USER'S GUIDE FOR

NET/80™ SINGLE BOARD MICROCOMPUTER

AND

EXP/80™ EXPANSION INTERFACE BOARD

25% LARGER

MUSYS
Multi-user Microsystems

1451 Irvine Blvd., Suite 11, Tustin, CA 92680
(714) 730-5692 TWX: 910-595-1967
CABLE: MUSYSTSTN

662-7387

TABLE OF CONTENTS

APPENDICIES:

[TM] NET/80 and EXP/80 are trademarks of MUSYS CORP.

I. INTRODUCTION

The NET/80 and EXP/80 are the first in a series of products designed to give minicomputer throughput rates using microcomputer components. This is achieved by Technologies variously referred to an Multiprocessing, Distributed Processing, and Networking. While there is some feasability in a single user having the ability to spin off detached jobs into various processors, the primary demand for this type of architecture arises from requirements for multi-user computers. The two primary reasons leading to a decision in favor of a multi-user computer are a desire to minimize the need for costly peripheral equipment such as hard disks and letter quality printers, and a need to access shared data bases, usually in real time. It can truly be said with today's semiconductor technology that the Central Processing Unit (CPU) and it's memory are the cheapest part of a modern data processing system. In fact, hardware in general is becoming so inexpensive that the entire thrust of system implementation is changing. Because software costs are now the largest part of any business computer installation of any size, the requirements of the software now become the prime factors in the design of the hardware. Thus, one of the basic design criteria for new systems is to try to take advantage of an existing body of software while making available the benefits of a new hardware technology.

To this end, the NET/80 and EXP/80 were designed specifically to support the largest body of microcomputer software available today, the software which runs under the CP/M® operating system. This is made possible by

® and TM - CP/M, MP/M and CP/NET are trademarks of Digital Research.

two upwardly compatible software packages from the same vendor, the MP/M^{TM} resource sharing operating system and the CP/NET^{TM} networking package. The hardware on the NET/80 and EXP/80 is exactly what is required for these two packages, and therefore, is the optimum hardware selection for users desiring compatibility with older software, but needing greater throughput and flexibility.

Of course, the NET/80 and EXP/80 are not limited to these particular software packages. There are several software vendors with CP/M compatible operating systems specifically designed for this type of networking architecture. And CP/M is not the limiting factor either. The NET/80 is designed to be a flexible but low cost unit computer, and with the hardware flexibility offered by the local expansion bus and the software flexibility offered by 64K bytes of RAM and a Z80 processor, almost anything achievable by a computer is possible, if you only have enough of these computers working in parallel. So the last part of this discussion is a challenge to the user. Let us hear about your novel and unique applications for these hardware building blocks. If we feel there would be enough general interest, we will see to it that your idea is published in an appropriate place. Let your imagination run free, and the world will benefit.

II. NET/80 SINGLE BOARD MICROCOMPUTER

From a conceptual standpoint, the NET/80 is a very simple board. It has one switch and four connectors, and generally all you will do to use it is:

a. Set the DIP Switch.

b. Plug the NET/80 into an S-100 bus.

c. Mount the serial interface board and run the cable to the NET/80.

d. Wire RESET and Interrupt jumpers, if necessary for your application.

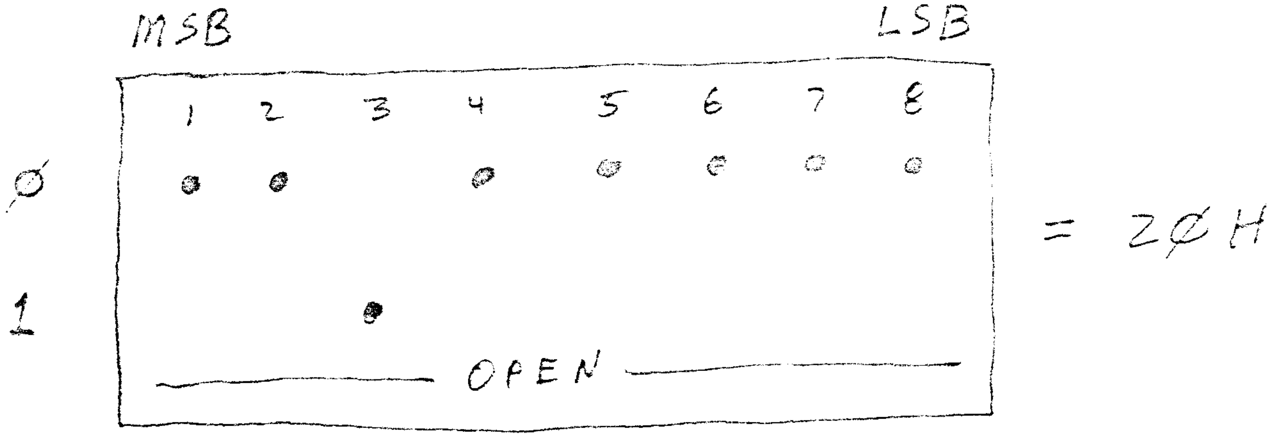
From the software viewpoint, it is just as simple. There is a serial port for a local CRT or other device, a parallel port for communicating with the master computer in the S-100 bus mainframe, and there is a bootstrap PROM to get you started. Beyond that, you have 64K bytes of RAM to fill with whatever 8080 or Z80 software strikes your fancy.

These instructions are written under the assumption you have purchased only the hardware from us and not the software. If this is not the case, you may skip this manual and go directly to the manual supplied with the software, which has the board set-up procedure for compatibility with that software.

A. DIP Switch Setting. The DIP switch has two functions on the NET/80. First, it selects between master and slave modes. Note that master mode is functional only in conjuncton with the EXP/80 Expansion Interface Board. Selecting master mode on a single board NET/80 will make the board non-functional.

The second function of the DIP switch is to select the even/odd pair of I/O port addresses as seen from the master. These two I/O ports will appear to the master as a simple parallel port interface through which the master can communicate with the slave. While any pair of addresses may be chosen, we recommend 20H-3FH as the range for slave processors.

Take the even numbered I/O port address and set it into the DIP switch as follows:



In the above diagram, the dot implies that the switch is pushed in at that point, and, of course, the other half of the same switch position will be sticking out. Because you will always set in the even address, switch position 8, the LSB will always be a zero, with the top pushed in and the bottom sticking out.

If you have connected an EXP/80 to this NET/80, and you desire this two board set to function as your S-100 permanent bus master, simply reverse the setting on position 8, pushing the bottom portion of switch 8 in, while the top part pops out. In master mode, the slave parallel port is disabled, and the setting of the other seven switch positions is ignored.

B. J1 - S-100 Bus INterface. J1 is the bottom edge connector on the NET/80, and will usually be plugged into a standard motherboard connector. The following pins are connected on J1:

1 = +8V (unregulated)
2 = +16V (unregulated)
29 = A5
30 = A4
31 = A3
35 = DO1 (-0.6 ma)*
36 = DO∅ (-0.6 ma)*
38 = DO4
39 - DO5
40 = DO6
41 = DI2
42 = DI3
43 = DI7
45 = SOUT
46 = SINP
50 = Ground
51 = +8V (unregulated)
52 = -16V (unregulated)
72 = RDY
77 = $\overline{\text{PWR}}$
78 = PDBIN
79 = A∅
80 = A1
81 = A2
82 = A6
83 = A7
88 = DO2
89 = DO3 (-0.6 ma)*
90 = DO7

* DENOTES the total BUS LOADING [illegible] This value is IN EXCESS OF THE -0.5ma ALLOWED BY IEEE STD 696 (PRELI

91	=	DI4
92	=	DI5
93	=	DI6
94	=	DI1
95	=	DIØ
100	=	Ground

All remaining pins of J1 are not connected. The usage of these pins on the NET/80 is as follows:

AØ-A7 = Decode the two I/O ports selected for the parallel port interface.

DIØ-DI7 = Send data from the slave to the master, under control of the master.

DOØ-DO7 = Receive data from the master.

SOUT and $\overline{\text{PWR}}$ = Output timing signals.

SINP and PDBIN = Input timing signals.

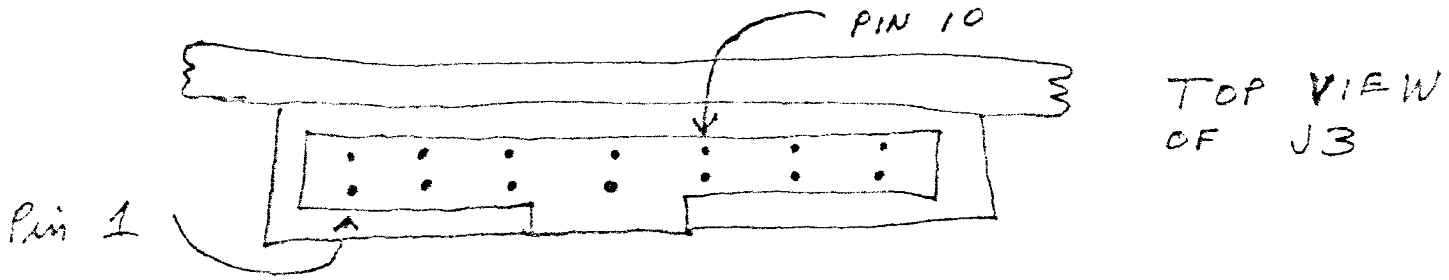
RDY = Pulled low by the slave when it's command port is written to by the master. This extends the slave response time by about four microseconds.

C. J2 - Front Panel Interface. J2 is a four pin connector which may optionally be used to connect the slave to a properly designed front panel. The pin usage on J2 is as follows:

1	=	Ground
2	=	$\overline{\text{RESET}}$ (Switch)
3	=	$\overline{\text{Compute}}$ (NET/80, TTL Output)
4	=	(No Connection)

Normally, an external RESET switch would be connected across pins 1 and 2. Pin 3 is high when the Z80 is halted or when an IORQ is active, such as when the slave is awaiting service from the master. Pin 3 is a TTL level and would require an external lamp driver circuit, unless a TTL compatible LED were used.

D. J3-Serial Port Interface. J3 connects the 2661 serial controller on the NET/80 to a personality board, which usually would be mounted on the rear panel of the S-100 mainframe. Generally, a null modem type of personality board is supplied with the NET/80, but certain other types of personality boards are available as options. If it is desired to run the NET/80 with no personality board installed, a short piece of wire wrap wire, or some other connection, must be used to jumper J3 pin 10 to J3 pin 1. This will disable the serial port RESET circuit, which otherwise would hold the NET/80 in a continuous RESET state. Appendix A contains a reprint of the data sheet for the Signetics 2661 serial interface controller. Appendix B defines the interface to the personality boards, as well as the standard boards available for the NET/80.

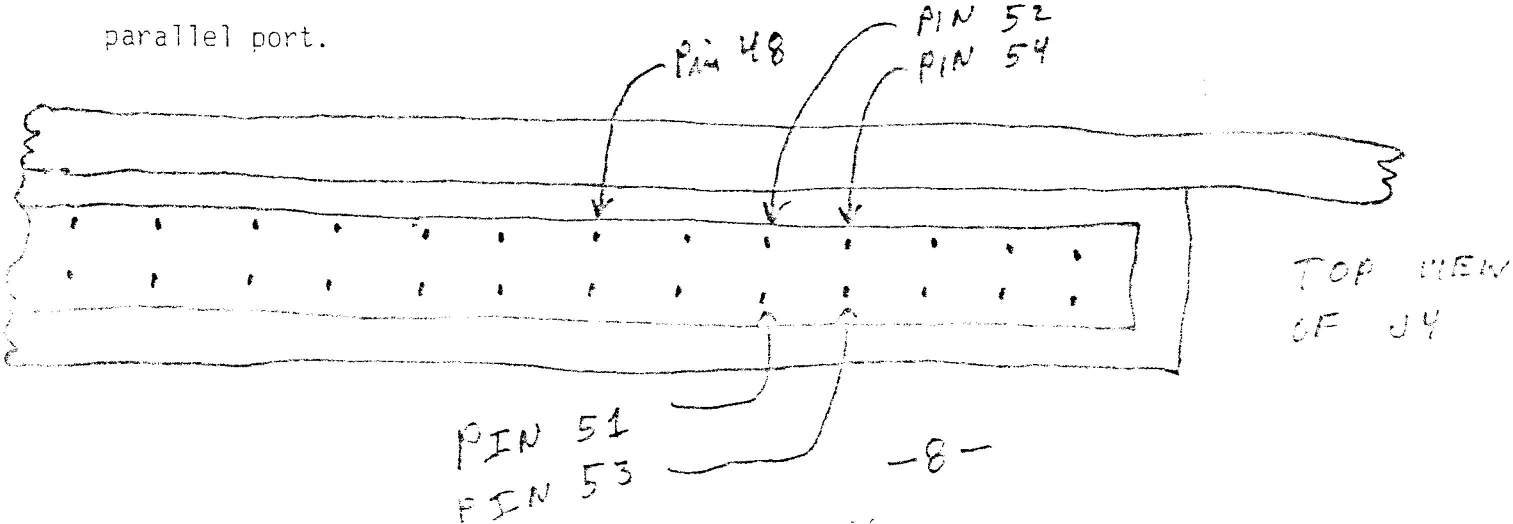


E. J4 - Expansion Bus Interface. One of the key design features of the NET/80 processor is the expansion bus, defined in detail in Appendix C. When an EXP/80 is connected to the NET/80, the 60 pin cable from the EXP/80 will plug into J4 on the NET/80, joining the two boards. See Section III of this manual for a complete description of the use of the EXP/80.

If no EXP/80 is used, J4 would usually remain open. However, the user is free to design his own expansion boards for the NET/80 by following the design specifications given in Appendix C.

One additional use of J4 is possible. The user may implement a single level interrupt on the NET/80 by using BERG style jumper plugs to connect J4-53 to J4-54 ($\overline{\text{SPINT}}$ to $\overline{\text{INT}}$) and J4-51 to J4-52 ($\overline{\text{SINTA}}$ to $\overline{\text{LCLINTA}}$). Because of the wait state generated when the master processor writes to the slaves command port, the slave has about five microseconds to recognize and respond to an interrupt request from the master. The software in the slave must not re-enable interrupts for at least six microseconds, and the software in the master should be programmed to re-issue the interrupt if the slave fails to respond in a predetermined amount of time.

The normal RESTART vector is to location 0018H (RST 3). The Z80 will vector to location 0038H (RST 7) if the jumper between J4-51 and J4-52 is removed. If, in place of any other jumpers, you wire from J4-48 to J4-54, the master processor interrupt becomes a non-maskable interrupt ($\overline{\text{NMI}}$) to the Z80, causing a vector to location 0066H. Note that no interrupt will be effective if the slave Z80 is locked in a wait state requesting service on the slave parallel port.



F. Programming the NET/80 (Slave Side). The slave environment is fairly simple. The memory consists of 64K bytes of dynamic RAM, the bottom 4K bytes of which can be overlaid by a shadow PROM or EPROM. The NET/80 comes with a 32 byte PROM (74S288 or 82S123) preprogrammed with the slave processor boot program. If the EXP/80 is connected, the 32 byte PROM may be removed and replaced with an EPROM containing up to 4K bytes of program and data. In any case, the PROM or EPROM will appear throughout the entire address range of 0000H-0FFFH, regardless of it's actual size. The remainder of the memory address space will always appear as RAM, and remember, the shadow PROM is controlled by a bit in the board control port, so the lower 4K will appear as either RAM or PROM, depending on the state of that bit.

The I/O ports on the NET/80 and EXP/80 take up two identical ranges of 16 port addresses, 00H-0FH and 80H-8FH. There is no difference between port 00H and port 80H, and so forth. This is caused by the fact that A7 is ignored during on board decoding of the port address. The complete list of I/O ports is as follows:

LOW RANGE	HIGH RANGE	FUNCTION
00H-03H	80H-83H	NET/80 2661
04H-07H	84H-87H	EXP/80 2661
08H-0BH	88H-8BH	EXP/80 Centronics Parallel Port
0CH-0DH	8CH-8DH	EXP/80 9519A Interrupt Controller
0EH-0FH	8EH-8FH	NET/80 S-100 Parallel Port

The port addresses for the EXP/80 are decoded whether the EXP/80 is present or not. Programming for the 2661 is covered in Appendix A and programming for the EXP/80 is covered in Section III. The remainder of this section will be concerned with programming the S-100 Parallel Port.

Throughout this discussion, use of addresses 8EH and 8FH are assumed. Addresses 0EH and 0FH may also be used, but this will be the last time that they are mentioned.

I/O port 8EH is slave data port. An input from port 8EH will place the slave into a wait state until the master supplies a byte of data on the other side of the interface. Conversely, an output to port 8EH will place the slave into a wait state until the master takes the slave data. If the slave is not in one of these data wait states when the master accesses the data port on the master side, then a data overrun condition is assumed, and the OVERRUN flip/flop is set, which disables any future slave wait states until the overrun condition is cleared. Thus, the most basic aspects of the protocol required of the software can be summarized as follows:

1. The slave initiates all transactions. However, the master is allowed to RESET the slave or generate a slave interrupt, if necessary.
2. The number of bytes to transfer must be known in advance, or imbedded in the data. If the slave must react to imbedded data (length, status bytes, etc.), then the master protocol must allow enough time for the slave to perform it's function before resuming the transfer.
3. The slave will generally use Z80 block I/O instructions (INIR and OTIR) to access port 8EH, allowing the transfer to proceed at a rate determined by the master.
4. If the above rules are followed, the master need not check the slave status during the data transfer. The Z80 instructions INI

and OTI may be used, with a NOP as the previous instruction and a non-zero jump back to the NOP as the subsequent instruction. The status of the slave should be checked immediately before and after the I/O operation.

Port 8FH contains certain bit sensitive command and status information. On input, the bits have the following meaning:

7-4 = Unused, read as ones.
3 = Request bit status (1=Set)
2 = Overrun status (1=Overrun)
1 = Modem Present (from serial personality board)
0 = PROM Enabled

The REQUEST flip/flop is general purpose in nature. It may be set or cleared by the slave, and may also be cleared by the master. It may have any function desired by the software, but will generally be used to discriminate between the types of data being transferred across the interface (request block or data). The overrun status may be set or cleared by the slave software, as well as being set by the hardware when a data overrun is detected. Thus, the OVERRUN flip/flop could also be used as part of a general purpose protocol. However, data may not be transferred when OVERRUN is set. Modem Present is a signal which was designed to indicate what type of personality board was installed on the serial interface (Direct or Modem), However, since the same signal determines whether or not a remote RESET over the serial cable is recognized, the actual signal condition may be more of a reflection of the state of that jumper selectable option than a reflection of the type of personality board in use. However, this bit may still be useful in some situations. The final bit reflects whether the lower 4K bytes of memory are RAM or PROM.

and may be marginally useful in certain situations.

The control bits which are set or cleared by an output to port 8FH are echos of the three software controllable bits in the status read.

7-4 = No effect
3 = New Request bit status
2 = New Overrun status
1 = No effect
Ø = New PROM Enable status

An output to port 8FH will change all three bits, and if not all are to be changed, the status may be read from port 8FH, the one or two bits which need to be altered can be changed, and the result would then be written back to port 8FH.

G. Programming the NET/80 (Master Side). The NET/80 appears as two consecutive I/O ports to the master processor. The even numbered port is the data port while the next higher odd numbered port is the command/status port. The master should never access the data port unless it is sure that the slave has entered it's wait state. If the slave is not in it's wait state, accessing the data port will cause an overrun condition, disabling the interface until it is cleared. The section above on slave programming contains some suggestions for master/slave protocols.

The odd numbered port contains bit sensitive command/status information. Reading the odd port gives the following data:

7-4 = Unused, indeterminate state.
3 = REQUEST is set.
2 = OVERRUN is set.
1 = Slave is reading, and waiting for the master to write.
0 = Slave is writing, and waiting for the master to read.

The request and overrun status bits reflect the state of the associated flip/flop on the slave. The read and write status bits can be used to manage the protocol and check for ready conditions and protocol errors.

Writing to the odd port will cause a four microsecond wait state to be generated via S-100 bus pin 72. This wait is necessary when performing RESET or INTERRUPT commands, but will be generated whenever the odd port is written to. The command bits are as follows:

7-4 = Unused
3 = Clear the REQUEST flip/flop
2 = Unused
1 = INTERRUPT the slave
0 = RESET the slave

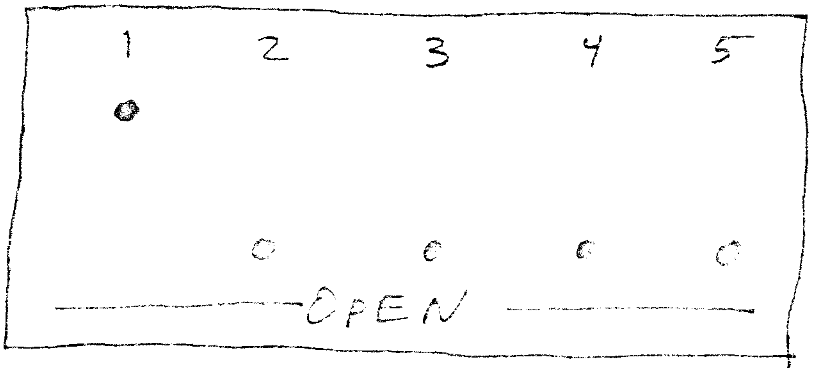
Note that the slave must be properly set up in order for the interrupt to function. If the 9519A Priority Interrupt Controller on the EXP/80 is being used, the interrupt will be latched and handled normally, according to the hardware/software configuration. In the single Board mode, the slave must recognize the interrupt before the wait state expires on the master. This is necessary because the $\overline{\text{INT}}$ input on the Z80 is level sensitive rather than edge sensitive. If edge sensitive interrupts are necessary, the $\overline{\text{NMI}}$ interrupt may be wired, as described earlier in the J4 section.

The master should be programmed with timeouts in any control loop so that if the slave does not react in the expected manner, the master will either retry the operation or declare the slave to be down.

III. EXP/80 EXPANSION INTERFACE BOARD

The EXP/80 is only usable by connecting it to a NET/80 through the 60 pin expansion interface, J4 on both boards. The EXP/80 gives the user several useful features, which may be used either in master or slave mode. The DIP switch on the NET/80 controls the selection of master or slave mode, enabling or disabling the S-100 permanent bus master logic as appropriate. The NET/80 also decodes the I/O port addresses used on the EXP/80, and these are described above in the section on the NET/80. The remainder of this section describes the features and set-up procedures unique to the EXP/80.

A. DIP Switch Setting. The DIP switch is part of the S-100 permanent master interface, and is therefore only functional in master mode. The switch is used if you have memory mapped peripherals which do not assert $\overline{\text{PHANTOM}}$ when they are addressed. When enabled, the switch will assert $\overline{\text{PHANTOM}}$ whenever the system address bus is in the selected 4K byte range. Closing switch position 1 (Top pushed in) will disable the comparison. If switch 1 is open (bottom pushed in) then switch positions 2 through 5 select the 4K byte block of NET/80 memory which is to be mapped onto the S-100 bus by asserting $\overline{\text{PHANTOM}}$. Switch 2 is the MSB and switch 5 is the LSB, the top pressed in is a zero while the bottom pressed in is a one.



In the above diagram, the dot implies that the switch is pushed in at that point, and, of course, the other half of the same switch position will be sticking out. The setting shown above disables the comparison because switch 1 is closed. If you were to reverse the setting for switch position 1 in the above diagram so that all five switches were open, then the NET/80 RAM from F000H to FFFFH would be disabled, and those addresses would access memory on the S-100 bus.

E. <u>J1 - S-100 Bus Interface</u>. J1 is the bottom edge connector on the EXP/80, and will usually be plugged into a standard motherboard connector. Every attempt has been made to have the EXP/80 be compatible with the proposed IEEE Standard 696, the Standard Specification for S-100 Bus Interface Devices. However, there are three things which mitigate against absolute compliance:

a. The design was based upon the proposal published in the July, 1979 issue of "COMPUTER" magazine. The proposal has been modified slightly since then and still has not been adopted in final form.

b. There is some desire on the part of users to be compatible with older devices which are not necessarily compatible with the standard.

c. As of this date, every 4MHz Z80 CPU board we have examined violates one or more of the timing specifications given in the 1979 proposal. This includes, without exception, those boards designed by the people who loudly trumpet their support of the standard.

Therefore, all we will say is that a good faith effort has been made to comply with the standard, but extreme measures were avoided where no practical benefit would be achieved. In the list of signals below, non-standard usages have been noted, along with pull-up and bus loading situations which technically violate the standard. These were implemented in non-standard ways due to considerations unique in the S-100 world, namely putting a whole bunch of identical processors on one motherboard, while still having a two board set which would function as a traditional CPU/Memory/I-O board set. (Note that the overloading on address lines A12 through A15 is not critical since the NET/80 boards are No Connect on these pins.)

Pin	Signal	Special Notes
1	+8 Volts	
2	+16 Volts	
3	XRDY	10K P.U. (Not specified O.C.)
4	VI0*	10K P.U.
5	VI1*	10K P.U.
6	VI2*	10K P.U.
7	VI3*	10K P.U.
8	VI4*	10K P.U.
9	VI5*	10K P.U.
10	VI6*	10K P.U.
11	VI7*	10K P.U.
12	NMI*	10K P.U.
13	PWRFAIL*	10K P.U.
14	DMA3*	10K P.U.
15	A18	No Connect
16	A16	No Connect
17	A17	No Connect
18	SDSB*	10K P.U.
19	CDSB*	10K P.U.

Therefore, all we will say is that a good faith effort has been made to comply with the standard, but extreme measures were avoided where no practical benefit would be achieved. In the list of signals below, non-standard usages have been noted, along with pull-up and bus loading situations which technically violate the standard. These were implemented in non-standard ways due to considerations unique in the S-100 world, namely putting a whole bunch of <u>identical</u> processors on one motherboard, while still having a two board set which would function as a traditional CPU/Memory/I-O board set. (Note that the overloading on address lines A12 through A15 is not critical since the NET/80 boards are No Connect on these pins.)

Pin	Signal	Special Notes
1	+8 Volts	
2	+16 Volts	
3	XRDY	10K P.U. (Not specified O.C.)
4	VI0*	10K P.U.
5	VI1*	10K P.U.
6	VI2*	10K P.U.
7	VI3*	10K P.U.
8	VI4*	10K P.U.
9	VI5*	10K P.U.
10	VI6*	10K P.U.
11	VI7*	10K P.U.
12	NMI*	10K P.U.
13	PWRFAIL*	10K P.U.
14	DMA3*	10K P.U.
15	A18	No Connect
16	A16	No Connect
17	A17	No Connect
18	SDSB*	10K P.U.
19	CDSB*	10K P.U.

Pin	Signal	Special Notes
20	GND	No Connect
21	NDEF	
22	ADSB*	10K P.U.
23	DODSB*	10K P.U.
24	Φ	
25	pSTVAL*	
26	pHLDA	10K P.U. (Not specified O.C.)
27	RFU	(Jumper option to use as PWAIT)
28	RFU	
29	A5	
30	A4	
31	A3	
32	A15	(-1.4ma, 80 μa loading)
33	A12	(-1.4ma, 80 μa loading)
34	A9	
35	DO1	
36	DO0	
37	A10	
38	DO4	
39	DO5	
40	DO6	
41	DI2	
42	DI3	
43	DI7	
44	sM1	
45	sOUT	
46	sINP	
47	sMEMR	
48	sHLTA	
49	CLOCK	
50	GND	
51	+8 Volts	
52	-16 Volts	
53	GND	No Connect

Pin	Signal	Special Notes
54	SLAVE CLR*	Generated, 10K P.U.
55	DMA0*	10K P.U.
56	DMA1*	10K P.U.
57	DMA2*	10K P.U.
58	sXTRQ*	
59	A19	No Connect
60	SIXTN*	10K P.U.
61	A20	No Connect
62	A21	No Connect
63	A22	No Connect
64	A23	No Connect
65	NDEF	
66	NDEF	
67	PHANTOM*	10K P.U.
68	MWRT	(Jumper Selectable)
69	RFU	
70	GND	No Connect
71	RFU	
72	RDY	10K P.U.
73	INT*	10K P.U.
74	HOLD*	10K P.U.
75	RESET*	Generated and Received, 10K P.U.
76	pSYNC	
77	pWR*	
78	pDBIN	
79	A0	
80	A1	
81	A2	
82	A6	
83	A7	
84	A8	
85	A13	(-1.4 ma, 80 μa loading)
86	A14	(-1.4 ma, 80 μa loading)
87	A11	

Pin	Signal	Special Notes
88	DO2	
89	DO3	
90	DO7	
91	DI4	
92	DI5	
93	DI6	
94	DI1	
95	DI0	
96	sINTA	
97	sWO*	
98	ERROR*	10K P.U.
99	POC*	Generated, 10K P.U. (Not specified O.C.)
100	GND	

As mentioned above, there are two jumper options associated with the S-100 interface. Both are two pin Berg type jumpers, located just above J1. The one on the left enables MWRT onto pin 68. The one on the right enables PWAIT onto pin 27, a non-standard usage required by some older boards.

C. J2 - Serial Port Interface. J2 connects the 2661 serial controller on the EXP/80 to the same type of personality boards used by the NET/80. As with the NET/80, a null modem type of personality board is the standard board supplied with the EXP/80. Again, Appendix A contains the data sheet for the 2661 while Appendix B defines the interface to the personality boards.

D. J3 - CENTRONICS Type Parallel Port. J3 is set up to drive any printer or other device which supports a CENTRONICS compatible parallel interface. Optionally, the port may be wired for 8 bits of data in and 8 bits of data out. Depending on the mode selected, the handshaking can take place automatically, allowing use of Z80 block output instructions when transferring data to the printer.

Even though I/O port addresses 08H-0BH are valid, the parallel port is generally programmed using I/O port addresses 88H-8BH. These for I/O ports have the functions defined in the following paragraphs.

First, the three pin BERG type jumper just below J3 should be set with the middle and left pins jumpered for a CENTRONICS type of interface and the middle and right pins jumpered for 8 bit, bidirectional data.

I/O port 88H is the command/status port for the parallel interface. On input, the printer status or, optionally, the 8 bits of data from the parallel input, are read. Note that if the data mode is being used, the parallel port interrupt must be wired in order to detect an acknowledge (strobe from the other parallel device). In the printer mode (jumper to the left) the status bits have the following meaning, provided that the particular printer supports the associated signal.

Normal data writes are done to I/O port address 8BH. This loads the output data, waits about 1.5 μs, generates a 1.5 μs $\overline{\text{STROBE}}$, and leaves the data active on the output data lines until the next write to port 8BH. This timing is compatible with the CENTRONICS parallel printer interface specification. I/O ports 89H and 8AH may be used when the above timing is not adequate. However, the $\overline{\text{STROBE}}$ will always be 1.5 μs in width (active low).

The parallel port logic could be wired to exchange data between computers over a short distance. Assuming a NET/80 and an EXP/80 in each computer, both wired for 8 bit data transfers with the jumper selection moved to the right, all that would be necessary would be to wire the output data and $\overline{\text{STROBE}}$ of each computer to the input data and acknowledge of the other computer. Normally in this configuration, the acknowledge would cause a parallel port interrupt in the receiving machine to start a data transfer. However, the protocol must be designed to resolve a dispute when both computers want to talk at the same time, or, have an ability to initiate bidirectional transfers so that an incoming acknowledge not only responds to the last data byte sent, but also flags the availability of a new data byte from the other computer.

The pin configuration of J3 is as follows:

PIN	NAME	I or O	PIN	NAME	I or O
1	$\overline{\text{STROBE}}$	O	8	GND	-
2	GND	-	9	DO3	O
3	DOØ	O	10	GND	-
4	GND	-	11	DO4	O
5	DO1	O	12	GND	-
6	GND	-	13	DO5	O
7	DO2	O	14	GND	-

Bit	Meaning	J3 Pin
7	ACKNOWLEDGE IS SET (On board)	None (29 in data mode)
6	DEMAND	36
5	POWER ON	35
4	PAPER MOVE or PRESENT	32
3	$\overline{\text{FAULT}}$	28
2	SELECT	25
1	PAPER EMPTY	23
0	BUSY	21

Bit 7 is a reflection of the state of the Acknowledge flip/flop, and can be used by the software to determine whether or not the printer has acknowledged for the last character transmitted. The remaining bits come directly from the printer, through a 1K ohm pullup.

Writing to I/O port 88H samples the state of bits 6 and 7 in the output byte. When bit 7 is set, the $\overline{\text{PRIME}}$ signal is sent to the printer (J3-26), When bit 6 is set, acknowledge wait states (described later) are enabled. The port must be written to again to clear any set bit. Upon RESET, acknowledge wait states are disabled and $\overline{\text{PRIME}}$ is asserted. Therefore, the initialization program after a RESET should write a byte of 00H to port 88H in order to remove the $\overline{\text{PRIME}}$ signal to the printer. (40H may be used if acknowledge wait states are desired.)

The remaining three I/O ports, 89H, 8AH, and 8BH are undefined for input operations. For output operations they are defined as follows:

89H = Loads data into the output register
8AH = Generates a data strobe
8BH = Both loads the data and generates a strobe

PIN	NAME	I or O
15	DO6	O
16	GND	-
17	DO7	O
18	GND	-
19	$\overline{\text{ACK}}$	I
20	GND	-
21	BUSY (0)	I
22	GND	-
23	PAPER MT(1)	I
24	GND	-
25	SELECT(2)	I
26	$\overline{\text{PRIME}}$	O
27	GND	-
28	$\overline{\text{FAULT}}$(3)	I
29	OPTIONAL DATA(7)	I
30	No Connect	-
31	GND	-
32	$\overline{\text{PAPER MOVE}}$ or $\overline{\text{PRESENT}}$(4)	I
33	GND	-
34	No Connect	-
35	POWER ON(5)	I
36	DEMAND(6)	I

NOTE: Numbers in parentheses indicate the bit for data input (0-7) when used for a general purpose interface.

E. J4 - Expansion Bus Interface. This connector is cabled to the NET/80 processor board which is associated with this EXP/80. Nothing prevents additional boards from being hooked into this same cable, except for design considerations involving functionality and signal loading. See Appendix C for a description of the expansion bus.

F. EPROM Option. The basic NET/80 is shipped with a preprogrammed 74S288 or 82S123 PROM in IC socket U51. This PROM is good for down loading a program from the master into the slave. In cases where the NET/80 is to perform as the master processor, or where diagnostics, monitor programs, and the like are desired, the PROM may be removed from U51 on the NET/80 and replaced with an EPROM on the EXP/80. There are 3 acceptable types of EPROM, with the installed type selected by the jumper area between J3 and J4. This jumper area consists of two six pin BERG type jumper areas, each with a

BERG type jumper plug. Each jumper plug has three legal positions, across the top two pins, across the middle two pins, or across the bottom two pins. The same selection should be made for both jumper areas. The top position is used for a 2708, the middle for a 2716 (or 2516). and the bottom for a 2732 (the lower half of a 2532 could be used by selecting the 2716 setting). Be sure that you have removed U51 on the NET/80 before installing an EPROM, and vice versa, be sure you install U51 if you remove the EPROM.

G. Interrupts. The selectable interrupt options consist of eight inputs into the 9519A Priority Interrupt Controller, plus the Z80 non-maskable interrupt line. There are 19 possible interrupt sources, and nothing prevents the user from using wire wrap wire to achieve any desired interrupt configuration. However, with a little planning, BERG type jumper plugs should be used for most or all interrupt selections.

The interrupt jumper area consists of 30 pins in a block above pin 1 on the EXP/80. The layout is as follows:

	$\overline{PWRFAIL}$	$\overline{INT}$	$\overline{PARINT}$	$\overline{T2RDY}$	$\overline{T1RDY}$	RTC	$\overline{R2RDY}$	$\overline{R1RDY}$	$\overline{SPINT}$
$\overline{ERROR}$	$\overline{ENMI}$	$\overline{IRQ7}$	$\overline{IRQ6}$	$\overline{IRQ5}$	$\overline{IRQ4}$	$\overline{IRQ3}$	$\overline{IRQ2}$	$\overline{IRQ1}$	$\overline{IRQØ}$
	$\overline{NMI}$	$\overline{VI7}$	$\overline{VI6}$	$\overline{VI5}$	$\overline{VI4}$	$\overline{VI3}$	$\overline{VI2}$	$\overline{VI1}$	$\overline{VIØ}$

The signals $\overline{\text{VI0-VI7}}$, $\overline{\text{NMI}}$, $\overline{\text{ERROR}}$, $\overline{\text{PWRFAIL}}$, and $\overline{\text{INT}}$ are buffered versions of the equivalent S-100 bus signal, and are only available to the master processor. $\overline{\text{PARINT}}$ comes from the acknowledge flip/flop in the parallel port logic. $\overline{\text{R1RDY}}$, $\overline{\text{R2RDY}}$, $\overline{\text{T1RDY}}$, and $\overline{\text{T2RDY}}$ are the receiver and transmitter ready signals from the 2661 on the NET/80 ($\overline{\text{R1RDY}}$ and $\overline{\text{T1RDY}}$) and the 2661 on the EXP/80 ($\overline{\text{R2RDY}}$ and $\overline{\text{T2RDY}}$). The Real Time Clock (RTC) is derived from the band rate oscillator, and produces a 60 Hz interrupt rate. The $\overline{\text{SPINT}}$ signal is the interrupt from the S-100 port from the master processor.

The interrupt destination consists of the eight inputs into the 9519A, $\overline{\text{IRQ0}}$-$\overline{\text{IRQ7}}$, plus the $\overline{\text{NMI}}$ line to the Z80.

```
               ;       NET/80 BOOTSTRAP PROM
               ;
               ;       (C) 1981, MUSYS CORP.
               ;
008F =         CTLPORT EQU     08FH            ; NET/80 CONTROL/STATUS PORT
0008 =         REQUEST EQU     008H            ; REQUEST BIT
0001 =         PROMON  EQU     001H            ; PROM ON BIT
0001 =         BLKTYP  EQU     001H            ; TYPE OF REQUEST BLOCK
004C =         BLKCOD  EQU     'L'             ; SECONDARY BOOT REQUEST
0000 =         SYSCOD  EQU     000H            ; SYSTEM CODE - 00H = STD. CP/M
1000 =         BOOT2   EQU     01000H          ; SECONDARY BOOT LOCATION
0080 =         BOOT2L  EQU     128             ; LENGTH OF SECONDARY BOOT PROGRAM
               ;
0000 018F0B    RESET:  LXI     B,BLOCKL*256+CTLPORT    ; B = DATA LEN., C = I/O PORT
0003 211500            LXI     H,BLOCK                 ; HL POINTS TO REQUEST BLOCK
               ;
0006 EDA3              DB      0EDH,0A3H       ; OTI   ; OUTPUT TO CONTROL PORT
                                                       ;    SETS REQUEST, PROM ON.
0008 0D                DCR     C                       ; ADJUST TO DATA PORT
0009 EDB3              DB      0EDH,0B3H       ; OTIR  ; SEND THE REQUEST BLOCK
               ;
000B 0680              MVI     B,BOOT2L                ; LOAD SECONDARY BOOT LENGTH
000D 210010            LXI     H,BOOT2                 ; POINT TO SECONDARY BOOT
0010 EDB2              DB      0EDH,0B2H       ; INIR  ; READ SECONDARY BOOT BLOCK
0012 C30010            JMP     BOOT2                   ; GO TO IT
               ;
0015 09        BLOCK   DB      REQUEST+PROMON          ; CONTROL BYTE
               ;
0016 01                DB      BLKTYP          ; REQUEST BLOCK - TYPE CODE
0017 4C                DB      BLKCOD                  ; REQUEST CODE - BOOT PROCESSOR
0018 00                DB      SYSCOD                  ; SYSTEM TO BOOT
0019 FF                DB      0FFH                    ; UNUSED - FILL WITH ONES
001A FF                DB      0FFH
001B FF                DB      0FFH
001C FF                DB      0FFH
001D FF                DB      0FFH
001E FF                DB      0FFH
001F FF                DB      0FFH
               ;
000B =         BLOCKL  EQU     $-BLOCK         ; LENGTH OF DATA TO OUTPUT
               ;
0020                   END
```

```
                        DB    14,20,26,6,12
                        DB    18,24,4,10,16,22
                ;
                DPB1    EQU     DPB0
                DPB2    EQU     DPB0
                DPB3    EQU     DPB0
                ;
                XLT1    EQU     XLT0
                XLT2    EQU     XLT0
                XLT3    EQU     XLT0
                        ENDIF
                ;
0000 =          XLTM26  EQU     0000H
                ;
                ;
                ;**BIOS**
                ;
                ; IPC TRANSFER LOGIC
FA53 0E8E       PNX0:   MVI     C,8EH   ;DATA PORT
FA55 EDB3               DW      0B3EDH  ;Z80 OTIR INSTRUCTION
FA57 DB8F               IN      08FH    ;STATUS
FA59 E604               ANI     04H     ;CHECK FOR OVERRUN
FA5B C8                 RZ
FA5C 3E00               MVI     A,00    ;CLEAR OVER-RUN CONDITION
FA5E D38F               OUT     08FH
FA60 3E04               MVI     A,04H   ;RETURN WITH ZERO SET AND 04H IN A
FA62 C9                 RET             ;TO INDICATE ERROR
FA63 0E8E       PNX1:   MVI     C,8EH   ;DATA PORT
FA65 EDB2               DW      0B2EDH  ;Z80 INIR INSTRUCTION
FA67 DB8F               IN      8FH     ;STATUS PORT
FA69 E604               ANI     04H     ;CHECK FOR OVER-RUN
FA6B C8                 RZ
FA6C 3E00               MVI     A,00    ;CLEAR OVER-RUN CONDITION
FA6E D38F               OUT     08FH
FA70 3E04               MVI     A,04H   ;RETURN WITH ZERO SET AND 04H IN A
FA72 C9                 RET

                ;
                ;
                ;
                ; BOOT
                ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                ; THE CPM SYSTEM.
                ;
```

EXCERPT FROM
SLAVE BIOS

ON ENTRY:
B= # OF BYTES
TO TRANSFER

HL = ADDRESS OF
BUFFER

```
FA73 318000     BOOT:   LXI  SP,80H      ;SET STACK POINTER.
FA76 AF                 XRA  A
FA77 320300             STA     IOBYTE   ;SET IOBYTE
FA7A 32D6FB             STA  NODISK
FA7D 320400             STA  CDISK       ;SELECT DRIVE ZERO
                        IF SIO2661
FA80 3ECA               MVI A,0CAH
FA82 D382               OUT CMODE
FA84 3E3E               MVI A,3EH
FA86 D382               OUT CMODE
FA88 3E27               MVI A,27H
FA8A D383               OUT CCOM
                        ENDIF
FA8C DB80               IN    CDATA      ;CLEAR KEYBOARD
FA8E 2132FD             LXI  H,SMSG      ;PRINT OPENING MESSAGE.
FA91 CDC6FC             CALL PMSG
FA94 CD9CFD     GOCPM:  CALL     BRKPT
FA97 3EC3               MVI  A,0C3H      ;PUT JMP TO WBOOT
 A99 320000             STA  0           ;ADR AT ZERO.
FA9C 2103FA             LXI  H,WBOOTE
FA9F 220100             SHLD 1
FAA2 320500             STA  5
FAA5 2106EC             LXI  H,BDOS      ;PUT JUMP TO BDOS
FAA8 220600             SHLD 6           ;AT ADR 5,6,7.
 AAB 218000             LXI  H,80H       ;SET DEFAULT DMA ADR.
FAAE 22E4FD             SHLD DMAADD
FAB1 3A0400             LDA  CDISK       ;GET DISK NUMBER TO
FAB4 4F                 MOV  C,A         ;PASS TO CCP IN C.
FAB5 C300E4             JMP  CPMB        ;JUMP TO CCP.

                ;
                ; WARM-BOOT:  READ ALL OF CPM BACK IN
                ; EXCEPT BIOS, THEN JUMP TO CCP.
                ;
FAB8 318000     WBOOT:   LXI SP,80H      ;SET STACK POINTER.
FABB 3A0400             LDA  CDISK       ;SAVE DISK NUMBER.
FABE 32E3FD             STA  TEMP
FAC1 3A00FA             LDA     BIOS     ;SAVE 1ST BYTE OF BIOS
FAC4 32E2FD             STA     SAVBUF
FAC7 3E4C               MVI     A,IPL
FAC9 322CFB             STA     WRQCOD
FACC 3E00               MVI     A,0
FACE 3230FB             STA     WRECNO
FAD1 2180E3             LXI     H,CPMB-80H
```

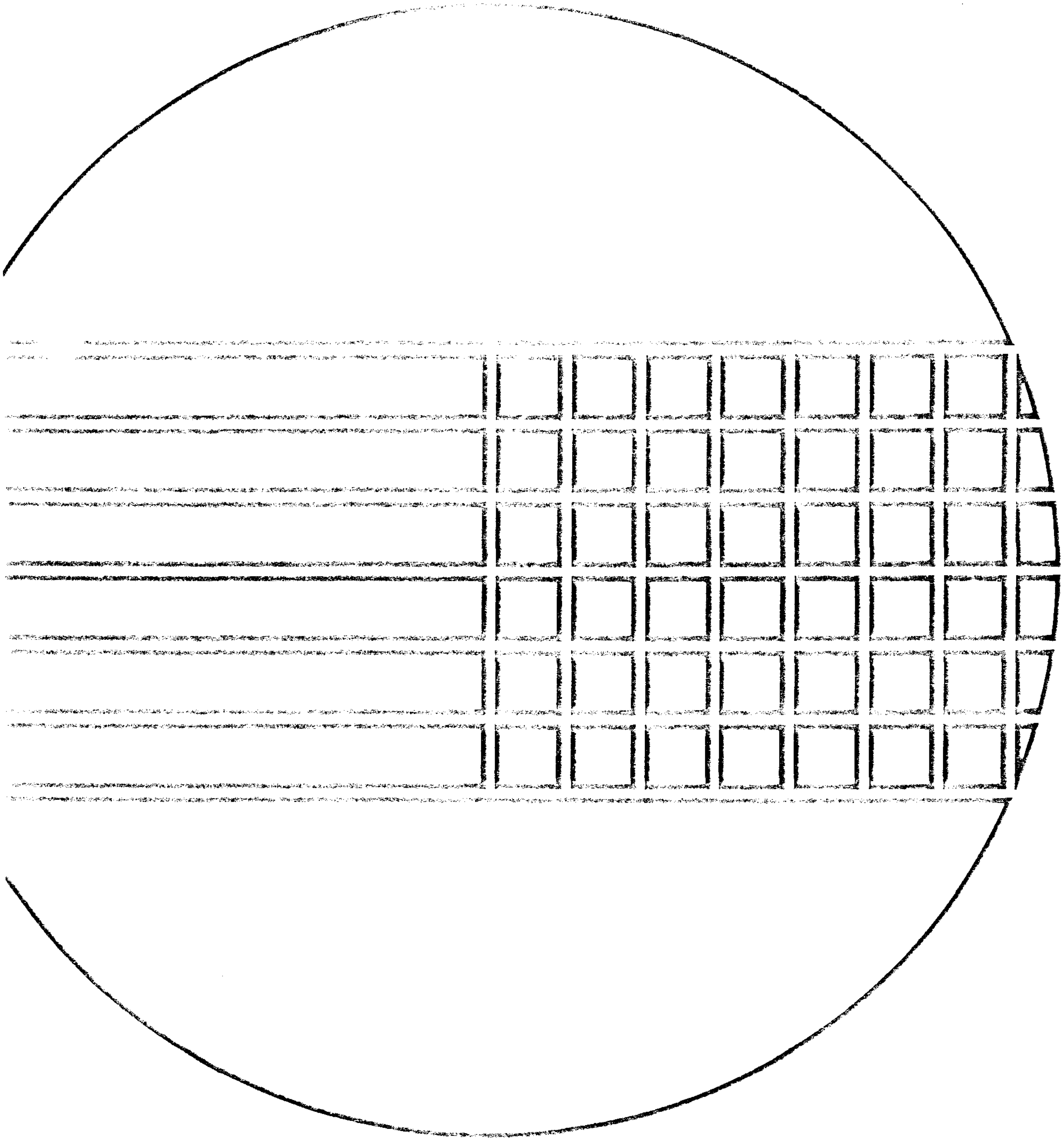
CONSOLE
INITIALIZATION

APPENDIX A

SIGNETICS 2661 DATA SHEET

Reproduced with the permission of
Signetics Corporation

ANCED
OGRAMMABLE
MMUNICATIONS
ERFACE (EPCI)



Signetics reserves the right to make changes in the products contained in this book in order to improve design or performance and to supply the best possible products. Signetics also assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representations that the circuits are free from patent infringement. Applications for any integrated circuits contained in this publication are for illustration purposes only and Signetics makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification. Reproduction of any portion hereof without the prior written consent of Signetics is prohibited.

signetics

DESCRIPTION

The Signetics 2661 EPCI is a universal synchronous/asynchronous data communications controller chip that is an enhanced pin compatible version of the 2651. It interfaces directly to most 8-bit microprocessors and may be used in a polled or interrupt driven system environment. The 2661 accepts programmed instructions from the microprocessor while supporting many serial data communications disciplines—synchronous and asynchronous—in the full or half-duplex mode. Special support for BISYNC is provided.

The EPCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The 2661 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (-1, -2, -3) has a different set of baud rates.

The EPCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- Synchronous operation
  - 5 to 8-bit characters plus parity
  - Single or double SYN operation
  - Internal or external character synchronization
  - Transparent or non-transparent mode
  - Transparent mode DLE stuffing (Tx) and detection (Rx)
  - Automatic SYN or DLE-SYN insertion
  - SYN, DLE and DLE-SYN stripping
  - Odd, even, or no parity
  - Local or remote maintanance loop back mode
  - Baud rate: dc to 1M bps (1X clock)
- Asynchronous operation
  - 5 to 8-bit characters plus parity
  - 1, 1½ or 2 stop bits transmitted
  - Odd, even, or no parity
  - Parity, overrun and framing error detection
  - Line break detection and generation
  - False start bit detection
  - Automatic serial echo mode (echoplex)
  - Local or remote maintanance loop back mode
  - Baud rate: dc to 1M bps (1X clock)
    dc to 62.5K bps (16X clock)
    dc to 15.625K bps (64X clock)

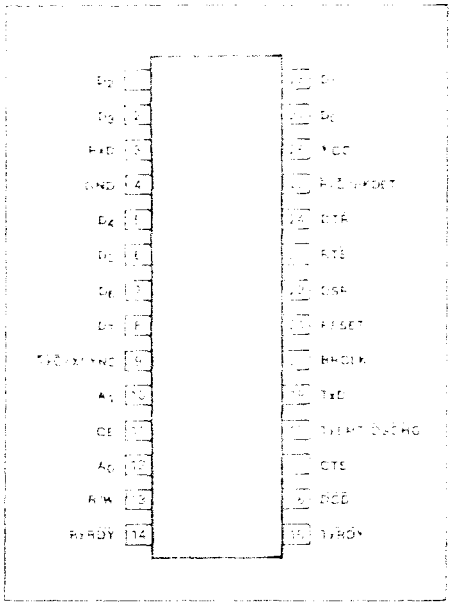
OTHER FEATURES

- Internal or external baud rate clock
- 3 baud rate sets (2661-1, -2, -3)
- 16 internal rates for each set
- Double buffered transmitter and receiver
- Dynamic character length switching
- Full or half duplex operation
- Fully compatible with 2650 CPU
- TTL compatible inputs and outputs
- RxC and TxC pins are short circuit protected
- 3 open drain MOS outputs can be wire-ORed
- Single 5V power supply
- No system clock required
- 28-pin dual in-line package

APPLICATIONS

- Intelligent terminals
- Network processors
- Front end processors
- Remote data concentrators
- Computer to computer links
- Serial peripherals
- BISYNC adaptors

PIN CONFIGURATION



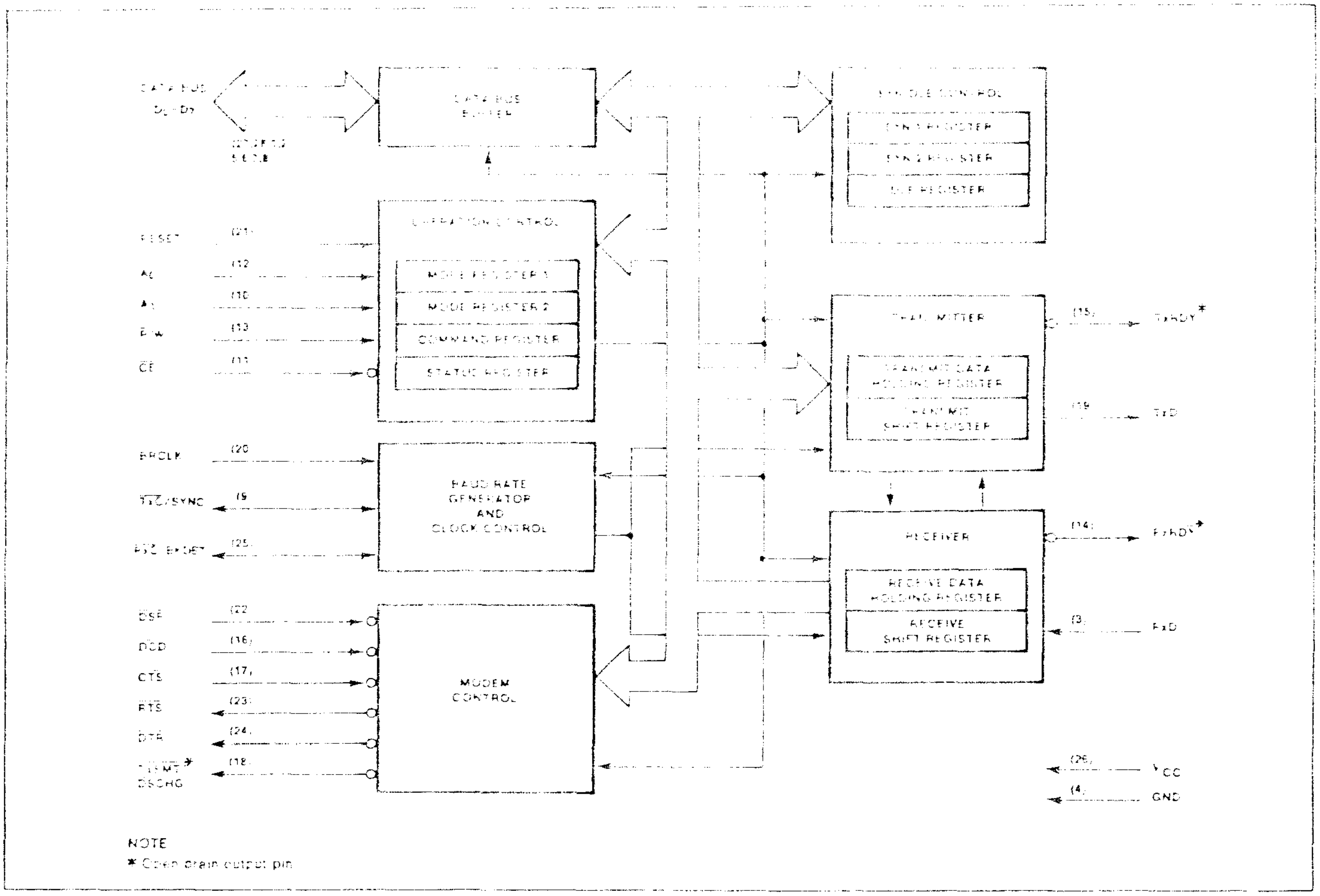
ORDERING CODE

PACKAGES		COMMERCIAL RANGES V_{CC} = 5V ± 5%, T_A = 0°C to 70°C
Ceramic DIP	2661-1I 2661-2I 2661-3I	See table 1 for baud rates
Plastic DIP	2661-1N 2661-2N 2661-3N	See table 1 for baud rates

PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1, 2,5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	$R/\overline{W}$	Read or write command	I
11	$\overline{CE}$	Chip enable input	I
22	$\overline{DSR}$	Data set ready	I
24	$\overline{DTR}$	Data terminal ready	O
23	$\overline{RTS}$	Request to send	O
17	$\overline{CTS}$	Clear to send	I
16	$\overline{DCD}$	Data carrier detected	I
18	$\overline{TxEMT}/\overline{DSCHG}$	Transmitter empty or data set change	O
9	$\overline{TxC}/XSYNC$	Transmitter clock/external SYNC	I/O
25	$\overline{RxC}/BKDET$	Receiver clock/break detect	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	$\overline{TxRDY}$	Transmitter ready	O
14	$\overline{RxRDY}$	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

BLOCK DIAGRAM



BLOCK DIAGRAM

The EPCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS 2661–1 (BRCLK = 4.9152MHz)

MR23–20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS (Cont'd)

2661-2 (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

2661-3 (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE

16X clock is used in asynchronous modes. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC

Table 2 CPU-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal EPCI registers.
$\overline{R}$/W	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{R}$/W, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the three-state condition.
D_7-D_0	8,7,6,5, 2,1,28,17	I/O	8-bit, three-state data bus used to transfer commands, data and status between EPCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}$/ $\overline{DSCHG}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

OPERATION

The functional operation of the 2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2661 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{RxC}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (*framing error*), the *receiver will* interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (*RxD is low for the entire* character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27-MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the break detection timing diagram.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	PIN NO	INPUT/ OUTPUT	FUNCTION
BRCLK	20	I	Clock input to the internal baud rate generator (see table 1). Not required if external receiver and transmitter clocks are used.
*Rx$\overline{\text{C}}$/BKDET	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
*Tx$\overline{\text{C}}$/XSYNC	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output or an external jam synchronization input.
RxD	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{\text{TxEMT/DSCHG}}$ when its state changes if CR2 or CR0 = 1.
$\overline{\text{DCD}}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{TxEMT/DSCHG}}$ when its state changes if CR2 or CR0 = 1. If $\overline{\text{DCD}}$ goes high while receiving, the RxC is internally inhibited.
$\overline{\text{CTS}}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. If the transmit shift register is not empty when CR5 is reset (1 to 0), then $\overline{\text{RTS}}$ will go high one TxC time after the last serial bit is transmitted.

NOTE
*Rx$\overline{\text{C}}$ and Tx$\overline{\text{C}}$ outputs have short circuit protection max. C_L = 100pF. Outputs become open circuited upon detection of a zero pulled high or a one pulled low

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN (CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization.) When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27-MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The 2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the TxEMT/DSCHG output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

In the synchronous mode, when the 2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the 2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the initialization process appears in figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}/W$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and

Table 4 2661 REGISTER ADDRESSING

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	Tri-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers 1/2
0	1	0	1	Write mode registers 1/2
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC characteristics section for timing requirements.

2661 INITIALIZATION FLOW CHART

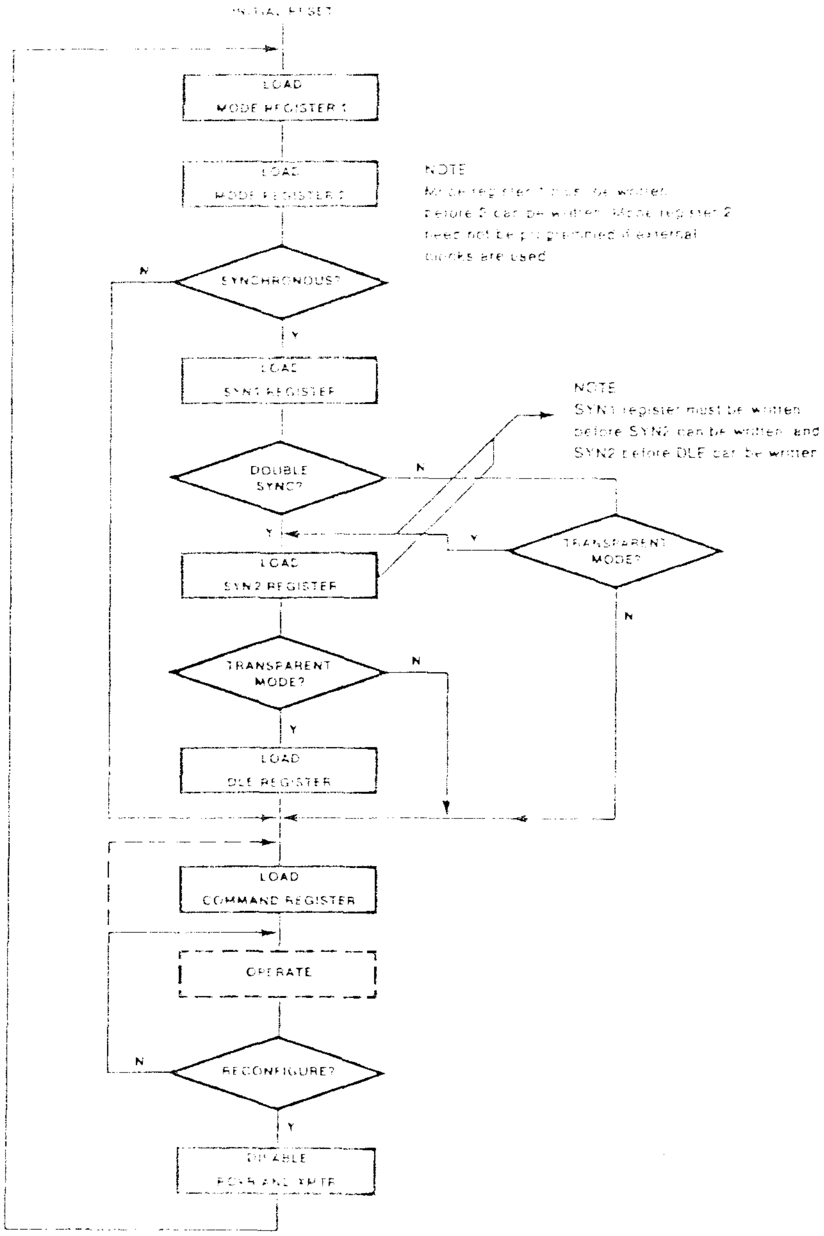


Figure 1

R/W = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The 2661 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12-MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN=0 or when TxEN = 1 and the transmitter is marking in half duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12-15 must be changed within n bit times of the active going state of RxRDY/TxRDY. Transparent and non-transparent mode changes (MR16) must occur within n-1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (−1, −2, −3). Version 1 and 2 specify a 4.9152 MHz TTL input at BRCLK (pin 20); version 3 specifies a 5.0688 MHz input which is identical to the Signetics 2651. MR23-20 are don't cares if external clocks are selected (MR25-MR24 = 0). The individual rates are given in table 1.

MR24-MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 9 and 25. Refer to table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{RxC}$ rising edge. Disabling the receiver causes $\overline{RxRDY}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state

Table 5 MODE REGISTER 1 (MR 1)

MR17	MR16	MR15	MR14	MR13 MR12	MR11 MR10
Sync/Async		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor
Async: Stop Bit Length 00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate
Sync: Number of SYN char 0 = Double SYN 1 = Single SYN	Sync: Transparency Control 0 = Normal 1 = Transparent				

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6 MODE REGISTER 2 (MR2)

MR27–MR24											MR23–MR20
	TxC	RxC	Pin 9	Pin 25		TxC	RxC	Pin 9	Pin 25	Mode	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	XSYNC[1]	RxC/TxC	sync	See baud rates in table 1
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	async	
0010	I	E	1X	RxC	1010	I	E	XSYNC[1]	RxC	sync	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	async	
0100	E	E	TxC	RxC	1100	E	E	XSYNC	RxC/TxC	sync	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	async	
0110	I	E	16X	RxC	1110	I	E	XSYNC[1]	RxC	sync	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	async	

NOTES

[1] When pin 9 is programmed as XSYNC input, SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled.

E = External clock

I = Internal clock (BRG)

1X and 16X are clock outputs

Table 7 COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	Request To Send	Reset Error	Sync/Async	Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{RTS}$ output high one clock time after TxSR serialization 1 = Force $\overline{RTS}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect)	Async: Force break 0 = Normal 1 = Force break Sync: Send DLE 0 = Normal 1 = Send DLE	0 = Disable 1 = Enable	0 = Force $\overline{DTR}$ output high 1 = Force $\overline{DTR}$ output low	0 = Disable 1 = Enable

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ input is high 1 = $\overline{DSR}$ input is low	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	Async: 0 = Normal 1 = Framing Error Sync: 0 = Normal 1 = SYN detected	0 = Normal 1 = Overrun Error	Async: 0 = Normal 1 = Parity error Sync: 0 = Normal 1 = Parity error or DLE received	0 = Normal 1 = Change in $\overline{DSR}$, or $\overline{DCD}$, or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty

(high) while $\overline{TxRDY}$ and $\overline{TxEMT}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{DTR}$ and $\overline{RTS}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE—non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

Table 9 2651 EPCI vs 2651 PCI

FEATURE	EPCI	PCI
1 MR2 Bit 6, 7	Control pin 9, 25	Not used
2 DLE detect-SR3	SR3 = 0 for DLE-DLE, DLE SYNC1	SR3 = 1 for DLE-DLE, DLE SYNC1
3 Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4 Send DLE-CR3	One time command	Reset via CR3 on next $\overline{TxRDY}$
5 DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6 SYNC1 stripping in double sync non-transparent mode	All SYNC1	First SYNC1 of pair
7. Baud rate versions	Three	One
8 Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{TxRDY}$ changing from 0 to 1	Reset CR0 when $\overline{TxEMT}$ goes from 1 to 0. Then reset CR5 when $\overline{TxEMT}$ goes from 0 to 1
9 Break detect	Pin 25[1]	FE and null character
10. Stop bit searched	One	Two
11. External jam sync	Pin 9[2]	No
12. Data bus timing	Improved over 2651	—
13 Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES

[1] Internal BRG used for RxC

[2] Internal BRG used for TxC.

When CR5 (RTS) is set, the $\overline{RTS}$ pin is forced low and the transmit serial logic is enabled. A 1 to 0 transition of CR5 will cause $\overline{RTS}$ to go high (inactive) one TxC time after the last serial bit has been transmitted (if the transmit shift register was not empty).

The EPCI can operate in one of four sub-modes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7-CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{TxRDY}$ output = 1.
4. The $\overline{TxEMT/DSCHG}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loop back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{DTR}$ is connected to $\overline{DCD}$ and $\overline{RTS}$ is connected to $\overline{CTS}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{DTR}$, $\overline{RTS}$ and $\overline{TxD}$ outputs are held high.
5. The $\overline{CTS}$, $\overline{DCD}$, $\overline{DSR}$ and RxD inputs are ignored.

Additional requirements to operate in the local loop back mode are that CR0 (TxEN), CR1 (DTR) and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loop back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{RxRDY}$, $\overline{TxRDY}$, and $\overline{TxEMT/DSCHG}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{TxRDY}$ output pin is low. In the automatic echo and remote loop back modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{RxRDY}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{TxEMT/DSCHG}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit, i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{DCD}$ and $\overline{DSR}$ inputs respectively. A low input sets its corresponding status bit, and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	$I_{OL} = 2.2mA$			0.4	
V_{OH}[7]	High	$I_{OH} = -400\mu A$	2.4			
I_{IL}	Input leakage current	$V_{IN} = 0$ to 5.5 V			10	μA
	3-state output leakage current					μA
I_{LH}	Data bus high	$V_O = 4.0V$			10	
I_{LL}	Data bus low	$V_O = 0.45V$			10	
I_{CC}	Power supply current				150	mA

CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance					pF
C_{IN}	Input				20	
C_{OUT}	Output	fc = 1MHz			20	
$C_{I/O}$	Input/Output	Unmeasured pins tied to ground			20	

Notes on following page

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ± 5%[4,5,6]

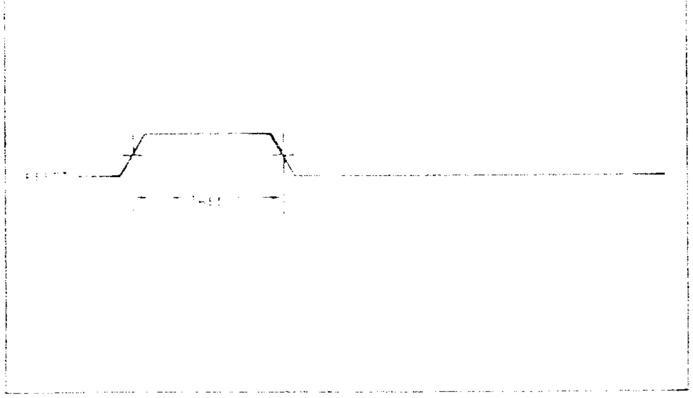
PARAMETER		TEST CONDITIONS	Min	Typ	Max	UNIT
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		250			
	Setup and hold time					ns
t_{AS}	Address setup		10			
t_{AH}	Address hold		10			
t_{CS}	$\overline{R}$/W control setup		10			
t_{CH}	$\overline{R}$/W control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	C_L = 150pF			200	ns
t_{DF}	Data bus floating time for read	C_L = 150pF			100	
t_{CED}	CE to CE delay		600			
	Input clock frequency					MHz
f_{BRG}	Baud rate generator (2661-1,-2)		1.0	4.9152	4.9202	
f_{BRG}	Baud rate generator (2661-3)		1.0	5.0688	5.0738	
$f_{R/T}$	$\overline{TxC}$ or $\overline{RxC}$		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high (2661-1,-2)		75			
t_{BRH}[9]	Baud rate high (2661-3)		70			
t_{BRL}[9]	Baud rate low (2661-1,-2)		75			
t_{BRL}[9]	Baud rate low (2661-3)		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		480			
$t_{R/TL}$	$\overline{TxC}$ or $\overline{RxC}$ low		480			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	C_L = 150pF			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	C_L = 150pF		0		

NOTES

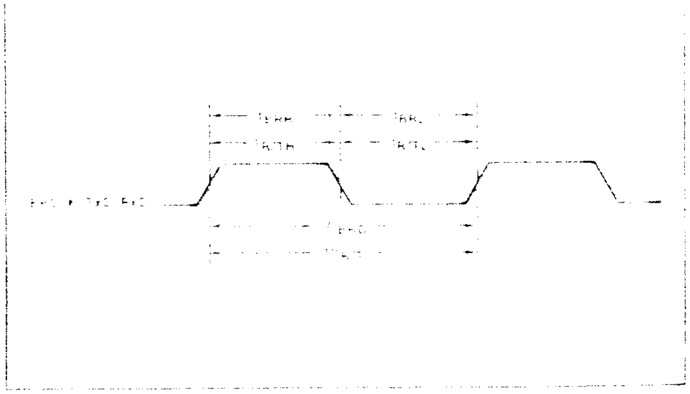
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (IC ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2.0V for outputs. Input levels swing between 0.4V and 2.4V, with a transition time of 20ns maximum.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. $\overline{TxRDY}$, $\overline{RxRDY}$ and $\overline{TxEMT/DSCHG}$ outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688 MHz f_{BRG} (2661-3) and 4.9152 MHz f_{BRG} (2661-1,-2), t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.

TIMING DIAGRAMS

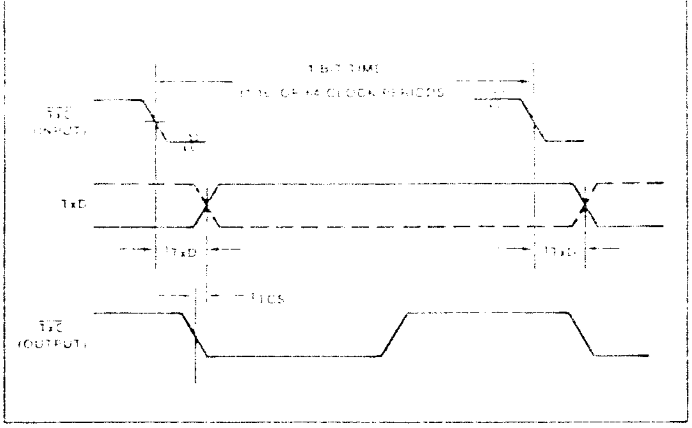
RESET



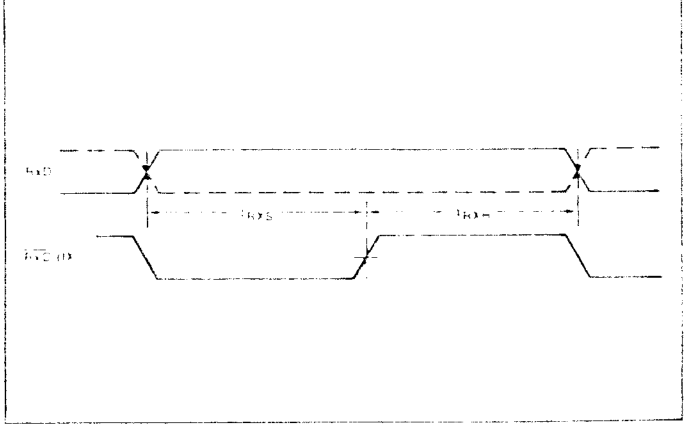
CLOCK



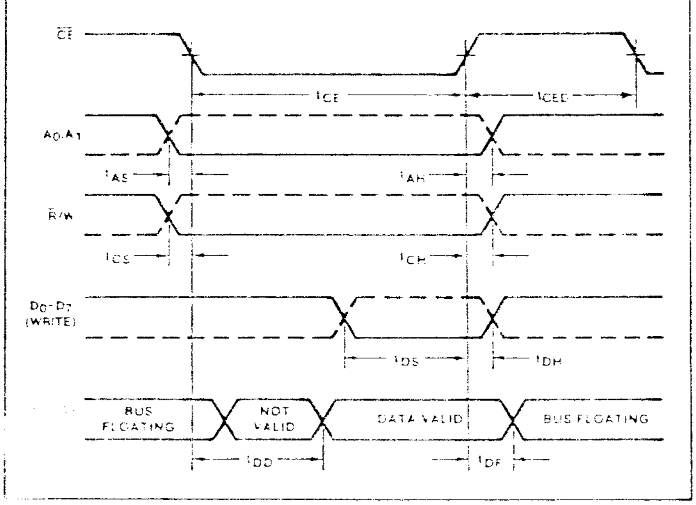
TRANSMIT



RECEIVE

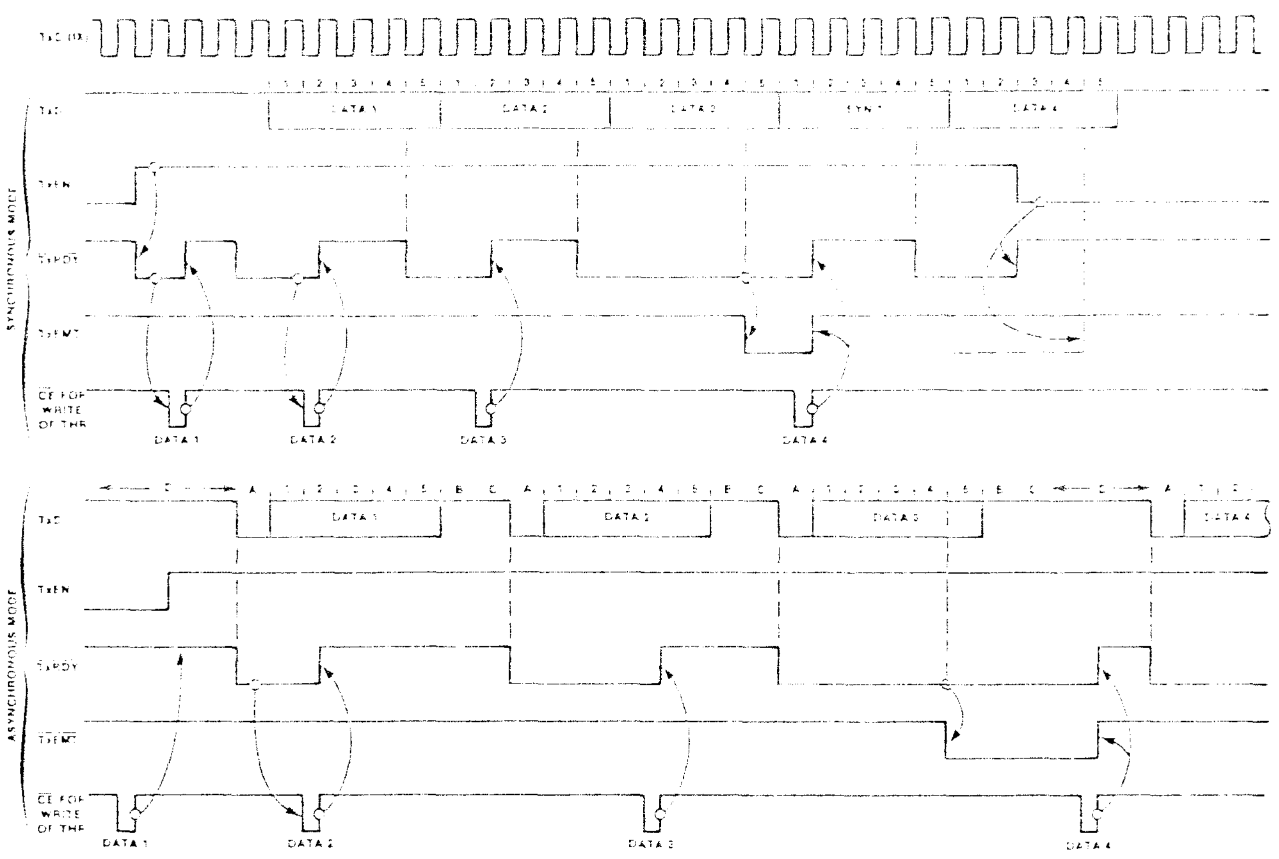


READ AND WRITE



TIMING DIAGRAMS (Cont'd)

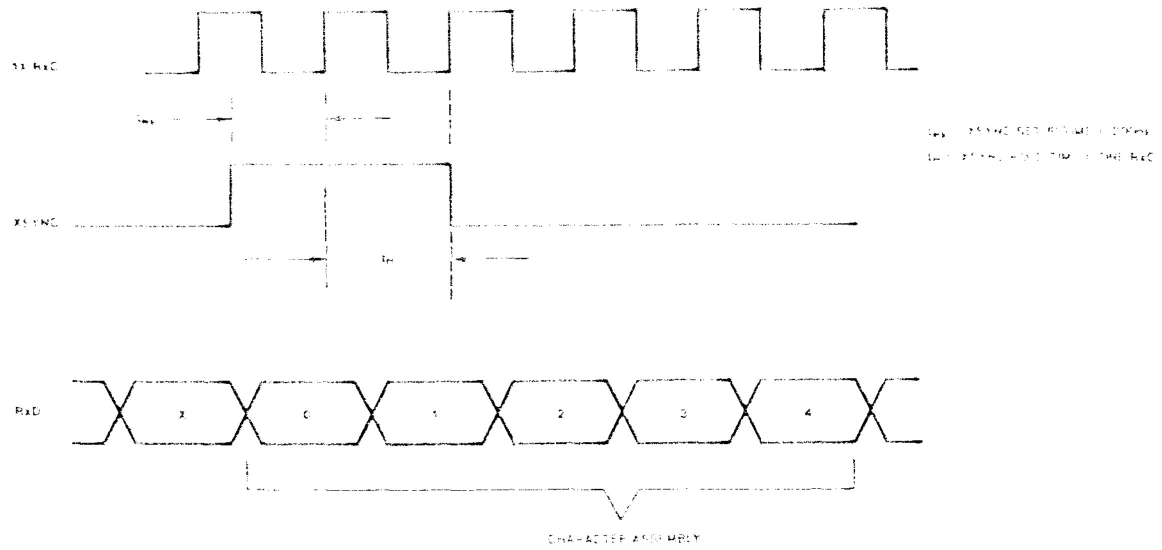
TxRDY, TxEMT (Shown for 5 bit characters, no parity, 2 stop bits [in asynchronous mode])



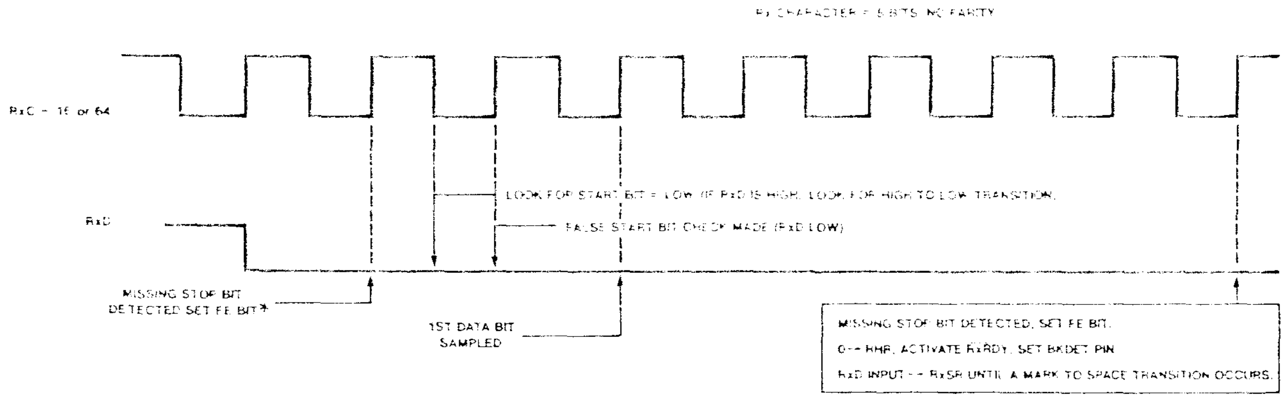
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

TIMING DIAGRAMS (Cont'd)

EXTERNAL SYNCHRONIZATION WITH XSYNC


1X RxC
tSXS
tHXS
XSYNC
tHXS
RxD
x
0
1
2
3
4
CHARACTER ASSEMBLY
tSXS = XSYNC SETUP TIME = 300ns
tHXS = XSYNC HOLD TIME = ONE RxC

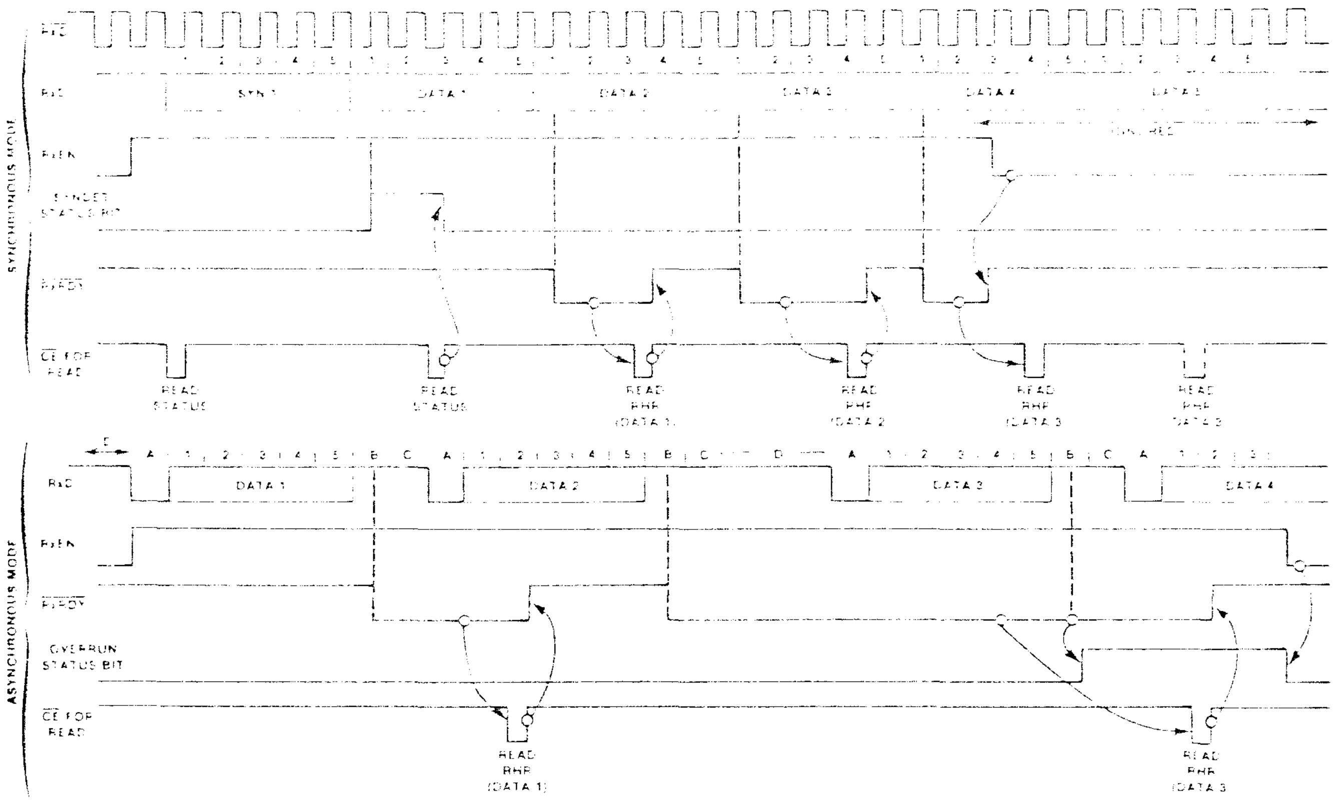
BREAK DETECTION TIMING


Rx CHARACTER = 5 BITS, NO PARITY
RxC = 16 or 64
RxD
LOOK FOR START BIT = LOW (IF RxD IS HIGH, LOOK FOR HIGH TO LOW TRANSITION)
FALSE START BIT CHECK MADE (RxD LOW)
MISSING STOP BIT DETECTED SET FE BIT*
1ST DATA BIT SAMPLED
MISSING STOP BIT DETECTED, SET FE BIT.
0 → RHR, ACTIVATE RxRDY, SET BKDET PIN
RxD INPUT → RxSR UNTIL A MARK TO SPACE TRANSITION OCCURS.

NOTE
* If the stop bit is present, the start bit search will commence immediately.

TIMING DIAGRAMS (Cont'd)

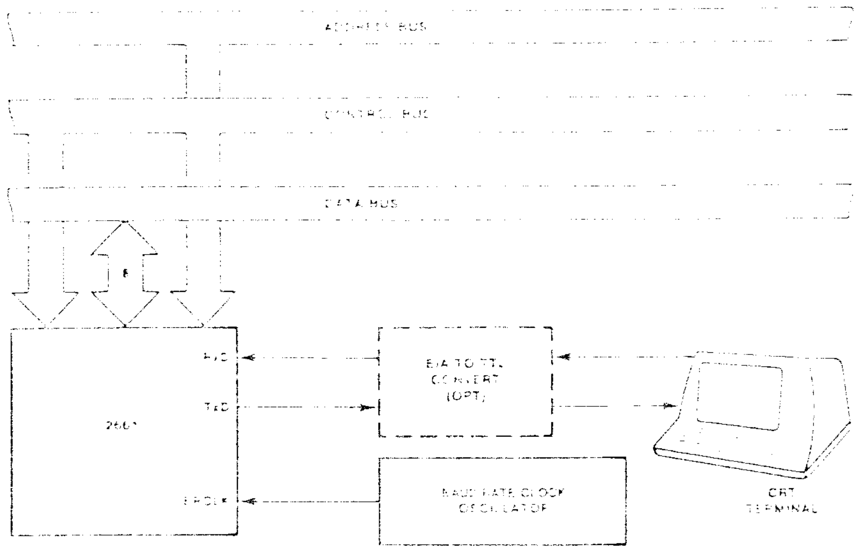
RxRDY (Shown for 5 bit characters, no parity, 2 stop bits [in asynchronous mode])



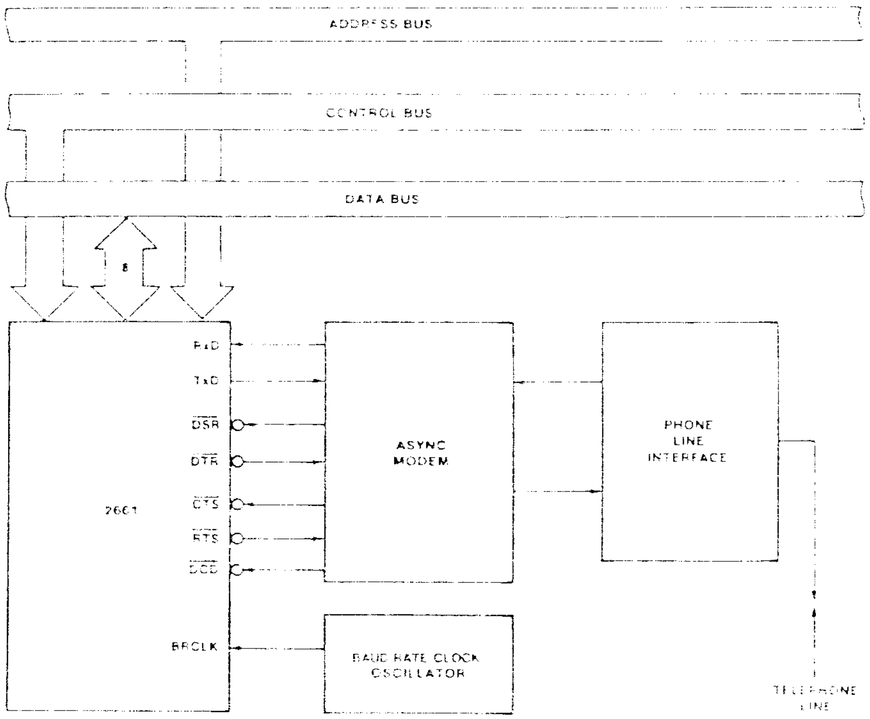
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected

TYPICAL APPLICATIONS

ASYNCHRONOUS INTERFACE TO CRT TERMINAL

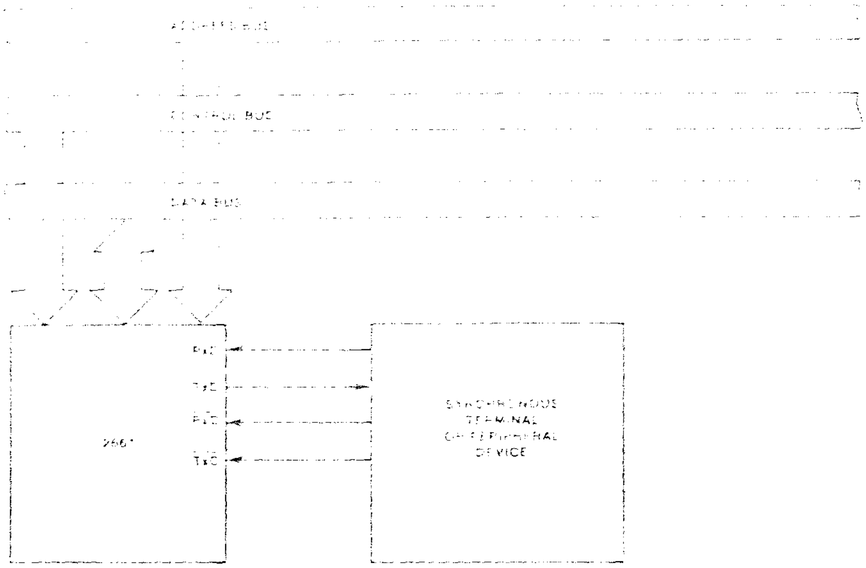


ASYNCHRONOUS INTERFACE TO TELEPHONE LINES

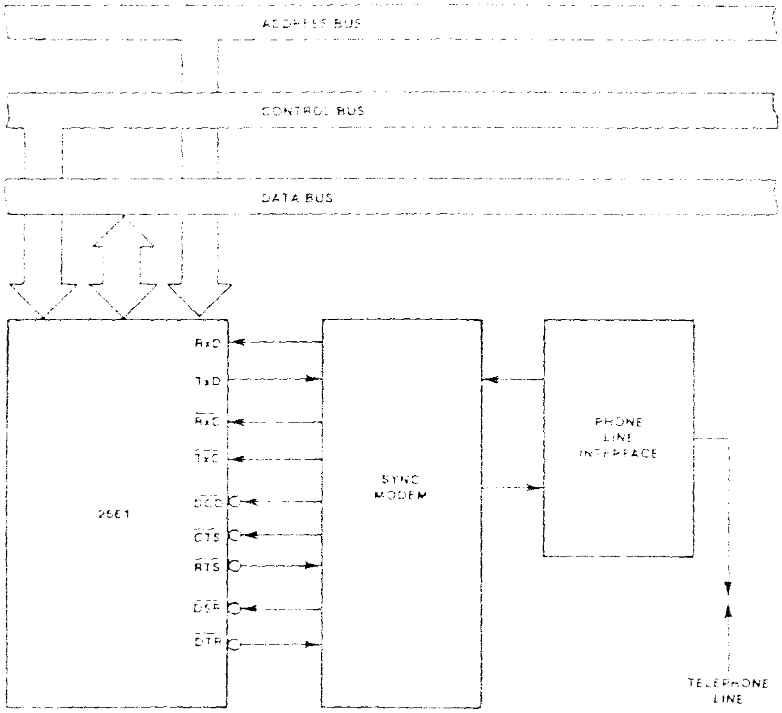


TYPICAL APPLICATIONS (Cont'd)

SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES



APPENDIX D

AMD 9519A DATA SHEET

Universal Interrupt Controller
Advanced Micro Devices
Advanced MOS LSI

DISTINCTIVE CHARACTERISTICS

- Eight individually maskable interrupt inputs reduce CPU overhead
- Unlimited interrupt channel expansion with no extra hardware
- Programmable 1- to 4-byte response provides vector address and message protocol for 8-bit CPUs
- Rotating and fixed priority resolution logic
- Software interrupt request capability
- Common vector and polled mode options
- Automatic hardware clear of in-service interrupts reduces software overhead
- Polarity control of interrupt inputs and outputs
- Reset minimizes software initialization by automatically generating CALL to location zero
- Single +5V supply
- 100% MIL-STD-883 quality assurance testing increases reliability

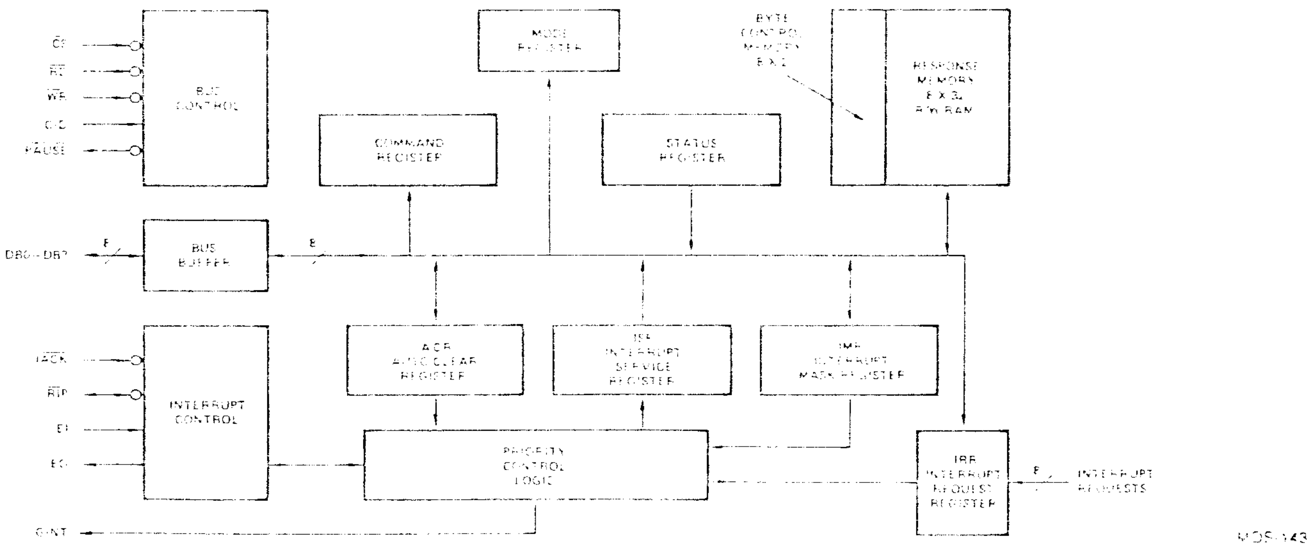
GENERAL DESCRIPTION

The Am9519A Universal Interrupt Controller is a processor support circuit that provides a powerful interrupt structure to increase the efficiency and versatility of microcomputer-based systems. A single Am9519A manages up to eight maskable interrupt request inputs, resolves priorities and supplies up to four bytes of fully programmable response for each interrupt. It uses a simple expansion structure that allows many units to be cascaded for control of large numbers of interrupts. Several programmable control features are provided to enhance system flexibility and optimization.

The Universal Interrupt Controller is designed with a general purpose interface to facilitate its use with a wide range of digital systems, including most popular 8-bit microprocessors. Since the response bytes are fully programmable, any instruction or vectoring protocol appropriate for the host processor may be used.

When the Am9519A controller receives an unmasked Interrupt Request it issues a Group Interrupt output to the CPU. When the interrupt is acknowledged, the controller outputs the one-to-four byte response associated with the highest priority unmasked interrupt request. The ability of the CPU to set interrupt requests under software control permits hardware prioritization of software tasks and aids system diagnostic and maintenance procedures.

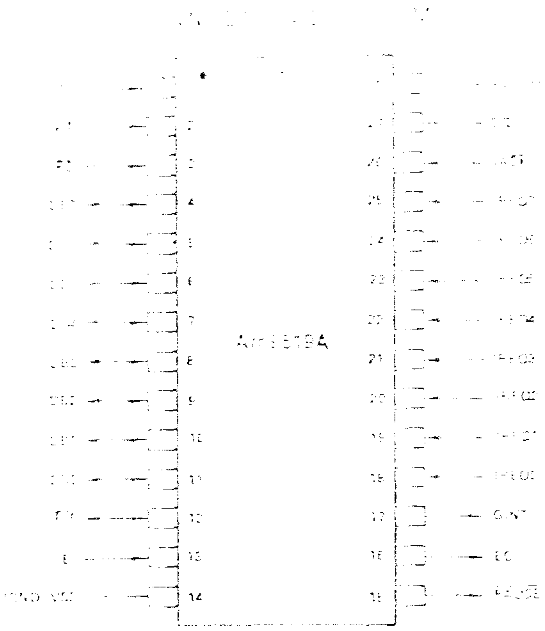
BLOCK DIAGRAM



ORDERING INFORMATION

Package Type	Ambient Temperature	Timing Options	
		Am9519A	Am9519A-1
Hermetic DIP*	0°C ≤ T_A ≤ +70°C	AM9519ADC CC	AM9519A-1DC CC
	−55°C ≤ T_A ≤ +125°C	AM9519ADM	
Molded DIP	0°C ≤ T_A ≤ +70°C	AM9519APC	AM9519A-1PC

*DC = Side Brazed Ceramic   CC = Cerdip



Top View

Pin 1 is marked for orientation.

MOS-019

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt Power Supply
VSS: Ground

DB0 – DB7 (Data Bus, Input/Output)

The eight bidirectional data bus signals are used to transfer information between the Am9519A and the system data bus. The direction of transfer is controlled by the $\overline{\text{IACK}}$, and $\overline{\text{RD}}$ input signals. Programming and control information are written into the device; status and response data are output by it.

$\overline{\text{CS}}$ (Chip Select, Input)

The active low Chip Select input enables read and write operations on the data bus. Interrupt acknowledge responses are not conditioned by $\overline{\text{CS}}$.

$\overline{\text{RD}}$ (Read, Input)

The active low Read signal is conditioned by $\overline{\text{CS}}$ and indicates that information is to be transferred from the Am9519A to the data bus.

$\overline{\text{WR}}$ (Write, Input)

The active low Write signal is conditioned by $\overline{\text{CS}}$ and indicates that data bus information is to be transferred from the data bus to a location within the Am9519A.

C/$\overline{\text{D}}$ (Control/Data, Input)

The C/$\overline{\text{D}}$ control signal selects source and destination locations for data bus read and write operations. Data read or write transfers are made to or from preselected internal registers or memory locations. Control write operations load the command register and control read operations output the status register.

IREQ0 – IREQ7 (Interrupt Request, Input)

The Interrupt Request signals are used by external devices to indicate that service by the host CPU is desired. IREQ inputs are accepted asynchronously and they may be programmed for either a high-to-low or low-to-high [illegible]

$\overline{\text{RIP}}$ (Response In Process, Input/Output)

Response In Process is a bidirectional signal used when two or more Am9519A circuits are cascaded. It permits multibyte response transfers to be completed without interference from higher priority interrupts. An Am9519A that is responding to an acknowledged interrupt will treat $\overline{\text{RIP}}$ as an output and hold it low until the acknowledge response is finished. An Am9519A without an acknowledged interrupt will treat $\overline{\text{RIP}}$ as an input and will ignore $\overline{\text{IACK}}$ pulses as long as $\overline{\text{RIP}}$ is low. The $\overline{\text{RIP}}$ output is open drain and requires an external pullup resistor to VCC.

$\overline{\text{IACK}}$ (Interrupt Acknowledge, Input)

The active low Interrupt Acknowledge line indicates that the external system is asking for interrupt response information. Depending on the programmed state of the Am9519A, it will accept 1, 2, 3 or 4 $\overline{\text{IACK}}$ pulses; one response byte is transferred per pulse. The first $\overline{\text{IACK}}$ pulse causes selection of the highest priority unmasked pending interrupt request and generates a $\overline{\text{RIP}}$ output signal.

$\overline{\text{PAUSE}}$ (Pause, Output)

The active low Pause signal is used to coordinate interrupt responses with data bus and control timing. Pause goes low when the first $\overline{\text{IACK}}$ is received and remains low until $\overline{\text{RIP}}$ goes low. The external system can use Pause to stretch the acknowledge cycle and allow the control timing to automatically adjust to the actual priority resolution delays in the interrupt system. Second, third and fourth response bytes do not cause Pause to go low. Pause is an open drain output and requires an external pullup resistor to VCC.

EO (Enable Out, Output)

The active high EO signal is used to implement daisy-chained cascading of several Am9519A circuits. EO is connected to the EI input of the next lower priority chip. On receipt of an interrupt acknowledge, each EO will go inactive until it has been determined that no valid interrupt request is pending on that chip. If an active request is present, EO remains low. EO is also held low when the master mask bit is active, thus disabling all lower priority chips.

EI (Enable In, Input)

The active high EI signal is used to implement daisy-chained cascading of several Am9519A circuits. EI is connected to EO of the next higher priority chip. It may also be used as a hardware disable input for the interrupt system. When EI is low $\overline{\text{IACK}}$ inputs are ignored. EI is internally pulled up to VCC so that no external pullup is needed when EI is not used.

GINT (Group Interrupt, Output)

The Group Interrupt output signal indicates that at least one unmasked interrupt request is pending. It may be programmed for active high or active low polarity. When active low, the output is open drain and requires an external pull up resistor to VCC.

Interrupt Request Register (IRR): The 8-bit IRR is used to store pending interrupt requests. A bit in the IRR is set whenever the corresponding IREQ input goes active. Bits may also be set under program control from the CPU, thus permitting software generated interrupts. IRR bits may be cleared under program control. An IRR bit is automatically cleared when its interrupt is acknowledged. All IRR bits are cleared by a reset function.

Interrupt Service Register (ISR): The 8-bit ISR contains one bit for each IREQ input. It is used to indicate that a pending interrupt has been acknowledged and to mask all lower priority interrupts. When a bit is set by the acknowledge logic in the ISR, the corresponding IRR bit is cleared. If an acknowledged interrupt is not programmed to be automatically cleared, its ISR bit must be cleared by the CPU under program control when it is desired to permit interrupts from lower priority devices. When the interrupt is programmed for automatic clearing, the ISR bit is automatically reset during the acknowledge sequence. All ISR bits are cleared by a reset function.

Interrupt Mask Register (IMR): The 8-bit IMR is used to enable or disable the individual interrupt inputs. The IMR bits correspond to the IREQ inputs and all eight may be loaded, set or cleared in parallel under program control. In addition, individual IMR bits may be set or cleared by the CPU. A reset function will set all eight mask bits, disabling all requests. A mask bit that is set does not disable the IRR, and an IREQ that arrives while a corresponding mask bit is set will cause an interrupt later when the mask bit is cleared. Only unmasked interrupt inputs can generate a Group Interrupt output.

Response Memory: An 8 x 32 read/write response memory is included in the Am9519A. It is used to store up to four bytes of response information for each of the eight interrupt request inputs. All bits in the memory are programmable, allowing any desired vector, opcode, instruction or other data to be entered. The Am9519A transfers the interrupt response information for the highest priority unmasked interrupt from the memory to the data bus when the $\overline{IACK}$ input is active.

Auto Clear Register: The 8-bit Auto Clear register contains one bit for each IREQ input and specifies the operating mode for each of the ISR bits. When an auto clear bit is off, the corresponding ISR bit is set when the interrupt is acknowledged and is cleared by software command. When an auto clear bit is on, the corresponding ISR bit is cleared by the hardware at the end of the acknowledge sequence. A reset function clears all auto clear bits.

Status Register: The 8-bit Status register contains information concerning the internal state of the chip. It is especially useful when operating in the polled mode in order to identify interrupting devices. Figure 1 shows the status register bit assignments. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit status register bit assignments. Bits S0-S2 are set asynchronously to a status register read operation. It is recommended to read the register twice and to compare the binary vectors for equality prior to the proceeding with device service in polled mode. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit 3). The Status register is read by executing a read operation ($\overline{CS}$ = 0, RS = 0) with the control location selected (C/$\overline{D}$ = 1).

Mode Register: The 8-bit Mode register controls the operating options of the Am9519A. Figure 2 shows the bit assignments for the Mode register. The five low order mode bits (0 through 4) are loaded in parallel by command. Bits 5, 6 and 7 are controlled by separate commands. (See Figure 4.) The Mode register cannot be read out directly to the data bus, but Mode bits 0, 2 and 7 are available as part of the Status register.

Command Register: The 8-bit Command register stores the last command entered. Depending upon the command opcode, it may initiate internal actions or precondition the part for subsequent data bus transfers. The Command register is loaded by executing a write operation ($\overline{WR}$ = 0) with the control location selected (C/$\overline{D}$ = 1), as shown in Figure 3.

Byte Count Register: The length in bytes of the response associated with each interrupt is independently programmed so that different interrupts may have different length responses. The byte count for each response is stored in eight 2-bit Byte Count registers. For a given interrupt the Am9519A will expect to receive a number of $\overline{IACK}$ pulses that equals the corresponding byte count, and will hold RIP low until the count is satisfied.

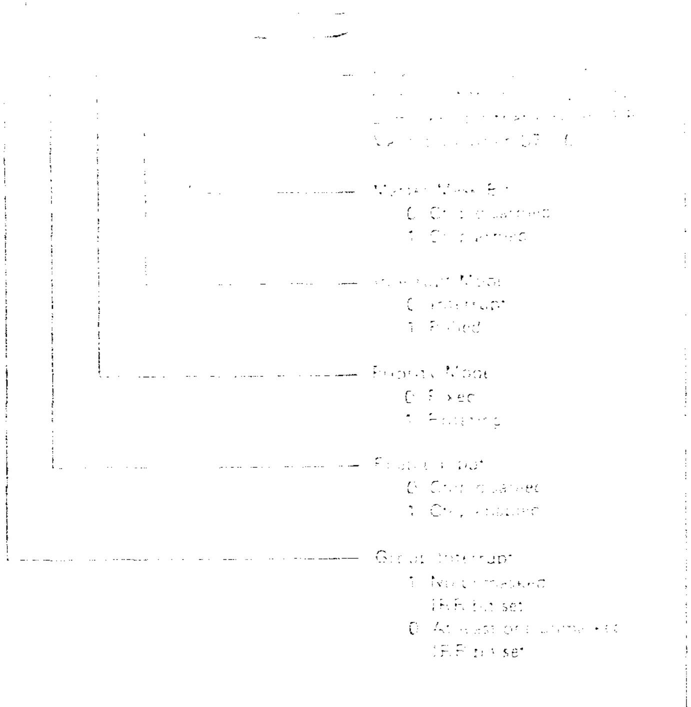


Figure 1. Status Register Bit Assignments.

MOS-025

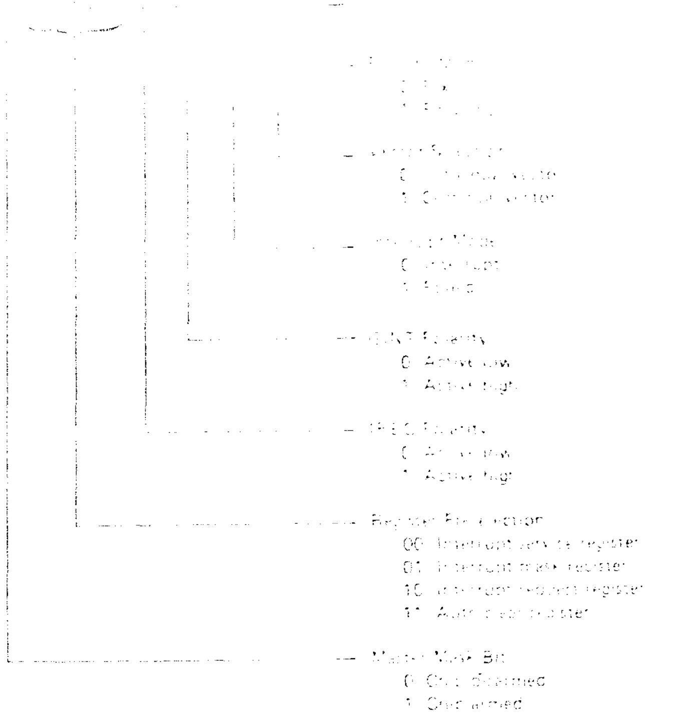


Figure 2. Mode Register Bit Assignments.

MOS-026

FUNCTIONAL DESCRIPTION

Interrupts are used to improve system throughput and response time by eliminating heavy dependence on software polling procedures. Interrupts allow external devices to asynchronously modify the instruction sequence of a program being executed. In systems with multiple interrupts, vectoring can further improve performance by allowing direct identification of the interrupting device and its associated service routine. The Am9519A Universal Interrupt Controller contains, on one chip, all of the circuitry necessary to detect, prioritize and manage eight vectored interrupts. It includes many options and operating modes that permit the design of sophisticated interrupt systems.

Reset

The reset function is accomplished by software command or automatically during power-up. The reset command may be issued by the CPU at any time. Internal power up circuitry is triggered when VCC reaches a predetermined threshold, causing a brief internal reset pulse. In both cases, the resulting internal state of the machine is that all registers are cleared except the Mask register which is set. Thus no Group Interrupt will be generated and no interrupt requests will be recognized. The response memory and Byte Count registers are not affected by reset. Their contents after power-up are unpredictable and must be established by the host CPU during initialization.

Operating Sequence

A brief description of a typical sequence of events in an operating interrupt system will illustrate the general interactions among the host CPU, the interrupt controller and the interrupting peripheral.

1. The Am9519A controller is initialized by the CPU in order to customize its configuration and operation for the application at hand. Both the controller and the CPU are then enabled to accept interrupts.

2. One (or more) of the interrupt request inputs to the controller becomes active indicating that peripheral equipment is asking for service. The controller asynchronously accepts and latches the request(s).

3. If the request is masked, no further action takes place. If the request is not masked, a Group Interrupt output is generated by the controller.

4. The GINT signal is recognized by the CPU which normally will complete the execution of the current instruction, insert an interrupt acknowledge sequence into its instruction execution stream, and disable its internal interrupt structure. The controller expects to receive one or more $\overline{\text{IACK}}$ signals from the CPU during the acknowledge sequence.

5. When the controller receives the $\overline{\text{IACK}}$ signal, it brings $\overline{\text{PAUSE}}$ low and selects the highest priority unmasked pending request. When selection is complete, the $\overline{\text{RIP}}$ output is brought low and the first byte in the response memory associated with the selected request is output on the data bus. $\overline{\text{PAUSE}}$ stays low until RIP goes low. $\overline{\text{RIP}}$ stays low until the last byte of the response has been transferred.

6. During the acknowledge sequence, the IRR bit corresponding to the selected request is automatically cleared, and the corresponding ISR bit is set. When the ISR bit is set, the Group Interrupt output is disabled until a higher priority request arrives or the ISR bit is cleared. The ISR bit will be cleared by either hardware or software.

7. If a higher priority request arrives while the current request is being serviced, GINT will be output by the controller, but will be recognized and acknowledged only if the CPU has its interrupt input enabled. If acknowledged, the corresponding higher priority ISR bit will be set and the requests nested.

Data Bus Transfers

Figure 3 shows the control input combinations for information transfers between the Am9519 and the data bus. The following conventions are adhered to: RD and WR active are mutually exclusive; RD, WR and C/$\overline{D}$ have no meaning unless $\overline{CS}$ is low; active $\overline{IACK}$ pulses occur only when $\overline{CS}$ is high.

For reading, the Status register is selected directly by the C/$\overline{D}$ control input. Other internal registers are read by preselecting the desired register with mode bits 5 and 6, and then executing a data read. The response memory can be read only with $\overline{IACK}$ pulses. For writing, the Command register is selected directly by the C/$\overline{D}$ control input. The Mask and Auto Clear registers are loaded following specific commands to that effect. To load each level of the response memory, the response preselect command is issued to select the desired level. An appropriate number of data write operations are then executed to load that level.

CONTROL INPUT					DATA BUS OPERATION
CS	C/D	RD	WR	IACK	
0	0	0	1	1	Transfer contents of preselected data register to data bus
0	0	1	0	1	Transfer contents of data bus to preselected data register
0	1	0	1	1	Transfer contents of status register to data bus
0	1	1	0	1	Transfer contents of data bus to command register
1	X	X	X	0	Transfer contents of selected response memory location to data bus
1	X	X	X	1	No information transferred

Figure 3. Summary of Data Bus Transfers.

The Pause output may be used by the host CPU to ensure that proper timing relationships are maintained with the Am9519A when $\overline{IACK}$ is active. The $\overline{IACK}$ pulse width required depends on several variables, including: operating temperature, internal logic delays, number of interrupt controllers chained together, and the priority level of the interrupt being acknowledged. When delays in these variables combine to delay selection of a request following the falling edge of the first $\overline{IACK}$, the Pause output may be used to extend the $\overline{IACK}$ pulse, if necessary. Pause will remain low until a request has been selected, as indicated by the falling edge of $\overline{RIP}$. Typically, the internal interrupt selection process is quite fast, especially for systems with a single Am9519A and Pause will consequently remain low for only a very brief interval and will not cause extension of the IACK timing.

Operating Options

The Mode register specifies the various combinations of operating options that may be selected by the CPU. It is cleared by power-up or by a reset command.

Mode bit 0 specifies the rotating/fixed priority mode (see Figure 2). In the fixed mode, priority is assigned to the request inputs based upon their physical location at the pins, with IREQ0 the highest and IREQ7 the lowest. In the rotating mode, the relative priority is rotated after each service and the most recently serviced request is assigned the lowest priority. In the fixed mode, a lower priority request might never receive service if enough higher priority requests are active. In the rotating mode, any request will receive service within a maximum of seven other service cycles no matter what pattern the request inputs follow.

Mode bit 1 selects the individual/common vector option. Individual vectoring provides a unique location in the response memory for each interrupt request. The common vector option always supplies the response associated with IREQ0 no matter which request is being acknowledged.

Mode bit 2 specifies interrupt or polled operation. In the polled mode the Group Interrupt output is disabled. The CPU may read the Status register to determine if a request is pending. Since $\overline{IACK}$ pulses are not normally supplied in polled mode, the IRR bit is not automatically cleared but may be cleared by command. With no $\overline{IACK}$ input the ISR and the response memory are not used. An Am9519A in the polled mode has EI connected to EO so that in multichip interrupt systems the polled chip is functionally removed from the priority hierarchy.

Mode bit 3 specifies the sense of the GINT output. When active high polarity is selected the output is a two-state configuration. For active low polarity, the output is open drain and requires an external pull-up resistor to provide the high logic level. The open drain output allows wired-or configurations with other similar output signals.

Mode bit 4 specifies the sense of the IREQ inputs. When active low polarity is selected, the IRR responds to falling edges on the request inputs. When active high is selected, the IRR responds to rising edges.

Mode bits 5 and 6 specify the register that will be read on subsequent data read operations (C/$\overline{D}$ = 0, $\overline{RD}$ = 0). This preselection remains valid until changed by a reset or a command.

Mode bit 7 is the master mask bit that disables all request inputs. It is used to disable all interrupts without modifying the IMR so that the previous IMR contents are valid when interrupts are re-enabled. When the master mask bit is low, it causes the EO line to remain disabled (low). Thus, for multiple-chip interrupt systems, one master mask bit can disable the whole interrupt structure. Alternatively, portions of the structure may be disabled. The state of the master mask bit is available as bit S3 of the Status register.

Programming

After reset, the Am9519A must be initialized by the CPU in order to perform useful work. At a minimum, the master mask bit and at least one of the IMR bits should be enabled. If vectoring is to be used, the response memory must be loaded; if not, the mode must be changed to a non-vectored configuration. Normally, the first step will be to modify the Mode register and the Auto clear register in order to establish the configuration desired for the application. Then the response memory and byte count will be loaded for those request levels that will be in use. Finally, the master mask bit and at least portions of the IMR will be enabled to allow interrupt processing to proceed.

[illegible]

[illegible] in Figure 4. An "X" entry in the table indicates a "don't care" entry. All commands are entered by directly loading the Command register as shown in Figure 3 (C/D = 1, WR

Figure 5 shows the coding assignments for the Byte Count registers. A detailed description of each command is contained in the Am9519A Application Note AMPUB-071.

BY1	BY0	[illegible]
0	0	1
0	1	2
1	0	3
1	1	4

Figure 5. Byte Count Coding.

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Reset
0	0	0	1	0	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0 - 4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Preselect IMR for subsequent loading from data bus
1	1	0	0	X	X	X	X	Preselect Auto Clear register for subsequent loading from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count register and preselect response memory level specified by L2, L1, L0 for subsequent loading from data bus

Figure 4. Am9519A Command Summary.

MAXIMUM RATINGS

Storage Temperature	-65°C to +150°C
Ambient Temperature Under Bias	-55°C to +125°C
VCC with Respect to VSS	-0.5V to +7.0V
All Signal Voltages with Respect to VSS	-0.5V to +7.0V
Power Dissipation (Package Limitation)	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VCC	VSS
Am9519ADC/CC Am9519A-1DC	0°C ≤ T_A ≤ +70°C	+5.0V ±5%	0V
Am9519ADM	-55°C ≤ T_A ≤ +125°C	+5.0V ±10%	0V

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

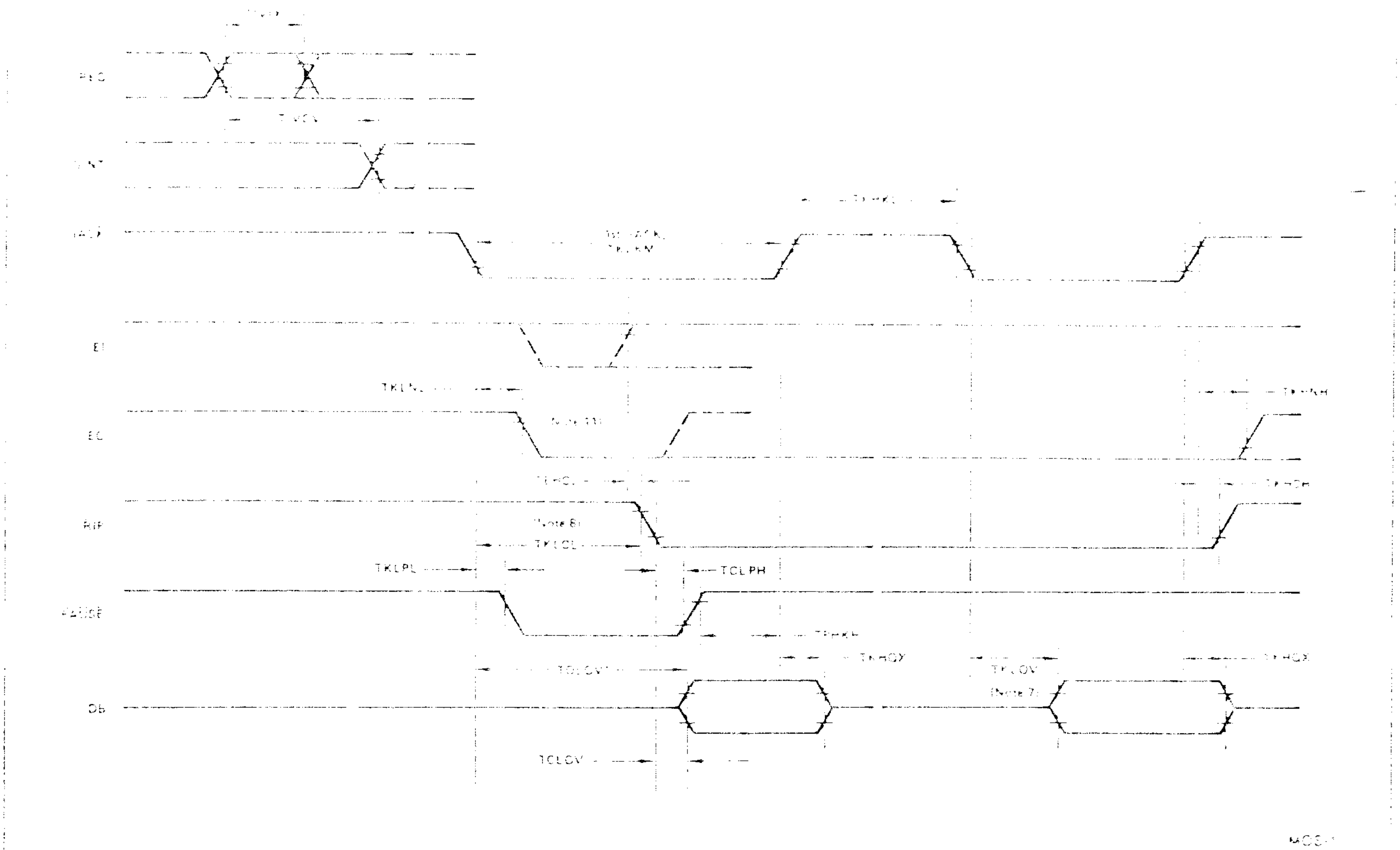
Parameter	Description	Test Conditions		Min.	Typ.	Max.	Unit
VOH	Output High Voltage (Note 12)	IOH = -200µA		2.4			Volts
		IOH = -100µA (EO only)		2.4			
VOL	Output Low Voltage	IOL = 3.2mA				0.4	Volts
		IOL = 1.0mA (EO only)				0.4	
VIH	Input High Voltage			2.0		VCC	Volts
VIL	Input Low Voltage			-0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VIN ≤ VCC	EI Input	-60		10	µA
			Other Inputs	-10		10	
IOZ	Output Leakage Current	VSS ≤ VOUT ≤ VCC, Output off		-10		10	µA
ICC	VCC Supply Current	T_A = +25°C			80	125	mA
		T_A = 0°C			100	145	
CO	Output Capacitance	fc = 1.0MHz				15	pF
CI	Input Capacitance	T_A = 25°C				10	
CIO	I/O Capacitance	All pins at 0V				20	

Parameters	Description	Am9519A Min	Am9519A Max	Am9519A-1 Min	Am9519A-1 Max	Units
TAVRL	C/$\overline{\text{D}}$ Valid and $\overline{\text{CS}}$ LOW to Read LOW	0		0		ns
TAVWL	C/$\overline{\text{D}}$ Valid and $\overline{\text{CS}}$ LOW to Write LOW	0		0		ns
TCLPH	$\overline{\text{RIP}}$ LOW to $\overline{\text{PAUSE}}$ HIGH (Note 6)	75	300	75	300	ns
TCLQV	$\overline{\text{RIP}}$ LOW to Data Out Valid (Note 7)		60		40	ns
TDVWH	Data In Valid to Write HIGH	150		200		ns
TEHCL	Enable In HIGH to $\overline{\text{RIP}}$ LOW (Notes 8, 9)	30	300	30	300	ns
TIVGV	Interrupt Request Valid to Group Interrupt Valid		800		650	ns
TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)	250		250		ns
TKHCH	$\overline{\text{IACK}}$ HIGH to $\overline{\text{RIP}}$ HIGH (Note 8)		400		350	ns
TKHKL	$\overline{\text{IACK}}$ HIGH to $\overline{\text{IACK}}$ LOW ($\overline{\text{IACK}}$ Recovery)	500		500		ns
TKHNH	$\overline{\text{IACK}}$ HIGH to EO HIGH (Notes 10, 11)		800		700	ns
TKHQX	$\overline{\text{IACK}}$ HIGH to Data Out Invalid	20	200	20	100	ns
TKLCL	$\overline{\text{IACK}}$ LOW to $\overline{\text{RIP}}$ LOW (Note 8)	75	600	75	450	ns
TKLKH	$\overline{\text{IACK}}$ LOW to $\overline{\text{IACK}}$ HIGH (1st IACK)	975		800		ns
TKLNL	$\overline{\text{IACK}}$ LOW to EO LOW (Notes 10, 11)		125		100	ns
TKLPL	$\overline{\text{IACK}}$ LOW to $\overline{\text{PAUSE}}$ LOW	25	175	25	125	ns
TKLQV	$\overline{\text{IACK}}$ LOW to Data Out Valid (Note 7)	25	300	25	200	ns
TKLQV1	1st $\overline{\text{IACK}}$ LOW to Data Out Valid	75	650	75	480	ns
TPHKH	$\overline{\text{PAUSE}}$ HIGH to $\overline{\text{IACK}}$ HIGH	0		0		ns
TRHAX	Read HIGH to C/$\overline{\text{D}}$ and $\overline{\text{CS}}$ Don't Care	0		0		ns
TRHQX	Read HIGH to Data Out Invalid	20	200	20	100	ns
TRLQV	Read LOW to Data Out Valid		300		200	ns
TRLQX	Read LOW to Data Out Unknown	50		50		ns
TRLRH	Read LOW to Read HIGH ($\overline{\text{RD}}$ Pulse Duration)	300		250		ns
TWHAX	Write HIGH to C/$\overline{\text{D}}$ and $\overline{\text{CS}}$ Don't Care	25		25		ns
TWHDX	Write HIGH to Data In Don't Care	25		25		ns
TWHRW	Write HIGH to Read or Write LOW (Write Recovery)	600		400		ns
TWLWH	Write LOW to Write HIGH ($\overline{\text{WR}}$ Pulse Duration)	300		250		ns

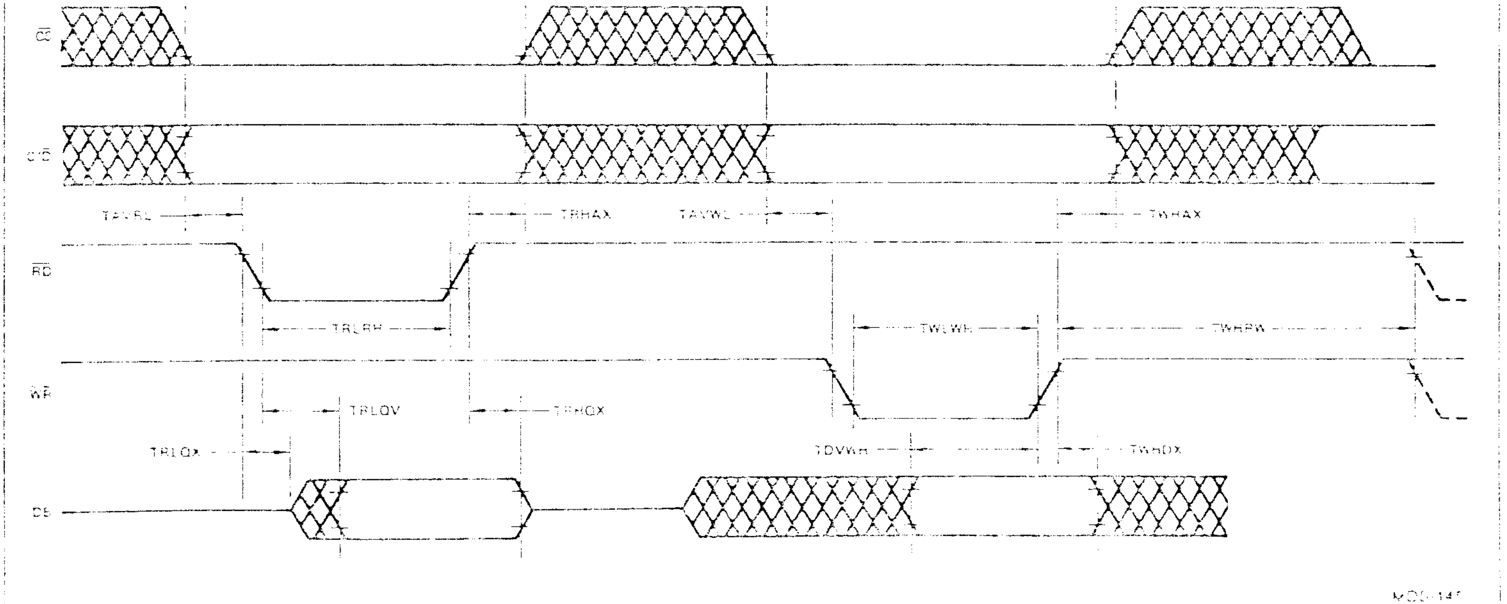
NOTES:

1. Typical values for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Test conditions assume transition times of 20ns or less, timing reference levels of 0.8V and 2.0V and output loading of one TTL gate plus 100pF, unless otherwise noted.
3. Transition abbreviations used for the switching parameter symbols include: H = High, L = Low, V = Valid, X = unknown or don't care, Z = high impedance.
4. Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address ($\overline{\text{CS}}$ and C/$\overline{\text{D}}$), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = $\overline{\text{RIP}}$.
5. Switching parameters are listed in alphabetical order.
6. During the first $\overline{\text{IACK}}$ pulse, $\overline{\text{PAUSE}}$ will be low long enough to allow for priority resolution and will not go high until after RIP goes low (TCLPH).
7. TKLQV applies only to second, third and fourth $\overline{\text{IACK}}$ pulses while $\overline{\text{RIP}}$ is low. During the first $\overline{\text{IACK}}$ pulse, Data Out will be valid following the falling edge of $\overline{\text{RIP}}$ (TCLQV).
8. $\overline{\text{RIP}}$ is pulled low to indicate that an interrupt request has been selected. $\overline{\text{RIP}}$ cannot be pulled low until EI is high following an internal delay. TKLCL will govern the falling edge of $\overline{\text{RIP}}$ when EI is always high or is high early in the acknowledge cycle. TEHCL will govern when EI goes high later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain. $\overline{\text{RIP}}$ remains low until after the rising edge of the $\overline{\text{IACK}}$ pulse that transfers the last response byte for the selected IREQ.
9. Test conditions for the EI line assume timing reference levels of 0.8V and 2.0V with transition times of 10ns or less.
10. Test conditions for the EO line assume output loading of two LS TTL gates plus 30pF and timing reference levels of 0.8V and 2.0V. Since EO normally only drives EI of another Am9519A, higher speed operation can be specified with this more realistic test condition.
11. The arrival of $\overline{\text{IACK}}$ will cause EO to go low, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return high when EI is high. If a pending request is selected, EO will stay low until after the last $\overline{\text{IACK}}$ pulse for that interrupt is complete and $\overline{\text{RIP}}$ goes high.
12. VOH specifications do not apply to $\overline{\text{RIP}}$ or to $\overline{\text{GINT}}$ when active-low. These outputs are open-drain and VOH levels will be determined by external circuitry.

SWITCHING WAVEFORMS



Interrupt Operations



Data Bus Transfers

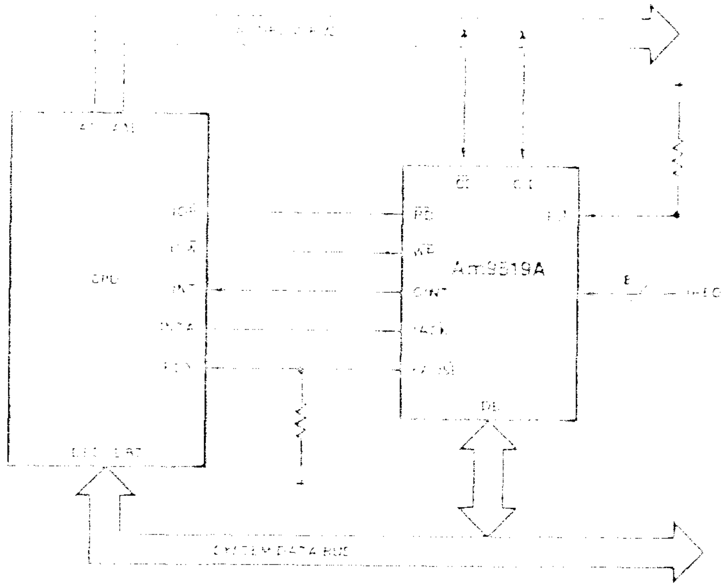


Figure 6. Base Interrupt System Configuration.

MOS 146

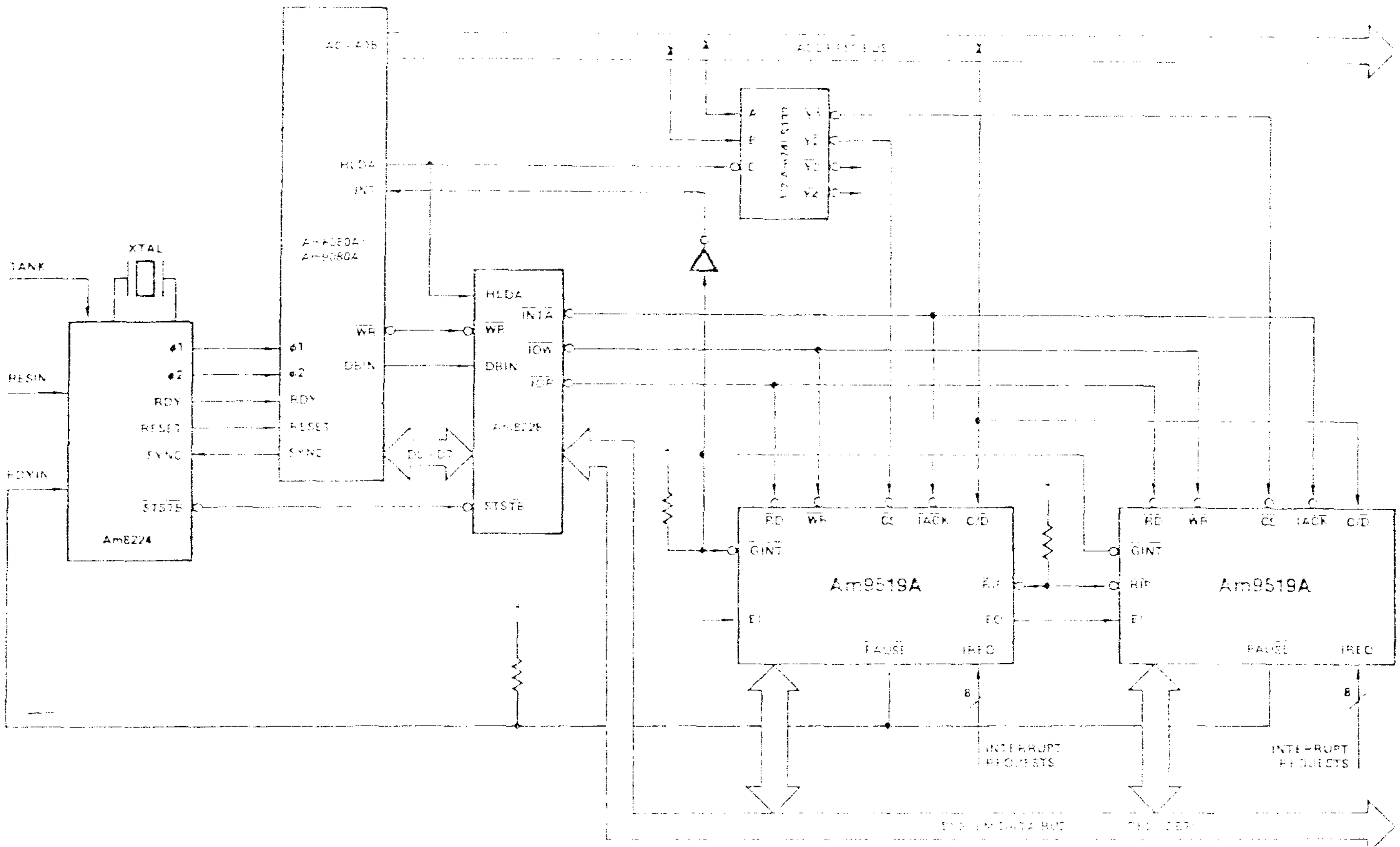
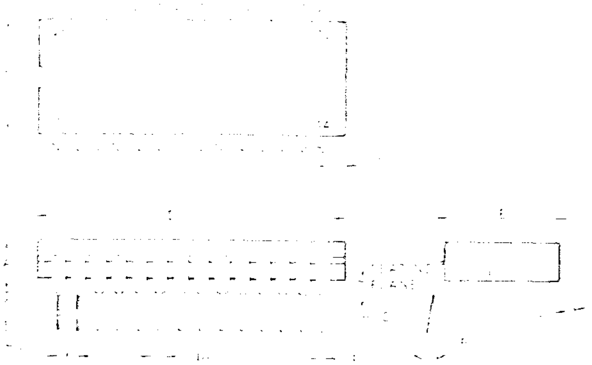


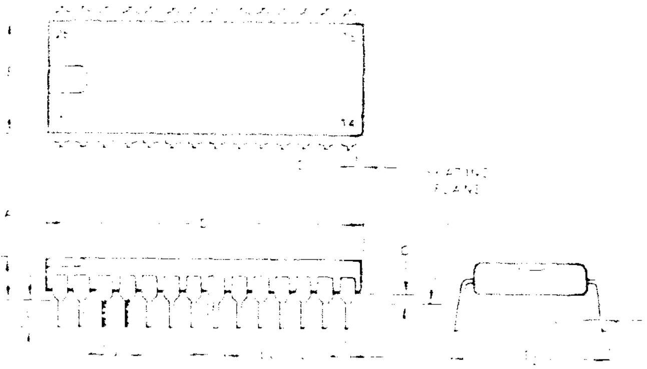
Figure 7. Expanded Interrupt System Configuration.

MOS 147

28-Pin Cerdip



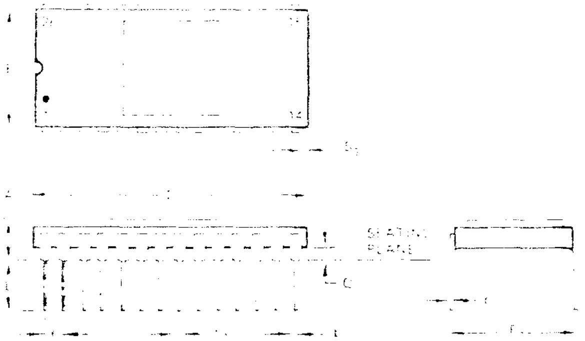
28-Pin Molded DIP



Reference Symbol	Inches	
	Min	Max
A	.150	.225
b	.015	.020
b₁	.045	.065
c	.009	.012
D	1.440	1.490
E	.510	.545
E₁	.600	.620
e	.090	.110
L	.125	.150
Q	.015	.060
S₁	.010	

Reference Symbol	Inches	
	Min	Max
A	.150	.200
b	.015	.020
b₁	.055	.065
c	.009	.011
D	1.450	1.440
E	.530	.550
E₁	.585	.700
e	.090	.110
L	.125	.150
Q	.015	.060
S₁	.040	.070

28-Pin Side-Brazed Ceramic



Reference Symbol	Inches	
	Min	Max
A	.100	.200
b	.015	.022
b₁	.030	.060
c	.008	.013
D	1.380	1.420
E	.560	.600
E₁	.580	.620
e	.090	.110
L	.120	.160
Q	.020	.060
S₁*	.005	
α	0	0

ADVANCED
MICRO
DEVICES, INC.
901 Thompson Place
P.O. Box 453
Sunnyvale,
California 94086
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL FREE
(800) 538-8450

12-80 MOS-147

Advanced Micro Devices cannot assume responsibility for use of any circuitry described other than circuitry embodied in an Advanced Micro Devices product.