μPD77230

Advanced Signal Processor

NEC Electronics Inc.

**ADVANCED
PRODUCT
INFORMATION
USER'S MANUAL**

March 1986

μPD77230

Advanced Signal Processor

The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Electronics Inc., at its own discretion, may withdraw the device prior to production.

The information in this document is subject to change without notice. NEC Electronics Inc. assumes no responsibility for any errors or omissions that may appear in this document. Devices sold by NEC Electronics Inc. are covered by the warranty and patent indemnification provisions appearing in NEC Electronics Inc. Terms and Conditions of Sale only. NEC Electronics Inc. makes no warranty, express, statutory, implied, or by description, regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. NEC Electronics Inc. makes no warranty of merchantability or fitness for any purpose. NEC Electronics Inc. makes no commitment to update or to keep current the information contained in this document. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics Inc.

uPD77230 User's Manual

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 General

The uPD77230R Advanced Signal Processor (ASP) is a digital signal processor designed for both high speed and high accuracy. It can execute arithmetic operations with 32-bit floating-point data (8-bit exponent, 24-bit mantissa) or with 24-bit fixed-point data at a rate as fast as 150 ns/instruction. Its internal circuitry consists of a floating-point multiplier (32 bits x 32 bits), a 55-bit floating-point ALU, 2K x 32 bits of instruction ROM, 1K x 32 bits of data ROM, and a pair of separately addressable data RAM blocks, each one 512 x 32 bits. Two operation modes are available: master, and slave, which are externally selected. With master mode specified, up to 8K x 32 bits of external memory can be used, with up to 4K of that space usable for instruction memory. In slave mode, the uPD77230R has a 16-bit I/O port under control of a host processor, and can also support up to 8K words of external memory.

1.2 Features

1) Arithmetic operation using 32-bit floating-point or 24-bit fixed-point data.

   - 32-bit floating-point multiplier (8-bit exponent, 24 bit mantissa input, and 8-bit exponent, 47-bit mantissa output).
   - 55-bit floating-point ALU (can also perform 47-bit fixed point ALU functions).
   - 8 Working Registers, each 55 bits long.
   - 47-bit bidirectional barrel shifter.

2) High-speed operation and effective data transfer.

   - Instruction cycle of 150 ns max.
   - Use of three-stage pipelining process.
   - Dedicated data busses for internal RAM, multiplier, and ALU.

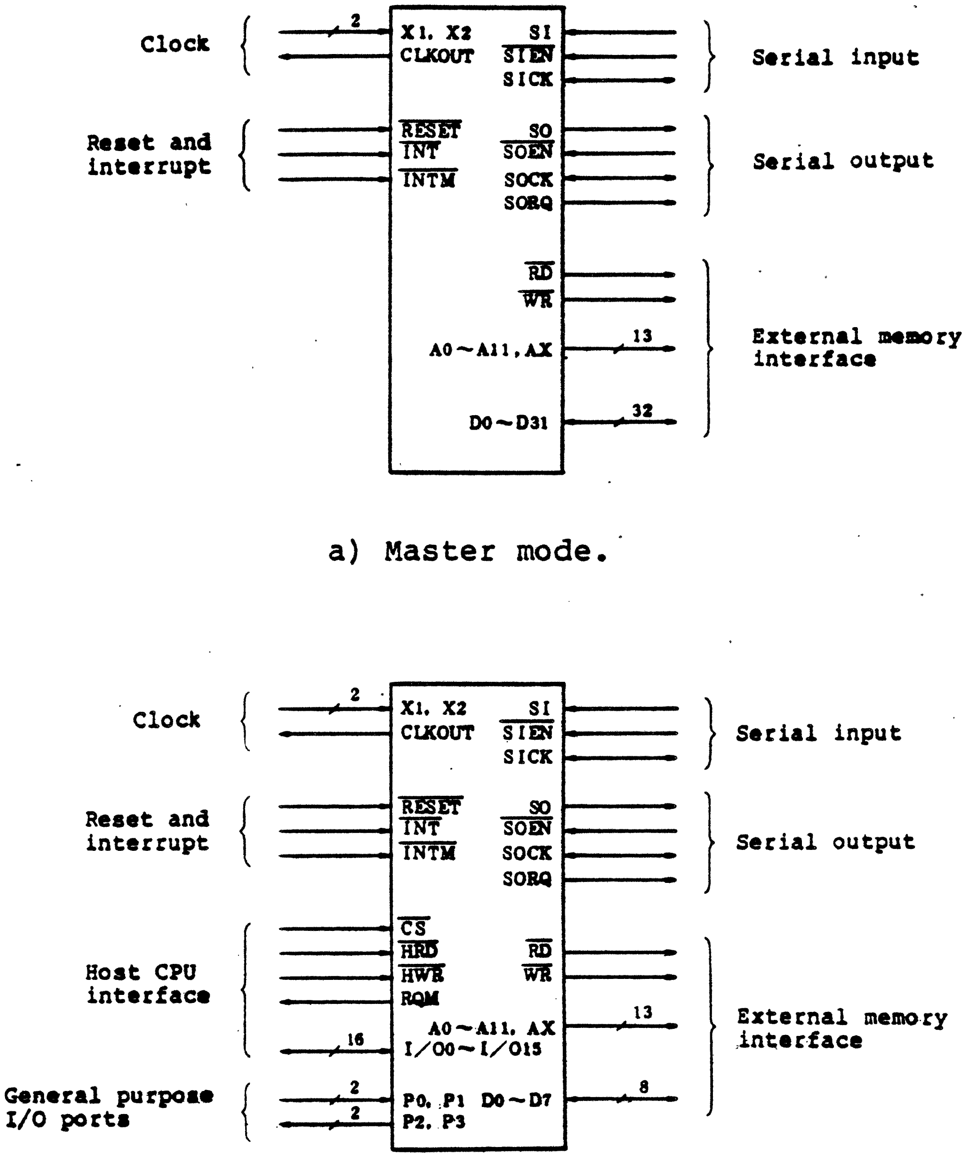
3) Architecture ideal for digital signal processing.

   - Two separately addressable data RAM blocks.
   - Data RAM pointers can perform ring count (modulo) operation.
   - Data ROM pointer can be incremented by 2**N, as well as ordinary increment and decrement.
   - Both inputs to the multiplier can be loaded simultaneously with an ALU operation (which has access to the previous multiplier result) and pointer modification.

4) External interface may be used in a variety of system configurations.

   * Master/slave mode selectable.

Figure 1-1. Master and slave mode functional pin groups.



a) Master mode.

b) Slave mode.

1.3 Functional Blocks

Figures 1-2 and 1-3 are the block diagrams of the uPD77230R. The internal circuitry can be broadly divided into the following functional blocks:

* Arithmetic operation processing section, including multiplier and ALU.
* Memory section, consisting of a data ROM and two separately addressable data RAM blocks.
* Instruction ROM.
* Parallel and serial I/O interface.

Table 1-1 lists the function of each block.

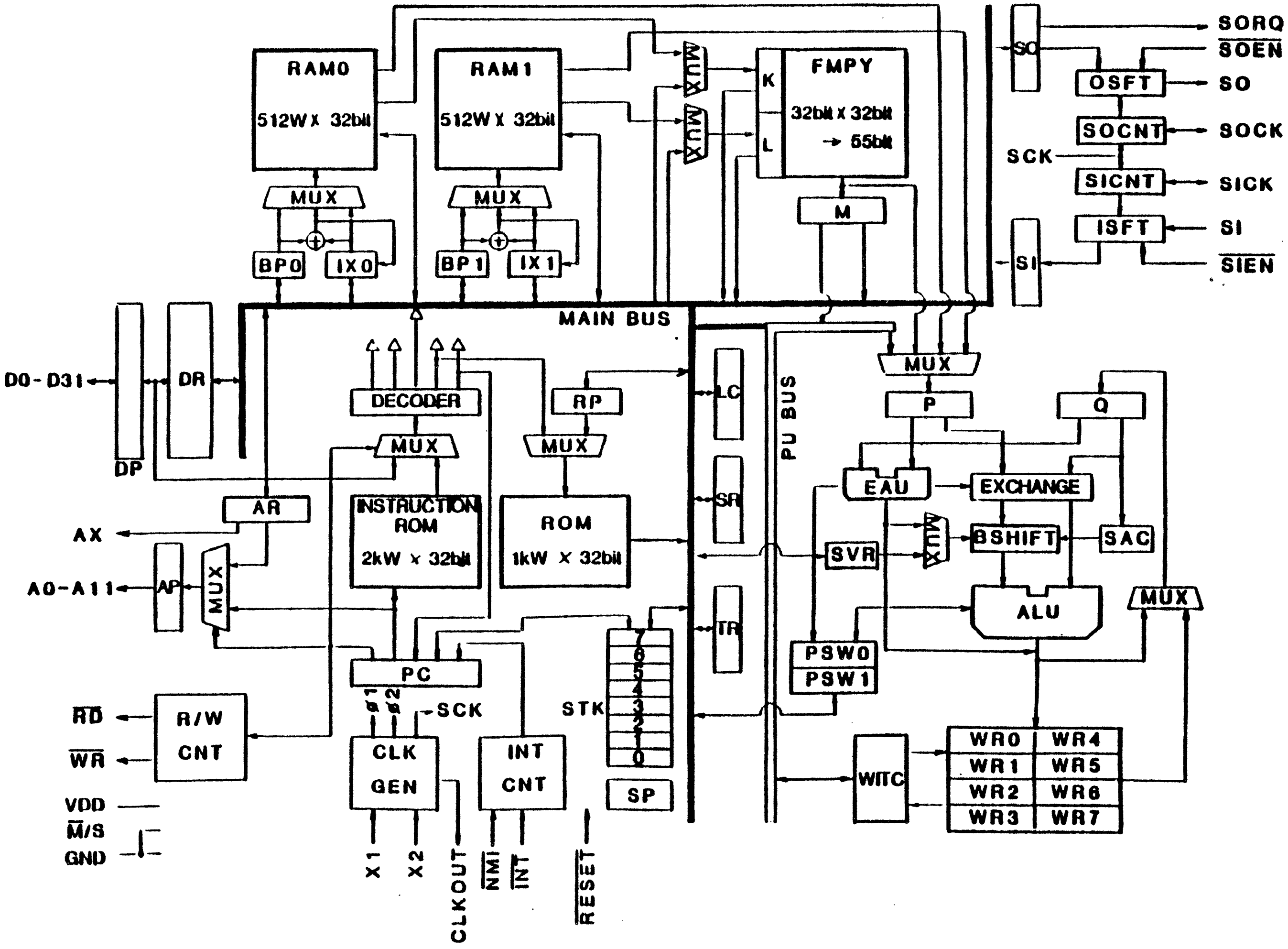


Figure 1-2. Master mode block diagram.

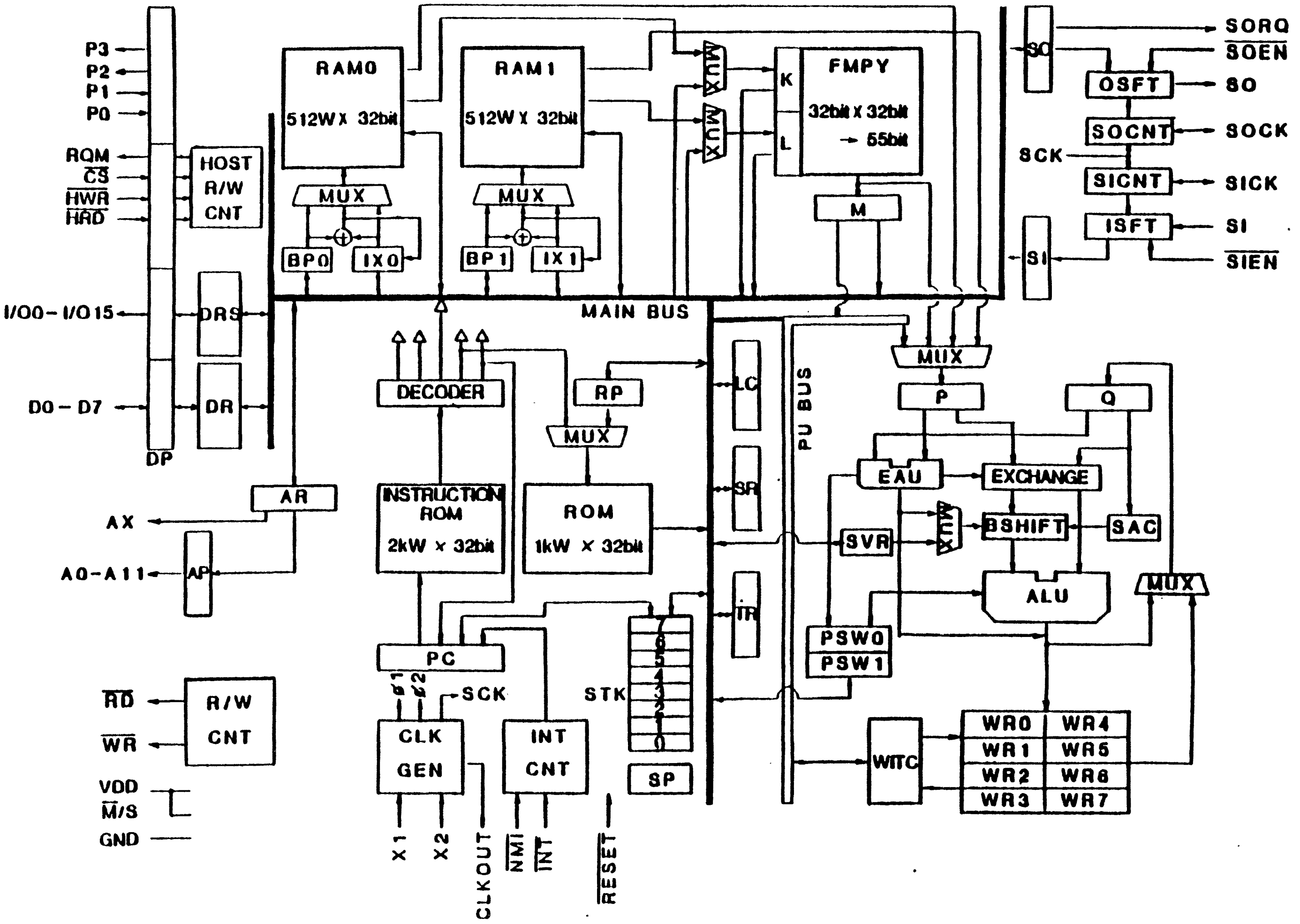


Figure 1-3. Slave mode block diagram.

Table 1-1. Functional blocks of the uPD77230R.

Multiplier and associated circuits

Abbreviation	Name	Function
FMPY	Floating-point multiplier	Multiplier for 32-bit floating-point data (8-bit exponent, 24-bit mantissa) 32 bits X 32 bits -> 55 bits
K	K register	Retains data input to FMPY (32 bits)
L	L register	Retains data input to FMPY (32 bits)
M	M register	Retains multiplication result of FMPY (55 bits)

ALU and associated circuits

Abbreviation	Name	Function
ALU	Arithmetic and logical unit	Performs arithmetic and logical operations on 47-bit mantissa data
EAU	Exponent arithmetic and logical unit	Performs arithmetic and logical operations on 8-bit exponent data
P	P register	Retains data input to ALU and EAU (55 bits)
Q	Q register	Retains data input to ALU and EAU (55 bits)
SAC	Shift and count circuit	Detects shift value of mantissa value in Q
BSHIFT	Barrel shifter	Barrel shifter for mantissa in Q and P registers
SVR	Shift value register	Sets shift value
WR0 to WR7	Working registers 0 to 7	Retain arithmetic operation results of ALU and EAU
PSW0	Processor status word 0	Indicates arithmetic operation result status of ALU and EAU
PSW1	Processor status word 1	Indicates arithmetic operation result status of ALU and EAU
WRTC	Working register transfer control	Control format of transfer to/from working registers

Table 1-1 cont'd.

Data memories and associated circuits

Abbreviation	Name	Function
DATA ROM	Data ROM	Stores fixed data (1K words X 32 bits)
RP	ROM pointer	Indicates current data ROM address
DATA RAM 0	Data RAM 0	Stores data (512 words X 32 bits)
BASE 0	Base pointer 0	Indicates current base address of data RAM 0
INDEX 0	Index register 0	Indicates current index address of data RAM 0
DATA RAM 1	Data RAM 1	Stores data (512 words X 32 bits)
BASE 1	Base pointer 1	Indicates current base address of data RAM 1
INDEX 1	Index register 1	Indicates current index address of data RAM 1
ADD	Adder	Adder for base address and index address

Instruction ROM and associated circuits

Abbreviation	Name	Function
INSTRUCTION ROM	Instruction ROM	Stores instructions (2K words X 32 bits)
PC	Program counter	Indicates current address of instruction ROM (13 bits)
STACK	Stack	Eight-level 13-bit stack
SP	Stack pointer	Indicates current address of stack
IR	Instruction register	Retains data output from instruction ROM
DECODE	Instruction decoder	Decodes instructions

Table 1-1 cont'd.

Parallel interface

Abbreviation	Name	Function
DP	Data port	In master mode, a 32-bit parallel data bus for external memory
		In slave mode, consists of an 8-bit parallel data bus for external memory, a 16-bit parallel data bus for host I/O interfacing, read/write control signals from the host, and a general-purpose parallel I/O port
AP	Address port	In master mode, a 12-bit address bus for external instruction and data memory
		In slave mode, a 12-bit address bus for external data memory
DR	Data register	In master mode, a register for interfacing the DP with the internal data bus
		In slave mode, a register for interfacing the DP (8-bit parallel data bus for external data memory) with the internal data bus
DRS	Data register for slave	In slave mode, a register for interfacing the DP (16-bit parallel data bus for host I/O) with the internal data bus
AR	Address register	Indicates current address of external data memory
HOST R/W CNT	Host CPU read/write control circuit	In slave mode, controls interfacing the uPD77230R with a host CPU
R/W CNT	Read/write control circuit	Controls read/write of external memory

Table 1-1 cont'd.

Serial input/output interface

Abbreviation	Name	Function
SO	Serial output data register	Retains serial output data
OSFT	Output shift register	Serially outputs SO data
SOCNT	Serial output control circuit	Controls serial output
SI	Serial input data register	Retains serial input data
ISFT	Input shift register	Inputs serial data
SICNT	Serial input control circuit	Controls serial input

Other control circuits

Abbreviation	Name	Function
CLK GEN	Clock generator	Generates internal system clock and serial I/O clock
INT CNT	Interrupt controller	Controls external interrupts
TR	Temporary register	General-purpose temporary register (32 bits)
LC	Loop counter	Controls the number of times a program will loop
SR	Status register	Specifies or indicates current mode of operation

1.4 Data Format

The uPD77230R is capable of processing both fixed-point and floating-point data.

1.4.1 Floating-point data format

The data on the 32-bit internal data bus, and associated 32-bit registers, consists of an 8-bit exponent in the highest 8 bits, and a 24 bit mantissa in the lower 24 bits. The 55-bit processing unit bus and associated 55-bit registers (M, P, Q, WR0-WR7) contain data that consists of an 8-bit exponent in the highest 8 bits, and a 47-bit mantissa in the lower 47 bits.

Both the exponent and mantissa are represented in two's complement notation, with their most significant bit being a sign bit.

55 bit representation:

```
Exponent (8 bits)                 Mantissa (47 bits)
-----------------------------------------------------------------
| Sign + 7 bits  | Sign + 46 bits                               |
-----------------------------------------------------------------
     Both exponent and mantissa are two's complement
```

32 bit representation:

```
Exponent (8 bits)                 Mantissa (24 bits)
-----------------------------------------------------------------
| Sign + 7 bits  | Sign + 23 bits                               |
-----------------------------------------------------------------
     Both exponent and mantissa are two's complement
```

Figure 1-4. Floating point data formats.

Table 1-2. Internal 32-bit floating point data format.

	Binary representation		Hexadecimal representation		Decimal representation
	Exponent	Mantissa	Exponent	Mantissa	
Max. positive value	01111111	0111......1111	7F	7FFFFF	$(1.0-2^{-23})x2^{127} \approx 1.7x10^{38}$
:	01111111	0111......1110	7F	7FFFFE	$(1.0-2^{-22})x2^{127}$
:	:		:		
:	00000110	0100......0000	60	400000	$(1.0-2^{-1})x2^{6} = 32$
:	:		:		
Min. positive value	10000000	0000......0001	80	000001	$(2^{-23})x2^{-128} \approx 3.5x10^{-46}$
Zero	10000000	0000......0000	80	000000	0.0
Max. negative value	10000000	1111......1111	80	FFFFFF	$-(2^{-23})x2^{-128} \approx -3.5x10^{-46}$
:	:		:		
:	00000110	1100......0000	60	C00000	$(-1.0+2^{-1})x2^{6} = -32$
:	:		:		
:	01111111	1000......0001	7F	800001	$(-1.0+2^{-23})x2^{127}$
Min. negative value	01111111	1000......0000	7F	800000	$-1.0x2^{127} \approx -1.7x10^{38}$

Table 1-3. Internal 55-bit floating point data format.

	Binary representation		Hexadecimal representation		Decimal representation
	Exponent	Mantissa	Exponent	Mantissa	
Max. positive value	01111111	0111......1111	7F	7FFFFFFFFFFE	$(1.0-2^{-46})x2^{127} \approx 1.7x10^{38}$
:	01111111	0111......1110	7F	7FFFFFFFFFFC	$(1.0-2^{-45})x2^{127}$
:		: :		: :	
:	00000110	0100......0000	60	400000000000	$(1.0-2^{-1})x2^{6} = 32$
:		: :		: :	
Min. positive value	10000000	0000......0001	80	000000000002	$(2^{-46})x2^{-128} \approx 4.2x10^{-53}$
Zero	10000000	0000......0000	80	000000000000	0.0
Max. negative value	10000000	1111......1111	80	FFFFFFFFFFFE	$-(2^{-46})x2^{-128} \approx -4.2x10^{-53}$
:		: :		: :	
:	00000110	1100......0000	60	C00000000000	$(-1.0+2^{-1})x2^{6} = -32$
:		: :		: :	
:	01111111	1000......0001	7F	800000000002	$(-1.0+2^{-46})x2^{127}$
Min. negative value	01111111	1000......0000	7F	800000000000	$-1.0x2^{127} \approx -1.7x10^{38}$

1.4.2 Fixed-point data format

The data on the 32-bit internal data bus, and associated 32-bit registers, consists of a 24 bit mantissa in the lower 24 bits, with the 8 MSBs unused. The 55-bit Processing Unit bus and associated 55-bit registers (M, P, Q, WR0-WR7) contain data that consists of a 47-bit mantissa in the lower 47 bits, with the upper 8 bits unused.

The mantissa is represented in two's complement notation, with the most significant bit being a sign bit.

47 bit representation:

```
Unused (8 bits)                 Mantissa (47 bits)
-------------------------------------------------------------------
|                |  Sign + 46 bits                                |
-------------------------------------------------------------------
     Mantissa is two's complement.
```

24 bit representation:

```
Unused (8 bits)                  Mantissa (24 bits)
-------------------------------------------------------------------
|                |  Sign + 23 bits                                |
-------------------------------------------------------------------
     Mantissa is two's complement.
```

Figure 1-5. Fixed point data formats.

Table 1-4. Internal 24-bit fixed point data format.

	Binary representation		Hexadecimal representation		Decimal representation
	Exponent	Mantissa	Exponent	Mantissa	
Max. positive value	-	0111......1111	-	7FFFFF	$1.0-2^{-23} \approx 1.0$
:	-	0111......1110	-	7FFFFE	$1.0-2^{-22}$
:	:		:		
:	-	0100......0000	-	400000	$1.0-2^{-1} = 0.5$
:	:		:		
Min. positive value	-	0000......0001	-	000001	$2^{-23} \approx 1.2x10^{-7}$
Zero	-	0000......0000	-	000000	0.0
Max. negative value	-	1111......1111	-	FFFFFF	$-2^{-23} \approx -1.2x10^{-7}$
:	:		:		
:	-	1100......0000	-	C00000	$-1.0+2^{-1} = -0.5$
:	:		:		
:	-	1000......0001	-	800001	$-1.0+2^{-23}$
Min. negative value	-	1000......0000	-	800000	-1.0

Table 1-5. Internal 47-bit fixed point data format.

	Binary representation		Hexadecimal representation		Decimal representation
	Exponent	Mantissa	Exponent	Mantissa	
Max. positive value	-	0111......1111	-	7FFFFFFFFFFE	$1.0-2^{-46} \approx 1.0$
:	-	0111......1110	-	7FFFFFFFFFFC	$1.0-2^{-45}$
:		: :		: :	
:	-	0100......0000	-	400000000000	$1.0-2^{-1} = 0.5$
:		: :		: :	
Min. positive value	-	0000......0001	-	000000000002	$2^{-46} \approx 1.4x10^{-14}$
Zero	-	0000......0000	-	000000000000	0.0
Max. negative value	-	1111......1111	-	FFFFFFFFFFFE	$-2^{-46} \approx -1.4x10^{-14}$
:		: :		: :	
:	-	1100......0000	-	C00000000000	$-1.0+2^{-1} = -0.5$
:		: :		: :	
:	-	1000......0001	-	800000000002	$-1.0+2^{-46}$
Min. negative value	-	1000......0000	-	800000000000	-1.0

CHAPTER 2 PIN FUNCTIONS

The uPD77230R is housed in a 68-pin grid array package, as shown in Figure 2-1. Pin assignments are listed in Table 2-1, and functional descriptions of each pin are given in Tables 2-2 thru 2-4.

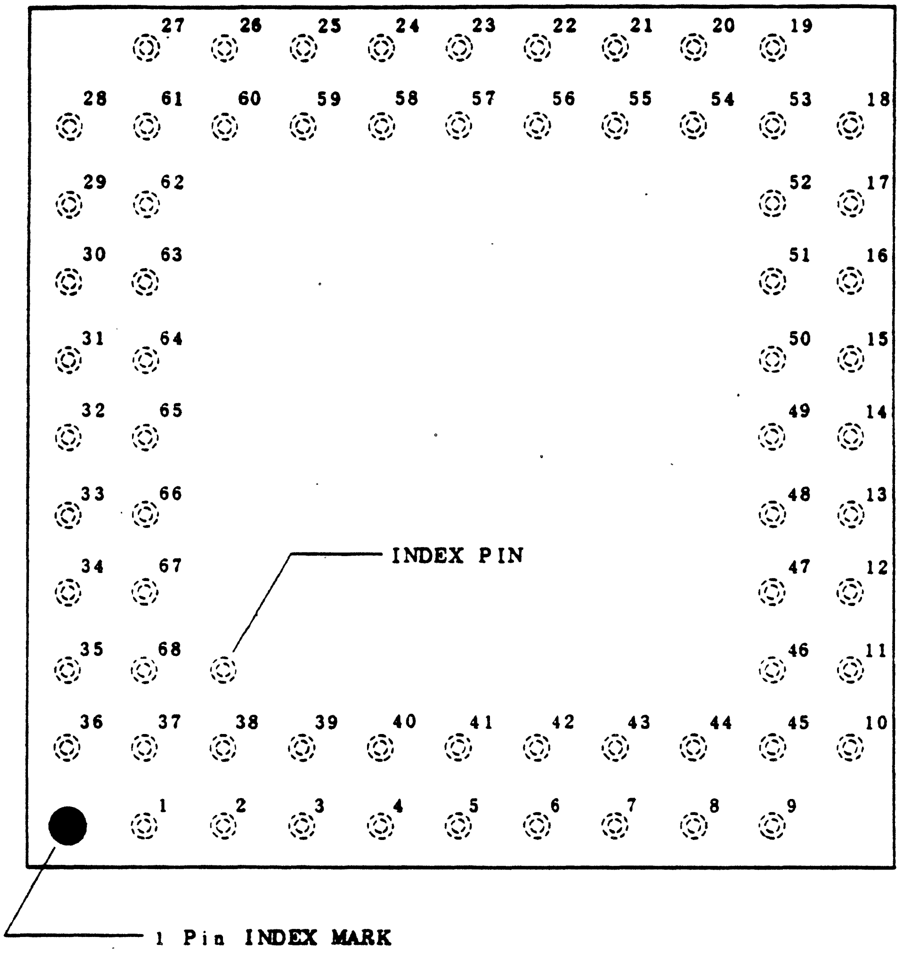


Figure 2-1. 68-pin grid array package (top view).

Table 2-1. uPD77230R pin assignments.

PIN #	MASTER	SLAVE	PIN #	MASTER	SLAVE
1	D0		35	D2	
2	A1		36	D1	
3	A3		37	A0	
4	A5		38	A2	
5	A6		39	A4	
6	A8		40	Vdd	
7	A10		41	A7	
8	Ax		42	A9	
9	WR/		43	A11	
10	RD/		44	GND	
11	SORQ		45	SO	
12	SOCK		46	SICK	
13	SOEN/		47	SIEN/	
14	INT/		48	NC (No Connect)	
15	INTM/		49	RESET/	
16	(M/)/S		50	SI	
17	CLKOUT		51	X2	
18	X1		52	Vdd	
19	D31	P3	53	D30	P2
20	D29	P1	54	D28	P0
21	D27	RQM	55	D26	CS/
22	D25	HWR/	56	GND	
23	D24	HRD/	57	D23	I/O 15
24	D22	I/O 14	58	D21	I/O 13
25	D20	I/O 12	59	D19	I/O 11
26	D18	I/O 10	60	Vdd	
27	D17	I/O 9	61	D15	I/O 7
28	D16	I/O 8	62	D13	I/O 5
29	D14	I/O 6	63	D11	I/O 3
30	D12	I/O 4	64	D9	I/O 1
31	D10	I/O 2	65	D7	
32	D8	I/O 0	66	D5	
33	D6		67	D3	
34	D4		68	GND	
			INDEX PIN		

FUNCTIONAL PIN DESCRIPTIONS

Table 2-2. Pin functions shared by master and slave modes.

* POWER SUPPLY

NAME	#	I/O	FUNCTION
Vdd	40 52 60	-	+5V power supply pins. All of these pins must be connected.
GND	44 56 68	-	Ground pins. All of these pins must be connected.

* MODE SETTING

NAME	#	I/O	FUNCTION
(M/)/S	16	Input	Selects the operation mode. The operation mode must not be switched during operation, however. 0: Master mode 1: Slave mode

* CLOCK

NAME	#	I/O	FUNCTION
X1	18	Input	A crystal oscillator is connected across these pins. External clock should be input via the X1 pin.
X2	51		
CLKOUT	17	Output	Outputs internal system clock of uPD77230R. The output signal frequency is half the oscillation frequency of the crystal connected across the X1 and X2 pins.

* RESET/INTERRUPT

NAME	#	I/O	FUNCTION
RESET/	49	Input	Inputs internal system reset signal which is active-low and must be at least 3 system clock pulses wide.
INT/	14	Input	Inputs non-maskable interrupt signal which is active-low and must be at least 3 system clock pulses wide. The interrupt signal is detected at the falling edge. The interrupt address is 10H.

Table 2-2 cont'd.

NAME	#	I/O	FUNCTION
INTM/	15	Input	Inputs maskable interrupt signal which is active-low and must be at least 3 system clock pulses wide. The interrupt signal is detected at the falling edge. The interrupt address is 100H.

* SERIAL INTERFACE

NAME	#	I/O	FUNCTION
SOCK	12	I/O	Inputs or outputs clock for serial output data. The serial output data is synchronized with the clock that is input to or output from this pin. Whether the clock is to be input from an external source or the internal clock is to be output is determined by the setting of the status register.
SORQ	11	Output	Outputs serial output request signal which is active-high. When data is ready in the serial output register, this signal becomes 1. It will become 0 after the data has been output.
SOEN/	13	Input	Enables the SO pin to output serial data. This pin is active-low.
SO	45	Output (Tri-State)	Outputs serial data synchronized with the rising edge of the SOCK pin.
SICK	46	I/O	Inputs or outputs the clock for serial input data. The serial data is internally latched at the falling edge of the clock that is input to or output from this pin. Whether the clock is to be input from an external source or the internal clock is to be output is determined by the setting of the status register.
SIEN/	47	Input	Enables the SI pin to input serial data. This pin is active-low.
SI	50	Input	Inputs serial data synchronized with the falling edge of the SICK pin.

Table 2-3. Pin functions available only in master mode.

* EXTERNAL MEMORY INTERFACE

NAME	#	I/O	FUNCTION
WR/	9	Output	Controls data write to external memory. This signal becomes 0 after the output address is valid and data is output to the data port formed by pins D0 to D31.
RD/	10	Output	Controls data read from external memory. This signal becomes 0 after the output address is valid, and data is input at the rising edge to the data port formed by pins D0 to D31.
AX	8	Output	Outputs the highest bit of the memory address. When accessing external instruction memory, the highest bit of the program counter (PC12) is output to this pin. When accessing external data memory, the highest bit of the external address register is output to this pin. 0: High-speed memory area. 1: Low-speed memory area.
A0 to A11	%%	Output	Address bus for access to external memory. When accessing external instruction memory, the lower 12 bits of the program counter are output to these pins. When accessing external data memory, the lower 12 bits of the external address register are output to these pins.
D0 to D31	%%	I/O (Tri-State)	Data bus. These pins form a 32-bit data bus for external memory (data or instruction).

%% Refer to Table 2-1 for pin numbers.

Table 2-4. Pin functions available only in slave mode.

* HOST CPU INTERFACE

NAME	#	I/O	FUNCTION
CS/	55	Input	Active-low chip select input signal. When this pin becomes 0, the host CPU may perform read/write operations on the 16-bit port formed by pins I/O 0 through I/O 15.
HWR/	22	Input	Active-low host write input signal. In conjunction with CS/, this signal allows the host CPU to write data into the DRS register via the 16-bit port formed by pins I/O 0 to I/O 15.
HRD/	23	Input	Active-low host read input signal. In conjunction with CS/, this signal allows the host CPU to read data from the DRS register via the 16-bit port formed by pins I/O 0 to I/O 15.
I/O 0 to I/O 15	%%	I/O (Tri-State)	These pins form the I/O port to the host CPU bidirectional data bus. It is used for input to or output from the DRS register under the control of the host CPU signals CS/, HWR/, and HRD/. The data transfer format can be specified in the status register as either a 16-bit or a 32-bit transfer.
RQM	21	Output	Requests the host CPU to read or write data via the host CPU data bus.

* EXTERNAL DATA MEMORY INTERFACE

NAME	#	I/O	FUNCTION
WR/	9	Output	Controls data write to external memory. This signal becomes 0 after the output address is valid and data is output to the data port formed by pins D0 to D7.
RD/	10	Output	Controls data read from external memory. This signal becomes 0 after the output address is valid, and data is input at the rising edge to the data port formed by pins D0 to D7.

Table 2-4 cont'd.

NAME	#	I/O	FUNCTION
AX	8	Output	When accessing external data memory, the highest bit of the external address register is output to this pin. 0: High-speed memory area. 1: Low-speed memory area.
A0 to All	%%	Output	Address bus for accessing external memory. When accessing external data memory, the lower 12 bits of the external address register are output to these pins.
D0 to D7	%%	I/O (Tri-State)	Data bus. These pins form an 8-bit data bus for external data memory access. Data may be transferred in one of four formats (1-, 2-, 3-, or 4-byte words), depending on the setting of the status register.

* GENERAL PURPOSE I/O PORT

NAME	#	I/O	FUNCTION
P0,P1	54, 20	Input	These pins form a general-purpose input port. The status of either of these pins may be tested by a conditional branch instruction.
P2,P3	53, 19	Output	These pins form a general-purpose output port. The data output by these pins can be set directly by an instruction and will be retained until explicitly changed.

%% Refer to Table 2-1 for pin numbers.

CHAPTER 3 HARDWARE ARCHITECTURE

This chapter describes the operation and function of each functional block of the uPD77230R.

3.1 Instruction ROM and associated circuits.

The uPD77230R has an internal instruction ROM which consists of 2K 32-bit words. In master mode, instruction memory can be expanded by the addition of an external 4K 32-bit words. Figure 3-1 shows the internal instruction ROM and its associated circuits.

3.1.1 Instruction ROM.

The instruction ROM, which stores the microprogram of the uPD77230R, has a capacity of 2K words by 32 bits. The current address of the instruction ROM is indicated by the program counter (PC). Each instruction is stored in the instruction register (IR) prior to decoding by the decoder.

3.1.2 Program counter (PC).

The program counter consists of a 13-bit register that indicates the current address of the instruction ROM, and a counter that increments the lower 12 bits of that address. The highest bit of the PC, PC12, indicates whether the memory to be accessed is the internal instruction ROM or external instruction ROM. When PC12 is 0, internal memory is accessed, while PC12 = 1 is used for access to external memory.

3.1.3 Stack and stack pointer (SP).

The stack is 8 words deep and 13 bits wide, and works on a last-in, first-out (LIFO) basis. The current address of the stack is contained in the stack pointer. The stack is connected to the lower 13 bits of the main bus. Therefore, the contents of the stack location indicated by the stack pointer can be read from or written to, if desired, by appropriate specification in the SRC or DST fields of a transfer instruction.

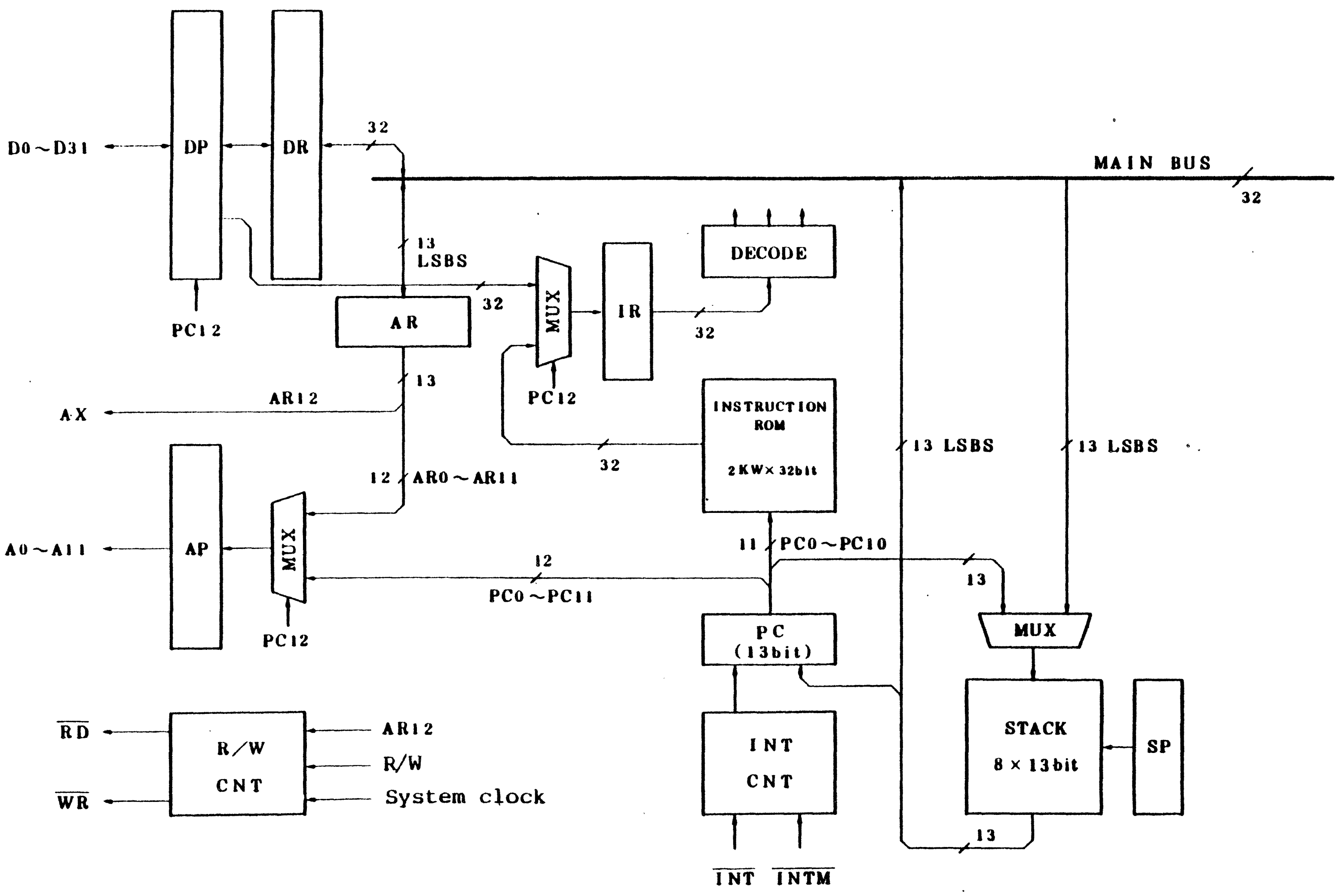


Figure 3-1. Instruction ROM and its associated circuits (in master mode).

3.2 Data ROM and its associated circuits.

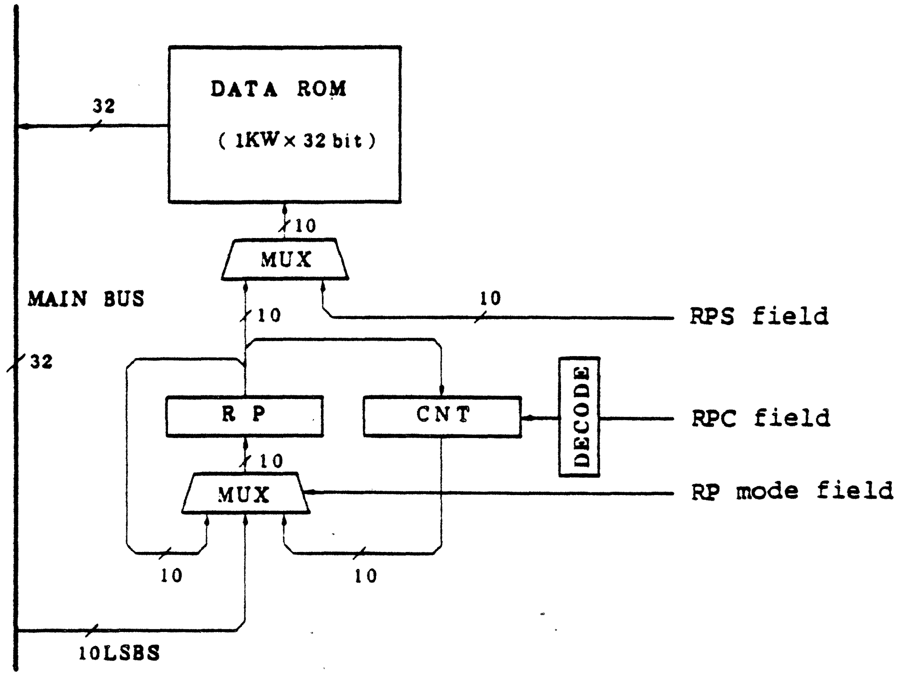
3.2.1 Data ROM.

Data ROM capacity is 1K words by 32 bits. The current address of this memory is specified by the ROM pointer (RP), or by the RPS field in the CNT field of the OP instruction.

3.2.2 ROM pointer (RP).

The ROM pointer is a 10-bit register that contains the current address of the data ROM. It is also connected to the lower 10 bits of the main bus. RP modifications are specified by the two RP bits in the CNT field of an OP instruction. The modifications which are available are increment RP, decrement RP, and add 2^n to RP.

Figure 3-2. Data ROM and its associated circuits.



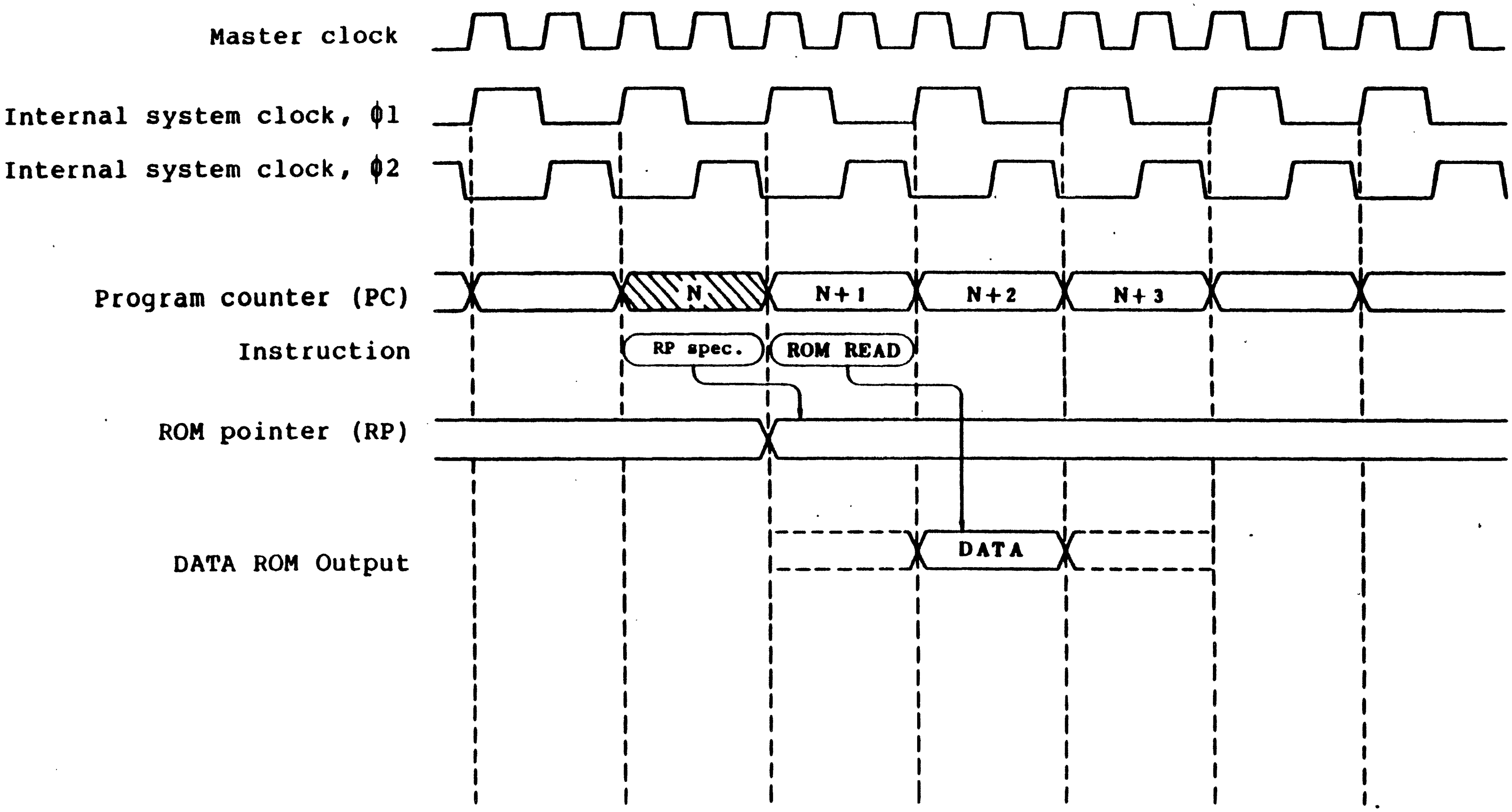


Figure 3-3. Internal data ROM access timing.

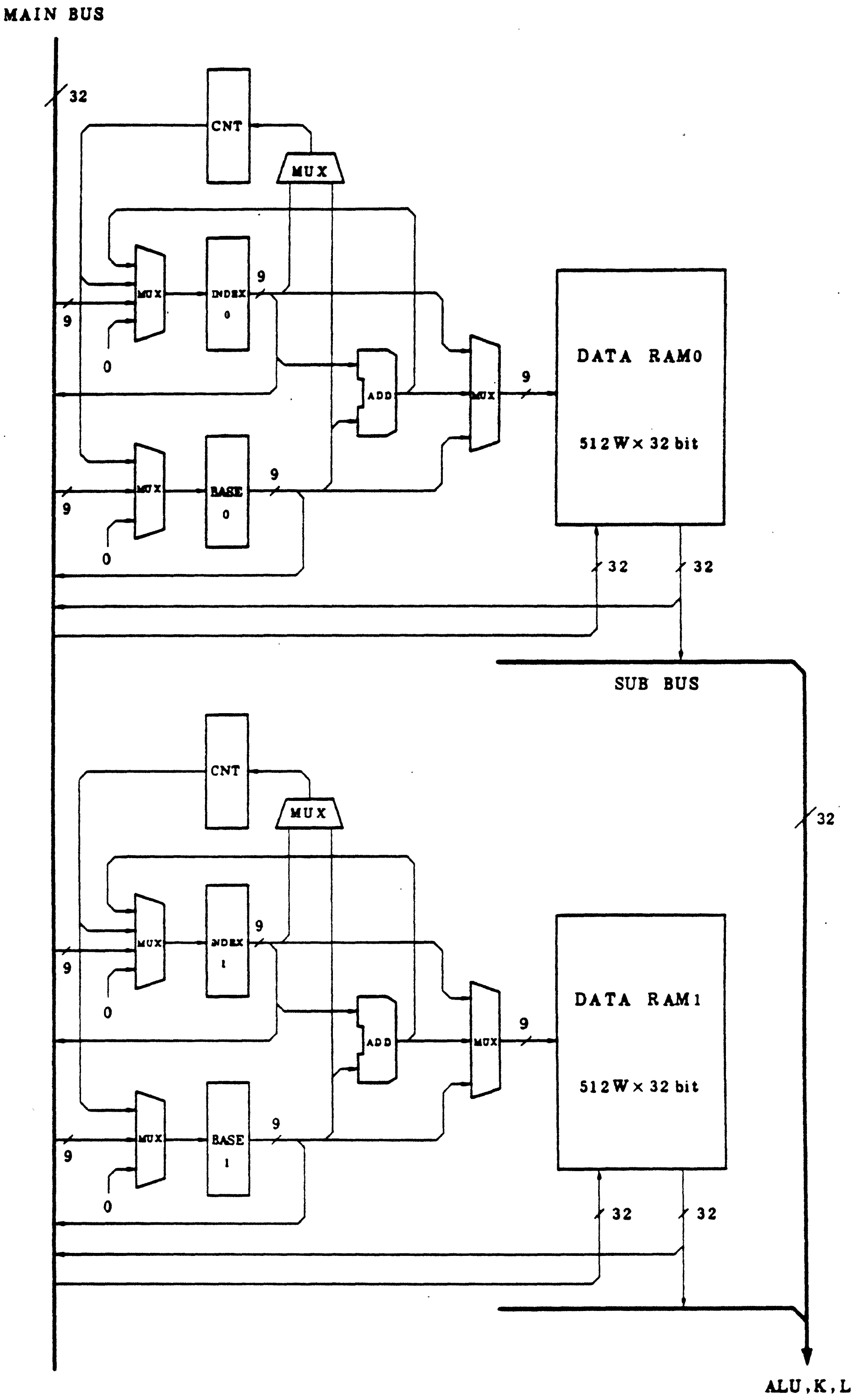
3.3 Data RAMs and their associated circuits.

3.3.1 Data RAMs 0 and 1.

Two blocks of RAM are provided for data storage. Each block consists of 512 words by 32 bits. The current address for data RAM 0 is specified by base pointer 0 (BASE0) and index register 0 (INDEX0). Likewise, the address for data RAM 1 is specified by base pointer 1 (BASE1) and index register 1 (INDEX1). Figure 3-4 is a block diagram of this configuration.

The output from each data RAM block is connected to the main bus. In addition, each data RAM block is connected to a sub-bus, which routes the RAM data to the P register of the ALU, and another sub-bus which routes data to the K and/or L registers at the input to the multiplier. Thus, data transfers from the RAM blocks can occur in parallel with other data transfers across the main bus, and in parallel with an ALU operation.

Figure 3-4. Data RAMs and their associated circuits.



3.3.2 Data RAM address.

The address for each data RAM block is specified by the appropriate base pointer and index register. Each of these registers is 9 bits wide, and is connected to the lower 9 bits of the main bus.

The following addressing modes are available:

1) address = base pointer
2) address = index register
3) address = base pointer + index register

The contents of the CNT field in the OP instruction specify which mode is used. The specification made becomes valid on the following instruction cycle.

3.3.3 BASE0 and BASE1.

Base pointer 0 (BASE0) specifies the base address for data RAM 0, while base pointer 1 specifies the base address for data RAM 1. Each of these 9-bit pointers are connected to the lower 9 bits of the main bus, and are therefore accessible to/from the other registers on the bus.

(1) General base pointer operation.

The base pointers can be modified as follows, in accordance with the CNT field of the OP instruction:

* Increment
* Decrement
* Clear
* NOP

(2) Special base pointer operation.

The base pointers can count either as ordinary binary counters, or as modulo counters, with the modulus specified by the BASEn bits (n=0,1) in the CNT field of the OP instruction. In modulo count mode, the carry to a specified bit position is inhibited, so that the bits higher than and including the one specified are not affected by any counting operations. The bits lower than the one specified will go through a ring count (i.e. wrap around) upon successive increments or decrements.

Example: n=4 specifies a modulo 2^4 count; when incremented, the base pointer will count between 0 and 15 and then wrap around to 0 again and continue counting.

Figure 3-5. Base pointer modulo count operation.

```
|<------no change------->|<---ring count---->|
 ____________________________________________
|    |    |    |     |     |     |    |    |    | base
-------------------------------------------- pointer
 8                   4     3              0
```

The modulo count number n can be between 0 and 7. When the modulo count number is 0, the base pointer will behave as an ordinary 9-bit binary counter.

Note that the starting address of the count need not be zero; i.e. it is possible to specify a count of 2^2 starting at address 014H, for example, and count through the range 014H - 017H.

3.3.4 INDEX0 and INDEX1.

Index register 0 (INDEX0) specifies the index address for data RAM 0, and index register 1 (INDEX1) specifies the index address for data RAM 1. Each of these 9-bit registers is connected to the lower 9 bits of the main bus, and both are therefore accessible to/from the other registers on the bus.

(1) Index register operation.

The index registers can be modified as follows, in accordance with the CNT field of the OP instruction:

* Increment
* Decrement
* Clear
* Store
* NOP

When the store operation is specified, the base pointer and the index register contents are added together, and the sum replaces the previous value in the index register.

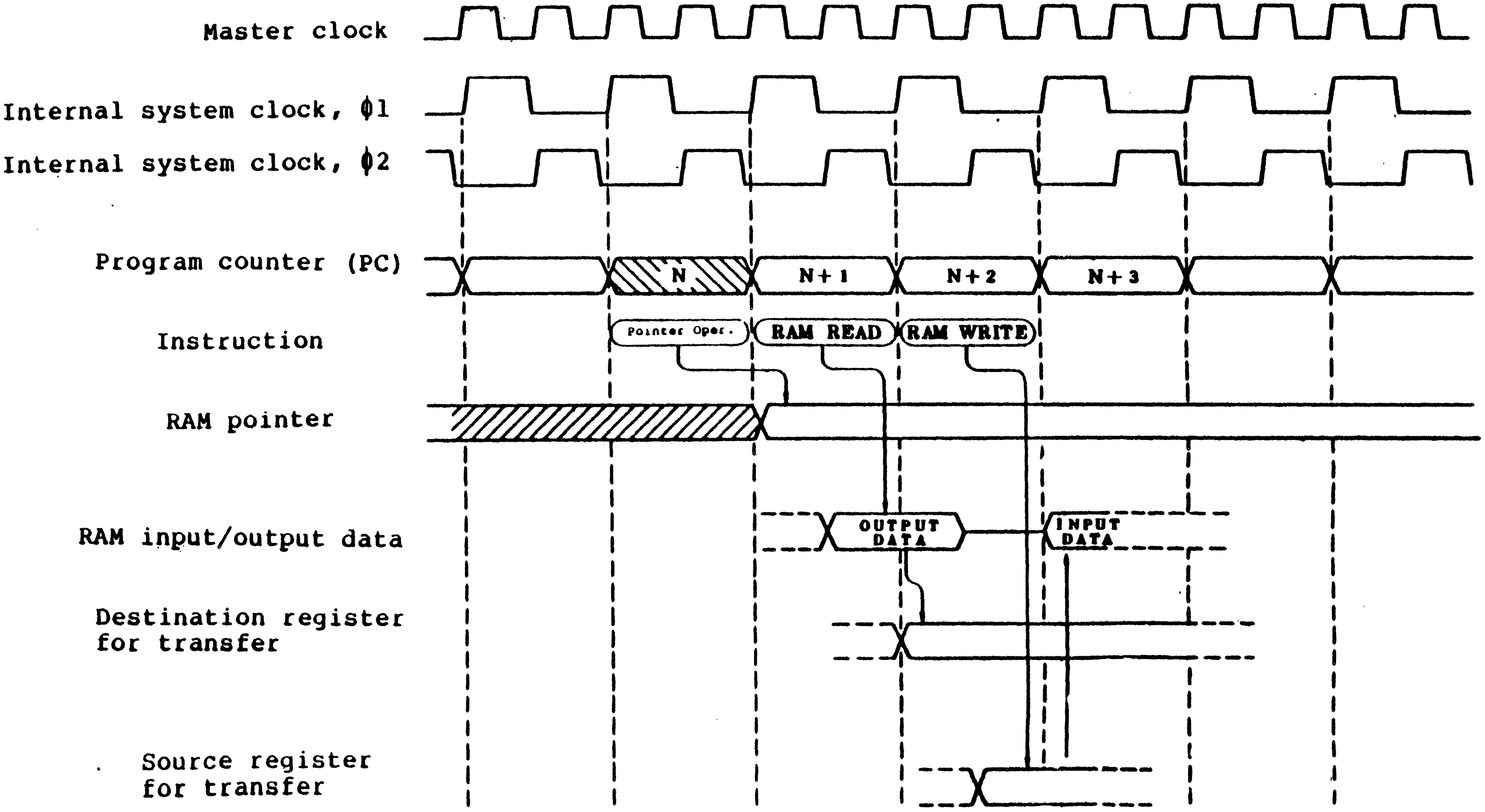


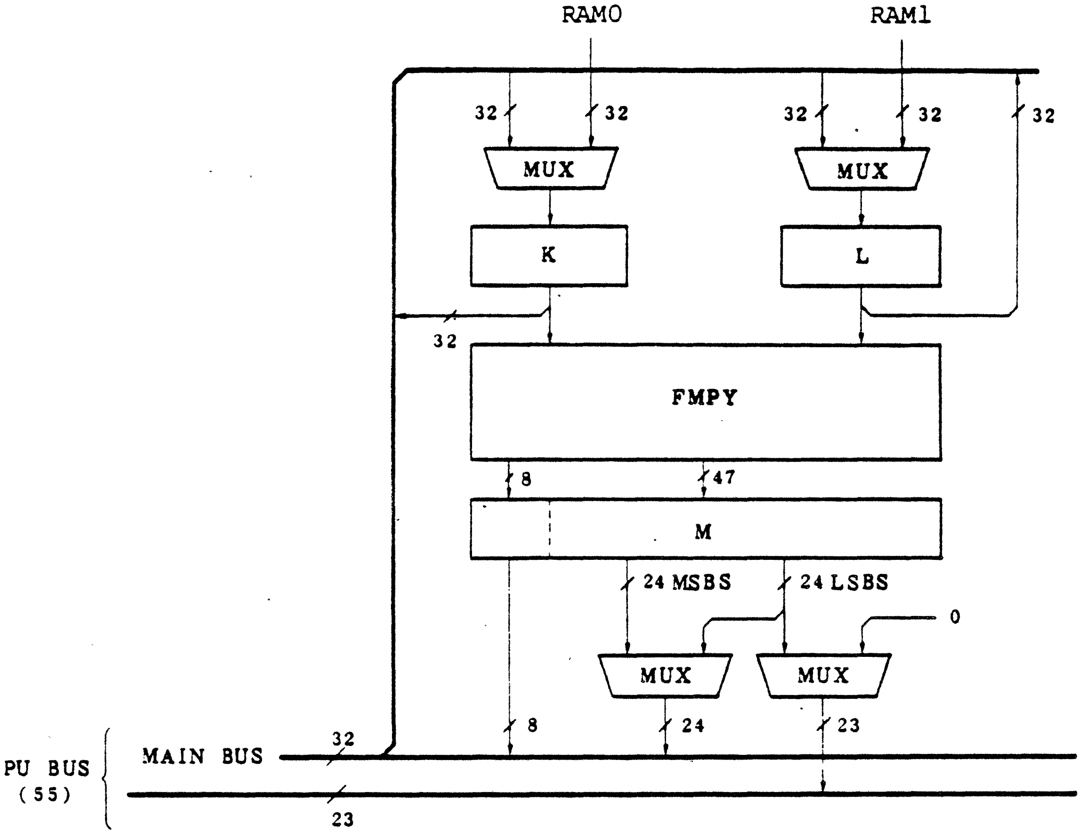
Figure 3-6. Internal data RAM access timing.

3.4 Floating point multiplier section.

3.4.1 Floating point multiplier (FMPY).

The multiplier generates the product of the 32-bit data in the K register and the 32-bit data in the L register. The K and L register contents are in 8-bit exponent, 24-bit mantissa format. The 55-bit product (8-bit exponent, 47-bit mantissa) is stored in the M register. Refer to Figure 3-7 for a block diagram of the FMPY and its associated circuits.

Figure 3-7. FMPY and associated circuits.



3.4.2 K and L registers.

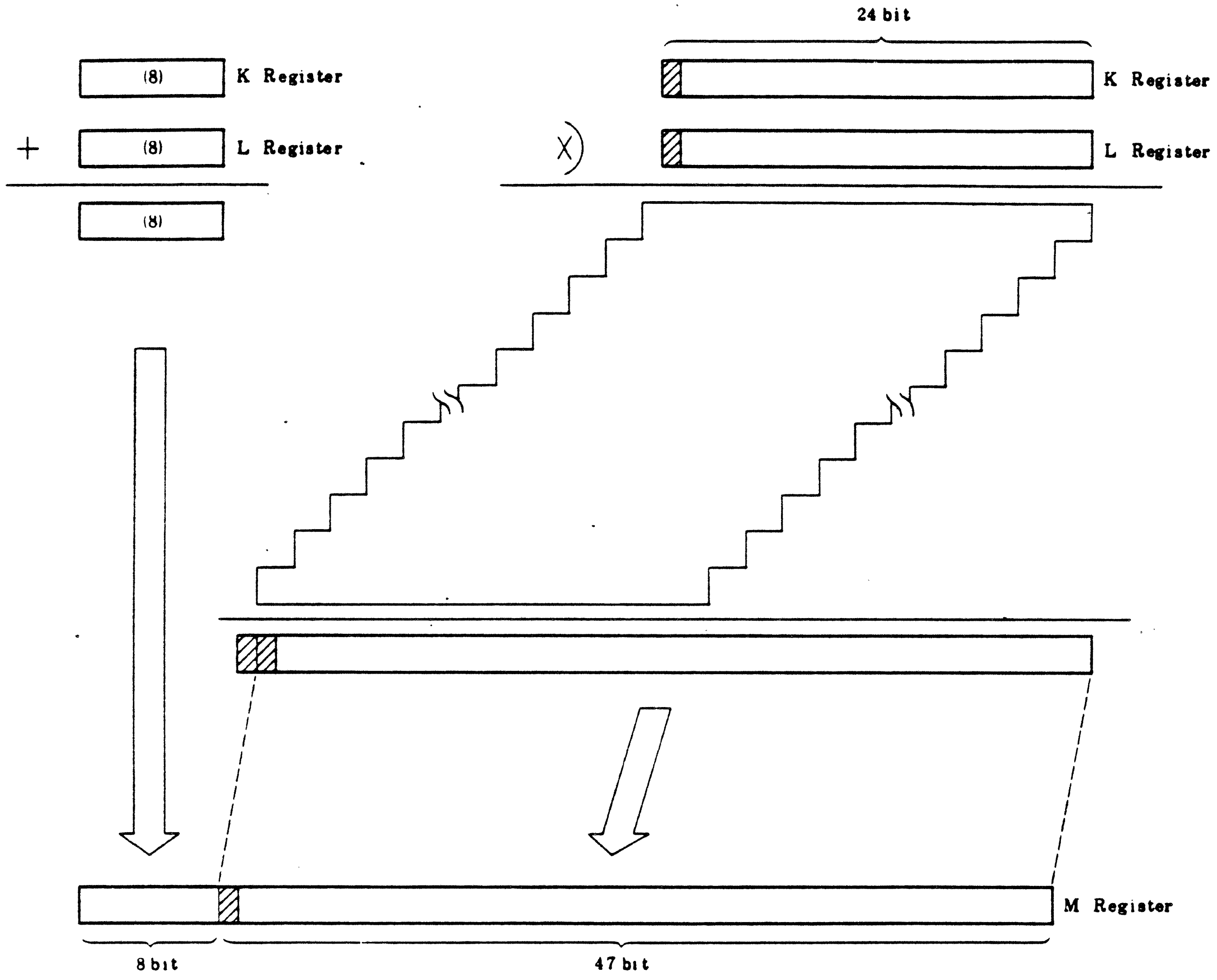
These are 32-bit registers which hold the inputs to the floating point multiplier (FMPY). They can also be used as general purpose registers because they can be written to or read from the main bus. In addition, data from data RAM 0 can be loaded into the K register, or data from data RAM 1 can be loaded into the L register, via independent sub-busses.

3.4.3 M register.

Once the numbers to be multiplied are stored in the K and L registers, the product is automatically generated and stored in the M register. Refer to Figure 3-8 for a representation of the multiplication process. The 55-bit M register is in 8-bit exponent, 47-bit mantissa format. The timing of a multiplication is shown in Figure 3-9.

The data in the M register can be transferred to the main bus (32 bits), the PU bus (55 bits), and the P register (55 bits) of the ALU, according to the transfer format specifications described below.

Figure 3-8. Arithmetic operation of FMPY.



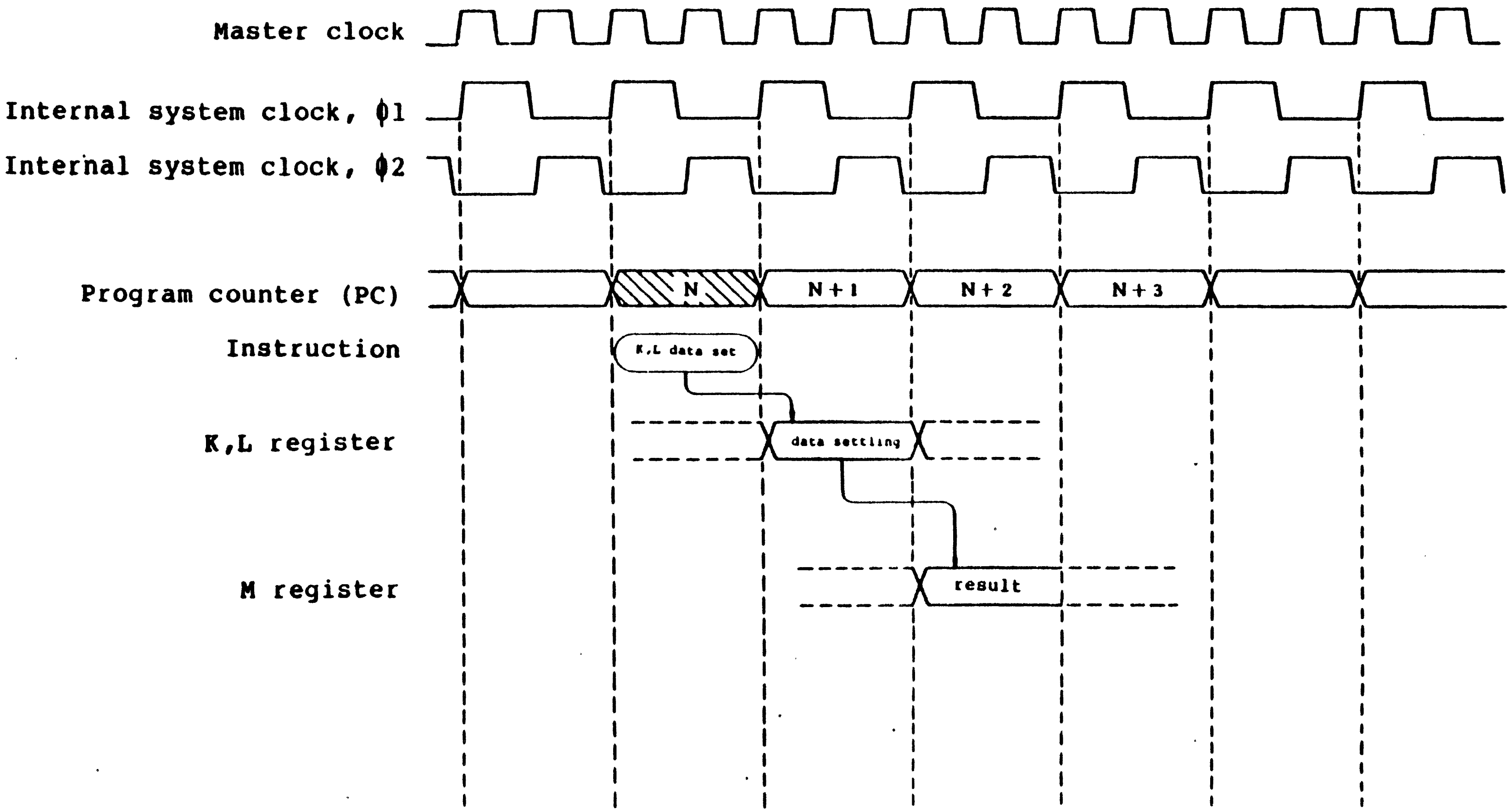


Figure 3-9. Multiplication timing.

There are two formats which are available for transferring data from the M register to any other register. The desired format is specified in the SRC field of the OP instruction. The following transfer formats are depicted in Figure 3-10:

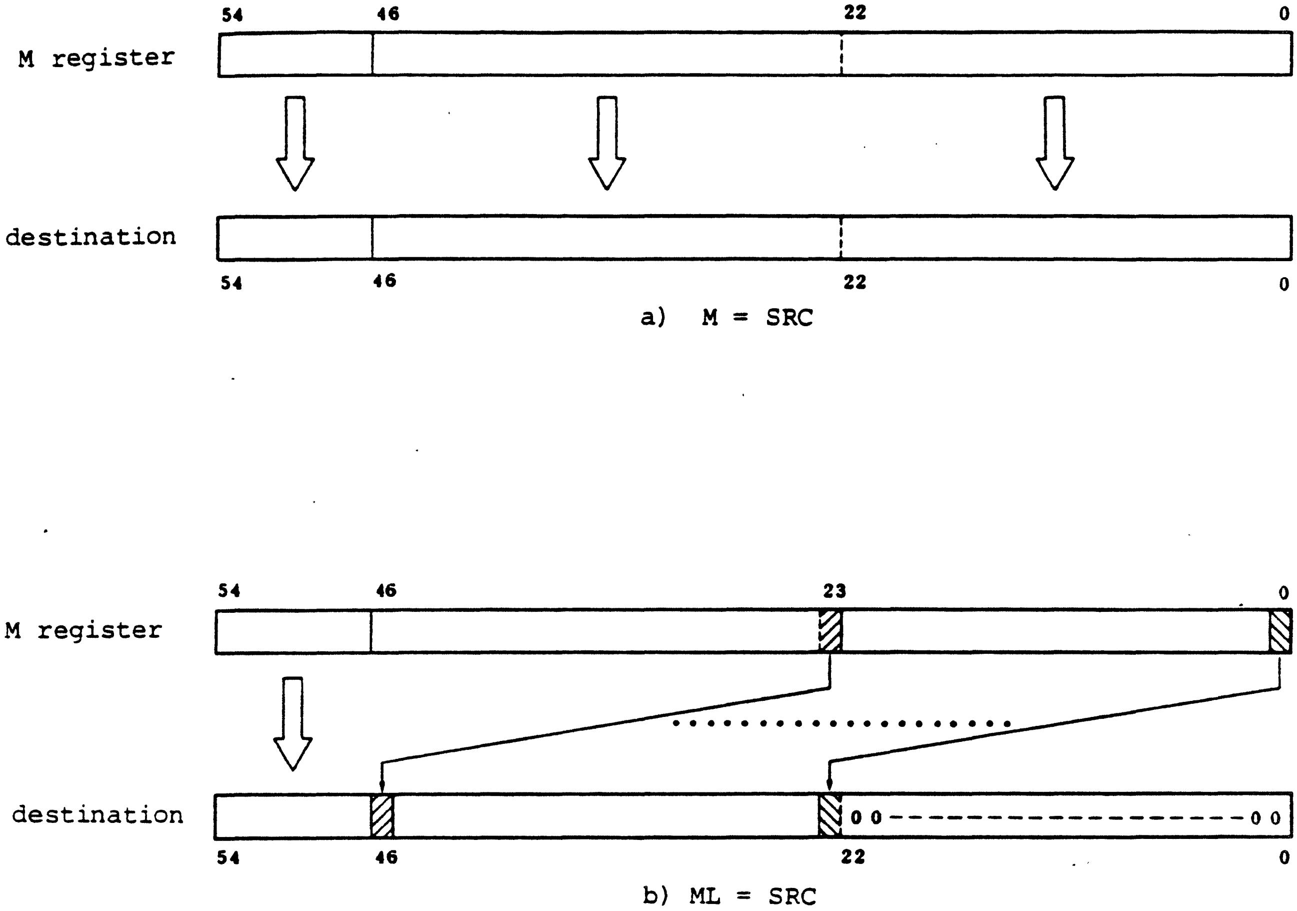
(1) SRC field = M

If the destination register is 32 bits wide, then the higher 32 bits of the M register are transferred to the destination register. If the destination register is 55 bits wide, the 55 bits of the M register are transferred unaltered to the destination register.

(2) SRC field = ML

The exponent (i.e. highest 8 bits) of the M register is transferred directly to the exponent of the destination register, and the lower 24 bits of the M register are transferred to the 24 bit mantissa of the destination register. If the destination register is 55 bits wide, then the lower 23 bits of the destination register are filled with zeros.

Figure 3-10. M register data transfer formats.



3.5 Processing Unit (PU).

The processing unit (PU) contains those sub-sections shown in Figure 3-11. The P and Q registers hold the inputs for the two arithmetic logic units (ALU and EAU), and the eight working registers may be used to store the results.

3.5.1 P register.

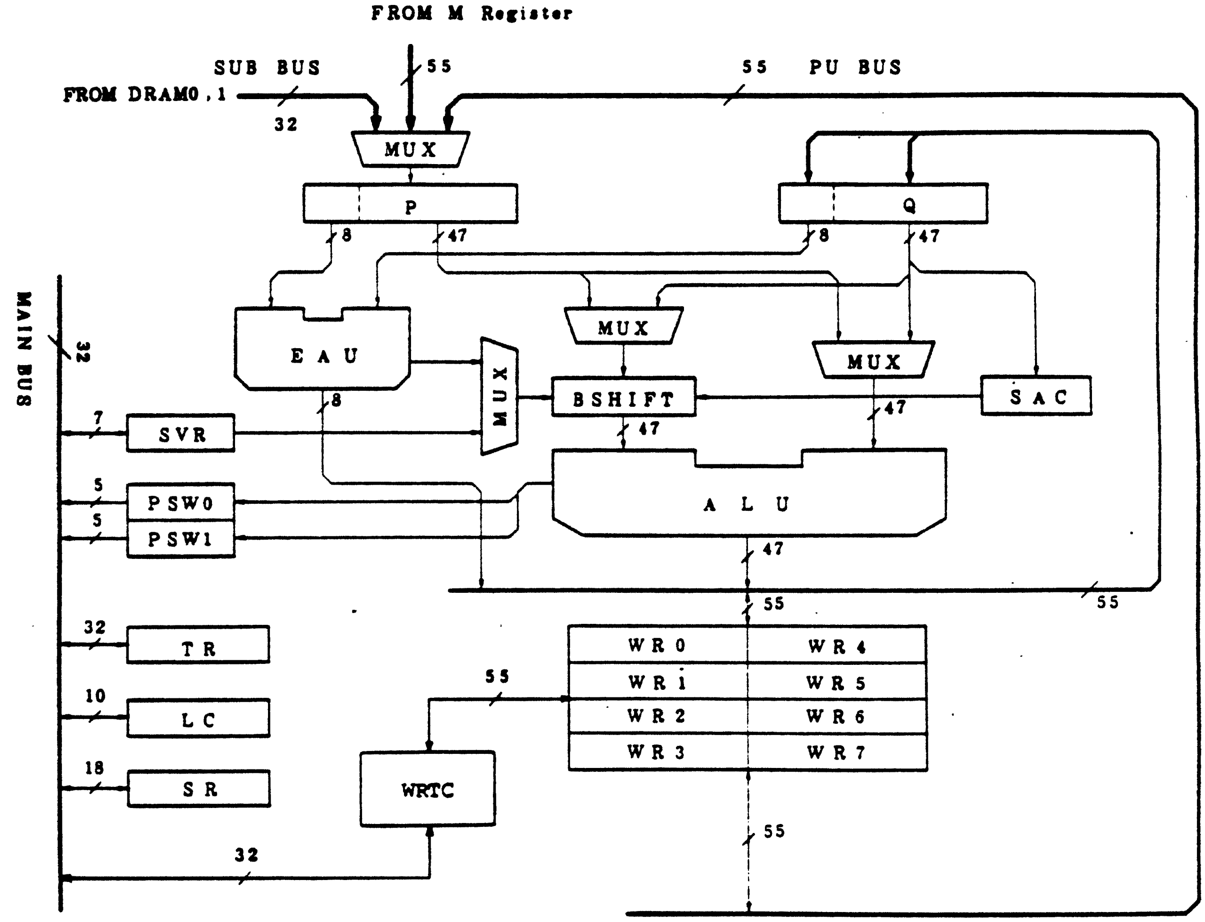
The P register serves as one of the two inputs to the ALUs. This 55-bit register is in 8-bit exponent, 47-bit mantissa format. Data may be input to the P register from the PU bus, M register, data RAM 0, or data RAM 1, in accordance with the P field specification in the OP instruction.

3.5.2 Q register.

The Q register is the other input to the ALUs. The data format in the Q register is the same as the format in the P register. Data may be input to the Q register from any of the eight working registers (WR0 thru WR7), in accordance with the Q field specification in the OP instruction.

If the same working register is specified as the input to the Q register in consecutive instructions, then the output of the ALU is fed directly to the Q register. This allows a single accumulator (i.e. working register) to be used for successive arithmetic operations. However, an intermediate result in a continuous accumulation will not be stored in the working register.

Figure 3-11. Processing Unit (PU) and associated circuits.



3.5.3 Shift and count circuit (SAC).

This circuit determines the "shift value" of the mantissa stored in the Q register. The shift value is the number of bits by which the mantissa must be left shifted so that the MSB of the fractional part of the mantissa (i.e. the bit to the right of the sign bit) becomes a numerical 1. For example, a 4 bit two's complement mantissa equal to 0001 would have a shift value of 2, while 1100 would have a shift value of 1. This shift value is used by the barrel shifter (BSHIFT) in performing normalization and format conversion.

3.5.4 Barrel shifter (BSHIFT).

This unit performs shifting operations on the mantissa stored in the P register or the Q register. The number of bits shifted is determined by the SAC or EAU in the processing unit, or by shift control fields in the OP instruction (e.g. SHV, SHLM fields). The shift operations performed by BSHIFT are described below.

(1) Mantissa alignment in floating point operations.

In a floating point arithmetic operation, the EAU compares the exponents of the data stored in the P and Q registers. The mantissa of the number with the smaller exponent will be right shifted, and its exponent incremented, until the two exponents are equal. The floating point arithmetic operation is then performed on the numbers with equal exponents.

(2) Mantissa shift in fixed point operation.

When working with fixed point numbers, the data in the P or Q register may be shifted by the shift value contained in the shift value register (SVR). See the description of the SHV field in chapter 4 for details.

(3) Normalization.

Data can be normalized by specifying NORM in the operation field of an OP instruction. "Normalization" means that the exponent and mantissa of the data in the Q register are adjusted in accordance with the value from the SAC (see above). After normalization, the 2^{-1} bit will be a numerical 1.

(4) Floating point to fixed point conversion.

The mantissa stored in the Q register will be shifted by the value of its exponent. If the exponent is negative, the mantissa will be shifted to the right; if the exponent is positive, the mantissa will be shifted to the left.

(5) Unary shift.

The mantissa of the data in the Q register can be shifted by specifying a shift instruction in the operation field of an OP instruction. The exponent will not be affected.

3.5.5 Arithmetic and logic unit (ALU).

The ALU performs arithmetic and logical operations on the 47-bit mantissas of the P and Q registers. The ALU performs the operation specified in the OP field of an instruction. If the operation is unary (i.e. only one input is needed), then the Q register is the source of the operand. Otherwise, the data from both the P and Q registers are used.

The result of an operation is stored in the working register selected in the Q field of the instruction. The ALU operation will affect the flag register (i.e. PSW) selected by the FIS and FC bits of the OP instruction.

In general, the results of an ALU operation are valid two instruction cycles after the instruction specifying the operation (refer to Figures 3-12 and 3-13). However, because of the pipelined nature of instruction execution, the results of one instruction are always available for the next instruction (except for certain external memory read operations).

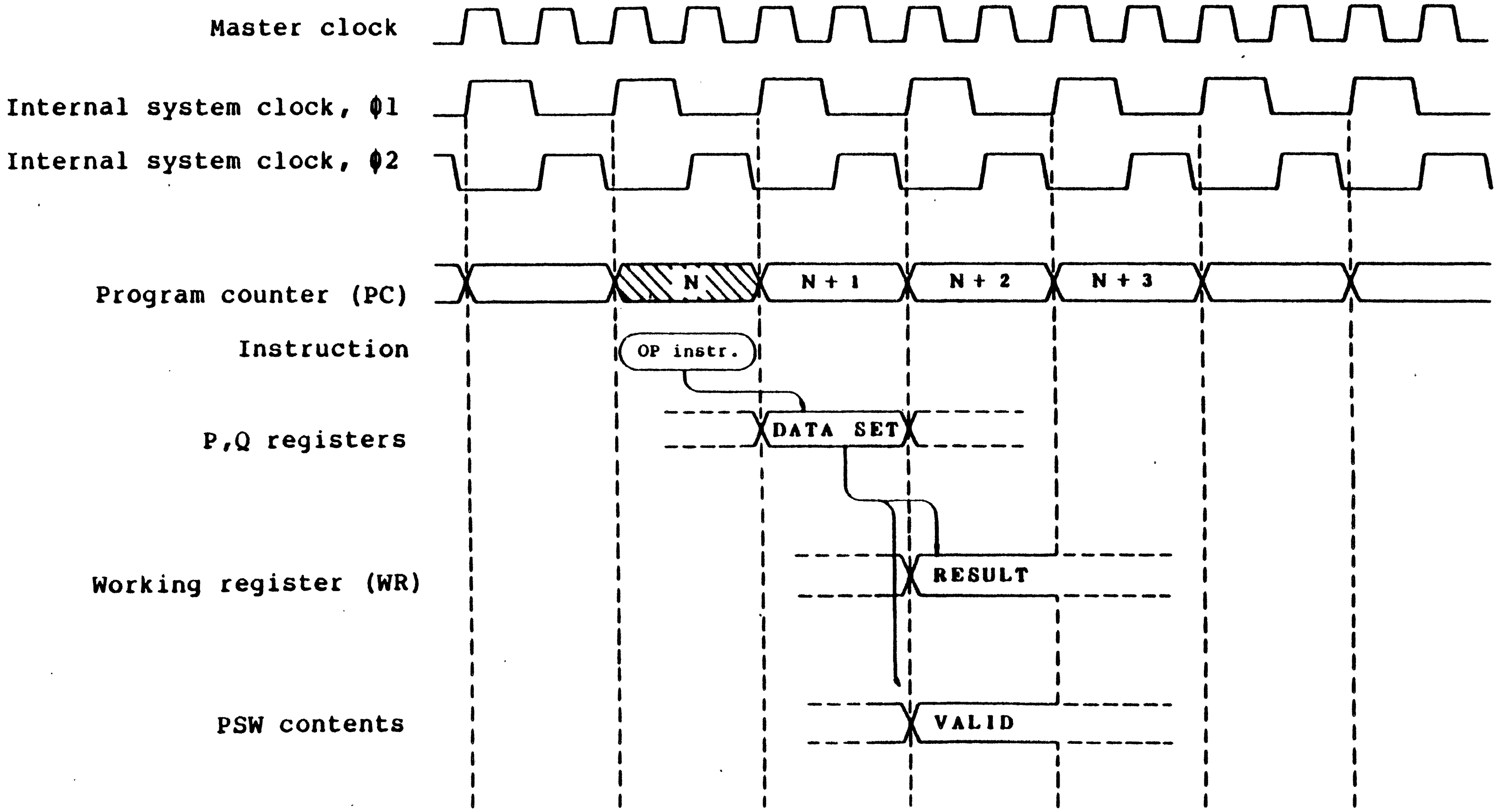


Figure 3-12. ALU operation timing.

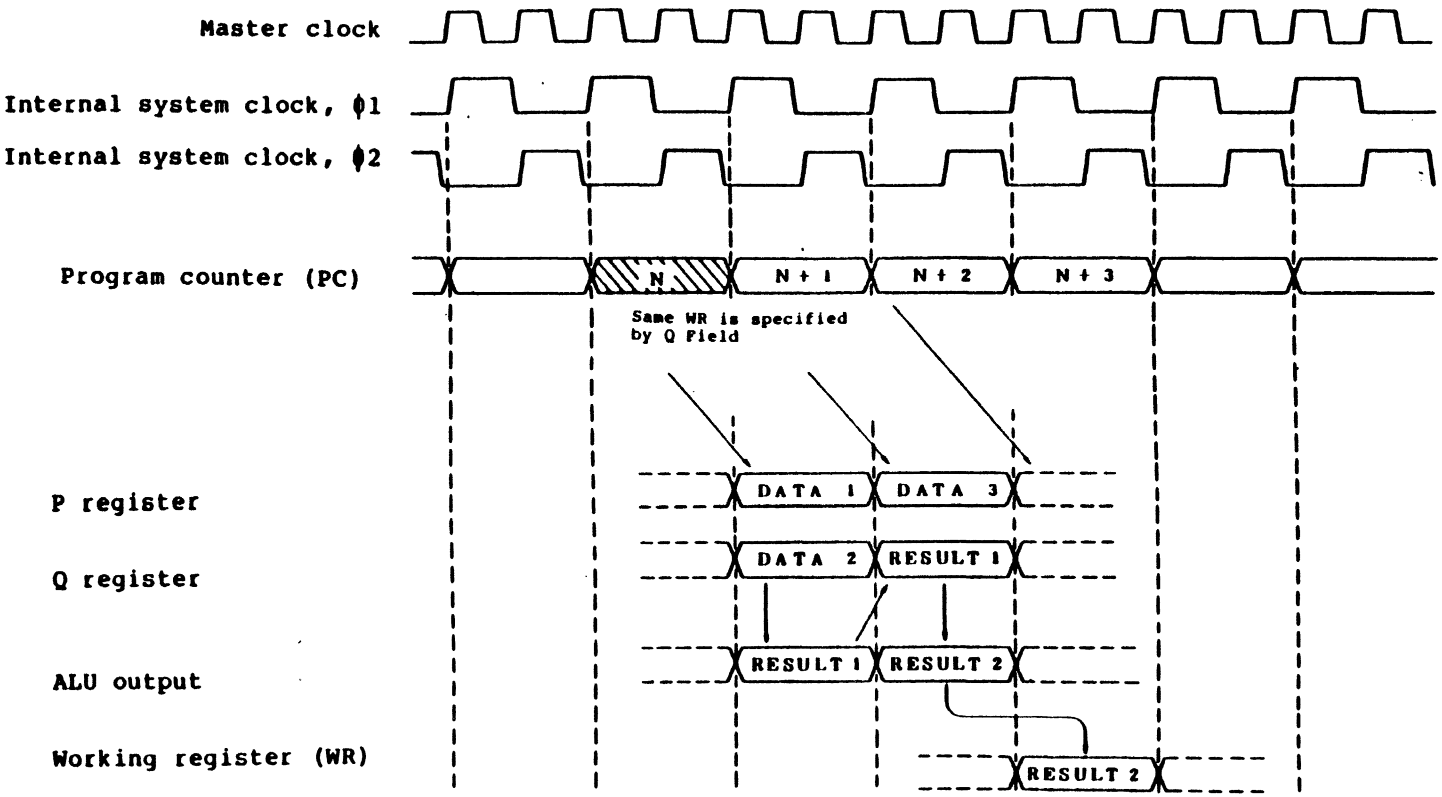


Figure 3-13. ALU operation timing when a single working register is used on successive operations.

3.5.6 Exponent arithmetic and logic unit (EAU).

This unit performs the required operations on the 8-bit exponents stored in the P and Q registers.

3.5.7 Working registers 0 thru 7 (WR0 - WR7).

These 55-bit registers store the results of the operations performed by the ALU and EAU. The current working register is specified by the Q field of the OP instruction. Data can be transferred to or from the working registers according to the transfer format specified in the OP instruction. The WT field controls the transfer format when the working register is the source of data for the transfer, while the WI field controls the transfer format when the working register is the destination. The transfer formats are described in Tables 3-1 and 3-2.

Table 3-1. Data transfer format, WRn = SRC.

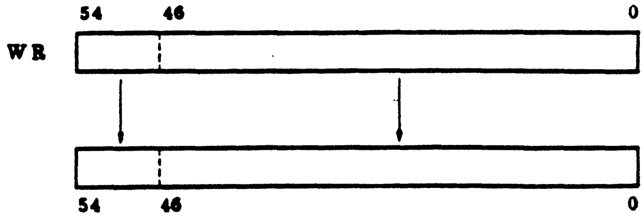
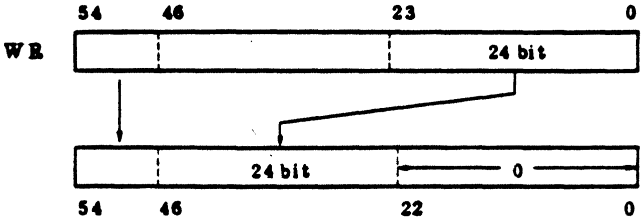
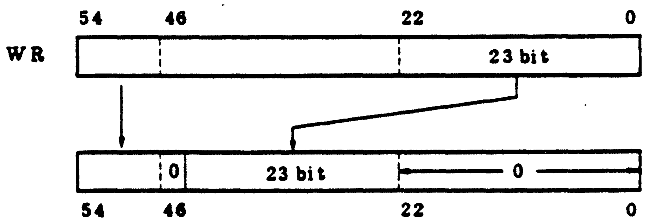
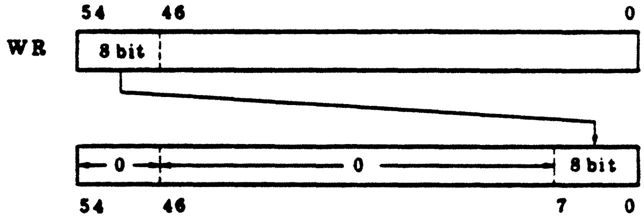
Mnemonic	WT field			Operation
	D21	D20	D19	
WRBORD	0	0	1	Ordinary data transfer from a working register 
WRBL24	0	1	0	Shifts lower 24 bits of WR to higher 24 bits of mantissa  54 46 23 0 WR 24 bit 24 bit 0 54 46 22 0
WRBL23	0	1	1	Shifts lower 23 bits of WR to higher 24 bits of mantissa; set sign bit of mantissa to 0.  54 46 22 0 WR 23 bit 0 23 bit 0 54 46 22 0
WRBL8E	1	0	0	Shifts exponent of WR to lower 8 bits of mantissa  54 46 0 WR 8 bit 0 0 8 bit 54 46 7 0

Table 3-1 (cont'd). Data transfer format, WRn = SRC.

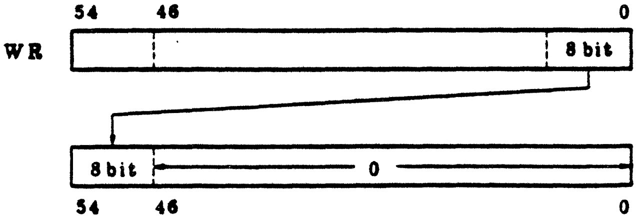
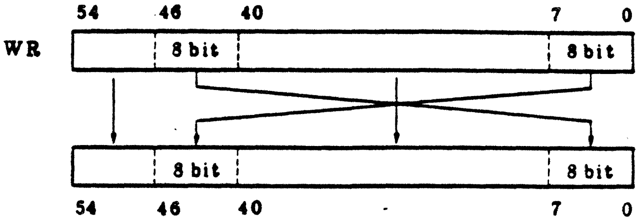
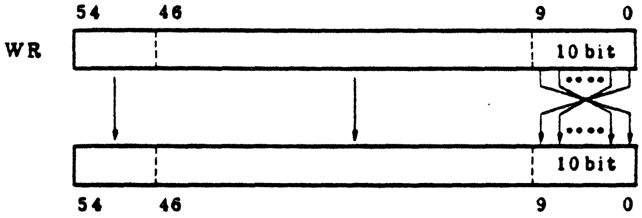
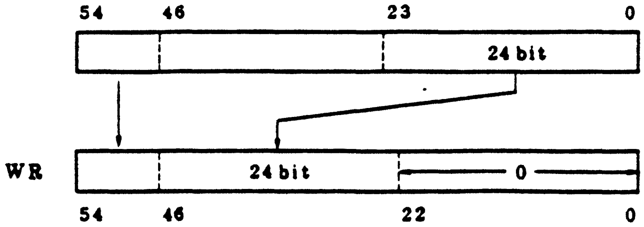
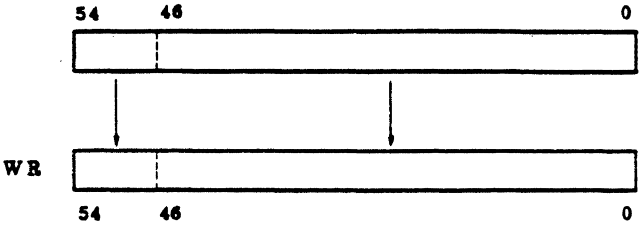
Mnemonic	WT field			Operation
	D21	D20	D19	
WRBEL8	1	0	1	Shifts the lower 8 bits of WR to the exponent 
WRBXCH	1	1	0	Exchanges the lower and higher 8 bits of the mantissa in a working register  54 46 40 7 0 WR 8 bit 8 bit 8 bit 8 bit 54 46 40 7 0
WRBRV	1	1	1	Reverses the lower 10 bits of a working register  54 46 9 0 WR 10 bit 10 bit 54 46 9 0

Table 3-2. Data transfer format, WR = DST.

Mnemonic	WI field		Operation
	D19	D18	
BWRL24	0	1	Shifts the lower 24 bits of the mantissa of a WR to the higher 24 bits  54 46 23 0 24 bit WR 24 bit 0 54 46 22 0
BWRORD	1	0	Ordinary data transfer to a WR 

3.5.8 Processor status words 0 and 1 (PSW0 and PSW1).

These 5-bit registers hold the ALU flags corresponding to the results of an ALU operation.

Figure 3-14. Processor status word bit configuration.

OVFE	C	Z	S	OVFM

Processor status word

OVFE	:	exponent overflow flag
C	:	carry flag
Z	:	zero flag
S	:	sign flag
OVFM	:	mantissa overflow flag

Both PSW0 and PSW1 are connected to the 5 lower bits on the main bus, and may be specified as a source or destination in a transfer operation. The flags in the PSW may also be tested by conditional branch instructions. Some noteworthy characteristics of the PSW are noted below.

(1) The FIS and FC bits of the CNT field in an OP instruction determine which PSW will be affected by the current instruction. This PSW specification becomes valid as soon as the specifying instruction is executed.

(2) If a PSW clear operation and a data transfer to the PSW are both specified in the same instruction, then clearing the PSW takes precedence, and the data is not transferred to the PSW.

(3) A PSW clear operation takes effect as soon as it is executed. Therefore, if the PSW is cleared by an OP instruction immediately following an OP instruction that performs an ALU operation, then the result of that ALU operation will not be seen in the PSW.

(4) A transfer to the PSW takes effect immediately upon execution of the transfer instruction. Therefore, if the transfer instruction immediately follows an OP instruction that performs an ALU operation, then the result of that ALU opertaion will not be seen in the PSW.

Table 3-3. PSW functions

PSW bit	Description
OVFE (overflow of exponent)	This bit is set to 1 if an overflow or underflow occurrs as a result of an EAU (exponent arithmetic unit) operation
C (carry)	This bit reflects the carry or borrow resulting from an arithmetic operation
Z (zero)	This bit is set to 1 when the mantissa in the result of an arithmetic operation is zero
S (sign)	This bit holds the sign bit (i.e. MSB) of the mantissa of the result of an arithmetic operation
OVFM (overflow of mantissa)	This bit is set to 1 if an overflow or underflow occurrs as a result of an ALU operation

Table 3-4. Effects of ALU operations on PSW.

ALU	CONTENTS OF PSW				
OPERATION	OVFE	C	Z	S	OVFM
NOP	*	*	*	*	*
INC	*	$	$	$	$
DEC	*	$	$	$	$
ABS	*	0	$	$	$+
NOT	*	0	$	$	0
NEG	*	$	$	$	$+
SHLC	*	$	$	$	0
SHRC	*	$	$	$	0
ROL	*	0	$	$	0
ROR	*	0	$	$	0
SHLM	*	0	$	$	0
SHRM	*	0	$	$	0
SHRAM	*	0	$	$	0
CLR	0	0	1	0	0
NORM (NORM.)	$	0	$	$	0
(ROUNDING)	$	$	$	$	$
(FLT - FIX)	0	$	$	$	$
(FIX M. A.)	0	$	$	$	$
CVT	X	0	$	$	0
ADD	*	$	$	$	$
SUB	*	$	$	$	$
ADDC	*	$	$	$	$
SUBC	*	$	$	$	$
CMP	$	$	$	$	$
AND	*	0	$	$	0
OR	*	0	$	$	0
XOR	*	0	$	$	0
ADDF	$	$	$	$	$
SUBF	$	$	$	$	$

$: Flag will be affected by the result of the operation.
0 : Flag will be reset to "0".
1 : Flag will be set to "1".
* : Previous condition of flag will be preserved.
X : Undecided.
+ : If the original data in the mantissa was 80...0H,
OVFM = "1" after operation.

3.5.9 Loop Counter (LC).

The loop counter is a 10-bit down counter connected to the lower 10 bits of the main bus. It can be specified as either the source or destination of a transfer operation.

The contents of the LC can be decremented when so specified in the CNT field of an OP instruction. When an LC decrement generates a borrow (it has its own borrow flag, not to be confused with the PSW), the count operation is disabled, and the next instruction to be executed is executed as though it were a NOP.

3.5.10 Shift value register (SVR).

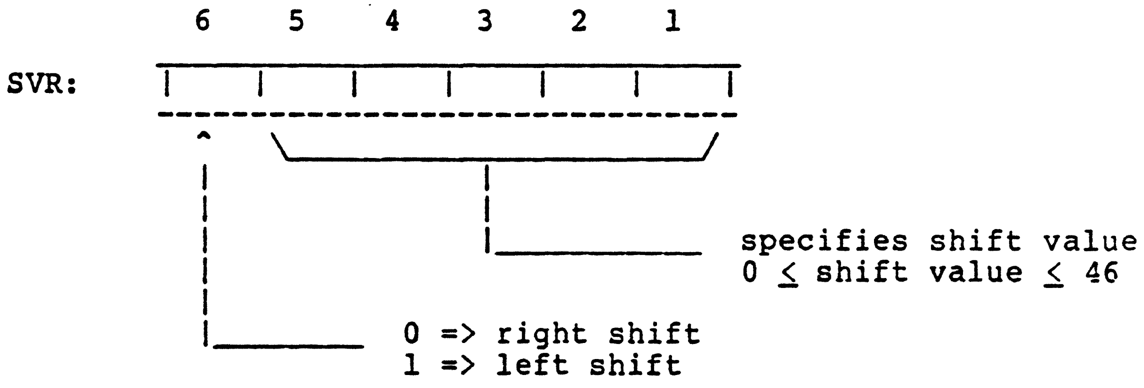
This 7-bit register specifies a shift value for the barrel shifter (BSHIFT). It is connected to the lower 7 bits of the main bus. The characteristics of the SVR are listed below.

(1) The SVR can be specified as either the source or destination in a transfer operation.

(2) The contents of the SVR can be changed by the SHV bits of the CNT field in an OP instruction.

(3) If the SVR is specified as the destination of a transfer in the same instruction in which the SHV field would change the contents of the SVR, then the transfer gets priority over the SHV field.

Figure 3-15. Shift value register bit configuration.



3.5.11 Temporary register (TR).

This is a 32-bit general purpose register connected to the main bus. Use of the TR is illustrated in the following example.

Temporary register application example:

In the uPD77230R, the Load Immediate Data (LDI) instruction can only load 24 bits at a time. In order to load a 32-bit floating point number, the TR must be used as follows:

(1) Specify TR as the destination in an LDI instruction. The 24-bit immediate data is then loaded into the 24 lower bits of the TR, and the 8 MSBs of the TR are set to zero.

(2) Specify TRE as the destination in an LDI instruction. The lower 8 bits of the 24-bit immediate data field are then loaded into the upper 8 bits of the TR (i.e. the exponent). The lower 24 bits of the TR will not be affected by this transfer to TRE.

3.5.12 Overflow and underflow processing.

The uPD77230R will automatically compensate for any overflow or underflow that occurs during a normalization or floating point operation. The overflow or underflow is detected inside the ALU or EAU and is compensated for at the output section of the ALU and EAU.

(1) Compensation in floating point operation.

Refer to Table 3-5 below.

Condition	Compensation
Overflow/ underflow of mantissa	(1) 1 bit right shift of mantissa; carry of PSW input to MSB. (2) Add 1 to exponent. NOTE: If exponent overflows as a result of this operation, the following process is performed: * If mantissa is positive (MSB = 0), then exponent = max. positive value (7FH) & mantissa=max. positive value (7F...FEH) * If mantissa is negative (MSB=1), then exponent = max. positive value (7FH) & mantissa = min. negative value (80...0H)
Mantissa = 0	(1) Mantissa = 0 (0...0H) & exponent = 80H

Table 3-5. Compensation for overflow/underflow in floating point operation.

(2) Compensation in normalization.

If an underflow occurs in the exponent as a result of a normalization, the exponent and mantissa are set to the following values:

- * Exponent = 0 (80H)
- * Mantissa = 0 (0...0H)

(3) Compensation in normalization with rounding.

If an overflow occurs in the mantissa as a result of a normalization with rounding operation, it is compensated for in the same manner as for a floating point mantissa overflow/underflow, described in Table 3-5.

(4) Compensation during format conversion.

If the exponent is greater than the shift value determined by the SAC during a conversion from floating point to fixed point format, the mantissa is set to its positive or negative saturation value, depending on its sign, as shown below:

- * If mantissa is positive (MSB=0), then
  mantissa = max. positive value (7F...FEH).

- * If mantissa is negative (MSB=1), then
  mantissa = min. negative value (80...0H).

3.6 System control section.

3.6.1 Clock generator.

The clock generator supplies the internal system clocks (including the serial clocks, if desired) by means of a crystal oscillator connected across the X1 and X2 pins, as shown in Figure 3-16. If an externally generated clock is available, it may be supplied to the X1 pin, and the X2 pin should be grounded.

The internal system clock consists of two phases, and is obtained by dividing the master clock (i.e. crystal oscillator) frequency by two. The relation between the two phases and the crystal oscillator is shown in Figure 3-17.

The serial clock (SCK) is obtained by dividing the master clock by eight. The SCI bit of the status register determines if SCK is output from the clock generator to the serial input clock pin (SICK), or if the serial input is performed using an external clock input to the SICK pin.

Likewise, the SCO bit of the status register determines whether the serial output transfer is clocked by SCK or by an external clock input to the SOCK pin.

Figure 3-16. Crystal oscillator connection to X1 and X2 pins.

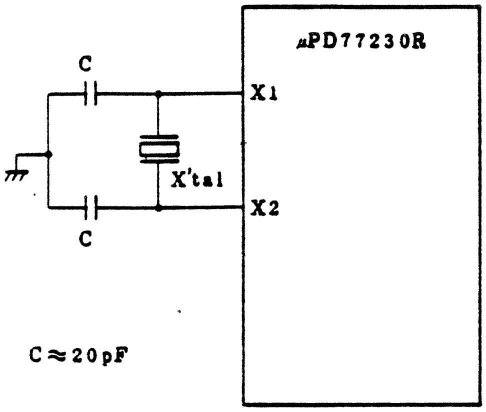
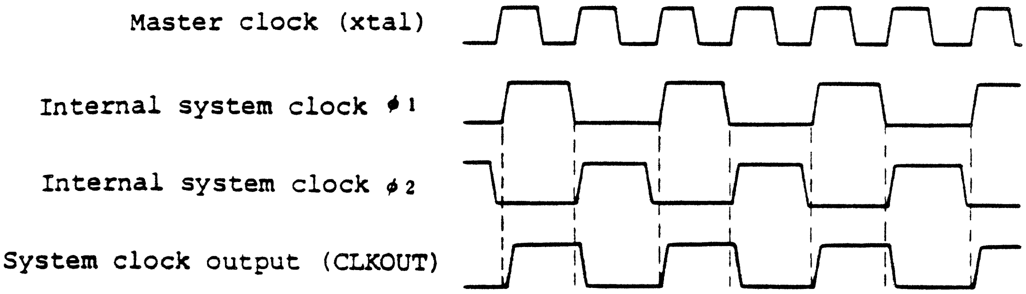


Figure 3-17. Internal system clock phases.



3.6.2 Interrupt controller.

The uPD77230R has two interrupt pins: INT\, for nonmaskable interrupts, and INTM\ for maskable interrupt.

(1) INT\ (nonmaskable interrupt).

- INT\ has higher priority than INTM\.
- Interrupts are detected at the falling edge of INT\.
- The interrupt address is 10H.
- INT\ must be at least 3 system clocks wide.

(2) INTM\ (maskable interrupt).

- INTM\ has lower priority than INT\.
- Interrupts are detected at the falling edge of INTM\ when the EM bit of the status register is set to 1.
- When INTM\ is acknowledged, the EM bit of the status register is cleared to 0.
- The interrupt address is 100H.
- INTM\ must be at least 3 system clocks wide.

(3) Normal servicing procedure for INT\ and INTM\.

After an interrupt has been recognized (i.e after the falling edge of INT\ or INTM\), the interrupt is processed in the following sequence (refer to Figure 3-18):

(a) When execution of the current instruction is completed, the address of the currently fetched instruction (i.e. the return address) is pushed onto the stack.

(b) The currently fetched instruction is temporarily replaced by a NOP and executed. The execution of this NOP is required by the pipelined nature of program execution.

(c) While the NOP is executed, the program counter (PC) is loaded with the interrupt address, and the first instruction of the interrupt routine will be executed immediately following the NOP.

(d) After the interrupt has been processed, the return address is popped off the stack and loaded into the PC.

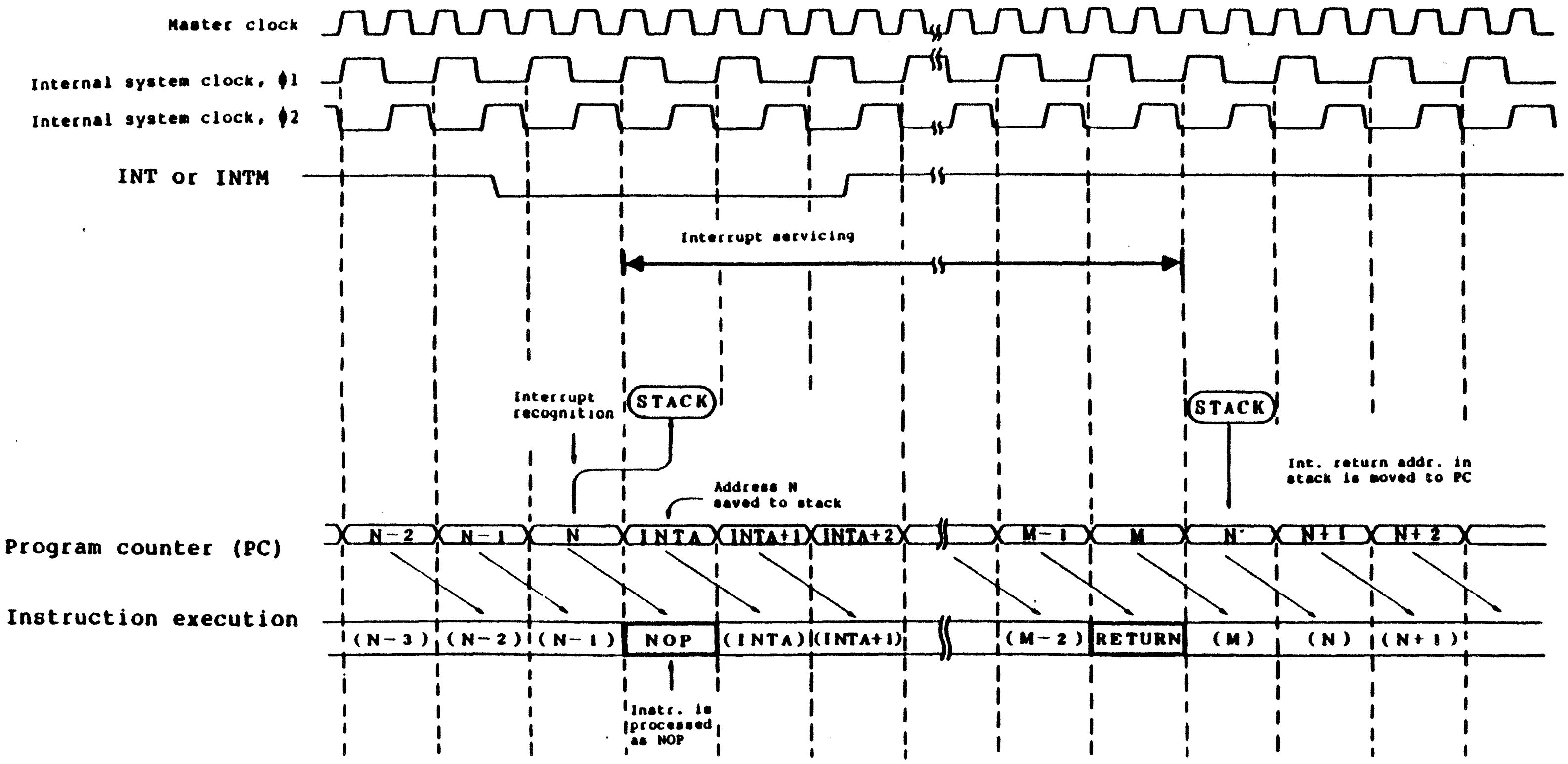


Figure 3-18. Normal interrupt servicing.

(4) Interrupt acknowledge during CALL instruction.

When an interrupt signal is acknowledged during the execution of a CALL instruction, the interrupt is processed in the following sequence (refer to Figure 3-19):

(a) The address two addresses after the CALL instruction address (i.e. the subroutine return address) is pushed onto the stack.

(b) The instruction following the CALL instruction is executed normally.

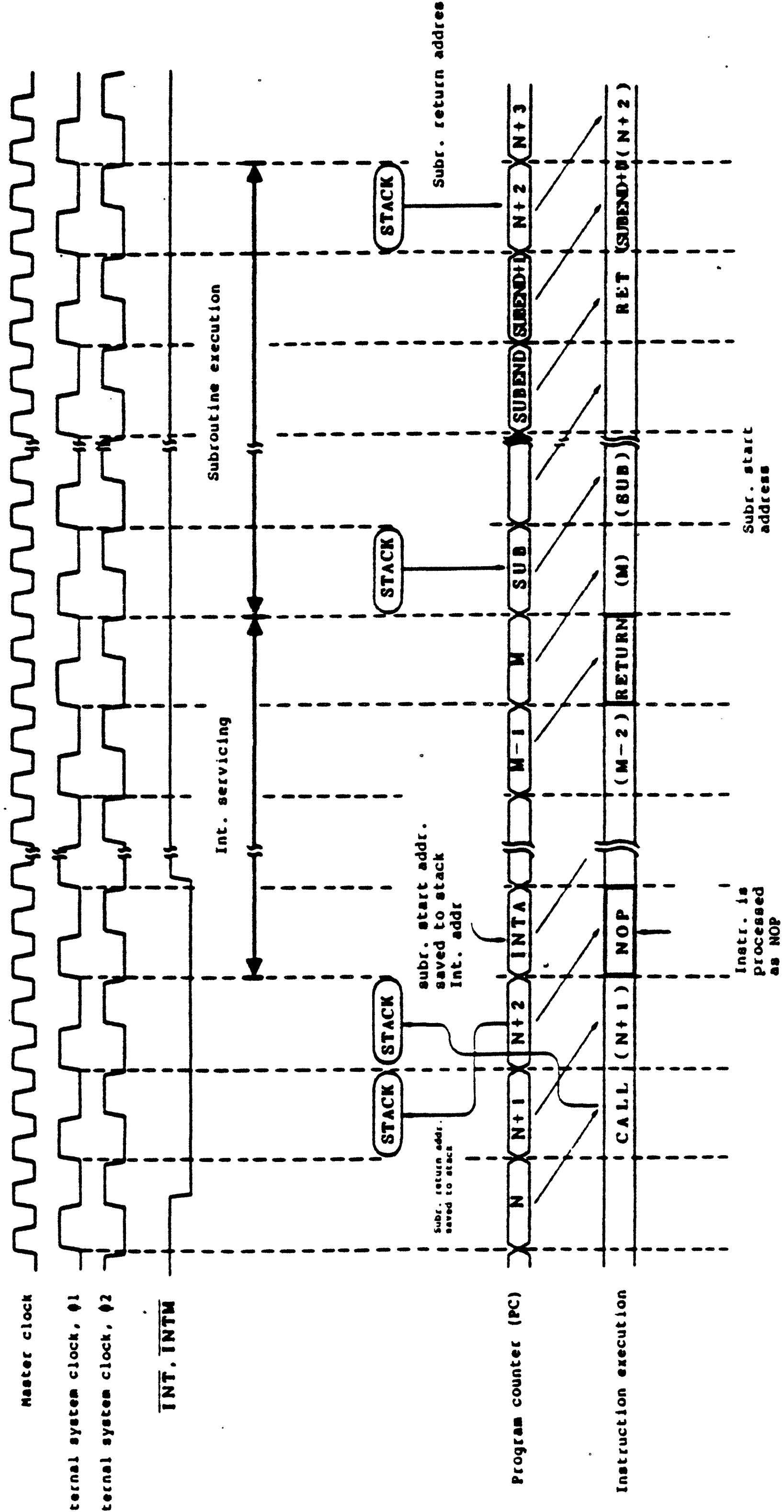
(c) The address of the instruction that would have been executed had the interrupt not occurred (i.e. the subroutine start address) is pushed on to the stack, and that instruction is temporarily replaced by a NOP and executed.

(d) While the NOP is executed, the program counter (PC) is loaded with the interrupt address, and the first instruction of the interrupt routine will be executed immediately following the NOP.

(e) After the interrupt has been processed, the subroutine start address is popped off the stack and loaded into the PC.

(f) After the subroutine has been executed, the subroutine return address is popped off the stack and loaded into the PC.

Figure 3-19. Interrupt acknowledge during CALL instruction execution.



(5) Interrupt acknowledge during a branch instruction other than a CALL instruction.

When an interrupt signal is acknowledged during the execution of a branch instruction (other than a CALL instruction), the interrupt is processed in the following sequence (refer to Figure 3-20):

(a) The address two addresses after the branch instruction address (i.e. the interrupt return address) is pushed onto the stack.

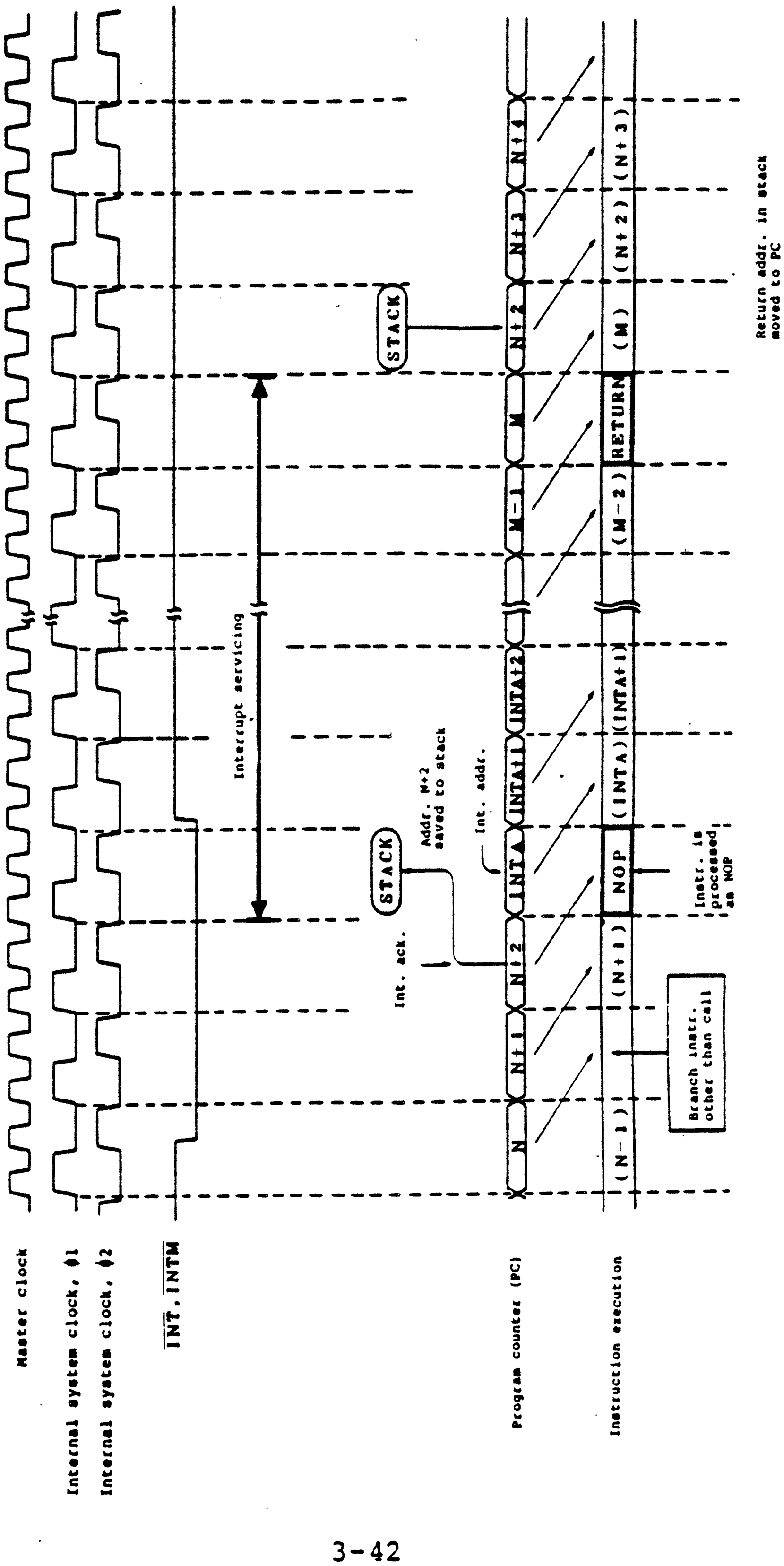
(b) The instruction following the branch instruction is executed normally.

(c) The instruction at the interrupt return address is temporarily replaced by a NOP and executed.

(d) While the NOP is executed, the program counter (PC) is loaded with the interrupt address, and the first instruction of the interrupt routine will be executed immediately following the NOP.

(e) After the interrupt has been processed, the interrupt return address is popped off the stack and loaded into the PC.

Figure 3-20. Interrupt acknowledge during execution of branch instruction other than CALL.



3.6.3 Hardware reset.

The hardware is reset by an active low RESET\ signal which must be at least 3 system clocks wide.

(1) Internal status after hardware reset.

The following flags and registers are reset to 0 after a hardware reset.

Registers:

* Status register (SR)
* PSW0 and PSW1
* Program counter (PC)
* ROM pointer (RP)

Flags:

* SIAK, SOAK
* RQM
* Loop counter borrow
* P2 and P3 (in slave mode)

In addition, the following conditions will be in effect after RESET\.

* The address used for data RAM 0 and 1 is the base register address plus index register address.

* The address used for the data ROM is RP.

* The modulus of the base pointer count and the value of n in 2^n addition to the RP are reset to 0.

* The loop count operation is inactive.

* The contents of these registers just mentioned are not affected by RESET\.

(2) Output pin status after hardware reset.

The effect of a RESET\ on the output pins of the uPD77230R for both master and slave modes is listed in Table 3-6.

Table 3-6. Effect of RESET\ on output pins.

(a) Pins used in both master and slave modes.

Pin name	Status
A0 to A11	High impedance
RD	High level
WR	High level
CLKOUT	Internal system clock is output
SORQ	High impedance
SO	High impedance
A_x	Low level

(b) Pins used in master mode only.

Pin name	Status
D0 to D31	High impedance

(c) Pins used in slave mode only.

Pin name	Status
RQM	Low level
D0 to D7	High impedance
P2 and P3	Low level
I/O 0 to I/O 15	Status dependent on HRD\, HWR\, and CS\ pins.

3.6.4 Status register (SR).

SR is a 20-bit register that specifies the internal status of the uPD77230R. It is connected to the lower 20 bits of the main bus. The following modes are set by bits in the SR:

- Serial I/O interface operation mode.
- Data memory bus transfer format for operation in slave mode.
- Maskable interrupt status.

The bits corresponding to the above functions can be either read from or written to. The remaining bits in the SR are read-only bits which are used to check the following internal conditions:

- Master/slave mode.
- Serial I/O transfer error status.
- Parallel I/O transfer error status.

The SR bit configuration is shown in Figure 3-21, and the functions of the individual bits are described in Table 3-7.

Figure 3-21. Status register bit configuration.

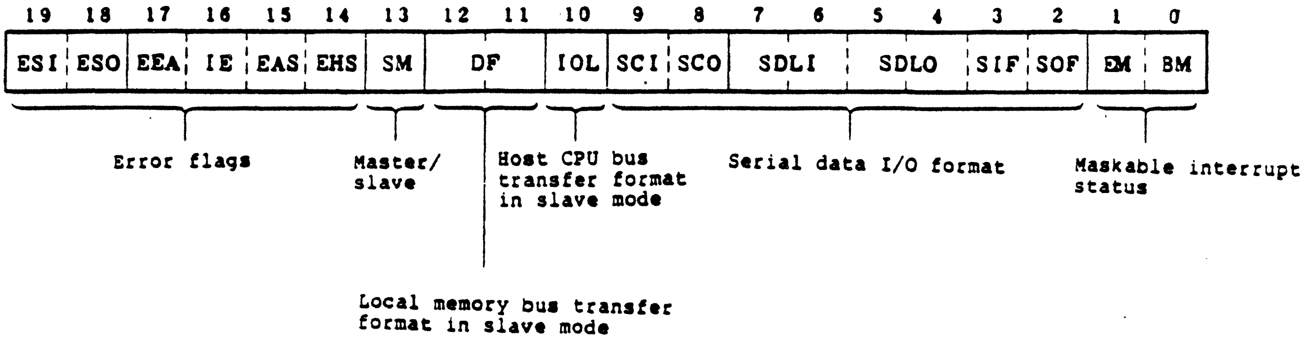


Table 3-7. Status register bit functions.

Bit name	Function
SCI (serial clock for input)	Specifies whether internal or external clock is used as the serial input clock (SICK) SCI 0 Internal clock 1 External clock
SCO (serial clock for output)	Specifies whether internal or external clock is used as the serial output clock (SOCK) SCO 0 Internal clock 1 External clock
SDLI (serial data length for input)	Specifies serial input data length SDLI 7 6 0 0 32 bits 0 1 8 bits 1 0 16 bits 1 1 24 bits
SDLO (serial data length for output)	Specifies serial output data length SDLO 5 4 0 0 32 bits 0 1 8 bits 1 0 16 bits 1 1 24 bits
SIF (serial data input format)	Specifies serial data input order SIF 0 MSB in first 1 LSB in first

Table 3-7. Status register bit functions (cont'd).

<table>
<tr><th>Bit name</th><th>Function</th></tr>
<tr><td>SOF
(serial data output format)</td><td>Specifies serial output data order
<table>
<tr><th>SOF</th><th></th></tr>
<tr><td>0</td><td>MSB out first</td></tr>
<tr><td>1</td><td>LSB out first</td></tr>
</table></td></tr>
<tr><td>EM
(enable maskable interrupt)</td><td>Enable/disable maskable interrupt
<table>
<tr><th>EM</th><th></th></tr>
<tr><td>0</td><td>disable</td></tr>
<tr><td>1</td><td>enable</td></tr>
</table></td></tr>
<tr><td>BM
(booked maskable interrupt)</td><td>Specifies whether INTM is booked (memorized)
<table>
<tr><th>BM</th><th></th></tr>
<tr><td>0</td><td>reset</td></tr>
<tr><td>1</td><td>memorize</td></tr>
</table></td></tr>
<tr><td>IOL
(I/O bus data length for slave)</td><td>Specifies data transfer format for host CPU bus in slave mode
<table>
<tr><th>IOL</th><th></th></tr>
<tr><td>0</td><td>32 bits</td></tr>
<tr><td>1</td><td>16 bits</td></tr>
</table></td></tr>
<tr><td>DF
(local bus data format for slave)</td><td>Specifies data transfer format for local bus in slave mode
<table>
<tr><th colspan="2">DF</th><th></th></tr>
<tr><td>12</td><td>11</td><td></td></tr>
<tr><td>0</td><td>0</td><td>4 bytes (32 bits)</td></tr>
<tr><td>0</td><td>1</td><td>1 byte (8 bits)</td></tr>
<tr><td>1</td><td>0</td><td>2 bytes (16 bits)</td></tr>
<tr><td>1</td><td>1</td><td>3 bytes (24 bits)</td></tr>
</table></td></tr>
<tr><td>SM
(slave/ master mode)</td><td>Indicates master/slave mode
<table>
<tr><th>SM</th><th></th></tr>
<tr><td>0</td><td>master mode</td></tr>
<tr><td>1</td><td>slave mode</td></tr>
</table></td></tr>
</table>

Table 3-7. Status register bit functions (cont'd).

<table>
<tr><th>Bit name</th><th>Function</th></tr>
<tr><td>ESI
(error status for serial input)</td><td>Indicates that serial data input was aborted with data length shorter than specified by the SDLI bits
<table><tr><th>ESI</th><th></th></tr><tr><td>0</td><td>no error</td></tr><tr><td>1</td><td>error occurred</td></tr></table></td></tr>
<tr><td>ESO
(error status for serial output)</td><td>Indicates that serial data output was aborted with data length shorter than specified by the SDLO bits
<table><tr><th>ESO</th><th></th></tr><tr><td>0</td><td>no error</td></tr><tr><td>1</td><td>error occurred</td></tr></table></td></tr>
<tr><td>EHS</td><td>Indicates external instruction access was specified during a read/write of external memory
<table><tr><th>EHS</th><th></th></tr><tr><td>0</td><td>no error</td></tr><tr><td>1</td><td>error occurred</td></tr></table></td></tr>
<tr><td>EAS</td><td>Indicates that read/write of low speed external memory was specified before current read/write is completed
<table><tr><th>EAS</th><th></th></tr><tr><td>0</td><td>no error</td></tr><tr><td>1</td><td>error occurred</td></tr></table></td></tr>
<tr><td>EEA</td><td>Indicates address register contents were changed during read/write of low speed external memory
<table><tr><th>EEA</th><th></th></tr><tr><td>0</td><td>no error</td></tr><tr><td>1</td><td>error occurred</td></tr></table></td></tr>
</table>

Table 3-7. Status register bit functions (cont'd).

<table>
<tr><th>Bit name</th><th>Function</th></tr>
<tr><td>IE</td><td>Indicates external instruction ROM access attempted when in slave mode
<table>
<tr><th>IE</th><th></th></tr>
<tr><td>0</td><td>no error</td></tr>
<tr><td>1</td><td>error occurred</td></tr>
</table>
</td></tr>
</table>

3.7 Serial I/O Interface.

3.7.1 Serial input interface configuration.

The serial input interface consists of the following circuits: the serial input shift register (ISFT), which converts the serial input bit stream into parallel data; the serial input register (SI), which holds the data generated by the ISFT and can be accessed from the main bus; and the serial input controller (SICNT), which regulates the serial input circuitry.

3.7.2 Serial input data format.

The serial input data format is specified by several bits in the status register. In particular, the two SR bits labelled SDLI determine the length of the serial input word, which may be 8, 16, 24, or 32 bits wide (as shown in Fig. 3-22), and the SIF bit specifies the data input order as either MSB in first or LSB in first. The serial input data format settings written to the status register take effect upon the next transition of SIEN\ from high to low.

Figure 3-22. Serial input register data format for SIF=0 (MSB in first).

```
               31        23        15         7         0
               ----------------------------------------
8 bit mode     |<-- 0 -->|////1////|<------- 0 ------->|
               ----------------------------------------
                         ^ MSB

               31        23        15         7         0
               ----------------------------------------
16 bit mode    |<-- 0 -->|////1////|////2////|<-- 0 -->|
               ----------------------------------------
                         ^ MSB

               31        23        15         7         0
               ----------------------------------------
24 bit mode    |<-- 0 -->|////1////|////2////|////3////|
               ----------------------------------------
                         ^ MSB

               31        23        15         7         0
               ----------------------------------------
32 bit mode    |////1////|////2////|////3////|////4////|
               ----------------------------------------
               ^ MSB
```

Shaded areas indicate bit storage locations for each mode; the number in the shaded areas indicates the order in which the bytes are filled.

3.7.3 Serial data input timing.

1) Serial data input is enabled by setting the SIEN\ input to 0.

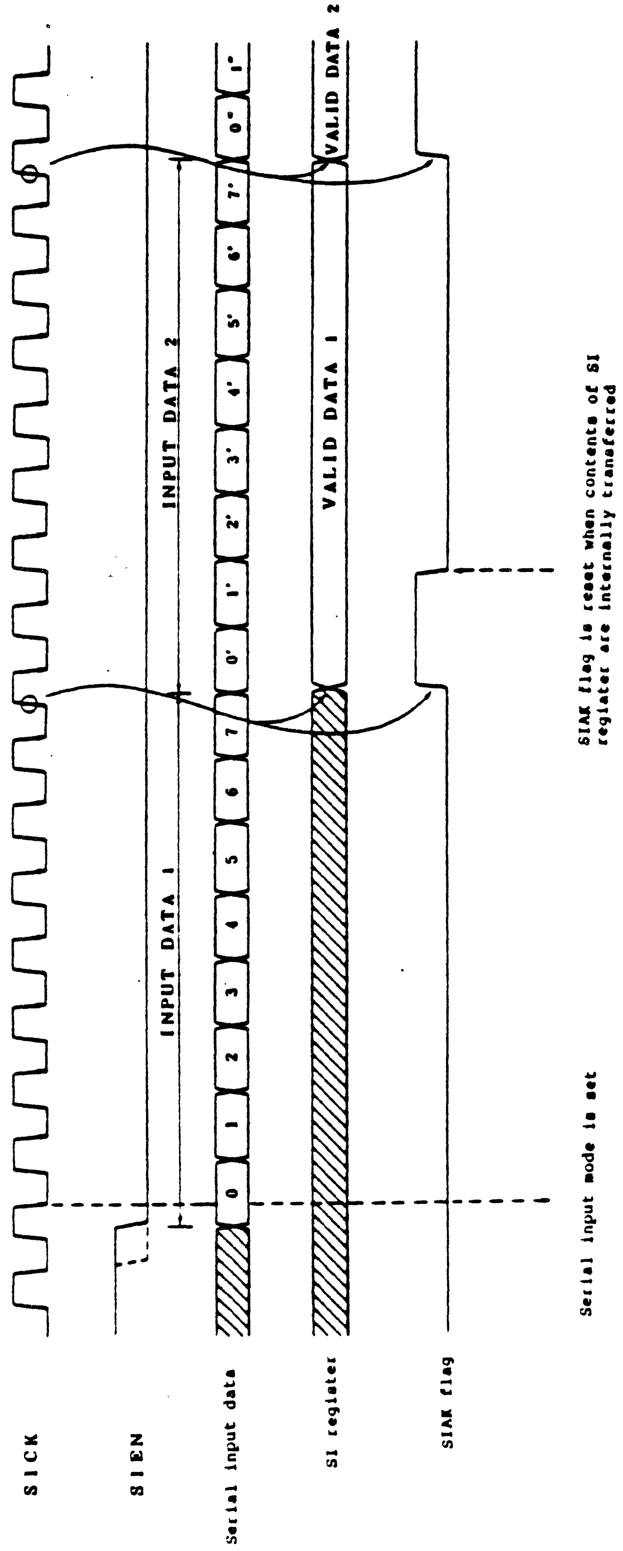
2) When SIEN\ is low, serial input data is clocked into ISFT on the falling edge of the serial input clock (SICK).

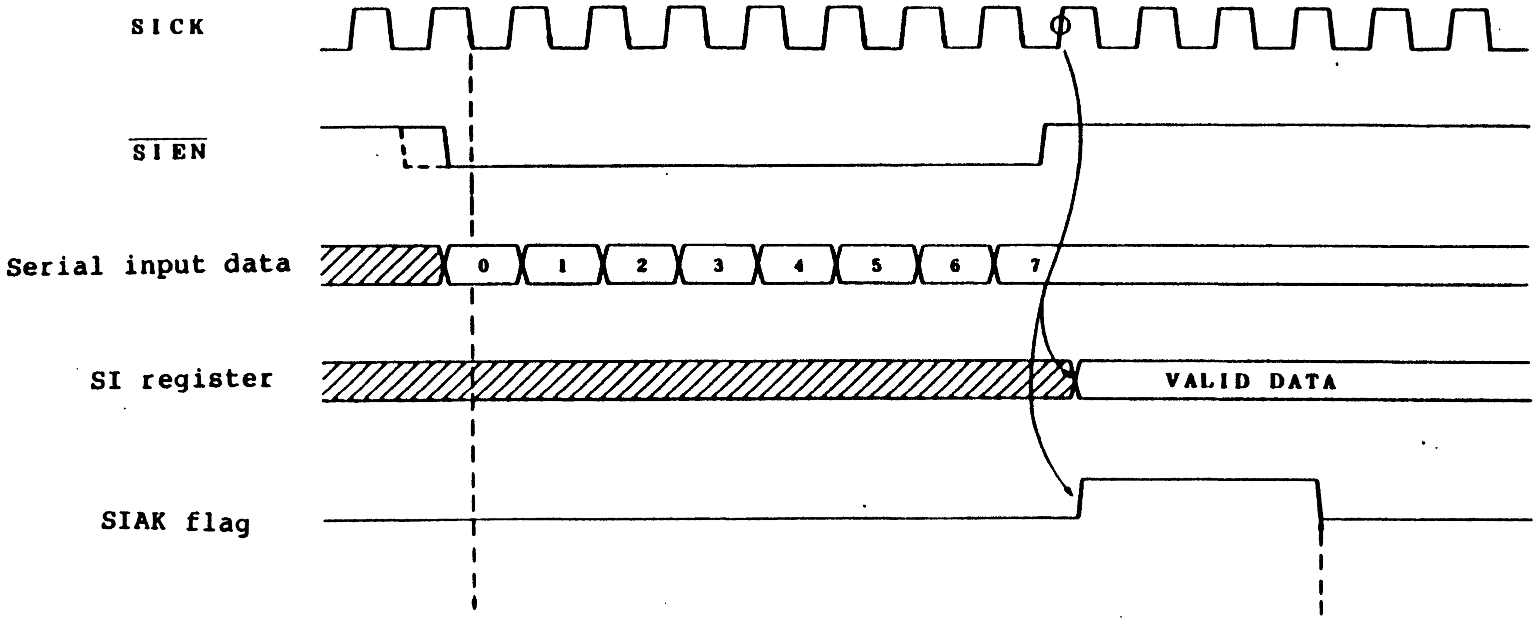
3) A serial data input word is considered complete when the number of bits specified by the SDLI bits of the status register have been loaded into ISFT. If serial data input is aborted by changing SIEN\ from 0 to 1 before the serial data input word is complete, then the serial input error flag (ESI) in the status register will be set.

4) When the serial data input word is complete, it is transferred from ISFT to SI, the serial input acknowledge flag (SIACK) is set, and ISFT is reset. Each of these actions is triggered by the rising edge of SICK immediately following the falling edge of SICK which completed the serial input word. This timing is depicted in Figures 3-23 and 3-24.

5) Serial data can be continuously input by leaving SIEN\ in its active low state. The bit latched into ISFT after the last bit of the previous word is handled as the first bit of the next word, as shown in Figure 3-23.

Figure 3-23. Serial input timing for successive 8-bit words.





Serial input mode is set

SIAK flag is reset when contents of SI register are internally transferred

Figure 3-24. Serial input timing for a single 8-bit word.

3.7.4 Internal transfers of serial input data.

1) The serial input register (SI) contents can be transferred to any register that can be specified in the DST field of an instruction. Such a transfer will not change the format of the SI register contents.

2) The data in the SI register will be overwritten by the contents of ISFT when the next serial input word is complete. To avoid loss of data, the SI register contents should be transferred to another location before the next serial input word is complete.

3) The serial input acknowledge flag (SIACK) goes high when data has been transferred from ISFT to SI. This flag can be tested by a conditional branch instruction, and ought to be used to avoid data loss as described in (2) above.

4) When data is read from the SI register, the SIACK flag is reset to 0.

3.7.5 Serial output interface configuration.

The serial output interface consists of the following circuits: the serial output register (SO), which holds the data sent from the main internal bus; the serial output shift register (OSFT), which converts the data in SO from parallel to serial form; and the serial output controller (SOCNT) which regulates the functioning of the serial output circuitry.

3.7.6 Serial output data format.

The serial output data format is specified by several bits in the status register. In particular, the two bits labelled SDLO determine the length of the serial output word, which may be 8, 16, 24, or 32 bits wide (as shown in Figure 3-25), and the SOF bit specifies the data output order as either MSB out first or LSB out first. The serial data output format settings written to the status register take effect upon the next transfer of data from the main bus to the serial output register (SO).

Figure 3-25. Serial output register data format for SOF=0 (MSB out first).

```
              31          23          15          7          0
              ---------------------------------------------
8 bit mode    |<-- 0 -->|////1////|<------- 0 ------->|
              ---------------------------------------------
                        ^ MSB

              31          23          15          7          0
              ---------------------------------------------
16 bit mode   |<-- 0 -->|////1////|////2////|<-- 0 -->|
              ---------------------------------------------
                        ^ MSB

              31          23          15          7          0
              ---------------------------------------------
24 bit mode   |<-- 0 -->|////1////|////2////|////3////|
              ---------------------------------------------
                        ^ MSB

              31          23          15          7          0
              ---------------------------------------------
32 bit mode   |////1////|////2////|////3////|////4////|
              ---------------------------------------------
              ^ MSB
```

Shaded areas indicate bit storage locations for each mode; the number in the shaded areas indicate the order in which the bytes are shifted out.

3.7.7 Serial output timing.

1) A serial output data transfer is initiated by writing the output data word to the SO register. This results in the serial output acknowledge flag (SOAK) going from low to high.

2) The data written to the SO register is transferred to the OSFT register as soon as the OSFT is finished shifting out the previously transferred data. When this transfer occurs, the serial output request (SORQ) goes high, telling the external device that data is ready for a serial transfer. In addition, the SOAK flag is reset, indicating that new data may be written into the SO register without over-writing the previous word.

   If a serial output data transfer is in progress when data is written to the SO register, then the transfer from SO to OSFT is not executed until the serial output data transfer is complete.

3) After the external device senses the serial output request (SORQ=1), it initiates the serial transfer by setting SOEN\ to 0.

4) Serial data is clocked out by the rising edge of the serial output clock (SOCK).

Serial output timing is depicted in Figures 3-26 and 3-27

Figure 3-26. Serial output timing for single 8-bit transfer.

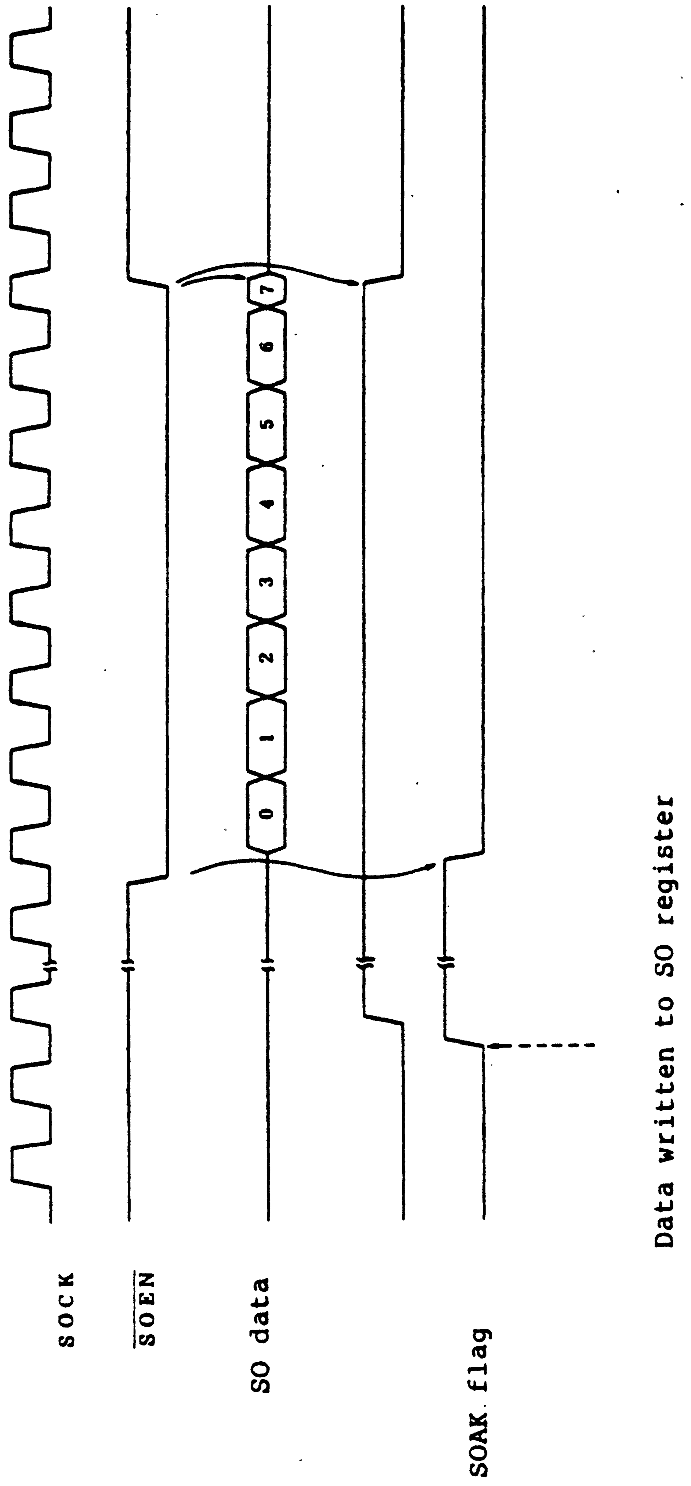
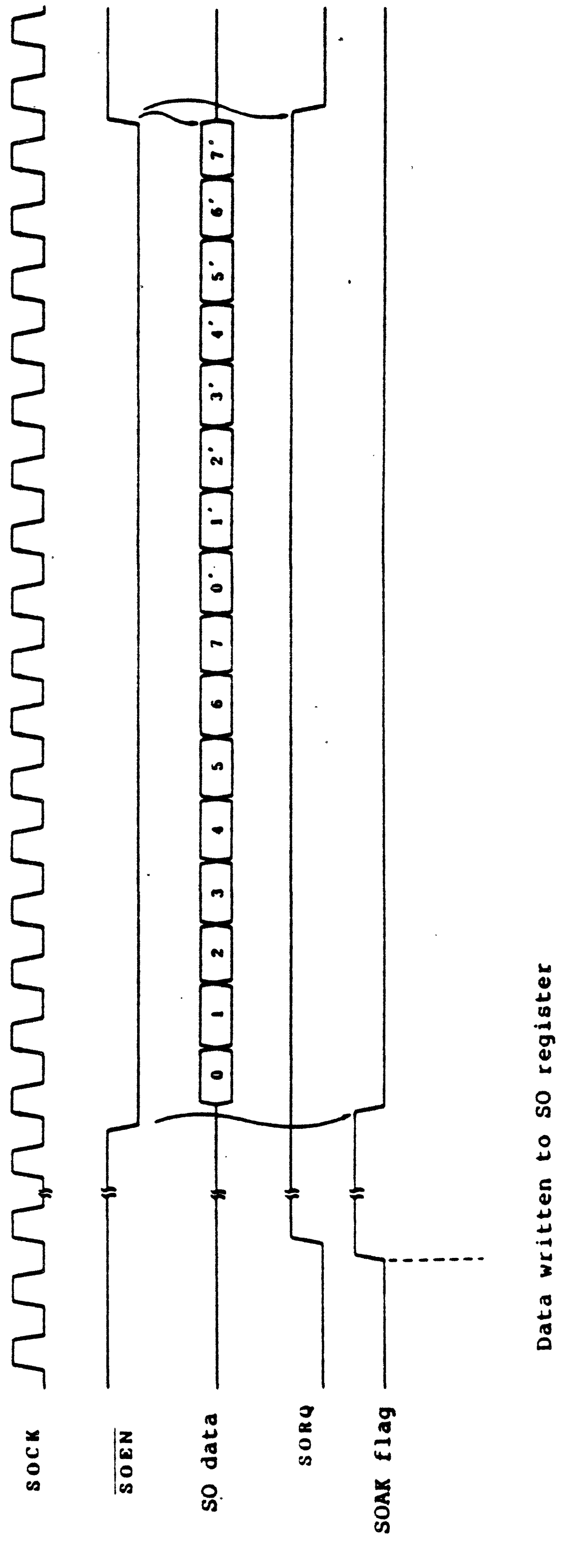


Figure 3-27. Serial output timing for successive 8-bit transfers.

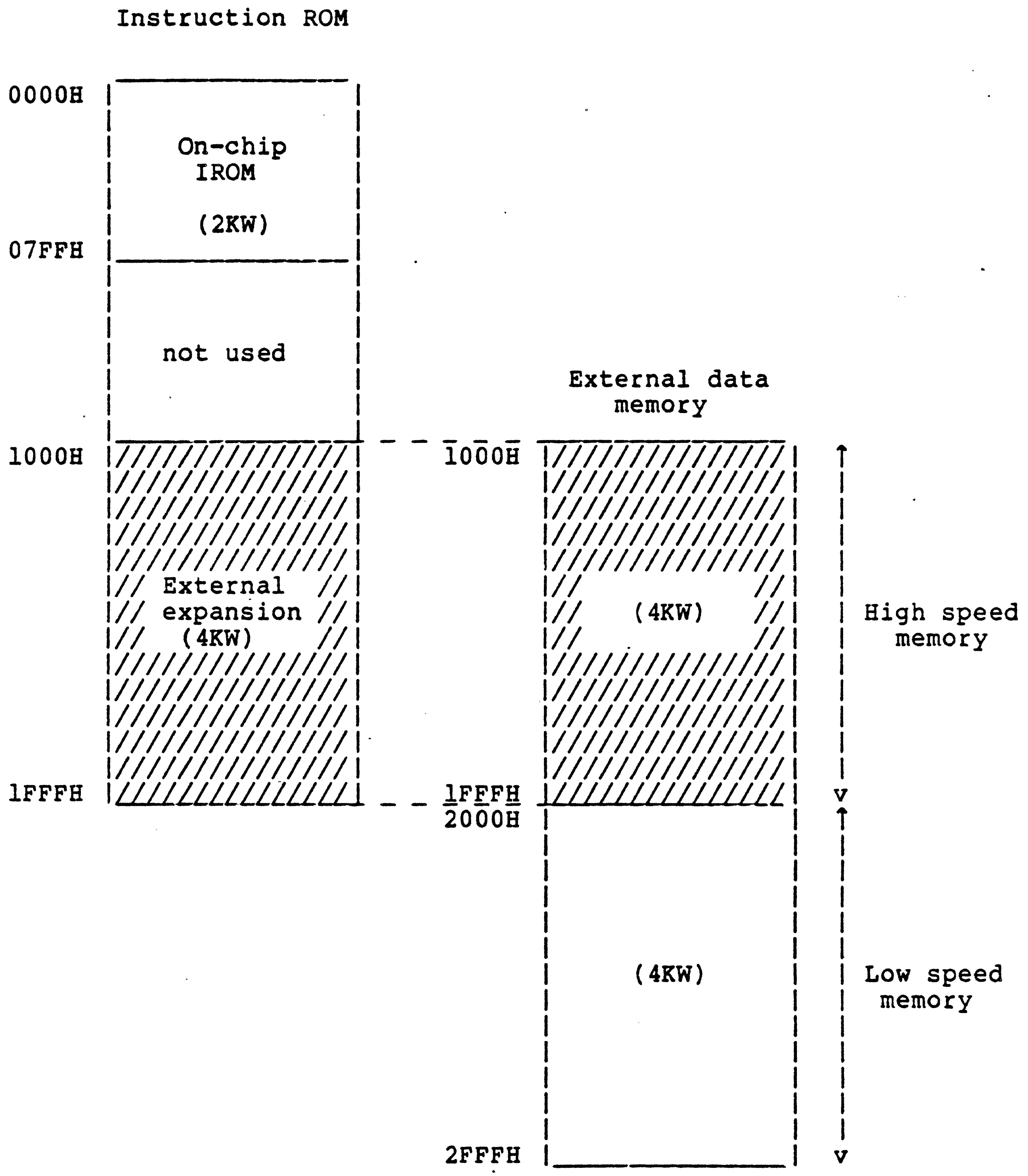


3.8 Parallel interface in Master mode.

3.8.1 Memory map.

In master mode, the uPD77230R is capable of handling 8K words of external memory expansion. This 8K word area consists of a 4K high speed memory area and a 4K low speed memory area, as shown in Figure 3-28. The high speed memory area (lower 4K of 8K expansion) can be accessed in a single instruction cycle, and may be used for either instruction or data storage. The low speed memory area (upper 4K of 8K expansion) can be accessed in three instruction cycles, and may be used for data only.

Figure 3-28. Master mode memory map.



3.8.2 External expansion of instruction ROM.

The uPD77230R can use a 4K word external expansion of instruction memory only in master mode. Only the high speed external memory area (i.e. the lower 4K) can be used for instruction storage. The most significant bit of the program counter (PC12) determines whether internal instruction ROM or external instruction memory is selected. To set and clear PC12, an appropriate 13-bit address must be specified as the destination address in a branch instruction. As this implies, the program counter will not automatically count from FFFH to 1000H; rather, it will wrap around from FFFH to 000H.

1) When PC12 = 0, internal instruction ROM is selected, and when PC12 = 1, external instruction memory is selected.

2) When PC12 becomes 1, the lower 12 bits of the PC (i.e. PC00 thru PC11) are output to the address port (AP), and the data port (DP) is set to input mode. The 32-bit instruction from external memory will be input via the DP and latched into the instruction register (IR) for decoding.

A typical instruction memory expansion in master mode is shown in Figure 3-29. Figure 3-30 shows the external instruction memory access timing.

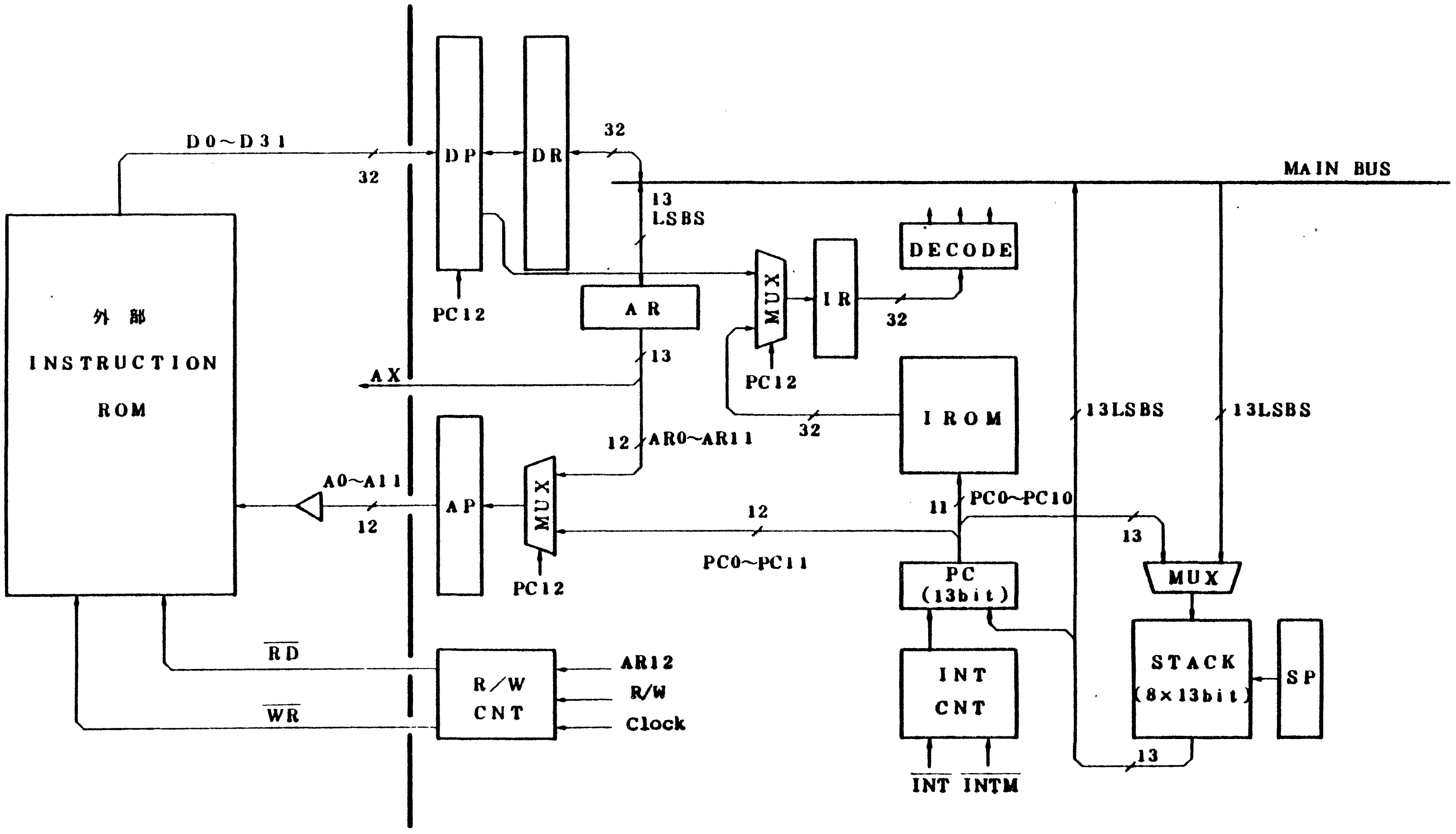


Figure 3-29. Instruction ROM expansion in master mode.

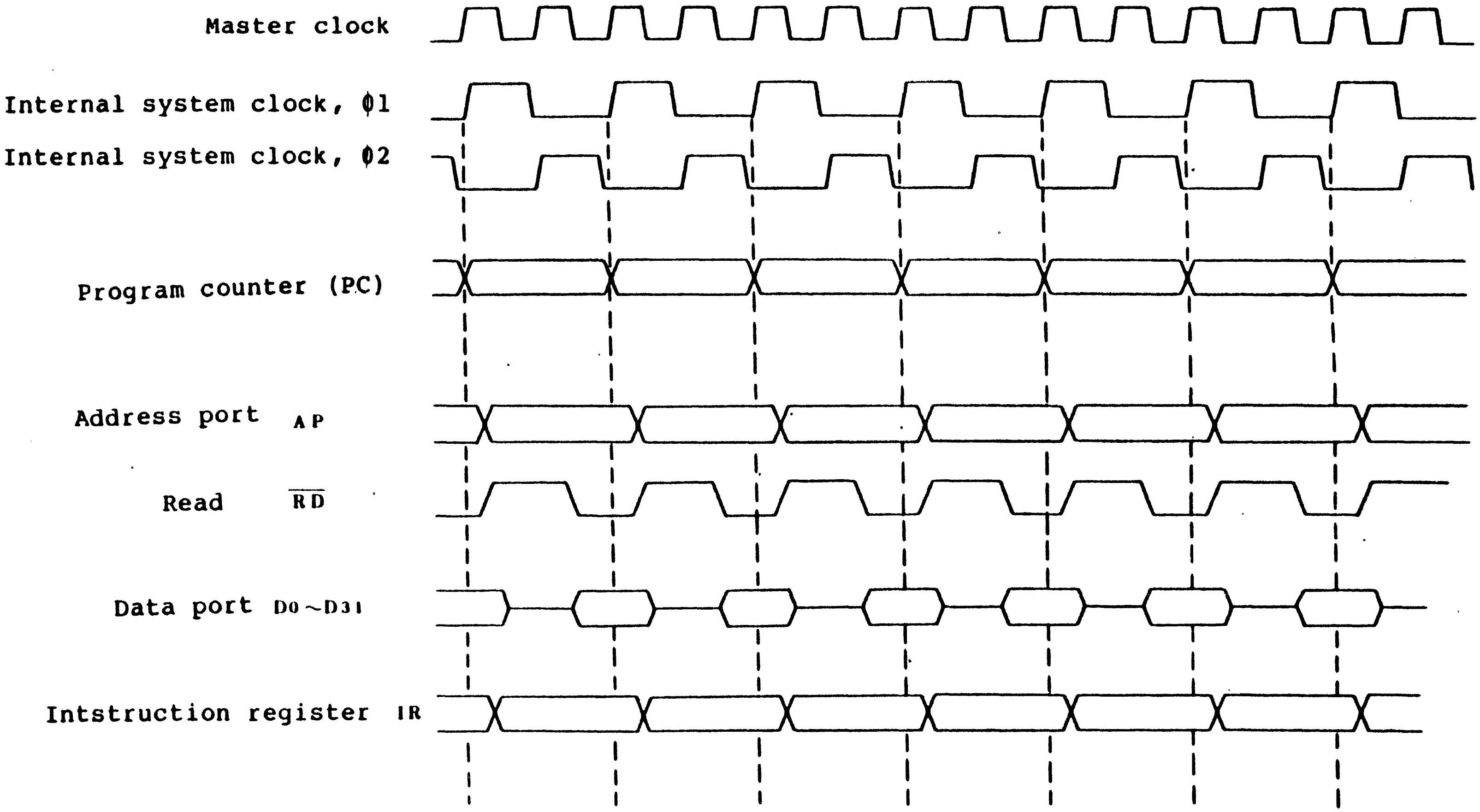


Figure 3-30. External instruction memory access timing.

3.8.3 External data memory expansion.

When instruction ROM is being externaly expanded in master mode, a 4K word data memory expansion is ALSO possible, using the low speed memory area (i.e. upper 4K of 8K expansion). If instruction ROM is not being externally expanded, then a total of 8K words of data memory expansion may be used, with 4K words in the low speed memory area and 4K words in the high speed memory area. Figure 3-31 depicts external data memory expansion in master mode.

1) Specification of a read from or a write to external data memory is done via the RW̄ bits in the CNT field of an OP instruction.

2) When accessing external data memory, the lower 12 bits of the address register (AR) are output to external memory through the address port (AP). The MSB of the address is output on the A_X pin.

3) The access time for the high speed external memory area is one instruction cycle, as shown in Figures 3-32 and 3-33.

4) The access time for the low speed external memory area is three instruction cycles, as shown in Figures 3-34 and 3-35.

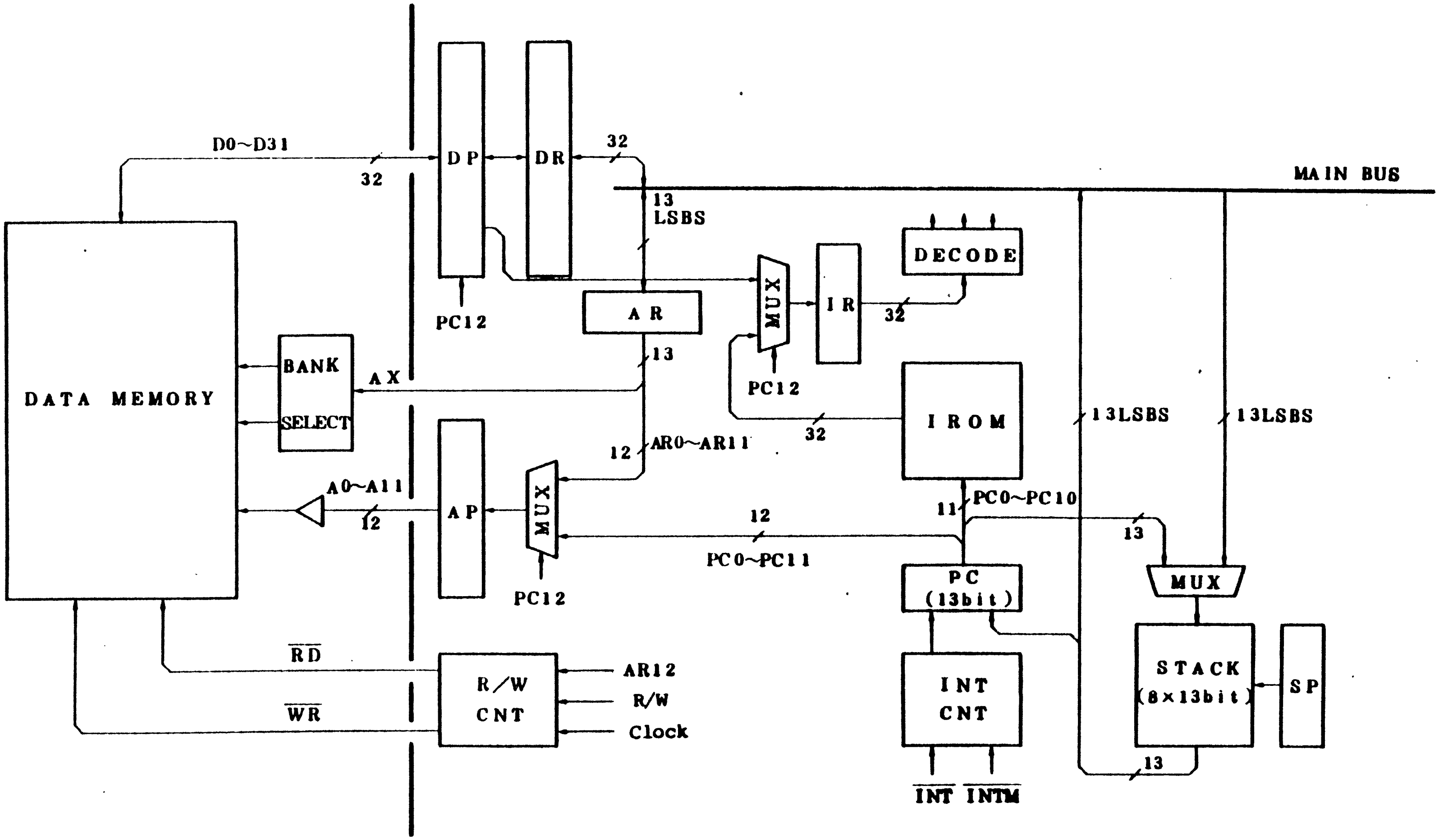


Figure 3-31. External data memory expansion in master mode.

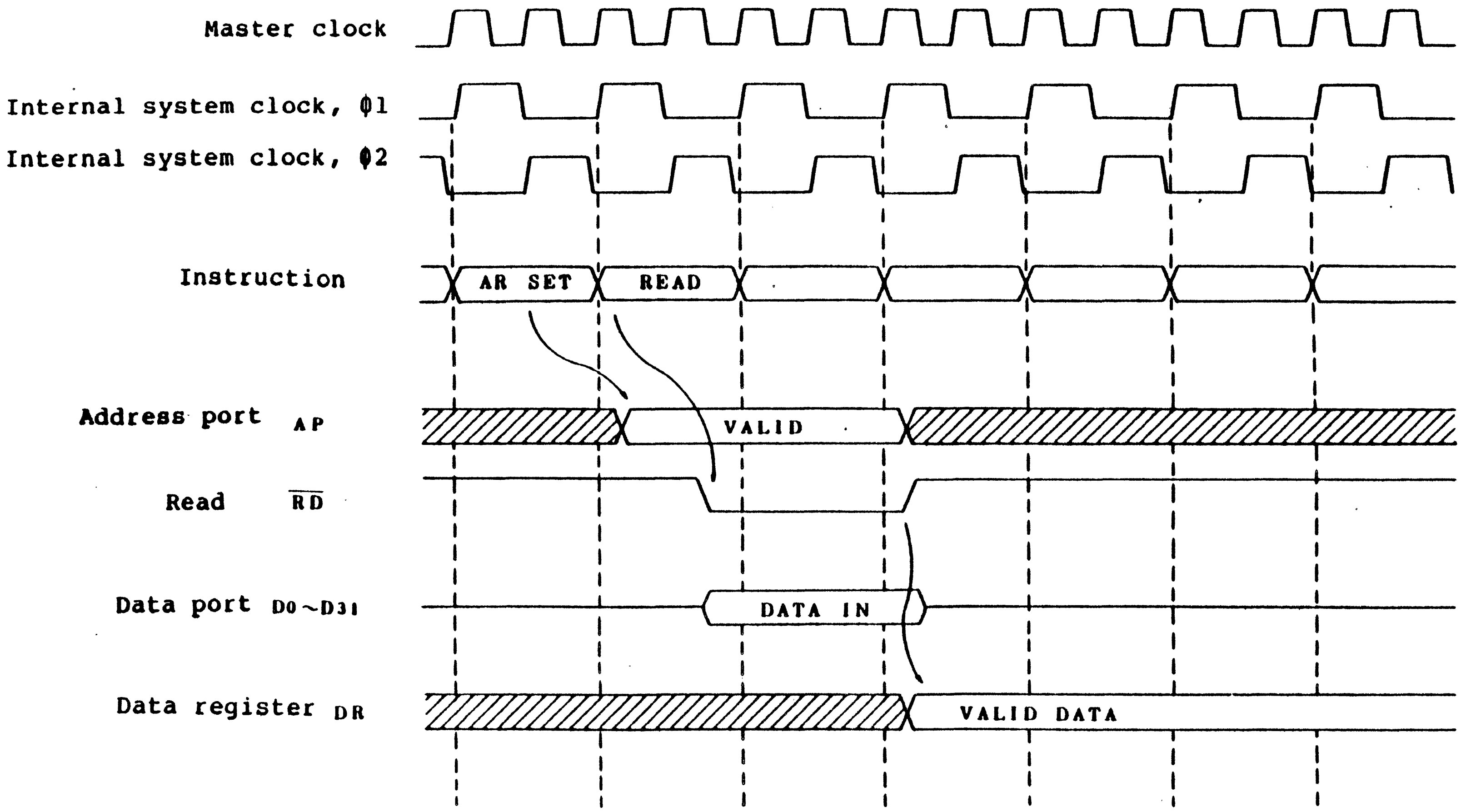


Figure 3-32. Read timing in high speed external data memory (master mode).

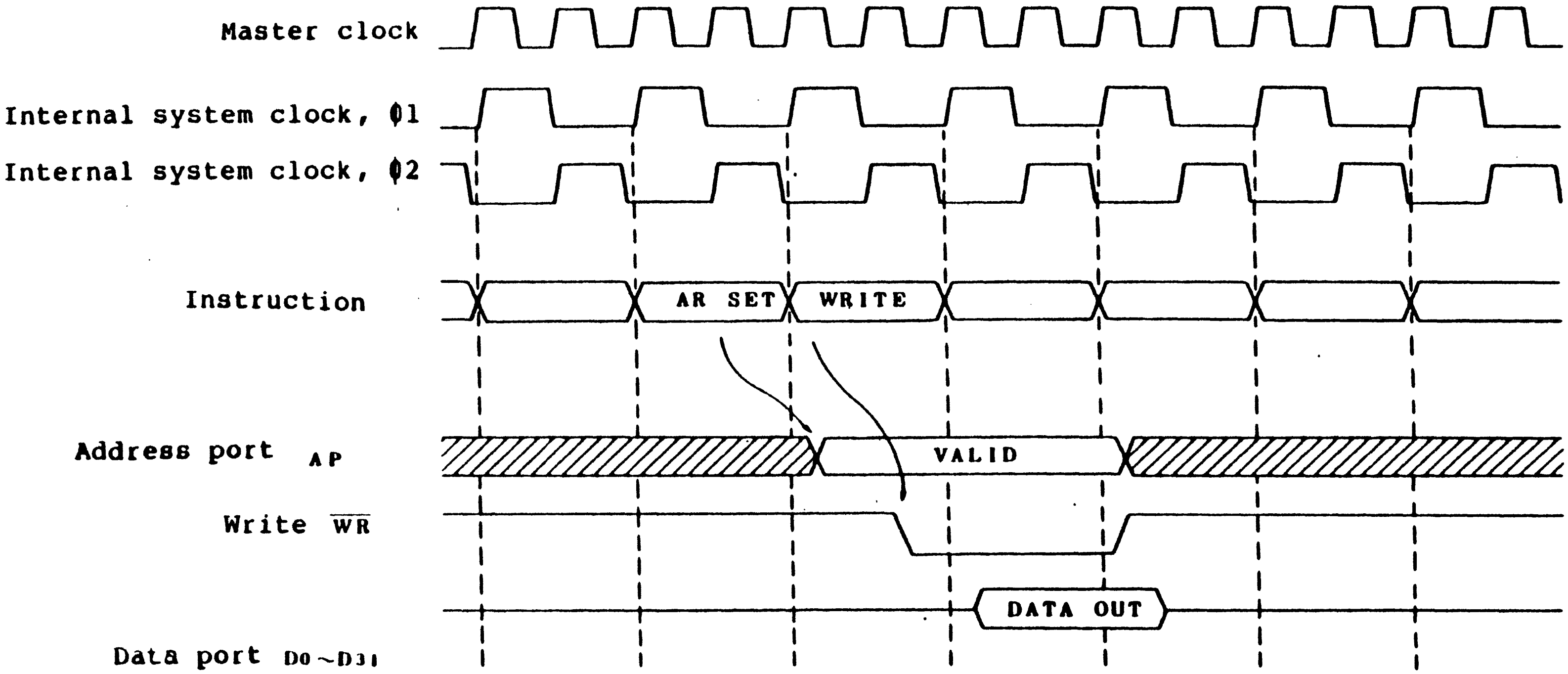


Figure 3-33. Write timing in high speed external data memory (master mode).

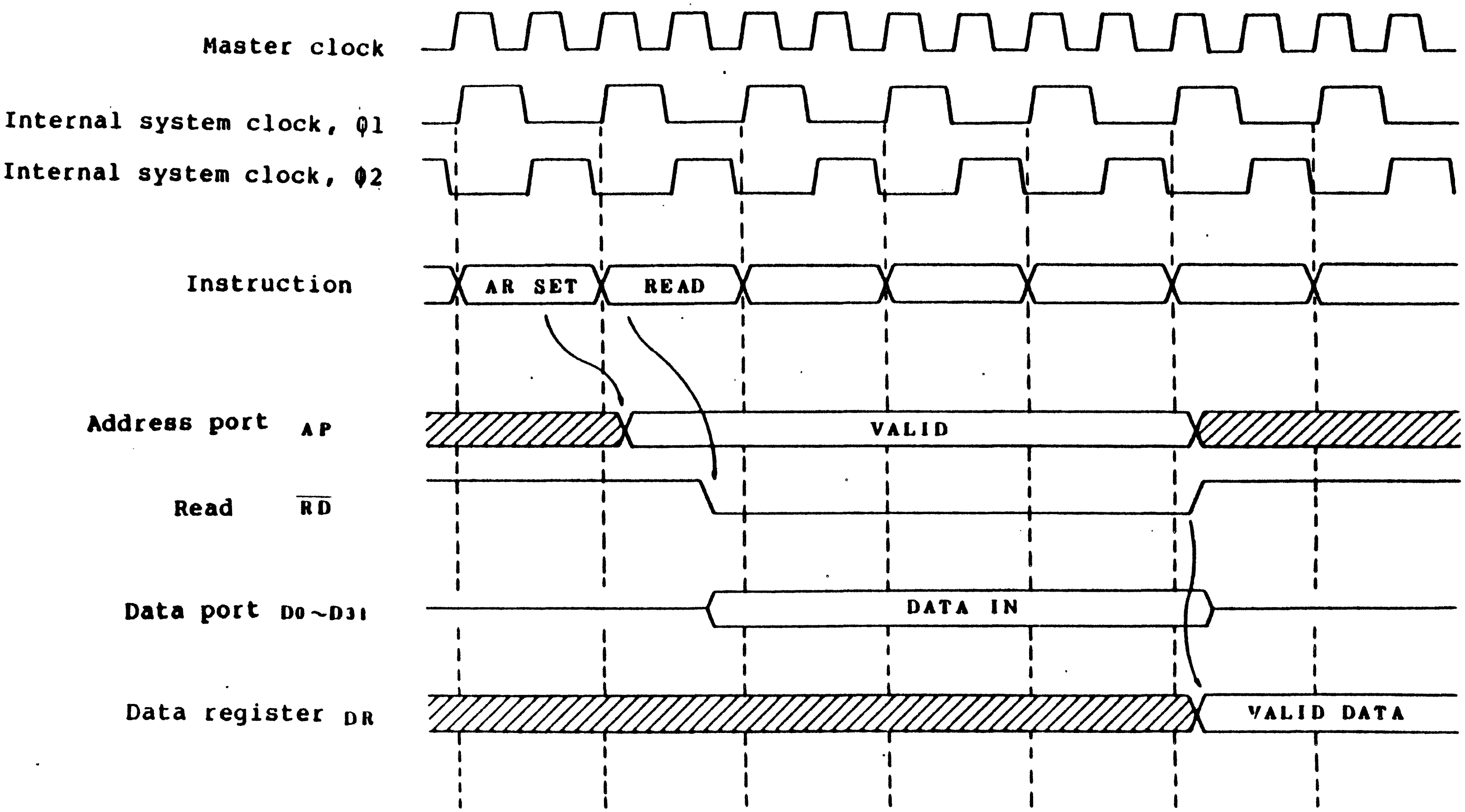


Figure 3-34. Read timing in low speed external data memory (master mode).

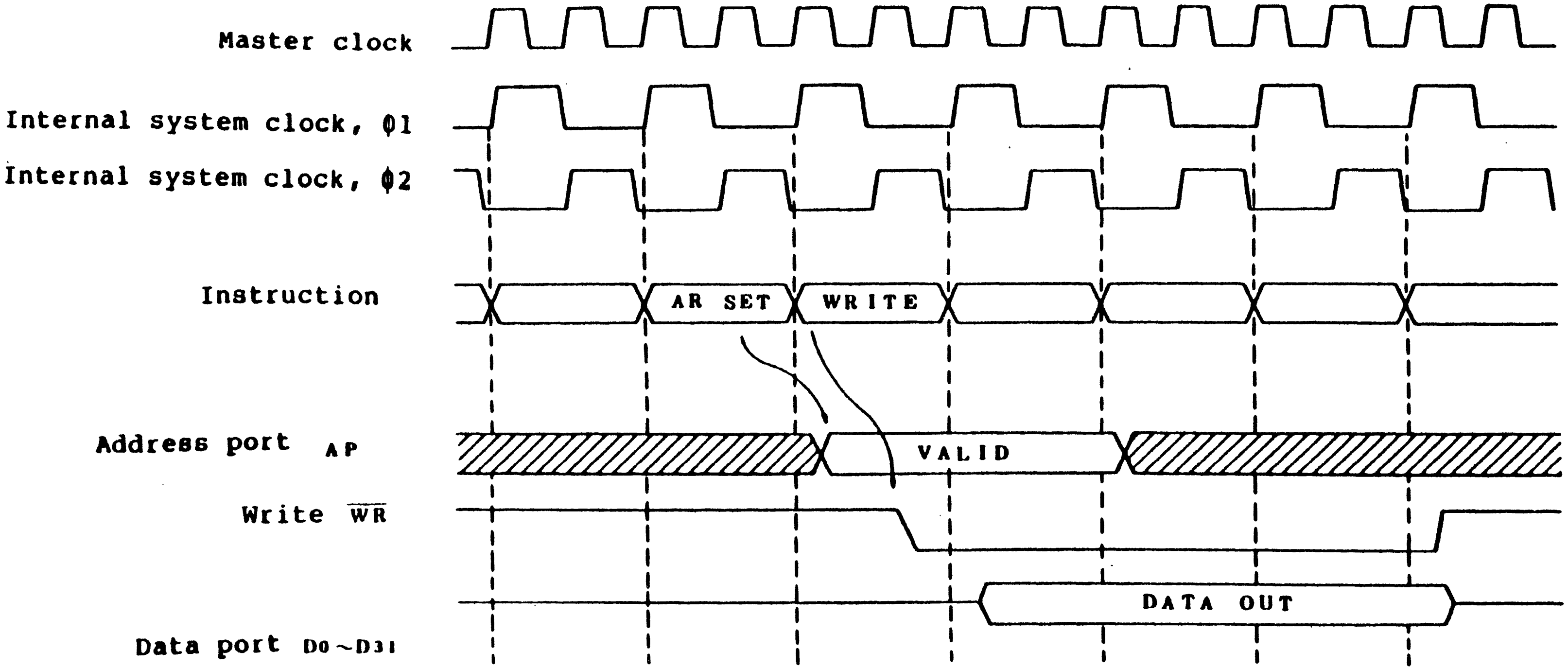


Figure 3-35. Write timing in low speed external data memory (master mode).

3.9 Parallel interface in slave mode.

3.9.1 Slave mode.

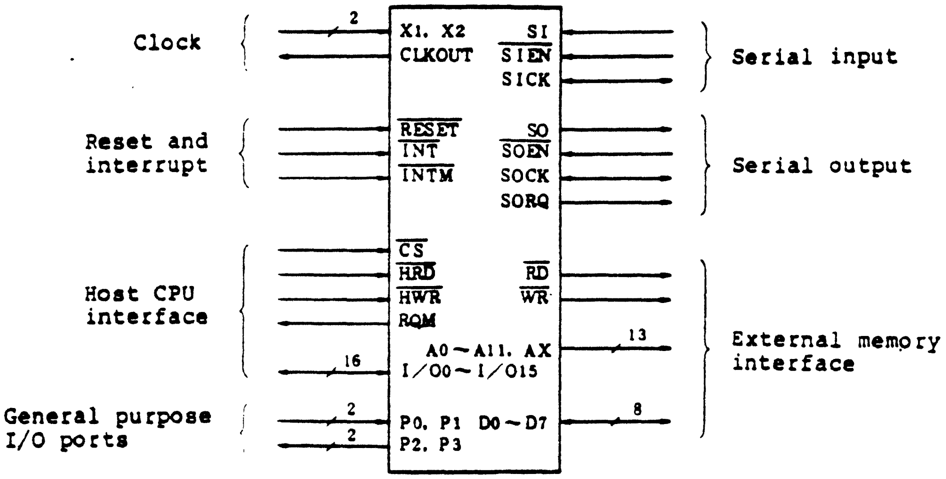
The uPD77230R can be configured for slave mode operation by setting the mode set pin (M/S) to a high level. In slave mode, the pin configuration for the parallel interface to external devices is shown in Table 3-8 and Figue 3-36.

Function	Pin name	
Host CPU interface	I/O 0 to I/O 15 HRD\ HWR\ RQM CS\	(data bus) (read) (write) (request) (chip select)
External data memory interface	D0 to D7 A0 to A11,A_x RD\ WR\	(data bus) (addr. bus) (read) (write)
General purpose input port	P0 P1	
General purpose output port	P2 P3	

NOTE: The status of the mode set pin (M/S) must be established prior to power-up of the device. Device behavior cannot be guaranteed if the setting of the M/S pin is changed after power-up.

Table 3-8. Pin configuration in slave mode.

Figure 3-36. Pin configuration in slave mode.



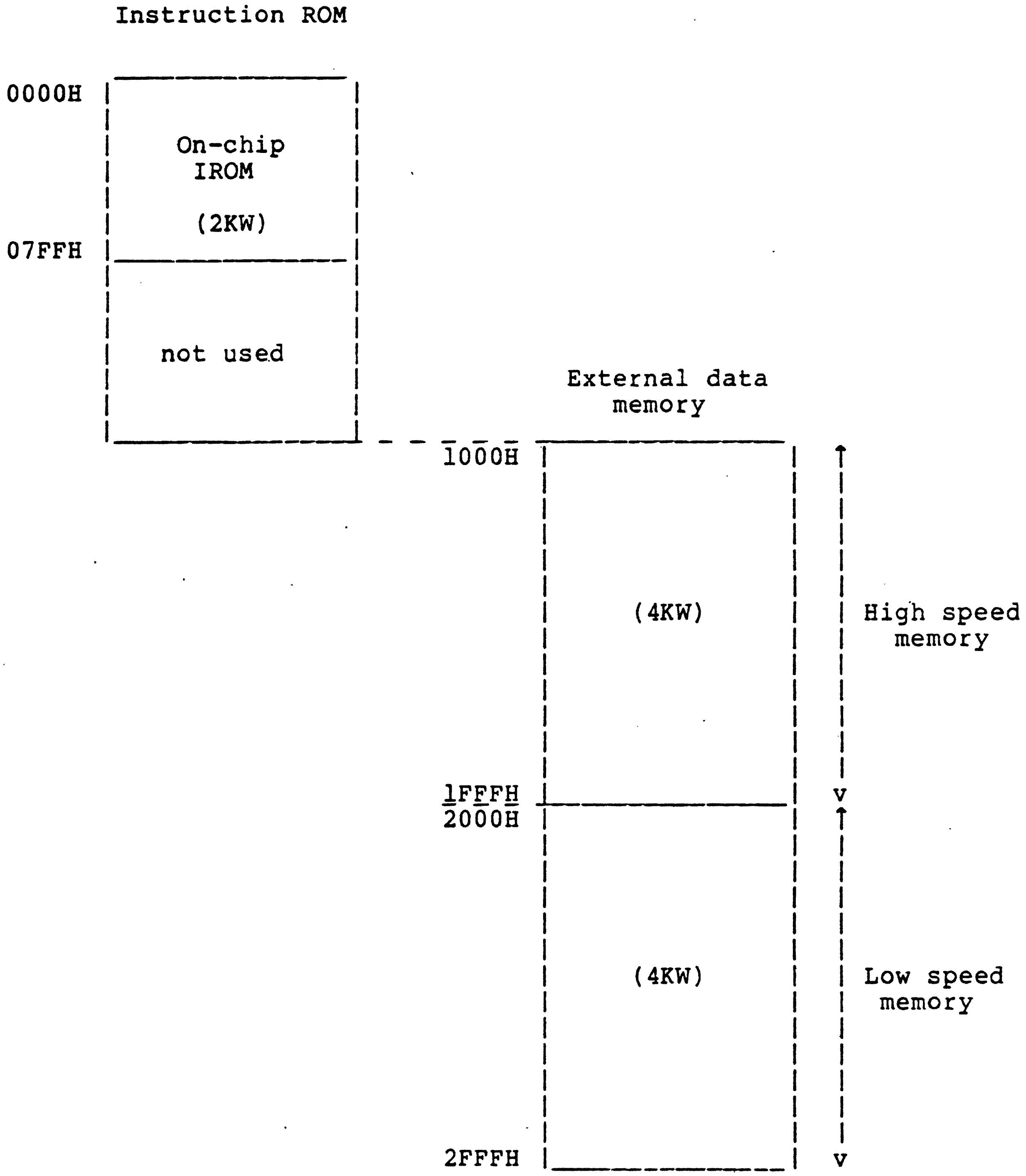
3.9.2 Memory map in slave mode.

In slave mode, it is possible to add up to 8K words of external data memory. The first 4K words reside in the high speed memory area, and the second 4K words reside in the low speed memory area, as shown in Figure 3-37.

The interface to the external data memory is made through an 8-bit bus comprised of pins D0 thru D7. The data transfer format is specified by the DF bits of the status register (SR), which allows selection of 8, 16, 24, or 32-bit word lengths.

Note that the instruction ROM cannot be externally expanded in slave mode.

Figure 3-37. Memory map in slave mode.



3.9.3 External data memory expansion in slave mode.

Figure 3-38 shows an example of data memory expansion in slave mode. The data memory interface is an 8-bit bus comprised of pins D0 thru D7.

1) The read/write specification for the external data memory is made by the RW bits of the CNT field of an OP instruction.

2) When access to the external data memory is specified, the lower 12 bits of the address register (AR) are output through the address port (AP), and the MSB of AR is output through the A_x pin to the external memory.

3) Data transfer format is specified by the DF bits of the status register (SR). Word lengths of 8, 16, 24, or 32 bits may be chosen. These word lengths are assembled in the data register (DR) from the 8-bit pieces brought in through the data bus. Figure 3-39 shows the data format for the various wordlengths. The timing of data transfers is shown in Figures 3-40 thru 3-43.

Figure 3-38. Data memory expansion in slave mode.

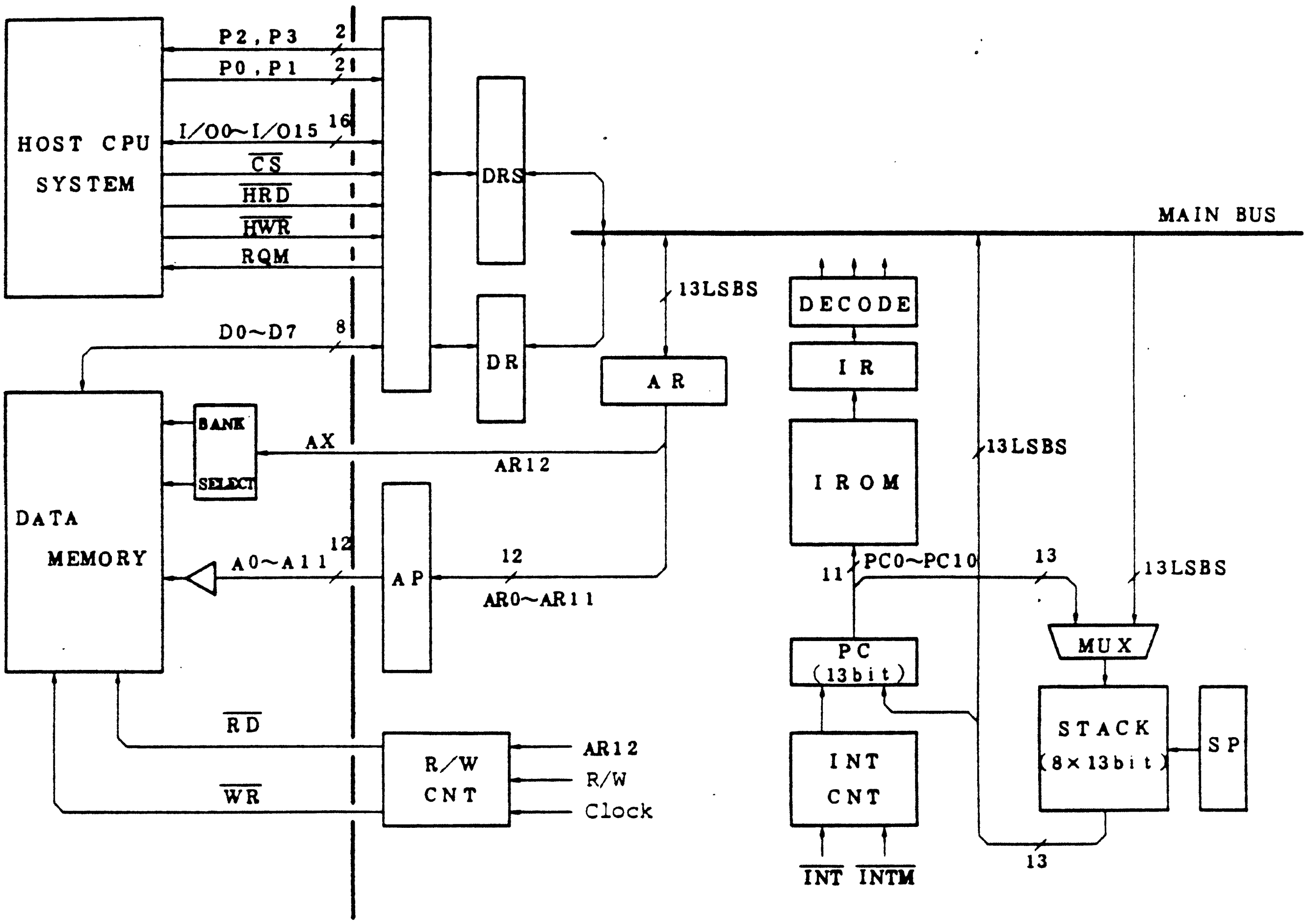


Figure 3-39. External data memory transfer formats in the data register.

```
               31         23        15        7        0
                ---------------------------------------
8 bit mode     |<-- 0 -->|////1////|<------- 0 ------->|
                ---------------------------------------
                         ^ MSB

               31         23        15        7        0
                ---------------------------------------
16 bit mode    |<-- 0 -->|////2////|////1////|<-- 0 -->|
                ---------------------------------------
                         ^ MSB

               31         23        15        7        0
                ---------------------------------------
24 bit mode    |<-- 0 -->|////3////|////2////|////1////|
                ---------------------------------------
                         ^ MSB

               31         23        15        7        0
                ---------------------------------------
32 bit mode    |////4////|////3////|////2////|////1////|
                ---------------------------------------
               ^ MSB
```

Shaded areas indicate bit storage locations for each mode; the number in the shaded areas indicate the order in which the bytes are filled.

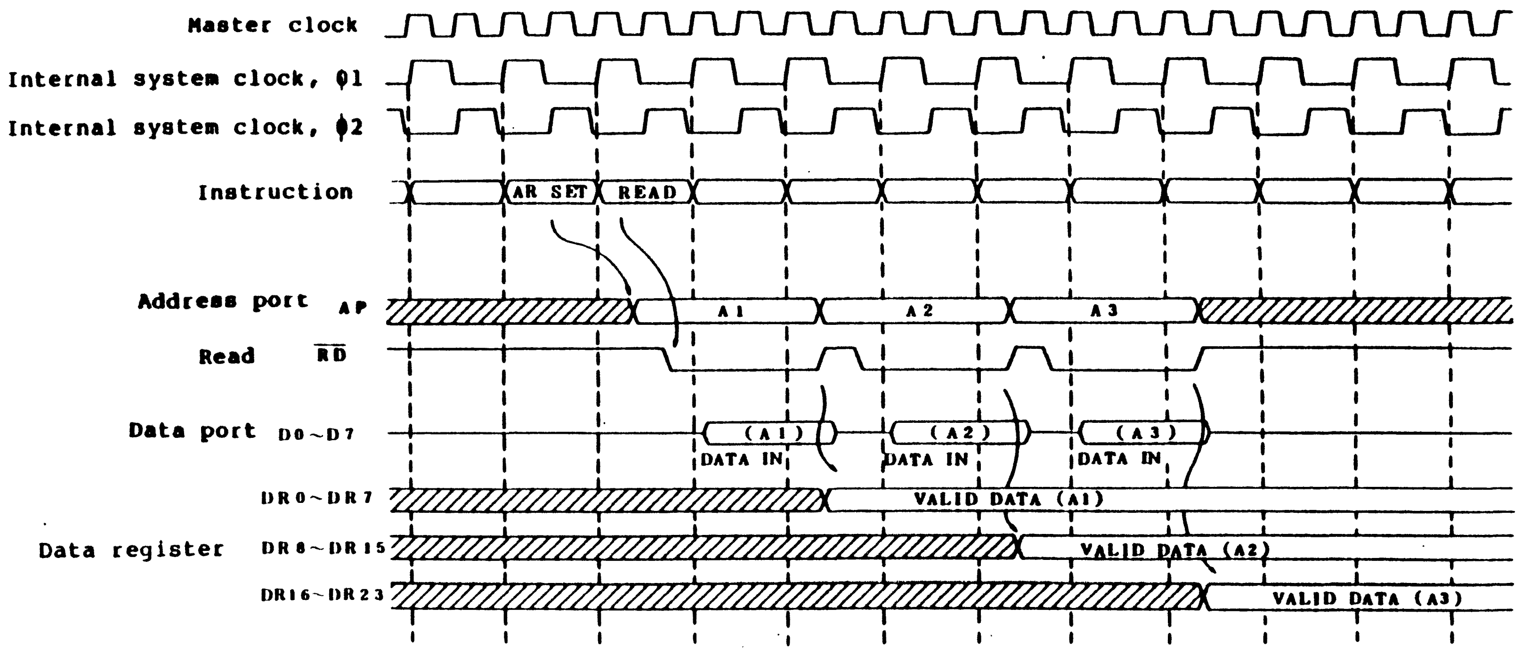


Figure 3-40. External data memory read timing for 24-bit transfer in slave mode.

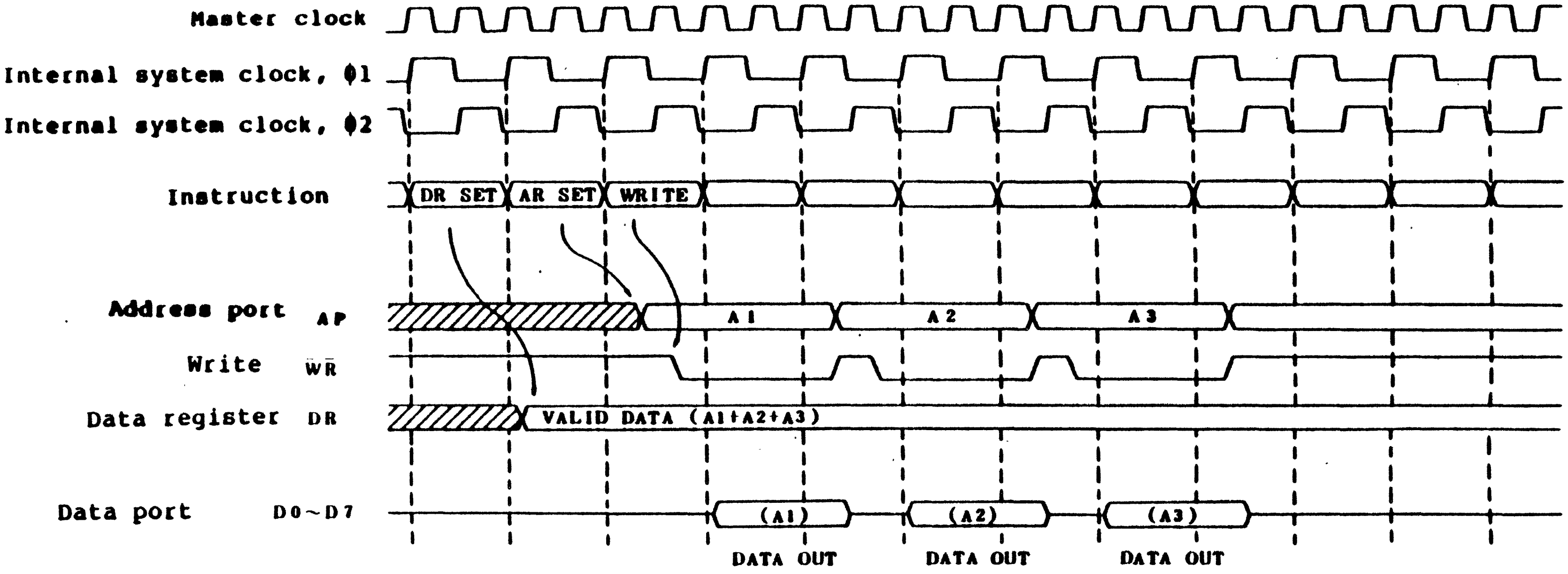


Figure 3-41. External data memory write timing for 24-bit transfer in slave mode.

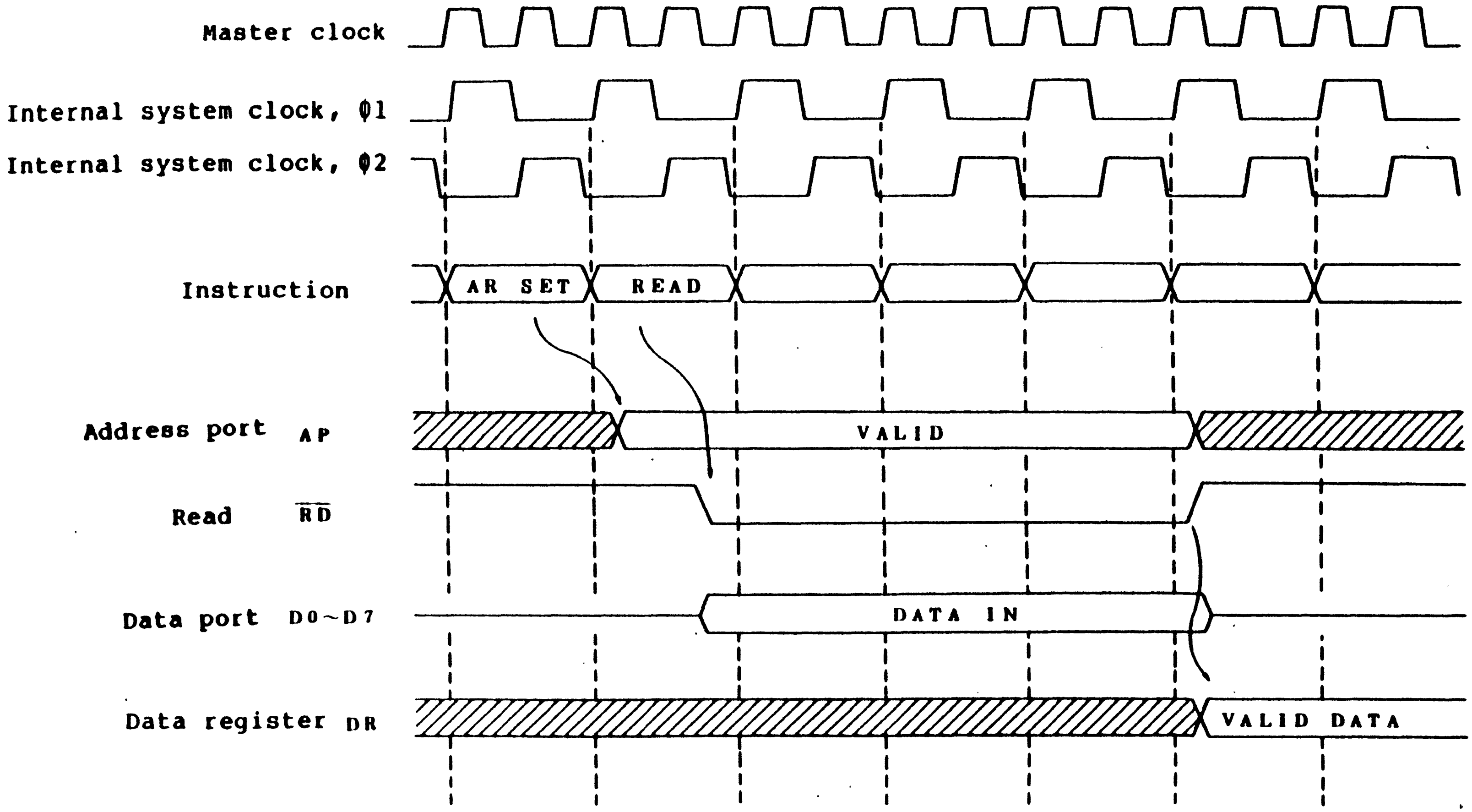


Figure 3-42. External low speed data memory read timing in slave mode.

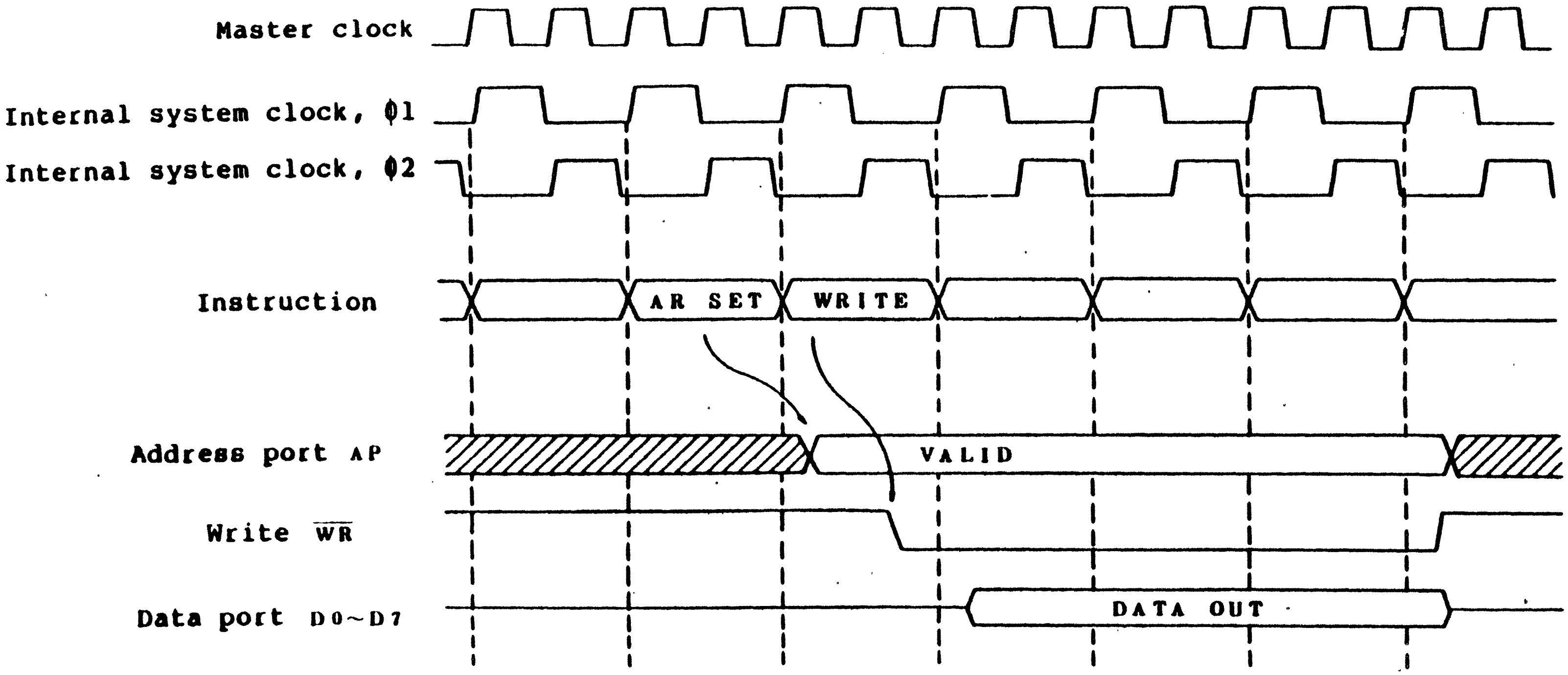


Figure 3-43. External low speed data memory write timing in slave mode.

3.9.4 Interfacing with the host CPU in slave mode.

In slave mode, data transfer between the uPD77230R and the host CPU is done over a 16-bit bus comprised of pins I/O 0 thru I/O 15. The data transfer format is specified by the IOL bit of the status register (SR), which selects either a 16-bit or a 32-bit word length. The data transfer formats are shown in Figure 3-44. Data transfer timing is shown in Figures 3-45 and 3-46.

Figure 3-44. Formats for data transfers to and from host CPU.

```
                  31         23         15         7         0
                  ----------------------------------------
16 bit mode       |<------- 0 ------->|//////////1//////////|
                  ----------------------------------------

                  31         23         15         7         0
                  ----------------------------------------
32 bit mode       |//////////2//////////|//////////1//////////|
                  ----------------------------------------
```

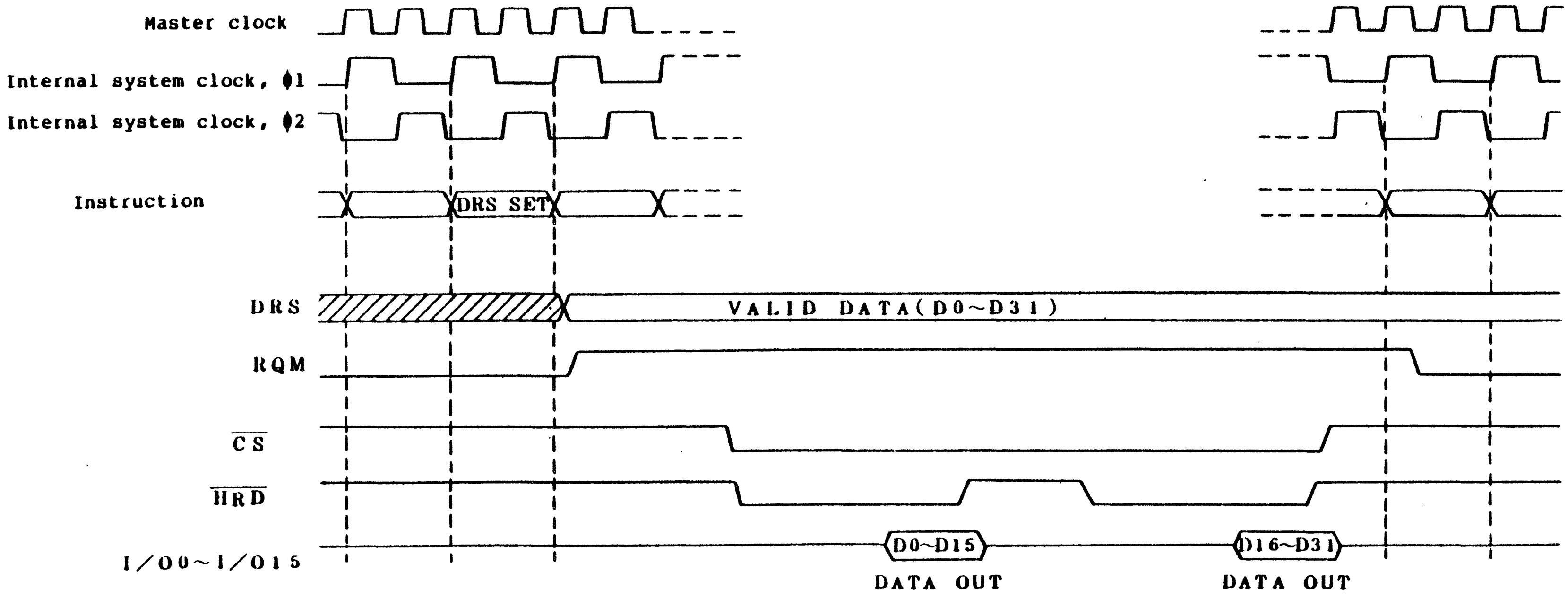
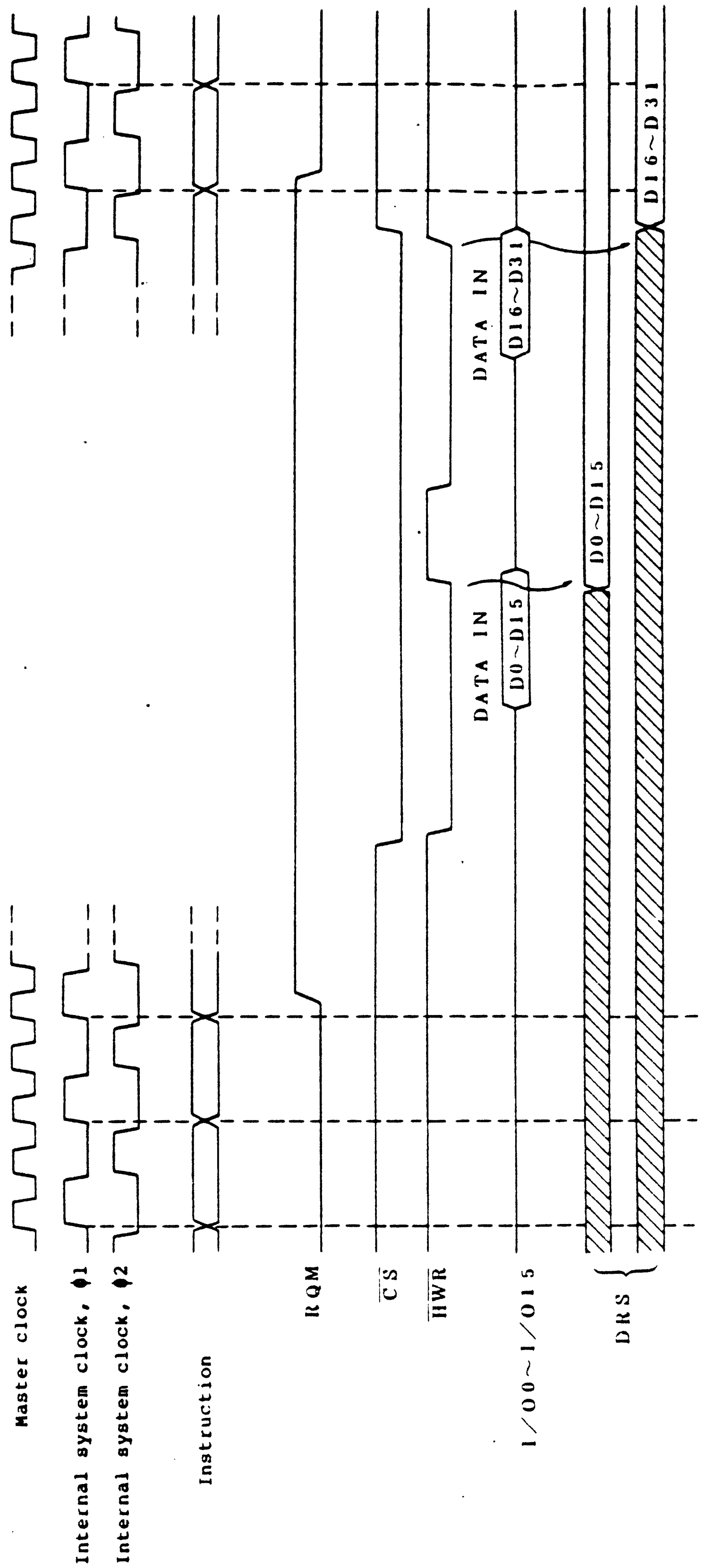


Figure 3-45. Host CPU read timing for 32-bit transfer.

Figure 3-46. Host CPU write timing for 32-bit transfer.



3.9.5 General purpose I/O ports in slave mode.

P0 and P1 can be used as general purpose input ports in slave mode. An internal flag is set according to the state of each of these pins, and these flags may be tested by a conditional branch instruction.

P2 and P3 are general purpose output ports whose state is determined by the P2 and P3 bits in the CNT field of an OP instruction. The state specified in the OP instruction is maintained until it is changed, or until a hardware reset is done. P2 and P3 are reset to 0 by a hardware reset.

CHAPTER 4 INSTRUCTION SET

4.1 General.

The ASP will decode microinstructions, loaded into the instruction register either from the on-chip instruction ROM or the external memory, and generate microcode for each functional block. Each block will follow this microcode from the instruction register decoder and perform the specified function.

Instructions are all 32 bits long (single 32-bit word) and are divided into different fields to specify each function. These specifications may be grouped into the following types:

- Specification of operation (ALU function and operands).
- Specification of transfer (Move from source to destination).
- Specification of pointer modifications.
- Specification of control modes for each functional block.
- Data Load Instruction (Load Immediate Data).
- Branch instructions (jump, call, return, conditional jump).

4.2 Instruction Types

ASP instructions all consist of a single 32-bit word. There are three basic types of instructions:

(i) OP type instruction.

This is an ALU operation instruction. 26 different operations may be specified in the upper 5 bits. Transfers may also be specified within an OP instruction, by use of the SRC and DST fields. Pointer modifications and control modes may be specified in the CNT field. By use of the various fields, a single OP instruction may perform several different tasks at once.

(ii) Branch type instruction.

This is the branch instruction. There are four types of branch: unconditional jump, conditional jump, subroutine call, and return. A branch instruction may also simultaneously specify a transfer operation.

(iii) Load type instruction.

This is the Load Immediate Data instruction. It performs a load of 24-bit immediate data to the specified destination register.

Figure 4-1 shows the format of these basic instruction types.

OP TYPE INSTRUCTION:

```
 31     27 26                    15 14 13 12 10 9      5 4      0
| OP(5)  |          CNT(12)         |P 2| Q(3)|  SRC(5) | DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

BRANCH TYPE INSTRUCTION:

```
 31   28 27                      15 14      10 9       5 4      0
| B(4) |            NA(13)        |   C(5)   |  SRC(5) | DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

LOAD TYPE INSTRUCTION:

```
 31 29 28                                              5 4      0
|LDI 3|                     IM(24)                     | DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

Figure 4-1. Instruction type formats.

4.3 OP Instruction

This instruction specifies the ALU operation and operand(s), transfer operation (source and destination), pointer modifications and control mode changes. It consists of six fields, which perform the following functions:

(i) The upper 5 bits of the instruction (OP field, IR31 - IR27) determine if the instruction is an OP instruction, and which type of ALU function is to be performed. 26 different OP codes (00H - 19H) may be specified. Note that codes 1AH and 1BH (1101X) specify a branch type instruction, and codes 1CH - 1FH (111XX) specify a load immediate data instruction.

(ii) The CNT (control) field (bits IR26 - IR15) determines such specifications as addressing modes, pointer modifications, flag register switching, transfer format specification, loop counter decrementing, interrupt control, shift value specification, local branch address, ROM address value, etc.

(iii) The P field (bits IR14 and IR13) specifies the P register input for the ALU operation, if an ALU operation that requires two operands is specified. The internal data bus (or processing unit bus, if a working register is specified as the source), M (multiplier result) register, RAM0 or RAM1 may be specified.

(iv) The Q field (bits IR12 - IR10) specifies the Q register input for the ALU operation. One of the eight working registers (WR0 - WR7) may be specified in this field. The operation result is stored back into the working register specified by the Q field.

(v) The SRC field (bits IR9 - IR5) specifies the source register for a transfer via the internal data bus. 32 different source specifications are available.

(vi) The DST field (bits IR4 - IR0) specifies the destination register for a transfer via the internal data bus. 32 different destination specifications are available (most are the same as the SRC specifications) including two which perform a simultaneous load of the two multiplier input registers.

Except when a local jump is specified in the CNT field, the program counter automatically increments for the next instruction.

The bit format of the OP instruction is shown in Figure 4-2.

Figure 4-2. Op instruction format.

```
 31      27 26                      15 14 13 12 10 9        5 4        0
| OP(5)    |          CNT(12)         |P 2| Q(3)|  SRC(5)  |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
           | | | | | | | | | | | | |
           | | | | | | | | | | | | |
          / / / / /  / |  \  \ \ \ \
         / / / / |  |  |  |  | \ \ \ \
        / / /  | |  |  |  |  |  |  \ \ \
       / / |   | |  |  |  |  |  |  |  \ \
      /  | |   | |  |  |  |  |  |  |  |  \
|26 |25 |24 |23 |22 |21 |20 |19 |18 |17 |16 |15 |
| 0   0 |  M0   |  M1   |    DP0    |    DP1    |
| 0   1   0   0 |  EA   |    DP0    |    DP1    |
| 0   1   0   1 |  RP   |  M0   |    DP0    |FC |
| 0   1   1   0 |  RP   |  M1   |    DP1    |FC |
| 0   1   1   1 |  RP   |  M0   |  M1   | L |FC |
| 1   0   0   0   0 |   BAS0    |   BAS1    |FC |
| 1   0   0   0   1 |      RPC      |-- | L |FC |
| 1   0   0   1   0 |P3 |P2 |EM |  BM   | L |FC |
| 1   0   0   1   1 |  RW   |---------  | L |FC |
| 1   0   1   0   0 |    WT     |-----  | L |FC |
| 1   0   1   0   1 |    NF     |  WI   | L |FC |
| 1   0   1   1   0 |    FIS    |  FD   | L |-- |
| 1   0   1   1   1 |            SHV            |
| 1   1   0 |                RPS                |
| 1   1   1 |                NAL                |
```

4.3.1 OP Field

This is the field which specifies the operation the ASP will perform in the floating-point ALU (FALU). It can specify the 26 operations shown in Table 4-1.

Table 4-1. Op field specifications.

MNEMONIC	OP FIELD	OPERATION
	31,30,29,28,27	
NOP	00000	NO OPERATION
INC	00001	INCREMENT
DEC	00010	DECREMENT
ABS	00011	ABSOLUTE VALUE
NOT	00100	NOT--ONE'S COMPLEMENT
NEG	00101	NEGATE--TWO'S COMPLEMENT
SHLC	00110	SHIFT LEFT WITH CARRY
SHRC	00111	SHIFT RIGHT WITH CARRY
ROL	01000	ROTATE LEFT
ROR	01001	ROTATE RIGHT
SHLM	01010	SHIFT LEFT MULTIPLE
SHRM	01011	SHIFT RIGHT MULTIPLE
SHRAM	01100	SHIFT RIGHT ARITHMETIC MULTIPLE
CLR	01101	CLEAR
NORM	01110	NORMALIZE
CVT	01111	CONVERT FLOATING POINT FORMAT
ADD	10000	FIXED POINT ADD
SUB	10001	FIXED POINT SUBTRACT
ADDC	10010	FIXED POINT ADD WITH CARRY
SUBC	10011	FIXED POINT SUBTRACT WITH BORROW
CMP	10100	COMPARE (FLOATING POINT)
AND	10101	LOGICAL AND
OR	10110	LOGICAL OR
XOR	10111	LOGICAL EXCLUSIVE OR
ADDF	11000	FLOATING POINT ADD
SUBF	11001	FLOATING POINT SUBTRACT

The operation of all OP codes, along with their effect on the processor status word (PSW), are described in detail on the following pages.

Note that the PSW operation is valid only for the PSW selected at the time of the operation (although in some cases the carry bit of the other PSW will affect the operation). The abbreviations used for the effects of operations on the PSW are: OVFE, overflow of exponent; C, carry; Z, zero; S, sign; OVFM, overflow of mantissa. The indicators used are: $, affected by operation result; 0, reset to zero; 1, set to one; *, retains the previous value; X, undefined.

ABS (Absolute Value)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	0	1	1

Operand(s):

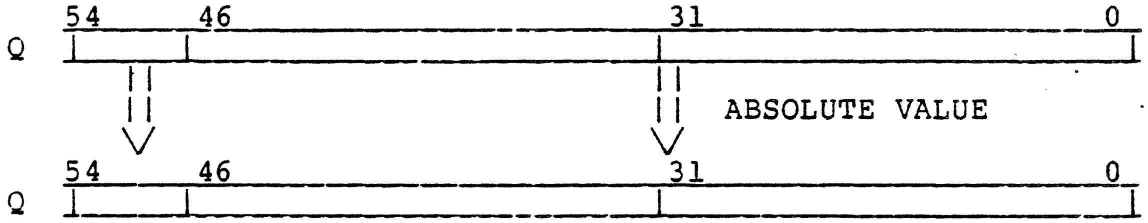
The working register specified by the Q field.

Description of operation:

The absolute value of the mantissa stored in the working register specified by the Q field is returned to the same working register. The exponent is not affected.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	$

Special Case(s):

If the input is 80...0H, the output will be 7F...FEH. This is the only case in which the OFVM bit will be set.

ADD (Add Fixed Point)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	0	0	0

Operand(s):

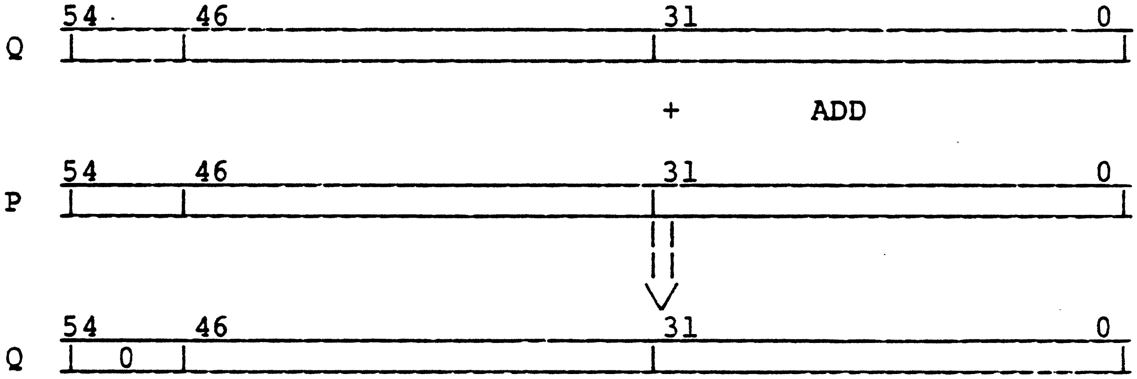
1) The working register specified by the Q field.
2) The contents of either the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

The mantissas of the two inputs will be added together. The exponent of the result will be set to 0.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

Special Case(s):

None.

ADDC (Add fixed point with carry)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	0	1	0

Operand(s):

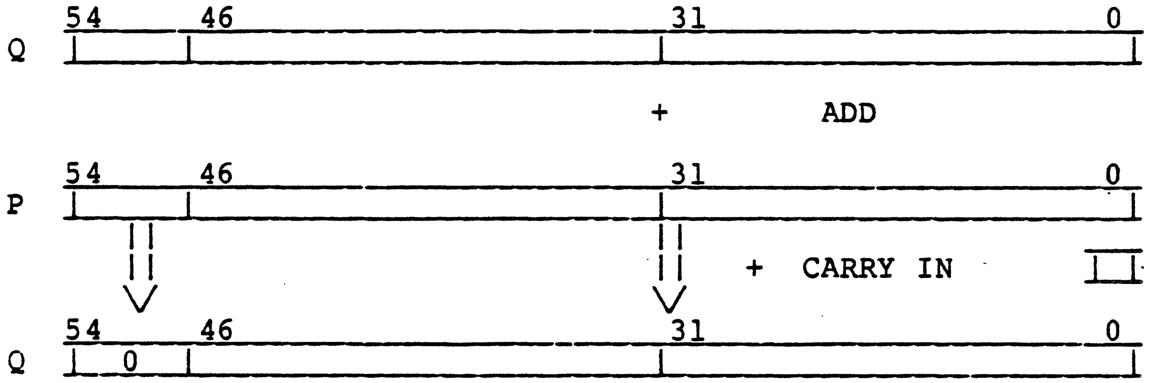
1) The working register specified by the Q field.
2) The contents of either the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

The mantissas of the two inputs will be added together (with carry). The carry bit of the PSW not selected is used as the carry input to the operation. The exponent of the result will be set to 0.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

Note that the carry bit of the PSW not selected will affect this operation's results.

Special Case(s):
None.

ADDF (Add Floating Point)

Contents of OP field:

D31	D30	D29	D28	D27
1	1	0	0	0

Operand(s):

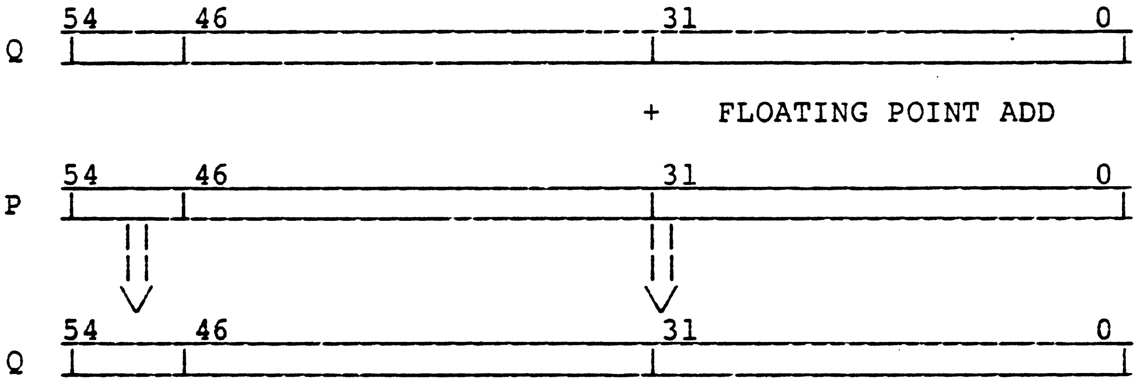
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs floating point addition on the inputs specified by the Q and P fields.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
$	$	$	$	$

Special Case(s):

None.

AND (Logical AND)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	1	0	1

Operand(s):

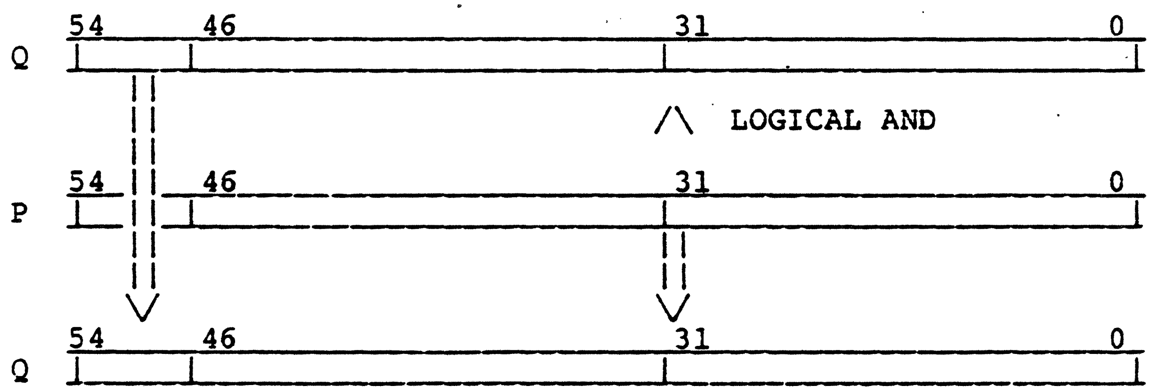
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs a logical AND operation on the mantissas of the inputs. The exponent of the Q input will be preserved in the result.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

CLR (Clear)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	1	0	1

Operand(s):

The working register specified by the Q field.

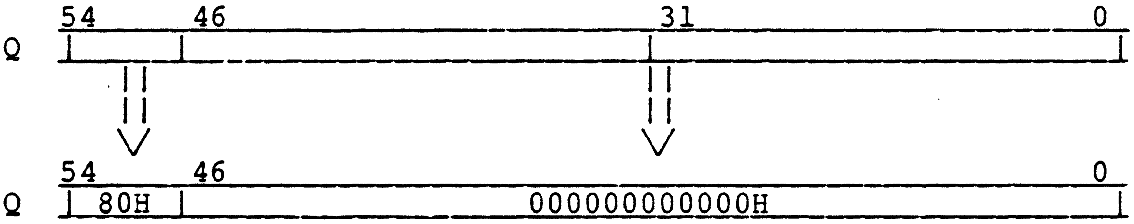
Description of operation:

Clears the working register specified in the Q field. Note that a floating-point 0 is defined by:

Exponent = 80H, mantissa = 00...0H

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
0	0	1	0	0

Special Case(s):

None.

CMP (Compare)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	1	0	0

Operand(s):

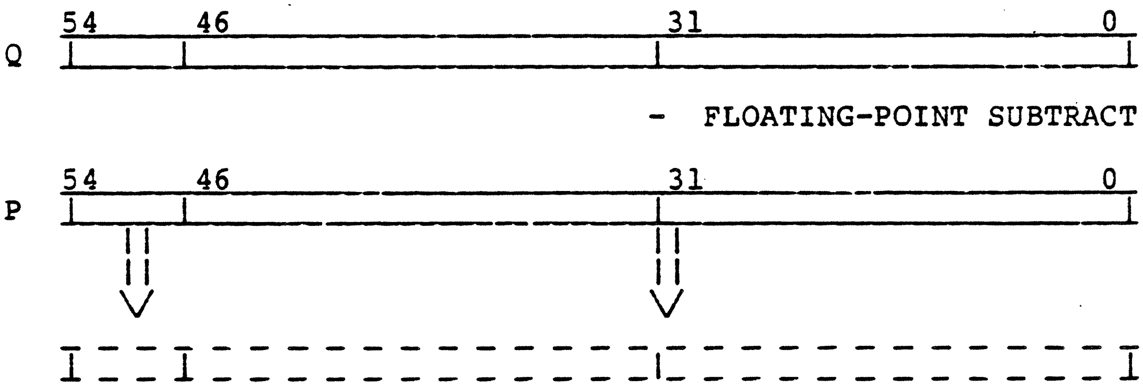
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

The P input is subtracted from the Q input (floating point subtract). The result is not stored. Since this is a compare instruction, only the flags in the PSW will change.

Operation result output:

The result is not stored. Only the PSW flags will change.



PSW Operation:

OVFE	C	Z	S	OVFM
$	$	$	$	$

Special Case(s):

None.

CVT (Convert Floating-Point Format)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	1	1	1

Operand(s):

The working register specified by the Q field.

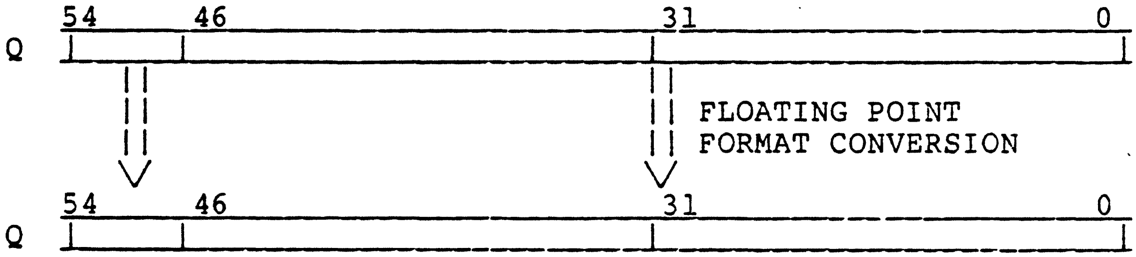
Description of operation:

Converts the format of the contents of the working register specified by the Q field according to the condition specified by the FD bits of the CNT field, as shown below.

Operation result output:

The result is stored to the working register specified by the Q field after two instructions (see section on pipelining).

MNEMONIC	FD FIELD	DATA CONVERSION FORMAT SPECIFICATION
	18,17	
(NON)	00	NO CHANGE IN SPECIFICATION
SPIE	01	CONVERSION OF ASP FORMAT INTO IEEE FORMAT (DEFAULT)
IESP	10	CONVERSION OF IEEE FORMAT TO ASP FORMAT
	11	USE PROHIBITED



PSW Operation:

OVFE	C	Z	S	OVFM
X	0	$	$	0

Difference between ASP and IEEE floating-point data formats:

ASP format has an 8-bit two's complement exponent, and either 24 or 47 bit mantissa, also in two's complement. IEEE format has an offset binary exponent, a sign-magnitude mantissa format (with the sign bit as the leading bit for the entire word), and the mantissa is normalized with the leading 1 bit "hidden".

The conversion from ASP to IEEE format consists of the following steps:

1. Add 7EH to the exponent part.
2. Convert the mantissa to absolute value, rotate to the left one bit, fill LSB with 0, insert sign bit with sign of original.
3. Rotate right 1 bit the most significant 9 bits (this will properly place the sign bit in front of the exponent). Store result back to Working Register.

The conversion from IEEE to ASP format consists of the following steps:

1. Rotate left 1 bit the most significant 9 bits (this will bring the sign bit back to the MSB of the mantissa field, and the exponent will be in the 8 MSBs of the entire word).
2. Add 82H to the exponent part.
3. If the sign bit is 1, rotate right 1 bit the mantissa, and then two's complement it. If the sign bit is 0, make it 1 and then rotate right 1 bit the mantissa (without two's complementing). Store result back to working register.

Note that in either case, the conversion is performed in a single instruction.

Special Case(s):

If the exponent in IEEE format is FFH or 00H, it is not regarded as a numeric value, and therefore has no meaning when converted. These correspond to exponents of 81H and 82H in ASP format; therefore, these values should be trapped by the user's program.

The exponent 80H (negative) in ASP format would be incorrectly translated as FEH (positive) in IEEE format; therefore, these values should also be trapped by the user's program.

DEC (Decrement)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	0	1	0

Operand(s):

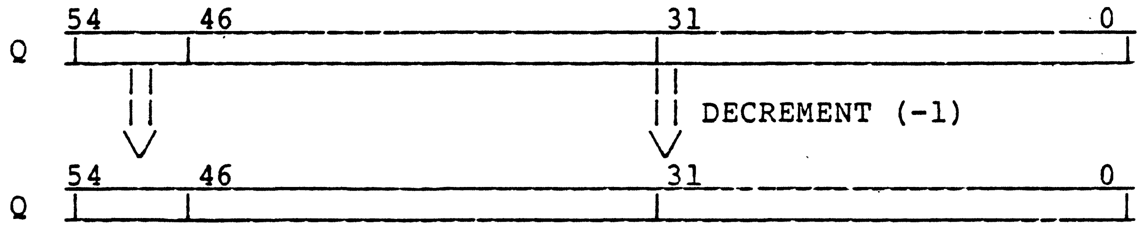
The working register specified by the Q field.

Description of operation:

Decrements (subtracts 1 from) the mantissa of the working register specified in the Q field. The exponent part will be unchanged.

Operation result output:

The result is stored to the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	$	$	$	$

Special Case(s):

A mantissa input value of 00..00H will decrement to FF..FEH, and a mantissa input value of 80..00H will decrement to 7F..FEH.

INC (Increment)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	0	0	1

Operand(s):

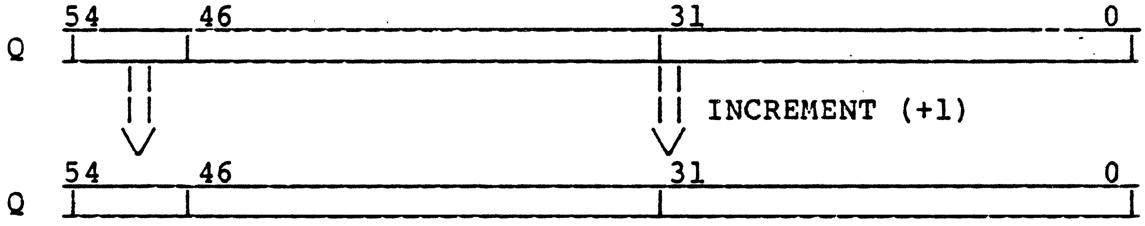
The working register specified by the Q field.

Description of operation:

Increments (adds 1 to) the mantissa of the working register specified in the Q field. The exponent part will be unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	$	$	$	$

Special Case(s):

A mantissa input value of FF..FEH will increment to 00..00H, and a mantissa input value of 7F..FEH will increment to 80..00H.

NEG (Negate)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	1	0	1

Operand(s):

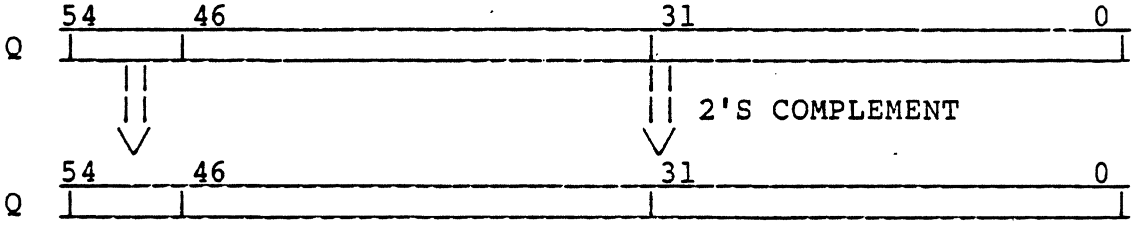
The working register specified by the Q field.

Description of operation:

Takes 2's complement of the mantissa of the working register specified by the Q field. The exponent part remains unchanged.

Operation result output:

The result is stored to the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	$	$	$	$

Special Case(s):

A mantissa input value of 80..00H will negate to itself (80..00H). This is the only case in which the OVFM bit of the PSW will be set.

NOP (No Operation)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	0	0	0

Operand(s):

None.

Description of operation:

No operation.

Operation result output:

None of the contents of any register will change.

PSW Operation:

OVFE	C	Z	S	OVFM
*	*	*	*	*

Special Case(s):

None.

NORM (Normalize)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	1	1	0

Operand(s):

The working register specified by the Q field.

Description of operation:

Converts the data in the working register specified by the Q field according to the procedure specified by the NF bits of the CNT field, as shown below.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).

MNEMONIC	NF FIELD 21,20,19	NORMALIZATION FORMAT SPECIFICATION
(NON)	000	NO CHANGE OF SPECIFICATION
TRNORM	010	TRUNCATING NORMALIZATION (DEFAULT)
RDNORM	100	ROUNDING NORMALIZATION
FLTFIX	110	CONVERT FLOATING TO FIXED POINT
FIXMA	111	FIXED POINT MULTIPLE* ALIGNMENT

* Multiple value is in SVR.

1) Normalization with truncation.

The value obtained by the SAC circuit is input as a shift amount to the barrel shifter. Mantissa is shifted by the barrel shifter and input to the ALU. The value obtained by the SAC circuit will also be subtracted from the exponent. If underflow occurs at this time, both the exponent and mantissa will be set to their underflow saturation values. "Truncation" implies that if the normalized data is transferred to a 32 bit register, the lower 23 bits will be truncated.

PSW Operation:

OVFE	C	Z	S	OVFM
$	0	$	$	0

2) Normalization with rounding.

After normalizing as in (1), the number is rounded to 32 bits using bit 22 of the result as a rounding input (i.e. if it is set, 1 will be added into bit 23). This minimizes the error that would be caused by truncating off the lower 23 bits of the working register when moving it to another register that is only 32 bits long. If overflow occurs from the rounding addition, a 1 bit right shift is performed and the exponent is incremented by 1. If exponent overflow occurs, both the exponent and the mantissa are set to their overflow saturation values.

PSW Operation:

OVFE	C	Z	S	OVFM
$	$	$	$	$

3) Floating to fixed point conversion.

Converts floating-point data to fixed-point data. If the sign bit of the exponent is positive, the mantissa is arithmetically shifted to the left N bits, where N equals the exponent. If the exponent value is greater than the value obtained by the SAC circuit, then the mantissa is set to its overflow saturation value and the overflow flag is set.

If the sign bit of the exponent is negative, the mantissa is arithmetically shifted N bits to the right, where N is the two's complement (absolute value) of the exponent.

The exponent of the result is set to 0.

PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

4) Fixed-point multiple (digit) alignment.

Arithmetically shifts the mantissa N bits to the left. The value of N is specified by either the SVR register or the SHV bits in the CNT field. If the SVR/SHV value is greater than or equal to 46, the mantissa becomes 0 and the overflow flag is set. The exponent is set to 0.

PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

Special Case(s):

See separate descriptions above.

NOT (Not)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	1	0	0

Operand(s):

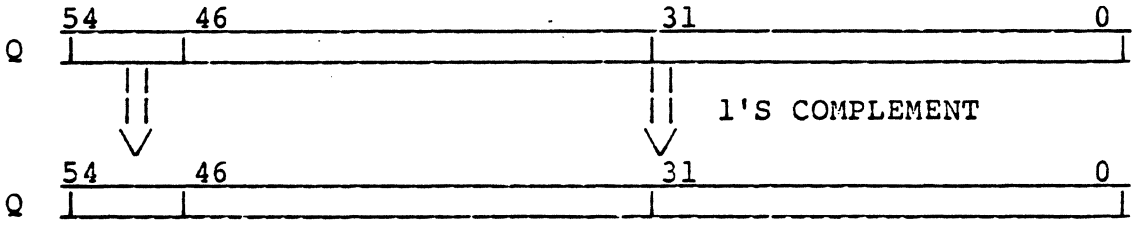
The working register specified by the Q field.

Description of operation:

Inverts (takes the 1's complement of) the mantissa of the working register specified by the Q field. The exponent part remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

OR (Logical OR)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	1	1	0

Operand(s):

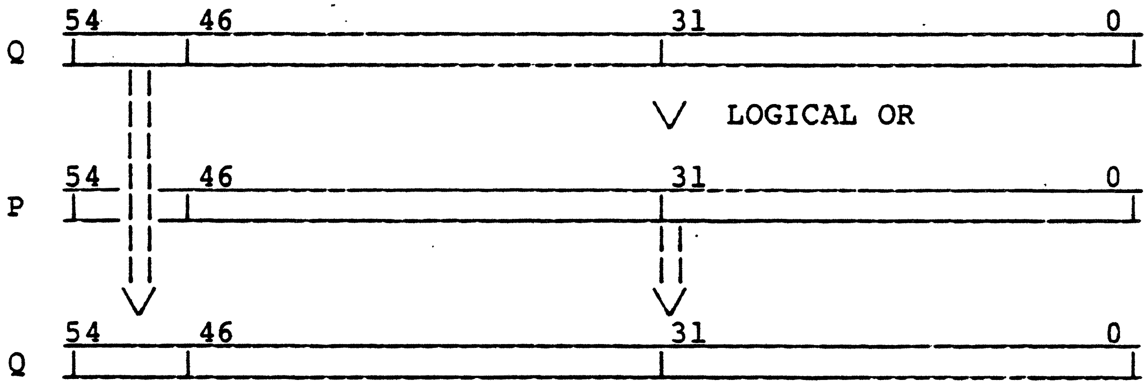
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs a logical OR of the mantissas of the P and Q inputs. The exponent of the Q input will be preserved in the result.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

ROL (Rotate Left)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	0	0	0

Operand(s):

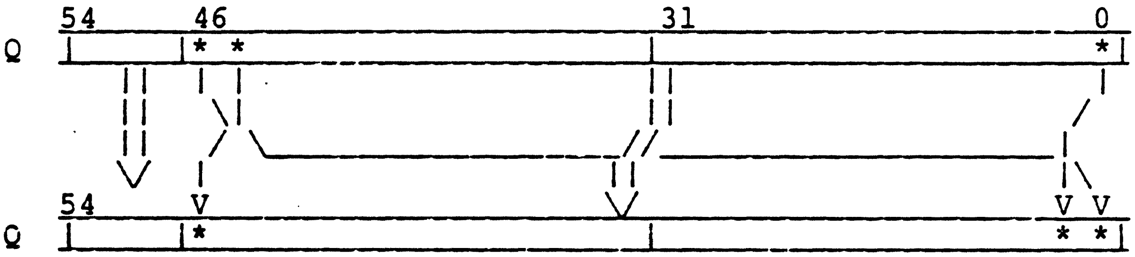
The working register specified by the Q field.

Description of operation:

Rotates the mantissa of the working register specified in the Q field one bit to the left. The MSB of the input mantissa will appear as the LSB of the output mantissa. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

ROR (Rotate Right)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	0	0	1

Operand(s):

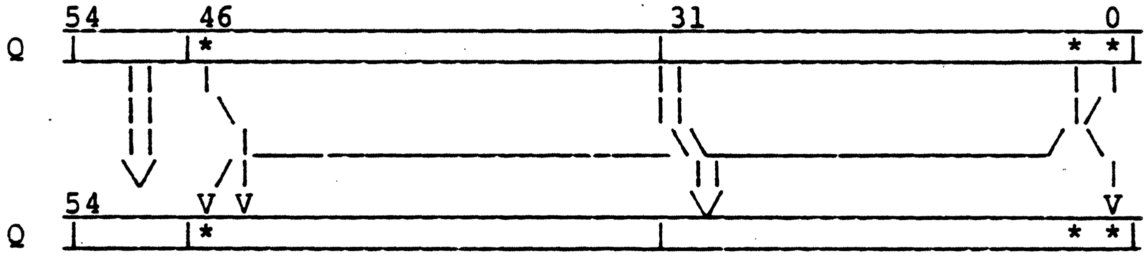
The working register specified by the Q field.

Description of operation:

Rotates the mantissa of the working register specified in the Q field one bit to the right. The LSB of the input mantissa will appear as the MSB of the output mantissa. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

SHLC (Shift Left with Carry)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	1	1	0

Operand(s):

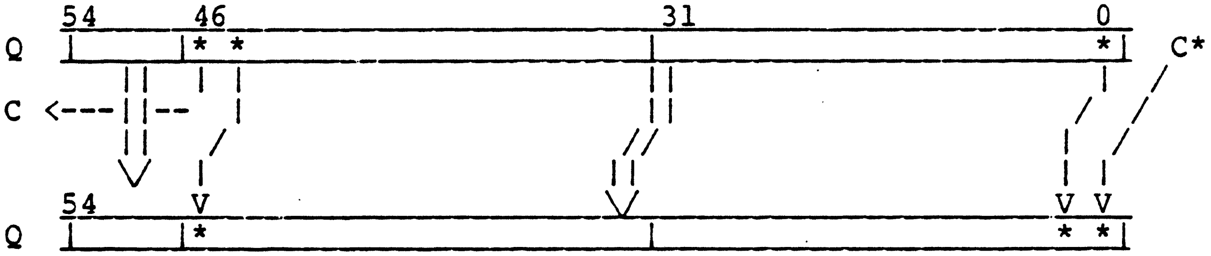
The working register specified by the Q field.

Description of operation:

Shifts the mantissa of the working register specified in the Q field one bit to the left. The MSB of the input mantissa will appear in the carry bit of the PSW selected. The carry bit of the PSW not selected will appear as the LSB of the output mantissa. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	$	$	$	0

Note that the carry bit of the PSW not selected affects this operation's result.

Special Case(s):

None.

SHLM (Shift Left Multiple)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	0	1	0

Operand(s):

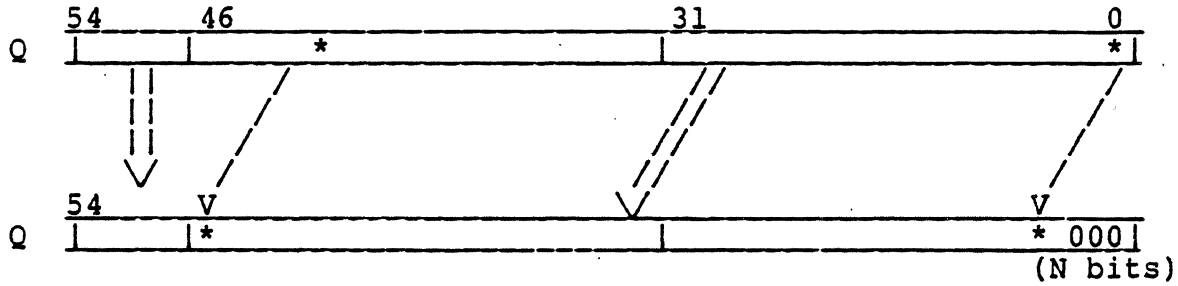
The working register specified by the Q field.

Description of operation:

Shifts the mantissa of the working register specified in the Q field N bits to the left. The value of N is specified in either the SHV bits of the CNT field, or in the SVR register (0 <= N <= 46). The direction bit of the SHV or SVR is ignored. The N MSBs of the input will be discarded, and the N LSBs of the result will be filled with zeroes. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

SHRC (Shift Right with Carry)

Contents of OP field:

D31	D30	D29	D28	D27
0	0	1	1	1

Operand(s):

The working register specified by the Q field.

Description of operation:

Shifts the mantissa of the working register specified in the Q field one bit to the right. The LSB of the input mantissa will appear in the carry bit of the PSW selected. The carry bit of the PSW not selected will appear as the MSB of the output mantissa. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).

```
    54     46                          31                          0
Q   |      |*                          |                         * *|
       ||  |                           ||                        | |
C*  ---||- |                           ||                        \ ---> C
       ||  \ \                          \\                        |
       \/   | |                          ||                       |
    54      V V                          \/                       V
Q   |      |* *                        |                          *|
```

PSW Operation:

OVFE	C	Z	S	OVFM
*	$	$	$	0

Note that the carry bit of the PSW not selected affects this operation's result.

Special Case(s):

None.

SHRAM (Shift Right Arithmetic Multiple)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	1	0	0

Operand(s):

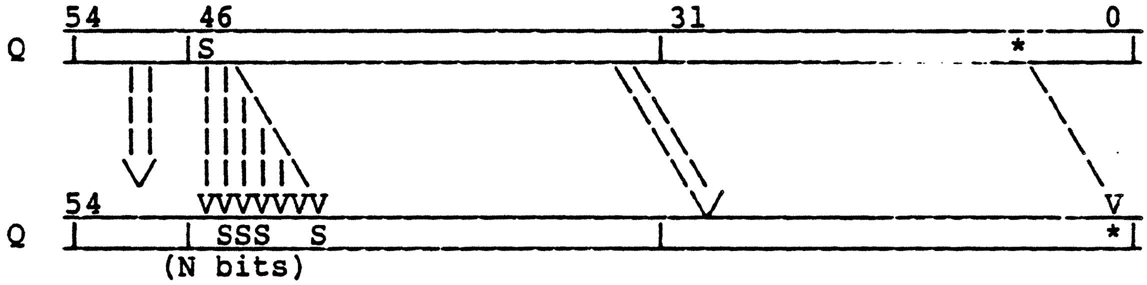
The working register specified by the Q field.

Description of operation:

Arithmetically shifts the mantissa of the working register specified in the Q field N bits to the right. The value of N is specified in either the SHV bits of the CNT field, or in the SVR register (0 <= N <= 46). The direction bit of the SHV or SVR is ignored. The N LSBs of the input will be discarded, and the N MSBs of the result will be filled with the value of the sign bit before the shift. The exponent remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

SHRM (Shift Right Multiple)

Contents of OP field:

D31	D30	D29	D28	D27
0	1	0	1	1

Operand(s):

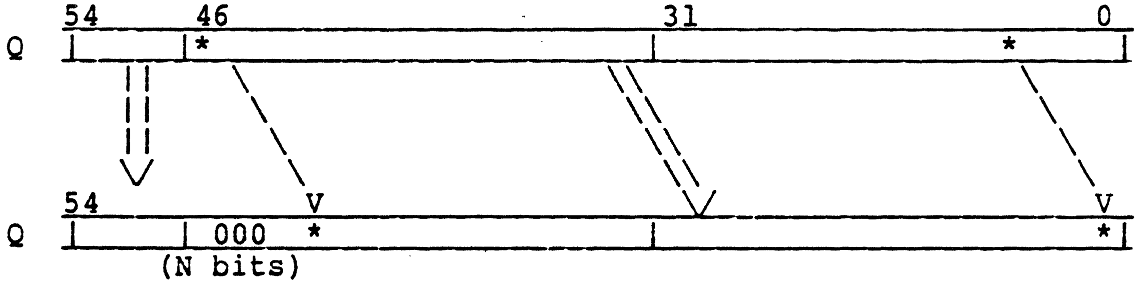
The working register specified by the Q field.

Description of operation:

Shifts the mantissa of the working register specified in the Q field N bits to the right. The value of N is specified in either the SHV bits of the CNT field, or in the SVR register (0 <= N <= 46). The direction bit of the SHV or SVR is ignored. The N LSBs of the input will be discarded, and the N MSBs of the result will be filled with zeroes. The exponent part remains unchanged.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

SUB (Subtract Fixed-Point)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	0	0	1

Operand(s):

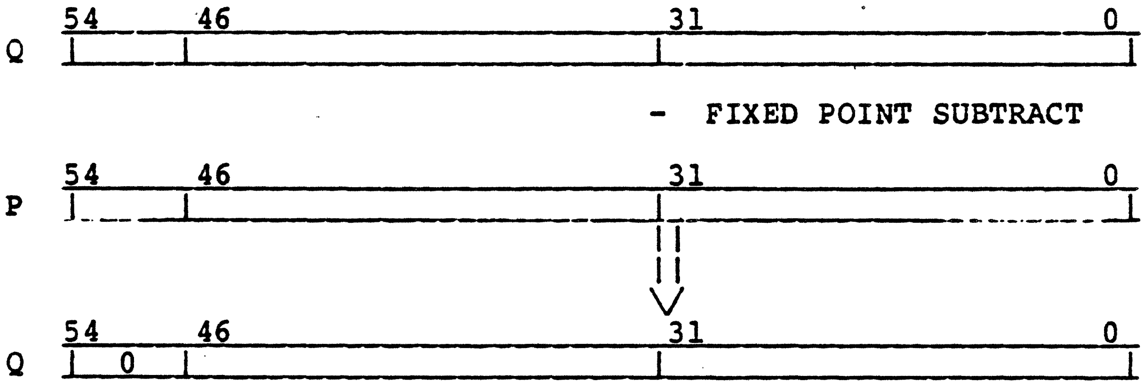
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs fixed-point subtraction. The mantissa value of the P input is subtracted from the mantissa value of the Q input. The exponent of the result will be set to 0.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

Special Case(s):

None.

SUBC (Subtract fixed point with carry)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	0	1	1

Operand(s):

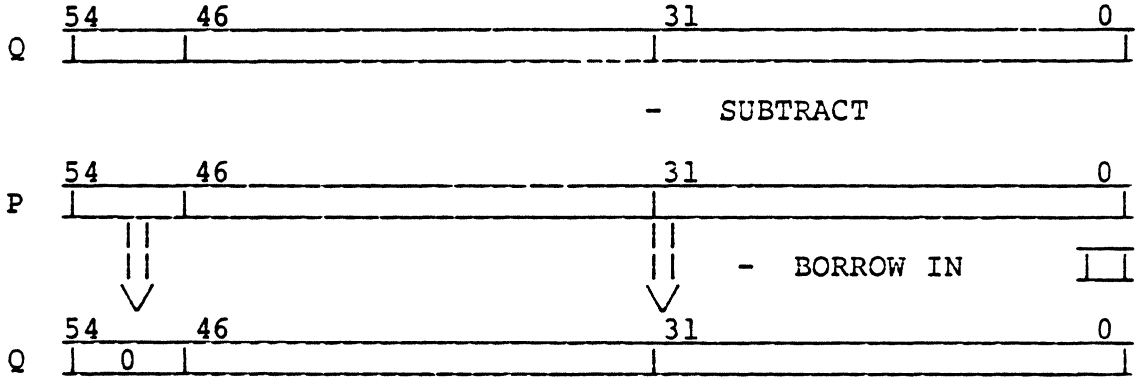
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

The mantissa of the P input will be subtracted from the mantissa of the Q input. The carry bit of the PSW not selected is used as the borrow input to the operation. The exponent of the result will be set to 0.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
0	$	$	$	$

Note that the carry bit of the PSW not selected will affect this operation's results.

Special Case(s):

None.

SUBF (Subtract Floating Point)

Contents of OP field:

D31	D30	D29	D28	D27
1	1	0	0	1

Operand(s):

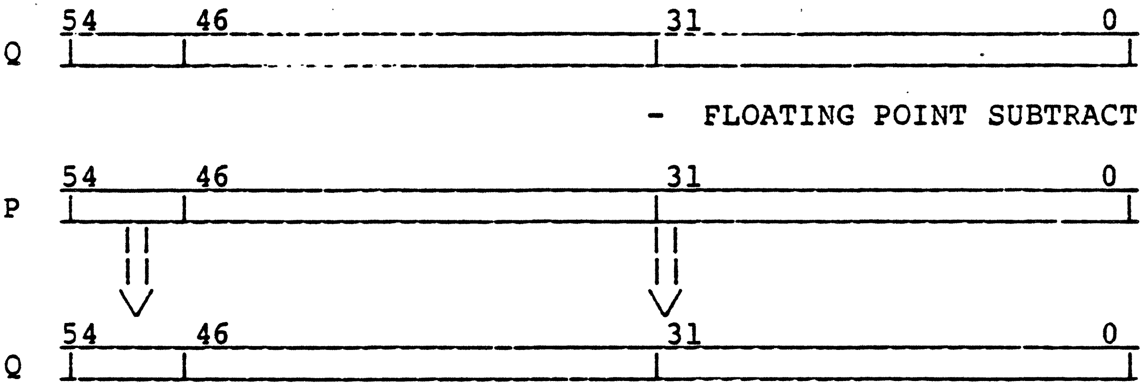
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs floating point subtraction on the inputs specified by the Q and P fields. The P input will be subtracted from the Q input.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
$	$	$	$	$

Special Case(s):

None.

XOR (Logical Exclusive OR)

Contents of OP field:

D31	D30	D29	D28	D27
1	0	1	1	1

Operand(s):

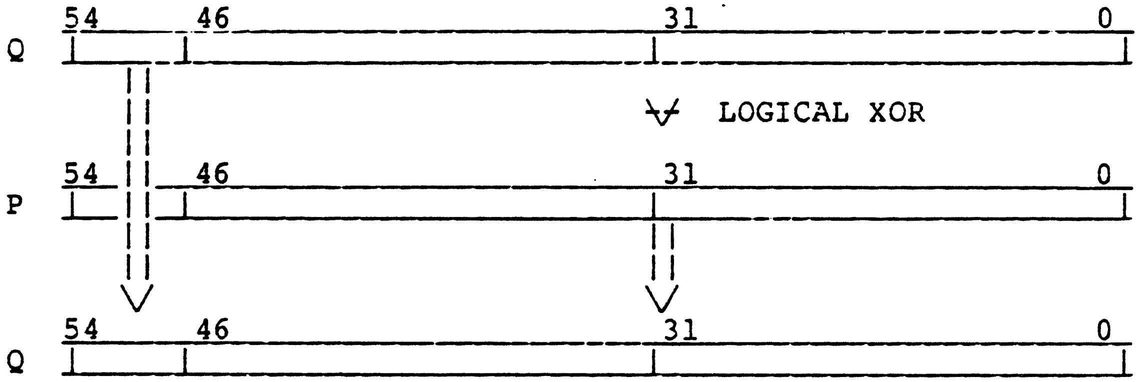
1) The working register specified by the Q field.
2) The contents of the internal data bus, M register, RAM0, or RAM1, as specified by the P field.

Description of operation:

Performs a logical XOR (exclusive OR) of the mantissas of the P and Q inputs. The exponent part of the Q input will be preserved in the output.

Operation result output:

The result is stored in the working register specified by the Q field after two instructions (see section on pipelining).



PSW Operation:

OVFE	C	Z	S	OVFM
*	0	$	$	0

Special Case(s):

None.

Table 4-2 summarizes the bits of the PSW that will be affected by the various instructions.

ALU	CONTENTS OF PSW				
OPERATION	OVFE	C	Z	S	OVFM
NOP	*	*	*	*	*
INC	*	$	$	$	$
DEC	*	$	$	$	$
ABS	*	$	$	0	$+
NOT	*	0	$	$	0
NEG	*	$	$	$	$+
SHLC	*	$	$	$	0
SHRC	*	$	$	$	0
ROL	*	0	*	$	0
ROR	*	0	*	$	0
SHLM	*	0	$	$	0
SHRM	*	0	$	$	0
SHRAM	*	0	$	$	0
CLR	0	0	1	0	0
NORM (NORM.)	$	0	$	$	0
(ROUNDING)	$	$	$	$	$
(FLT - FIX)	*	0	$	$	$
(FIX M. A.)	*	0	$	$	$
CVT	X	0	$	$	0
ADD	*	$	$	$	$
SUB	*	$	$	$	$
ADDC	*	$	$	$	$
SUBC	*	$	$	$	$
CMP	$	$	$	$	$
AND	*	0	$	$	0
OR	*	0	$	$	0
XOR	*	0	$	$	0
ADDF	$	$	$	$	$
SUBF	$	$	$	$	$

$: Flag will be affected by the result of the operation.
0 : Flag will be reset to "0".
1 : Flag will be set to "1".
* : Previous condition of flag will be preserved.
X : Undecided.
+ : If the original data in the mantissa was 80---0H,
    OVFM = "1" after operation.

Table 4-2. Effects of ALU operations on PSW flags.

4.3.2 Q Field.

The Q field specifies the input to the Q register, which is used for the ALU operation. The Q field is a 3-bit field which specifies one of the 8 working registers (WR0 thru WR7).

1) The working register specified by this field is input to the Q register and is operated on by the ALU.

2) The result of the ALU operation is output back to the working register specified by this field.

3) The working register specified by this field may also be specified in the SRC field at the same time. In this case, the data in the register prior to the ALU operation is sent to the internal bus.

4) If the working register specified by this field is also specified by the DST field at the same time, the contents of the internal bus are input to the Q register, and then the ALU operation is performed.

5) If the working register specified by this field is also specified in the Q field of the next instruction, the result (ALU output) of the present instruction becomes the input to the Q register for the next instruction (this means that continuous operations on the same working register are possible).

6) If the working register specified by this field is also specified by the SRC field of the next instruction, the value output to the internal bus (for the transfer operation of the next instruction) is the value of the working register before the ALU operation of the current instruction. In other words, the ALU result value is not saved in the working register in time to be used as the source for the next instruction.

MNEMONIC	Q FIELD	REGISTER SPECIFICATION
	12,11,10	
WR0	000	WORKING REGISTER 0
WR1	001	WORKING REGISTER 1
WR2	010	WORKING REGISTER 2
WR3	011	WORKING REGISTER 3
WR4	100	WORKING REGISTER 4
WR5	101	WORKING REGISTER 5
WR6	110	WORKING REGISTER 6
WR7	111	WORKING REGISTER 7

TABLE 4-3. Q field specificatons.

4.3.3 P Field.

The P field is a 2-bit field which specifies the source of the input data to the P register, which is used as an input to the ALU for operations requiring 2 operands.

1) If the internal bus (IB) is specified in this field, the contents of the register specified in the SRC field are input to the P register. If a 55-bit register (either a working register or the multiplier result register M) is selected, it is input unchanged (55 bits wide). If a register with 32 or fewer bits is selected, the value on the internal bus is input into the upper 32 bits of the P register, with the lower 23 bits of the P register set to 0. If no SRC specification is made (NON), all 55 bits of the P input are set to 1.

2) If the M register is specified, the contents of the M register are sent to the P register via a sub-bus (55 bits wide) which is independent of the internal bus.

3) If RAM0 or RAM1 is specified, the contents of RAM0 or RAM1 are sent to the P register via a sub-bus (32 bits wide) which is independent of the internal bus. The 23 lower bits of the P register are filled with 0's.

MNEMONIC	P FIELD	INPUT OF P REGISTER SPECIFICATION
	14,13	
IB	00	INTERNAL BUS
M	01	MULTIPLIER OUTPUT REGISTER
RAM0	10	RAM BLOCK 0
RAM1	11	RAM BLOCK 1

Table 4-4. P field specifications.

4.3.4 CNT Field

The CNT field is a 12-bit field in the OP instruction. It has the following functions: specification and modification of pointers, change of flag register, specification of format to be used for transfers, loop counter decrement, and mode controls of other functional blocks.

Each function field in the CNT field is described in one of the following sections. Note that some function sub-fields affect the operation of the current instruction, while some do not take effect until the next instruction.

The format of the fields within the CNT field is determined by the upper 2 - 5 bits of the CNT field. Control field format is shown in Figure 4-3.

A summary of the various CNT sub-fields, grouped by functional area, is given in Table 4-5. Also shown is whether any given sub-field is effective on the current instruction (*), or on the next instruction (-->).

<table>
<tr><td>0</td><td>0</td><td colspan="2">M0</td><td colspan="2">M1</td><td colspan="3">DP0</td><td colspan="3">DP1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td colspan="2">EA</td><td colspan="3">DP0</td><td colspan="3">DP1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td colspan="2">RP</td><td colspan="2">M0</td><td colspan="3">DP0</td><td>FC</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td colspan="2">RP</td><td colspan="2">M1</td><td colspan="3">DP1</td><td>FC</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td colspan="2">RP</td><td colspan="2">M0</td><td colspan="2">M1</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="3">BAS0</td><td colspan="3">BAS1</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="4">RPC</td><td>--</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>P3</td><td>P2</td><td>EM</td><td colspan="2">BM</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td colspan="2">RW</td><td colspan="3">---------</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td colspan="3">WT</td><td colspan="2">-----</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td colspan="3">NF</td><td colspan="2">WI</td><td>L</td><td>FC</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td colspan="3">FIS</td><td colspan="2">FD</td><td>L</td><td>--</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td colspan="7">SHV</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td colspan="9">RPS</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="9">NAL</td></tr>
</table>

Figure 4-3. Control field format.

Table 4-5. Control fields.

GROUP	FIELD	FUNCTION	*/-->
Interrupt	EM, BM	Enable and disable maskable interrupt, and control interrupt memorization.	-->
PSW	FIS	PSW control (select and clear)	*
	FC	Select other PSW	*
Data ROM Pointer	RP	Controls ROM pointer operation	-->
	RPC	Specifies n value for special manipulation of ROM pointer	-->
	RPS	Specifies 9 lower bits of data ROM address	-->
Data RAM0 and RAM1 Pointers	M0	Specifies RAM0 addressing mode	-->
	M1	Specifies RAM1 addressing mode	-->
	DP0	Controls modification of base pointer 0 and index register 0	-->
	DP1	Controls modification of base pointer 1 and index register 1	-->
	BASE0	Specifies counter length of modulo count operation of base pointer 0	-->
	BASE1	Specifies counter length of modulo count operation of base pointer 1	-->
Data Format Conversion	FD	Controls conversion mode for floating point CVT.	*
	WI	Controls transfer format when working register is specified in DST field.	-->
	WT	Controls transfer format when working register is specified in SRC field.	-->
Normalization Specification	NF	Specifies normalization, normalization with rounding, floating-point to fixed-point conversion, or digit alignment	*

* Effective starting with the current instruction

--> Effective starting with the next instruction

Table 4-5. cont'd.

Shift Specification	SHV	Controls amount of shift for 47-bit mantissa	*
Data Memory Access	RW	Specifies read/write operation for external memory	*
	EA	Increments or decrements external address register	*
General-Purpose Output Port	P2	Controls state of P2 pin	-->
	P3	Controls state of P3 pin	-->
Loop Counter	L	Decrements loop counter	-->
Jump	NAL	Specifies unconditional local jump address	*

* Effective starting with the current instruction

--> Effective starting with the next instruction

BASE0

Function:

This 3-bit field determines the modulo count for base register 0 (part of the data RAM0 pointer functional block) by specifying the bit length n of the 2**n modulo counter. A modulo count of 2**1 through 2**7 can be specified. If 0 is specified in this field, a normal binary count operation is performed.

Notes for use:

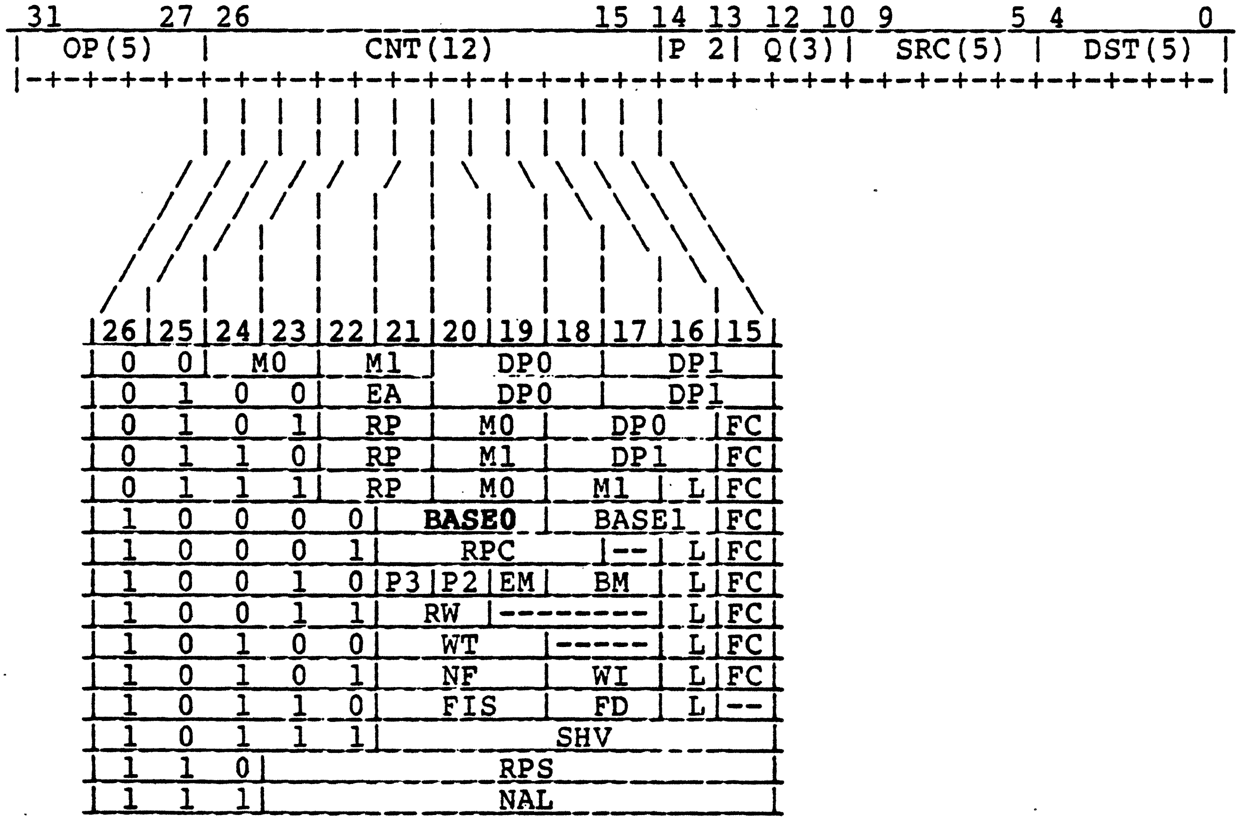
- * The default condition after hardware reset is normal binary count.
- * This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	BASE0 FIELD 21,20,19	SPECIFY MODULO COUNT NUMBER
MCNBP0 imm	(imm)B	SPECIFY BIT LENGTH N FOR 2**N FOR INCREMENTING BASE POINTER 0

* imm (=n) is 1 through 7; 0 specifies ordinary count

Field format:



```
 31     27 26                  15 14 13 12 10 9      5 4      0
| OP(5)  |         CNT(12)        |P 2| Q(3)|  SRC(5) |  DST(5) |
```

26	25	24	23	22	21	20	19	18	17	16	15
0	0	M0		M1		DP0			DP1		
0	1	0	0	EA		DP0			DP1		
0	1	0	1	RP		M0		DP0			FC
0	1	1	0	RP		M1		DP1			FC
0	1	1	1	RP		M0		M1		L	FC
1	0	0	0	0	**BASE0**			BASE1			FC
1	0	0	0	1	RPC				--	L	FC
1	0	0	1	0	P3	P2	EM	BM		L	FC
1	0	0	1	1	RW		---------			L	FC
1	0	1	0	0	WT			-----		L	FC
1	0	1	0	1	NF			WI		L	FC
1	0	1	1	0	FIS			FD		L	--
1	0	1	1	1	SHV						
1	1	0	RPS								
1	1	1	NAL								

BASE1

Function:

This 3-bit field determines the modulo count for base register 1 (part of the data RAM1 pointer functional block) by specifying the bit length n of the 2**n modulo counter. A modulo count of 2**1 through 2**7 can be specified. If 0 is specified in this field, a normal binary count operation is performed.

Notes for use:

* The default condition after hardware reset is normal binary count.
* This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	BASE1 FIELD 18,17,16	SPECIFY MODULO COUNT NUMBER
MCNBP1 imm	(imm)B	SPECIFY MODULO COUNT NUMBER (2**N) FOR INCREMENTING BASE POINTER 1

* imm (=n) is 1 through 7, 0 specifies ordinary count

Field format:

```
 31      27 26                         15 14 13 12 10 9       5 4        0
| OP(5)    |           CNT(12)           |P 2| Q(3) |  SRC(5)  |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
           | | | | | | | | | | | |
           | | | | | | | | | | | |
          / / / / / / | \ \ \ \ \
         / / / / |  | | | | \ \ \ \
        / / / |  |  | | | |  | \ \ \
       / / |  |  |  | | | |  |  | \ \
      / |  |  |  |  | | | |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |  DP0   |   DP1   |
    | 0  1  0  0|  EA |  DP0   |   DP1   |
    | 0  1  0  1|  RP |  M0 |  DP0   |FC|
    | 0  1  1  0|  RP |  M1 |  DP1   |FC|
    | 0  1  1  1|  RP |  M0 | M1  | L|FC|
    | 1  0  0  0  0|  BASE0 | BASE1 |FC|
    | 1  0  0  0  1|   RPC  |--|  L|FC|
    | 1  0  0  1  0|P3|P2|EM|  BM | L|FC|
    | 1  0  0  1  1|  RW |---------| L|FC|
    | 1  0  1  0  0|   WT   |-----| L|FC|
    | 1  0  1  0  1|   NF   |  WI | L|FC|
    | 1  0  1  1  0|   FIS  |  FD | L|--|
    | 1  0  1  1  1|         SHV         |
    | 1  1  0|           RPS             |
    | 1  1  1|           NAL             |
```

BM, EM

Function:

A combination of the EM and BM fields enables or disables the maskable interrupt, and determines if a maskable interrupt input signal, occurring while in the interrupt disabled condition, will be memorized for future use (see table below) . If the memory function is turned off, any interrupt that was already memorized will be reset. After hardware reset, the maskable interrupt and the memory function are disabled. When either a maskable or non-maskable interrupt occurs, the maskable interrupt is disabled.

Notes for use:

* This field is effective starting with the instruction after the one in which it is specified.
* Because P2, P3, EM, and BM are all specified together in the same instruction (they co-exist in one particular CNT field format), care should be exercised when specifying any one of them, to avoid inadvertantly changing the state of any of the others. For example, suppose P2 was previously set high, and it should remain that way. Then, if EM is specified to enable interrupts, the P2 field should again specify 1. If this specification is omitted in the instruction, it will default to 0.

Mnemonic table:

MNEMONIC		EM	BM FIELD		OPERATION FOR MASKABLE INTERRUPT
EM	BM	19	18	17	
(NOP)	(NOP)	0	0	0	NO OPERATION
(NOP)	CLRBM	0	0	1	CLEAR BOOKING FLAG
(NOP)	SETBM	0	1	0	SET BOOKING FLAG
DI	(NOP)	0	1	1	INTERRUPT DISABLED
EI	(NOP)	1	0	0	INTERRUPT ENABLED
EI	CLRBM	1	0	1	INTERRUPT ENABLED AND CLEAR BOOKING FLAG
EI	SETBM	1	1	0	INTERRUPT ENABLED AND SET BOOKING FLAG
		1	1	1	USE PROHIBITED

* DEFAULT: INTERRUPT DISABLED AND CLEAR BOOKING FLAG
* WRITING "(NOP)" IS NOT NECESSARY, JUST USEFUL FOR REMEMBERING THE AVAILABLE COMBINATIONS AND THEIR EFFECTS

Field format:

```
 31     27 26                  15 14 13 12 10 9      5 4       0
_________________________________________________________________
|  OP(5)  |        CNT(12)        |P 2| Q(3)| SRC(5)  | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    _____________________________________
    | 0  0|  M0 |  M1 |  DP0   |  DP1   |
    _____________________________________
    | 0  1  0  0|  EA |  DP0   |  DP1   |
    _____________________________________
    | 0  1  0  1|  RP |  M0 |  DP0   |FC|
    _____________________________________
    | 0  1  1  0|  RP |  M1 |  DP1   |FC|
    _____________________________________
    | 0  1  1  1|  RP |  M0 | M1  | L|FC|
    _____________________________________
    | 1  0  0  0  0|  BASE0 | BASE1  |FC|
    _____________________________________
    | 1  0  0  0  1|   RPC     |--| L|FC|
    _____________________________________
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    _____________________________________
    | 1  0  0  1  1| RW |---------| L|FC|
    _____________________________________
    | 1  0  1  0  0|  WT    |-----| L|FC|
    _____________________________________
    | 1  0  1  0  1|  NF    | WI  | L|FC|
    _____________________________________
    | 1  0  1  1  0|  FIS   | FD  | L|--|
    _____________________________________
    | 1  0  1  1  1|        SHV         |
    _____________________________________
    | 1  1  0|           RPS            |
    _____________________________________
    | 1  1  1|           NAL            |
    _____________________________________
```

DP0

Function:

The DP0 field (3 bits) specifies counting and clearing operations for the base and index registers, which are used for addressing data RAM 0. The following functions may be specified:

- Clear the contents of either the base or the index register.
- Increment or decrement either the base or the index register.
- Increment or decrement operations performed on the base register will conform to the modulo count specified in the BASE0 field.
- Add the values of base and index registers and store the sum in the index register.

Notes for use:

* This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	DP0 FIELD	POINTER MODIFICATION OPERATION
(NOP)	000	NO OPERATION
INCBP0	001	INCREMENT BASE POINTER 0
DECBP0	010	DECREMENT BASE POINTER 0
CLRBP0	011	CLEAR BASE POINTER 0
STIX0	100	STORE BASE + INDEX TO INDEX REGISTER 0
INCIX0	101	INCREMENT INDEX REGISTER 0
DECIX0	110	DECREMENT INDEX REGISTER 0
CLRIX0	111	CLEAR INDEX REGISTER 0

Field format:

```
 31     27 26                  15 14 13 12 10 9      5 4       0
|  OP(5)  |        CNT(12)        |P 2| Q(3)|  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |   DP0  |   DP1  |
    | 0  1  0  0|  EA |   DP0  |   DP1  |
    | 0  1  0  1|  RP |  M0 |    DP0 |FC|
    | 0  1  1  0|  RP |  M1 |    DP1 |FC|
    | 0  1  1  1|  RP |  M0 |  M1 | L|FC|
    | 1  0  0  0  0|  BASE0 |  BASE1 |FC|
    | 1  0  0  0  1|    RPC    |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM|  BM | L|FC|
    | 1  0  0  1  1|  RW |--------| L|FC|
    | 1  0  1  0  0|   WT   |-----| L|FC|
    | 1  0  1  0  1|   NF   |  WI | L|FC|
    | 1  0  1  1  0|   FIS  |  FD | L|--|
    | 1  0  1  1  1|        SHV         |
    | 1  1  0|           RPS            |
    | 1  1  1|           NAL            |
```

DP1

Function:

The DP1 field (3 bits) specifies counting and clearing operations for the base and index registers, which are used for addressing data RAM 1. The following functions may be specified:

- Clear the contents of either the base or the index register.
- Increment or decrement either the base or the index register.
- Increment or decrement operations performed on the base register will conform to the modulo count specified in the BASE1 field.
- Add the values of base and index registers and store the sum in the index register.

Notes for use:

* This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	DP1 FIELD	POINTER MODIFICATION OPERATION
(NOP)	000	NO OPERATION
INCBP1	001	INCREMENT BASE POINTER 1
DECBP1	010	DECREMENT BASE POINTER 1
CLRBP1	011	CLEAR BASE POINTER 1
STIX1	100	STORE BASE + INDEX TO INDEX REGISTER 1
INCIX1	101	INCREMENT INDEX REGISTER 1
DECIX1	110	DECREMENT INDEX REGISTER 1
CLRIX1	111	CLEAR INDEX REGISTER 1

Field format:

```
_31_____27_26_________________15_14_13_12_10_9______5_4_______0_
|  OP(5)  |        CNT(12)        |P 2| Q(3)| SRC(5)  | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |  DP0   |  DP1   |
    | 0  1  0  0|  EA |  DP0   |  DP1   |
    | 0  1  0  1|  RP |  M0 |   DP0  |FC|
    | 0  1  1  0|  RP |  M1 |  DP1   |FC|
    | 0  1  1  1|  RP |  M0 |  M1 | L|FC|
    | 1  0  0  0  0| BASE0  | BASE1  |FC|
    | 1  0  0  0  1|    RPC    |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    | 1  0  0  1  1| RW  |--------| L|FC|
    | 1  0  1  0  0|  WT    |-----| L|FC|
    | 1  0  1  0  1|  NF    | WI  | L|FC|
    | 1  0  1  1  0|  FIS   | FD  | L|--|
    | 1  0  1  1  1|        SHV         |
    | 1  1  0|           RPS            |
    | 1  1  1|           NAL            |
```

EA

Function:

This 2-bit field specifies a modification of the lower 12 bits of the external address register. Possible specifications are increment, decrement, and no change.

Notes for use:

* This field is effective for the instruction in which it is specified.

Mnemonic table:

MNEMONIC	EA FIELD 22,21	OPERATION FOR EXTERNAL ADDRESS REGISTER
(NOP)	00	NO OPERATION
INCAR	01	INCREMENT EXTERNAL ADDRESS REGISTER
DECAR	10	DECREMENT EXTERNAL ADDRESS REGISTER
	11	USE PROHIBITED

Field format:

```
 31      27 26                        15 14 13 12 10 9       5 4        0
| OP(5)   |          CNT(12)           |P 2| Q(3)| SRC(5)  | DST(5)   |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
            | | | | | | | | | | | |
            | | | | | | | | | | | |
           / / / / / / | \ \ \ \ \
          / / / / |  | |  | | \ \ \
         / / / |  |  | |  | |  | \ \
        / / |  |  |  | |  | |  |  \ \
       / |  |  |  |  | |  | |  |  |  \
|26|25|24|23|22|21|20|19|18|17|16|15|
| 0  0| M0  | M1  |  DP0   |  DP1   |
| 0  1  0  0| EA  |  DP0   |  DP1   |
| 0  1  0  1| RP  | M0  |  DP0   |FC|
| 0  1  1  0| RP  | M1  |  DP1   |FC|
| 0  1  1  1| RP  | M0  | M1  | L|FC|
| 1  0  0  0  0| BASE0  | BASE1  |FC|
| 1  0  0  0  1|   RPC     |--| L|FC|
| 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
| 1  0  0  1  1| RW  |--------| L|FC|
| 1  0  1  0  0|  WT    |-----| L|FC|
| 1  0  1  0  1|  NF    | WI  | L|FC|
| 1  0  1  1  0|  FIS   | FD  | L|--|
| 1  0  1  1  1|        SHV         |
| 1  1  0|           RPS            |
| 1  1  1|           NAL            |
```

FC

Function:

This bit can be used to toggle the selection of the active processor status word (PSW). When FC=1,. the PSW which was not active for the previous instruction becomes active for the current instruction. When FC=0, the PSW which was active during the previous instruction remains active.

Notes for use:

* After hardware reset, PSW0 will be selected.
* If FC is specified as 1 in combination with an ALU operation, the change of flag is valid for that operation. For example, if PSW 0 is already selected, and an ADDC instruction is combined with FC = 1, the carry out from the operation will be saved in PSW 1, while the carry into the LSB will come from PSW 0.
* The PSWs may also be selected and cleared by use of the FIS field specification.

Mnemonic table:

MNEMONIC	FC BIT 15	FLAG CHANGE OPERATION
(NOP)	0	NO OPERATION
XCHPSW	1	EXCHANGE PSW FOR OPERATION

Field format:

```
 31      27 26                        15 14 13 12 10 9       5 4        0
| OP(5)   |           CNT(12)           |P 2| Q(3)|  SRC(5)   |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / / / | \ \ \ \ \ \
        / / / / |  |  |  |  |  \ \ \ \
       / / / |  |  |  |  |  |  |  \ \ \
      / / |  |  |  |  |  |  |  |  |  \ \
     / |  |  |  |  |  |  |  |  |  |  |  \
|26|25|24|23|22|21|20|19|18|17|16|15|
| 0  0|  M0  |  M1  |  DP0  |  DP1  |
| 0  1  0  0|  EA  |  DP0  |  DP1  |
| 0  1  0  1|  RP  |  M0  |  DP0  |FC|
| 0  1  1  0|  RP  |  M1  |  DP1  |FC|
| 0  1  1  1|  RP  |  M0  |  M1  | L|FC|
| 1  0  0  0  0|  BASE0  |  BASE1  |FC|
| 1  0  0  0  1|    RPC     |--| L|FC|
| 1  0  0  1  0|P3|P2|EM|  BM  | L|FC|
| 1  0  0  1  1|  RW  |---------| L|FC|
| 1  0  1  0  0|   WT    |-----| L|FC|
| 1  0  1  0  1|   NF    |  WI | L|FC|
| 1  0  1  1  0|   FIS   |  FD | L|--|
| 1  0  1  1  1|         SHV          |
| 1  1  0|             RPS            |
| 1  1  1|             NAL            |
```

FD

Function:

This 2-bit field specifies the operation that will be performed when the CVT instruction is executed by the ALU. The possible CVT operations that may be specified either convert the internal ASP floating point format to the IEEE floating point format, or vice versa.

Notes for use:

* After hardware reset, the default is ASP to IEEE format conversion.
* This field is effective starting with the instruction after the one in which it is specified, and remains in effect until respecified.

Mnemonic table:

MNEMONIC	FD FIELD 18,17	DATA CONVERSION FORMAT SPECIFICATION
(NON)	00	NO CHANGE OF SPECIFICATION
SPIE	01	CONVERSION OF ASP FORMAT INTO IEEE FORMAT (DEFAULT)
IESP	10	CONVERSION OF IEEE FORMAT TO ASP FORMAT
	11	USE PROHIBITED

Field format:

```
_31_____27_26__________________15_14_13_12_10_9______5_4_______0_
| OP(5)  |          CNT(12)            |P 2| Q(3)|  SRC(5) | DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / / /  |  \  \ \ \ \ \
       / / / /  |  |  |  |  | \ \ \ \
      / / /  |  |  |  |  |  |  | \ \ \
     / /  |  |  |  |  |  |  |  |  | \ \
    / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |  DP0   |  DP1   |
    | 0  1  0  0|  EA |  DP0   |  DP1   |
    | 0  1  0  1|  RP |  M0 |  DP0   |FC|
    | 0  1  1  0|  RP |  M1 |  DP1   |FC|
    | 0  1  1  1|  RP |  M0 |  M1 | L|FC|
    | 1  0  0  0  0| BASE0  | BASE1  |FC|
    | 1  0  0  0  1|   RPC     |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    | 1  0  0  1  1|  RW |--------| L|FC|
    | 1  0  1  0  0|   WT   |-----| L|FC|
    | 1  0  1  0  1|   NF   | WI  | L|FC|
    | 1  0  1  1  0|   FIS  | FD  | L|--|
    | 1  0  1  1  1|        SHV         |
    | 1  1  0|          RPS             |
    | 1  1  1|          NAL             |
```

FIS

Function:

This 3-bit field controls the selection of the processor status word (PSW). It may also be used to clear the contents of either PSW, or both at once.

Notes for .use:

* After hardware reset, PSW 0 is the default selection.
* As with the FC specification, if the FIS field selects a PSW in the same instruction that an ALU operation is being performed, the selection made in the FIS field will take effect for that ALU operation.
* If the PSW Clear specification is made in the FIS field, it will take precedence over the flag change resulting from an ALU operation.
* If the PSW Clear specification is made in the FIS field, and the same PSW is specified in the DST field (destination of a transfer operation), the contents of the PSW are cleared and the transfer operation is neglected.

Mnemonic table:

MNEMONIC	FIS FIELD	FLAG INITIALIZE AND SELECT
	21,20,19	
(NOP)	000	NO OPERATION
SPCPSW0	001	SPECIFY PSW 0 FOR OPERATION (DEFAULT)
SPCPSW1	010	SPECIFY PSW 1 FOR OPERATION
CLRPSW0	100	CLEAR PSW 0
CLRPSW1	101	CLEAR PSW 1
CLRPSW	110	CLEAR PSW 0 AND PSW 1

Field format:

```
_31_____27_26__________________15_14_13_12_10_9______5_4_______0_
|  OP(5)  |        CNT(12)        |P 2| Q(3)|  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    |_0__0|__M0_|__M1_|___DP0__|___DP1__|
    |_0__1__0__0|__EA_|___DP0__|___DP1__|
    |_0__1__0__1|__RP_|__M0_|___DP0__|FC|
    |_0__1__1__0|__RP_|__M1_|___DP1__|FC|
    |_0__1__1__1|__RP_|__M0_|__M1_|__L|FC|
    |_1__0__0__0__0|__BASE0_|__BASE1_|FC|
    |_1__0__0__0__1|____RPC____|--|__L|FC|
    |_1__0__0__1__0|P3|P2|EM|__BM_|__L|FC|
    |_1__0__0__1__1|__RW_|--------|__L|FC|
    |_1__0__1__0__0|___WT___|-----|__L|FC|
    |_1__0__1__0__1|___NF___|__WI_|__L|FC|
    |_1__0__1__1__0|___FIS__|__FD_|__L|--|
    |_1__0__1__1__1|_________SHV________|
    |_1__1__0|____________RPS___________|
    |_1__1__1|____________NAL___________|
```

L

Function:
Specifies the decrementing of the loop counter.

Notes for use:
- * This field is effective starting with the instruction after the one in which it is specified.
- * Refer to the hardware description section for more details on the use of the loop counter.

Mnemonic table:

MNEMONIC	L BIT	LOOP COUNTER OPERATION
	16	
(NOP)	0	NO OPERATION
DECLC	1	DECREMENT LOOP COUNTER

Field format:

```
 31     27 26                  15 14 13 12 10 9      5 4       0
|  OP(5)  |        CNT(12)        |P 2| Q(3)|  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / /  / / | \ \  \ \ \ \
        / / /  / /  / | \  \ \  \ \ \
      /  /  / /  / /  |  \ \  \ \  \  \
     /  /  /  / /  /  |  \  \ \  \  \  \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    |0  0 | M0  |  M1 |   DP0  |   DP1  |
    |0  1  0  0 |  EA |   DP0  |   DP1  |
    |0  1  0  1 |  RP | M0  |  DP0   |FC|
    |0  1  1  0 |  RP | M1  |  DP1   |FC|
    |0  1  1  1 |  RP | M0  | M1  | L|FC|
    |1  0  0  0  0 | BASE0  | BASE1  |FC|
    |1  0  0  0  1 |    RPC    |--| L|FC|
    |1  0  0  1  0 |P3|P2|EM| BM  | L|FC|
    |1  0  0  1  1 | RW  |--------| L|FC|
    |1  0  1  0  0 |   WT   |-----| L|FC|
    |1  0  1  0  1 |   NF   | WI  | L|FC|
    |1  0  1  1  0 |   FIS  | FD  | L|--|
    |1  0  1  1  1 |       SHV          |
    |1  1  0 |          RPS             |
    |1  1  1 |          NAL             |
```

M0

Function:

Specifies the addressing mode to be used for accessing RAM0. Available modes are base pointer 0 only, index register 0 only, and base pointer 0 + index register 0.

Notes for use:

- After hardware reset, the default is base + index.
- In base + index mode, the lower 9 bits of the sum are used, and the carry (overflow) into the 10th bit is ignored.
- This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	M0 FIELD	SPECIFY RAM POINTER
(NON)	00	NO CHANGE IN SPECIFICATION
SPCBP0	01	BASE POINTER 0
SPCIX0	10	INDEX REGISTER 0
SPCBI0	11	BASE POINTER 0 + INDEX REGISTER 0 (DEFAULT)

Field format:

```
 31      27 26                        15 14 13 12 10 9       5 4        0
| OP(5)   |          CNT(12)          |P 2| Q(3)|  SRC(5)  |  DST(5)   |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
           | | | | | | | | | | | | |
           | | | | | | | | | | | | |
          / / / / /  /  |  \   \ \ \ \ \
         / / / /  |   |  |   |   | \ \ \ \
        / / / |   |   |  |   |   |  | \ \ \
       / / |  |   |   |  |   |   |  |  | \ \
      / |  |  |   |   |  |   |   |  |  |  | \
     |26 |25 |24 |23 |22 |21 |20 |19 |18 |17 |16 |15 |
     | 0   0 |   M0  |   M1    |   DP0    |    DP1    |
     | 0   1   0   0 |   EA    |   DP0    |    DP1    |
     | 0   1   0   1 |   RP    |   M0  |    DP0   |FC |
     | 0   1   1   0 |   RP    |   M1  |    DP1   |FC |
     | 0   1   1   1 |   RP    |   M0  |   M1  | L |FC |
     | 1   0   0   0   0 |   BASE0   |   BASE1  |FC |
     | 1   0   0   0   1 |      RPC      |-- | L |FC |
     | 1   0   0   1   0 |P3 |P2 |EM |   BM  | L |FC |
     | 1   0   0   1   1 |   RW  |---------- | L |FC |
     | 1   0   1   0   0 |    WT     |------ | L |FC |
     | 1   0   1   0   1 |    NF     |  WI   | L |FC |
     | 1   0   1   1   0 |    FIS    |  FD   | L |-- |
     | 1   0   1   1   1 |          SHV              |
     | 1   1   0 |              RPS                  |
     | 1   1   1 |              NAL                  |
```

M1

Function:

Specifies the addressing mode to be used for accessing RAM1. Available modes are base pointer 1 only, index register 1 only, and base pointer 1 + index register 1.

Notes for use:

- * After hardware reset, the default is base + index.
- * In base + index mode, the lower 9 bits of the sum are used, and the carry (overflow) into the 10th bit is ignored.
- * This field is effective starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	M1 FIELD	SPECIFY RAM POINTER
(NON)	00	NO CHANGE IN SPECIFICATION
SPCBP1	01	BASE POINTER 1
SPCIX1	10	INDEX REGISTER 1
SPCBI1	11	BASE POINTER 1 + INDEX REGISTER 1 (DEFAULT)

Field format:

```
_31_____27_26__________________15_14_13_12_10_9______5_4_______0_
| OP(5)  |        CNT(12)        |P 2| Q(3)| SRC(5) | DST(5)   |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |  DP0   |  DP1   |
    | 0  1  0  0| EA  |  DP0   |  DP1   |
    | 0  1  0  1| RP  | M0  |  DP0   |FC|
    | 0  1  1  0| RP  | M1  |  DP1   |FC|
    | 0  1  1  1| RP  | M0  | M1  | L|FC|
    | 1  0  0  0  0| BASE0  | BASE1  |FC|
    | 1  0  0  0  1|    RPC    |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    | 1  0  0  1  1| RW  |--------| L|FC|
    | 1  0  1  0  0|   WT   |-----| L|FC|
    | 1  0  1  0  1|   NF   | WI  | L|FC|
    | 1  0  1  1  0|  FIS   | FD  | L|--|
    | 1  0  1  1  1|        SHV         |
    | 1  1  0|          RPS             |
    | 1  1  1|          NAL             |
```

NAL

Function:

This field specifies 9 bits which will replace the lower 9 bits of the program counter (PC). The upper 4 bits of the PC will remain unchanged. Therefore, a local jump is performed, within a 512-word segment of instruction memory. For example, if this instruction is executed at location 4F3H, the extent over which this instruction can branch by using this field is 400H - 5FFH.

Notes for use:

- * Because of the pipelined nature of instruction execution, a branch instruction can not occupy the next instruction location.

Mnemonic table:

MNEMONIC	NAL FIELD 23,22,21,20,19,18,17,16,15	LOCAL BRANCH
JBLK imm	(imm)B	JUMP TO imm ADDRESS IN LOCAL BLOCK

* 0 =< imm =< 511

Field format:

```
 31     27 26                           15 14 13 12 10 9       5 4       0
| OP(5)   |          CNT(12)              |P 2| Q(3)|  SRC(5) |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | |
          | | | | | | | | | | | |
         / / / / / / | \ \ \ \ \
        / / / /  | | | | \ \ \ \
       / / /  |  | | | | | \ \ \
      / / |  |  | | | | | | \ \
     / | |  |  | | | | | | | \
   |26|25|24|23|22|21|20|19|18|17|16|15|
   | 0  0|  M0 |  M1 |  DP0  |   DP1   |
   | 0  1  0  0|  EA |  DP0  |   DP1   |
   | 0  1  0  1|  RP |  M0  |   DP0  |FC|
   | 0  1  1  0|  RP |  M1  |   DP1  |FC|
   | 0  1  1  1|  RP |  M0  |  M1 | L|FC|
   | 1  0  0  0  0|  BASE0  |  BASE1 |FC|
   | 1  0  0  0  1|    RPC     |--| L|FC|
   | 1  0  0  1  0|P3|P2|EM|  BM  | L|FC|
   | 1  0  0  1  1| RW |---------| L|FC|
   | 1  0  1  0  0|  WT    |-----| L|FC|
   | 1  0  1  0  1|  NF    |  WI | L|FC|
   | 1  0  1  1  0|  FIS   |  FD | L|--|
   | 1  0  1  1  1|         SHV          |
   | 1  1  0|            RPS             |
   | 1  1  1|            NAL             |
```

NF

Function:

This 3-bit field specifies the operation that will be performed when the NORM instruction is executed by the ALU. Possible specifications include truncating normalization, rounding normalization, floating to fixed point data conversion, and fixed point data (multiple bit) alignment.

Notes for use:

- * This specification is valid for the instruction in which it is specified, and it is latched until respecified.
- * After hardware reset, the default specification is truncating normalization (TRNORM = 010).

Mnemonic table:

MNEMONIC	NF FIELD 21,20,19	NORMALIZATION FORMAT SPECIFICATION
(NON)	000	NO CHANGE OF SPECIFICATION
TRNORM	010	TRUNCATING NORMALIZATION (DEFAULT)
RDNORM	100	ROUNDING NORMALIZATION
FLTFIX	110	CONVERT FLOATING TO FIXED POINT
FIXMA	111	FIXED POINT MULTIPLE* ALIGNMENT

* Multiple value is in SVR.

Field format:

```
   31     27 26                  15 14 13 12 10 9      5 4       0
  | OP(5)   |        CNT(12)        |P 2| Q(3)| SRC(5)  |  DST(5) |
  |-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
            | | | | | | | | | | | | |
            | | | | | | | | | | | | |
          / /  / / /  / | \  \ \ \  \ \
        /  / /  /  / /  |  \ \  \  \ \  \
      /  /  /  /  /  /  |  \  \  \  \  \  \
    /  /   /  /  /   /  |  \   \  \  \   \  \
  /   /  /   /   /  /   |   \  \   \   \  \   \
|26 |25 |24 |23 |22 |21 |20 |19 |18 |17 |16 |15 |
| 0   0 |  M0   |  M1   |    DP0    |    DP1    |
| 0   1   0   0 |  EA   |    DP0    |    DP1    |
| 0   1   0   1 |  RP   |  M0   |    DP0    |FC |
| 0   1   1   0 |  RP   |  M1   |    DP1    |FC |
| 0   1   1   1 |  RP   |  M0   |  M1   | L |FC |
| 1   0   0   0   0 |   BASE0   |   BASE1   |FC |
| 1   0   0   0   1 |      RPC      |-- | L |FC |
| 1   0   0   1   0 |P3 |P2 |EM |  BM   | L |FC |
| 1   0   0   1   1 |  RW   |-----------| L |FC |
| 1   0   1   0   0 |    WT     |-------| L |FC |
| 1   0   1   0   1 |    NF     |  WI   | L |FC |
| 1   0   1   1   0 |    FIS    |  FD   | L |-- |
| 1   0   1   1   1 |            SHV            |
| 1   1   0 |                RPS                |
| 1   1   1 |                NAL                |
```

P2

Function:

This 1-bit field directly controls (sets or resets) the state of the P2 output pin. If this bit is set to 1, the state of the output pin is high level; if 0, low level. This applies only if the ASP is in slave mode.

Notes for use:

* After hardware reset, the output of the P2 pin is set to a low level.
* If the ASP is in master mode, the P2 pin has a different function, and therefore this bit is meaningless.
* This specification takes effect starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	P2 BIT	P2 PIN CONTROL (SLAVE MODE ONLY)
	20	
CLRP2	0	CLEAR OUTPUT PORT PIN 2
SETP2	1	SET OUTPUT PORT PIN 2

Field format:

```
 31      27 26                       15 14 13 12 10 9        5 4         0
| OP(5)    |          CNT(12)           |P 2| Q(3)|  SRC(5)  |  DST(5)   |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
            | | | | | | | | | | | | |
            | | | | | | | | | | | | |
           / / / / /  / | \  \ \ \ \ \
          / / / / |   | |  |  | \ \ \ \
         / / / |  |   | |  |  |  | \ \ \
        / / |  |  |   | |  |  |  |  | \ \
       / |  |  |  |   | |  |  |  |  |  | \
      |26|25|24|23|22|21|20|19|18|17|16|15|
      | 0  0| M0  | M1  |  DP0  |   DP1  |
      | 0  1  0  0| EA  |  DP0  |   DP1  |
      | 0  1  0  1| RP  |  M0 |   DP0  |FC|
      | 0  1  1  0| RP  |  M1 |   DP1  |FC|
      | 0  1  1  1| RP  |  M0 |  M1 | L|FC|
      | 1  0  0  0  0| BASE0 | BASE1  |FC|
      | 1  0  0  0  1|   RPC    |--| L|FC|
      | 1  0  0  1  0|P3|P2|EM|  BM | L|FC|
      | 1  0  0  1  1| RW |--------| L|FC|
      | 1  0  1  0  0| WT    |-----| L|FC|
      | 1  0  1  0  1| NF    |  WI | L|FC|
      | 1  0  1  1  0| FIS   |  FD | L|--|
      | 1  0  1  1  1|        SHV        |
      | 1  1  0|          RPS            |
      | 1  1  1|          NAL            |
```

P3

Function:

This 1-bit field directly controls (sets or resets) the state of the P3 output pin. If this bit is set to 1, the state of the output pin is high level; if 0, low level. This applies only if the ASP is in slave mode.

Notes for use:

* After hardware reset, the output of the P3 pin is set to a low level.
* If the ASP is in master mode, the P3 pin has a different function, and therefore this bit is meaningless.
* This specification takes effect starting with the instruction after the one in which it is specified.

Mnemonic table:

MNEMONIC	P3 BIT 21	P3 PIN CONTROL (SLAVE MODE ONLY)
CLRP3	0	CLEAR OUTPUT PORT PIN 3
SETP3	1	SET OUTPUT PORT PIN 3

Field format:

```
 31       27 26                         15 14 13 12 10 9       5 4       0
|  OP(5)  |            CNT(12)          |P  2| Q(3)|  SRC(5)  |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
           | | | | | | | | | | | | |
           | | | | | | | | | | | | |
          / / / / / / | \ \ \ \ \ \
         / / / / | |  |  | | \ \ \ \
        / / /  | | |  |  | |  | \ \ \
       / / |   | | |  |  | |  |  | \ \
      / |  |   | | |  |  | |  |  |  | \
|26 |25 |24 |23 |22 |21 |20 |19 |18 |17 |16 |15 |
| 0   0 |  M0   |  M1   |    DP0    |    DP1    |
| 0   1   0   0 |  EA   |    DP0    |    DP1    |
| 0   1   0   1 |  RP   |  M0   |    DP0    |FC |
| 0   1   1   0 |  RP   |  M1   |    DP1    |FC |
| 0   1   1   1 |  RP   |  M0   |  M1   | L |FC |
| 1   0   0   0   0 |   BASE0   |   BASE1   |FC |
| 1   0   0   0   1 |      RPC      |-- | L |FC |
| 1   0   0   1   0 |P3 |P2 |EM |  BM   | L |FC |
| 1   0   0   1   1 |  RW   |---------- | L |FC |
| 1   0   1   0   0 |    WT     |------ | L |FC |
| 1   0   1   0   1 |    NF     |  WI   | L |FC |
| 1   0   1   1   0 |    FIS    |  FD   | L |-- |
| 1   0   1   1   1 |            SHV            |
| 1   1   0 |                RPS                |
| 1   1   1 |                NAL                |
```

RP

Function:

The RP field specifies the modification operations performed on the ROM pointer (RP), which is used for addressing the data ROM area. This field is 2 bits long, and can specify the following operations to be performed on the (RP): increment, decrement, and add 2**N.

Notes for use:

- * This field is effective starting with the instruction after the one in which it is specified.
- * For the special case of addition of 2**n to the RP, the value of n is specified in the RPC field.

Mnemonic table:

MNEMONIC	RP FIELD	ROM POINTER MODIFICATION OPERATION
	22,21	
(NOP)	00	NO OPERATION
INCRP	01	INCREMENT ROM POINTER
DECRP	10	DECREMENT ROM POINTER
INCBRP	11	INCREMENT SPECIFIED BIT OF ROM POINTER (I.E. ADD 2**N)

Field format:

```
 31     27 26                  15 14 13 12 10 9      5 4       0
| OP(5)   |        CNT(12)        |P 2| Q(3)|  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / /  / / | \ \  \ \ \ \
        / /  / / /  / | \  \ \ \  \ \
       / /  / /  / /  |  \ \  \ \  \ \
      /  / /  /  / /  |  \ \  \  \ \  \
     /  / /  /  / /   |   \ \  \  \ \  \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0| M0  | M1  |  DP0   |  DP1   |
    | 0  1  0  0| EA  |  DP0   |  DP1   |
    | 0  1  0  1| RP  | M0  |  DP0   |FC|
    | 0  1  1  0| RP  | M1  |  DP1   |FC|
    | 0  1  1  1| RP  | M0  | M1  | L|FC|
    | 1  0  0  0  0| BASE0  | BASE1  |FC|
    | 1  0  0  0  1|    RPC    |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    | 1  0  0  1  1| RW  |--------| L|FC|
    | 1  0  1  0  0|  WT    |-----| L|FC|
    | 1  0  1  0  1|  NF    | WI  | L|FC|
    | 1  0  1  1  0|  FIS   | FD  | L|--|
    | 1  0  1  1  1|       SHV          |
    | 1  1  0|          RPS             |
    | 1  1  1|          NAL             |
```

RPC

Function:

This 4-bit field is used to specify the value of N used for the special addition of 2**N to the ROM pointer (RP).

Notes for use:

* This value is latched, starting with the instruction after the one in which it is specified, and will be used if the RP field specifies the 2**N addition operation. The value of N can be 1 - 9. If 0 is specified, the contents of RP will be incremented when the 2**N addition is specified.
* 0 is the default value after hardware reset.

Mnemonic table:

MNEMONIC	RPC FIELD 21,20,19,18	SPECIFY 2**N ADDITIVE VALUE
BITRP imm	(imm)B	SPECIFY N FOR ADDING 2**N TO ROM POINTER

* imm (=n) is 0 through 9.

Field format:

```
 31     27 26                      15 14 13 12 10 9     5 4       0
|  OP(5)  |         CNT(12)          |P 2| Q(3)|  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | |
          | | | | | | | | | | | |
         / / / / / / | \ \ \ \ \ \
        / / / / | | | | | \ \ \ \ \
       / / / | | | | | | | \ \ \ \
      / / | | | | | | | | | | \ \
     / | | | | | | | | | | | | \

|26 |25 |24 |23 |22 |21 |20 |19 |18 |17 |16 |15 |
| 0   0 |  M0   |  M1   |    DP0    |    DP1    |
| 0   1   0   0 |  EA   |    DP0    |    DP1    |
| 0   1   0   1 |  RP   |  M0   |    DP0    |FC |
| 0   1   1   0 |  RP   |  M1   |    DP1    |FC |
| 0   1   1   1 |  RP   |  M0   |  M1   | L |FC |
| 1   0   0   0   0 |  BASE0    |  BASE1    |FC |
| 1   0   0   0   1 |      RPC      |-- | L |FC |
| 1   0   0   1   0 |P3 |P2 |EM |  BM   | L |FC |
| 1   0   0   1   1 |  RW   |-----------| L |FC |
| 1   0   1   0   0 |   WT      |-------| L |FC |
| 1   0   1   0   1 |   NF      |  WI   | L |FC |
| 1   0   1   1   0 |   FIS     |  FD   | L |-- |
| 1   0   1   1   1 |           SHV             |
| 1   1   0 |              RPS                  |
| 1   1   1 |              NAL                  |
```

RPS

Function:

This field specifies a 9-bit data ROM address that may be used once by a subsequent instruction. Only the lower half of the data ROM may be accessed in this manner. When a subsequent instruction specifies ROM in the SRC field, the 9-bit RPS field is used as the data ROM address, with the MSB of the address set to 0. The data at the address specified in the RPS field will be transferred through the internal bus.

Notes for use:

* This field is valid starting with the instruction after the one in which it is specified.
* This ROM address is available only once, the first time ROM is specified in the SRC field after the RPS specification. Once an instruction uses ROM as the SRC field, accessing the address specified in the RPS field, then the value that had previously been in the RP register will be valid again.

Mnemonic table:

MNEMONIC	RPS FIELD	SPECIFY IMMEDIATE
	23,22,21,20,19,18,17,16,15	ROM ADDRESS
SPCRA imm	(imm)B	SPECIFY IMMEDIATE ROM ADDRESS

* 0 =< imm =< 511

Field format:

```
_31_____27_26_________________15_14_13_12_10__9_____5_4________0_
|  OP(5)  |        CNT(12)        |P 2| Q(3)|  SRC(5) | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    |_0__0|__M0_|__M1_|___DP0__|___DP1__|
    |_0__1__0__0|__EA_|___DP0__|___DP1__|
    |_0__1__0__1|__RP_|__M0_|___DP0__|FC|
    |_0__1__1__0|__RP_|__M1_|___DP1__|FC|
    |_0__1__1__1|__RP_|__M0_|_M1__|_L|FC|
    |_1__0__0__0__0|_BASE0__|_BASE1__|FC|
    |_1__0__0__0__1|___RPC_____|--|_L|FC|
    |_1__0__0__1__0|P3|P2|EM|_BM__|_L|FC|
    |_1__0__0__1__1|_RW__|--------|_L|FC|
    |_1__0__1__0__0|__WT____|-----|_L|FC|
    |_1__0__1__0__1|__NF____|_WI__|_L|FC|
    |_1__0__1__1__0|__FIS___|_FD__|_L|--|
    |_1__0__1__1__1|________SHV_________|
    |_1__1__0|____________RPS___________|
    |_1__1__1|____________NAL___________|
```

RW

Function:

This 2-bit field specifies an external memory read or write operation. In the case of master mode, a 32-bit transfer is performed through pins D0 - D31, and in the case of slave mode, an 8-bit transfer is performed (1, 2, 3, or 4 times, depending on the length of data to be transferred, as specified in the status register) through pins D0 - D7.

Notes for use:

* Only instructions contained in the internal instruction ROM can perform this operation. External instructions are prohibited from performing this operation.
* If an internal instruction performs a branch to an external instruction, the next internal instruction is not permitted to specify RW, due to the pipelined operation of the ASP (this would require performing the RW and the external instruction fetch simultaneously on the external data bus).
* The external memory access is initiated during the instruction in which it is specified.

Mnemonic table:

MNEMONIC	RW FIELD 21,20	OPERATION FOR EXTERNAL DATA MEMORY
(NOP)	00	NO OPERATION
RD	01	READ
WR	10	WRITE
	11	USE PROHIBITED

Field format:

```
_31_____27_26__________________15_14_13_12_10_9______5_4______0__
|  OP(5)  |       CNT(12)         |P 2| Q(3)| SRC(5)  | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / /  |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0| M0  | M1  |  DP0   |  DP1   |
    | 0  1  0  0| EA  |  DP0   |  DP1   |
    | 0  1  0  1| RP  | M0  |  DP0   |FC|
    | 0  1  1  0| RP  | M1  |  DP1   |FC|
    | 0  1  1  1| RP  | M0  |  M1 | L|FC|
    | 1  0  0  0  0| BASE0  | BASE1  |FC|
    | 1  0  0  0  1|   RPC     |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM| BM  | L|FC|
    | 1  0  0  1  1| RW  |--------| L|FC|
    | 1  0  1  0  0|  WT    |-----| L|FC|
    | 1  0  1  0  1|  NF    | WI  | L|FC|
    | 1  0  1  1  0|  FIS   | FD  | L|--|
    | 1  0  1  1  1|        SHV         |
    | 1  1  0|           RPS            |
    | 1  1  1|           NAL            |
```

SHV

Function:

This 7-bit field specifies a shift value for use by the barrel shifter in the ALU functional block area. The highest bit specifies left or right shift operation (although it is ignored for SHLM, SHRM, and SHRAM), and the remaining bits specify the number of bits to be shifted (0 - 46).

Notes for use:

* This specification is valid for the instruction in which it is specified, and it is latched in the SVR register until changed again.
* SHV/SVR value will shift Q for the SHLM, SHRM, SHRAM, and NORM (FIXMA only) ALU operations. The direction bit will be ignored in these cases.
* SHV/SVR value will shift P for ADD, SUB, ADDC, SUBC, AND, OR, and XOR ALU operations.
* If a floating point ALU operation is being performed, the SHV/SVR value will be ignored and the barrel shifter will automatically adjust P or Q to align their mantissas.

Mnemonic table:

MNEMONIC	SHV FIELD		SET SHIFT VALUE TO SVR
	21	20,19,18,17,16,15	
SETSVL imm	0	(imm)B	imm BITS LEFT SHIFT (DEFAULT)
SETSVR imm	1	(imm)B	imm BITS RIGHT SHIFT

* 0 =< imm =< 46

Field format:

```
 31     27 26                          15 14 13 12 10 9       5 4       0
|  OP(5)  |            CNT(12)            |P 2| Q(3) |  SRC(5)  |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

26	25	24	23	22	21	20	19	18	17	16	15
0	0	M0		M1		DP0			DP1		
0	1	0	0	EA		DP0			DP1		
0	1	0	1	RP		M0		DP0			FC
0	1	1	0	RP		M1		DP1			FC
0	1	1	1	RP		M0		M1		L	FC
1	0	0	0	0	BASE0			BASE1			FC
1	0	0	0	1	RPC				--	L	FC
1	0	0	1	0	P3	P2	EM	BM		L	FC
1	0	0	1	1	RW		--------			L	FC
1	0	1	0	0	WT			-----		L	FC
1	0	1	0	1	NF			WI		L	FC
1	0	1	1	0	FIS			FD		L	--
1	0	1	1	1	SHV						
1	1	0	RPS								
1	1	1	NAL								

WI

Function:

This 2-bit field specifies the format to be used when transferring data from the internal bus to a working register (i.e, when WRn is specified in DST field).

Notes for use:

- * This specification is valid starting with the instruction after the one in which it is specified, and it is latched until changed again by the use of this field or a reset.
- * After hardware reset, the default specification is normal transfer mode.

Mnemonic table:

MNEMONIC	WI FIELD 18,17	SPECIFICATION OF TRANSFER FORMAT WHEN DATA IS MOVED FROM IB TO WR
(NON)	00	NO CHANGE OF SPECIFICATION
BWRL24	01	TRANSFER LOW 24 BITS OF MANTISSA TO HIGH 24 BITS
BWRORD	10	ORDINARY TRANSFER (DEFAULT)
	11	USE PROHIBITED

Field format:

```
 31     27 26                    15 14 13 12 10 9       5 4        0
| OP(5)  |          CNT(12)        |P 2| Q(3)|  SRC(5)  |  DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

26	25	24	23	22	21	20	19	18	17	16	15
0	0	M0		M1		DP0			DP1		
0	1	0	0	EA		DP0			DP1		
0	1	0	1	RP		M0		DP0			FC
0	1	1	0	RP		M1		DP1			FC
0	1	1	1	RP		M0		M1		L	FC
1	0	0	0	0	BASE0			BASE1			FC
1	0	0	0	1	RPC				--	L	FC
1	0	0	1	0	P3	P2	EM	BM		L	FC
1	0	0	1	1	RW		---------			L	FC
1	0	1	0	0	WT			------		L	FC
1	0	1	0	1	NF			**WI**		L	FC
1	0	1	1	0	FIS			FD		L	--
1	0	1	1	1	SHV						
1	1	0	RPS								
1	1	1	NAL								

WT

Function:

This 3-bit field specifies the format to be used when transferring data from a working register to the internal bus (i.e, when WRn is specified in the SRC field).

Notes for use:

- * This specification is valid starting with the instruction after the one in which it is specified, and it is latched until changed again by the use of this field or a reset.
- * After hardware reset, the default specification is normal transfer mode.

Mnemonic table:

MNEMONIC	WT FIELD	SPECIFICATION OF TRANSFER FORMAT WHEN
	21,20,19	DATA IS MOVED FROM WR TO IB
(NON)	000	NO CHANGE OF SPECIFICATION
WRBORD	001	ORDINARY TRANSFER (DEFAULT)
WRBL24	010	LOW 24 BITS OF MANTISSA TO HIGH 24
WRBL23	011	LOW 23 BITS (BIT 23=0) TO HIGH 24
WRBEL8	100	EXPONENT PART TO MANTISSA LOW 8 BITS
WRBL8E	101	MANTISSA LOW 8 BITS TO EXPONENT PART
WRBXCH	110	EXCHANGE HIGH 8 BITS OF MANTISSA WITH LOW 8 BITS OF MANTISSA
WRBBRV	111	BIT REVERSE ENTIRE MANTISSA

Field format:

```
 31     27 26                  15 14 13 12 10 9      5 4       0
|  OP(5)  |        CNT(12)        |P 2| Q(3)|  SRC(5) | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
          | | | | | | | | | | | | |
          | | | | | | | | | | | | |
         / / / / /  / | \  \ \ \ \ \
        / / / / |  |  |  |  | \ \ \ \
       / / / |  |  |  |  |  |  | \ \ \
      / / |  |  |  |  |  |  |  |  | \ \
     / |  |  |  |  |  |  |  |  |  | \
    |26|25|24|23|22|21|20|19|18|17|16|15|
    | 0  0|  M0 |  M1 |  DP0   |  DP1   |
    | 0  1  0  0|  EA |  DP0   |  DP1   |
    | 0  1  0  1|  RP |  M0 |  DP0   |FC|
    | 0  1  1  0|  RP |  M1 |  DP1   |FC|
    | 0  1  1  1|  RP |  M0 |  M1 | L|FC|
    | 1  0  0  0 .0|  BASE0 |  BASE1 |FC|
    | 1  0  0  0  1|    RPC    |--| L|FC|
    | 1  0  0  1  0|P3|P2|EM|  BM | L|FC|
    | 1  0  0  1  1|  RW |--------| L|FC|
    | 1  0  1  0  0|   WT   |-----| L|FC|
    | 1  0  1  0  1|   NF   |  WI | L|FC|
    | 1  0  1  1  0|   FIS  |  FD | L|--|
    | 1  0  1  1  1|         SHV        |
    | 1  1  0|           RPS           |
    | 1  1  1|           NAL           |
```

4.3.5 SRC field.

The SRC field is a 5-bit field which specifies the source register for the transfer operation that can occur in an OP or branch instruction. The contents of the specified register will be put onto the internal bus for transfer to the register specified in the DST field.

SRC can specify any of 32 different registers. If the NON source specification is used, the internal bus bits are all set to 1, for transfer to the destination.

If a 32-bit register is specified in the SRC field for a transfer to a 55-bit register, the upper 32 bits of the destination receive the data from that register, and the lower 23 bits are all set to 0.

If a working register is specified in the SRC field, it is put onto the internal bus according to the transfer format specified by the WT field.

If the M register is specified in the SRC field, and the destination is a 32-bit register, then the most significant 32 bits of the M register are moved to the destination. However, if the destination is a 55-bit register (i.e. a working register), then the full 55 bits of the M register will be transferred.

If ML is specified in the SRC field, then the lower 24 bits of the M register will be put into the lower 24 bits of the 32 bit internal bus, while the 8 MSBs (i.e. exponent) of the M register will be put into the 8 MSBs (i.e. exponent) of the internal bus.

Table 4-6 lists the SRC field specifications.

Table 4-6. SRC field specifications.

MNEMONIC	SRC FIELD	SELECTED SOURCE REGISTER
	9,8,7,6,5	
NON	00000	NO SOURCE SELECTED
RP	00001	ROM POINTER
PSW0	00010	PROGRAM STATUS WORD 0
PSW1	00011	PROGRAM STATUS WORD 1
SVR	00100	SVR (SHIFT VALUE REGISTER)
SR	00101	STATUS REGISTER
LC	00110	LOOP COUNTER
STK	00111	TOP OF STACK
M	01000	M REGISTER (MULTIPLIER OUTPUT)
ML	01001	LOW 24 BITS OF M REGISTER
ROM	01010	DATA ROM OUTPUT
TR	01011	TEMPORARY REGISTER
AR	01100	EXTERNAL ADDRESS REGISTER
SI	01101	SERIAL INPUT REGISTER
DR	01110	DATA REGISTER
DRS	01111	DATA REGISTER FOR SLAVE
WR0	10000	WORKING REGISTER 0
WR1	10001	WORKING REGISTER 1
WR2	10010	WORKING REGISTER 2
WR3	10011	WORKING REGISTER 3
WR4	10100	WORKING REGISTER 4
WR5	10101	WORKING REGISTER 5
WR6	10110	WORKING REGISTER 6
WR7	10111	WORKING REGISTER 7
RAM0	11000	RAM BLOCK 0
RAM1	11001	RAM BLOCK 1
BP0	11010	BASE POINTER 0
BP1	11011	BASE POINTER 1
IX0	11100	INDEX REGISTER 0
IX1	11101	INDEX REGISTER 1
K	11110	K REGISTER
L	11111	L REGISTER

4.3.6 DST field.

The DST field is a 5-bit field which specifies the destination register for the transfer operation that can occur in an OP, branch, or load immediate data instruction. The data on the internal bus (which comes from the register specified in the SRC field) will be written into the specified register.

DST can specify any of 32 different registers. If the NON specification is used, the contents of the register specified in the SRC field are output only to the internal bus, and are not written to any other register.

If a working register is specified in the DST field, it is transferred from the internal bus into the working register according to the transfer format specified by the WI field.

If the temporary register (TR) is specified in the DST field, the 32 bits of the internal bus are input to the TR. If TRE is specified in the DST field, the lower 8 bits of the internal bus are input to the upper 8 bits (exponent) of the TR, and the lower 24 bits of the TR retain their former value.

If a pointer register (RP, BP0, IX0, etc.) is specified in the DST field and a modification of that pointer register is specified in the CNT field of the same instruction, the transfer operation takes precedence, and the pointer modification function is not performed.

If LKR0 is specified in the DST field, then the data on the internal bus is written into the L register in parallel with data at the current address of RAM0 being written into the K register. If KLR1 is specified in the DST field, then the data on the internal bus is written into the K register in parallel with the data at the current address of RAM1 being written into the L register.

If the same register is specified in both the SRC field and the DST field, the contents of the register is put onto the internal bus, and the contents are not changed at the end of the instruction. An exception to this is the case where a working register (WR) is specified. In this case, the contents are varied according to the transfer formats specified by the WI and WT fields.

Table 4-7 shows the various DST field specifications.

Table 4-7. DST field specifications.

MNEMONIC	DST FIELD 4,3,2,1,0	SELECTED DESTINATION REGISTER
NON	00000	NO DESTINATION SELECTED
RP	00001	ROM POINTER
PSW0	00010	PROGRAM STATUS WORD 0
PSW1	00011	PROGRAM STATUS WORD 1
SVR	00100	SVR (SHIFT VALUE REGISTER)
SR	00101	STATUS REGISTER
LC	00110	LOOP COUNTER
STK	00111	TOP OF STACK
LKR0	01000	L REGISTER (RAM 0 TO K REGISTER)
KLR1	01001	K REGISTER (RAM 1 TO L REGISTER)
TRE	01010	EXPONENT PART OF TEMPORARY REG.
TR	01011	TEMPORARY REGISTER
AR	01100	EXTERNAL ADDRESS REGISTER
SO	01101	SERIAL OUTPUT REGISTER
DR	01110	DATA REGISTER
DRS	01111	DATA REGISTER FOR SLAVE
WR0	10000	WORKING REGISTER 0
WR1	10001	WORKING REGISTER 1
WR2	10010	WORKING REGISTER 2
WR3	10011	WORKING REGISTER 3
WR4	10100	WORKING REGISTER 4
WR5	10101	WORKING REGISTER 5
WR6	10110	WORKING REGISTER 6
WR7	10111	WORKING REGISTER 7
RAM0	11000	RAM BLOCK 0
RAM1	11001	RAM BLOCK 1
BP0	11010	BASE POINTER 0
BP1	11011	BASE POINTER 1
IX0	11100	INDEX REGISTER 0
IX1	11101	INDEX REGISTER 1
K	11110	K REGISTER
L	11111	L REGISTER

4.4 Data Load Instruction

This is the Load Data Immediate (LDI) instruction that performs a direct load of immediate data to virtually any register in the ASP. This instruction can specify 24 bits of immediate data. The register in which the data will be loaded is specified in the DST field, as in the OP instruction.

If the TRE register is specified in the DST field, the lower 8 bits of the immediate data field are input to the upper 8 bits (exponent) of the TR. The lower 24 bits of the TR are not affected.

Figure 4-4. LDI instruction format.

```
 31 29 28                                              5 4        0
|LDI 3|                     IM(24)                     | DST(5)  |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

Table 4-8. Load immediate data instruction type field.

TYPE	LD FIELD 31,30,29	INSTRUCTION
LDI	111	LOAD IMMEDIATE DATA

4.5 Branch Instruction

The branch instruction type includes the unconditional jump, conditional jump, subroutine call, and return instructions. The target address of the jump (NAL field) is specified in 13 bits; the most significant bit determines whether the target is in internal or external instruction space. Figure 4-5 shows the configuration of the Branch Type Instruction. Table 4-9 shows that the B field is always "1101".

Figure 4-5. Branch type instruction format.

```
 31   28 27                          15 14     10 9       5 4       0
| B(4) |              NA(13)            |  C(5)  |  SRC(5) |  DST(5) |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
```

Table 4-9. Branch instruction type field.

TYPE	B FIELD	INSTRUCTION
	31,30,29,28	
B *	1101	BRANCH TYPE INSTRUCTION

* TAKES THE MNEMONIC OF THE C (CONDITION) FIELD

All branch instructions are delayed jumps, due to the pipelined architecture of the ASP. This also applies to the local jump feature of the OP instruction, using the NAL field in the CNT field.

The 5-bit C field of the branch instruction specifies unconditional jump, subroutine call, return instruction, and the condition of the conditional jump. The definitions of the various bit combinations in the C field are shown in Table 4.10.

A data transfer operation may be included as part of a branch instruction, due to the fact that the SRC and DST fields are available for use in the branch instruction.

Due to the pipelined architecture of the ASP, a branch instruction may not occur immediately following another branch instruction. In this case the results are undefined.

Similarly, if a branch instruction specifies a branch to an external instruction address, the instruction immediately following the one in which the branch is specified may not have an RW specification. This would cause a conflict between the RW operation, which would be in the pipeline when the branch executes, and the instruction fetch from external instruction memory.

4.5.1 C Field

The 5-bit C field of the branch instruction specifies unconditional jump, subroutine call, return instruction, and the condition of the conditional jump. The definitions of the various bit combinations in the C field are shown in Table 4-10.

Table 4-10. Branch instruction C field specifications.

MNEMONIC	C FIELD	JUMP WITH CONDITION
	14,13,12,11,10	
JMP	00000	JUMP UNCONDITIONALLY
CALL	00001	SUBROUTINE CALL
RET	00010	RETURN FROM INTRPT OR SUBROUTINE
JNZRP	00011	JUMP IF ROM POINTER NOT ZERO
JZ0	00100	JUMP IF ZERO FLAG 0 IS SET
JNZ0	00101	JUMP IF ZERO FLAG 0 IS RESET
JZ1	00110	JUMP IF ZERO FLAG 1 IS SET
JNZ1	00111	JUMP IF ZERO FLAG 1 IS RESET
JC0	01000	JUMP IF CARRY FLAG 0 IS SET
JNC0	01001	JUMP IF CARRY FLAG 0 IS RESET
JC1	01010	JUMP IF CARRY FLAG 1 IS SET
JNC1	01011	JUMP IF CARRY FLAG 1 IS RESET
JS0	01100	JUMP IF SIGN FLAG 0 IS SET
JNS0	01101	JUMP IF SIGN FLAG 0 IS RESET
JS1	01110	JUMP IF SIGN FLAG 1 IS SET
JNS1	01111	JUMP IF SIGN FLAG 1 IS RESET
JV0	10000	JUMP IF OVERFLOW FLAG 0 IS SET
JNV0	10001	JUMP IF OVERFLOW FLAG 0 IS RESET
JV1	10010	JUMP IF OVERFLOW FLAG 1 IS SET
JNV1	10011	JUMP IF OVERFLOW FLAG 1 IS RESET
JEV0	10100	JUMP IF EXP. OVFL FLAG 0 IS SET
JEV1	10101	JUMP IF EXP. OVFL FLAG 1 IS SET
JNFSI	10110	JUMP IF SI REGISTER IS NOT FULL
JNESO	10111	JUMP IF SO REGISTER IS NOT EMPTY
JIP0	11000	JUMP IF INPUT PORT 0 IS ON
JIP1	11001	JUMP IF INPUT PORT 1 IS ON
JNZIX0	11010	JUMP IF INDEX REGISTER 0 NONZERO
JNZIX1	11011	JUMP IF INDEX REGISTER 1 NONZERO
JNZBP0	11100	JUMP IF BASE POINTER 0 NONZERO
JNZBP1	11101	JUMP IF BASE POINTER 1 NONZERO
JRDY	11110	JUMP IF READY IS ON
JRQM	11111	JUMP IF REQUEST FOR MASTER IS ON

4.5.2 NA Field

The 13-bit NA field specifies the target address of the jump; the most significant bit determines whether the target is in internal or external instruction space. Since the capacity of the internal instruction ROM is 2K words, the addresses that may be specified for an internal branch are 0 - 7FFH. For an external branch, the address range is 1000H - 1FFFH. Because the PC does not provide counter operation of the most significant bit, movement between internal and external instruction spaces must be done by use of a branch instruction. After hardware reset, the PC starts operation at location 0 of the internal instruction ROM.

The instruction address space is shown below:

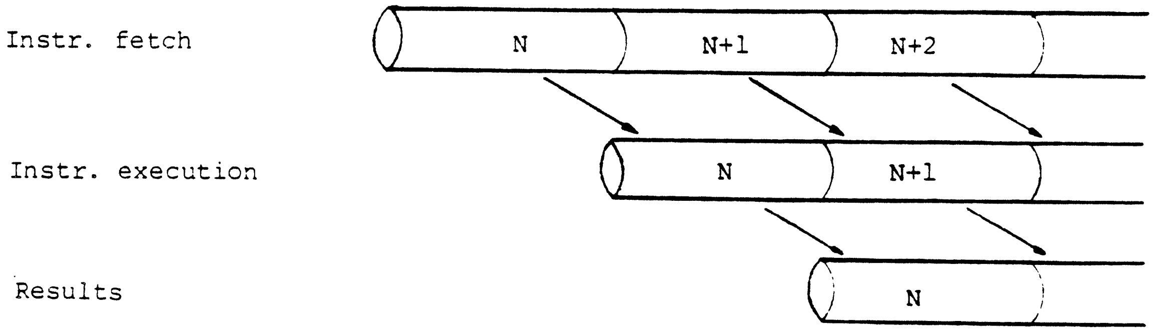
```
0000H     |------------------------------------|
          |  2K Internal Instruction ROM       |
          |                                    |
07FFH     |____________________________________|
0800H     |                                    |
          |  Unavailable Space                 |
          |                                    |
0FFFH     |____________________________________|
1000H     |                                    |
          |  4K Space Available For Use        |
          |  As Instruction Space In           |
          |  External Memory Space             |
          |                                    |
          |                                    |
          |                                    |
1FFFH     |____________________________________|
```

APPENDIX A. INSTRUCTION PIPELINING

As indicated in Chapter 4, the result of an ALU operation is, in most cases, available in the designated working register two instruction cycles after the instruction was fetched. At first glance, this may appear to be a shortcoming. However, it is actually a characteristic of the uPD77230's highly efficient instruction pipelining scheme.

Figure A-1 shows the nature of pipelined instruction execution in the uPD77230. Conceptually, there exists three "pipelines:" an instruction fetch pipeline, an instruction execution pipeline, and a results pipeline. Instructions are fetched in accordance with the sequence indicated by the program counter (PC). One instruction is fetched each instruction cycle. Likewise, one instruction is executed each instruction cycle. However, there is a one cycle delay between the fetch and the execution of a particular instruction, as well as a one cycle delay between the execution of an instruction and the availability of the result in a working register. This accounts for the two cycle "delay" mentioned in the ALU operations section in Chapter 4.

Figure A-1. Pipelined nature of instruction execution.



This "delay" is not a true delay in that the result of one instruction is available as an input for the next instruction. Figure A-2 shows the timing of the execution of a series of instructions. Instruction N generates a result which is required as an input for instruction N+1. In this case, the results from instruction N are not stored in a working register, but instead are routed directly back to the input of the ALU for the next instruction. If instruction N+1 were the last in the series of instructions, then the final result would be available in a working register two cycles later, as shown.

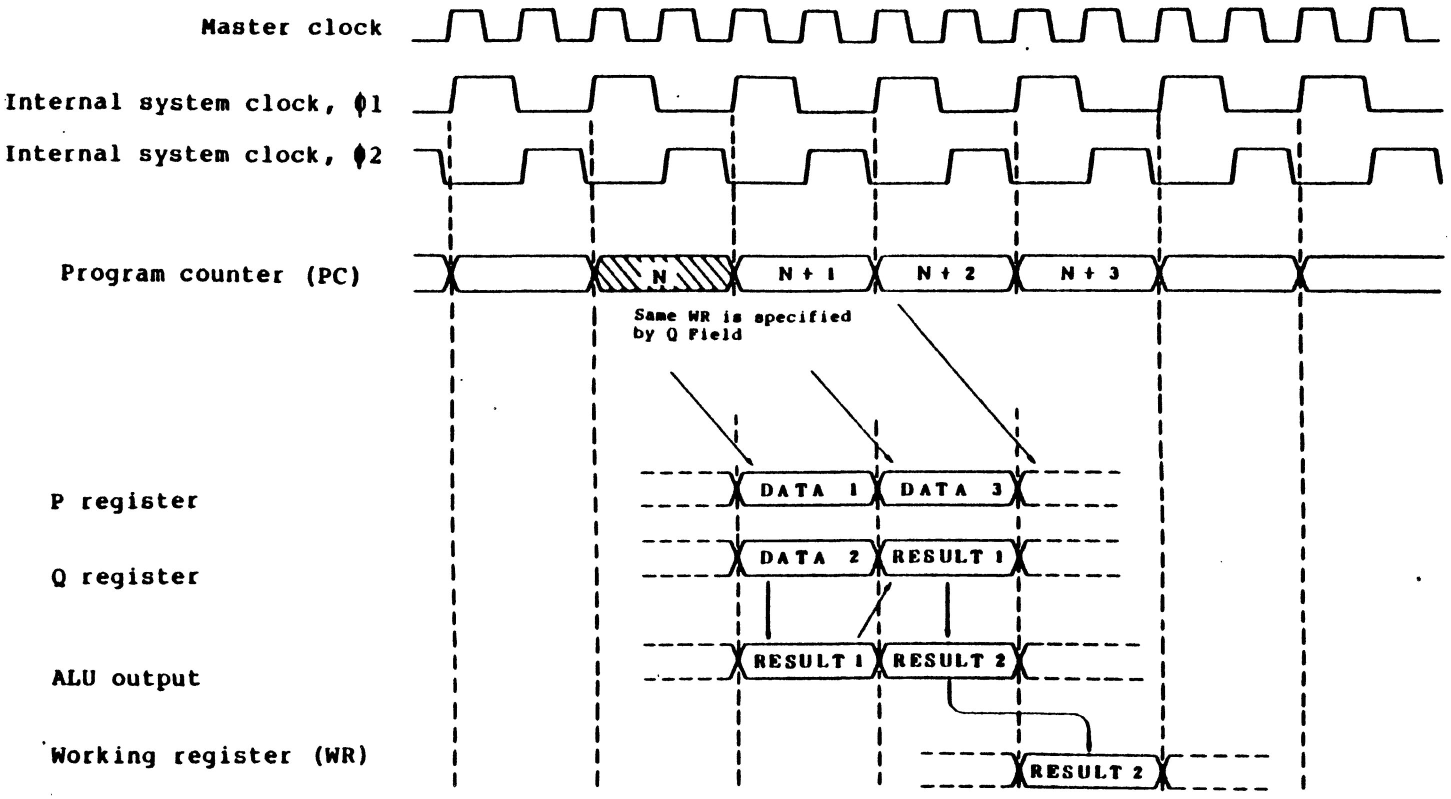


Figure A-2. ALU operation timing for a series of instructions performing continuous accumulation.

NEC

NEC Electronics Inc.

CORPORATE HEADQUARTERS

401 Ellis Street
P.O. Box 7241
Mountain View, CA 94039
TEL 415-960-6000
TWX 910-379-6985

For Literature Call Toll Free: 1-800-632-3531
1-800-632-3532 (In California)

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics Inc. The information in this document is subject to change without notice. Devices sold by NEC Electronics Inc. are covered by the warranty and patent indemnification provisions appearing in NEC Electronics Inc. Terms and Conditions of Sale only. NEC Electronics Inc. makes no warranty, express, statutory, implied, or by description, regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. NEC Electronics Inc. makes no warranty of merchantability or fitness for any purpose. NEC Electronics Inc. assumes no responsibility for any errors that may appear in this document. NEC Electronics Inc. makes no commitment to update or to keep current the information contained in this document.

NECEL-000611-0386
STOCK NO. 500291