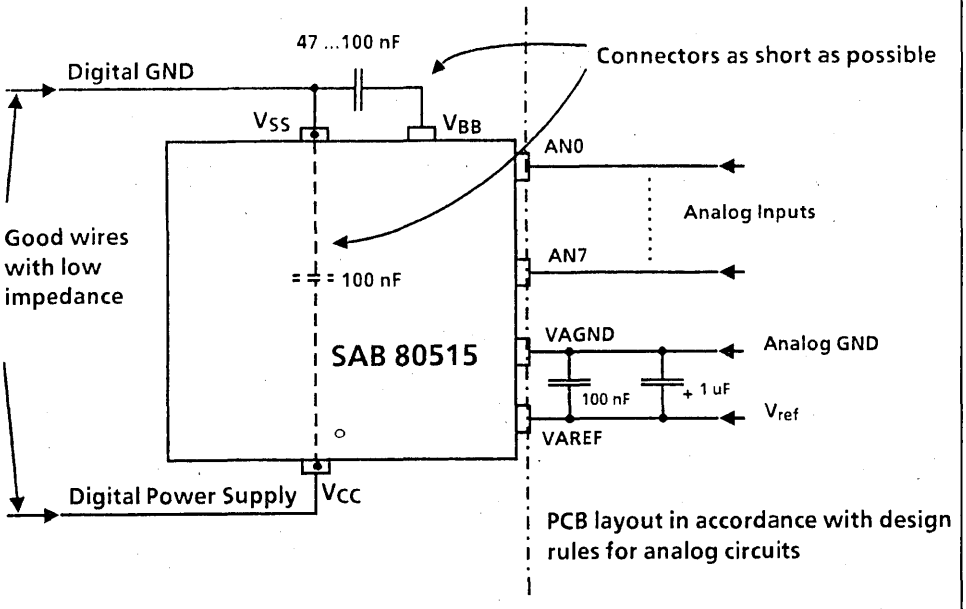
Good Example:	Bad Example:
Addr.	Addr.
XX X0H JB	0F FFH JB
XX X1H BSY	10 00H BSY
XX X2H rel. address	10 01H rel. address

To ensure highest accuracy, self-calibration routines should be performed each time before the A/D converter is used, to monitor the offset error at baseline, and gain error of the device. A simple test for offset may be performed by inputting an analog zero, and checking the digital result. Likewise, gain error may be determined by feeding known voltages into the A/D converter and comparing the digital outputs.

References

1. Siemens SAB 80515/535 User's Manual.

Figure 25
Capacitors for the SAB 80515/535



8

Appendix A

```
                        1   $MOD515
                        2   $PAGELENGTH(80)
                        3
                        4   ;*******************************************************************
                        5   ;*                                                                 *
                        6   ;*  Subroutine_Name: AD_CONV                                       *
                        7   ;*  Function: This subroutine performs an A/D conversion on the ch-*
                        8   ;*            annel selected by the variable CHAN_SEL and at the   *
                        9   ;*            end of the conversion saves a 10-bit result in locat-*
                       10   ;*            ions AD_VALUE_L (lower eight bits) and AD_VALUE_H    *
                       11   ;*            (has the two MSBs).                                 *
                       12   ;*                                                                 *
                       13   ;*******************************************************************
0020                   14           DSEG    AT      20H
0020                   15   AD_VALUE_L:     DS      1       ;CONTAINS LOWER 8 BITS
0021                   16   AD_VALUE_H:     DS      1       ;CONTAINS TWO MSBs OF 10-BIT RESULT
                       17
0000                   18           CSEG    AT      0H
0000                   19   AD_CONV:
0000 C0E0              20           PUSH    ACC
0002 C0D0              21           PUSH    PSW
0004 53D8F8            22           ANL     ADCON,#11111000B
0007 E500              23           MOV     A,CHAN_SEL
0009 42D8              24           ORL     ADCON,A
000B 75DA00            25           MOV     DAPR,#00H
000E 20DCFD            26           JB      BSY,$           ;FIRST CONVERSION
0011 E5D9              27           MOV     A,ADDAT
0013 54F0              28           ANL     A,#0F0H
0015 C4                29           SWAP    A               ;A = FIRST MEAS./16
0016 241E              30           ADD     A,#REFER_OFFSET
0018 83                31           MOVC    A,@A+PC         ;LOOK-UP TABLE FOR SECOND DAPR
0019                   32   REFERENCE_HELP_LABEL:
0019 F5DA              33           MOV     DAPR,A
001B 20DCFD            34           JB      BSY,$           ;SECOND CONVERSION
001E AED9              35           MOV     R6,ADDAT        ;RESULT=SECOND CONVERSION
0020 03                36           RR      A               ;A=(DAPR0)000 0(DAPR3)(DAPR2)(DAPR1)
0021 5407              37           ANL     A,#07H          ;A=0000 0(DAPR3)(DAPR2)(DAPR1)
0023 FF                38           MOV     R7,A
0024 5401              39           ANL     A,#01H          ;A=0000 000(DAPR1)
0026 03                40           RR      A               ;A=(DAPR1)000 0000
0027 2E                41           ADD     A,R6
0028 F520              42           MOV     AD_VALUE_L,A
002A EF                43           MOV     A,R7
002B 03                44           RR      A
002C 5403              45           ANL     A,#03H          ;A=0000 00(DAPR3)(DAPR2)
002E 3400              46           ADDC    A,#00H
0030 F521              47           MOV     AD_VALUE_H,A
0032 D0D0              48           POP     PSW
0034 D0E0              49           POP     ACC
0036 22                50           RET
                       51
0037                   52   REFERENCE_TABLE:
0037 40                53           DB      40H
0038 40                54           DB      40H
0039 40                55           DB      40H
003A 62                56           DB      62H
003B 62                57           DB      62H
003C 84                58           DB      84H
003D 84                59           DB      84H
003E A6                60           DB      0A6H
003F A6                61           DB      0A6H
0040 C8                62           DB      0C8H
0041 C8                63           DB      0C8H
0042 EA                64           DB      0EAH
0043 EA                65           DB      0EAH
0044 0C                66           DB      00CH
0045 0C                67           DB      00CH
0046 0C                68           DB      00CH
                       69
  001E                 70   REFER_OFFSET    EQU     REFERENCE_TABLE-REFERENCE_HELP_LABEL
  0000                 71   CHAN_SEL        EQU     00H
                       72
                       73           END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

Bidirectional, Speed Regulated Moving Message Display by Using the Timer 2 & 8-Bit A/D Converter of the SAB 80515/80535

SAB 80515/80535

8

Application Note

Introduction

This application note introduces the user to one of the features of Timer 2 and A/D converter of the SAB 80515/535. Included in this application note is a description of both the software and hardware implementations of the SAB 80515/535 to use its Timer 2 and 8-bit A/D converter for the bidirectional, speed regulated moving message display. The program listing demonstrates how the Timer 2 and the 8-bit A/D converter of the SAB 80515/535 can be combined to generate time delays controlled by analog levels. The hardware circuitry shows an interface of the SAB 80515/535 with a simulated analog input, a 2 kbyte EPROM, and intelligent display chips of Siemens used in memory mapped I/O scheme.

The SAB 80515/535 microcontroller with on-chip A/D converter and a 16-bit Timer (Timer 2) with reload capability offers a solution which can be applied to a wide range of industrial applications. These applications vary from analog controlled digital delays to controlled frequency converters for pulse width modulation.

In the present application example, the above features of the SAB 80515/535 are used in conjunction to generate the software delays. The software delay results in varying the voltage level of the analog signal applied to the A/D converter of the SAB 80515/535.

A/D Converter

The SAB 80515/535 provides an 8-bit A/D converter with eight multiplexed analog input channels on-chip. In addition, the A/D converter has a sample and hold circuit and offers the feature of software-programmable reference voltages. For the conversion, the method of successive approximation with a capacitor network is used.

"Figure 26 Block Diagram of A/D Converter" shows a block diagram of the A/D converter. There are three user-accessible special function registers:

- ADCON (A/D converter control register)
- ADDAT (A/D converter data register)
- DAPR (D/A converter program register) for the programmable reference voltages.

Special function register ADCON is used to select one of the eight analog input channels to be converted, to specify a single or continuous conversion, and to check the status bit BSY which signals whether a conversion is in progress or not.

The special function register ADDAT holds the converted digital 8-bit data result. The data remains in ADDAT until it is overwritten by the next converted data. The new converted value will appear in ADDAT in the 15th machine cycle after a conversion has been started. ADDAT can be read and written to under software control. If the A/D converter of the SAB 80515/535 is not used, register ADDAT can be used as an additional general-purpose register.

The special function register DAPR is provided for programming the internal reference voltages IVAREF and IVAGND. In the present application DAPR holds a value of 00H. For this value of DAPR, IVAREF and IVAGND are same as VAREF and VAGND respectively.

A/D Conversion

A conversion is started by writing to the special function register DAPR. A "Write-to-DAPR" will start a new conversion even if a conversion is currently in progress. The conversion begins with the next machine cycle. The busy flag BSY will be set in the same machine cycle as the "write-to-DAPR" operation occurs. If the value written to DAPR is 00H, meaning that no adjustment of the internal reference voltages is desired, the conversion needs 15 machine cycles to be completed. Thus, the conversion time is 15 µs for 12 MHz oscillator frequency.

After a conversion has been started by writing into the special function register

DAPR, the analog voltage at the selected input channel is sampled for 5 machine cycles (5 µs at 12 MHz oscillator frequency), which will then be held at the sampled level for the rest of the conversion time. The external analog source must be strong enough to source the current in order to load the sample & hold capacitance, being 25 pF, within those 5 machine cycles.

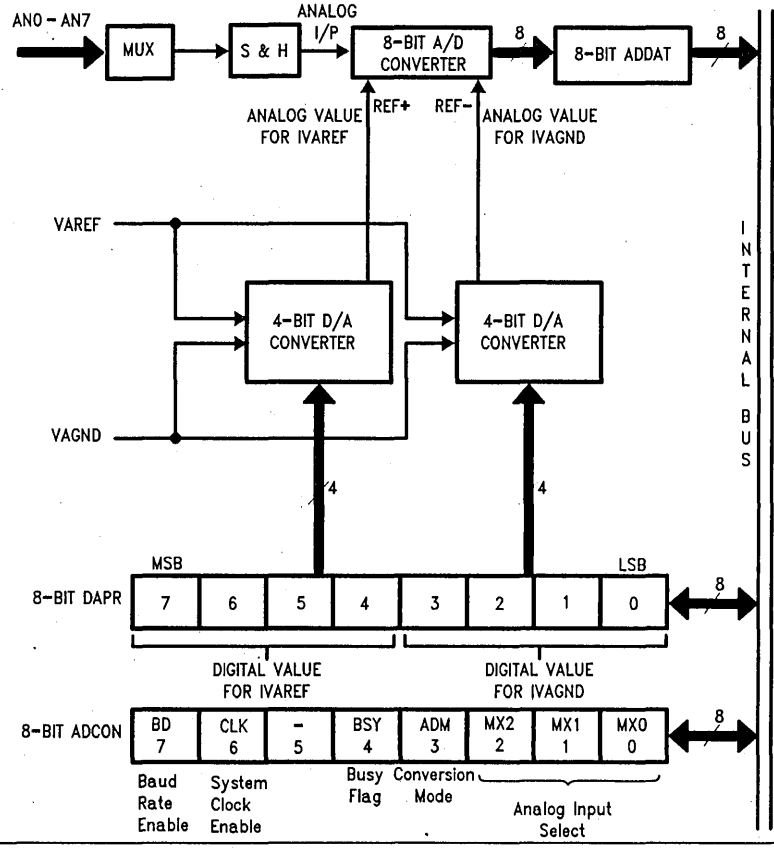
Conversion of the sampled analog voltage takes place between the 6th and 15th machine cycle after sampling has been completed. In the 15th machine cycle the converted result is moved to ADDAT.

Timer 2

The SAB 80515 has three 16-bit Timer/Counters: Timer 0, Timer 1 and Timer 2. These Timers can be configured to operate either as timers or event counters. Timer 2 is the time base of the programmable Timer/Counter Register Array (PTRA) unit. In addition to the operational modes "Timer" or "counter", Timer 2, being the time base for the PTRA unit, provides the features of:

- 16-bit reload
- 16-bit compare
- 16-bit capture

Figure 26
Block Diagram of A/D Converter



The reload mode of Timer 2 is used in this application to generate software delays. For explanation of the other modes please refer to the user's manual.

Reload

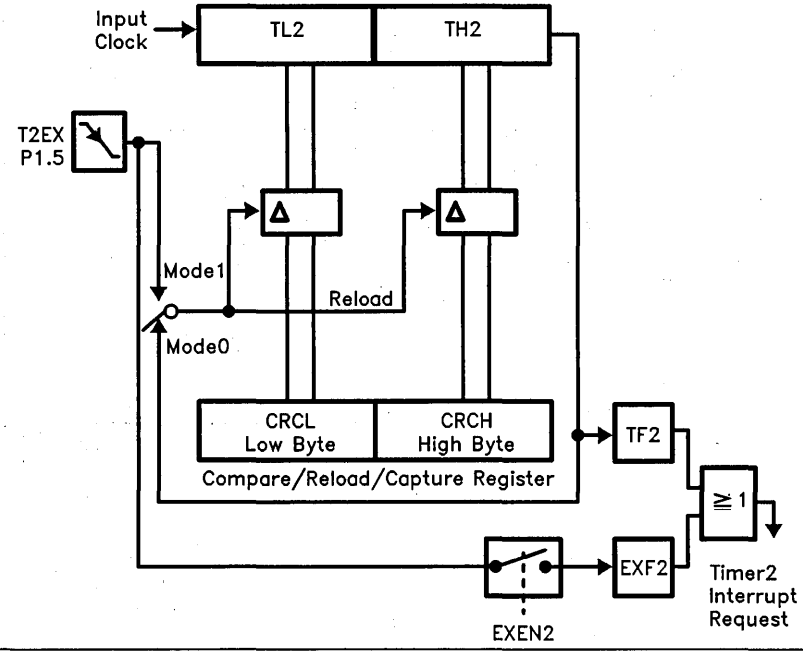
The reload mode for Timer 2 is selected by bits T2R0 and T2R1 in special function register T2CON as illustrated in Table 2. In mode 0, when Timer 2 rolls over from all 1s to all 0s, it not only sets TF2 but also causes the Timer 2 registers to be loaded with the 16-bit value in the CRC (compare/reload/capture) register which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000H.

"Figure 27 Timer 2 in Reload Mode" shows a functional diagram of the Timer 2 reload modes.

Table 2 Timer 2 Reload Mode Selection

T2R1	T2R0	Mode
0	X	Reload Disabled
1	0	Mode 0: Auto-Reload upon Timer 2 Overflow (TF2)
1	1	Mode 1: Reload upon Falling Edge at Pin T2EX/P1.5

Figure 27
Timer 2 in Reload Mode



PD2435

PD2435 is a CMOS
4-character
5 x 7 dot matrix
alphanumeric
programmable display with ROM to decode 96 ASCII alphanumeric characters and enough RAM to store the display's complete four digit ASCII message with software programmable attributes. The CMOS IC incorporates special interface control circuitry to allow the user to control the module as a fully supported microprocessor peripheral.

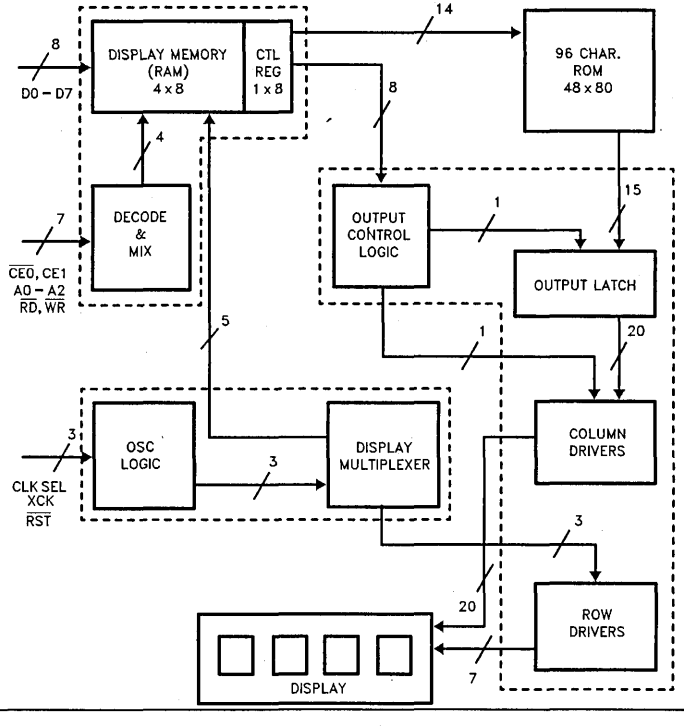
Microprocessor Interface

The interface to the microprocessor is through the address lines (A0-A2), the data bus (D0-D7), two chip select lines ($\overline{CE0}$, CE1), and ($\overline{RD}$) and ($\overline{WR}$) lines. The $\overline{CE0}$ should be held low when executing a read or write operation. The read and write lines are both active low. A valid write will enable the data as input lines.

Programming the PD2435

There are five registers within the PD2435. Four of the registers are used to hold the ASCII code of the four display characters. The fifth register is the Control Word, which is used to blink, blank, clear or dim the entire display to change the presentation (attributes) of individual characters.

Figure 28
PD2435 Block Diagram
The PD2435 block diagram shows the major blocks and internal registers.



Application

The speed regulated moving message display is an example where a digitized value of the controlling analog signal is used to compute a reload value for the Timer 2. The Timer 2 is operated in mode 0 where this reload value becomes a starting point for the Timer to count up. On overflow the Timer automatically takes the restart value for counting from reload register CRC. While the Timer is counting up, a new reload value is computed using the present A/D value.

Hardware

The circuit used in this application offers the advantage in requiring a minimum of components. The single chip microcomputer SAB 80535 operates in conjunction with four alphanumeric programmable display chips PD 2435 to form a 16-digit long display.

The ASCII-coded data is transferred from the SAB 80535 to the display ICs via the data port P0 and using the control signal WR (P3.6) of the SAB 80535. The address pins from the ports P0 and P2 of the SAB 80535 are used to address the EPROM as well as the display chips in a memory-mapped I/O scheme. The display chips are addressed as memory locations with the following addresses.

Display Chip	Control Register Address	Digits Address
1	1000H	1004H - 1007H
2	2000H	2004H - 2007H
3	4000H	4004H - 4007H
4	8000H	8004H - 8007H

A push button is interfaced to port P3.2 of the SAB 80535 to provide an external interrupt to the microcontroller.

Firmware Description

Besides controlling the speed of the moving message, there is a provision to interrupt the moving message to roll it backwards until the beginning of the message. The microcontroller reads the code and the message to display from an EPROM 2716A interfaced to the ports P0 and P2 of the SAB 80535. A virtual image of the message is created in the internal RAM of the SAB 80535. Four display chips PD2435 are interfaced to the SAB 80535 in a memory-mapped scheme and can be addressed as external memory to the SAB 80535. The virtual image of the message in internal RAM of the SAB 80535 is used to manipulate data to be displayed on the display chips. The internal RAM used for the display can be viewed as an area divided into two portions:

1. For active display
2. As a data buffer

The active display area is the replica of the data being displayed on the display chips. In this case the 16-digit display would need 16 RAM locations which correspond to 16 digits currently being displayed. The data buffer contains the rest of the message which is not being displayed. The message is shifted character by character in the RAM area. When the message on the display moves from right to left, the RAM buffer acts in "First In First Out" mode and when the message on the display moves from left to right, the data to the display from the microcontroller RAM buffer is supplied in the "Last In First Out" scheme.

Between display of every character there is a software delay which depends upon the level of the analog signal supplied to the AN0 pin of the SAB 80535. The external interrupt 0 (at port P3.2) is used to interrupt the microcontroller to inform that the message needs to be scrolled backwards. On getting this interrupt the software sets the flag bit 0 which remains set until the message is scrolled back to the beginning of the message.

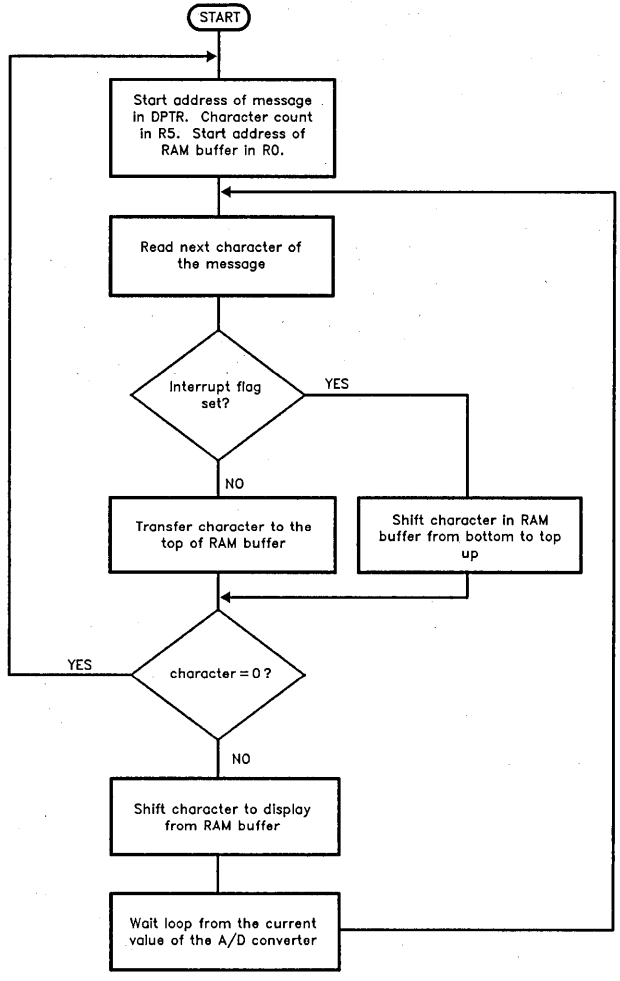
List of Components

Name	Number
SAB 80535	1
2716A	1
PD2435	4
12 MHz Crystal	1
74LS373	1
22 pF Capacitors	2
100 nF Capacitor	1
4.7 f Capacitor	1
1k Resistor	1
0k Pot	1

Reference Material for ICs

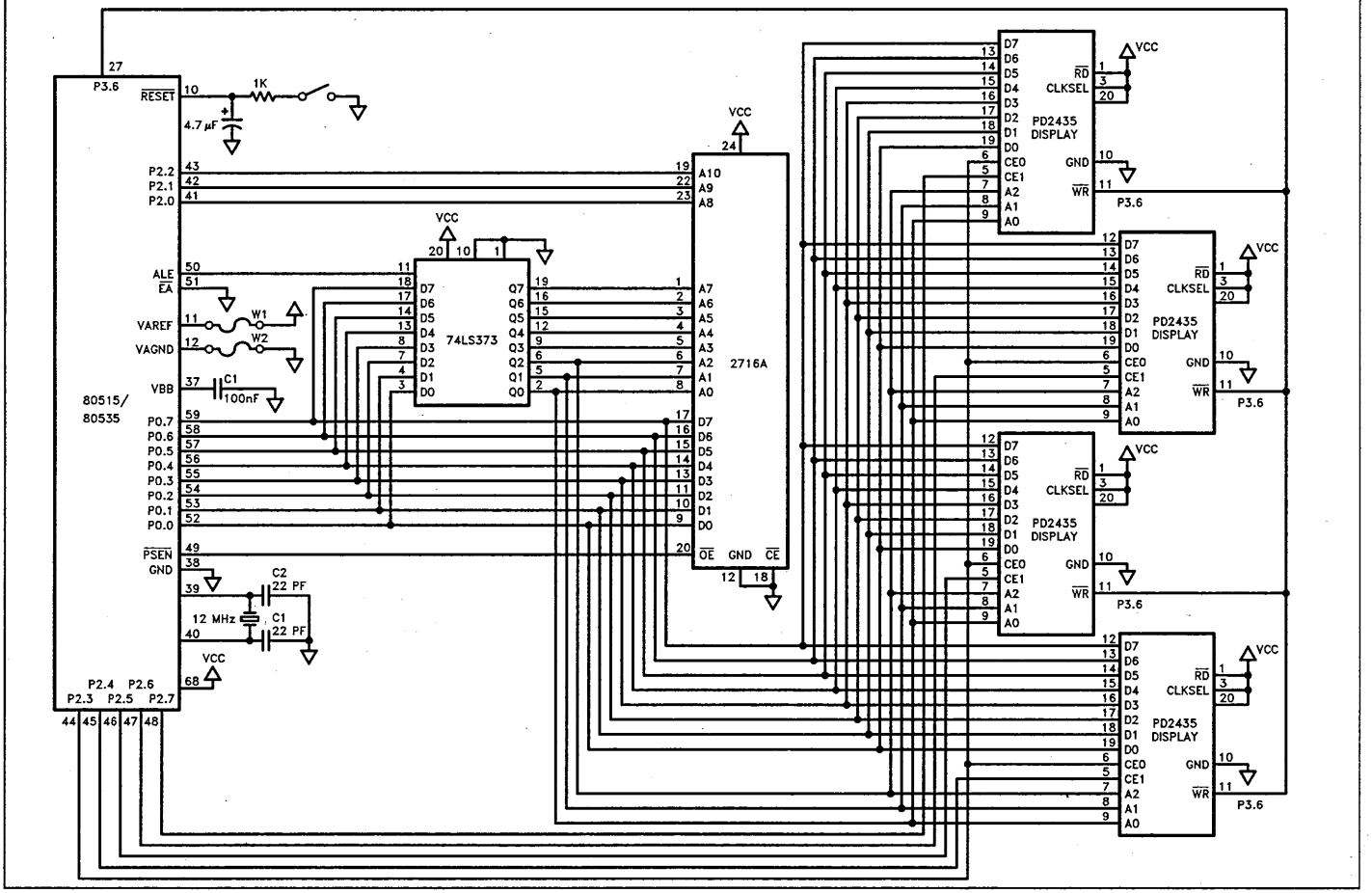
1. SAB 80515/80535 User's Manual.
2. PD2435 Data-Sheet or Optoelectronic Data Book (1987/88).

Figure 29
Program Flow-Chart



8

Figure 30
Interface Circuit



Program Listing

```
UDISP'PD 2435 DISPLAY PROGRAM'
                    1     $TITLE ('PD 2435 DISPLAY PROGRAM')
                    2     $MOD515
                    3     $NOSYMBOLS
                    4
....                5     CSEG
                    6     $DEBUG
                    7
                    8
0000                9           ORG   OOH
                   10
0000 02000C        11           LJMP  BEGIN            ;Jump on reset
                   12
                   13           ;*************************************************************
                   14           ;*    This is the interrupt subroutine for INTO. This        *
                   15           ;*    is used to set a flag which then indicates that        *
                   16           ;*    the message needs to be rolled back.                   *
                   17           ;*************************************************************
                   18
0003               19           ORG   03H
                   20
0003 C0E0          21           PUSH  ACC
0005 D2D5          22           SETB  F0               ;Set flag for external interrupt
0007 D0E0          23           POP   ACC
0009 C289          24           CLR   IEO
000B 32            25           RETI
                   26
                   27           ;*************************************************************
                   28           ;*    MAIN PROGRAM                                           *
                   29           ;*************************************************************
                   30
000C D2B2          31     BEGIN:SETB  P3.2             ;Set bit for INTO
000E 758110        32           MOV   SP,#10H
0011 750800        33           MOV   ADCON,#00H       ;Select analog channel 0
                   34
0014 C2D5          35     OPTS: CLR   F0               ;Clear flag 0
0016 7800          36           MOV   R3,#00H          ;Character pointerin the message
0018 79FF          37           MOV   R1,#00H          :R1 used as flag
001A 90F000        38           MOV   DPTR,#0F000H     ;Control register of all displays
001D 7403          39           MOV   A,#03H           ;Control word for display
001F F0            40           MOVX  @R0PTR,A
0020 9000C2        41           MOV   DPTR,#(TEXT - 1) ;Beginning of text
0023 7820          42           MOV   R0,#20H          ;Internal RAM location
0025 7D65          43           MOV   R5,#101          ;A count for 101 characters
0027 7420          44           MOV   A,#20H           ;ASCII for space
0029 F6            45     BLANK:MOV   @RO,A            ;Fill all location with blank
002A 08            46           INC   RO
002B DDFC          47           DJNZ  R5, BLANK
                   48
002D 12006C        49     SHIF: CALL  NEXTC            ;Read the next character
0030 20D501        50           JB    F0,TEMP          ;Check if the interrupt was raised
0033 0B            51           INC   R3               ;If no interrupt
0034 7D65          52     TEMP: MOV   R5,#101          ;Character count in message
0036 7820          53           MOV   R0,#20H          ;RAM location 20H
0038 20D506        54           JB    F0,REV0
003B C6            55     SHFT: XCH   A,@RO            ;If no interrupt
003C 08            56           INC   R0               ;Add the character
003D DDFC          57           DJNZ  R5,SHFT          ;To the top of the RAM buffer
003F 0158          58           AJMP  CONT0
0041 7421          59     REV0: MOV   A,#21H.          ;If there is an interrupt
0043 2B            60           ADD   A, R3            ;Offset for the RAM buffer
0044 F8            61           MOV   R0,A             ;Pointer in the RAM buffer
0045 7600          62           MOV   R0, #00H         ;Displayed so far
0047 7820          63           MOV   RO, #20H         ;Beginning of the RAM buffer
0049 E6            64           MOV   A,@RO            ;Read the character
004A C0E0          65           PUSH  ACC              ;Save it
004C 08            66     AGAIN:INC   R0               ;Next location in RAM buffer
004D E6            67           MOV   A,@R0            ;Read the next character
004E 18            68           DEC   R0               ;Back to first character
004F F6            69           MOV   @R0, A           ;Replace with second character
0050 08            70           INC   R0               ;Process repeats
0051 DDF9          71           DJNZ  R5, AGAIN        ;Moving character backwards
0053 08            72           INC   R0
0054 7600          73           MOV   @R0,#00H         ;End of character buffer
0056 D0E0          74           POP   ACC              ;Restore character
0058 7820          75     CONT0:MOV   R0,#20H          ;Beginning of character buffer
005A E9            76           MOV   A, R1            ;Check if end of character buffer
0058 6087          77           JZ    OPTS
005D 120071        78           CALL  OUTC
0060 C2AF          79           CLR   IEN0.7           ;Disable interrupt
0062 1200A4        80           CALL  WAITA            ;Before delay
0065 75A881        81           MOV   IEN0,#81H        ;Enable interrupt
0068 D288          82           SETB  IT0              ;INTO control bit
```

8

Program Listing

```
006A 012D          83          AJMP    SHIF
                   84
                   85          ;*************************************************
                   86          ;* The routine moves a character of the message to ACC. *
                   87          ;*************************************************
                   88
006C A3            89  NEXTC:  INC     DPTR
006D 7400          90          MOV     A,#0
006F 93            91          MOVC    A,@A+DPTR               ;Move the character to Acc.
0070 22            92          RET
                   93
                   94          ;*************************************************
                   95          ;*   This routine displays and moves a character over   *
                   96          ;*    the four digits of the PD2435 and then repeats    *
                   97          ;*          for the next display chip and so on.        *
                   98          ;*************************************************
                   99
0071 C0E0          100 OUTC:   PUSH    ACC
0073 C082          101         PUSH    DPL
0075 C083          102         PUSH    DPH
0077 7A04          103         MOV     R2,#4               ;For four digits(0 to 3) in a chip
0079 901004        104         MOV     DPTR,#1004H         ;Digit 0 in first display chip
007C 120098        105         CALL    OUTC0
007F 902004        106         MOV     DPTR,#2004H         ;Digit 0 in second display chip
0082 120098        107         CALL    OUTC0
0085 904004        108         MOV     DPTR,#4004H         ;Digit 0 in third display chip
0088 120098        109         CALL    OUTC0
008B 908004        110         MOV     DPTR,#8004H         ;Digit 0 in fourth display chip
008E 120098        111         CALL    OUTC0
0091 D083          112         POP     DPH
0093 D082          113         POP     DPL
0095 D0E0          114         POP     ACC
0097 22            115         RET
                   116
                   117         ;*************************************************
                   118         ;* This is a nested subroutine. It moves a nonzero hex  *
                   119         ;*  value (ASCII) from left in right of the four digit  *
                   120         ;*                          display.                   *
                   121         ;*************************************************
                   122
0098 E6            123 OUTC0:  MOV     A,@R0
0099 6007          124         JZ      FIN
009B F0            125         MOVX    @DPTR,A
009C 08            126         INC     R0
009D A3            127         INC     DPTR
009E DAF8          128         DJNZ    R2,OUTC0
00A0 7A04          129         MOV     R2,#4
00A2 F9            130 FIN:    MOV     R1,A
00A3 22            131         RET
                   132
                   133         ;*************************************************
                   134         ;*  This subroutine generates the software delay. The   *
                   135         ;*  delay is generated by the timer 2. The start count  *
                   136         ;*  of the timer 2 is computed from the present value   *
                   137         ;*                of the A/D converter.                *
                   138         ;*************************************************
                   139
00A4 7E03          140 WAITA:  MOV     R6,#03H
00A6 7D10          141 WAITB:  MOV     R5,#10H
00A8 75DA00        142 WAITC:  MOV     DAPR,#00H
00AB E5D9          143         MOV     A,ADDAT
00AD 75F0FF        144         MOV     B,#255              ;For computing reload value
00B0 A4            145         MUL     AB                  ;Reload value is computed
00B1 F5CA          146         MOV     CRCL,A              ;Load the reload value low
00B3 85F0CB        147         MOV     CRCH,B              ;Load the reload value high
00B6 75C811        148         MOV     T2CON,#11H
00B9 10C602        149 WAITD:  JBC     TF2,WAITE
00BC 01B9          150         AJMP    WAITD
00BE DDE8          151 WAITE:  DJNZ    R5,WAITC
00C0 DEE4          152         DJNZ    R6,WAITB
00C2 22            153         RET
                   154
                   155         ;*************************************************
                   156         ;*                       MESSAGE                 *
                   157         ;*************************************************
                   158
00C3 20202020      159 TEXT:   DB      '               '
00C7 20202020
00CB 20202020
00CF 20202020
00D3 53494540      160         DB      'SIEMENS MICROCONTROLLER SAB 80515/535'
00D7 454E5320
00DB 4D494352
00DF 4F434F4E
```

Program Listing

```
00E3 54524F4C
00E7 4C455220
00EB 53414220
00EF 38303531
00F3 352F3533
00F7 35
00F8 20202020          161             DB      '               SAB 80515/535               ',0
00FC 20202020
0100 20202020
0104 53414220
0108 38303531
010C 352F3533
0110 35202020
0114 20202020
0118 20202020
011C 20202020
0120 00
                       162             END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

8

Heating and Air Conditioning Control in Cars with the Microcontroller SAB 80515/80535

SAB 80515/80535

8

Application Note

Introduction

The heating and air conditioning unit in the car should provide the driver with conditions of comfortable temperature, fresh air flow, defogged and defrosted windows, low energy consumption and easy operation.

The currently mass-produced systems meet these requirements to a limited degree. In response, Siemens began to develop µC-controlled installations and equipped several test vehicles with various systems which proved to be highly satisfactory.

Its performance-oriented processor and flexible on-chip periphery (e.g. analog-to-digital converter, timer function, large number of inputs/outputs) make the SAB 80515 especially suitable for this type of application. The majority of the peripheral components are fully utilized in this application.

The temperature in the car reaches its nominal value and is kept constant by means of a two-stage mixing valve control. A rise in outside temperature automatically activates the compressor of the air conditioning unit. The air entering the car is distributed upwards and downwards by an electrically controlled valve, depending on the temperature of the air. The optimal speed is also determined by the microcontroller as a function of the various input valves. The electronics also control actuators such as air circulation and water valve.

The device is operated by means of several keys. An LED display indicates the nominal or outside temperature. Individual LEDs indicate special conditions which can be selected by the user independent of the automatic functions.

Control Elements and Sensors (Figure 31)

The temperature inside the car is controlled in accordance with a selectable nominal value. The most important actuators for this purpose are the mixing valve and the compressor of the air conditioning unit. The mixing valve determines which part of the air entering the passenger area must pass through the heat exchanger of the heating unit. The valve can be fine-tuned by the microcomputer.

If there is no heating requirement, the water flow to the heat exchanger is stopped by a digitally controlled valve. As a result, the temperature is further reduced in the summertime.

Depending on the output of the system, the air conditioning unit ensures that the nominal value of the temperature inside the car is obtained despite higher outside temperatures. The compressor of the air conditioning unit is enabled/disabled by the microcontroller.

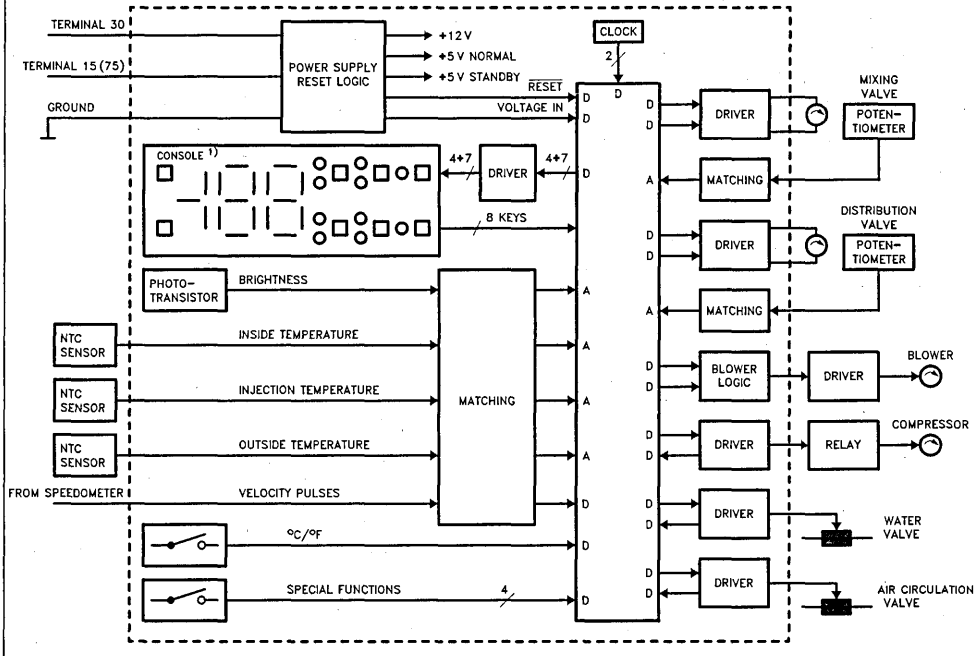
In addition, the electronics influences the distribution of the temperature layers inside the car by a nearly stepless adjustable distribution valve. The valve determines whether the air is to be moved towards the roof or the floor of the car. Through this type of control, the air close to the roof of the car should be at a temperature lower than that close to the floor.

The fresh air flow is also electronically controlled. Depending on the different temperatures and the road speed of the car, the microcontroller computes the optimal speed for the blower, which can be changed almost continuously.

On the basis of the temperature conditions, the processor determines the requirement for fresh air flow or circulation of the air inside the car. The corresponding valve is digitally controlled.

In order to achieve the described functions, the system uses three sensors to measure the temperature inside the car, the temperature of the air entering the car as well as the outside temperature. A speed sensor informs the processor about the car's current road speed.

Figure 31
Block Diagram



D_ Digital Signal
A_ Analog Signal
1See Figure 32 for Detail

8

Operating and Displaying Unit (Figure 32)

A display optionally indicates the nominal or outside temperature. Functions which deviate from standard operations are indicated by LEDs located next to the keys. The brightness level of the display and the LEDs is controlled by the processor in accordance with the ambient light measured by a phototransistor

With the aid of eight keys the following functions can be performed (Figure 32):

S1, S2: Changes in nominal temperature ("+" and "-" key)
Through instantaneous pressure or sustained pressure on the key, the nominal value can be changed in 1C/1F steps, that is from 16C--30C/60F--86F. In addition, the extreme values "LO" and "HI" can be set, and the mixing valve will continue to remain in the minimal (cold) or maximal (hot) position.

By depressing the keys, the following functions allow the user to switch over from normal (automatic) setting to one, two or three fixed values. After a fixed value has been selected, the corresponding LED or a combination of two LEDs lights up.

S3: Distribution key for switching the air distribution to automatic, only upwards, in the center (upwards and downwards, both LEDs light up) or only downwards.

S4: Blower key for switching the blower to automatic, full speed, half speed (both LEDs light up) and "OFF".

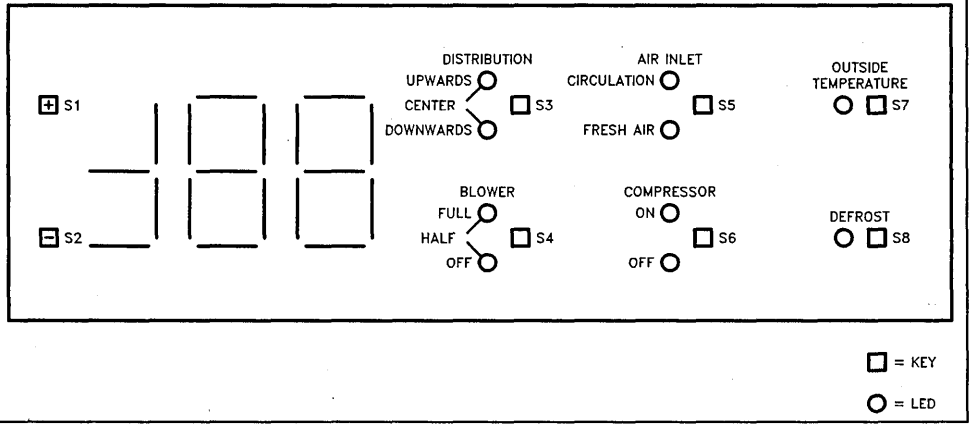
S5: Air supply key for switching the air supply to automatic, air circulation, or fresh air.

S6: Compressor key for switching the compressor to automatic, "ON" and "OFF".

S7: Outside temperature key for switching the display to nominal temperature (standard function) or outside temperature. The outside temperature is displayed in 1 °C/1 °F steps at a range between -40 °C/-40 °F and +60 °C/+140 °F.

S8: Defrost key for switching the device from its previous function (standard) to the defrost function.

Figure 32
Console



The windows of the car can thus rapidly be defrosted and defogged.

The setting elements take the following positions during the defrost function which has priority over all other settings:

Mixing valve:	Max. Heating
Distributor valve:	Only Upwards
Blower:	Max. Number of Speed
Air supply:	Fresh Air
Compressor:	ON
Water Valve:	ON

As long as the defrost function is in operation, the remaining functions (with the exception of display switch-over for the temperature) cannot be operated. The corresponding LEDs are not driven. After the defrost status is finished, the previous functions apply again.

The nominal temperature as well as set fixed values are saved after the car ignition has been turned off. During initial start-up or after reconnecting the battery, a mean nominal temperature (22 °C/71 °F) is set and the automatic functions apply.

Major Hardware and Software Functions

Voltage Supply, Reset Logic (Figure 33)

Since various conditions—e.g. nominal temperature—are to be stored after the ignition has been turned off, a continuous 5V supply is required which is supplied directly by the battery (terminal 30). A diode/ capacitor combination protects against reversed polarity and extreme voltage peaks. The voltage regulator, which is used, is characterized by a lower power dissipation and continues to operate during low input voltages. The SAB 80515 stores the data; 40 bytes of its internal RAM are saved during standby operation with a typ. supply current of 1 mA.

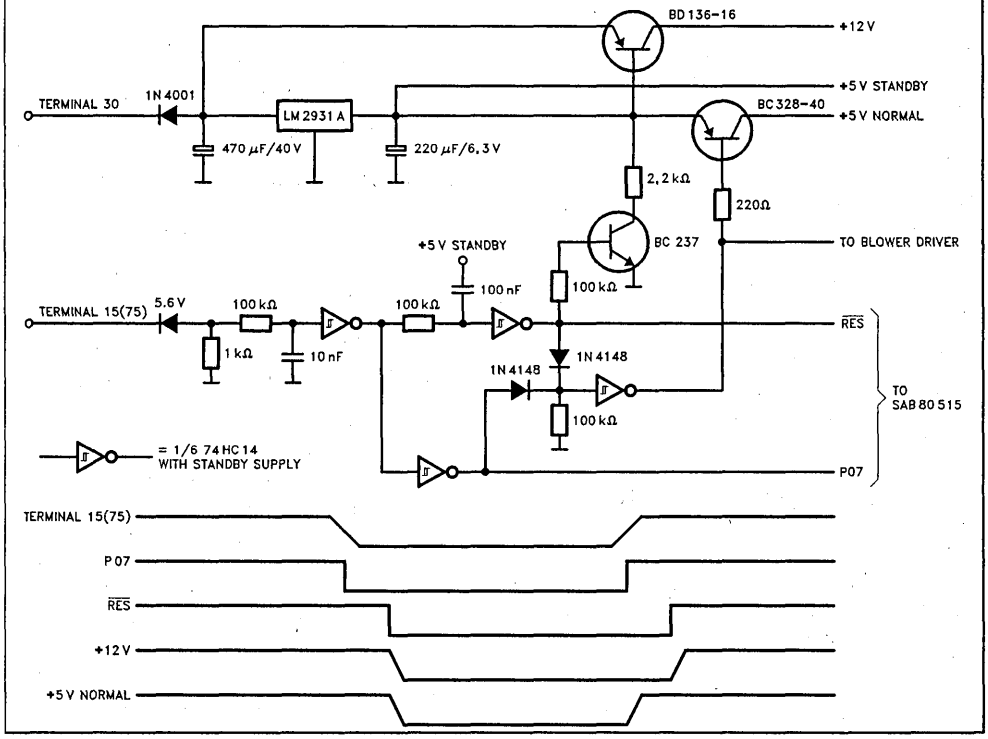
When the car ignition is turned on, the normal 5V operating voltage as well as a filtered 12V voltage are available for supplying the drivers. The criterion for the connection of these voltages is the status of terminal 15. Preferably terminal 75 should be used if included in the device, since it will remain disabled while the car is started.

When the ignition is turned off, the processor receives a signal via P07 prior to the drop in voltage of the standard 5V supply. Subsequently, the processor will wait for the reset signal which immediately precedes the voltage switch off. After the ignition has been turned on again, RES continues to be in "L" to reset the SAB 80515. This time period required for reset is ensured by an RC network in combination with diodes and Schmitt triggers.

Clock Supply

The SAB 80515 oscillator resonates at a frequency of 6 MHz by means of a ceramic resonator. The result is an instruction cycle time of 2 µs.

Figure 33
Voltage Supply, Reset Logic



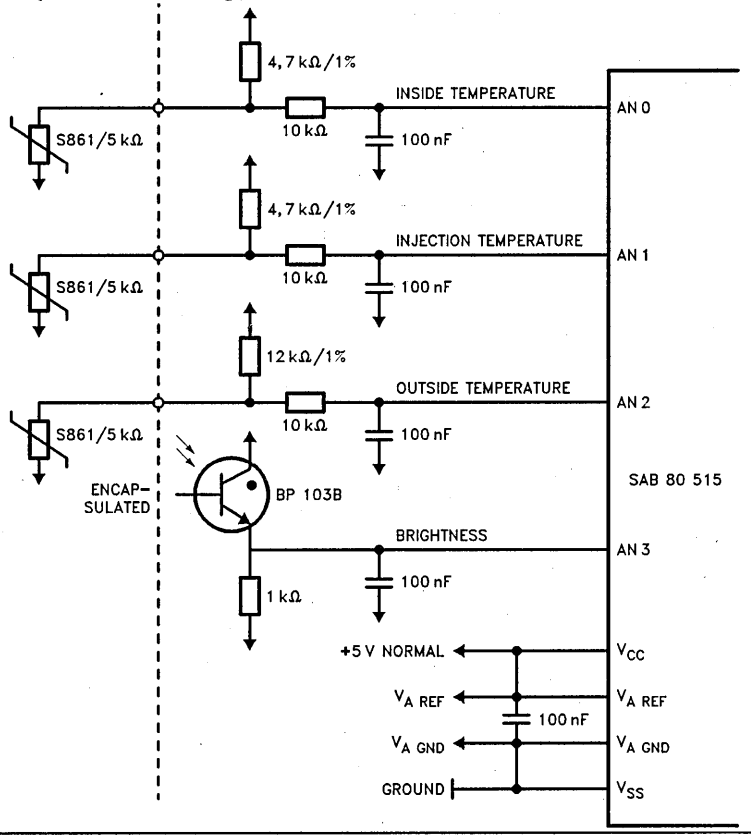
Acquisition and Preprocessing of Input Value Parameters

– Temperature and brightness (Figure 34)

The inside air, fresh air, and outside air temperatures are measured with an NTC sensor S 861 or S 867 (encapsulated). The most suitable locations for installations of the sensors in particular car types must be determined experimentally. The values of the pull-up resistors between the signal line and the analog reference voltage have been selected to ensure optimal accuracy within the required temperature range. Short-term interference pulses are filtered out by an RC network. The phototransistor BP 103B on the face of the device generates a voltage across a resistor in accordance with the ambient light.

The SAB 80515 reads these analog units at regular intervals via the multiplexer and the analog-to-digital converter located on the chip. Since the sensors are connected to the analog references, the result is not affected by the absolute value of these voltages. After the conversion low-frequency fluctuations are suppressed via software averaging. On the basis of tables and linear interpolation, the SAB 80515 computes the values required, i.e. the temperature and the value for control of the display's brightness.

Figure 34
Temperature and Brightness



– Speed (Figure 35)

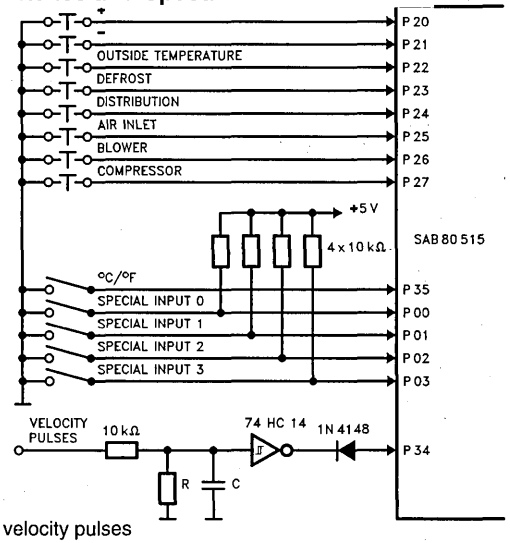
The speed is derived from the generator for electronic speedometers included in most cars. An RC wiring (perhaps with voltage division) and a Schmitt trigger filter out interference in the sensor pulses and adjust the voltage amplitude. With timer 1, the processor counts the pulses received within a defined timer period.

– Keys/Switch (Figure 35)

Since these components are located inside the device, they can be protected against bouncing using software. Because of its many I/O ports, the SAB 80515 can read in information directly. A matrix with decoupling diodes is not required. Also, pull-up resistors are not required at the inputs of the SAB 80515—with the exception of PO. With the hidden F switch, the unit for displaying the nominal and outside temperatures can be selected. The special inputs are used for activating special test functions (see section on "testing and optimization support").

8

Figure 35
Inputs for Keys, Switches and Speed



R:Value according to velocity pulses
C:Value according to max. frequency of the velocity pulses

– Display (Figure 36)

The display is comprised of a three-digit 7-segment LED display (configured from HG 1107 elements). The foremost digit utilizes only four segments. There are also 10 single LEDs LG 3160 for indicating special conditions. The processor drives the display in a four-step multiplex method. For selecting the digit, the outputs P54-P57 go to HIGH in successive order. During this time the information for the segments is present at outputs P40-P46. Four Darlington transistors BC 517 are used as actuator drivers, and seven transistors BC 237 as segment drivers. The voltage source is the 12V supply.

The multiplexed display and the brightness are controlled by timer 2 of the SAB 80515. The component is used as timer in the auto-reload with the oscillator frequency divided by 12. During each overflow the timer is automatically loaded with the content of the CRC register—in this case FF00. This leads to a time interval of 256 X 2 = 512 µs between two overflows.

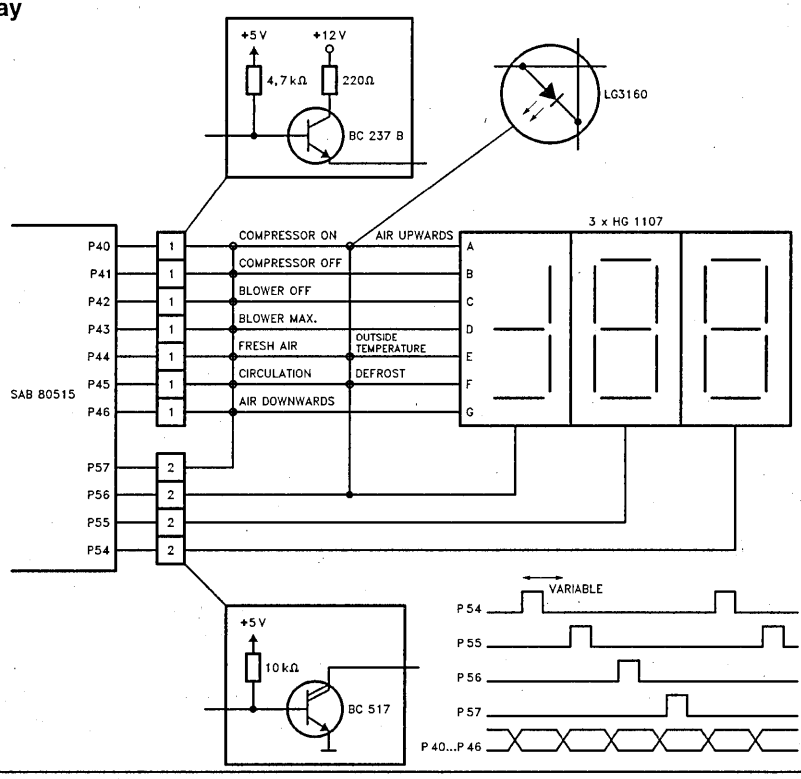
The interval determines the length of a multiplex clock. The interrupt triggered by each overflow results in the output of the new segment information at port 4. The allocated multiplex location is released through port 5.

The display brightness level is determined by the processor on the basis of the ambient light measured by the phototransistor, and a table stored in the ROM. The compare function of timer 2 sets the brightness level: as soon as the timer reaches the value of the compare register, an additional interrupt is triggered. In the associated routine, the processor sets the actuator outputs P54-P57 to "L". This creates an off-period until the timer overflows, the duration of which depends on the content of the compare register. This register can be loaded at any time with the value determined from the ambient light.

– Regulating the inside temperature with the mixing valve (Figure 37)

The temperature inside the car depends largely on the position of the mixing valve, which the SAB 80515 computes by means of the so-called cascade control.

Figure 36
Display



The deviation of the temperature inside the car T_I act from the set nominal value T_{Inom} determines in an outer control circuit the nominal value for the injection temperature T_{Inom}. Through the inner, faster control circuit the mixing valve is adjusted so that the injection temperature actually reaches the value TIn nom. When compared with a less complex control of the mixing valve by means of the difference between the nominal and actual value of the temperature inside the car, this two-stage system results in improved stability. In addition, interference which influences the injection temperature can be quickly rectified (e.g. changes in motor or outside temperature, activation/ deactivation of compressor, switchover from/ to fresh air/air circulation). Also, the min. and max. ratings for the injection temperature can easily be established providing the necessary comfort for the passengers. With properly set parameters the time characteristics as compared to a simple control are equally satisfactory.

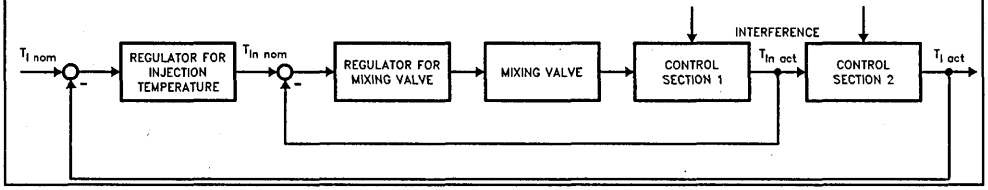
The nominal values for the injection temperature and the mixing valve position are computed according to a digital PID (proportional, integral, and differential) algorithm. Although the variety of parameters which can be set for both controls permit a wide range of adjustments, the expenditure is considerable. Therefore, to facilitate the test and optimization phase, all parameters can be displayed and changed during travel by depressing the respective key (see section 3.10 "testing and optimization support"). For example, by setting the differential portion to zero, a PI characteristic can be obtained.

When "HI" or "LO" is displayed in place of

the nominal temperature, the control algorithm is switched off and the mixing valve is positioned at maximum or minimum heating output.

The control algorithm is also inactive in the defrost status which calls for max. heat output. When switching over to normal operation, the valve returns to its former position.

Figure 37
Block Diagram of Temperature Regulation



Establishing the Nominal Value for the Distribution Valve

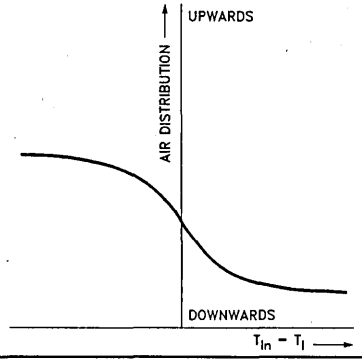
As can be seen in Figure 38, the position of the distribution valve normally depends on the difference between the injection temperature and the temperature inside the car. Cooler air is usually injected upwards while heating air flows downwards towards the floor of the car. The effective nominal value is set by the SAB 80515 with reference to the actual end positions of the valve.

During the special functions "upward air distribution" and "defrost", the air is blown only upwards or during "downward air distribution" only downwards at floor level. When "center air distribution" has been selected, half of the air volume is blown upwards and half downwards.

Setting the Mixing and Distribution Valve (Figure 39)

Both valves are set in the same manner, that is by motors and gears which run or stop in both directions. The components TLE 4201 drive the motor, while the C controls them via ports P50 and P51 (mixing valve) as well as P52 and P53 (distribution valve). The analog-to-digital converter of the processor is informed of the valve position by means of the voltage on a potentiometer which is connected to the valve and supplied by the analog reference voltages. An RC network filters out interference. When the difference between the nominal and actual value of a valve exceeds a certain tolerance margin, the motor is driven in the respective direction.

Figure 38
Position of Distribution Valve versus Injection and Inside Temperature



The valves should reach their end positions (mechanical stops) but the motor, for mechanical reasons, should not be driven continuously in these positions. Since it is difficult to solve this problem by an accurate adjustment of the potentiometer, the system recognizes a mechanical stop when the difference between the actual and the nominal value remains the same although the motor is running. In response, the motor is switched off and the actual value is stored. After that the system will stop when this value has been reached. Only after a certain period of time (approx. 10 minutes) or each time the ignition has been turned on, the user can change the stop by depressing the respective key for selection of a max. position. As a prerequisite for this type of stop recognition, the electrical region of the potentiometer should not be fully utilized by the valve angle.

The slight, relatively rapid fluctuations in the valve nominal value in the regulation or control mode are suppressed to prevent mechanical wear and tear. This suppression is of no consequence to the passengers.

Switching Over the Air Supply

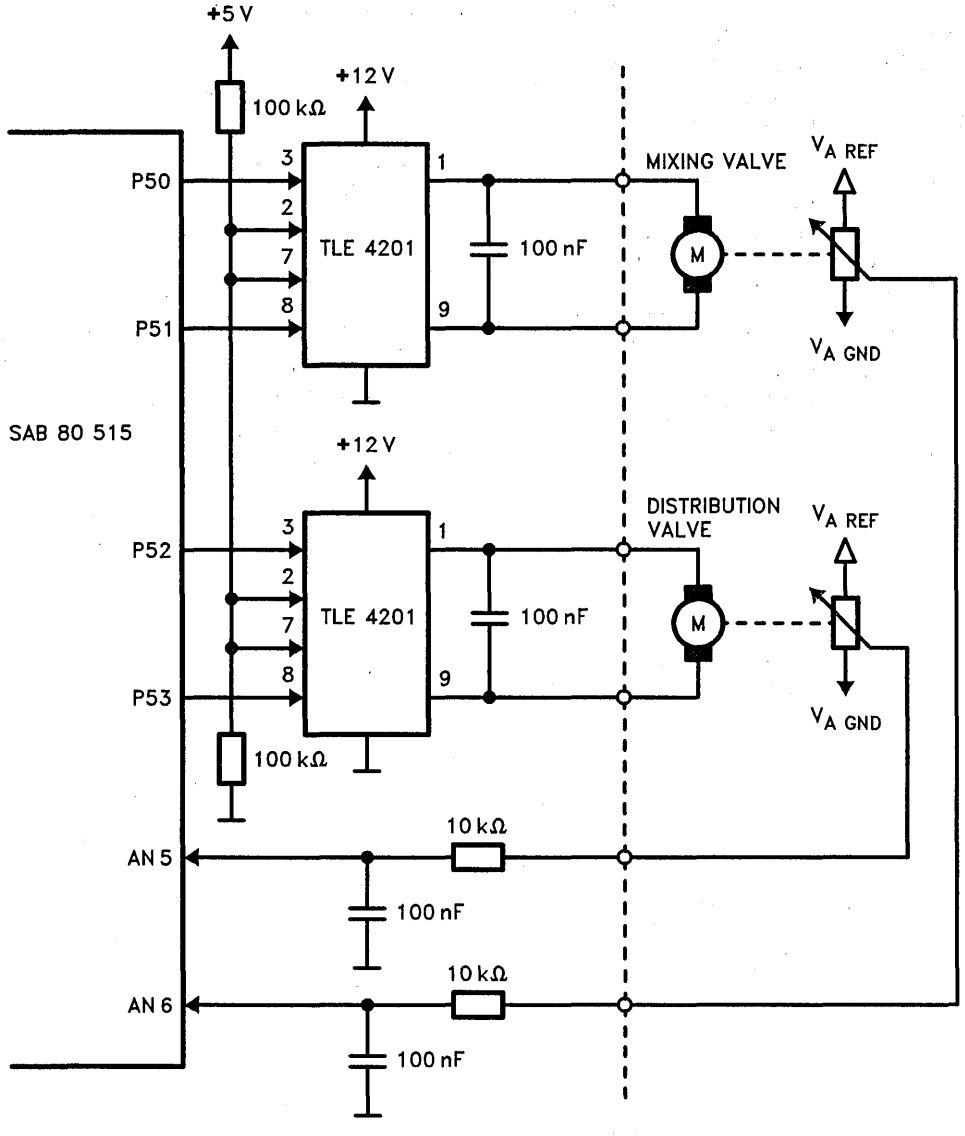
During the automatic function, fresh air will be supplied when the following conditions for air circulation are not met:

— outside temperature > nominal temperature +10K (C)

— outside temperature > inside temperature

To prevent the valve from switching continuously, a hysteresis of 2K in each direction is used.

Figure 39
Control of Mixing and Distribution Valve



During "defrost" or "fresh air supply" the system takes in fresh air; during "ambient air supply", the air in the car is circulated.

The magnetic valve for the air circulation valve is switched on via P15 by the SAB 80515 with a Darlington transistor BC 879. A resistor on the output of the transistor protects against short-term interference (see Figure 40). A short-circuit in the electrical supply can be detected via P05. In this case, the processor immediately stops the control of the valve, but periodically attempts to reactivate it every few seconds. The magnetic valve controls a vacuum motor for activating the air circulation valve.

Water Valve Activation

When there is no heating requirement, the electronics inhibit the water flow to the heat exchanger by means of a valve. As a result, the temperature inside the car can be reduced by several degrees in the summer when compared to operation with closed mixing valve. The criterion for inhibiting the water flow is the mixing valve's position at the lower stop. Only after the mixing valve has changed its position by a defined distance from the stop will the processor enable the valve again.

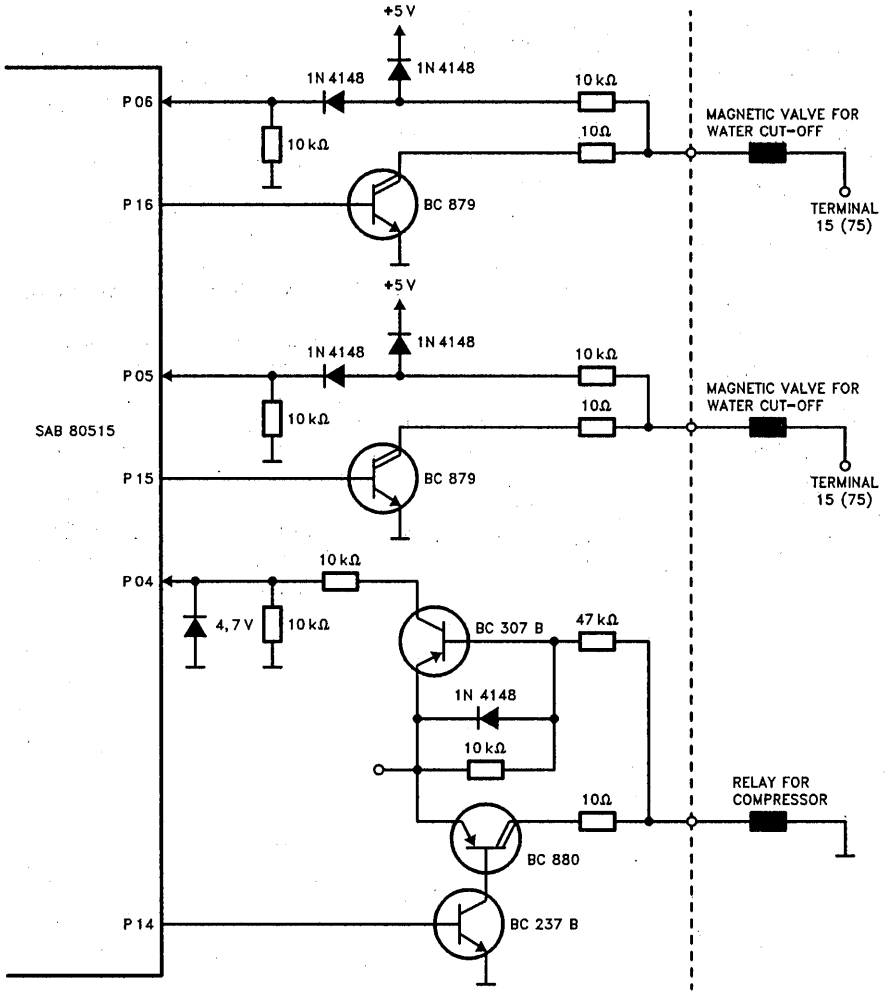
The magnetic valve for inhibiting the water flow is controlled in a manner similar to that used for the air circulation valve (see Figure 40). P16 and P06 are used as outputs or feedback pins. As can be seen in Figure 40, the magnetic valve pneumatically activates the inhibit valve.

Enabling/Disabling the Compressor

During the automatic function, the compressor is disabled only if the outside temperature drops by more than 10K below the nominal temperature. Again a hysteresis of 2K (K = °C) is applied for the switching procedure.

During "defrost" or "compressor ON", the air conditioning unit operates continuously, but stops completely during "compressor OFF".

Figure 40
Control of Water Valve, Circulation Valve and Compressor



The compressor relay is driven (see Figure 40) by P14 of the SAB 80515 as well as two transistors BC 237 and BC 880 for increasing the current and converting the levels. A drive signal short-circuited to ground—after level conversion by transistor BC 307—can be detected via P04 and the system is then switched off for several seconds. The external compressor relay activates the magnetic coupling for driving the compressor, however, only if the (electronically independent) defroster for the carburetor does not respond.

To Drive the Blower

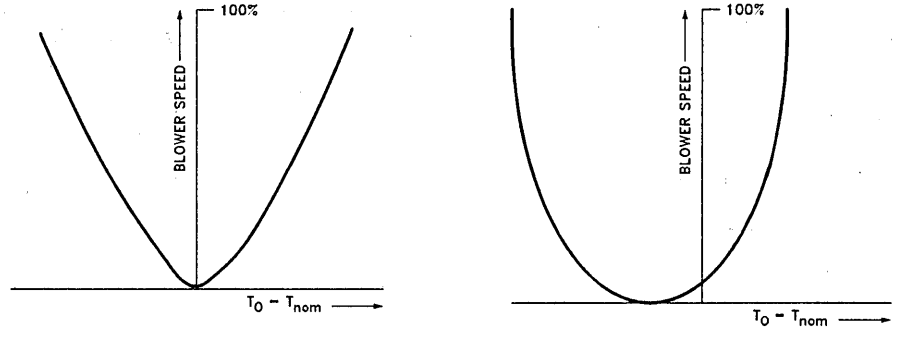
Initially, the speed of the blower is a function of two variables as can be seen in Figure 41. An increase in speed as a function of the nominal-actual temperature difference leads to rapid temperature adjustment. During extreme outside temperatures, the heating or air conditioning effect has to be supported continuously by the blower. The curve minimum is therefore displaced since the average thermal effect of sun rays has been taken into account. The two functions are additively combined.

When both the inside temperature and the injection temperature lie below or above the nominal value for the inside temperature, the blower speed is reduced. Otherwise the already uncomfortably cold car would get colder or, if already too hot, hotter.

Two points were included when considering the dependence on the road speed; during higher speeds the dynamic air pressure increasingly replaces the blower output. In response the blower speed is reduced in proportion to the speed or set to zero, if required. However, during lower speeds or when the car is parked, the noise generated by the blower is irritating, and polluted air is brought in e.g. during heavy traffic. The speed of the blower is therefore reduced.

These automatic functions can be overriden with the blower or defrost key. The blower reaches its max. speed during "full speed blower capacity" or "defrost", operates at a medium speed at "half blower capacity" or not at all in "blower off".

Figure 41
Examples for Blower Speed Characteristics



8

The blower is driven (Figure 42) by the pulse-width modulated signal generated by the microcontroller at P12 with the aid of timer 2. The timer—as described in section 2—has been programmed for an overflow every 512 µs. The compare/capture register 2 operates in the compare mode (mode 0), port 12 is in "L" during timer overflow, or in "H" when the content of the timer and compare register is the same. Therefore, by changing the content of the compare register, the pulse duty factor at P12 can be varied. An HCMOS inverter and an RC combination convert the microcontroller output signal into an analog voltage ranging between 0V and 5V. This voltage drives the blower driver located outside the electronics.

During the standby status, a signal from the voltage supply prevents a voltage from being applied to the blower.

When the blower is operating at full capacity, there should be no voltage drop across the power transistor of the blower driver. Therefore the transistor is by-passed with a relay. For this part of the driver, the SAB 80515 connects a 12V signal to the blower output using two transistors (BC 237 and BC 307). The relay is switched off—to prevent wear and tear—when the speed of the blower is reduced.

Testing and Optimization Support

By encoding at ports P00-P02, the following quantities can be displayed in place of the nominal or outside temperature:

1. Inside temperature T_I in C
2. Injection temperature T_{In} in C
3. Mixing valve setting in %
4. Distribution valve setting in %
5. Blower drive in %
6. Status of compressor, air circulation as well as water valve

7a. Memory address of internal RAM, which can be set

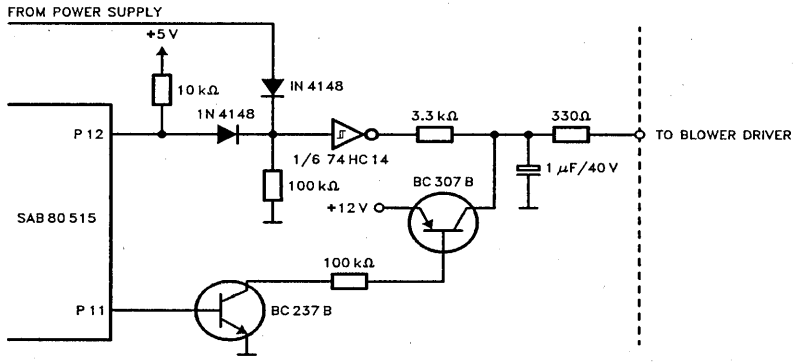
7b. Memory content associated with this address, which can be varied

The controlling and regulating procedures of the system can be monitored with displays 1 through 6.

During operation, the system can be accessed and all memory contents can be indicated with display 7. The settings are performed in the same manner as the changes in nominal temperature, namely with keys "+", and "-". The outside temperature key is used to switch between the memory address and content. There are functions suitable for manual control, and there are those which should be left in the automatic mode.

The possibility of user access has been provided for adjusting the parameters of the two-stage PID control to the respective vehicle. The parameters are not established by the program. Instead they are stored in the RAM in the memory area saved during standby operation. Only after the voltage has been switched off, the parameters can be loaded with fixed values during initialization. The test engineer can therefore change the parameters according to the test results, although the device has already been installed.

Figure 42
Blower Control



In order to provide defined start conditions for a test, all controlling and regulating functions can be switched off during the setting procedures with P03.

After the test has been completed, the established optimal parameters values can be permanently programmed in the EPROM or in the masked ROM.

Use of the On-Chip Periphery of the SAB 80515

Table 3 includes the functions of the integrated periphery of the SAB 80515. As can be seen, almost all elements are utilized.

Table 3 Use of the SAB 80515 On-Chip Periphery

Periphery	Application
Analog-to-Digital	for measuring
Converter	—temperatures
	—brightness level and for setting mixing and distribution value
Timer 0	for generating a standard time clock
Timer 1	for measuring road speed
Timer 2	for controlling the time for LED multiplex display (with brightness control) for controlling the pulse/pause ratio for the blower
Watchdog Timer	for system monitoring
Serial Interface	for diagnostic purposes
Ports	for interrogating and driving digital and time analog inputs/ outputs

8

Alternatives and Upgrades

Changes in the functions of the described sample—provided the sensors and actuators remain the same—can easily be realized by merely modifying the program or the stored tables. However, when different or additional sensors or actuators are used, the hardware must be changed as well.

For example the mixing valve can be replaced by a clocked valve which alternately releases and interrupts the water flow between the cooler and heat exchanger. According to the pulse/pause ratio of the drive signal the heat exchanger temperature changes and thus the air injected into the car. The previously described hot water valve is in this case omitted.

Also, in addition to the distribution valve, other elements for changing the air distribution can be controlled, e.g. the vent flaps at the dashboard.

If the serial interface of the SAB 80515 is not used, the system could be diagnosed during practical application and inspections.

E^2PROM Interface with a Siemens 8031 based Microcontroller

SAB-51 Family

8

Application Note

Introduction

This application note provides users with a solution to interface an E^2PROM to an 8031 based microcontroller over the I^2C bus. In this application example, a Siemens microcontroller the SAB 80535 and a Siemens E^2PROM the SDA 2526 or the SDA 2516 are used.

An on-chip Electrically Erasable Programmable Read-Only Memory (*E^2PROM*) in a microcontroller becomes a very useful peripheral because it allows the system parameters to be stored and reprogrammed without having to remove the microcontroller from the board. An *E^2PROM* can also store internally the system specific information and can quite often replace a battery backed-up CMOS RAM. The write-cycle time of an *E^2PROM* is considerably longer than that of most RAM chips and therefore the device is referred to as a "read-only" memory. Most *E^2PROM*s have a limited number of write/erase endurance cycles and for this reason alone an on-chip *E^2PROM* in a microcontroller is sometimes not desirable. However, Siemens microcontrollers do not have an on-chip *E^2PROM*. Therefore, the user either has to rely on an external *E^2PROM* chip or perform the same function by battery back-up of the microcontroller's internal RAM.

The following application example offers the users a solution of interfacing an *E^2PROM* to a Siemens microcontroller over a two line I^2C interface.

Inter-Integrated Circuit Bus

The Inter-Integrated Circuit (I^2C) bus is a mechanism for serially communicating with peripheral devices. The bi-directional bus consists of two wires, and it can support multiple masters and operate at various data rates.

The physical part of an I^2C interface is a set of two wires, the serial clock (SCL) line and the serial data (SDA) line. The clock line is used to clock data out of a transmitting device into a receiving device. The data line is used to carry the data bits from a transmitting device to a receiving device. It is also used to transfer the acknowledge signal from a receiving device to a transmitting device.

When transferring address, data or acknowledge bits the SDA line can change its state only while the clock is low. If the data line changes from high to low while the SCL line is high, a start condition is initiated. A change from low to high while the SCL line is high initiates a stop condition. The basic relationship between the SCL and SDA lines is shown in Figure 43 Start and Stop Condition.

Hardware

Figure 45 Schematic Diagram illustrates the hardware interface between a Siemens microcontroller such as the SAB 80535 and a Siemens *E^2PROM* such as the SDA 2526 or the SDA 2516. The bit manipulation capability of the SAB 80515/535 allows any two port pins of the microcontroller to be used for the *E^2PROM* with an I^2C bus interface. In this example, port pin P1.0 is used for the SDA line and port pin P1.1 is used for the SCL line of the I^2C bus. One could select any two port pins for the interface by making an appropriate change in the software.

Via the I^2C bus the memory is controlled by the microcontroller (master) during two operating modes:

a. Read-out cycle.

b. Reprogramming cycle or the write cycle.

In both operating modes the microcontroller has to provide 3 bytes and an additional acknowledge clock on the bus after the start condition. These three bytes contain information like chip select for data input, memory word address and chip select for data output. For more information on the operation of the SDA 2526 or the SDA 2516,

please refer to their data-sheets.

Software

The software listing of the subroutines used to read and program (write) the SDA 2526 or the SDA 2516 is attached.

The subroutine Re_EEPROM allows the user to read the *E²PROM*. The following parameters need to be transferred to this subroutine to complete the read operation successfully.

a. Register R1 holds the starting address of the *E²PROM*.

b. Register R2 holds the number of bytes to be read.

c. Register R0 holds the destination address in the internal RAM of the SAB 80535.

The subroutine Pr_EEPROM allows the user to program the *E²PROM*. After programming a byte, the software executes a time delay of about 30 ms (worst case programming duration for the SDA 2526 or the SDA 2516) before writing the next byte. Registers R0, R1 and R2 need to be programmed with the following values before calling this subroutine.

a. Register R1 holds the starting address of the *E²PROM*

b. Register R2 holds the number of bytes to be programmed.

c. Register R0 holds the source address of the data in the internal RAM of the SAB 80535.

References

1. Siemens SAB 80515/535 User's Manual.
2. Siemens SDA 2526/2516 Data Sheets.
3. Siemens I2C Bus Driver Subroutines by R. Mirthes and P. Walter

Figure 43
Start and Stop Condition

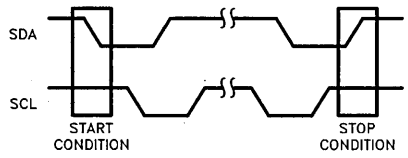
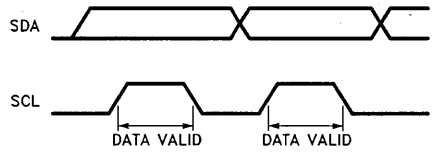
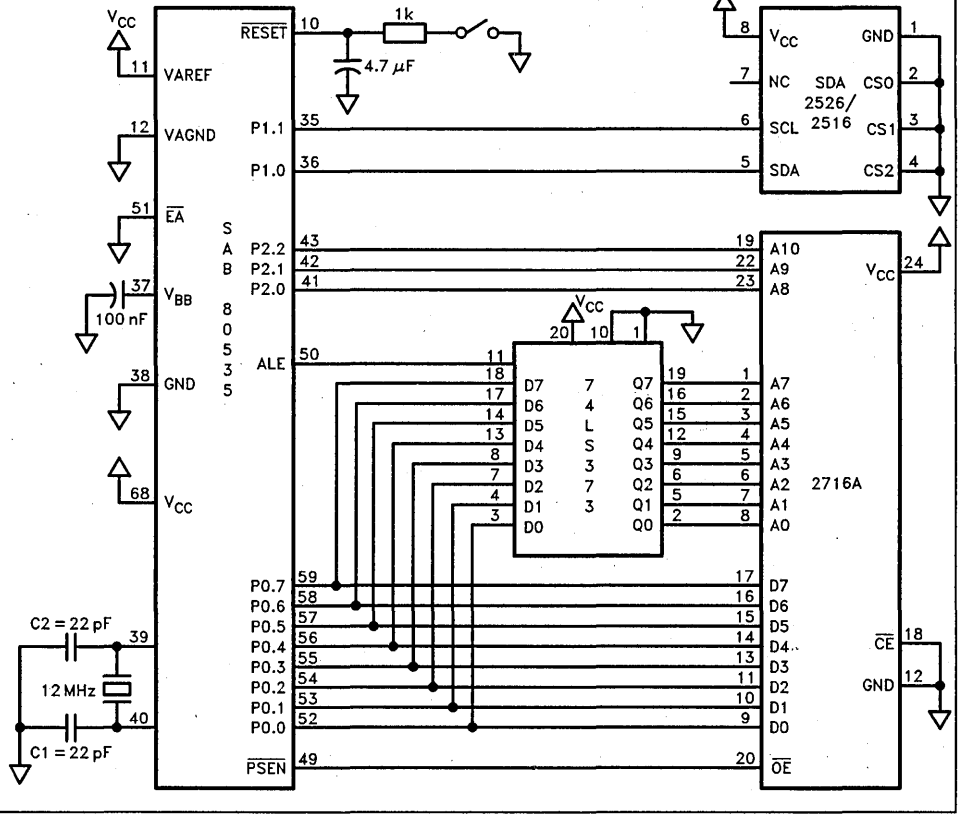


Figure 44
Data and Clock Relationship



8

Figure 45
Schematic Diagram



Program Listings

```
                        1   $MOD515
                        2   $PAGELENGTH(80)
                        3   $TITLE(            SAB 80515/80535 IIC-BUS Software for SDA 2526/2516)
                        4
                        5
                        6
0020                    7   FLAGS           Data    20H
0021                    8   Scratch1        Data    21H
0022                    9   Scratch2        Data    22H
0023                   10   Scratch3        Data    23H
                       11
0090                   12   SDA             Bit     P1.0
0091                   13   SCL             Bit     P1.1
0000                   14   AckFlag Bit     FLAGS.0
                       15
00A1                   16   Adr_EEPROM_L    EQU     10100001B
00A0                   17   Adr_EEPROM_S    EQU     10100000B
                       18
                       19   ;*********************************************************************
                       20   ;*                                                                   *
                       21   ;*  Program Name : IICSDA                                            *
                       22   ;*  Function : This program has the subroutines to read and write    *
                       23   ;*             the Siemens EEPROMs - the SDA 2526 or the SDA 2516.   *
                       24   ;*                                                                   *
                       25   ;*********************************************************************
                       26
0000                   27           Org     0
                       28
0000                   29   Main_Program:
                       30
0000 80FE              31           Sjmp    $
                       32
                       33   ;*********************************************************************
                       34   ;*                                                                   *
                       35   ;*  Subroutine_Name : Re_EEPROM                                      *
                       36   ;*  Function : This subroutine is called to read the EEPROM. The     *
                       37   ;*             number of bytes to be read are loaded in register R2* 
                       38   ;*             Register R1  holds the  starting address  of  the    *
                       39   ;*             location in the EEPROM. Register R0 holds the        *
                       40   ;*             destination address of RAM in the microcontroller.   *
                       41   ;*                                                                   *
                       42   ;*********************************************************************
                       43
0002                   44   Re_EEPROM:
0002 1200D5            45           Call    SaveParam       ; Save input parameters
                       46
0005                   47   Read_EEPROM:
0005 1200DC            48           Call    RestoreParam    ; Restore input parameter
0008 74A0              49           Mov     A,#Adr_EEPROM_s
000A 12002A            50           Call    Cs_EEPROM       ; EEPROM chip select
000D 4011              51           Jc      Re_EEPROM_exit  ; Device not available
                       52
000F 74A1              53           Mov     A,#Adr_EEPROM_L ; C/S for data O/P out of memory
0011 120052            54           Call    Start_IIC       ; Initialize with the address
0014 40EF              55           Jc      Read_EEPROM     ; If error then repeat
                       56
0016                   57   Read_loop:
0016 D200              58           Setb    AckFlag         ; Acknowledge flag
                       59
0018                   60   Lastbyte:
0018 DA09              61           Djnz    R2,Re_EEPROM_Loop
                       62
001A C200              63           Clr     AckFlag         ; It is the last byte
001C 1200B0            64           Call    Datain          ; Read databyte in
001F F6                65           Mov     @R0,A           ; Read databyte in internal RAM
0020                   66   Re_EEPROM_Exit:
0020 020070            67           Jmp     Stop_IIC        ; Stop_IIC condition
                       68
0023                   69   Re_EEPROM_Loop:
0023 1200B0            70           Call    Datain          ; Read databyte in
0026 F6                71           Mov     @R0,A           ; Read databyte in internal RAM
0027 08                72           Inc     R0              ; Inc data pointer
0028 80EE              73           Jmp     Lastbyte        ; Read next byte
                       74
                       75   $EJ
```

8

Program Listings

```
                        76  ;*************************************************************************
                        77  ;*                                                                       *
                        78  ;*  Subroutine_Name   : Cs_EEPROM                                        *
                        79  ;*  Function : This subroutine creates a start condition & sends         *
                        80  ;*             out the device address and the memory word address        *
                        81  ;*             to the SDA 2526/2516.  Register R1 contains the add-*
                        82  ;*             ress. If the device is not available the carry flag *
                        83  ;*             is set as an error condition.                             *
                        84  ;*                                                                       *
                        85  ;*************************************************************************
                        86
002A                    87  Cs_EEPROM:
002A 120052             88          Call    Start_IIC       ; Start condition & device address
002D 4006               89          Jc      Cs_EEPROM_Exit  ; Device not available
002F E9                 90          Mov     A,R1            ; Memory word address
0030 120085             91          Call    Dataout
0033 40F5               92          Jc      Cs_EEPROM       ; If error, repeat
0035                    93  Cs_EEPROM_Exit:
0035 22                 94          Ret
                        95
                        96  ;*************************************************************************
                        97  ;*                                                                       *
                        98  ;*  Subroutine_Name   : Pr_EEPROM                                        *
                        99  ;*  Function:  This subroutine is called to program the  EEPROM.         *
                       100  ;*             The # of bytes to be programmed is loaded in R2. R1 *
                       101  ;*             holds the address of the first byte in EEPROM. R0  *
                       102  ;*             holds the address of the first byte in internal RAM *
                       103  ;*             of the microcontroller.                                   *
                       104  ;*                                                                       *
                       105  ;*************************************************************************
                       106
0036                   107  Pr_EEPROM:
0036 74A0              108          Mov     A,#adr_EEPROM_s
0038 112A              109          Call    Cs_EEPROM       ; EEPROM chip select
003A 4015              110          Jc      Pr_EEPROM_exit  ; Device not available
003C E6                111          Mov     A,@R0           ; Load data byte
003D 120085            112          Call    Dataout
0040 40F4              113          Jc      Pr_EEPROM       ; If error, repeat
0042 120070            114          Call    Stop_IIC        ; Stop_IIC condition
0045 7864              115          Mov     R3,#100         ; 30 ms wait
0047                   116  Wait_Loop:
0047 7C96              117          Mov     R4,#150
0049 DCFE              118          Djnz    R4,$
004B DBFA              119          Djnz    R3,Wait_Loop
004D 08                120          Inc     R0
004E 09                121          Inc     R1
004F DAE5              122          Djnz    R2,Pr_EEPROM    ; Next byte
0051                   123  Pr_EEPROM_Exit:
0051 22                124          Ret
                       125
                       126  ;*************************************************************************
                       127  ;*                                                                       *
                       128  ;*  Subroutine_Name : Start_IIC                                          *
                       129  ;*  Function : Creates a start condition on the I2C bus & then puts*
                       130  ;*             the device address from the accumulator on to the    *
                       131  ;*             bus. After three trials if no acknowledge results    *
                       132  ;*             from the device then the carry flag is set as an     *
                       133  ;*             error condition.                                          *
                       134  ;*                                                                       *
                       135  ;*************************************************************************
                       136
0052                   137  Start_IIC:
0052 7C03              138          Mov     R4,#3           ; Maximum 3 adressing attempts
0054                   139  Init_IIC:
0054 D290              140          Setb    SDA
0056 1200AA            141          Call    Wait_6
0059 D291              142          Setb    SCL
005B 1200AA            143          Call    Wait_6          ; wait
005E C290              144          Clr     SDA             ; Start condition
0060 1200AA            145          Call    Wait_6          ; wait
0063 C291              146          Clr     SCL
0065 1200AA            147          Call    Wait_6
0068 120085            148          Call    Dataout         ; Send device address
006B 5002              149          Jnc     Init_IIC_Exit   ; Acknowledge received
006D DCE5              150          Djnz    R4,Init_IIC     ;Next addressing attempt
006F                   151  Init_IIC_Exit:
006F 22                152          Ret
                       153  $EJ
```

Program Listings

```
                        154   ;*****************************************************************
                        155   ;*                                                               *
                        156   ;*  Subroutine_Name : Stop_IIC                                   *
                        157   ;*  Function : This sobroutine creates a stop condition on the IIC *
                        158   ;*             bus. Then the status of the SDA line is checked - if*
                        159   ;*             low then clock pulses are sent out until SDA line  *
                        160   ;*             goes high.                                        *
                        161   ;*                                                               *
                        162   ;*****************************************************************
                        163
0070                    164   Stop_IIC:
0070 C291               165           Clr     SCL
0072 1200AA             166           Call    Wait_6
0075 C290               167           Clr     SDA
0077 1200AA             168           Call    Wait_6
007A D291               169           Setb    SCL
007C 1200AA             170           Call    Wait_6
007F D290               171           Setb    SDA
                        172
0081                    173   IsSDALow:
0081 3090EC             174           Jnb     SDA,Stop_IIC    ; SDA is not high
0084 22                 175           Ret
                        176
                        177
                        178   ;*****************************************************************
                        179   ;*                                                               *
                        180   ;*  Subroutine_Name : Dataout                                    *
                        181   ;*  Function : This subroutine transmits a databyte on the IIC bus *
                        182   ;*             from the accumulator. The carry flag is set if an  *
                        183   ;*             error results.                                    *
                        184   ;*                                                               *
                        185   ;*****************************************************************
                        186
0085                    187   Dataout:
0085 7F08               188           Mov     R7,#8           ; 8 bits in a byte
                        189
0087                    190   Dataout_Loop:
0087 33                 191           Rlc     A
0088 9290               192           Mov     SDA,C           ; Send data bit out
008A 00                 193           Nop
008B D291               194           Setb    SCL             ; Set the clock bit
008D 1200AA             195           Call    Wait_6
0090 C291               196           Clr     SCL
0092 1200AA             197           Call    Wait_6
0095 DFF0               198           Djnz    R7,Dataout_Loop
0097 33                 199           Rlc     A
0098 D290               200           Setb    SDA
009A 00                 201           Nop
009B D291               202           Setb    SCL             ; Clock pulse for acknowledge
009D 1200AA             203           Call    Wait_6
00A0 A290               204           Mov     C,SDA           ; Get acknowledge bit
00A2 C291               205           Clr     SCL
00A4 1200AB             206           Call    Wait_36
00A7 40C7               207           Jc      Stop_IIC
00A9 22                 208           Ret
                        209
                        210   ;*****************************************************************
                        211   ;*                                                               *
                        212   ;*  Subroutine_Name : Wait_6, Wait_36                            *
                        213   ;*  Function : These subroutines cause delays.                   *
                        214   ;*                                                               *
                        215   ;*****************************************************************
                        216
00AA                    217   Wait_6:
00AA 22                 218           Ret
                        219
00AB                    220   Wait_36:
00AB 7B06               221           Mov     R3,#6
00AD DBFE               222           Djnz    R3,$
00AF 22                 223           Ret
                        224
                        225   $EJ
```

8

Program Listings

```
                        226   ;*********************************************************************
                        227   ;*                                                                   *
                        228   ;*  Subroutine_Name : Datain                                         *
                        229   ;*  Function : This subroutine reads a databyte from the IIC bus     *
                        230   ;*             into the accumulator. If the AckFlag   was set as     *
                        231   ;*             this subroutine was called then an acknowledge is     *
                        232   ;*             given out during the 9TH clock pulse.                 *
                        233   ;*                                                                   *
                        234   ;*********************************************************************
                        235
00B0                    236   Datain:
00B0 D290               237           Setb    SDA
00B2 00                 238           Nop
00B3 7F08               239           Mov     R7,#8           ; 8 bits
                        240
00B5                    241   Datain_Loop:
00B5 D291               242           Setb    SCL             ; Clock pulse
00B7 11AA               243           Call    Wait_6
00B9 A290               244           Mov     C,SDA
00BB 33                 245           Rlc     A
00BC C291               246           Clr     SCL
00BE 11AA               247           Call    Wait_6
00C0 DFF3               248           Djnz    R7,Datain_Loop  ; Next bit
                        249
00C2 D290               250           Setb    SDA
00C4 11AA               251           Call    Wait_6
00C6 300002             252           Jnb     AckFlag ,No_Ack
00C9 C290               253           Clr     SDA
                        254
00CB                    255   No_Ack:
00CB 11AA               256           Call    Wait_6
00CD D291               257           Setb    SCL
00CF 11AA               258           Call    Wait_6
00D1 C291               259           Clr     SCL
00D3 8006               260           Jmp     Wait_36
                        261
                        262   ;*********************************************************************
                        263   ;*                                                                   *
                        264   ;*  Subroutine_Name : SaveParam                                      *
                        265   ;*  Function : This subroutine saves registers R0, R1 & R2.          *
                        266   ;*                                                                   *
                        267   ;*********************************************************************
                        268
00D5                    269   SaveParam:
00D5 8821               270           Mov     Scratch1,R0
00D7 8922               271           Mov     Scratch2,R1
00D9 8A23               272           Mov     Scratch3,R2
00DB 22                 273           Ret
                        274
                        275   ;*********************************************************************
                        276   ;*                                                                   *
                        277   ;*  Subroutine_Name : RestoreParam                                   *
                        278   ;*  Function : This subroutine restores the registers.               *
                        279   ;*                                                                   *
                        280   ;*********************************************************************
                        281
00DC                    282   RestoreParam:
00DC A821               283           Mov     R0,Scratch1
00DE A922               284           Mov     R1,Scratch2
00E0 AA23               285           Mov     R2,Scratch3
00E2 22                 286           Ret
                        287
                        288           End

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

Programmable Timer/Counter Register Array in Microcontrollers

SAB 80515/80535

8

Article Reprint

Introduction

In the past the addition of on-chip memory to microprocessors was the distinctive feature which separated microcontrollers from microprocessors. Since then many bells and whistles have been added to microcontrollers to clearly distinguish them from microprocessors. To appreciate the depth of the microcontroller function cornucopia, consider the following (seemingly endless, but nevertheless partial) list of available features: ROM, RAM, EPROM, E^2PROM, A/D converter, PWM, full-duplex serial ports, on-chip counter/timer, latched I/O, Boolean manipulation etc. Far from dead, the 8-bit microcontroller market continues to grow, as more functions with ever increasing performance are integrated on a single-chip by taking advantage of the advanced NMOS and CMOS processes, tighter design rules, and architectural improvements. One such enhanced functional unit of the microcontroller is a timer/counter unit, which in the past had only been used to generate delays and count pulses. The microcontroller architectures have continued to evolve and expand the scope of the timer/counter functions. This article makes an attempt to re-classify a timer unit of the Siemens 80535/515 (a member of Siemens 8051-based microcontrollers) to give a new wrinkle to control applications. If you have ever felt a need to measure frequencies and pulse widths, or had an application to combine digital and analog functions, or wanted to distinguish noise from real signals, or needed to generate PWM signals, then your solution is the PTRA (Programmable Timer/Counter Register Array) of the Siemens 80535/515 microcontroller. Intel's 8051FA microcontroller has a similar timer unit called the PCA (Programmable Counter Array). But before we draw any conclusions it would be worthwhile to review the features of the PTRA unit available in the Siemens 80535/515 and compare it to the PCA unit of Intel's 8051FA. We will then discuss an application example to evaluate the capabilities and limitations of such timer units.

What is in a name?

A Timer circuit is not new in microcomputers, but of all the on-chip peripheral functions of the 8-bit SAB 80535/515 microcomputer—RAM, ROM, analog-to-digital converter, serial port—none is more useful or important. Two of the SAB 80535/515's forerunners, the SAB 8031/51 and SAB 8032/52, incorporated timers, and it is from these circuits that a new enhanced timer function is derived. The time-base for the PTRA of the SAB 80535/515 is a programmable 16-bit timer/counter also known as the Timer 2. Timer 2 is the only one that can serve the PTRA.

The PTRA consists of the following registers.

T2CON	Timer 2 Control Register
TL2	Timer 2 Register, Low-Byte
TH2	Timer 2 Register, High-Byte
CRCL	Compare/Reload/Capture Register, Low-Byte
CRCH	Compare/Reload/Capture Register, High-Byte
CCL1	Compare/Capture Register 1, Low-Byte
CCH1	Compare/Capture Register 1, High-Byte
CCL2	Compare/Capture Register 2, Low-Byte
CCH2	Compare/Capture Register 2, High-Byte
CCL3	Compare/Capture Register 3, Low-Byte
CCH3	Compare/Capture Register 3, High-Byte
CCEN	Compare/Capture Enable Register

For brevity, the double-byte Compare/Reload/Capture register is called CRC register, and the three double-byte Compare/Capture registers are called CC registers 1 to 3. The PTRA shares Port 1 pins for hardware interfacing as shown in Table 4.

In addition to supporting the operational modes of a timer or a counter, the PTRA provides the following features for the Timer 2.

- –16 bit reload
- –16 bit compare
- –16 bit capture

Figure 46 Block Diagram of the PRTA Unit shows a block diagram of the PTRA unit.

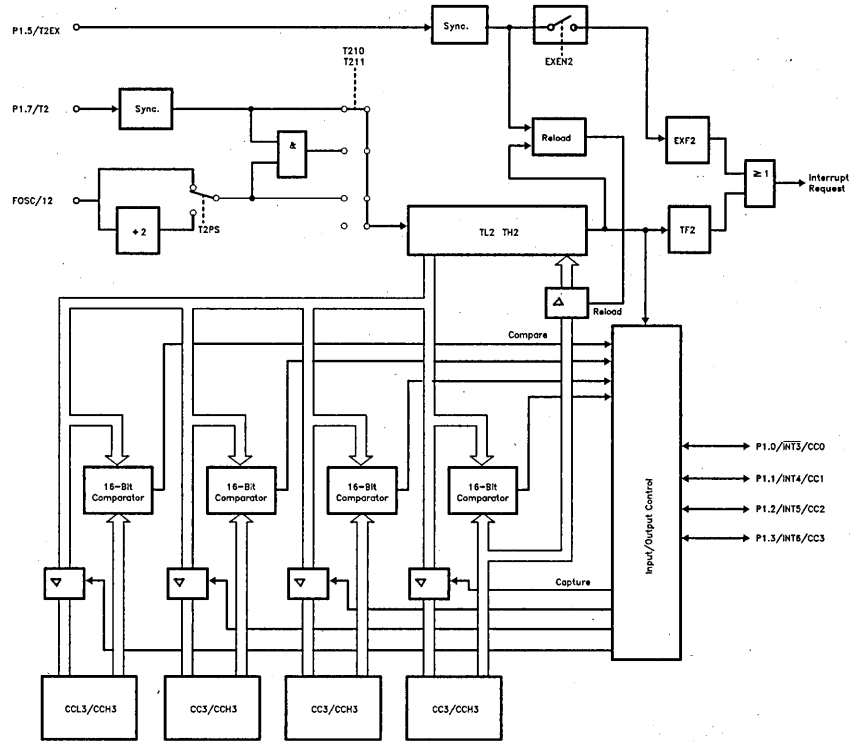
The Timer 2 of PTRA can operate either as timer, event counter, or gated timer. In timer mode, the count rate is derived from the oscillator frequency. A 2:1 prescaler makes it possible to select a count rate of 1/12 or 1/24 of the oscillator frequency. In either case, no matter whether Timer 2 is configured as timer, event counter, or gated timer; when the count rolls over from all 1s to all 0s it sets the Timer 2 overflow flag which can generate an interrupt. The timer overflow function can serve as a periodic interrupt for performing a keyboard scan or for refreshing the screen display. It can also be used to indicate end of delay for the execution of time delayed events. In counter mode, the Timer 2 register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7).

The reload mode of the PTRA unit allows contents of Timer 2 to be reloaded from the CRC register at the time when Timer 2 rolls over from all 1s to all 0s—a feature which is quite useful in varying periods of pulse width modulated signals. The 16-bit reload from the CRC register to Timer 2 registers can also be caused by a negative transition at port pin P1.5.

Table 4 Hardware Interfacing for the PTRA Unit

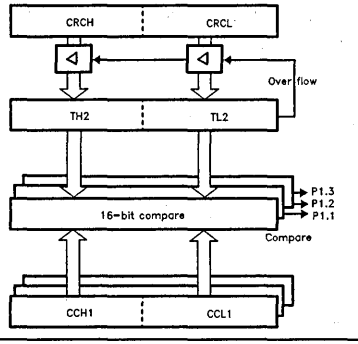
Port Pin	Name	Function
P1.0	INT3/CC0	Compare Output/Capture Input for the CRC Register
P1.1	INT4/CC1	Compare Output/Capture Input for CC Register 1
P1.2	INT5/CC2	Compare Output/Capture Input for CC Register 2
P1.3	INT6/CC3	Compare Output/Capture Input for CC Register 3
P1.5	T2EX	External Reload Trigger Input
P1.7	T2	External Count or Gate Input to Timer 2

Figure 46
Block Diagram of the PRTA Unit



In the compare mode, the 16-bit values stored in the dedicated compare registers of the PTRA are compared with the contents of the Timer 2 registers TL2 and TH2. If the count value in the Timer 2 register matches the stored value, an appropriate output signal is generated at the corresponding port 1 pin, and an interrupt can also be generated. In one of the two compare modes, the Timer 2 of PTRA can be used to generate PWM signals as shown by the functional diagram in Figure 47 PWM Using the PTRA Unit.

Figure 47
PWM Using the PTRA Unit



In the PTRA set-up (compare mode-0) for PWM signal generation, the output signals at port pins P1.1 through P1.3 change from low to high upon a match of the corresponding CC1 through CC3 register contents with contents of the Timer 2 registers. The outputs then go back to a low level on Timer 2 overflow, and Timer 2 restarts by reloading the CRC register contents. The CRC value controls the period and the registers CC1 through CC3 control the pulse widths of the PWM signals generated at port pins P1.1 through P1.3 respectively. In the second mode of operation the output port pins P1.0 through P1.3 can be caused to toggle at compare events, thereby resulting in the generation of square waves at these outputs. In this mode the Timer 2 overflow has no affect on the output port pins. In this mode of operation one can also generate pulses of varying widths from a few to several thousand microseconds without any major software interruption.

Each of the three compare/capture and the CRC registers of the PTRA unit can be used to latch the current 16-bit value of the Timer 2 in one of them. This latching could either be caused by an external event or upon writing to the low-order byte of the dedicated 16-bit capture register. Either a rising or a falling edge can be selected to cause the capture from Timer 2 into the CRC register, whereas for the CC registers 1 to 3, a positive transition causes the capture. In the former case, this capability is useful in applications to determine the state of an input pin, or in discriminating noise (a narrow signal) from real signal, or in measuring pulse widths.The other application of input capture is to determine the frequency of a signal applied to the input pin. By capturing the counter value on successive rising or falling edges, the timer determines the number of elapsed counts. Then, knowing the count rate, the frequency of the input signal can be computed.

The PCA unit of Intel's 8051FA can perform similar functions including 16-bit capture/ compare with capture activated by negative or positive or both edge transition. In addition, one of the compare registers could be preset and programmed to cause an internal reset on a match, thereby acting as a watchdog timer—a function which is supported by another timer in the SAB 80535/515. The PCA unit of 8051FA supports 8-bit PWM as opposed to 16-bit by the PTRA, and has no reload capability for the timer. On the other hand the timer of the PCA can be programmed to count at a rate of FOSC/4 and supports High Speed Output (HSO) operation. In HSO mode of operation the PCA is configured such that a dedicated software bit can be pre-set and when a match occurs, the module reverses the logic level of its I/O pin.

The 16-bit Timer 2 of the PTRA can also be compared with the timers available on 16-bit microcomputers such as NEC 78312 and Intel's 8096, which have similar performance but only have two and one PWM outputs respectively. Even Motorola's 68HC11 offers capabilities similar to that of the PTRA of the SAB 80515/535 but it lacks a PWM output.

8

PWM Outputs

A microcontroller with PWM outputs becomes a useful component in a converter design (Figure 48 Circuit Diagram of a Converter for 3-Phase Motors) to control three phase motors. If the three phase voltage supplied by the converter is variable in frequency and amplitude (a function which can easily be performed by the PWM outputs of the PTRA unit) an efficient control of a three phase motor can be realized. This three phase switching control through a microcontroller will allow motors to have

—Long Operating Times
—High Speed
—Variable RPM
—Frequency Reversing Cycles
—Low Noise, Maintenance-Free Operation

The key role of a microcontroller in such an application is to provide PWM control signals which in effect could be used directly to control the converter.

The PWM outputs characterized by their variable duty cycle and fixed frequency waveforms can also be integrated to provide an approximation to analog outputs. Converting the PWM signal to an analog signal varies in difficulty, depending upon the requirements of the system. Simply, analog signals can be generated from PWM waveforms by using a simple RC or an active filter. Applications such as motor control and switching power supplies actually require a PWM signal, not a true analog one.

PWM and 3 Phase Motor Control

In the control of a 3-phase motor, the motor is driven by a power amplifier stage which is controlled by PWM signals as shown in the block diagram of Figure 48 Circuit Diagram of a Converter for 3-Phase Motors. The micro- controller in this converter diagram acts as a driver and generates the necessary signals to drive the power amplifier stage. The amplifier performs an important function in the converter by providing the correct amount of current and voltage needed to drive the motor. However, a continuous sinusoidal voltage applied to this amplifier results in operational inefficiency caused by the dissipation of large amounts of power in the output transistors. Therefore a pulsed voltage is applied to the motor, which at short intervals switches each motor winding on at full voltage, then switches it off. On and off times of the pulse sequence using PWM method are controlled by the microcontroller in such a way that the average value of all on-cycles represents a sine-wave shape. The running of a 3-phase motor requires a sinusoidal current flow in its windings and does not depend on the shape of the applied voltage. The following section discusses the synthesis of a sinusoidal voltage waveform from a PWM output.

Synthesis of a Sine-Waveform Using PWM

As shown in Figure 50 Synthesis of a Sine-Wave Using PWM, a sine-wave can be formed by a number of synthesis points—each represented by a PWM signal corresponding to the amplitude of the sampled point. If Tp is the time period of the PWM signal then the maximum amplitude (positive peak) of the sine-wave will be represented by a PWM signal which has a high-time = Tp. The high-time of the PWM signal can be varied by selecting different compare values (in compare mode of the PTRA) which corresponds to different amplitude levels in a sine-wave. If T_S is the time period of the synthesized sine-wave and S is the number of synthesis points for one full period of the sine-wave, then

$$T_S = S.T_p$$

It is essential that a timer unit must have the following features to generate PWM signals,

1. Compare registers.
2. PWM outputs (at least three needed for a 3-phase motor control).
3. Capability to auto-reload the timer (a feature which allows to vary the T_p).

Figure 48
Circuit Diagram of a Converter for 3-Phase Motors

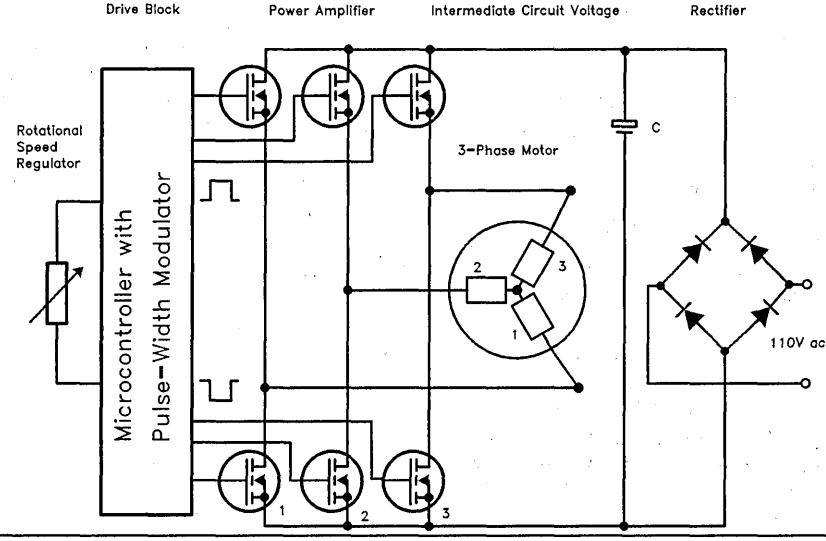
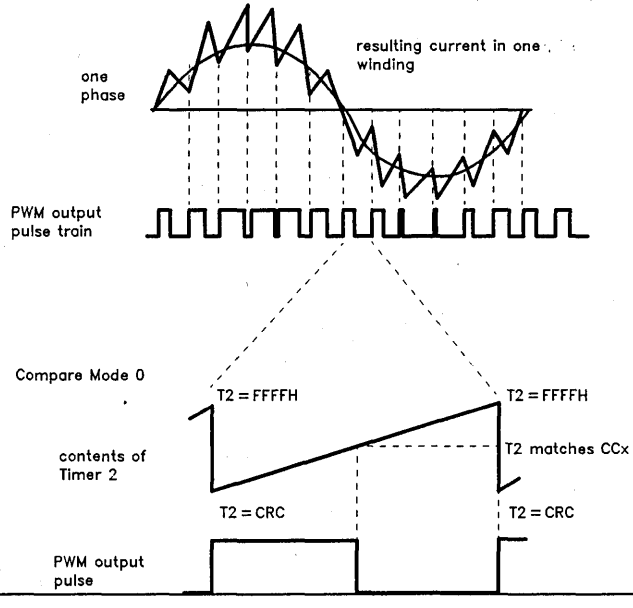


Figure 49
Pulse Width Modulation Using Timer 2 Cycle



The pulse width modulation of samples representing a sine-wave is shown in Figure 49 Pulse Width Modulation Using Timer 2 Cycle. A high-time of 50% duty cycle represents an amplitude of zero magnitude, whereas an almost 100% high-time

8

represents the positive peak and an almost 0% high-time represents the negative peak. In the SAB 80535/515 the maximum count rate of Timer 2 is 750 ns at 16 MHz oscillator frequency. The full count value (0FFFFH) minus the reload value of Timer 2 represents the T_p. For example, a reload value of 0FC00H will result in a T_p of 1024 counts (0FFFFH-0FC00H) or 1024 X 0.75 = 770.25 s. The count of 1024 can also be interpreted as 1024 different compare values to represent 1024 different sample points in a given period of a sine-wave whereas T_p when expressed as a unit of time represents the PWM frequency. This would imply a positive peak could be represented by a pulse which will remain high for 770.25 µs. The full count value minus the reload value of Timer 2 is also referred to as the resolution of the PWM signal. As the reload value approaches the full count, the T_p count decreases, implying an increase in the PWM frequency but a decrease in the resolution of the PWM signal. For example, using a reload value of 0FC00H in the PTRA unit of the SAB 80535/515 one can achieve a PWM frequency of 1.302 kHz with a resolution of 10 bits. Similarly, using the PTRA unit at 16 MHz oscillator frequency, one can achieve 8-bit resolution with a PWM frequency of 5.208 kHz, a 6-bit resolution with a PWM frequency of 20.83 kHz and a 5-bit resolution with a PWM frequency of 41.67 kHz. Clearly, there is a compromise between the PWM frequency and resolution of the PWM signal. One can also have more than one PWM pulse per sample point. If N is the number of equal pulses per synthesis point then

$$Ts = S.N.Tp$$

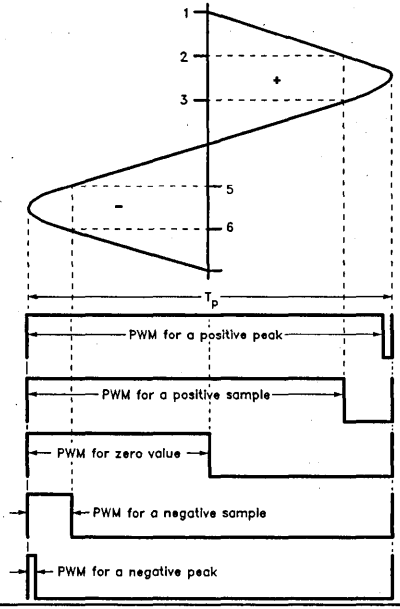
or $$fs = fp/S.N$$

where,

fs = frequency of the synthesized sine-wave

fp = frequency of the PWM signal

Figure 50
Synthesis of a Sine-Wave Using PWM



The PWM frequency depends upon the crystal oscillator frequency of the microcontroller. The number of synthesis points for each full sine-wave cycle must be a multiple of six as otherwise a 120 degree phase displacement cannot be achieved for a three phase operation. Based on these requirements, Table 5 illustrates some values for f_s.

From Table 5 it is clear that a series of sine-waves with a wide range of frequencies can be synthesized using the PWM technique. However, an improper selection of the f_p could cause oscillations and audible "chirps" when a mechanical resonance is met at a particular frequency while controlling a 3-phase motor. As f_p increases the resolution of the PWM decreases and only fewer samples are available to synthesize a sine-wave. The maximum synthesized sine-wave frequency f_s is obtained at a high PWM frequency with fewer synthesis points per period and as the f_p decreases or the number of synthesis points increase the frequency of the synthesized sine-wave decreases. Clearly, continuous sinusoidal frequency variations require alterations of the PWM frequency—a job which the SAB 80535/515 can perform quite easily.

Conclusion

Besides performing the PWM support to control 3-phase motors, the SAB 80535/515 has additional on-chip features which makes it the ideal solution for many control applications. On-chip peripherals such as three timers/counters, 6 x 8-bit ports, A/D converter with programmable reference voltages, watchdog timer, baud-rate generator and a serial interface allow versatile applications in the field of automotive, industrial and consumer electronics.

References

1. Siemens SAB 80535/515 user's manual.
2. Frequency converter for driving asynchronous three-phase motors designed with new micro and power electronic components Siemens application note.

8

Table 5 Different Values for fs

Timer 2 Reload Value	fp in Hz	Clock in MHz	Timer 2 Count Rate	S	N	fs in Hz
0FFC0H	20833	16	750 ns	18	1	1157.4
0FFC0H	20833	16	750 ns	18	2	578.69
0FFC0H	20833	16	750 ns	18	4	289.35
0FFC0H	20833	16	750 ns	54	1	385.80
•	•	•	•	•	•	•
•	•	•	•	•	•	•
0F000H	326	16	750 ns	12	1	27.12
0F000H	326	16	750 ns	12	2	13.56
•	•	•	•	•	•	•
•	•	•	•	•	•	•
0FFC0H	15625	12	1000 ns	24	1	651.04
0FFC0H	15625	12	1000 ns	24	2	325.52
0FFC0H	15625	12	1000 ns	54	1	289.35
•	•	•	•	•	•	•
0F000H	244	12	1000 ns	24	1	10.16

Implementation of the ISDN Oriented Modular (IOM) Interface Using the SAB 80515/80535 Microcontroller

SAB 80515/80535

8

Application Note

Introduction

This application note describes how any two port pins of the SAB 80515/535 microcontroller can be programmed to perform the data transmit and receive line functions of the ISDN Oriented Modular (IOM) interface.

Digitization of telephone networks not only increases transmission rates but allows more than one signal to be sent over a telephone line simultaneously. This capability to offer voice and a multiplicity of data services over one network with standard interfaces and call set-up procedures is the real driving force behind today's telecom/datacom revolution.

This one network is called ISDN—the Integrated Services Digital Network.

Integrated	Telecommunication and data communication services together on one system.
Services	Voice, data, text, picture, video, telex, facsimile, telemetry, alarms, and more.
Digital	Digital transmission from end to end, with voice digitization handled at the terminal.
Network	One worldwide network, based on existing public telephone lines, providing standard interfaces and call procedures familiar to everyone.

ISDN–Standards for Interconnectability

The International Telegraph and Telephone Consultative Committee (CCITT) has published a series of recommendations that have become the industry standard. These recommendations include ISDN definitions for two different groups of transmission speeds:

Basic Access Rate: 144 Kbits/s user information.

Basic access is configured as two 64 Kbits/s B-channels for voice or data transmission and one 16 Kbits/s D-channel for signaling or packet switched data. This is abbreviated as 2xB + D. It is used to connect voice/data terminals to either a private branch exchange (PBX) or directly to the public central office exchange (CO).

Primary Access Rate: 1.544 Mbits/s user information in USA and Japan.

Primary rates are used to transmit large amounts of data. The increased transmission rate helps to reduce the time of transmission and hence the cost of transmission. 64 Kbits/s channels are configured as 23xB + D in USA/Japan.

Siemens provides a number of dedicated solutions in the form of devices, system integration, and software for ISDN. To make the interconnectivity of such ISDN devices possible, Siemens developed the ISDN Oriented Modular (IOM) architecture. The Siemens IOM (rev 2) architecture is fast becoming the de-facto standard for telecom designs. Many IC companies (Siemens, AMD, National and AT&T) now support this standard. Although a rich array of devices currently exists to support the bus, the Telecom market is expanding at such a rate that not all applications are covered. It is now Siemens' intent to have IOM interface available on all ISDN devices. Complete ISDN systems from terminals, terminal adapters, network terminators and transmission repeaters to line cards for digital exchange systems can be optimally designed using the IOM family of ICs.

ISDN Oriented Modular (IOM) Architecture

The IOM interface is a standard 4-wire local interface for the interconnection of ISDN-devices within the ISDN basic access. It consists of a receive and a transmit data line, an 8 KHz frame signal (FSC) and a 512 KHz data clock signal (DCL). The interface is tailor-made to fit the needs of the ISDN basic access. Typical applications shown in"Figure 51 Application of the IOM-Interface in the Basic Access" are:

In a repeater; two identical transmission devices are set back-to-back to achieve an increase in range.

In an NT (Network Termination); two different transmission devices are combined to convert between the S and U interface.

In the terminal (TE) and the exchange; the IOM interface interlinks the layer 1 and layer 2 device. In addition to the point to point configuration, the IOM-interface can support a point-multipoint configuration.

Figure 51
Application of the IOM-Interface in the Basic Access

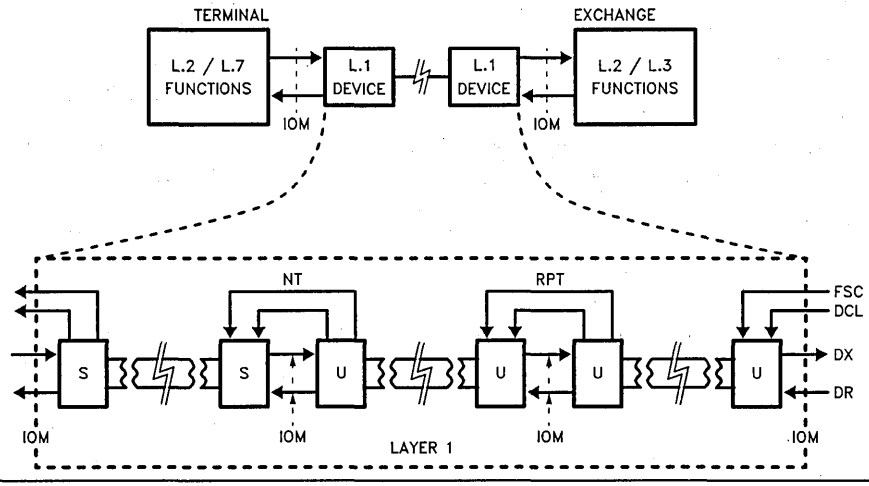
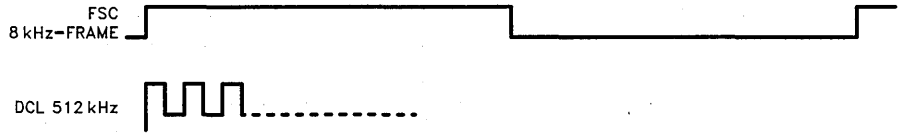


Figure 52
Transferred Information



IOM DIN	B1	B2	Monitor	D	C/I
IOM DOUT	B1	B2	Monitor	D	C/I

B1, B2: Circuit switched voice/data
D: D-channel
C/I: Command/Indicate

8

In all applications the ISDN data rate of 144 Kbits (2xB + 1xD) has to be transferred across the IOM interface transparently. In addition, there is a need for the exchange of control information for activating/ deactivating layer 1 and for switching test loops. A few applications require further capacity to transfer maintenance information via this interface. All this information is transferred in a time-division multiplexed mode based on an 8 KHz frame structure ("Figure 52 Transferred Information") and assigned to the following four octets per frame and direction:

The 64 Kbits/sec channels B1 and B2 occupy the first two octets.

The third octet (monitor channel) is used for transferring the maintenance information.

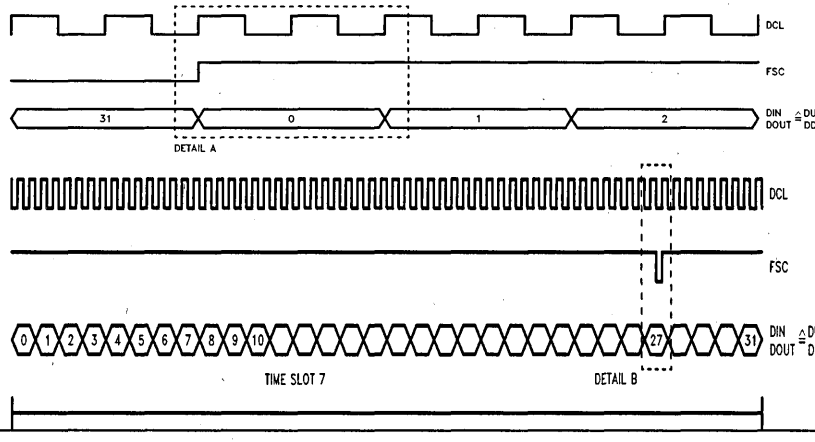
The two D-channel-bits, the four Command/ Indicate-bits (which control the activation/ deactivation procedure), the T-bit and the E-bit are transferred in the fourth octet. The E-bit supports the handling of the monitor channel and the T-bit is reserved for a transparent 1 Kbit channel.

These four octets per frame constitute a bit rate of 256 Kbps. The transmission rate depends on one of the following modes:

Normal Mode: Data Rate: 256 Kbps
Bit Rate: 256 Kbps
Clock Frequency: 512 KHz

MUX Mode: Data Rate: 256 Kbps
Bit Rate: 2.048 Mbps
Clock Frequency: 4.096 MH
Normal Mode
MUX-Mode

Figure53
IOM Interface Timing



In the multiplex mode the outputs of up to eight layer 1 devices may be connected together to form an eight-time-slot IOM interface bus with a 2 MHz data rate. The physical timing of the IOM-interface in different modes is shown in "Figure53 IOM Interface Timing".

SAB 80515/535 Role

The primary role of the SAB 80515/535 microcontroller in generating the IOM interface in normal mode is to accept the frame synchronization (FSC) signal as an input and based on this signal identify the four octets on the receive and transmit data lines. Any two port pins of the SAB 80515/ 535 can be used as the transmit and receive data lines. The PTRA unit of the SAB 80515/ 535 plays a major role not only in defining the octets but also in generating and receiving the data bit stream over the transmit and

receive lines in synchronization with the frame and data clock signals. In this respect the SAB 80515/535 falls into a unique family of microcontrollers which can perform this function.

PTRA Unit of the SAB 80515/535

The Programmable Timer/Counter Register Array (PTRA) of the SAB 80535/515 has a time-base which is a programmable 16-bit timer/counter also known as the Timer 2. Timer 2 is the only one that can serve the PTRA.

The PTRA consists of the following registers:

T2CON Timer 2 control register

TL2 Timer 2 register, low-byte

TH2 Timer 2 register, high-byte

CRCL Compare/reload/capture register, low-byte

CRCH Compare/reload/capture register, high-byte

CCL1 Compare/capture register 1, low-byte

CCH1 Compare/capture register 1, high-byte

CCL2 Compare/capture register 2, low-byte

CCH2 Compare/capture register 2, high-byte

CCL3 Compare/capture register 3, low-byte

CCH3 Compare/capture register 3, high-byte

CCEN Compare/capture register enable register

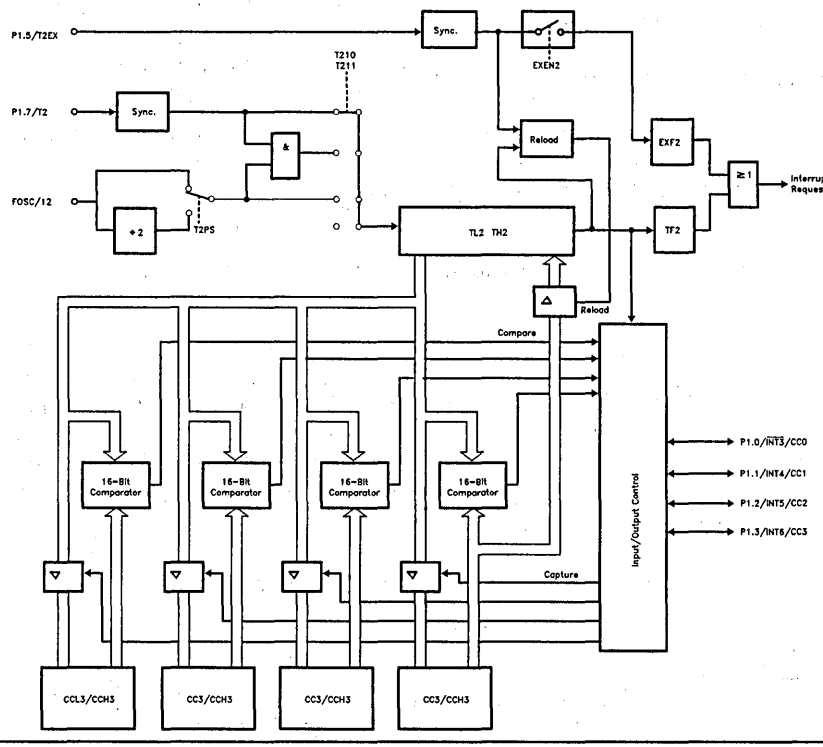
For brevity, the double-byte Compare/Reload/Capture register is called CRC register, and the three double-byte Compare/Capture registers are called CC registers 1 to 3. The PTRA shares Port 1 pins for hardware interfacing as shown in Table 6.

Table 6 Hardware Interfacing for the PTRA Unit

Port Pin	Name	Function
P1.0	INT3/CC0	Compare Output/Capture Input for the CRC Register
P1.1	INT4/CC1	Compare Output/Capture Input for CC Register 1
P1.2	INT5/CC2	Compare Output/Capture Input for CC Register 2
P1.3	INT6/CC3	Compare Output/Capture Input for CC Register 3
P1.5	T2EX	External Reload Trigger Input
P1.7	T2	External Count or Gate Input to Timer 2

8

Figure 54
Block Diagram of the PTRA Unit



In addition to supporting the operational modes of a timer or a counter, the PTRA provides the following features for the Timer 2.

—16-bit reload
—16-bit compare
—16-bit capture

"Figure 54 Block Diagram of the PTRA Unit" shows a block diagram of the PTRA unit.

The Timer 2 of PTRA can operate either as timer, event counter, or gated timer. In timer mode, the count rate is derived from the oscillator frequency. A 2:1 pre-scaler makes it possible to select a count rate of 1/12 or 1/24 of the oscillator frequency. In either case, no matter whether Timer 2 is configured as timer, event counter, or gated timer; when the count rolls over from all 1s to all 0s it sets the Timer 2 overflow flag which can generate an interrupt. In counter mode, the Timer 2 register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7).

The reload mode of the PTRA unit allows contents of Timer 2 to be reloaded from the CRC register at the time when Timer 2 rolls over from all 1s to all 0s. The 16-bit reload from the CRC register to Timer 2 registers can also be caused by a negative transition at port pin P1.5.

In the compare mode, the 16-bit values stored in the dedicated compare registers of the PTRA are compared with the contents of the Timer 2 registers TL2 and TH2. If the

count value in the Timer 2 register matches the stored value an appropriate output signal is generated at the corresponding port 1 pin, and an interrupt can also be enabled.

In the PTRA compare mode-0, the output signals at port pins P1.1 through P1.3 change from low to high upon a match of the corresponding CC1 through CC3 register contents with contents of the Timer 2 registers. The outputs then go back to a low level on Timer 2 overflow, and Timer 2 restarts by reloading the CRC register contents. In the second mode of operation the output port pins P1.0 through P1.3 can be caused to toggle at compare events, thereby resulting in the generation of square waves at these outputs. In this mode the Timer 2 overflow has no effect on the output port pins. Each of the three compare/capture and the CRC registers of the PTRA unit can be used to latch the current 16-bit value of the timer 2 in one of them. This latching could either be caused by an external event or upon writing to the low-order byte of the dedicated 16-bit capture register. Either a rising or a falling edge can be selected to cause the capture from Timer 2 into the CRC register, whereas for the CC registers 1 to 3 a positive transition causes the capture.

IOM Interface Using the SAB 80515/535

The high speed of the SAB 80515/535 microcontroller allows its machine cycle (1/12 of oscillator frequency) to be twice as fast as the DCL which means the microcontroller can easily handle two DCLs per data bit. To achieve this, the crystal oscillator frequency is selected at 24 times the data clock i.e., 24x512 KH = 12.288 MHz. As a result, the Timer 2, which is programmed to operate in mode 1, also counts at twice the rate of the data clock. Timer 2 is programmed to increment at the rate of 1024 KHz as opposed to counting at the data clock rate of 512 KHz. Though the latter could be accomplished by connecting the data clock to port pin P1.7 of the microcontroller and letting the Timer 2 count in the external count mode. But this is not desirable as the count rate of 1024 KHz gives a higher resolution for the compare feature. However, the synchronization is achieved by programming the Timer 2 to count at a value of 0FF80H every time the falling edge of the FSC (Frame Synchronization) signal is detected at port pin P1.5. The falling edge of the FSC signal coincides with the occurrence of the first data bit in octet 3 (monitor channel).

With 0FF80H being the start value of Timer 2 which also marks the beginning of the third octet, the other octets occur 32 counts apart. This derives from the fact that there are two DCLs per data bit (4 machine cycles or 4 counts per data bit) and there are 8 data bits per octet. Based on this analogy, the octets 3, 4, 1 and 2 occur at the Timer 2 count values of 0FF80H, 0FFA0H, 0FFC0H and 0FFE0H respectively as shown in "Figure 55 Octet Placement in FSC Signal". In other words, if the compare interrupt was invoked at these count values, the user will be in the desired octet to transmit or receive a data bit stream. However, the delay involved in processing the interrupt will also delay the occurrence of a dat bit at the desired DCL. To understand the compare interrupt handling in the SAB 80515/535, one will first have to understand the basic machine cycle timings of the SAB 80515/535.

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into two phases—phase 1 and phase 2. Thus a machine cycle consists of 12 phases or 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each phase lasts or one oscillator period and each state lasts for two oscillator periods. Timer 2 is incremented in S1P1 of every machine cycle. An internal compare signal is generated in S2P2, if Timer 2 increments to the compare count in one of the compare registers CC1 through CC3. The compare signal is active as long as the Timer 2 contents are equal to one of the compare registers. For the compare registers CC1 to CC3 an interrupt is always requested and a flag is set when the compare signal goes

active. The interrupt flags are sampled in S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition in S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine. As shown in "Figure 56 Interrupt Response Timing Diagram" a standard interrupt handling procedure can take about 4 machine cycles. For exceptions please refer to the SAB 80515/535 user's manual.

Based on above facts, in the present application, two DCLs (4 machine cycles) have already elapsed before the program enters into the compare subroutine. To avoid this delay, the compare interrupt subroutine is invoked 30 machine cycles or 15 DCLs before the desired octet. Therefore the compare event for different octets occur at the following count values:

Octet 1, count value = 0FFA2H
Octet 2, count value = 0FFC2H
Octet 3, count value = 0FFE2H
Octet 4, count value = 0FF82H

Figure 55
Octet Placement in FSC Signal

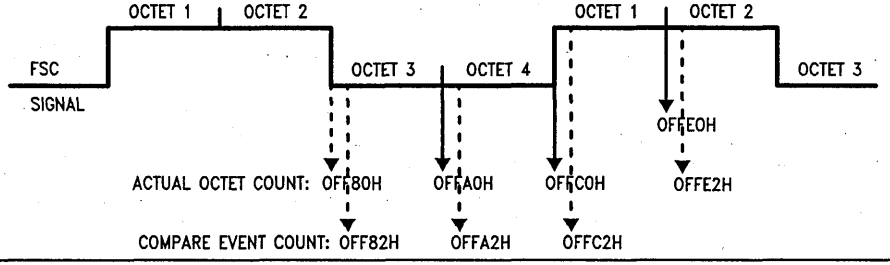
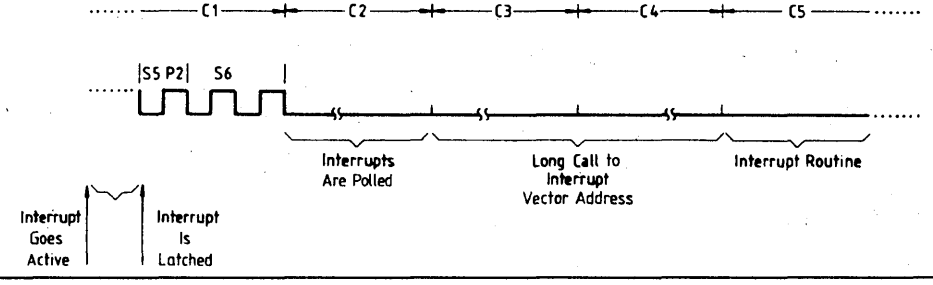


Figure 56
Interrupt Response Timing Diagram



In either transmit or receive mode, when compare interrupt occurs at one of the above count values, the microcontroller performs no-operation until 30 machine cycles elapse. At that instant, the microcontroller sends out or receives the data bit stream as that is the beginning of the desired octet.

System Implementation

The program listing (refer to Appendix A) has two subroutines—one for transmit and the other for receive mode. In either case the data variable "count" holds the lower byte of the desired octet to perform the compare match. In transmit mode the data is taken

from external memory and in the receive mode the data is stored in the external memory starting at address 000H. The total time spent in the compare interrupt subroutine to transmit or receive a byte varies from 75 to 82 machine cycles. This variation is caused due to the time spent in deciding whether or not the byte transmitted or received is the last byte. The next byte is not transmitted or received until the next octet in subsequent frame i.e., 128 machine cycles later. Therefore the CPU is free for 128 machine cycles minus the time spent in machine cycles in the compare interrupt subroutine. However more time could be made available for the CPU by cutting down the number of NOPs in the compare interrupt subroutine and by increasing the compare count to compensate for the number of NOPs removed. In that case the compare event could occur as close as 4 machine cycles before the desired octet count.

Any two port pins of the SAB 80515/535 can be used as the data transmit and data receive lines. In this application port pin P3.0 is used as the receive data line and the port pin P3.1 is used as the transmit data line. "Figure 57 SAB 80535 Schematic Diagram" shows the basic SAB 80535 circuit required to run the subroutines. The pins used for the IOM interface are also shown. The other on-chip peripherals namely the serial port, Timer 0 and Timer 1, the watchdog timer, the A/D converter and 30 I/O lines are still available to perform any other system functions.

"Figure 58 NT–Simulator Application Using the SAB 80515/5353 as a Terminal" shows an IDSN application where the SAB 80515/535 is being used as a terminal interfaced to an ISDN network simulator over the S-interface.

References

1. Siemens, 8-Bit Single-Chip Microcontroller Handbook, 1989/90.
2. Siemens, Telecommunications Data Book, 1987.
3. Total ISDN Commitment, a Siemens Brochure.

8

Figure 57
SAB 80535 Schematic Diagram

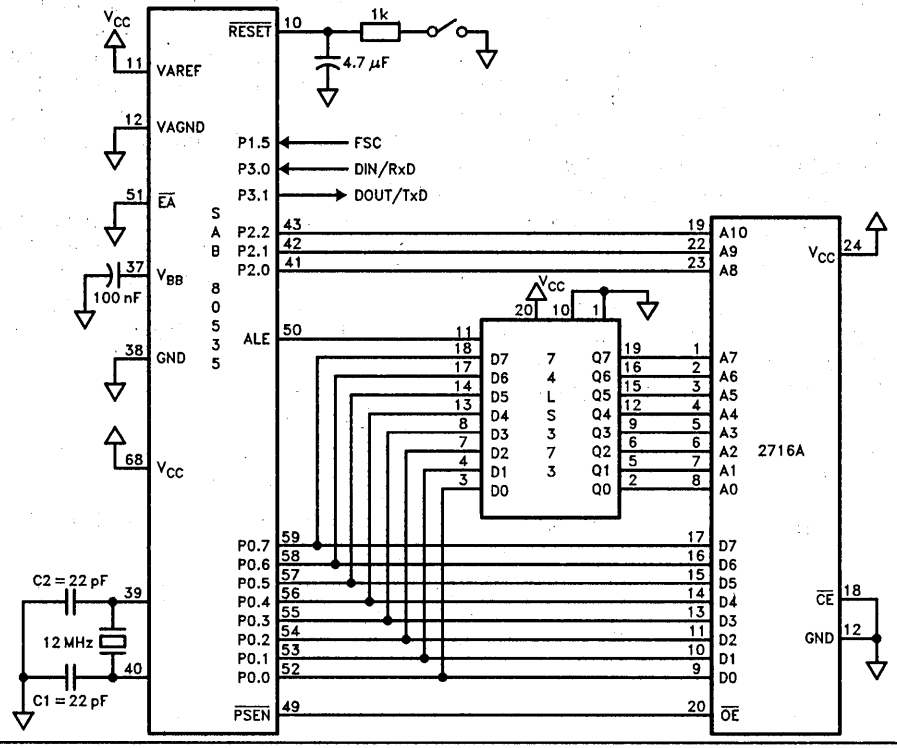
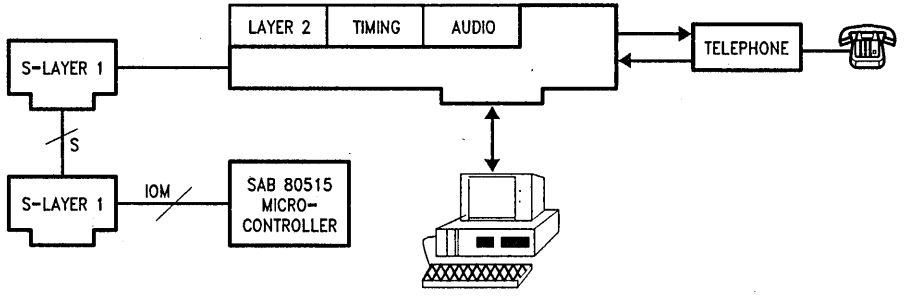


Figure 58
NT–Simulator Application Using the SAB 80515/5353 as a Terminal



```
IOMW        IOM     Interface           Appendix A
            1       $Mod515
            2       $Pagelength(70)
            3       $Debug
            4       $Title (IOM Interface Appendix A)
            5
            6
            7
            8       ;*****************************************************
            9       ;* IOM Interface Program:                            *
            10      ;*In the following program the variable "Count"      *
            11      ;*count in which the data byte is to be transmitted. *
            12      ;*The valid octet counts are: for octet 1, count =   *
            13      ;*0A2H; octet 2, count = 0C2H; octet 3, count = 0E2H; *
            14      ;*octet 4, count = 82H. These bytes are stored in the *
            15      ;*external data memory. The bytes appear in the      *
            16      ;*desired octet in every subsequent frame until      *
            17      ;*the last byte is transmitted. The external         *
            18      ;*oscillator frequency is 12.288 MHz (24 x 512 KHz). *
            19      ;*****************************************************
            20
            21
00A2        22      Count EQU 0A2H ;Octet 1 = A2,2=C2,3=E2,4=82.
            23
0000        24C     seg at OH
            25
RM 020100   26      LJMP Main_Begin
            27
            28      ;*****************************************************
            29      ;*Write_IOM_Subroutine:                             *
            30      ;*The following subroutine happens 30 machine cycles *
            31      ;*(15 DCLs) before the occurrence of the desired     *
            32      ;*octet. The data byte is output on port bit P3.1    *
            33      ;*coinciding with the occurrence of the desired      *
            34      ;*octet. The data byte is taken from external        *
            35      ;*memory and the data pointer to external           *
            36      ;*memory is incremented after a byte is transmitted  *
            37      ;*over the IOM interface. The subroutine is         *
            38      ;*exited before the arrival of the falling edge      *
            39      ;*of the FSC at port pin P1.5.                      *
            40      ;*****************************************************
005B        41                      Org 5BH       ;Compare 1 interrupt
routine.
            42
005B        43      Write_IOM_Subroutine: ;4 m/c for servicing the
            44                                    routine.
RM C2BB     45      CLR IEN1.3                    ;7th m/c into the frame.
            46
RM 00       47      NOP             ;8th m/c into the frame.
RM 00       48      NOP             ;9th m/c into the frame.
RM 00       49      NOP             ;10th m/c into the frame.
RM 00       50      NOP             ;11th m/c into the frame.
RM 00       51      NOP             ;12th m/c into the frame.
RM 00       52      NOP             ;13th m/c into the frame.
RM 00       53      NOP             ;14th m/c into the frame.
RM 00       54      NOP             ;15th m/c into the frame.
RM 00       55      NOP             ;16th m/c into the frame.
RM 00       56      NOP             ;17th m/c into the frame.
RM 00       57      NOP             ;18th m/c into the frame.
```

8

```
IOMW        IOM     Interface       Appendix A

RM 00       58      NOP             ;19th m/c into the frame.
RM 00       59      NOP             ;20th m/c into the frame.
RM 00       60      NOP             ;21th m/c into the frame.
RM 00       61      NOP             ;22th m/c into the frame.
RM 00       62      NOP             ;23th m/c into the frame.
RM 00       63      NOP             ;24th m/c into the frame.
RM 00       64      NOP             ;25th m/c into the frame.
RM 00       65      NOP             ;26th m/c into the frame.
RM 00       66      NOP             ;27th m/c into the frame.
RM 00       67      NOP             ;28th m/c into the frame.
            68
RM 13       69      RRCA            ;29th m/c into the frame.
RM 00       70      NOP             ;30th m/c into the frame.
RM 92B1     71      MOVP3.1,C       ;Two more cycles.
            72
RM 13       73      RRCA            ;New octet, first bit arrives.
RM 00       74      NOP             ;34th cycle.
RM 92B1     75      MOVP3.1,C       ;Second bit sent.
            76
RM 13       77      RRCA            ;Third bit to carry flag.
RM 00       78      NOP             ;38th cycle.
RM 92B1     79      MOVP3.1,C       ;Third bit sent
            80
RM 13       81      RRCA            ;Fourth bit to carry flag.
RM 00       82      NOP             ;42nd cycle.
RM 92B1     83      MOVP3.1,C       ;Fourth bit sent.
            84
RM 13       85      RRCA            ;Fifth bit to carry flag.
RM 00       86      NOP             ;46th cycle.
RM 92B1     87      MOVP3.1,C       ;Fifth bit sent.
            88
RM 13       89      RRCA            ;Sixth bit to carry flag.
RM 00       90      NOP             ;50th cycle.
RM 92B1     91      MOVP3.1,C       ;Sixth bit sent.
            92
RM 13       93      RRCA            ;Seventh bit to carry flag.
RM 00       94      NOP             ;54th cycle.
RM 92B1     95      MOVP3.1,C       ;Seventh bit sent.
            96
RM 13       97      RRCA            ;Eighth bit to carry flag.
RM 00       98      NOP             ;58th cycle.
RM 92B1     99      MOVP3.1,C       ;Eighth bit sent.
            100
RM 00       101     NOP             ;122th DCL (61st cycle).
RM 00       102     NOP             ;62nd cycle.
RM 00       103     NOP             ;63rd cycle.
RM D2B1     104     SETBP3.1        ;32nd DCL (64th cycle).
            105
RM A3       106     INCDPTR         ;Increment the data pointer.
RM E0       107     MOVXA,@DPTR     ;Load the acc. with next byte.
RM A883     108     MOVR0,DPH       ;To compare if end of array
RM B8FF08   109     CJNER0,#0FFH,CONTI;Compare the high byte first.
RM A882     110     MOVR0,DPL       ;Load the lower byte for comparison.
RM B8FF03   111     CJNER0,#0FFH,CONTI;Compare the lower byte.
112
00A3 900000 113     MOVDPTR,#00H    ;Initialize the data pointer.
114
```

```
IOMW          IOM      Interface          Appendix A

00A6          115      CONTI:
00A6 C2C3     116      CLR IRCON.3
00A8 D2BB     117      SETB IEN1.3
00AA 32       118      RETI
              119
              120
              121
              122      ;*****************************************************
              123      ;* Write_IOM_Set-Up:                                 *
              124      ;* In this program the registers are set up to       *
              125      ;* allow compare to happen at the "count" value. The *
              126      ;* PTRA unit of the SAB 80535 is programmed in the   *
              127      ;* compare mode 1. The "count" value is assigned     *
              128      ;* to the compare register 1 and the corresponding   *
              129      ;* interrupt is enabled. The Timer 2 of the PTRA     *
              130      ;* unit is programmed to restart at a value of FF80H *
              131      ;* on the arrival of the falling edge of the         *
              132      ;* FSC signal.                                       *
              133      ;*****************************************************
              134
0100          135      ORG 100H
              136      Main_Begin:
RM 900000     137      MOV DPTR,#00H      ;DPTR = (Start of data array).
RM 75C100     138      MOV CCEN,#00H      ;Disable compare mode.
RM 75B0FF     139      MOV P3,#0FFH       ;SDO, SDI lines are high.
              140
0109 75C3FF   141      MOV CCH1,#0FFH     ;Compare register has a value
010C 75C2A2   142      MOV CCL1,#Count    ;corresponding to the octet desired.
010F 75CBFF   143      MOV CRCH,#0FFH
0112 75CA80   144      MOV CRCL,#080H     ;Reload = 0FFFH - 128 (32x4)
0115 75CDFF   145      MOV TH2,#0FFH
0118 75CC81   146      MOV TL2,#81H ;To avoid compare at FF80H
011B D2CA     147      SETB T2CM          ;T2 in compare mode 1
011D 75C108   148      MOV CCEN,#08H      ;Compare mode enabled
0120 D2F      149      SETB EAL           ;Master interrupt Bit enabled.
              150
0122 758910   151      MOV TMOD,#10H      ;Timer1 to cause more than
0125 758DFF   152      MOV TH1,#0FFH      ;quarter of a frame delay
0128 758BD0   153      MOV TL1,#256-48    ;to avoid match before reload.
              154
012B          155      Start:
012B 2096FD   156      JB P1.6,$          ;Prepare for start pulse arrival.
012E 3096FD   157      JNB P1.6,$         ;Wait for the start pulse, 1 m/c.
0131 D28E     158      SETB TR1           ;So far 2 m/c into the frame.
0133 308FFD   159      JNB TF1,$          ;Wait for another 49 m/c
              160
0136 75C81D   161      MOV T2CON,#1DH     ;T2 auto reload from P1.5.
0139 D2BB     162      SETB IEN1.3        ;Compare interrupt 1 enable.
              163
013B 213B     164      AJMP $             ;Wait for a match.
              165
              166      end
ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

8

```
IOMRA       IOM     Interface          Appendix A

            1       $Mod515
            2       $Pagelength(73)
            3       $Debug
            4       $Title (IOM Interface Appendix A)
            5
            6
            7
            8       ;*************************************************************
            9       ;* IOM Interface Program:                                    *
            10      ;* In the following program the variable "Count"             *
            11      ;* holds the octet count from which the data byte is to be*
            12      ;* received. The valid octet counts are: for octet 1,        *
            13      ;* count = 0A2H; octet 2, count = 0C2H; octet 3,             *
            14      ;* count = 0E2H; octet 4, count = 82H. These bytes are       *
            15      ;* stored in the external data memory. The bytes are read *
            16      ;* from the desired octet in subsequent frames until the     *
            17      ;* last byte is transmitted. The external crystal            *
            18      ;* oscillator frequency is 12.288 MHz (24 x 512 KHz).        *
            19      ;*************************************************************
            20
            21
 00A2       22      Count EQU 0A2H    ;Octet 1=A2,2=C2,3=E2,4=82.
            23
0000        24      Cseg at OH
            25
RM 020100   26      LJMP Main_Begin
            27
            28      ;*************************************************************
            29      ;* Read_IOM_Subroutine:                                      *
            30      ;* The following subroutine happens 30 machine cycles        *
            31      ;* (15 DCLs) before the occurrence of the desired            *
            32      ;* octet. The data byte is read from port bit                *
            33      ;* P3.0, 3--4 DCLs after the occurrence of the desired       *
            34      ;* octet. The data byte is stored in external memory         *
            35      ;* and the data pointer to external memory                   *
            36      ;* is incremented after the byte is received                 *
            37      ;* over the IOM interface. The subroutine is                 *
            38;* exited before the next compare event.                          *
            39      ;*************************************************************
005B        40      Org 5BH           ;Compare 1 interrupt routine.
            41
005B        42      Write_IOM_Subroutine:;4 m/c for servicing the routine.
            43
RM C2BB     44      CLR IEN1.3        ;7th m/c into the frame.
45
RM 00       46      NOP               ;8th m/c into the frame.
RM 00       47      NOP               ;9th m/c into the frame.
RM 00       48      NOP               ;10th m/c into the frame.
RM 00       49      NOP               ;11th m/c into the frame.
RM 00       50      NOP               ;12th m/c into the frame.
RM 00       51      NOP               ;13th m/c into the frame.
RM 00       52      NOP               ;14th m/c into the frame.
RM 00       53      NOP               ;15th m/c into the frame.
RM 00       54      NOP               ;16th m/c into the frame.
RM 00       55      NOP               ;17th m/c into the frame.
RM 00       56      NOP               ;18th m/c into the frame.
RM 00       57      NOP               ;19th m/c into the frame.
```

```
IOMRA       IOM     Interface          Appendix A

RM 00       58      NOP                ;20th m/c into the frame.
RM 00       59      NOP                ;21th m/c into the frame.
RM 00       60      NOP                ;22th m/c into the frame.
RM 00       61      NOP                ;23th m/c into the frame.
RM 00       62      NOP                ;24th m/c into the frame.
RM 00       63      NOP                ;25th m/c into the frame.
RM 00       64      NOP                ;26th m/c into the frame.
RM 00       65      NOP                ;27th m/c into the frame.
RM 00       66      NOP                ;28th m/c into the frame.
RM 00       67      NOP                ;29th m/c into the frame.
RM 00       68      NOP                ;30th m/c into the frame.
RM 00       69      NOP                ;31th m/c into the frame.
RM 00       70      NOP                ;32th m/c into the frame.
            71
RM 00       72      NOP                ;New octet.
RM A2B0     73      MOV C,P3.0         ;First bit read.
RM 13       74      RRCA               ;First bit into the acc.
RM 00       75      NOP                ;36th m/c inth the frame
            76
RM 00       77      NOP                ;37th m/c into the frame.
RM A2B0     78      MOVC,P3.0          ;Second bit read.
RM 13       79      RRCA               ;Second bit into the acc.
RM 00       80      NOP                ;40th m/c into the frame.
            81
RM 00       82      NOP                ;41st m/c into the frame.
RM A2B0     83      MOVC,P3.0          ;Third bit read.
RM 13       84      RRCA               ;Third bit into the acc.
RM 00       85      NOP                ;44th cycle.
            86
RM 00       87      NOP                ;45th m/c into the frame.
RM A2B0     88      MOVC,P3.0          ;Fourth bit read.
RM 13       89      RRCA               ;Fourth bit into the acc.
RM 00       90      NOP                ;48th machine cycle.
            91
RM 00       92      NOP                ;49th m/c into the frame.
RM A2B0     93      MOVC,P3.0          ;Fifth bit read.
RM 13       94      RRCA               ;Fifth bit into the acc.
RM 00       95      NOP                ;52nd cycle.
            96
RM 00       97      NOP                ;53rd machine cycle.
RM A2B0     98      MOVC,P3.0          ;Sixth bit read.
RM 13       99      RRCA               ;Sixth bit into the acc.
RM 00       100     NOP                ;56th machine cycle.
            101
RM 00       102     NOP                ;57th machine cycle.
RM A2B0     103     MOVC,P3.0          ;Seventh bit read.
RM 13       104     RRCA               ;Seventh bit into the acc.
RM 00       105     NOP                ;60th machine cycle.
            106
RM 00       107     NOP                ;61st machine cycle.
RM A2B0     108     MOVC,P3.0          ;Eighth bit read.
RM 13       109     RRCA               ;Eighth bit into the acc.
            110
RM A3       111     INCDPTR            ;Increment the data pointer.
RM F0       112     MOVX@DPTR,A        ;Save the acc. at the next byte.
RM A883     113     MOVR0,DPH          ;To compare if end of array
RM B8FF08   114     CJNER0,#0FFH,CONTI;Compare the high byte first.
RM A882     115     MOVR0,DPL          ;Load the lower byte for comparison.
```

```
IOMRA          IOM       Interface          Appendix A

RM B8FF03      116       CJNE R0,#0FFH,CONTI;Compare the lower byte.
               117
00A9 900000    118       MOV DPTR,#00H      ;Initialize the data pointer.
               119
00AC           120       CONTI:
00AC C2C3      121       CLR IRCON.3
00AE D2BB      122       SETB IEN1.3
00B0 32        123       RETI
               124
               125
               126
               127       ;*********************************************************
               128       ;* Read_IOM_Set-Up:                                      *
               129       ;* In this program the registers are set up to           *
               130       ;* allow compare to happen at the "count" value. The     *
               131       ;* PTRA unit of the SAB 80535 is programmed in the       *
               132       ;* compare mode 1. The "count" value is assigned         *
               133       ;* to the compare register 1 and the corresponding       *
               134       ;* interrupt is enabled. The Timer 2 of the PTRA         *
               135       ;* unit is programmed to restart at a value of FF80H     *
               136       ;* on the arrival of the falling edge of the             *
               137       ;* FSC signal.                                           *
               138       ;*********************************************************
               139
0100           140       ORG 100H
0100           141       Main_Begin:
RM 900000      142       MOV DPTR,#00H      ;DPTR = (Start of data array).
RM 75C100      143       MOV CCEN,#00H      ;Disable compare mode.
RM 75B0FF      144       MOV P3,#0FFH       ;SDO, SDI lines are high.
               145
0109 75C3FF    146       MOV CCH1,#0FFH     ;Compare register has a value
010C 75C2A2    147       MOV CCL1,#Count    ;corresponding to the octet desired.
010F 75CBFF    148       MOV CRCH,#0FFH
0112 75CA80    149       MOV CRCL,#080H ;Reload = 0FFFFH - 128 (32x4)
0115 75CDFF    150       MOV TH2,#0FFH
0118 75CC81    151       MOV TL2,#81H       ;To avoid compare at FF80H
011B D2CA      152       SETB T2CM          ;T2 in compare mode 1
011D 75C108    153       MOV CCEN,#08H      ;Compoare mode enabled
0120 D2AF      154       SETB EAL           ;Master interrupt Bit enabled.
               155
0122 758910    156       MOV TMOD,#10H      ;Timer1 to cause more than
0125 758DFF    157       MOV TH1,#0FFH      ;quarter of a frame delay
0128 758BD0    158       MOV TL1,#256-48    ;to avoid match before reload.
               159
012B           160       Start:
012B 2096FD    161       JB P1.6,$          ;Prepare for start pulse arrival.
012E 3096FD    162       JNB P1.6,$         ;Wait for the start pulse, 1 m/c.
0131 D28E      163       SETB TR1           ;So far 2 m/c into the frame.
0133 308FFD    164       JNB TF1,$          ;Wait for another 49 m/c
               165
0136 75C81D    166       MOV T2CON,#1DH     ;T2 auto reload from P1.5.
0139 D2BB      167       SETB IEN1.3        ;Compare interrupt 1 enable.
               168
013B 213B      169       AJMP $             ;Wait for a match.
               170
               171       end

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

Memory Access Using the Eight Data Pointers of the SAB 80C517/80C537

SAB 80C517/80C537

8

Application Note

Memory Access Using the Eight Data Pointers of the SAB 80C517/80C537

The SAB 80C517/80C537 has eight 16-bit data pointers (DPTR) as opposed to one in other derivatives of the SAB 8051 family of microcontrollers. This allows a fast and efficient transfer of data bytes to, from and between program and external data memory spaces. An application of this feature and the resulting increase in processing power is illustrated in the following example. A comparison is made between a block transfer of data using the 8051 and the 80C517.

The microcontrollers based on the 8051 architecture have the capability to expand their memory addressing by going to external program and data memory spaces. This feature is quite useful in applications that require more memory space than furnished by the on-board ROM and RAM. However, data in the program memory and data in the external memory can only be accessed through indirect addressing using a 16-bit register called the Data Pointer (DPTR). The following instructions are used to access data or memory mapped peripheral addresses using the Data Pointer:

```
MOVC    A,@A+DPTR   ;Read from
                    Program memory
MOVX    A,@DPTR     ;Read
                    external Data
                    memory
MOVX    @DPTR,A     ;Write
                    external Data
                    memory
```

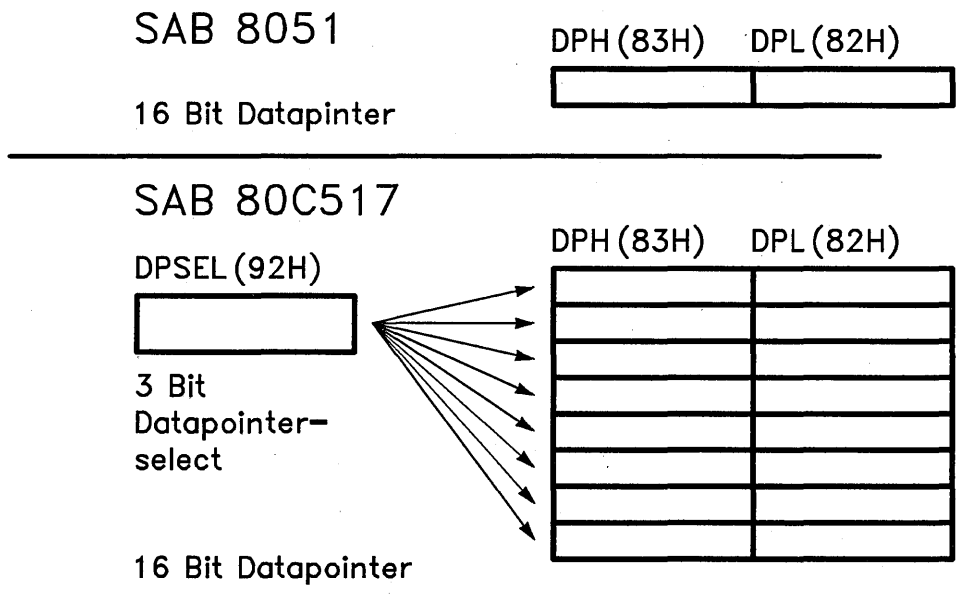
Almost all microcontrollers based on the 8051 architecture have only one such data pointer. This puts a heavy toll on data transfers requiring access to blocks of data stored at different addresses within the memory space. For example, in order to transfer a block of data from one address to another, one will first have to save the source address before using the Data Pointer for the destination address and vice versa. This operation not only slows down the data access as a certain amount of time is spent in saving and retrieving the data pointer but also uses the valuable internal RAM memory space to save the Data Pointer. The limitation of the 8051 architecture with one on-board Data Pointer is soon realized when the application requires multiple data transfers in the shortest possible time or when applications require access to a number of external peripherals and/or data memory locations. To some extent this limitation is eliminated in the SAB 80C517/ 80C537 by providing eight on-chip Data Pointers.

In order to keep the compatibility of the SAB 80C517/80C537 with the 8051, whose instruction set allows the handling of one single 16-bit Data Pointer only, it is not possible to add eight Data Pointers with eight different addresses. Instead all eight Data Pointers of the SAB 80C517/80C537 have the address 83H for the High-Byte (DPH) and 82H for the Low-Byte (DPL). Although the user's program may store up to eight different 16-bit addresses in these eight Data Pointer registers, only one register at any given time is active as a Data Pointer. The Data Pointer in use is selected by another special function register called DPSEL (Data Pointer SELect register). The bit 0 to 2 of this register DPSEL selects one of the eight Data Pointers. The selected Data Pointer can then be loaded, read and manipulated in the usual manner. It can also be used with MOVC/ MOVX types of instructions. Only the selected Data Pointer gets affected while the other seven Data Pointers remain unchanged. However, to activate any other Data Pointer, it is only sufficient to reprogram the lower three bits of the DPSEL register. This implementation guarantees the complete compatibility with the 8051 architecture for the Data Pointer in use.

The following programming example demonstrates the increase in processing power that results from using the additional Data Pointers of the SAB 80C517/80C537.

This program copies a block of data (e.g., a set of parameters) from the program memory (ROM) to the external data memory (RAM).

Figure 59
Data Pointer Register in the 8051-Type Microcontroller vs. the Data Pointer Array in the SAB 80C517/80C537



8

Transfer Program for the 8051

Figure 60
Program Listing using the 8051-Type Microcontroller vs the Program Listing using the SAB 80C517/80C537

```
;DECLARATION OF THE ARRAYS
     CSEG AT 800H                          ;ROM!
     PAR_ARRAY: DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
                DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
                                           ;.....etc.
                DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
END_OF_ARRAYEQU $
     XSEG AT 0000H ;EXTERNAL RAM!
     BUFFER:DS 512
;-------------------------------------------------------------------------------
DECLARATION OF THE SHADOW DATAPOINTERS
     DSEG AT 40H
SHADOW_DPL_0;    DS 1
SHADOW_DPH_0;    DS 1

SHADOW_DPL_1;    DS 1
SHADOW_DPH_1;    DS 1
;-------------------------------------------------------------------------------
     CSEG AT 00H
;LOAD THE    SHADOW DATAPOINTERS
     MOV     SHADOW_DPL_0,#LOW(PAR_ARRAY)
     MOV     SHADOW_DPH_0,#HIGH(PAR_ARRAY)
     MOV     SHADOW_DPL_1,#LOW(BUFFER)
     MOV     SHADOW_DPH_1,#HIGH(BUFFER)
LOOP: MOV    DPL,SHADOW_DPL_0              ;LOAD THE ACTUAL DPTR WITH
     MOV     DPH,SHADOW_DPH_0              ;POINTER TO PAR_ARRAY
     MOV     A,DPL                         ;CHECK FOR END_ADDRESS
     CJNE    A,#LOW(END_OF_ARRAY),GO_ON
     MOV     A,DPH
     CJNE    A,#HIGH(END_OF_ARRAY),GO_ON
     LJMP    END_LOOP
GO_ON:
     CLR     A
     MOVC    A, A+DPTR                     ;READ ELEMENT FROM PAR_ARRAY
     INC     DPTR                          ;AND INCREMENT POINTER
     MOV     SHADOW_DPL_0,DPL              ;SAVE THE ACTUAL DPTR
     MOV     SHADOW_DPH_0,DPH              ;TO SHADOW_POINTER
     MOV     DPL,SHADOW_DPL_1              ;LOAD THE ACTUAL DPTR WITH
     MOV     DPH,SHADOW_DPH_1              ;POINTER TO BUFFER
     MOVX    DPTR,A                        ;WRITE ELEMENT TO EXT. RAM
     INC     DPTR                          ;AND INCREMENT POINTER
     MOV     SHADOW_DPL_1,DPL              ;SAVE THE ACTUAL DPTR
     MOV     SHADOW_DPH_1,DPH              ;TO SHADOW_POINTER
     SJMP    LOOP
END
```

Transfer Program for the 80C517

Figure 60
Program Listing using the 8051-Type Microcontroller vs the Program Listing using the SAB 80C517/80C537 (continued)

```
;DECLARATION OF THE ARRAYS
     CSEG AT 800H                          ;ROM!
PAR_ARRAY:  DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
            DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
                                           .....
            DB 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
END_OF_ARRAYEQU $
     XSEG AT 0000H                         ;EXTERNAL RAM!
BUFFER:     DS 512
;------------------------
     CSEG AT 00H
;LOAD THE ACTUAL DATAPOINTERS
     MOV    DPSEL,#00H                     ;SELECT AND LOAD POINTER
     MOV    DPTR,#PAR_ARRAY                ;TO PAR_ARRAY
     INC    DPSEL                          ;SELECT AND LOAD POINTER
     MOV    DPTR,#BUFFER                   ;TO BUFFER
;------------------------
LOOP ;MOV   DPSEL,#00H                     ;SELECT FIRST DPTR (POINTER
                                           ;TO PAR_ARRAY)
     MOV    A,DPL                          ;CHECK FOR END ADDRESS
     CJNE   A,#LOW(END_OF_ARRAY),GO_ON
     MOV    A,DPH
     CJNE   A,#HIGH(END_OF_ARRAY),GO_ON
     LJMP   END_LOOP
GO_ON:
     CLR    A
     MOVC   A, A+DPTR                      ;READ ELEMENT FROM PAR_ARRAY
     INC    DPTR                           ;AND INCREMENT POINTER
     INC    DPSEL                          ;SELECT THE SECOND DPTR (POINTER
                                           ;TO BUFFER)
     MOVX   DPTR,A                         ;WRITE ELEMENT TO EXT. RAM
     INC    DPTR                           ;AND INCREMENT POINTER
     SJMP   LOOP
END
```

8

Program Description

First the memory spaces utilized are declared in the program—an operation which is the same for both the 8051 and the 80C517 program. This is done using the assembler directives CSEG for source data field in the ROM and XSEG for destination data field in the external RAM. However, in the 8051 program, internal RAM space needs to be reserved to allow saving of either the source or the destination pointer when one of them is not in use. This is implemented using the assembler directive DSEG. As the data pointer is a 16-bit register, two 8-bit locations are needed for both the source and destination pointers. These memory locations have the symbols SHADOW_DPL_0 and SHADOW_DPH_0 and SHADOW_DPL_1 and SHADOW_DPH_1 for the source and the destination pointers respectively. This procedure of memory declaration is however not needed in the 80C517 program.

Following the declaration is the program itself which in the case of the 8051 requires the loading of the shadow data pointers into the internal RAM whereas in the 80C517 program the source and the destination addresses are directly written into the Data Pointer array. In the data transfer subroutine of both the programs the initial part is quite similar in which it is checked whether or not the byte transferred is the last byte in the source array. Following this check is the actual transfer in which the byte read from source array in ROM is written into the appropriate location in the destination array in the external RAM. However, the way this transfer is handled by the 8051 routine is different from that of the 80C517 routine.

In the 8051 program, the Data Pointer must be loaded with the appropriate address from the shadow data pointers in the internal RAM, prior to each read and write operation. And after every read and write operation this address must again be saved in these shadow data pointers. This operation of loading and saving is completely eliminated in the 80C517 because of the abundance of the Data Pointer registers. However, in the 80C517 prior to every read and write operation the desired Data Pointer is selected by programming the DPSEL register.

The differences are obvious: In the 8051 program two bytes of memory space per address which need to be saved are needed in the internal RAM (4 bytes in the present example). However, using the 80C517 up to eight addresses can be directly stored in the new data pointer array, thus not occupying any space in the internal RAM. In addition the transfer of data is less time consuming with the 80C517. Most of the time spent in this routine is the time in performing the actual transfer of bytes. The transfer time increases as the number of bytes to be transferred increases. The 8051 program requires lots of time for loading and storing the Data Pointer. In 80C517, however, only the selection of the desired data pointer is necessary. One loop execution time in the 8051 program is 30 machine cycles while 80C517 for the same loop requires only 17 machine cycles; almost twice as fast.

Oscillator Design Considerations for SAB-51 Family of Microcontrollers Operating at Frequencies Higher than 16 MHz

SAB-51 Family

8

Application Note

For the SAB-51 (Mymos Technology only) family of microcontrollers specified to run at frequencies 18 and 20 MHz, the following care must be taken when they are used at these frequencies.

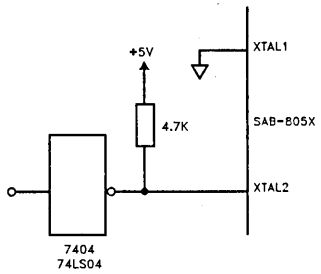
Problem

To use the on-chip oscillator, a crystal or ceramic resonator is connected between the XTAL1 and XTAL2 pins of the SAB-51 family of microcontrollers. In all the Mymos-based microcontrollers, XTAL1 is the input of the on-chip oscillator amplifier. XTAL2 is the output of this amplifier, driving the internal clock generator of all the microcontrollers. The gain of the on-chip oscillator amplifier decreases with increase in the crystal frequency, thus reducing the amplitude of the oscillation. This effect is more predominant at frequencies higher than 16 MHz. At crystal frequencies of 18 MHz and 20 MHz, the following work-arounds are recommended to have sufficiently high amplitude of oscillations.

Work Around #1

Using an external clock at XTAL2 pin (refer to "Figure 61 Driving from External Source") for driving the microcontrollers bypasses the on-chip oscillator amplifier, and the amplitude of the oscillations is then controlled by the external circuit. This is the best way to supply clock to microcontrollers at 18 MHz or 20 MHz clock oscillator frequency.

Figure 61
Driving from External Source



Work Around #2

If for some reason, work around #1 is not possible to implement, a crystal oscillator with a careful design can be used.

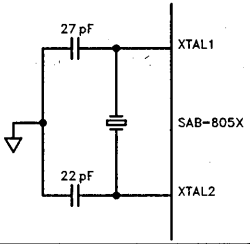
At such high frequencies, capacitive and inductive couplings between the oscillator circuitry and other signals are a major source of miscounts in the internal clocking circuitry. Surrounding the oscillator components with "quiet" traces (V_{CC} and ground, for example) will alleviate capacitive coupling to signals that have fast transition times. To minimize inductive coupling, the PCB layout should minimize the areas of the loops formed by the oscillator components. These are the loops that should be checked:

XTAL1 through the resonator to XTAL2;
XTAL1 through the capacitor to the V_{SS} pin;
XTAL2 through the capacitor to the V_{SS} pin;

The traces between the grounded ends of the capacitors and the V_{SS} pin should be kept as short as possible.

In addition, a higher amplitude at XTAL2 can be achieved by changing the ratio of the capacitors at XTAL1 and XTAL2. For frequencies higher than 16 MHz (18 MHz and 20 MHz), the configuration in "Figure 62 Crystal Oscillator Mode" increases the signal amplitude at XTAL2; the influence on the start-up time, however, is negligible. This is only a recommendation, the final design should however be verified against worst case conditions of temperature, V_{CC} levels, device tolerances, etc.

Figure 62
Crystal Oscillator Mode



Hardware-Power-Down In Use SAB 80C517A

8

Application Note

Data Saving With The SAB 80C517A In Power Down

In industrial and automotive fields, there is a strong need to save system information or data even while the main power supply is switched off. This information could be setup data, configuration data, error messages or other important data which must be available once the system restarts its normal operation.

There are basically two different methods to answer this demand. In method number one, data saving is achieved using "non-volatile" storage media, such as EEPROMs. This solution can make board designs rather uneconomical, due to the high price of such devices. Thus, this solution is normally used when a long data saving time is required or when there is no other alternative to buffer data. In this case the two above mentioned facts have a higher priority then the price level. The second solution is used more often. It is based on buffering the information with a weak standby supply which preserves the data after the main power supply is switched off.

With the growing availability of larger on-chip RAM on microcontrollers (SAB 80C515A with 1280-bytes, SAB 80C517A with 2304-bytes) this second way of data saving is becoming more and more economical for most board designs. The two following application examples therefore show two possibilities on how to save important data in the internal RAM of the SIEMENS line of high end 8-bit microcontroller.

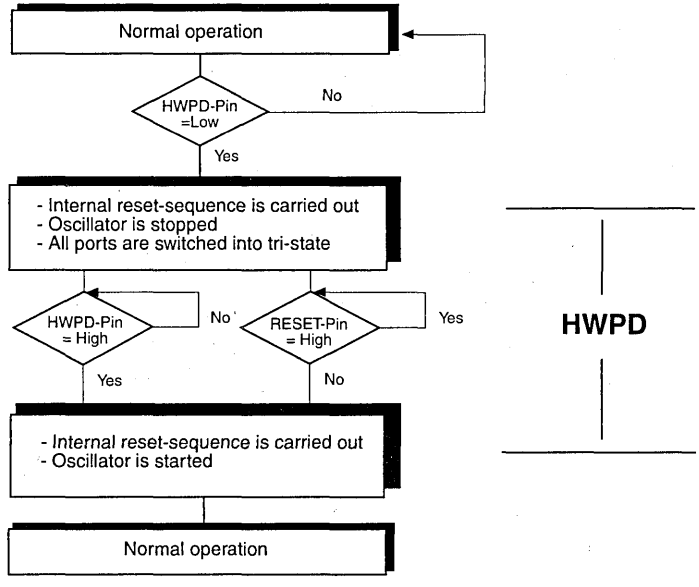
The SAB 80C517A and Its Power Down Modes

The heart of these two application examples is the microcontroller SAB 80C517A. Before getting into details of the first application example this paragraph gives a short technical overview of the SAB 80C517A.

It is an 8-bit microcontroller with powerful peripheral components such as Multiplication and Division Unit (MDU), Capture/Compare Unit, real 10-bit Analog Digital Converter (ADC), two Serial Interfaces, 68 I/O-lines, 2304-byte internal RAM, fail safe mechanisms and, last but not least, four power down modes. The latter are of particular significance to these application examples. During normal operation, the SAB 80C517A uses a supply current of lower than 28mA (fosz = 12MHz). The power down modes are used to reduce this consumption. Even in this mode the content of the internal RAM (2304-bytes) is preserved. The SAB 80C517A has in total four different types of power down modes, the Idle-mode (IDM), the Slow-down-mode (SDM), the Software-power-down (SWPD) and the Hardware-power-down (HWPD). The Idle-mode, Slow-down mode and the Software-power-down are software initiated. In the Idle-mode only the CPU is stopped and all peripheral components are still running. The ports keep their states. In the Slow-down-mode the controller runs for a certain time with a reduced speed (1/8 oscillator frequency). The last of the software initiated power down modes is the Software-power-down. In this mode not only the CPU is stopped, like in the Idle mode, but also the on-chip peripherals are stopped. During the SWPD the supply current is reduced, but the ports keep their last states active (can be inputs or outputs). This fact makes the estimation of the power consumption rather difficult because the port loads influence this value as well. For example, some of the port lines are defined as outputs before entering the SWPD, these lines are still output lines during the Software-power-down. For that configuration the port loads take their needed current from the controller. Therefore the supply current increases.

The Hardware-power-down is a power down mode which is initiated by an external hardware signal. For this mode the SAB 80C517A has one special input pin HWPD# (# = low active). A low signal at the HWPD# pin stops the running software and carries out the internal power down sequence (Figure 63 Internal Power-down Sequence). During this mode the SAB 80C517A is completely disconnected from

Figure 63
Internal Power-down Sequence



its periphery because all ports are in tri-state. Being independent of any port loads this powerful feature guarantees a supply current below 50uA.

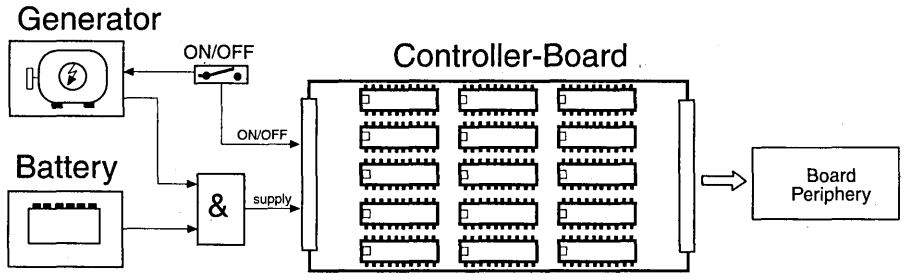
How to Enter and Leave the HWPD in the SAB 80C517A

To detect the power down request the microcontroller scans the level of the HWPD# pin in every machine cycle. If this pin is low, the internal power down sequence is executed. This sequence starts with a normal reset phase, then the internal oscillator is stopped and all ports are switched into tri-state. Now the SAB 80C517A is in power down. There are two possibilities to "wake up" the controller. The first is to set the HWPD# pin to logical high; the second is to set the reset pin (low-active). Furthermore the reset has a higher priority then the HWPD. In both cases the controller starts working with a normal reset sequence and runs the implemented software.

Application No. 1 - SAB 80C517A with the SIEMENS voltage regulator TLE 4258

Description of the system environment.

Figure 64
SAB 80C517A with the SIEMENS voltage regulator TLE 4258

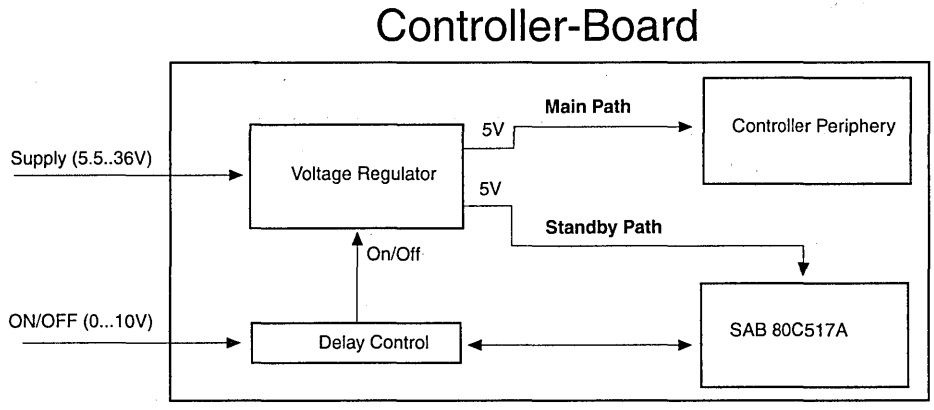


The controller-board is supplied from two external sources (generator and battery). During normal operation the generator is turned on and supplies the controller board. In the standby mode the generator is switched off and the battery serves the controller board. Additionally, the status line indicates the operation change.

To guarantee a long life span for the battery the most important point is to reduce the power consumption of the controller board to a minimum (automotive systems for example allow a minimum standby current of 1mA for the whole board).

In Figure 65 Structure of the Controller Board the structure of the controller board is demonstrated.

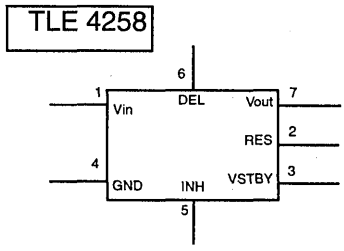
Figure 65
Structure of the Controller Board



Due to the requirement mentioned above, to reduce the power consumption to a minimum, the complete controller periphery must be switched off and the SAB 80C517A must still be supplied with 5V. The easiest way for this intention is to use a voltage regulator with two separate output paths (main path, standby path).

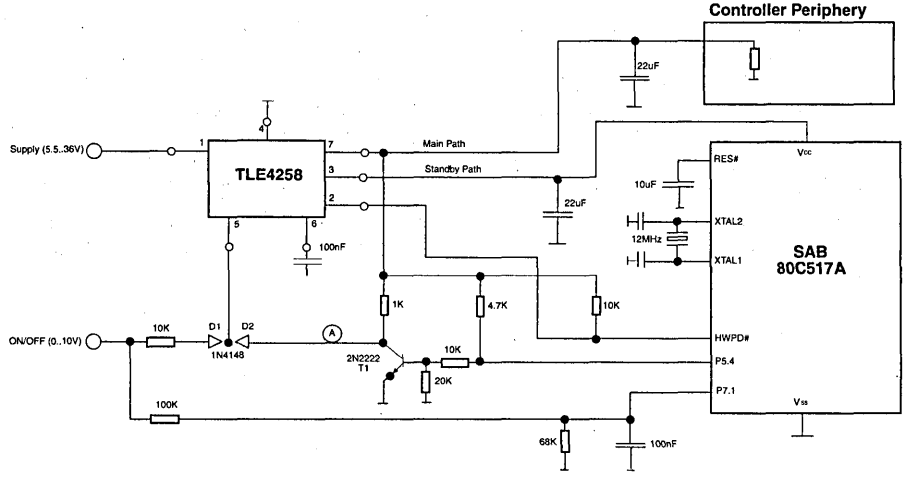
The main path (which can be switched on and off) to support the controllers periphery, and the standby path to serve only the SAB 80C517A in both operation modes (normal operation, power down mode). In this application example the SIEMENS low drop voltage regulator TLE 4258 is used (Figure 66 Application Example using the SIEMENS low drop voltage regulator TLE 4258).

Figure 66
Application Example using the SIEMENS low drop voltage regulator TLE 4258



Vin (pin 1): Input voltage 5.5V..36V
Res (pin 2): Reset pin (low active)
Vstby (pin 3): Standby path, Imax 35mA
GND (pin 4): Ground
INH (pin 5): Switch on/off (Vinh >3.6V --> switch on, Vinh < 3.6V --> off)
DEL (pin 6): Delay time for reset
Vout (pin 7): Main path, Vout = 5V+/- 15%, Imax = 300mA

Figure 67
Circuit Layout



The SAB 80C517A is directly supplied by the standby path of the TLE 4258. The maximum current at this path is determined by the microcontroller in normal operation mode. This current is lower than 28mA, if the SAB 80C517A is clocked with 12 MHz. In HWPD mode this current is reduced below 50uA. The HWPD# pin is connected to the reset pin of the TLE 4258, so an active reset signal (even in case of undervoltage!) of the voltage regulator causes the SAB 80C517A to go into Hardware-power-down mode.

As already explained, the main intention is to save important data in the internal RAM of the SAB 80C517A when the system is in standby mode. These data can be status/ setup information, error messages or other important system data. In order to write these data into the internal RAM the controller must know when there is an external need to force the controller board into standby-mode. For that purpose the status line is connected with one of the analog input pins (P7.1) to the internal Analog Digital Converter (ADC).

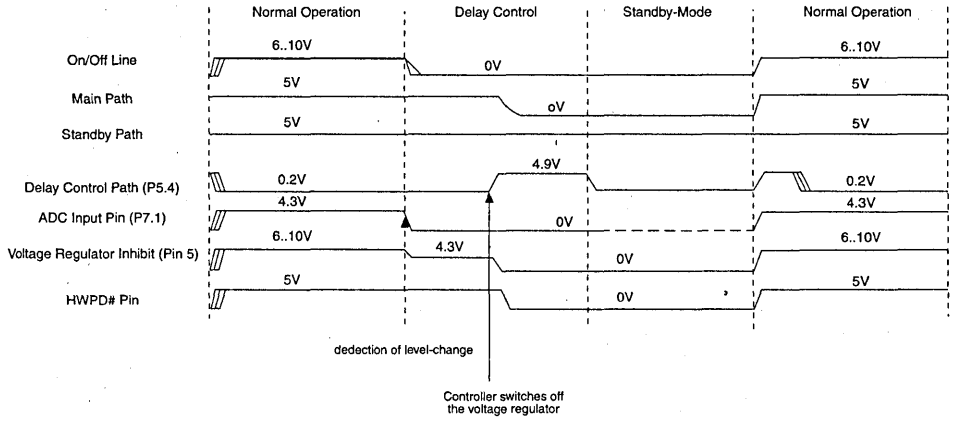
The connection is carried with two resistors and a capacitor to reduce the input voltage to 4V. The capacitor together with the 100K resistor works like a low-pass to avoid spikes at the ADC input pin. The software uses the internal ADC to monitor the status line. If its level is 4V the status line indicates that normal operation is to be continued. If it is well below 4V the controller recognizes that it should switch to standby mode. Under this condition there is no current flowing through the diode D1. Therefore now the state of the voltage regulator depends only on the state of the transistor T1. The circuit layout in Figure 67 Circuit Layout shows, that this transistor is controlled by the pin P5.4. If the controller outputs a low at this pin the transistor T1 is turned off. The voltage level is therefore 5V at point A. The regulator is then consequently still active. This control of the voltage regulator through diode D2 is called delay control. This delay control is required if the internal RAM of the SAB 80C517A does not contain the latest data when the status line becomes active. In this case the controller starts the updating sequence (e.g. reading data from external peripherals). After this is finished the SAB 80C517A turns off the regulator by outputting a high at P5.4. As a consequence the voltage of the main path drops down from 5V to 0V. The reset pin of the voltage regulator becomes active and pulls the HWPD# pin low as well. The voltage on the standby path is still at 5V and provides the power down current for the SAB 80C517A.

How to Switch on the Regulator

When the external generator starts working again, the status line becomes active and switches on the voltage regulator through diode D1. The voltage on the main path is then 5V again. This causes the reset pin of the TLE 4258 to go inactive. This level change will release the controller from hardware power down, because now there is a logical high at the HWPD# pin. After executing the internal reset sequence, the SAB 80C517A starts working. At the beginning of this startup the port pin P5.4 must be cleared to activate the delay control.

The Figure 68 Signal Timing of the Various Pins shows the context of these signal timing of the various pins.

Figure 68
Signal Timing of the Various Pins

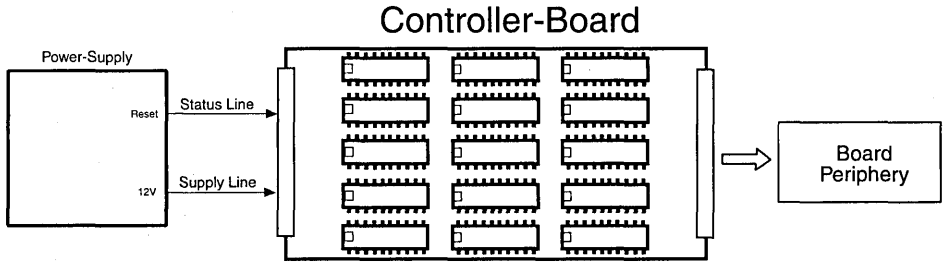


Application No. 2 - The SAB 80C517A in combination with the SIEMENS voltage regulator TLE 4261

The application example No. 1 described a system configuration with two supply sources (main power supply, standby supply) situated outside of the controller board. The supply switch between normal operation and standby was realized outside of the board as well. In application example No. 2 another possibility of data saving, with two separated supplies, is demonstrated.

The Figure 68 Signal Timing of the Various Pins shows the context of these signal timing of the various pins.

Figure 69
System Environment



Before starting with the details of the circuit layout, a brief description of the system environment is illustrated in Figure 69 System Environment.

In normal operation mode the controller board is served from a main power supply via a supply line. This power supply outputs 12V. If this supply is turned off it is indicated by a status line. As a consequence this status line becomes active (low active) and forces the controller board to go into the standby mode.

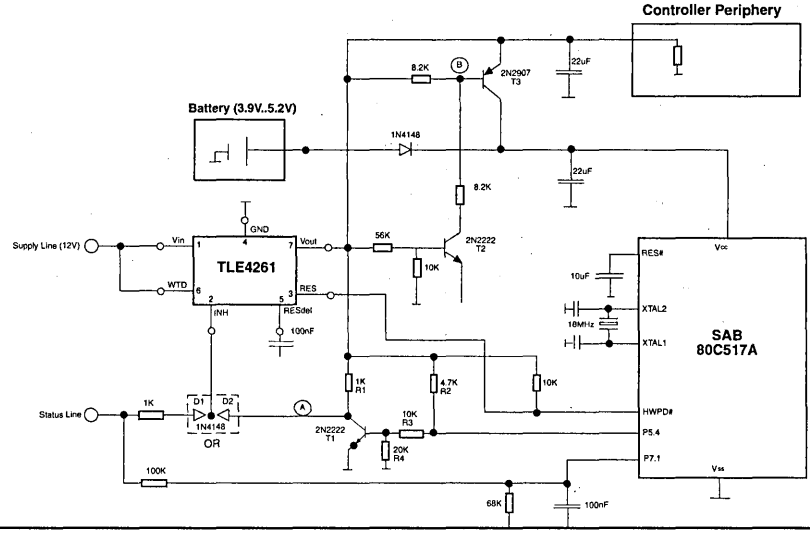
How the controller board manages the status/supply line

The core of this controller board is, like in the previous application example, the 8-bit microcontroller SAB 80C517A. The large on-chip RAM of this controller is used to buffer important data during standby mode. Because in standby mode the main power supply is switched off, there must be another source to provide the controllers standby current. This standby source for example a battery, is located on the controller board.

The supply line from the main power supply is connected with the input pin (pin 1) of the voltage regulator TLE 4261. As mentioned above this line has 12V.

8

Figure 70



In this design of the controller board the TLE 4261 is used to convert the 12V from the main power supply into 5V for the board. The TLE is a low drop voltage regulator with just one output path providing 5V +/- 5%. In the standby mode this regulator is switched off and only the SAB 80C517A is then supported by the battery. During this standby mode the controller is in Hardware-power-down (power consumption decreases below 50uA) and the content of the internal RAM is saved. When the main power supply is turned on again the status line goes to high and switches on the voltage regulator as well.

Description of the circuit layout

As in application example No. 1 the control of the voltage regulator is achieved with a delay control. This is also used in application example No.2 and consists of the transistor T1, the corresponding resistors R1, R2, R3 and R4 and the two diodes D1 and D2. These two diodes are representing an OR connection. The output of this OR serves the INH pin of the TLE 4261 (regulator on/off) and depends on the state of the transistor T1 and the status line. In normal operation the status line is approximately 10V. This results a voltage level of 9.3V at the INH pin. Under this condition the TLE 4261 is active and outputs 5V at pin 7. For activating the delay control, the SAB 80C517A clears the port pin P5.4. This turns off the transistor T1 and pulls the voltage level at point A to 5V. To monitor the status line and to detect whether there is a intention to switch off the main power supply, the internal 10 bit ADC of the SAB 80C517A is used. The status line is connected to one of the twelve analog input channels.

This is realized via two resistors (100K, 68K) and a capacitor (100nF). This kind of input protection reduces the voltage of the status line to 4V and avoids spikes at the ADC input pin (low-pass).

When the status line is driven to low (no current through D1) the TLE 4261 is still active, because the voltage level at point A is at 5V. From this moment on the voltage regulator is controlled solely by the port pin P5.4. The running software has now the possibility of delaying the "switch off" for the

TLE 4261 until all necessary data are written into the internal RAM. Once this write sequence is finished, the port pin P5.4 is set to high. Thus, the transistor is switched on and pulls the collector voltage down to 0V. This disables the regulator via the INH pin.

The main power supply

Before going on with the description of the circuit layout, a short look at the interdependency of the status line and supply line is provided. For this application example a power supply is used which indicates a power breakdown by setting the status line to low. The status line is also low when the output voltage falls under a defined limit. In normal operation the output voltage is 12V. Under this condition the status line is inactive (10V). If the output voltage falls under the limit of 9V (e.g. switch off or short connection) the status line indicates this by going to low. Because the output is buffered by a big capacitor, the changing of the output voltage is controlled by the following formula:

dt = (C * dU)/I dt = delta time [s]
C = capacity [F]
dU = delta voltage [V]
I = current [A]

However the problem is, that the voltage regulator TLE 4261 itself needs a minimum input voltage (pin 1) of 5.5V to guarantee an output voltage of 5V. Important now is the time range between the low at the status line and reaching the critical limit of 5.5V at the supply line. Exactly this time range is available for updating the internal RAM and also to go into hardware power down (described later). In the following an estimation of this range.

dt = (C * dU)/I

Typical values are: C = 10000uF, dU = 3.5V (9V - 5.5V), I = 1A

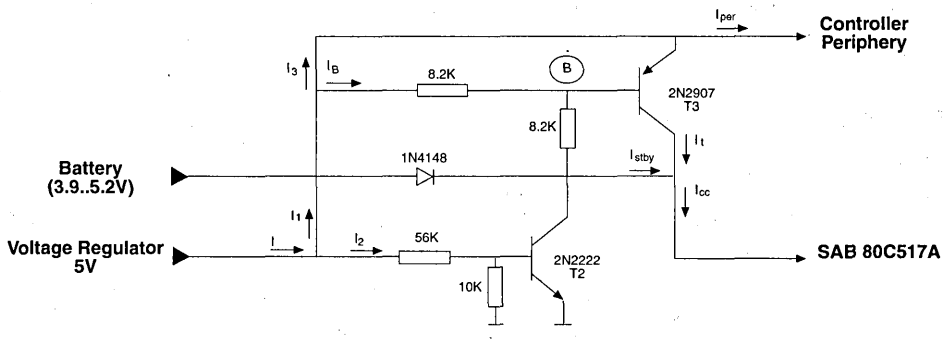
When these values are used for the formula, the result is a time span of 35ms. This is sufficient enough to carry out a complete update of the internal RAM (2304-bytes), and a few more lines of code (including the converting time of the Analog Digital Converter and jump into hardware power down).

Additionally, some bus systems (VME-bus, Multibus I and II) are providing an ACFAIL-Signal. This signal can also used as a status line. The time span of these signal is between 20ms and 40ms.

The supply switch

The main difference between application example No.1 and No.2 is the realization of the supply switch. In the previous example this unit was located outside of the controller board. In application example No.1 the used supply line was constantly active. In normal operation the board was served by a generator and in standby by a battery. But in application example No.2 the supply line is inactive in standby mode. Therefore the battery and the supply switch are located "on-board".

Figure 71



When and how does the supply switch become active

In normal operation mode the voltage regulator is active and the voltage output pin (pin 7) of the TLE 4261 provides currents I1 and I2. The slight current I2 is needed to switch on the NPN-transistor T2 (I2 approx. 80uA) and thus pull the voltage level down to 0.2V at point B. At the PNP-transistor T3 the current I3 is split into the current for the controller periphery (Iper) and the transistor current (It). Forced by the low level at the base of T3 an emitter base current causes an emitter collector current (It minus emitter base current). This current is the supply current for the SAB 80C517A (Icc) in normal operation mode. The supply current for the SAB 80C517A is lower than 37mA with a controller clock of 18MHz. In normal operation the standby current Istby is 0A.

When the voltage regulator TLE 4261 is switched off two things happen. Firstly the supply switch becomes active and secondly the SAB 80C517A must be forced into Hardware-power-down mode.

If the status line indicates the standby mode, the delay control becomes active and the TLE 4261 is turned off via the INH pin. As a result the output voltage of the regulator drops down from 5V to 0V (buffered by a 22uF capacitor). This occurs when the supply switch becomes active. Due to the absence of 5V at pin 7 of the TLE 4261, the transistor T2 is switched off (I2 = 0A) and results also I3 and IB to be 0A. Under this condition the transistor is turned off (It = 0) and the controllers periphery is also turned off. From this moment on the controllers supply current is provided by the battery (e.g. lithium battery). To guarantee a minimum of this supply current the SAB 80C517A must be in Hardware-power-down.

The branch into Hardware-power-down occurs in the following way. The reset pin of the TLE 4261 (pin 3) is connected with the HWPD# pin of the SAB 80C517A. A logical low at this line carries out the internal power down sequence of the SAB 80C517A (see Figure 63 Internal Power-down Sequence). In normal operation mode (TLE 4261 active) the reset pin of the TLE 4261 is inactive

(pulled up with 10Kohm) and the level at the HWPD# pin is also high. When the output voltage of the TLE 4261 drops below 4.7V the reset pin becomes active and forces the SAB 80C517A into Hardware-power-down. The same happens, if the regulator is switched off by the INH pin. As soon as the SAB 80C517A detects a low level at the HWPD# pin, it lasts approximately 1.8us until Hardware-power-down is reached. During this time the normal supply current (<37m)A is required. After this, the consumption is reduced below 50uA and all ports are switched into tri-state. From this moment on the controller is completely disconnected from its environment.

Finally some words about the standby source

For the standby source a lithium battery for example can be used. The typical cell voltage of this battery is 1.3V. To realize the data saving in the internal RAM of the SAB 80C517A in Hardware-power-down mode the supply voltage may not drop below 2V. This results the requirement of three (3.9V) or four (5.2V) cells. If a diode with a junction voltage of less than 0.6V is used, it is also possible to use only two cells (2.6V).

Their are a few things to know about the replacement of the lithium batteries when the main power supply is switched off. The SAB 80C517A addresses this concerns with starting a power on reset, because now the Vcc pin is supplied. In case two or three cells are used, the actual supply voltage does not reach the value for normal operation mode (Vcc = 5V +/- 10%). The SAB 80C517A is then in an undefined state. To avoid this, the main power supply must turned on immediately after the replacement of the battery.

Another possibility is to take four lithium cells (5.2V). If now the battery is replaced, the supply voltage of the SAB 80C517A is in the range of normal operation. Thus the power on is successful and the power down sequence is executed after the restart.

```
$DEBUG
$SYMBOLS
$ERRORPRINT
$DATE (JAN-25-91)
$NOMOD51
$INCLUDE (REG517.PDF)
;APPLICATION SAB80C517A IN HARDWARE-POWER-DOWN
;
;************** DECLARATION ***************************************
;
                CSEG AT RESET
                LJMP START
START:          CLR     EAL             DISABLE ALL INTERRUPTS
;
;************** TIMER 1 *******************************************
;
                MOV     TMOD,#10H       ;SET TIMER 1 MODE
                CLR     ET1             ; CLEAR TIMER 1 INTERRUPT
                CLR     TF1             ; CLEAR OVERFLOW BIT
                CLR     TR1             ; STOP TIMER 1
;
;************** A/D CONVERTER *************************************
;
                MOV     ADCON0,#01H     :CONFIGURATION OF THE A/D CONVERTER
                CLR     IADC            ; DISSABLE A/D INTERRUPT
;
;************** MAIN **********************************************
;
                ANL     P5,#0EFH        ; SET DELAY-CONTROL
I0001:          MOV     DAPR,#00H       ; START A/D CONVERTER
I0002:          JB      BSY,I0002       ; CHECK BUSY-FLAG
                MOV     A,ADDAT         ; READ VALUE
                MOV     B,#04H          ; DEFINE RANGE
                DIV     AB
                JZ      I0003           ; IF LOW-LEVEL JUMP TO I0003
                SJMP    I0001
I0003:          MOV     R4,#8DH         ; LOAD LOOP-COUNTER
                ANL     P5,#0FAH        ; SET MONITOR-BIT
I0005:          CLR     TF1             ; CLEAR OVERFLOW-FLAGT
                MOV     TL1,#00H        ; START-VALUE FOR TIMER 1
                MOV     TH1,#00H        ; ------------"----------
                SETB    TR1             ; START TIMER 1
I0004:          JNB     TF1,I0004       ; CHECK OVERFLOW-FLAG
                CLR     TR1
                DJNZ    R4,I0005
                ANL     P5,#0F0H        ; SET MONITOR-SIGN
I0006:          ORL     P5,#10H         ; CLEAR DELAY-CONTROL
                SJMP    I0006
                END
```

3-Phase Sine Wave Generation with the SAB 80C515A and SAB 80C517A

8

Application Note

3-Phase Sine Wave Generation with the SAB 80C515A and 80C517A

Almost every industrial application needs the generation of analog output voltages with variable frequency and level. Since, most of the time, additional control or monitoring tasks have to be performed, the simplest solution would be to have a single microcontroller covering both needs. In this case, the analog signal is generated using digital pulse width modulation (PWM). The generation of a symmetrical sinusoidal 3-phase system is one specific application example for the generation of analog voltages. Interesting target applications for such a 3-phase sine wave are controlling asynchronous motors (ASM) or the realization of uninterruptable power supplies (UPS). In the following description, methods of how to use the microcontrollers SAB 80C515A and the SAB 80C517A for generating a symmetrical 3-phase system are shown.

Features of the SAB 80C515A and the SAB 80C517A

The microcontrollers SAB 80C515A and SAB 80C517A are derived from their basic versions SAB 80C515 and SAB 80C517, respectively. They are backward compatible and yet show enhanced or new features resulting in higher performance (see Figures 72 and 73).

One speciality is the maximum clock input frequency (18 MHz) which means a 50% increase compared to the previous standard frequency of 12 MHz. The computational power of the SAB 80C517A is increased more than 10 times by the on-chip arithmetic co-processor. This Multiply/Divide Unit (MDU) executes multiply (16x16) and divide operations (32/16) as well as shift and normalize (32 bit) operations within 4 µsec. The normalize function of this math unit facilitates floating point arithmetic, a fact that is especially important in high-level language programming (e.g. C).

The ROM versions SAB 83C515A-5 and SAB 83C517A-5 offer 32 KBytes on-chip program memory, and the option to expand it externally up to 64 KByte. Even higher address ranges can be achieved through bank switching which is supported by special linkers/locators. The ROMless versions SAB 80C515A and SAB 80C517A need external program memory such as (EP)ROM. Other than that, the ROM and the ROMless parts do not differ. In the following text, the numbering SAB 80C515A and SAB 80C517A also applies for the ROM parts (unless otherwise noted).

The on-chip data memory of the SAB 80C515A is expanded by an extra 1024 Bytes, whereas the SAB 80C517A incorporates 2048 Bytes of internal data memory, in addition to the 256 Bytes in the basic types. This sums up to 1280 Bytes and 2304 Bytes, respectively. Thus, in most applications an external RAM is obsolete which helps to minimize board space and cut down system cost.

Processing of analog signals is supported by the integrated A/D converter with a real 10-bit resolution. The SAB 80C515A provides eight, the SAB 80C517A twelve multiplexed input channels. The total conversion time is less than 13 µsec, with the maximum total unadjusted error (TUE) being 2 LSBs. The ADC can be programmed to convert in single or continuous mode and can also be triggered externally.

For digital signal generation and measurement, the SAB 80C515A offers besides the two standard timers (T0 / T1) the Timer 2 block which provides 4 capture/compare channels. In the SAB 80C517A, this Timer 2 block is located in the Capture/Compare Unit (CCU) which also features a fast Compare Timer and up to 21 high speed outputs.

Serial communication, for example with a host or other microcontrollers within a multiprocessor network, is performed by one (SAB 80C515A) or two (SAB 80C517A) serial channels. Separate baudrate timers for each channel allow flexible baud rate

generation at almost any clock frequency without using any of the on-chip general purpose timers.

The SAB 80C515A / 80C517A are also designed to work in systems which require higher system reliability. Therefore, both a watchdog timer (WDT) and an oscillator watchdog (OWD) are implemented to increase system security. The WDT makes sure that the whole controller is reset if the software fails to clear it periodically when the correct software flow is deviated. The OWD in contrast, monitors the system clock. On detection of an oscillator failure (e.g. in the case of a broken crystal), it takes over and brings the controller into a reset state. It only releases the part after the clock has been recovered.

Finally, the system design engineer can choose between 4 power saving modes: slow-down mode, idle mode, software controlled power-down mode (SWPD) and the brand-new hardware controlled power-down mode (HWPD). One remarkable advantage of HWPD is that it can be invoked by applying a low level voltage, and exited by applying a high level voltage at a specific pin. The second advantage is that while in HWPD all port pins are floating (tri-state) which avoids sourcing current to external circuitry and helps to save power.

How to Generate Analog Voltages Using PWM

General Aspects

With digital PWM, a variable analog voltage level is achieved through variation of the duty cycle within a fixed time interval (PWM period TPWM, Figure 74). Integration over the interval TPWM with a low-pass filter leads to the average output voltage during TPWM. Depending on the length of the duty cycle, the level of the average voltage is a value between the digital supply voltages of the microcontroller (VSS and VV_{CC}).

Variation of the duty cycle from one interval to the next results in a variation of the average voltage level during each time interval. Thus, it is possible to change the output voltage step by step from one interval to the next. Appropriate variation of the duty cycle also allows to step-wise generate a sinusoidal voltage (see Figure 75). The sine wave values are stored in a table that is cyclically accessed. When the PWM intervals are short, and through filtering, a smooth sine wave can be produced. If the output voltage is not supposed to vary between VSS and V_{CC}, but instead between -V_{CC}/2 and +V_{CC}/2 (zero crossing), the DC voltage can be decoupled, e.g. with a serial capacitor.

The duty cycle t_D within TPWM can be calculated using the sine wave value u(i):

$t_D = T_{PWM}/2 + u(i) \times T_{PWM}/2$

where -1 <= u(i) <= +1 ; 0 <= i <=N (number of sample points)

Generally, you have free choice for the length of the PWM interval. However, there are limitations in selecting the range for the longest and the shortest time interval (lowest and highest PWM frequency fPWM). The lowest frequency depends on the quality of the low-pass filter or on the filtering capabilities of the motor connected to the PWM output. An additional requirement is the fact that many applications require fPWM to be supersonic (above the audible range). On the other hand, switching loss increases as the PWM frequency rises. From this point of view, engineers normally attempt to keep fPWM as low as possible. In addition, power switches have a maximum switching frequency (e.g. approximately 30 kHz for IGBTs).

Both amplitude and frequency of the sine wave to be generated can be varied.

The peaks of the sine wave depend on how much the rising edge of the duty cycle of a sine wave deviates from the center of T_{PWM}. This means that the amplitude can be

chosen to be between 0V and $V_{CC}/2$. In the above mentioned formula for tD, voltage value u(i) has to be exchanged with the new voltage value u'(i), which can be calculated as follows:

u'(i) = u(i) x a,

where a is the desired amplitude factor, which can be chosen to be between 0% and 100%.

Variation of the sine wave frequency can be achieved in three different ways:

1. Variation of time interval T_{PWM}

The shorter T_{PWM}, the shorter the sine wave period, that is the higher the sine wave frequency.

2. Variation of the number of sample points N

Sample points are the voltage values that represent the sine wave. The higher the number of sample points (while T_{PWM} remains the same), the longer the sine wave period, i.e. the lower the sine wave frequency. At the same time the resolution of the sine wave is increased.

3. Variation of repetition factor Z

By repeating every sample point once or several times, the sine wave length doubles, triples etc. (see Figure 76). Thus, the sine period can be varied without the need for more sample points (therefore saving memory space!). On the other hand, the smoothness of the sine wave becomes worse.

With these variation possibilities, the formula for the sine wave frequency is as follows:

$f_{SIN} = 1/T_{SIN} = 1 / (T_{PWM} \times N \times Z)$

T_{PWM}:	PWM interval
N:	number of sample points
Z:	repetition factor for each sample point

How to Generate a 3-Phase System

Every symmetrical 3-phase system consists of three sinusoidal voltages which are phase shifted by 120° from each other. Since a sine wave can be represented by a table of N sample points (using PWM), one "run through the table" corresponds to one sine wave period. A phase shift of an electrical angle of 120° therefore corresponds to a table shift of N/3, a phase shift of 240° corresponds to a table shift of 2N/3. Usually, N should be chosen such that it can be divided by 6. Thus, not only the number of shifts can be divided by 3 (you need three phases), but also the zero crossing point can then be included as a sample point. This is useful, because this way only one half of the wave needs to be sampled and stored in the table.

Conclusion: One sinusoidal phase can be represented by one single table with N sample points. The other two shifted phases are represented by the same table, but with the starting sample point being offset by N/3 and 2N/3, respectively.

Microcontroller Facilities for 3-Phase Generation

The Timer 2 in the SAB 80C515A and the SAB 80C517A

Figure 77 shows the principle diagram of the timer 2 (T2) block. The maximum input clock of this 16-bit wide, free running timer is 1/12 of the oscillator frequency. This corresponds to a time increment of 667 nsec at 18MHz. A programmable prescaler can divide the input frequency by 2 in the SAB 80C515A, which results in a maximum input frequency of $f_{OSC}/24$. The SAB 80C517A additionally allows the input frequency to be programmed to $f_{OSC}/48$ or $f_{OSC}/96$. Using the reload register, the overflow time of timer 2 can be varied. An interrupt is flagged after every T2 overflow.

Timer 2 works with 4 (SAB 80C515A) or 5 (SAB 80C517A) capture/compare registers

(CCx, x=0...3 or 4). Each of them is also 16-bits wide and controls one specific port pin (see Figure 77). These capture/compare registers can be programmed to have one of three functions:

1. Capture function

Upon a rising edge at the corresponding port pin, CCx captures the current T2 contents and flags an interrupt. Thus, the moment when the signal edge occurred can be measured very accurately.

2. Compare function

Upon a match of T2 and CCx, a compare event at the corresponding port pin is generated and an interrupt is flagged. The compare function is subdivided in 2 different modes:

2a) Compare mode 0 (PMW mode)

When the contents of T2 and CCx match, a HIGH level is output at the corresponding pin. Upon timer 2 overflow a LOW level appears at the corresponding compare output pin (Figure 78).

2b) Compare mode 1

In this mode, the output pin is physically disconnected from the internal port latch. Every write access to the corresponding latch does not affect the pin immediately. When T2 and CCx match, the value written to the latch appears at the pin. An overflow of timer 2 does not have any effects on the output pin.

3. Reload function

Register CC0 (also referred to as CRC, compare/reload/capture) can also be programmed to work as a reload register for T2. This means that after an overflow from FFFFH timer 2 does not re-start at 0000H, but at the value stored in register CRC.

For sine wave generation, the PWM mode (compare mode 0) is relevant. The falling edge at the pin after every T2 overflow characterizes the interval T_{PWM}. It can be varied by operating timer 2 in auto-reload mode. The value in the reload register CRC determines the overflow time of T2 and thus T_{PWM}. The rising edge which determines the duty cycle, can be selected by loading the compare register with the appropriate value.

Both a compare match and a timer overflow set their own interrupt flags. After every compare match, the software has to take care that the compare interrupt is disabled for the rest of the T_{PWM}. The reason is that after every match the compare value for the next PWM period has to be loaded into CCx. However, it must be insured that this new value does not cause another compare match in the current interval. This would be the case, if the new value (for the next timer cycle) is higher than the previous one (that just caused the compare match), and the timer is yet to reach this value in the current cycle. It is not so much that this would affect the level at the pin (it is outputting a HIGH level anyway), but it would flag another interrupt. In this case, yet the next compare value would be loaded into CCx without the current one causing the new interrupt (meant for the next timer cycle) having contributed to the PWM generation. Therefore, the compare interrupt should be disabled once it has been processed in a timer cycle. The timer overflow interrupt routine has to again enable the compare interrupt, of course only after having cleared the request flag that might have been set in the previous timer cycle.

The Compare Timer in the SAB 80C517A

The Compare/Capture Unit (CCU, see Figure 79) of the SAB 80C517A contains besides the timer 2, a second 16-bit ree running "compare timer" (CT). Up to 8 new compare registers (CM0 ... CM7) can be assigned to the CT, each of which have their dedicated output pins (alternative functions of port 4). The compare timer and the compare registers have been implemented especially to support PWM generation and therefore show some extraordinary features.

The maximum input clock for the compare timer is fosc/2. Thus, the increment time is 111 nsec (at f_{osc} = 18MHz). The input frequency can also be reduced by a 3-bit prescaler down to 1/128 of the oscillator frequency. The CT has its own auto-reload register CTREL which determines the overflow time and thus the PWM period TPWM.

An example shows how the fast CT can be used for PWM generation. At 9-bit resolution (CTREL = FE00H) the switching frequency f_{PWM} is:

$f_{PWM} = 1/T_{PWM} =$
$= 1 / [(10000H - FE00H) \times 111nsec] =$
$= 1 / (512 \times 111nsec) = 17.6$ kHz

A very important characteristic of the compare registers CM0 ... CM7 is their double register structure together with the timer overflow control loading (TOC, see Figure 80). The double register structure implies that the real register, which is constantly compared with the timer, is not accessible for the CPU. Whenever the CPU tries to write to one of the CMx registers via the internal bus, it instead accesses a shadow register. Its contents is only loaded to the actual compare register upon the CT overflow. This feature avoids the need to disable a compare interrupt and to re-enable it after the timer overflow (see chapter 4.2.4, "Generating the PWM Signals"). Thus, compare interrupts for PWM generation with registers CM0 ... CM7 are unnecessary and therefore not available.

Software Examples for 3-Phase Sine Wave Generation

General Aspects

The two application examples that are presented in the following sections, show one possible realization of a 3-phase sine wave system.

In the first example, timer 2 from the SAB 80C535/80C515 is used. This software solution can also run on the SAB 80C515A/ 83C515A-5 (A-versions) without changes. If the internal XRAM of the A-versions is to be used, it only needs to be initialized with a few instructions, and the memory address needs to be adjusted.

In the second example, the compare timer from the SAB 80C517A/83C517A-5 performs the PWM generation utilizing the internal XRAM. If the standard SAB 80C517/ 80C537 (non A-version) is to be used, the instructions for the XRAM initialization should be removed.

The microcontroller is connected to a terminal (z.B. VT100, 9600 Bd) via a serial link. This allows the user to communicate with the controller and select the following parameters: the number of sample points, the repetition factor, the PWM interval (represented by the timer reload value) and the desired amplitude factor.

The sample points are stored as hexadecimal values in a table located in the program memory. The resolution of the sample points is 8 bits (timer 2 example) or 16 bits (compare timer example). They are normalized relative to an amplitude of 100% and an 8- or a 16-bit wide PWM interval (100H or 10000H). However, only the sample points describing one half of the wave are actually stored. Their values have been reduced by the offset which marks the zero crossing.

Since the zero crossing line is represented by the value 80H (or 8000H), a value of 00H (or 0000H) corresponds to the minimum sine value at 0<198>, whereas the sine maximum at 90<198> is represented by the value 7FH (or 7FFFH). According to this, the positive half of the wave can be calculated by adding the offset value 80H (or 8000H) and the table value. The negative half of the wave in contrast, can be obtained by subtracting the table value from the offset value.

Example:

sin (0°) := 80H + 00H = 80H
(:= 8000H + 0000H = 8000H)
sin (90°) := 80H + 7FH = 8FH
(:= 8000H + 7FFFH = 8FFFH)
sin (270°) := 80H - 7FH = 01H
(:= 8000H - 7FFFH = 0001H)

The user is offered the choice of 4 tables. The number of sample points N in these tables can be divided by 6. The reason that N should be divisible by 3, is to realize an equal 120° shift for the three phases. At the same time, N should be divisible by 2 so that it is possible to cut the sine wave in two halves (the zero crossing point at N/2 should be an integer).

Program Structure

Other than the initialization routines, the program consists of three parts with different priority levels:

1. Communication with the user (low priority)
2. Table generation according to the chosen parameters (medium priority)
3. PWM generation and eventual table switching (high priority)

The principle function of the program is as follows:

The highest priority is assigned to the software section for PWM generation using the compare outputs (see Figure 81). This part is programmed to be non interruptable, since an interrupt might have fatal consequences for the application system. The compare values to be used are stored in the active table ("USED_TAB") which is located in the external RAM (non A-versions) or the XRAM (A-versions). Of course, for this purpose it is possible to use the standard 256 Bytes of internal data RAM, since the tables in these examples are relatively small.

The background task allows the user to select the various parameters. After final approval from the user, the microcontroller calculates the new table ("NEW_TAB") using the selected parameters and stores it at a different location in the XRAM than the table currently used. At a given time (after completion of a run through the table), the controller switches to the new table which is now regarded as the actual table. The memory location of the previous actual table is now available for the storage of a new table with new parameters.

Main Program

At the beginning, both the serial channel and the timer (timer 2 or compare timer) are initialized. The input of parameters is simulated which specify a sine wave with an amplitude factor of 0%.

Invoked subroutines: INIT (external module), START

After the initialization, the CPU enters a loop which asks for new parameters. After receiving the new set of parameters, the controller calculates the new table and stores it in the XRAM. However, it is made sure that the previously generated table has been used at least for one run (WAIT_FOR_SWITCH:).

Invoked subroutines: GET_PARAMETER, CHANGE_PAR

Communication (GET_PARAMETER)

The subroutine GET_PARAMETER controls the parameter input. The user has the choice of four normalized sine wave tables (N = 12, 18, 24, or 30), the amplitude factor (00H ... 100H), the repetition factor, and the PWM interval TPWM. The normalized sine wave tables are stored in the (EP)ROM under the label SINUS_TAB_12 (_18, _24, or _30) immediately behind the subroutine code. The corresponding pointers to these tables are stored under the label TAB_LOOK_UP immediately behind the tables.

At the end of GET_PARAMETER, the user is asked to confirm the input parameters ("Start New Waveform?").

Table Calculation (CHANGE_PARAMETER and GET_NEW_WAVE)

After the confirmation of the new parameters by the user, the subroutine CHANGE_PARAMETER calculates the new table (using the subroutine GET_NEW_WAVE) and stores it in the XRAM. When the timer 2 is used, the interrupt table has to be stored as well (see 4.2.4, "Generating the PWM Signals").

First, it has to find out which memory location is available for the new table (NEW_TAB_PTR), since the other location is currently being used for the PWM generation (USED_TAB_PTR). Then, the reload value for the timer is calculated according to the chosen value for the PWM period. The reload value determines the middle of the PWM interval corresponding to the sine value "zero" (ZERO_POINT). Third, the input amplitude factor is transformed from ASCII to the hexadecimal value. Finally, the length of the sine wave table (NEW_LENGTH) and its relative end address (NEW_H_ADDR) need to be provided, before GET_NEW_WAVE is invoked.

The actual calculation and the storage of the table values is done by GET_NEW_WAVE. Half of the normalized sine wave located in the (EP)ROM is fetched value by value, changed according the input parameters and stored in the XRAM. Now, this portion of the wave is expanded to a full wave with a positive and a negative half. The last (highest) table value is stored first (lowest address), since the 8051 architecture provides the instruction DJNZ (decrement and jump if not zero) which facilitates the down-counting of a variable. This is the reason, why the PWM generation is done "backwards" (from top address to bottom address; see 4.2.4 "Generating the PWM Signals").

Generating the PWM Signals

The highest prioritized task of the software is to provide in time the compare values for the PWM generation. The values need to be transferred from the table located in the XRAM to the shadow latches of the compare registers. Since the structures of the timer 2 and the compare timer as well as their assigned registers are different, this part of the application example is discussed separately for each timer in the following sections.

Generating the PWM Signals with the Timer 2 in the SAB 80C515(A) or the 80C517(A)

The overflow interrupt of the timer 2 has the duty to update the three pointers (R_PTR, S_PTR, T_PTR) to the next values in the current sine table. In case one of the pointers reaches the bottom of the table, it has to be set again to the top of the table (the table is run through from top to the bottom).

Whenever there is a request to switch to a new table (SWITCH_EN = 1), the new parameters (NEW_...) are incorporated. In this case, the values from the old table which had been loaded in the previous (just completed) timer cycle are still used for the current cycle. For the next cycle, the values are already loaded from the new table. The memory location of the old table is now available for the storage of a new table to be calculated. Figure 82 shows the flow chart of the timer 2 overflow routine.

Theoretically, there are three ways to load the compare registers:

a) Loading by timer 2 overflow interrupt

After every overflow, the timer 2 interrupt routine loads the compare registers for the PWM period that has just started (see Figure 83). The values are fetched directly from the table located in the XRAM.

This is the easiest way to implement the

loading. The disadvantage is that the access to the XRAM (via a MOVX instruction, e.g. MOVX A,@DPTR) takes relatively long, during which time no compare match can occur. This would mean that the rising edge of the PWM pulse could never be directly at the beginning of a timer cycle. The consequence is that it would be impossible to reach a duty cycle of 100% and thus the full amplitude of the sine wave.

b) Loading by the Compare Interrupts

Another possibility would be to have each compare interrupt routine load their own compare register. Since this has to be done very quickly, it makes sense to make available the compare values in "shadow latches" located in the standard internal data memory. Thus, the transfer time (667 nsec at 18MHz) and the execution time for the interrupt routine can be kept very short. In this case, the timer 2 overflow routine must take care that the compare values for the next (not the current!) PWM period are fetched from the XRAM and stored in the "shadow latches".

Since the loading of the compare registers is performed before the next timer cycle starts, the duty cycle gap described above is avoided. Still, each compare interrupt takes a certain execution time increases the CPU load.

c) Loading by Using "Interrupt Source Passing"

The principle of "Interrupt Source Passing" (ISR) is based on the method described in b). However, only one compare interrupt loads all three compare registers. The idea behind ISR takes into account that there is always one compare interrupt which occurs last in a timer cycle. The corresponding interrupt routine loads both its own and the other two compare registers. Since the phase shift between the last compare interrupt in the current TPWM and the first interrupt in the next TPWM is always 1/3 of the timer cycle (120<198> phase shift), the registers are certainly loaded before the next compare value is needed.

For speed reasons, even with ISP it is recommended to use "shadow latches" for the compare values which are pre-fetched from the XRAM by the timer overflow routine. It is necessary that the compare interrupt has a higher priority than the timer interrupt, because the last compare interrupt loads the actual needed values to the compare registers, whereas the timer interrupt only fetches the values for the "shadow latches". This means that a compare interrupt in process may last longer than the current timer cycle. It does not matter that the overflow routine is delayed, as long as the next values have already been loaded into the registers by the compare interrupt.

Which compare is going to occur last in the timer cycle, is determined by physics: if a 3-phase system is divided into 6 portions of 60<198> each, it is always the same phase within each 60<198>-portion that has the smallest sine value. This phase is also the one which has the rising edge of its duty cycle last within TPWM over the whole 60<198>-portion. This means that its compare match occurs last. Therefore, with every sine wave table, a corresponding interrupt table is always stored in the ROM which contains a reload byte for the interrupt enable register IEN1. Thus, only the last compare interrupt to occur is enabled and the other two are disabled.

Generating the PWM Signals with the Compare Timer in the SAB 80C517(A)

All tasks which would be performed by the timer 2 interrupt, must also be performed by the compare timer interrupt. This is true for the check to determine if a parameter switch is requested (and, if yes, to actually switch) as well as the pre-loading of the "shadow latches". One great feature of the compare registers CM0 ... CM7 (which are used with the compare timer) is that their structure provides exactly these "shadow latches" (see Figure 80). The loading of the actual compare registers with the values contained

in the "shadow latches" is done automatically by a compare timer overflow (TOC = timer overflow controlled). This TOC mechanism makes compare interrupts unnecessary. This is the reason, that compare interrupts for the CM0 ... CM7 registers have not been provided in PWM mode. In addition to the reduction of program execution time, the TOC feature allows the code size to be minimized. Figure 84 shows the flow chart of the compare timer routine.

Possible Modifications

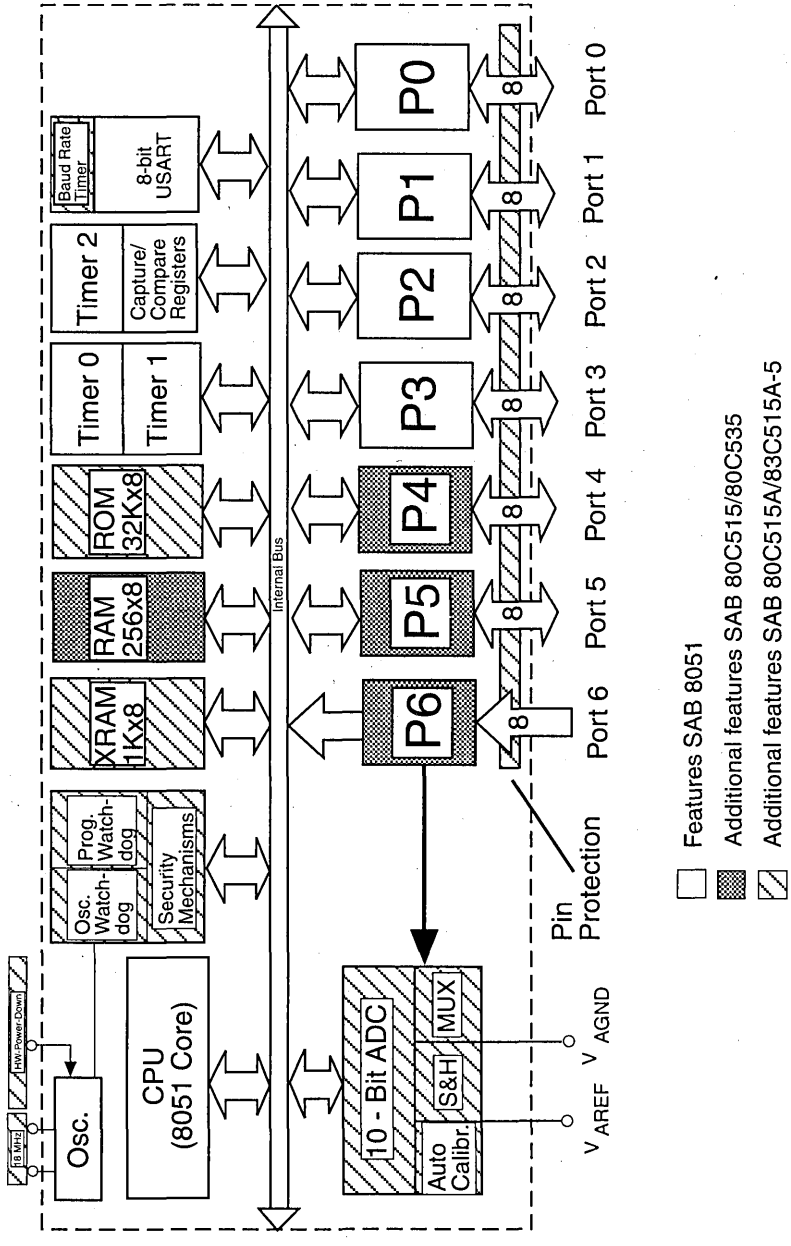
The presented program examples are only meant as application hints. They are not suited for the use in devices for series production. There are several possibilities to modify them in terms of motor speed ramp-up and/or table switch algorithms.

In these examples, a table switch is performed whenever there is a request to switch, no matter where in the table the phase pointers (R_PTR, S_PTR, T_PTR) point to, and the sine wave output starts from the top of the new table. This phase incontinuity might cause some ripples on the motor torque. They can be avoided, if the switching from one table to the next is done at a zero crossing of the sine wave or if the new sine wave is started at the same phase as the old wave ended.

The first case can easily be implemented (e.g. switch only, when R_PTR reaches the end of the table). In the second case, it would be necessary to calculate the relative position of one pointer within the table and take it into account at the initialization of the new pointer.

The annex contains listings of the two application examples, one using timer 2 of the SAB 80(C)515/80(C)535, the other using the compare timer in the SAB 80C517A/ 83C517A-5. Both programs can easily be run on the SAB 80C515A/83C515A-5 and the SAB 80C517/80C537. Only the XRAM initialization would have to be changed according to the target controller. No further changes are necessary.

Figure 72


Baud Rate Timer
8-bit USART
Timer 2
Capture/ Compare Registers
Timer 0
Timer 1
ROM 32Kx8
RAM 256x8
XRAM 1Kx8
Prog. Watch-dog
Osc. Watch-dog
Security Mechanisms
Internal Bus
P0
P1
P2
P3
P4
P5
P6
Port 0
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Pin Protection
CPU (8051 Core)
Osc.
HW-Power-Down
16 MHz
10 - Bit ADC
Auto Calibr.
S&H
MUX
V AREF
V AGND
Features SAB 8051
Additional features SAB 80C515/80C535
Additional features SAB 80C515A/83C515A-5

8

Figure 73

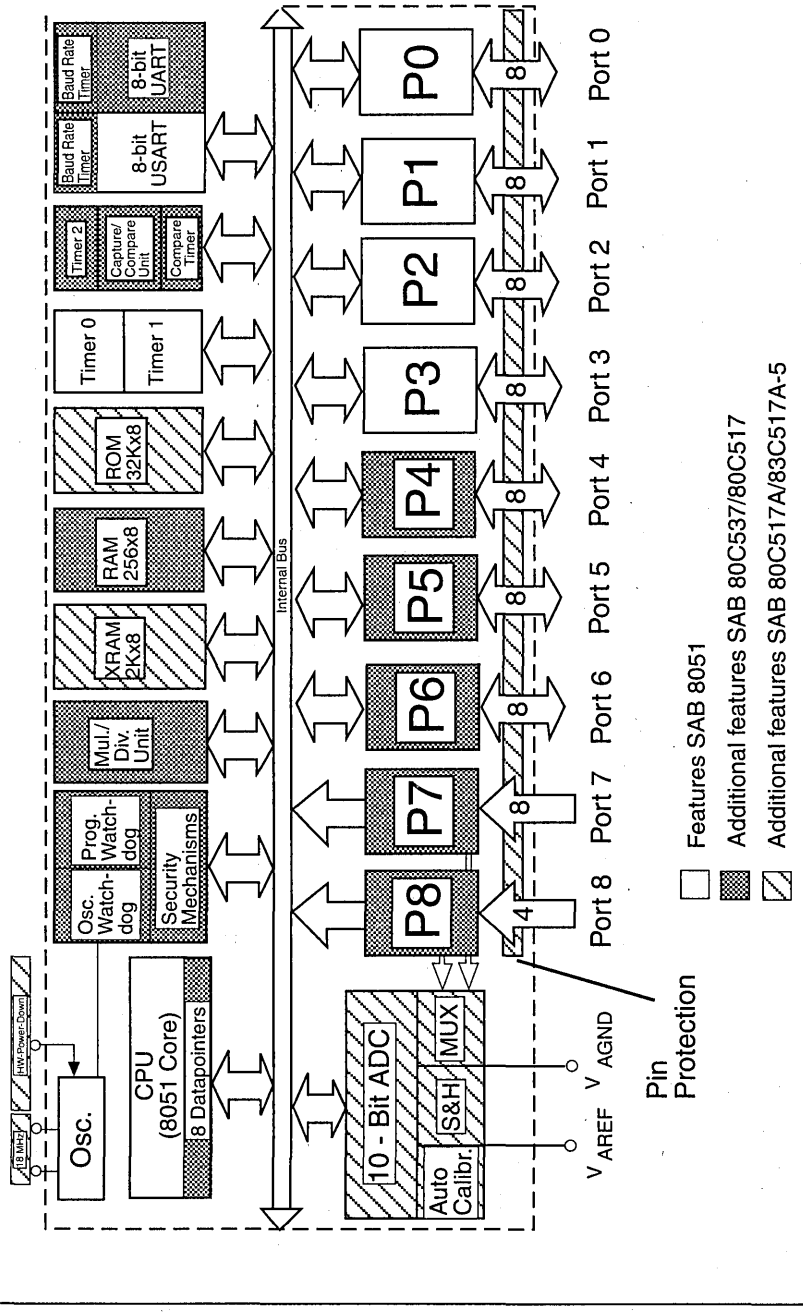
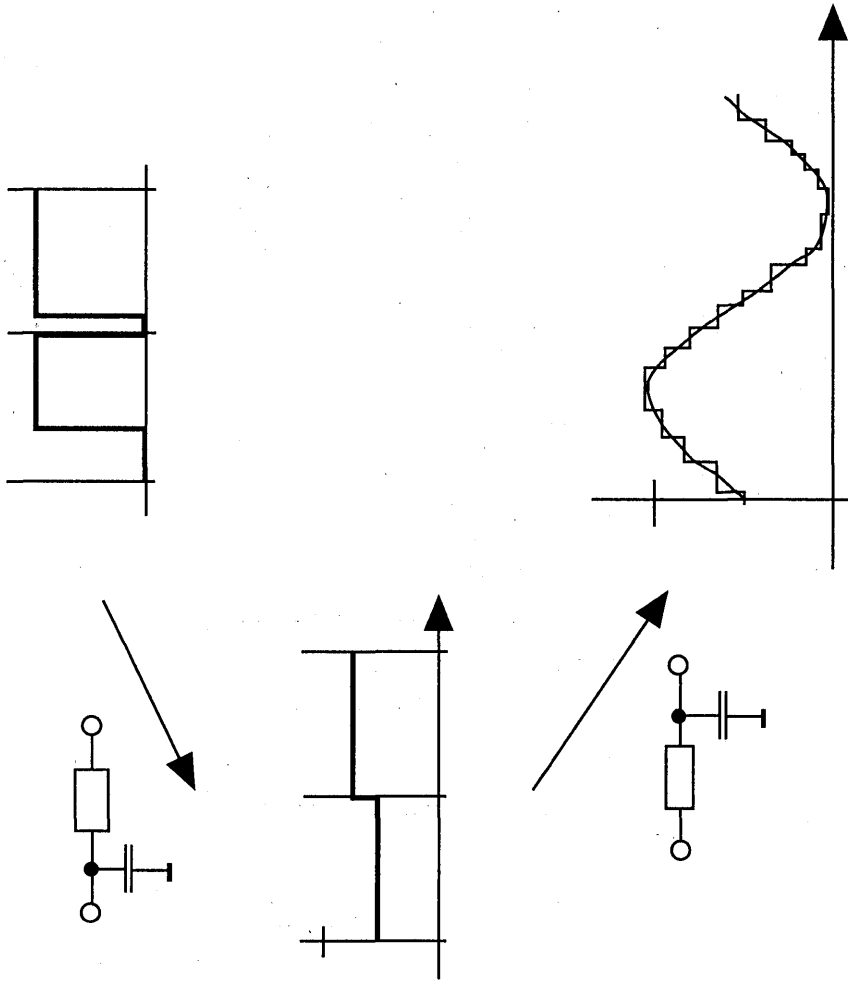

Osc.
(18 MHz)
HW-Power-Down
CPU
(8051 Core)
8 Datapointers
Osc. Watch-dog
Prog. Watch-dog
Security Mechanisms
Mul./ Div. Unit
XRAM 2Kx8
RAM 256x8
ROM 32Kx8
Timer 0
Timer 1
Timer 2
Capture/ Compare Unit
Compare Timer
Baud Rate Timer
8-bit USART
Baud Rate Timer
8-bit UART
Internal Bus
10 - Bit ADC
Auto Calibr.
S&H
MUX
P8
P7
P6
P5
P4
P3
P2
P1
P0
Port 8
Port 7
Port 6
Port 5
Port 4
Port 3
Port 2
Port 1
Port 0
V AREF
V AGND
Pin Protection
Features SAB 8051
Additional features SAB 80C537/80C517
Additional features SAB 80C517A/83C517A-5

Figure 74



8

Figure 75

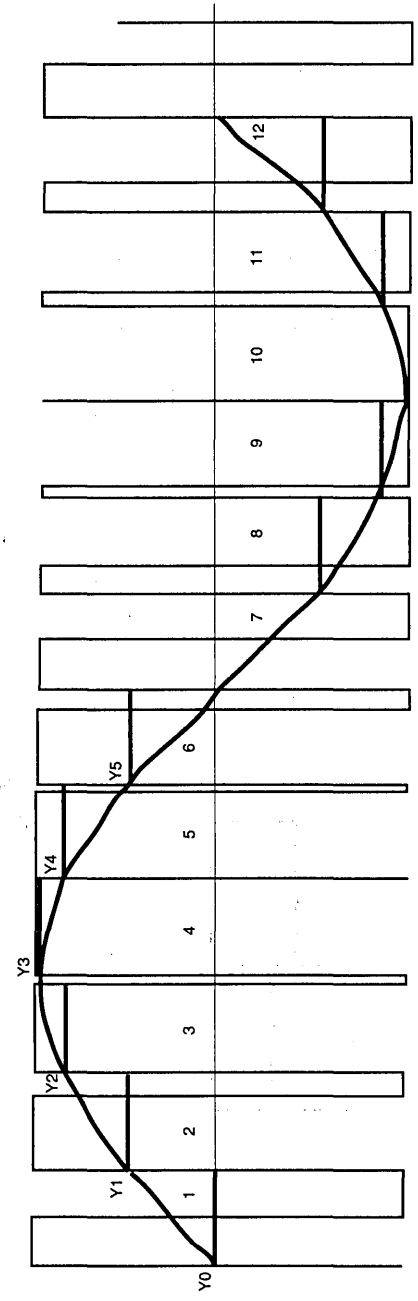
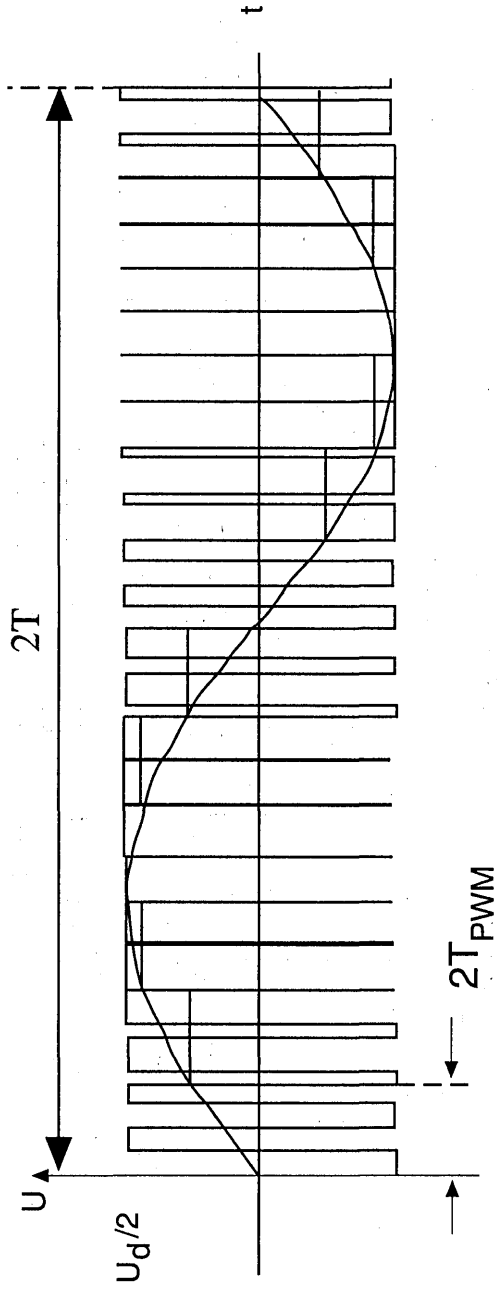


Figure 76



8

Figure 77

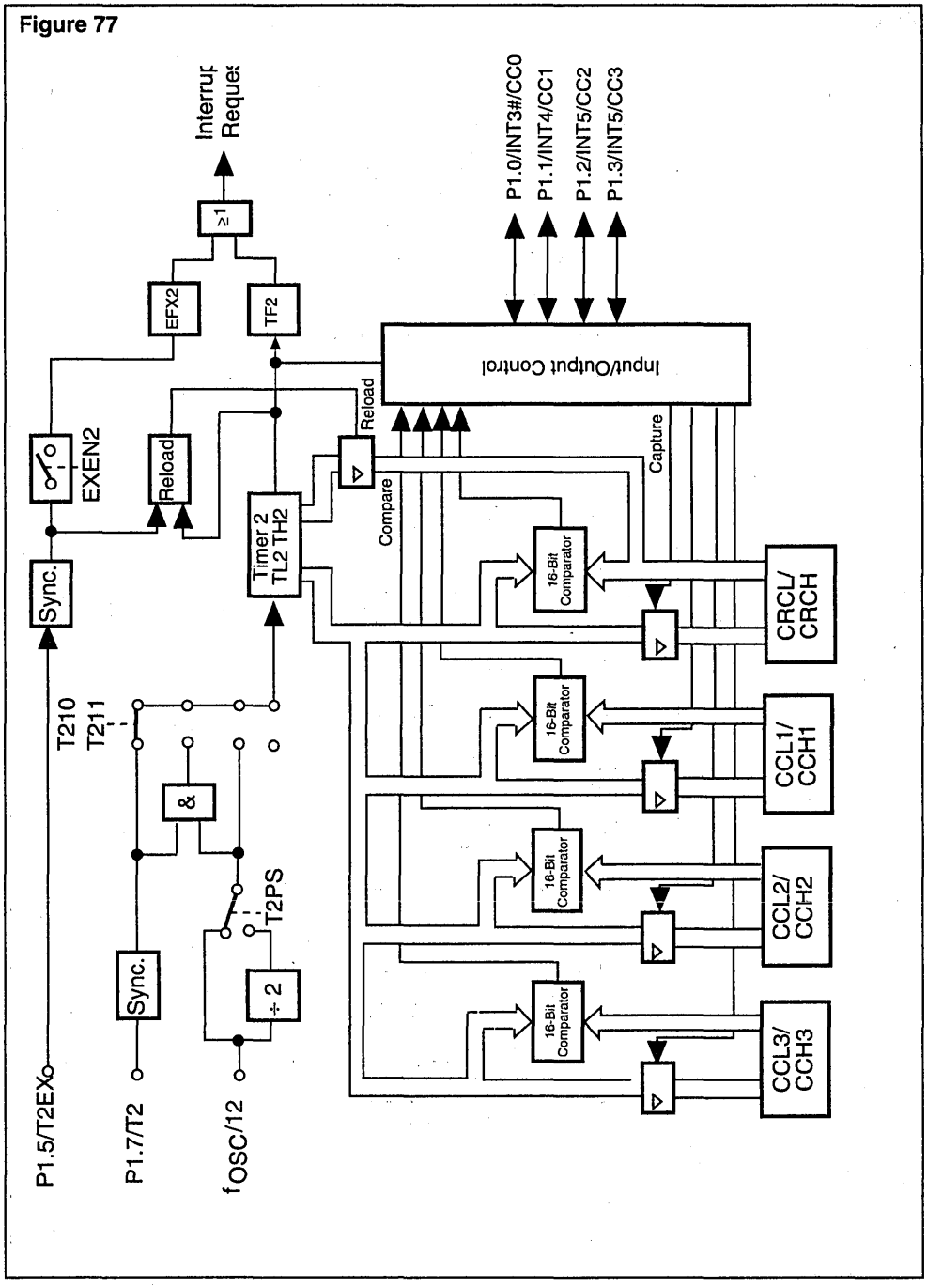

Interrupt Request
≥1
EFX2
TF2
P1.0/INT3#/CC0
P1.1/INT4/CC1
P1.2/INT5/CC2
P1.3/INT5/CC3
Input/Output Control
Reload
Capture
EXEN2
Reload
Compare
Timer 2
TL2 TH2
Sync.
16-Bit Comparator
CRCL/
CRCH
CCL1/
CCH1
CCL2/
CCH2
CCL3/
CCH3
T210
T211
&
T2PS
Sync.
÷ 2
P1.5/T2EX
P1.7/T2
fOSC/12

Figure 78

How to generate a PWM-signal with the SAB 80515
(e.g. PWM Period with 4 bit resolution):

Compare Mode 0

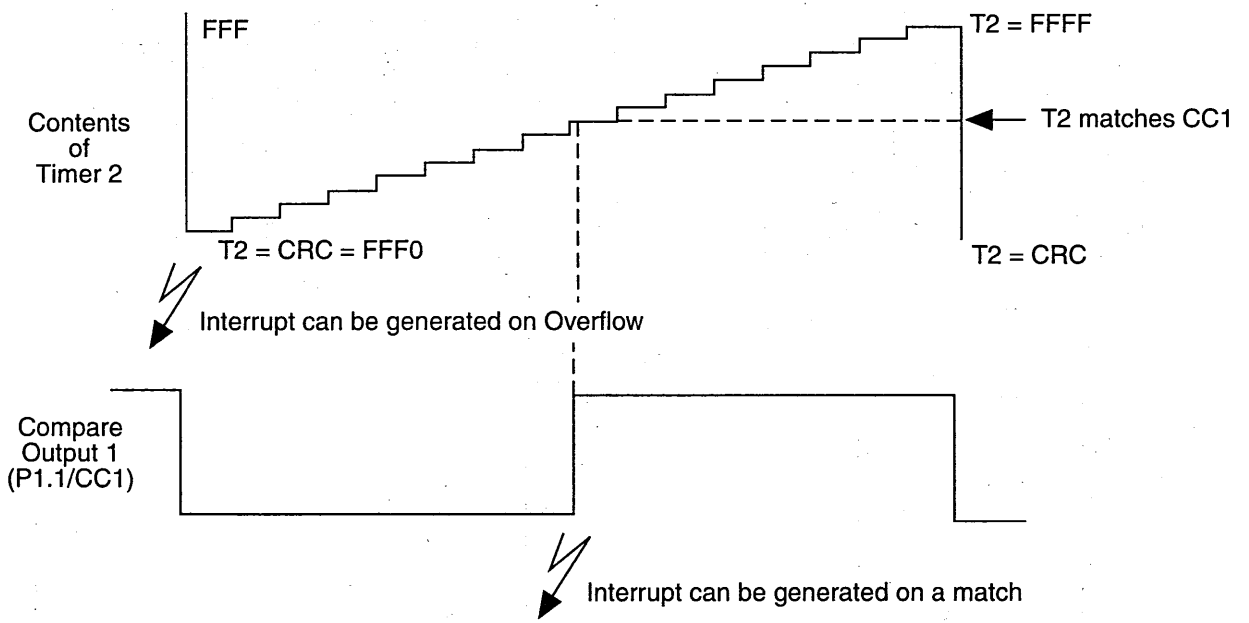


Figure 79

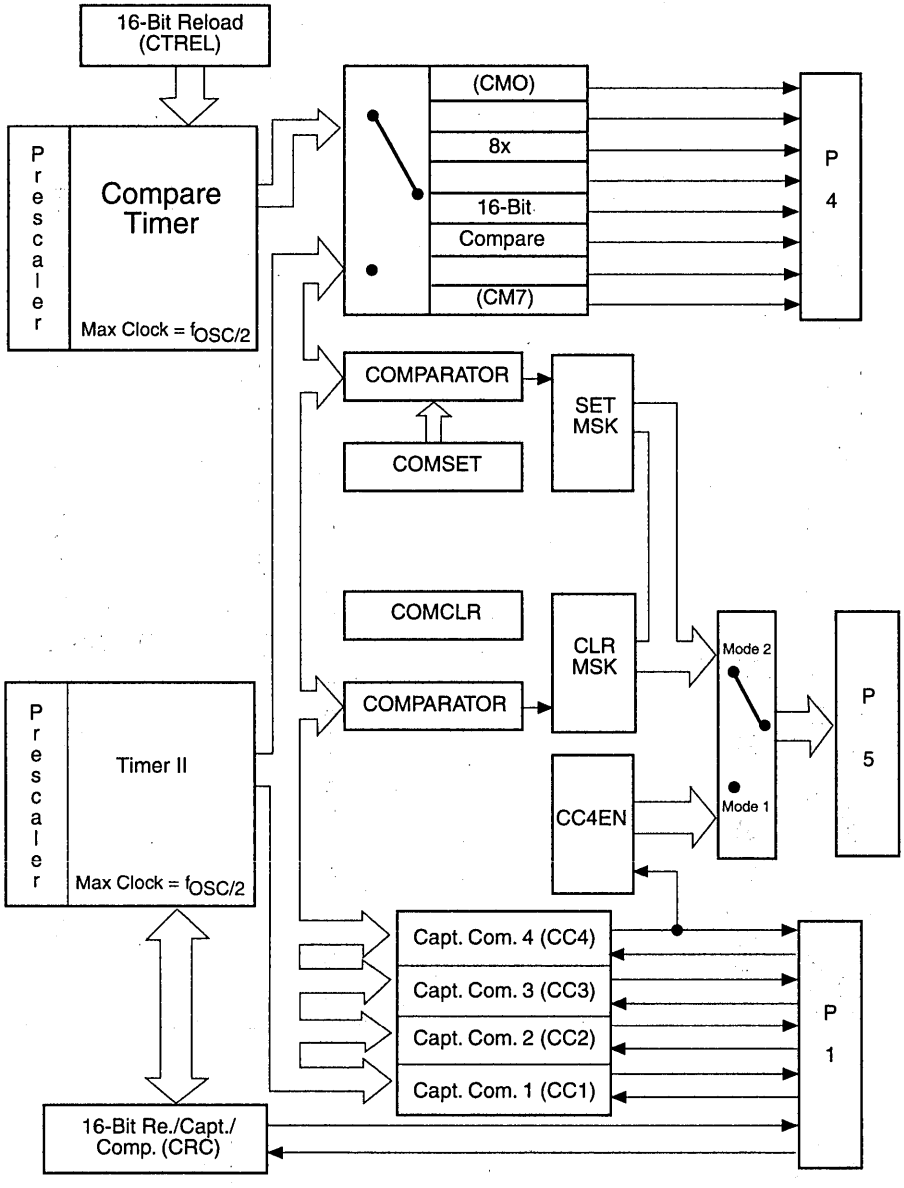
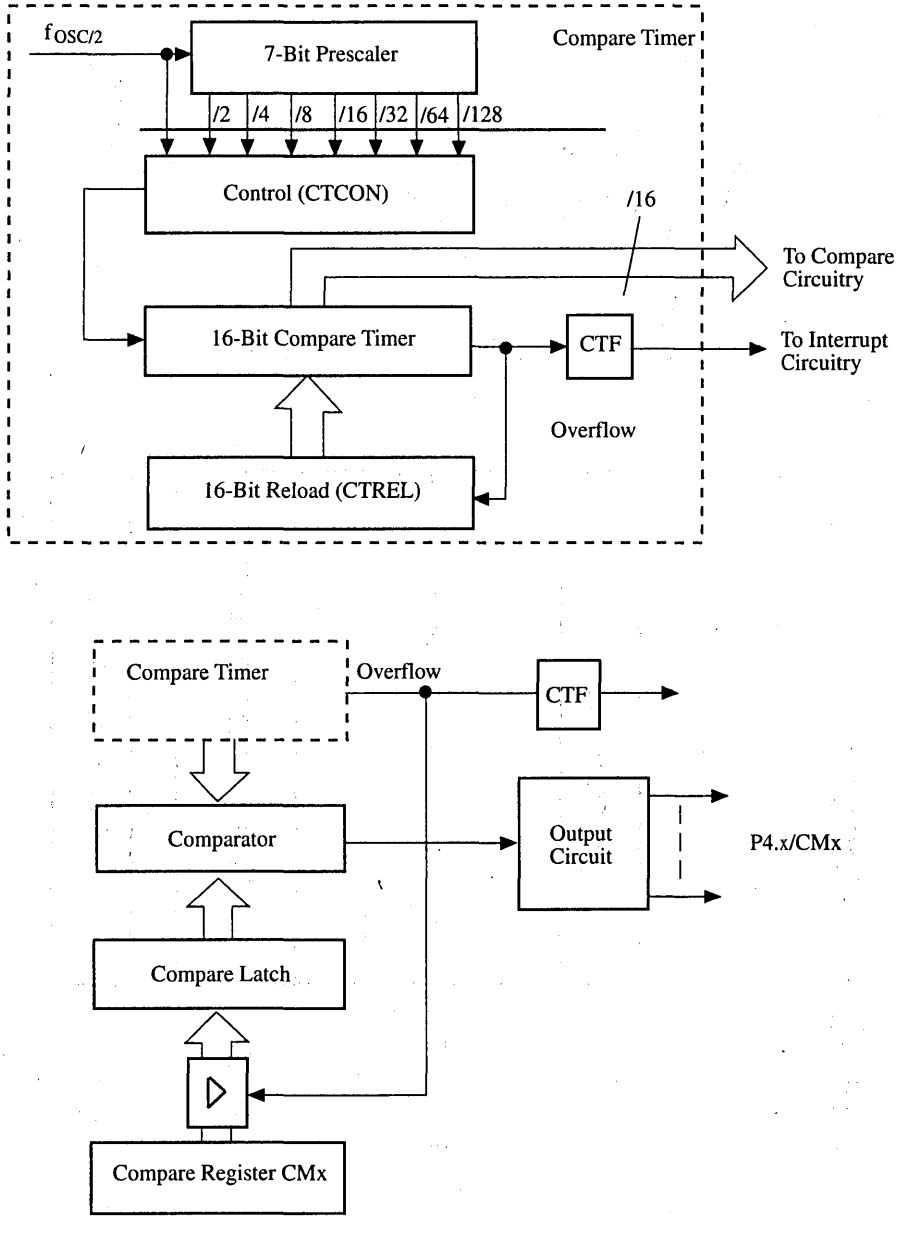

16-Bit Reload (CTREL)
Prescaler
Compare Timer
Max Clock = fOSC/2
(CMO)
8x
16-Bit Compare
(CM7)
P 4
COMPARATOR
COMSET
SET MSK
COMCLR
CLR MSK
COMPARATOR
Mode 2
Mode 1
P 5
CC4EN
Prescaler
Timer II
Max Clock = fOSC/2
Capt. Com. 4 (CC4)
Capt. Com. 3 (CC3)
Capt. Com. 2 (CC2)
Capt. Com. 1 (CC1)
P 1
16-Bit Re./Capt./ Comp. (CRC)

Figure 80


fOSC/2
7-Bit Prescaler
Compare Timer
/2
/4
/8
/16
/32
/64
/128
Control (CTCON)
/16
To Compare
Circuitry
16-Bit Compare Timer
CTF
To Interrupt
Circuitry
Overflow
16-Bit Reload (CTREL)
Compare Timer
Overflow
CTF
Comparator
Output
Circuit
P4.x/CMx
Compare Latch
Compare Register CMx

8

Figure 81

Multi-Layer Concept

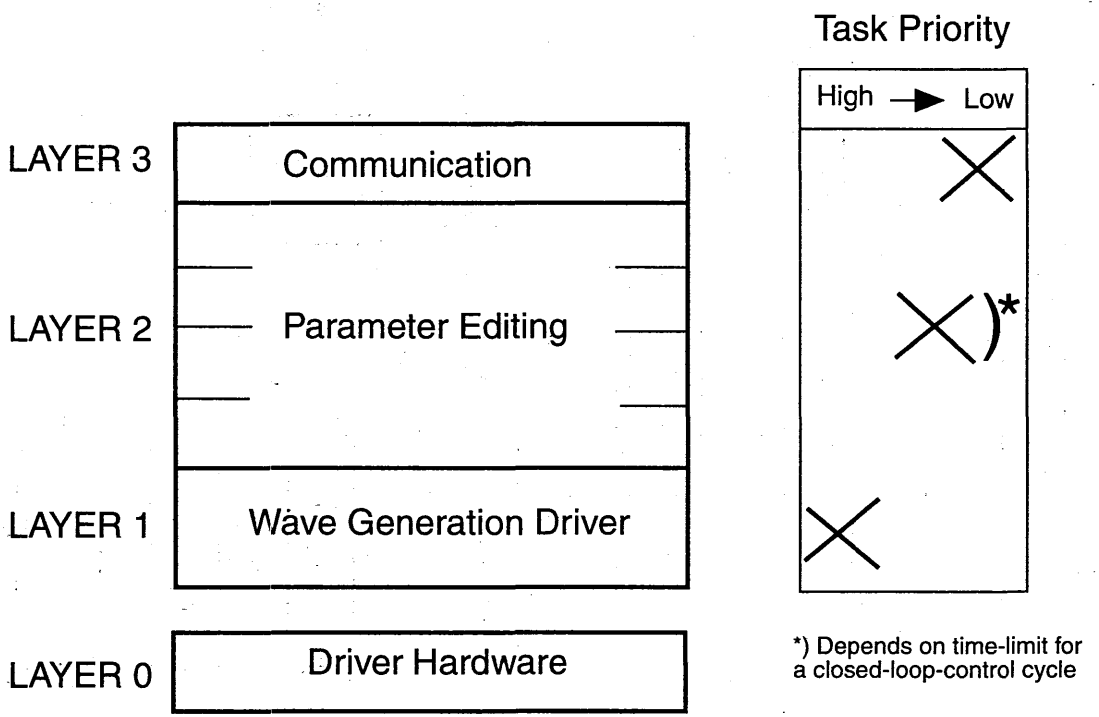
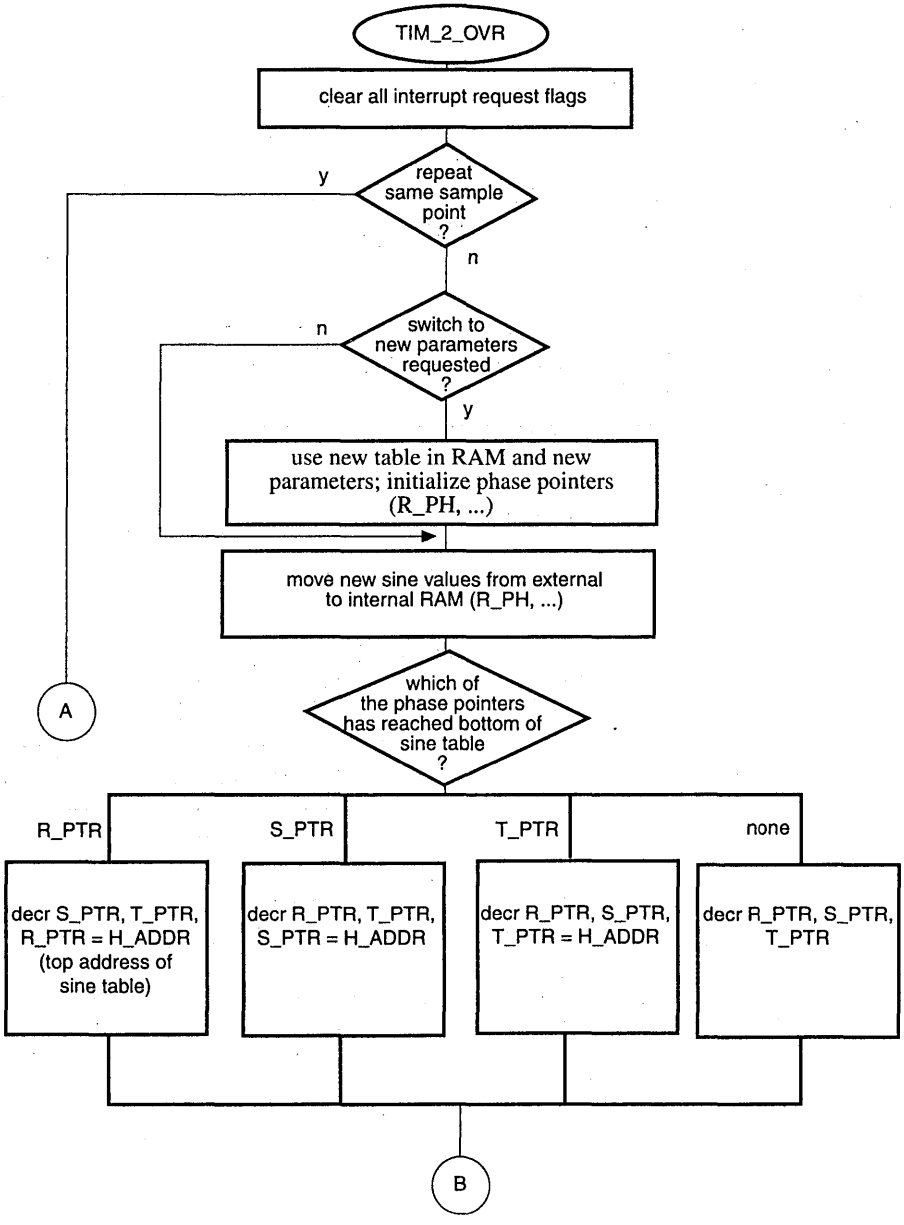


Figure 82


TIM_2_OVR
clear all interrupt request flags
repeat same sample point ?
y
n
switch to new parameters requested ?
n
y
use new table in RAM and new parameters; initialize phase pointers (R_PH, ...)
move new sine values from external to internal RAM (R_PH, ...)
which of the phase pointers has reached bottom of sine table ?
A
R_PTR
S_PTR
T_PTR
none
decr S_PTR, T_PTR, R_PTR = H_ADDR (top address of sine table)
decr R_PTR, T_PTR, S_PTR = H_ADDR
decr R_PTR, S_PTR, T_PTR = H_ADDR
decr R_PTR, S_PTR, T_PTR
B

8

Figure 83

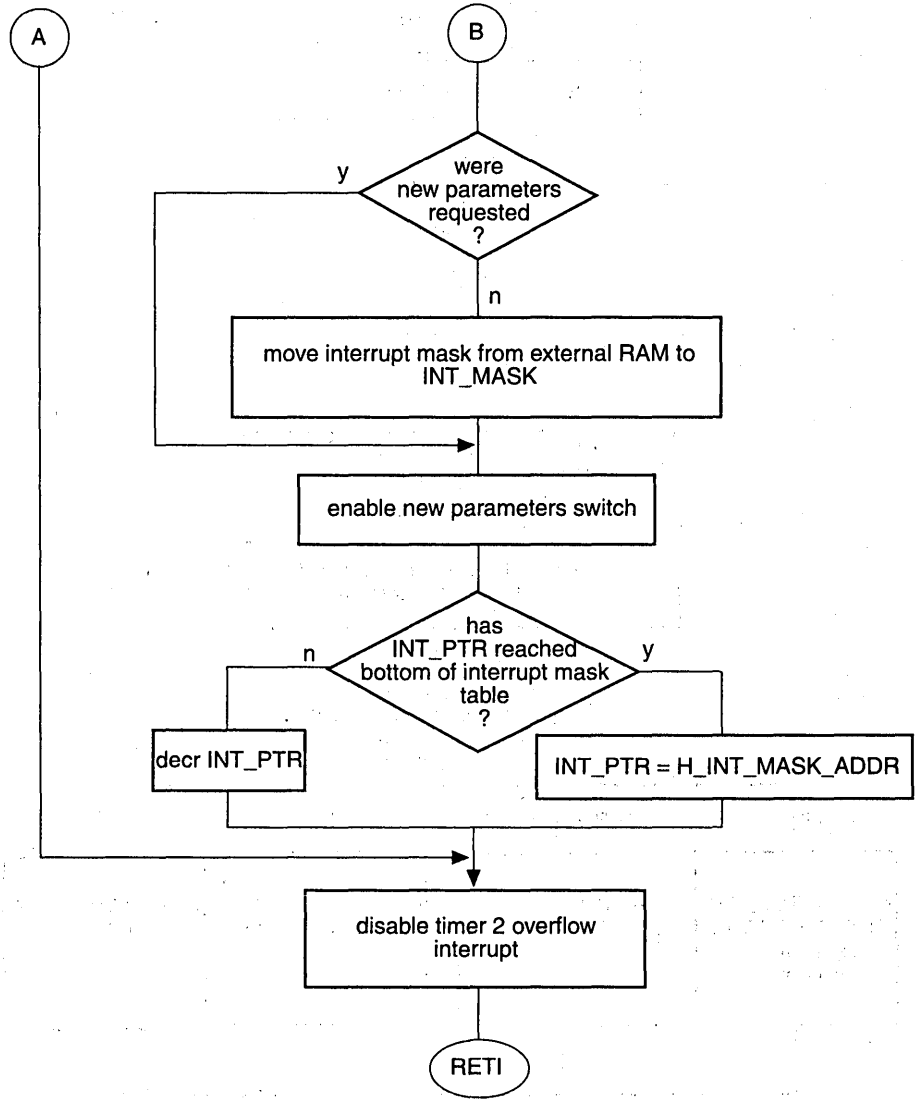

A
B
were new parameters requested ?
y
n
move interrupt mask from external RAM to INT_MASK
enable new parameters switch
has INT_PTR reached bottom of interrupt mask table ?
n
y
decr INT_PTR
INT_PTR = H_INT_MASK_ADDR
disable timer 2 overflow interrupt
RETI

Figure 84

Why "Interrupt Source Passing"?

Two proposals to think of a better way to generate a 3-Ph-PWM:

"Bad" Solution No. 1 — Easiest and least time consuming way for the controller: use timer-overflow interrupt to load PWM-values for all three phases.
Problem: Interrupt routine restricts minimal possible duty cycle.

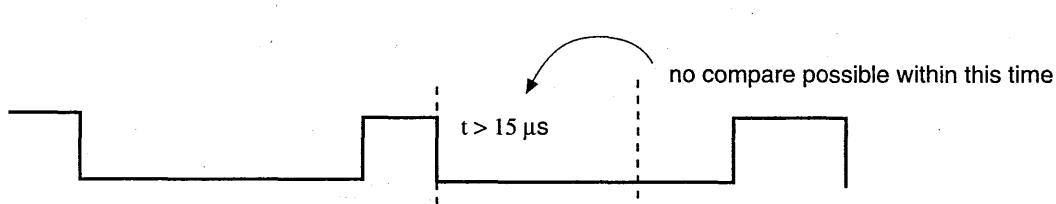
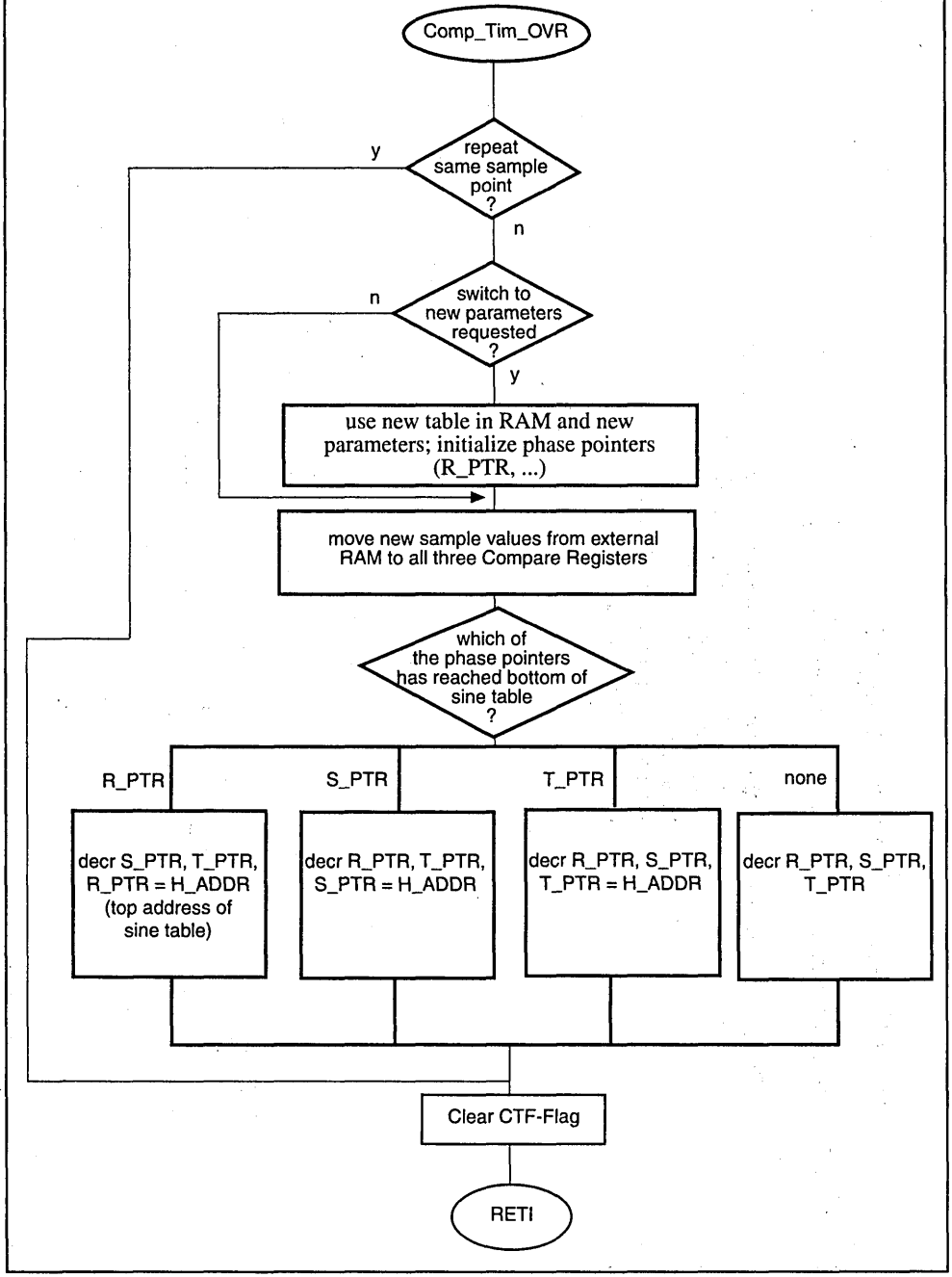


Figure 85


Comp_Tim_OVR
repeat same sample point ?
y
n
switch to new parameters requested ?
n
y
use new table in RAM and new parameters; initialize phase pointers (R_PTR, ...)
move new sample values from external RAM to all three Compare Registers
which of the phase pointers has reached bottom of sine table ?
R_PTR
S_PTR
T_PTR
none
decr S_PTR, T_PTR, R_PTR = H_ADDR (top address of sine table)
decr R_PTR, T_PTR, S_PTR = H_ADDR
decr R_PTR, S_PTR, T_PTR = H_ADDR
decr R_PTR, S_PTR, T_PTR
Clear CTF-Flag
RETI

Appendix A

Register Definitions for SAB 80515/80535

```
;         REG515.PDF for ASM51

;         REGISTER DECLARATIONS FOR SAB 80515

;********BYTE REGISTER********


PO        DATA 080H       ;PORT 0
SP        DATA 081H       ;STACK POINTER
DPL       DATA 082H       ;DATA POINTER-LOW BYTE
DPH       DATA 083H       ;DATA POINTER-HIGH BYTE
PCON      DATA 087H       ;POWER CONTROL
TCON      DATA 088H       ;TIMER CONTROL
TMOD      DATA 089H       ;TIMER MODE
TL0       DATA 08AH       ;TIMER 0-LOW BYTE
TL1       DATA 08BH       ;TIMER 1-LOW BYTE
TH0       DATA 08CH       ;TIMER 0-HIGH BYTE
TH1       DATA 08DH       ;TIMER 1-HIGH BYTE
P1        DATA 090H       ;PORT 1
SCON      DATA 098H       ;SERIAL PORT CONTROL
SBUF      DATA 099H       ;SERIAL PORT BUFFER
P2        DATA 0A0H       ;PORT 2
1EN0      DATA 0A8H       ;INTERRUPT ENABLE REGISTER 0
IP0       DATA 0A9H       ;INTERRUPT PRIORITY REGISTER 0
P3        DATA 0B0H       ;PORT3
1EN1      DATA 0B8H       ;INTERRUPT ENABLE REGISTER 1
1P1       DATA 0B9H       ;INTERRUPT PRIORITY REGISTER 1
IRCON     DATA 0C0H       ;INTERRUPT REQUEST CONTROL
CCEN      DATA 0C1H       ;COMPARE/CAPTURE ENABLE
CCL1      DATA 0C2H       ;COMPARE/CAPTURE REGISTER 1-LOW BYTE
CCH1      DATA 0C3H       ;COMPARE/CAPTURE REGISTER 1-HIGH BYTE
CCL2      DATA 0C4H       ;COMPARE/CAPTURE REGISTER 2-LOW BYTE
CCH2      DATA OC5H       ;COMPARE/CAPTURE REGISTER 2-HIGH BYTE
CCL3      DATA 0C6H       ;COMPARE/CAPTURE REGISTER 3-LOW BYTE
CCH3      DATA 0C7H       ;COMPARE/CAPTURE REGISTER 3-HIGH BYTE
T2CON     DATA 0C8H       ;TIMER 2 CONTROL
CRCL      DATA 0CAH       ;COMPARE/RELOAD/CAPTURE-LOW BYTE
CRCH      DATA 0CBH       ;COMPARE/RELOAD/CAPTURE-HIGH BYTE
TL2       DATA 0CCH       ;TIMER 2-LOW BYTE
TH2       DATA OCDH       ;TIMER 2-HIGH BYTE
PSW       DATA 0D0H       ;PROGRAM STATUS WORD
ADCON     DATA 0D8H             ;A/D CONVERTER CONTROL
ADDAT     DATA 0D9H       ;A/D CONVERTER DATA
DAPR      DATA 0DAH       ;D/A CONVERTER PROGRAM REGISTER
ACC       DATA 0E0H       ;ACCUMULATOR
P4        DATA 0E8H       ;PORT 4
B         DATA 0F0H       ;MULTIPLICATION REGISTER
P5        DATA 0F8H;PORT 5
```

8

Register Definitions for SAB 80515/80535 (continued)

```
********BIT REGISTERS********

IT0       BIT  088H       ;TCON.0-EXT.INTERRUPT 0 TYPE
IE0       BIT  089H       ;TCON.1-EXT.INTERRUPT 0 EDGE FLAG
IT1       BIT  08AH       ;TCON.2-EXT.INTERRUPT 1 TYPE
IE1       BIT  08BH       ;TCON.3-EXT.INTERRUPT 1 EDGE FLAG
TR0       BIT  08CH       ;TCON.4-TIMER 0 ON/OFF CONTROL
TF0       BIT  08DH       ;TCON.5-TIMER 0 OVERFLOW FLAG
TR1       BIT  08EH       ;TCON.6-TIMER 1 ON/OFF CONTROL
TF1       BIT  08FH       ;TCON.7-TIMER 1 OVERFLOW FLAG
INT3      BIT  090H       ;P1.0-EXTERNAL INTERRUPT 3/CAPTURE 0/COMPARE 0
INT4      BIT  091H       ;P1.1-EXTERNAL INTERRUPT 4/CAPTURE 1/COMPARE 1
INT5      BIT  092H       ;P1.2-EXTERNAL INTERRUPT 5/CAPTURE 2/COMPARE 2
INT6      BIT  093H       ;P1.3-EXTERNAL INTERRUPT 6/CAPTURE 3/COMPARE 3
INT2      BIT  094H       ;P1.4-EXTERNAL INTERRUPT 2
T2EX      BIT  095H;P1.5-TIMER 2 EXTERNAL RELOAD TRIGGER INPUT
CLKOUT    BIT  096H;P1.6-SYSTEM CLOCK OUTPUT
T2        BIT  097H;P1.7-TIMER 2 INPUT
RI        BIT  098H;SCON.0-RECEIVE INTERRUPT FLAG
TI        BIT  099H;SCON.1-TRANSMIT INTERRUPT FLAG
RB8       BIT  09AH;SCON.2-RECEIVE BIT 8
TB8       BIT  09BH;SCON.3-TRANSMIT BIT 8
REN       BIT  09CH;SCON.4 -RECEIVE ENABLE
SM2       BIT  09DH;SCON.5-SERIAL MODE CONTROL BIT 2
SM1       BIT  09EH;SCON.6-SERIAL MODE CONTROL BIT 1
SM0       BIT  09FH;SCON.7-SERIAL MODE CONTROL BIT0
EX0       BIT  0A8H;IEN0.0-EXTERNAL INTERRUPT 0 ENABLE
ET0       BIT  0A9H;IEN0.1-TIMER 0 INTERRUPT ENABLE
EX1       BIT  0AAH;IEN0.2-EXTERNAL INTERRUPT 1 ENABLE
ET1       BIT  0ABH;IEN0.3-TIMER 1 INTERRUPT ENABLE
ES        BIT  0ACH;IEN0.4-SERIAL PORT INTERRUPT ENABLE
ET2       BIT  0ADH;IEN0.5-TIMER 2 INTERRUPT ENABLE
WDT       BIT  0AEH;IEN0.6-WATCHDOG TIMER RESET
EAL       BIT  0AFH;IEN0.7-GLOBAL INTERRUPT ENABLE
RXD       BIT  0B0H;P3.0-SERIAL PORT RECEIVE INPUT
TXD       BIT  0B1H;P3.1-SERIAL PORT TRANSMIT OUTPUT
INT0      BIT  0B2H;P3.2-EXTERNAL INTERRUPT 0 INPUT
INT1      BIT  0B3H;P3.3-EXTERNAL INTERRUPT 1 INPUT
T0        BIT  0B4H;P3.4-TIMER 0 COUNT INPUT
T1        BIT  0B5H;P3.5-TIMER 1 COUNT INPUT
WR        BIT  0B6H;P3.6-WRITE CONTROL FOR EXT.MEMORY
RD        BIT  0B7H;P3.7-READ CONTROL FOR EXT.MEMORY
EADC      BIT  0B8H;IEN1.0-A/D CONVERTER INTERRUPT ENABLE
EX2       BIT  0B9H;IEN1.1-EXTERNAL INTERRUPT 2 ENABLE
EX3       BIT  0BAH;IEN1.2-EXTERNAL INTERRUPT 3 ENABLE
EX4       BIT  0BBH;IEN1.3-EXTERNAL INTERRUPT 4 ENABLE
EX5       BIT  0BCH;IEN1.4-EXTERNAL INTERRUPT 5 ENABLE
EX6       BIT  0BDH;IEN1.5-EXTERNAL INTERRUPT 6 ENABLE
SWDT      BIT  0BEH;IEN1.6-WATCHDOG TIMER START
EXEN2     BIT  0BFH;IEN1.7-TIMER 2 EXTERNAL RELOAD INTERRUPT ENABLE
IADC      BIT  0C0H;IRCON.0-A/D CONVERTER INTERRUPT REQUEST
IEX2      BIT  0C1H;IRCON.1-EXTERNAL INTERRUPT 2 EDGE FLAG
IEX3      BIT  0C2H;IRCON.2-EXTERNAL INTERRUPT 3 EDGE FLAG
IEX4      BIT  0C3H;IRCON.3-EXTERNAL INTERRUPT 4 EDGE FLAG
IEX5      BIT  0C4H;IRCON.4-EXTERNAL INTERRUPT 5 EDGE FLAG
IEX6      BIT  0C5H;IRCON.5-EXTERNAL INTERRUPT 6 EDGE FLAG
TF2       BIT  0C6H;IRCON.6-TIMER 2 OVERFLOW FLAG
EXF2      BIT  0C7H;IRCON.7-TIMER 2 EXTERNAL RELOAD FLAG
T2I0      BIT  0C8H;T2CON.0-TIMER 2 INPUT SELECT BIT 0
T2I1      BIT  0C9H;T2CON.1-TIMER 2 INPUT SELECT BIT 1
T2CM      BIT  0CAH;T2CON.2-COMPARE MODE
T2R0      BIT  0CBH;T2CON.3-TIMER 2 RELOAD MODE SELECT BIT 0
T2R1      BIT  0CCH;T2CON.4-TIMER 2 RELOAD MODE SELECT BIT 1
I2FR      BIT  0CDH;T2CON.5-EXTERNAL INTERRUPT 2 FALLING/RISING EDGE FLAG
I3FR      BIT  0CEH;T2CON.6-EXTERNAL INTERRUPT 3 FALLING/RISING EDGE FLAG
T2PS      BIT  0CFH;T2CON.7-PRESCALER SELECT BIT
P         BIT  0D0H;PSW.0-ACCUMULATOR PARITY FLAG
F1        BIT  0D1H;PSW.1-FLAG 1
OV        BIT  0D2H;PSW.2 -OVERFLOW FLAG
RS0       BIT  0D3H;PSW.3-REGISTER BANK SELECT 0
RS1       BIT  0D4H;PSW.4-REGISTER BANK SELECT 1
F0        BIT  0D5H;PSW.5-FLAG 0
AC        BIT  0D6H;PSW.6-AUXILIARY CARRY FLAG
CY        BIT  0D7H;PSW.7-CARRY FLAG
MX0       BIT  0D8H;ADCON.0-ANALOG INPUT CHANNEL SELECT BIT 0
MX1       BIT  0D9H;ADCON.1-ANALOG INPUT CHANNEL SELECT BIT 1
MX2       BIT  0DAH;ADCON.2-ANALOG INPUT CHANNEL SELECT BIT 2
ADM       BIT  0DBH;ADCON.3-A/D CONVERSION MODE
BSY       BIT  0DCH;ADCON.4-BUSY FLAG
CLK       BIT  0DEH;ADCON.6-SYSTEM CLOCK ENABLE
BD        BIT  0DFH;ADCON.7-BAUD RATE ENABLE
```

Register Definitions for SAB 80515/80535 (continued)

```
/*REG515.DCL for PLM51

/*Register declaration for SAB 80515 */

;********BYTE REGISTER********

DECLARE
P0        BYTE AT (080H)  REG,    /* PORT 0
PO        BYTE AT (081H)  REG,    /* STACK POINTER
DPL       BYTE AT (082H)  REG,    /* DATA POINTER-LOW BYTE
DPH       BYTE AT (083H)  REG,    /* DATA POINTER-HIGH BYTE
PCON      BYTE AT (087H)  REG,    /* POWER CONTROL
TCON      BYTE AT (088H)  REG,    /* TIMER CONTROL
TMOD      BYTE AT (089H)  REG,    /* TIMER MODE
TL0       BYTE AT (08AH)  REG,    /* TIMER 0-LOW BYTE
TL1       BYTE AT (08BH)  REG,    /* TIMER 1-LOW BYTE
TH0       BYTE AT (08CH)  REG,    /* TIMER 0-HIGH BYTE
TH1       BYTE AT (08DH)  REG,    /* TIMER 1-HIGH BYTE
P1        BYTE AT (090H)  REG,    /* PORT 1
SCON      BYTE AT (098H)  REG,    /* SERIAL PORT CONTROL
SBUF      BYTE AT (099H)  REG,    /* SERIAL PORT BUFFER
P2        BYTE AT (0A0H)  REG,    /* PORT 2
IEN0      BYTE AT (0A8H)  REG,    /* INTERRUPT ENABLE REGISTER 0
IP0       BYTE AT (0A9H)  REG,    /* INTERRUPT PRIORITY REGISTER 0
P3        BYTE AT (0B0H)  REG,    /* PORT 3
IEN1      BYTE AT (0B8H)  REG,    /* INTERRUPT ENABLE REGISTER 1
IP1       BYTE AT (0B9H)  REG,    /* INTERRUPT PRIORITY REGISTER 1
IRCON     BYTE AT (0C0H)  REG,    /* INTERRUPT REQUEST CONTROL
CCEN      BYTE AT (0C1H)  REG,    /* COMPARE/CAPTURE ENABLE
CCL1      BYTE AT (0C2H)  REG,    /* COMPARE/CAPTURE REGISTER 1-LOW BYTE
CCH1      BYTE AT (0C3H)  REG,    /* COMPARE/CAPTURE REGISTER 1-HIGH BYTE
CCL2      BYTE AT (0C4H)  REG,    /* COMPARE/CAPTURE REGISTER 2-LOW BYTE
CCH2      BYTE AT (0C5H)  REG,    /* COMPARE/CAPTURE REGISTER 2-HIGH BYTE
CCL3      BYTE AT (0C6H)  REG,    /* COMPARE/CAPTURE REGISTER 3-LOW BYTE
CCH3      BYTE AT (0C7H)  REG,    /* COMPARE/CAPTURE REGISTER 3-HIGH BYTE
T2CON     BYTE AT (0C8H)  REG,    /* TIMER 2 CONTROL
CRCL      BYTE AT (0CAH)  REG,    /* COMPARE/RELOAD/CAPTURE-LOW BYTE
CRCH      BYTE AT (0CBH)  REG,    /* COMPARE/RELOAD/CAPTURE-HIGH BYTE
TL2       BYTE AT (0CCH)  REG,    /* TIMER 2-LOW BYTE
TH2       BYTE AT (0CDH)  REG,    /* TIMER 2-HIGH BYTE
PSW       BYTE AT (0D0H)  REG,    /* PROGRAM STATUS WORD
ADCON     BYTE AT (0D8H)  REG,    /* A/D CONVERTER CONTROL
ADDAT     BYTE AT (0D9H)  REG,    /* A/D CONVERTER DATA
DAPR      BYTE AT (0DAH)  REG,    /* D/A CONVERTER PROGRAM REGISTER
ACC       BYTE AT (0E0H)  REG,    /* ACCUMULATOR
P4        BYTE AT (0E8H)  REG,    /* PORT 4
B         BYTE AT (0F0H)  REG,    /* MULTIPLICATION REGISTER
P5        BYTE AT (0F8H)  REG,    /* PORT 5
```

8

Register Definitions for SAB 80515/80535 (continued)

```
********BIT REGISTERS********
IT0       BIT  AT (088H)  REG,   /* TCON.0-EXT.INTERRUPT 0 TYPE
IE0       BIT  AT (089H)  REG,   /* TCON.1-EXT.INTERRUPT 0 EDGE FLAG
IT1       BIT  AT (08AH)  REG,   /* TCON.2-EXT.INTERRUPT 1 TYPE
IE1       BIT  AT (08BH)  REG,   /* TCON.3-EXT.INTERRUPT 1 EDGE FLAG
TR0       BIT  AT (08CH)  REG,   /* TCON.4-TIMER 0 ON/OFF CONTROL
TF0       BIT  AT (08DH)  REG,   /* TCON.5-TIMER 0 OVERFLOW FLAG
TR1       BIT  AT (08EH)  REG,   /* TCON.6-TIMER 1 ON/OFF CONTROL
TF1       BIT  AT (08FH)  REG,   /* TCON.7-TIMER 1 OVERFLOW FLAG
INT3      BIT  AT (090H)  REG,   /* P1.0-EXTERNAL INTERRUPT 3/CAPTURE 0/COMPARE 0
INT4      BIT  AT (091H)  REG,   /* P1.1-EXTERNAL INTERRUPT 4/CAPTURE 1/COMPARE 1
INT5      BIT  AT (092H)  REG,   /* P1.2-EXTERNAL INTERRUPT 5/CAPTURE 2/COMPARE 2
INT6      BIT  AT (093H)  REG,   /* P1.3-EXTERNAL INTERRUPT 6/CAPTURE 3/COMPARE 3
INT2      BIT  AT (094H)  REG,   /* P1.4-EXTERNAL INTERRUPT 2
T2EX      BIT  AT (095H)  REG,   /* P1.5-TIMER 2 EXTERNAL RELOAD TRIGGER INPUT
CLKOUT    BIT  AT (096H)  REG,   /* P1.6-SYSTEM CLOCK OUTPUT
T2        BIT  AT (097H)  REG,   /* P1.7-TIMER 2 INPUT
RI        BIT  AT (098H)  REG,   /* SCON.0-RECEIVE INTERRUPT FLAG
TI        BIT  AT (099H)  REG,   /* SCON.1-TRANSMIT INTERRUPT FLAG
RB8       BIT  AT (09AH)  REG,   /* SCON.2-RECEIVE BIT 8
TB8       BIT  AT (09BH)  REG,   /* SCON.3-TRANSMIT BIT 8
REN       BIT  AT (09CH)  REG,   /* SCON.4 -RECEIVE ENABLE
SM2       BIT  AT (09DH)  REG,   /* SCON.5-SERIAL MODE CONTROL BIT 2
SM1       BIT  AT (09EH)  REG,   /* SCON.6-SERIAL MODE CONTROL BIT 1
SM0       BIT  AT (09FH)  REG,   /* SCON.7-SERIAL MODE CONTROL BIT0
EX0       BIT  AT (0A8H)  REG,   /* IEN0.0-EXTERNAL INTERRUPT 0 ENABLE
ET0       BIT  AT (0A9H)  REG,   /* IEN0.1-TIMER 0 INTERRUPT ENABLE
EX1       BIT  AT (0AAH)  REG,   /* IEN0.2-EXTERNAL INTERRUPT 1 ENABLE
ET1       BIT  AT (0ABH)  REG,   /* IEN0.3-TIMER 1 INTERUPPT ENABLE
ES        BIT  AT (0ACH)  REG,   /* IEN0.4-SERIAL PORT INTERRUPT ENABLE
ET2       BIT  AT (0ADH)  REG,   /* IEN0.5-TIMER 2 INTERRUPT ENABLE
WDT       BIT  AT (0AEH)  REG,   /* IEN0.6-WATCHDOG TIMER RESET
EAL       BIT  AT (0AFH)  REG,   /* IEN0.7-GLOBAL INTERRUPT ENABLE
RXD       BIT  AT (0B0H)  REG,   /* P3.0-SERIAL PORT RECEIVE INPUT
TXD       BIT  AT (0B1H)  REG,   /* P3.1-SERIAL PORT TRANSMIT OUTPUT
INT0      BIT  AT (0B2H)  REG,   /* P3.2-EXTERNAL INTERRUPT 0 INPUT
INT0      BIT  AT (0B3H)  REG,   /* P3.3-EXTERNAL INTERRUPT 1 INPUT
T0        BIT  AT (0B4H)  REG,   /* P3.4-TIMER 0 COUNT INPUT
T1        BIT  AT (0B5H)  REG,   /* P3.5-TIMER 1 COUNT INPUT
WR        BIT  AT (0B6H)  REG,   /* P3.6-WRITE CONTROL FOR EXT.MEMORY
RD        BIT  AT (0B7H)  REG,   /* P3.7-READ CONTROL FOR EXT.MEMORY
EADC      BIT  AT (0B8H)  REG,   /* IEN1.0-A/D CONVERTER INTERRUPT ENABLE
EX2       BIT  AT (0B9H)  REG,   /* IEN1.1-EXTERNAL INTERRUPT 2 ENABLE
EX3       BIT  AT (0BAH)  REG,   /* IEN1.2-EXTERNAL INTERRUPT 3 ENABLE
EX4       BIT  AT (0BBH)  REG,   /* IEN1.3-EXTERNAL INTERRUPT 4 ENABLE
EX5       BIT  AT (0BCH)  REG,   /* IEN1.4-EXTERNAL INTERRUPT 5 ENABLE
EX6       BIT  AT (0BDH)  REG,   /* IEN1.5-EXTERNAL INTERRUPT 6 ENABLE
SWDT      BIT  AT (0BEH)  REG,   /* IEN1.6-WATCHDOG TIMER START
EXEN2     BIT  AT (0BFH)  REG,   /* IEN1.7-TIMER 2 EXTERNAL RELOAD INTERRUPT ENABLE
IADC      BIT  AT (0C0H)  REG,   /* IRCON.0-A/D CONVERTER INTERRUPT REQUEST
IEX2      BIT  AT (0C1H)  REG,   /* IRCON.1-EXTERNAL INTERRUPT 2 EDGE FLAG
IEX3      BIT  AT (0C2H)  REG,   /* IRCON.2-EXTERNAL INTERRUPT 3 EDGE FLAG
IEX4      BIT  AT (0C3H)  REG,   /* IRCON.3-EXTERNAL INTERRUPT 4 EDGE FLAG
IEX5      BIT  AT (0C4H)  REG,   /* IRCON.4-EXTERNAL INTERRUPT 5 EDGE FLAG
IEX6      BIT  AT (0C5H)  REG,   /* IRCON.5-EXTERNAL INTERRUPT 6 EDGE FLAG
TF2       BIT  AT (0C6H)  REG,   /* IRCON.6-TIMER 2 OVERFLOW FLAG
EXF2      BIT  AT (0C7H)  REG,   /* IRCON.7-TIMER 2 EXTERNAL RELOAD FLAG
T2I0      BIT  AT (0C8H)  REG,   /* T2CON.0-TIMER 2 INPUT SELECT BIT 0
T2I1      BIT  AT (0C9H)  REG,   /* T2CON.1-TIMER 2 INPUT SELECT BIT 1
T2CM      BIT  AT (0CAH)  REG,   /* T2CON.2-COMPARE MODE
T2R0      BIT  AT (0CBH)  REG,   /* T2CON.3-TIMER 2 RELOAD MODE SELECT BIT 0
T2R1      BIT  AT (0CCH)  REG,   /* T2CON.4-TIMER 2 RELOAD MODE SELECT BIT 1
I2FR      BIT  AT (0CDH)  REG,   /* T2CON.5-EXTERNAL INTERRUPT 2 FALLING/RISING EDGE FLAG
I3FR      BIT  AT (0CEH)  REG,   /* T2CON.6-EXTERNAL INTERRUPT 3 FALLING/RISING EDGE FLAG
T2PS      BIT  AT (0CFH)  REG,   /* T2CON.7-PRESCALER SELECT BIT
P         BIT  AT (0D0H)  REG,   /* PSW.0-ACCUMULATOR PARITY FLAG
F1        BIT  AT (0D1H)  REG,   /* PSW.1-FLAG 1
OV        BIT  AT (0D2H)  REG,   /* PSW.2-OVERFLOW FLAG
RS0       BIT  AT (0D3H)  REG,   /* PSW.3-REGISTER BANK SELECT 0
RS1       BIT  AT (0D4H)  REG,   /* PSW.4-REGISTER BANK SELECT 1
F0        BIT  AT (0D5H)  REG,   /* PSW.5-FLAG 0
AC        BIT  AT (0D6H)  REG,   /* PSW.6-AUXILIARY CARRY FLAG
CY        BIT  AT (0D7H)  REG,   /* PSW.7-CARRY FLAG
MX0       BIT  AT (0D8H)  REG,   /* ADCON.0-ANALOG INPUT CHANNEL SELECT BIT 0
MX1       BIT  AT (0D9H)  REG,   /* ADCON.1-ANALOG INPUT CHANNEL SELECT BIT 1
MX2       BIT  AT (0DAH)  REG,   /* ADCON.2-ANALOG INPUT CHANNEL SELECT BIT 2
ADM       BIT  AT (0DBH)  REG,   /* ADCON.3-A/D CONVERSION MODE
BSY       BIT  AT (0DCH)  REG,   /* ADCON.4-BUSY FLAG
CLK       BIT  AT (0DEH)  REG,   /* ADCON.6-SYSTEM CLOCK ENABLE
BD        BIT  AT (0DFH)  REG,   /* ADCON.7-BAUD RATE ENABLE
```

Software Support/Development Tools for the SAB-51 Family of Microcontrollers

SAB-51 Family

The hardware and software development support tools for the SAB-51 family of microcontrollers are available on IBM PC, XT, AT or a compatible. There are various solutions available from third party vendors, designed to be operated in an IBM PC compatible environment.

On account of the software compatibility of all the SAB-51 family of microcontrollers with the SAB 8051, the language and utility programs ASM51, PLM51, "C" and RL51 can be used without any restriction for any of the SAB-51 family of microcontrollers. However, the definitions of the new symbolic names as well as the register and bit addresses for the microcontrollers other than the SAB 8051 must be linked to the source program.

For example, in the case of the SAB 80515, a file called REG515.DCL is added to the PLM51 source program by an "Include" directive. Similarly, an "Include" directive is used to link a file called REG515.PDF to the ASM51 source program. In addition, the control instruction NOMOD51 (NOMO) must either be specified at the beginning of the source program, or in the assembler invocation, to avoid a multiple definition of the symbolic names. This control instruction causes default SAB 8051 declarations of register and bit symbols to be ignored during assembly. The file REG515.PDF includes all SAB 80515 register and bit definitions, including those that are declared with MOD51 for SAB 8051. The contents of the files REG515.PDF and REG515.DCL are listed in Appendix A.

Some third party vendors have already provided control directives such as MOD515, MOD512, etc. for their assemblers/compilers which eliminates the need to include files as mentioned above. In that case a mere mention of the control instructions MOD515, MOD512, etc. performs the job and the procedure of including a symbol declaration file is therefore transparent to the user.

Given below is a list of third party support products available for Siemens proprietary microcontrollers. These companies can also be contacted for development tools for the SAB 8031/8051/8032/8052 etc.

Third Party Hardware and Software Support Products

NOTE:

Note: For third party support contact the appropriate vendor for pricing, availability, and other information.

Metalink Corporation:
325 East Elliot Rd., Suite 23
Chandler, AZ 85225
Phone: (602) 926-0797

Hardware Support Product
Full In-Circuit-Emulator support for the SAB-51 family of microcontrollers for IBM PC compatible host system. Link to host system via RS232C interface.

Software Support Product

MetaWARE-51:
Cross assembly package for IBM PC compatible systems. Fully supports the SAB 80515/80535, SAB 80512/80532 and SAB 80C517/80C537.

Nohau Corporation:
51 East Campbell Ave., Suite 144
Campbell, CA 95008
Phone: (408) 866-1820

Hardware Support Product
Full In-Circuit-Emulator support for the SAB-51 family of microcontrollers for IBM PC compatible host system. Link to host system via RS232C serial interface or directly to the mother board.

Software Support Product

EMUL51-PC/C51:
"C" Cross compiler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

EMUL51-PC/AVO:
Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Signum Systems:
171 E. Thousand Oaks Blvd., Suite #202
Thousand Oaks, CA 91360
(805) 371-4608

Hardware Support Product
Full In-Circuit-Emulator support for the SAB-51 family of microcontrollers for IBM PC compatible host system. Link to host system via RS232C interface.

Software Support Product

ASM51:
Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

PL/M51:
Cross compiler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Allen Systems:
2151 Fairfax Road
Columbus, OH 43221
Phone: (614) 488-7122

Hardware Support Product
Prototype development board for SAB-51 family of microcontrollers comes with a monitor EPROM for program debugging.

Software Support Product

CA-51:
Cross assembler package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Rigel Corporation:
P.O. Box 90040
Gainesville, FL 32607
Phone: (904) 373-4629

Hardware Support Product
Prototype/Training development boards for SAB-51 family of microcontrollers.

TRI-L Data Systems, Inc.:
94-871 Farrington Hwy., 2nd Floor
Waipahu, HI 96797-3146
Phone: (800) 245-8745
(808) 671-5133

Hardware Support Product
Prototype development boards for SAB-51 family of microcontrollers.

Archemides Software:
2159 Union St.,
San Francisco, CA 94123
Phone: (415) 567-4010

Software Support Product

C-51:
"C" language cross-compiler package which includes a macro-assembler, linker and librarian for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

SimCASE:
Software simulator package for SAB-51 family of microcontrollers for IBM PC compatible systems.

BSO/Tasking:
P.O. Box 9164
Waltham, MA 02254-9164
Phone: (617) 320-9400
(800) 458-8276

Software Support Product

C-51 & PLM-51:
"C" language cross-compiler which includes a macro-assembler, linker and librarian for SAB-51 family of microcontrollers. Also available for PLM compiler. Both packages available on all popular hosts.

XRAY-51:
Source level Debugger for SAB-51 family of microcontrollers.

Franklin Software, Inc.:
888 Saratoga Ave., #2
San Jose, CA 95129
Phone: (408) 296-8051

Software Support Product

C-51:
"C" language C2055-compiler package which includes a macro-assembler, linker and librarian for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based) systems.

SIM-51:
Software simulator package for SAB-51 family of microcontrollers for IBM PC compatible systems.

Micro Computer Control:
P.O. Box 275,
Hopewell, NJ 08525
Phone: (609) 466-1751

Software Support Product

ASM51:
Cross-assembler package for SAB-51 family of microcontrollers. Available for IBM PC compatible systems (MSDOS based).

SIM-51:
SAB 8051 simulator.

C-51:
"C" language cross-compiler package for SAB-51 family of microcontrollers. Available for IBM PC compatible systems (MSDOS based).

2500AD Software Inc.:
109 Brookdale Ave.,
Box 480
Buena Vista, CO 81211
Phone: (719) 395-8683

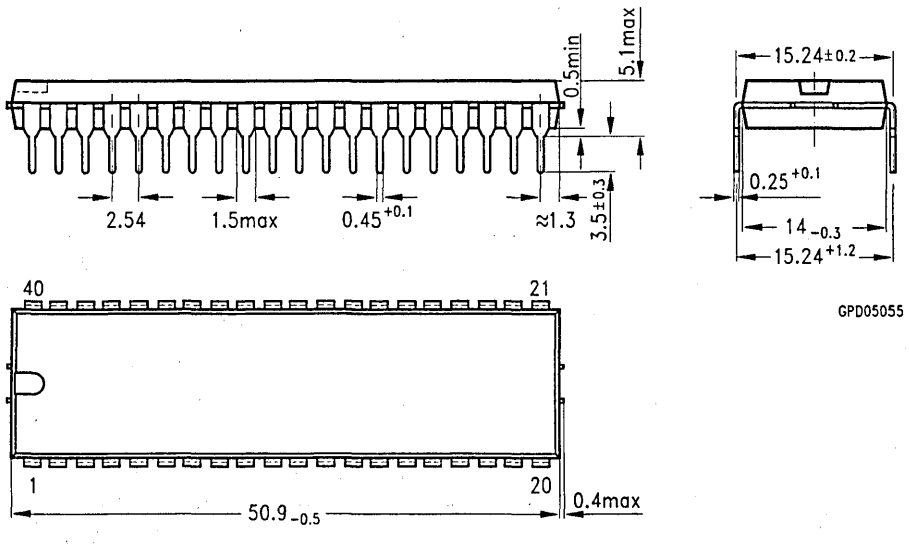
Software Support Product

A80515:
Cross-assembler package for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

S80515:
SAB 80515 simulator. Available for IBM PC compatible (MSDOS based), Microvax/VAX (running VMS/Ultrix) and SUN systems.

C80515:
"C" cross-compiler package for SAB-51 family of microcontrollers. Available for IBM PC compatible (MSDOS based), Microvax/ VAX (running VMS/Ultrix) and SUN systems.

Package Outlines

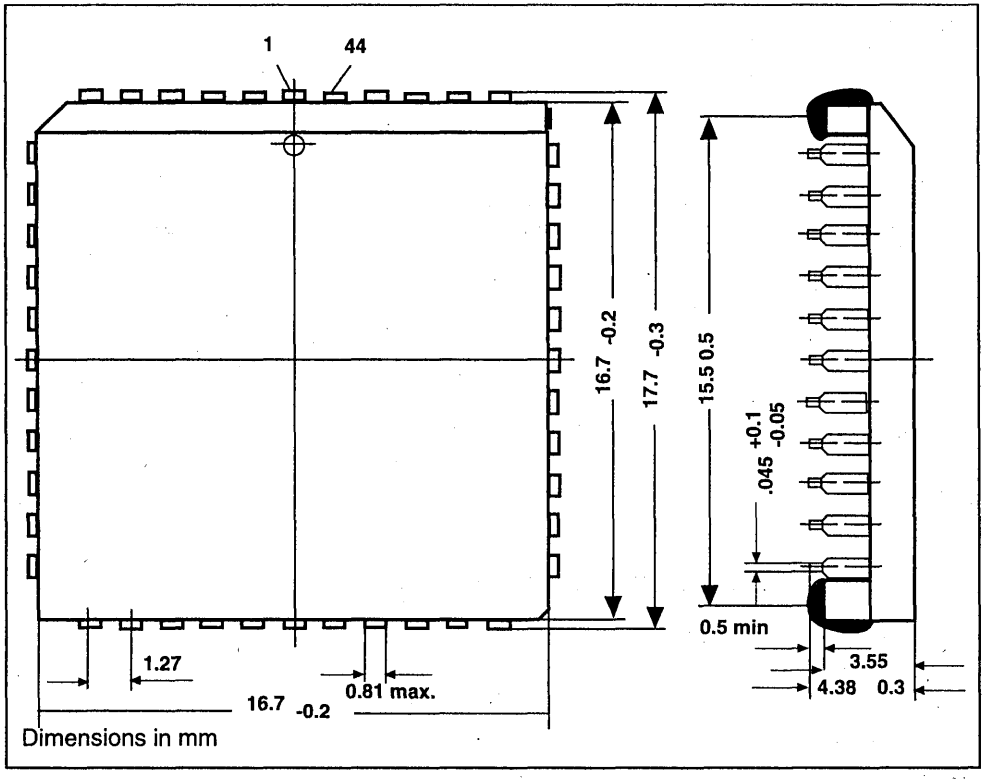


Approx. weight 5.9g Dimensions in mm

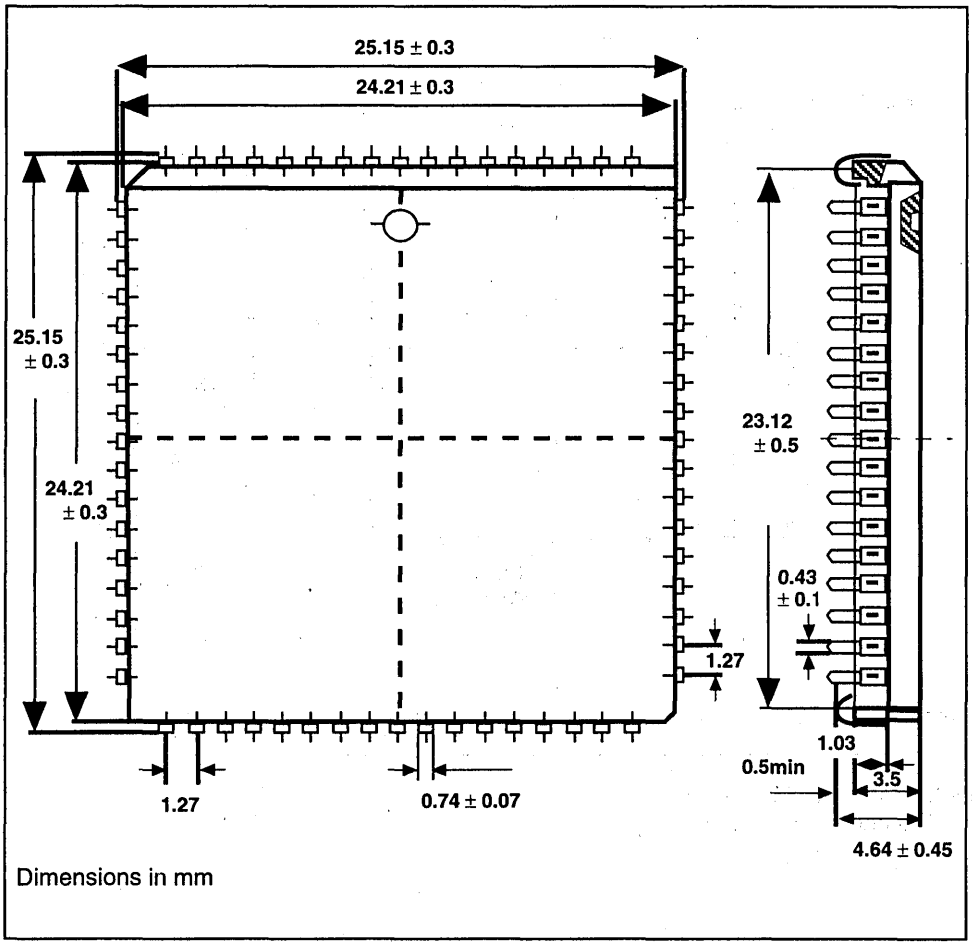
Plastic Package, P-DIP-40

(Dual-In-Line Package)
20 B 40 DIN 41870 T10

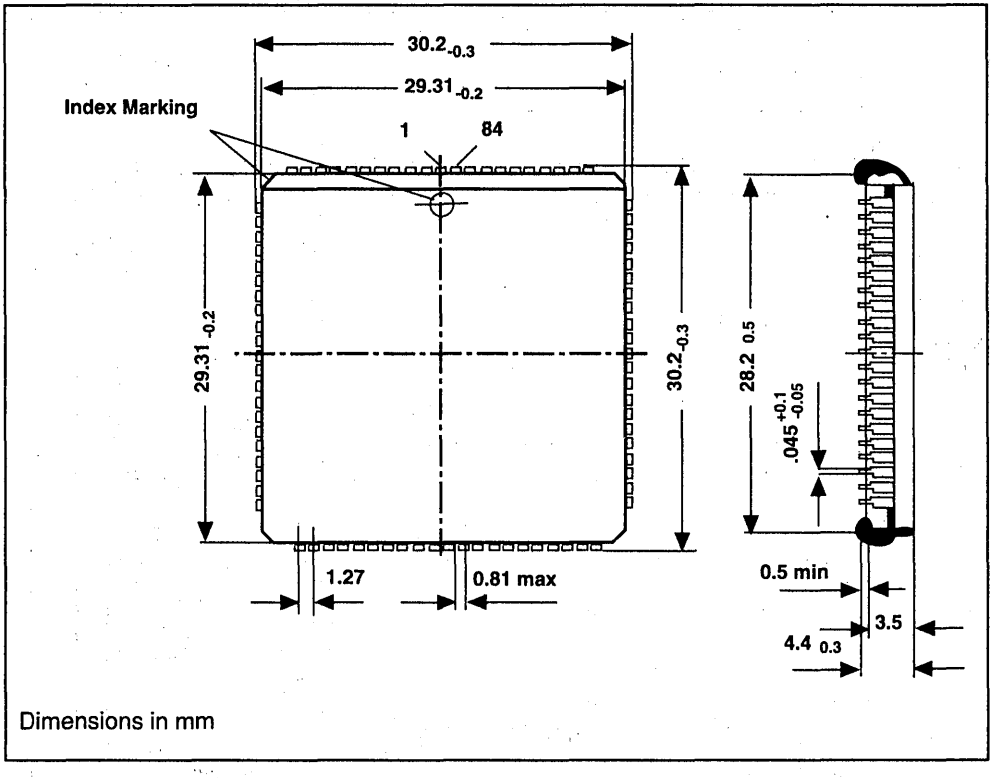
9



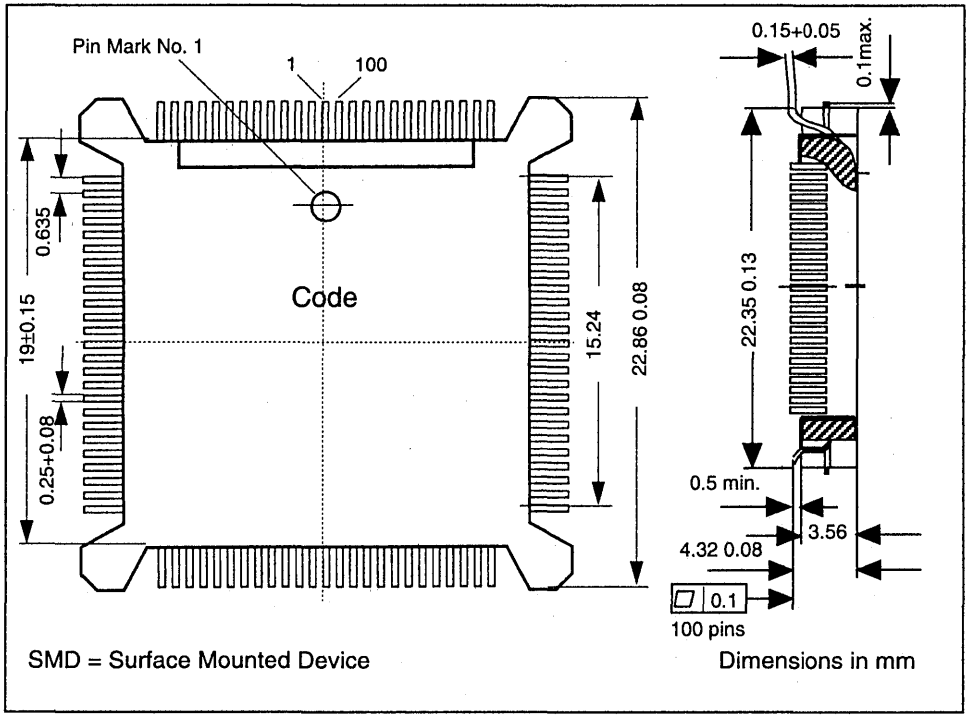
Plastic Package, P-LCC-44
(Plastic Leaded—Chip Carrier) **-SMD**



Plastic Package, PLCC-68 (SMD)
(plastic leaded chip carrier)



Plastic Package, PLCC-84 (SMD)
(plastic leaded chip carrier)



Plastic Package, P-QFP-100
(Plastic Quad-Flat-Pack) **- SMD**

9

Regional Sales Offices

Siemens Components, Inc.
238 Littleton Road
Westford, MA 01886
508/692-0550

Siemens Components, Inc.
Suite 202
307 Fellowship Road
Mt. Laurel, NJ 08054
609/273-6677

Siemens Components, Inc.
Suite 606
100 Wood Avenue South
Iselin, NJ 08830-2709
908/603-0600

Siemens Components, Inc.
Suite 206
6525 The Corners Parkway
Norcross, GA 30092
404/449-3981

Siemens Components, Inc.
Suite 435
38701 W. Seven Mile Road
Livonia, MI 48152
313/462-1195

Siemens Components, Inc.
Suite 565
1901 N. Roselle Road
Schaumburg, IL 60195
708/884-7009

Siemens Components, Inc.
Suite 320
625 The City Drive South
Orange, CA 92668
714/971-1274

Siemens Components, Inc.
10950 North Tantau Avenue
Cupertino, CA 95014
408/777-4968

Siemens Components, Inc.
Suite 690
16479 Dallas Parkway
Dallas, TX 75248
214/733-4511

Siemens Components, Inc.
Suite 103
1400 W. 122nd Avenue
Westminster, CO 80234
303/451-5513

Representatives

Alabama
Interep Associates, Inc.
Huntsville 205/881-1036
Daphne 205/621-1036

Arizona
Reptronix
Phoenix 602/230-2630

California—Northern
Siemens Components, Inc.
Santa Clara 408/777-4968

California—Southern
Dynarep, Inc.
Woodland Hills 818/346-2200
Tustin 714/258-8002

Varigon, Inc.
San Diego 619/576-0100

Colorado
Lange Sales, Inc.
Littleton 303/795-3600
Colorado Springs 719/632-8340

Florida
QXI, Inc.
Fort Lauderdale 305/978-0120
Melbourne 407/676-1491
Orlando 407/872-2321
St. Petersburg 813/894-4556

Georgia
Interep Associates, Inc.
Norcross 404/449-8680

Idaho
Lange Sales, Inc.
Boise 208/323-0713

Illinois
KMA Sales Company
Arlington Heights
708/398-5300

Indiana
Leslie M. DeVoe Company
Indianapolis 317/842-3245

Iowa
G.M. Brown & Associates
Cedar Rapids 319/393-9343

Kansas
G.M. Brown & Associates
Olathe 913/829-3086

Maryland
Micro-Comp, Inc.
Baltimore 301/644-5700

Massachusetts
Siemens Components, Inc.
Westford 508/692-0550

Paramount Technology
Waltham 617/890-8505

Michigan
Enco Marketing, Inc.
Bloomfield Hills
313/338-8600

Minnesota
Cahill-Schmitz-Cahill, Inc.
St. Paul 612/646-7217

Missouri
G.M. Brown & Associates
Florissant 314/839-3600

New Jersey—Northern
Emtec Sales
Englewood Cliffs 201/568-1151

New Jersey—Southern
See Pennsylvania—Eastern

New Mexico
Reptronix
Albuquerque 505/292-1718

New York
L-Mar Associates, Inc.
Cicero 315/699-0157
E. Aurora 716/655-4992
Fairport 716/425-9100
Syracuse 315/673-1325

North Carolina
ADI
Smithfield 919/934-8136
Matthews 704/847-4323

Ohio
Frank J. Campisano Company
Cincinnati 513/574-7111
Mentor 216/975-9300
Columbus 614/538-0469

Oklahoma
Quad States Sales & Marketing
Broken Arrow 918/357-1352

Oregon
Earl & Brown
Tigard 503/639-2100

Pennsylvania
Delta Tech
Hatboro 215/957-0600

Frank J. Campisano Company
Leechburg 412/845-2537

Puerto Rico
Klamco Electronics
Rio Piedras 809/748-5214

South Carolina
See North Carolina

Tennessee
Interep Associates, Inc.
Greenville 615/639-3491, 3492

Texas
Quad States Sales & Marketing
Dallas 214/669-8567
Austin 512/346-7002
Houston 713/467-7749

Utah
Lange Sales, Inc.
Salt Lake City 801/487-0843

Virginia
Micro Comp, Inc.
Lynchburg 804/239-2626

Washington
Earl & Brown
Redmond 206/885-5064

Washington, D.C.
See Maryland

Wisconsin
KMA Sales Company
Milwaukee 414/259-1771

Distributors

Alabama
Marshall • Huntsville 205/881-9235
Pioneer • Huntsville 205/837-9300

Arizona
Insight Electronics •
Tempe 602/829-1800
Marshall • Phoenix 602/496-029

California—Northern
Insight Electronics •
Sunnyvale 408/720-9222
Marshall • Milpitas 408/942-4600
Rancho Cordova 916/635-9700
Pioneer • San Jose 408/954-9100

California—Southern
Insight Electronics •
Irvine 714/727-3291
San Diego 619/587-1100
Westlake Village 818/707-2101
Marshall • Calabasas 818/878-7000
Chatsworth 818/407-4100
Irvine 714/859-5050
San Diego 619/627-4184
Pioneer • Irvine 714/753-5650
Agoura Hills 818/865-5800
San Diego 619/535-1035

Colorado
Insight Electronics •
Englewood 303/649-1800
Marshall • Thornton 303/451-8383

Connecticut
Marshall • Wallingford
203/265-3822
Pioneer • Shelton 203/929-5600

Florida
Marshall •
Altamonte Springs 407/767-8585
Fort Lauderdale 305/977-4880
St. Petersburg 813/573-1399
Pioneer •
Fort Lauderdale 305/428-8877
Orlando 407/834-9090

Georgia
Marshall • Norcross 404/923-5750
Pioneer • Duluth 404/623-1003

Illinois
Advent Electronics •
Rosemont 708/297-6200
Marshall • Schaumburg
708/490-0155
Pioneer • Chicago 708/495-9680

Indiana
Advent Electronics • Indianapolis
317/872-4910
Marshall • Indianapolis
317/297-0483
Pioneer • Indianapolis 317/573-0880

Iowa
Advent Electronics •
Cedar Rapids 319/363-0221

Kansas
Marshall • Lenexa 913/492-3121

Maryland
Marshall • Silver Springs
301/622-1118
Pioneer •
Gaithersburg 301/921-0660

Massachusetts
Marshall • Wilmington 508/658-0810
Pioneer • Lexington 617/861-9200

Michigan
Advent Electronics •
Farmington Hills 313/477-1650
Marshall • Livonia 313/525-5850
Pioneer • Livonia 313/416-2157

Minnesota
Marshall • Plymouth 612/559-2211
Pioneer • Twin Cities 612/944-3355

Missouri
Marshall • Bridgeton 314/291-4650

New Jersey
Marshall • Fairfield 201/882-0320
Mt. Laurel 609/234-9100
Pioneer • Fairfield 201/575-3510

New York
Marshall • Endicot 607/785-2345
Hauppauge 516/273-2695
Johnson City 607/798-1611
Rochester 716/235-7620
Pioneer • Binghamton 607/722-9300
Rochester 716/381-7070
Woodbury 516/921-8700

North Carolina
Marshall • Raleigh 919/878-9882
Pioneer • Raleigh 919/460-1530

Ohio
Marshall • Dayton 513/898-4480
Solon 216/248-1788
Pioneer • Cleveland 216/687-3600
Columbus 614/221-0043
Dayton 513/236-9900

Oklahoma
Pioneer • Tulsa 918/665-7840

Oregon
Insight Electronics •
Beaverton 503/644-3300
Marshall • Beaverton 503/644-5050

Pennsylvania
Pioneer • Horsham 215/674-4000
Pittsburgh 412/782-2300

Texas
Insight Electronics •
Austin 512/331-5887
Richardson 214/783-0800
Sugarland 713/448-0800
Marshall • Austin 512/837-1991
Carrollton 214/705-0600
Houston 713/467-1666
Pioneer • Austin 512/835-4000
Dallas 214/386-7300
Houston 713/495-4700

Utah
Marshall • Salt Lake City 801/973-2288

Washington
Insight Electronics •
Kirkland 206/820-8100
Marshall • Bothell 206/486-5747

Wisconsin
Marshall • Waukesha 414/797-8400
Pioneer • Brookfield 414/784-3480

Issued by Integrated Circuit Division
10950 North Tantau Avenue, Cupertino, CA 95014 (408) 777-4500

Siemens Components, Inc.

M14T146
Printed in U.S.A.
20K/0194/Delta