NEC Electronics Inc.

NEC

ADVANCE PRODUCT INFORMATION

March 1987

μPD77C25/ μPD77P25

16-Bit Fixed Point CMOS Digital Signal Processor

User's Manual

FABIO MONTORO

TABLE OF CONTENTS

CHAPTER 1 OVERVIEW

1.1 Overview

The uPD77C25/77P25 is a 16-bit fixed-point CMOS signal processor intended for real-time digital processing of audio signals.

It consists of a parallel multiplier (16 bits x 16 bits -> 31 bits), an ALU (16 bits), an instruction ROM (2,048 x 24 bits), a data ROM (1,024 x 16 bits), a data RAM (256 x 16 bits), I/O ports, and others. All instructions consist of 24 bits or one word instruction are executed in 122ns (at fCLK =16.384MHz) including product and sum computations.

Since signals that interfaces with the host CPU are provided, the uPD77C25/77P25 can cover a variety of applications, serving as an I/O processor. Moreover, it can also be used as a single-chip CPU.

The uPD77C25/77P25 provides an instruction ROM four times larger than that of the existing uPD7720 signal processor. Additionally, it has a data ROM and data RAM, both of which are two times larger, and a processing speed two times faster. Furthermore, the uPD77C25/77P25 can replace the uPD7720 as they have the same pin connections.

The instruction set of the uPD77C25/77P25 is upward-compatible with that of the uPD7720 at the assembler source program level.

The uPD77C25 is a version with on-chip resources including the instruction ROM and data ROM are constructed in mask ROMs; the uPD77P25 has an UVEPROM whose contents are erasable by ultraviolet light.

Note: In this document, the uPD77C25 refers to both the uPD77C25 and uPD77P25 unless otherwise specified.

1.2 Features

- o Biquad Digital Filter (with sampling performed at 8KHz) — Equivalent to 113 filters
- o On-chip exclusive parallel multiplier — 16 bits x 16 bits -> 31 bits
- o Instruction ROM — 2,048 words x 24 bits
- o Data ROM — 1,024 words x 16 bits
- o Data RAM — 256 words x 16 bits
- o Dual accumulator method
- o On-chip serial input and serial output interfaces
- o On-chip host CPU bus interface
- o On-chip DMA interface
- o Upward-compatible with the uPD7720 at assembler source program level
- o Pin-compatible with uPD7720

Application examples

(1) Digital filter
(2) DTMF (Dual Tone Multi-Frequency Receiver)
(3) Fast Fourier Transformation (FFT)
(4) Spectrum analyzer for voice recognition
(5) MODEM (Modulator and Demodulator)
(6) Speech synthesis device
(7) ADPCM (Adaptive differential pulse-coded modulation)
(8) Others

1.3 Internal Block Diagram

Fig. 1-1 shows the internal block diagram of the uPD77C25. The internal circuitry consists of a multiplier, an ALU and its peripherals, a data memory section (made up of a data ROM and a RAM), an instruction ROM, a parallel interface circuit, and a serial interface circuit.

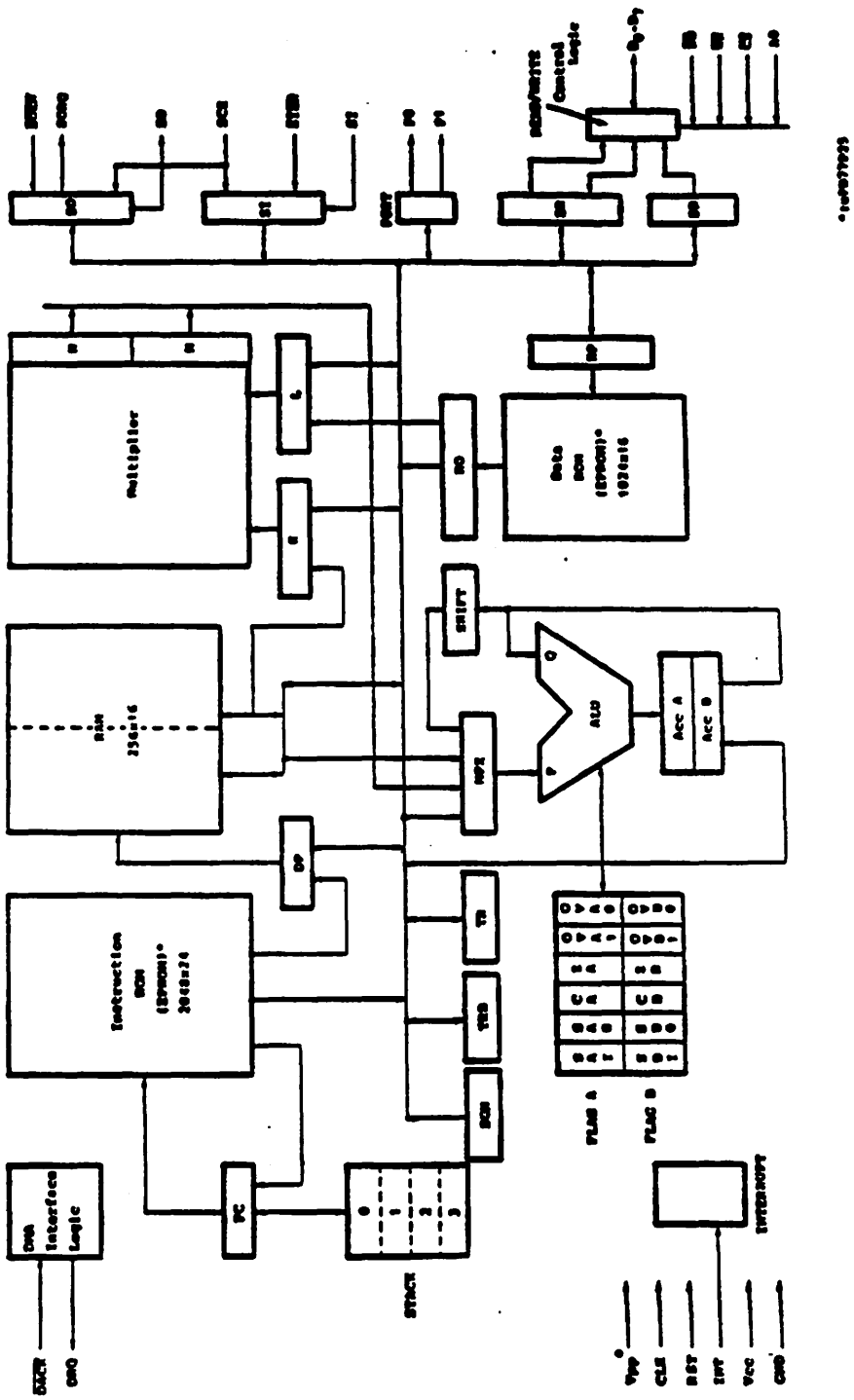


Fig. 1-1 Internal Block Diagram

1.4 Differences between uPD77C25 and uPD7720 Family

The uPD77C25 has enhanced functions of the conventional uPD7720 16-bit signal processor family and thus is compatible with the uPD7720 family at an assembler source program level.

Table 1-1 shows differences between uPD77C25 and uPD7720 family.

Table 1-1 Differences between uPD77C25 and uPD7720 Family

Item		uPD7720	uPD77C25
Memory	Instruction ROM	512 x 23 bits	2,048 x 24 bits
	Data ROM	510 x 13 bits	1,024 x 16 bits
	RAM	128 x 16 bits	256 x 16 bits
Register	PC	9 bits	11 bits
	STACK	9 bits x 4 levels	11 bits x 4 levels
	RP	9 bits	10 bits
	RO	13 bits	16 bits
	DP	7 bits	8 bits
	Added register	-	TRB
Instruction length		23 bits (w/3-bit DP_H.M field)	24 bits (w/4-bit DP_H.M field)
Added instructions		-	JDPLNO JDPLNF M8-MF (Modified DP)
RQM flag operations		Not affected in DMA mode	Affected even in DMA mode
Operation clock (Instruction cycle)		8.192MHz (244ns)	16.384MHz (122ns)

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Layout

The uPD77C25 comes in two package types: 28-pin DIP and 44-pin PLCC. Table 2-1 lists the product names and package types.

Figs. 2-1 and 2-2 show the pin layouts of the 28-pin DIP and 44-pin PLCC, respectively.

Table 2-1

Product name	On-chip ROM structure	Package
uPD77C25C	Mask ROM	28-pin plastic DIP
uPD77C25D		28-pin ceramic DIP
uPD77C25L		44-pin PLCC
uPD77P25D	UVEPROM	28-pin ceramic DIP with a window

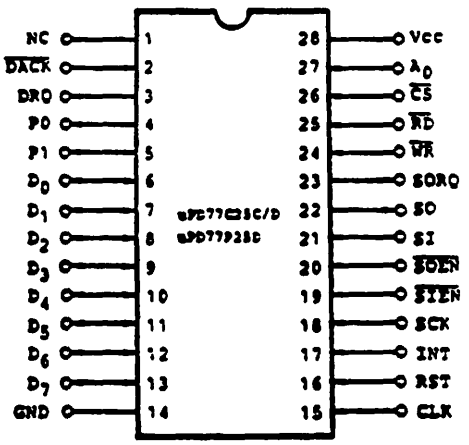


Fig. 2-1 28-Pin DIP Pin Layout

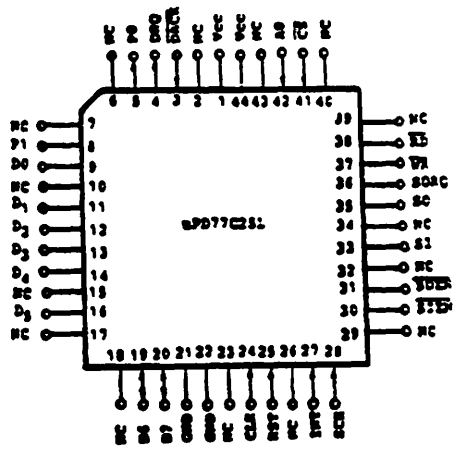


Fig. 2-2 44-Pin PLCC Pin Layout

2.2 Pin Functions

Table 2-2 Pin Function

Pin Designation	DIP pin number	PLCC pin number	I/O	Function
VDD	28	1 44	-	Power input pin. Inputs +5V.
GND	14	21 22	- -	Ground pin
VPP (Note)	1	2	-	Program power input pin of internal UVEPROM. Connected to +12.5V for programming UVEPROM or to +5 V for normal operation.
CLK	15	24	I	Inputs system clock having frequency twice as high as instruction cycle.
RST	16	25	I	Inputs system reset signal (active high). Width of signal must be wider than 4 system clock periods.
INT	17	27	I	Inputs mask able interrupt signal (active high). Program execution jumps to interrupt address at rising edge of this pin and with interrupt enabled.

Note: This pin serves as VPP only for the uPD77P25 and is NC for the uPD77C25.

$\overline{CS}$	26	41	I	Inputs chip select signal (active low). "0" input to this pin enables read/write operation by host CPU via D0 through D7.
A0	27	42	I	Inputs address signal. Signal input to this pin selects register whose contents are to be output from D0 through D7 during read operation. "0" selects DR and "1", SR.
$\overline{RD}$	25	38	I	Inputs read signal (active low). "0" to this pin causes D0 through D7 to output date (with $\overline{CS}$="0").
$\overline{WR}$	24	37	I	Inputs write signal (active low). "0" to this pin causes D0 through D7 to input data (with $\overline{CS}$="0").
D0-D7	Refer to Fig. 2-1.	Refer to Fig. 2-2.	I/O (3-state)	Constitute 8-bit data bus for host CPU and perform input/output according to $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$.
DRQ	3	4	O	Outputs DMA request signal (active high) and requests data transfer in DMA mode.

$\overline{DACK}$	2	3	I	Inputs DMA acknowledge signal (active low). "0" is input when DMA is enabled. When $\overline{DACK}$="0", this pin performs similar operation to when $\overline{CS}$="0" and A0="0". Since it is always valid, input "1" when DMA is not used.
P0, P1	4, 5	5, 8	O	Constitute general-purpose output port.
SI	21	33	I	Inputs serial data which is read into the processor in synchronization with rising edge of SCK clock.
$\overline{SIEN}$	19	30	I	Inputs serial input enable signal (active low) to enable serial data input from SI.
SO	22	35	O (3-state)	Outputs serial data which is output in synchronization with falling edge of SCK clock.
$\overline{SOEN}$	20	31	I	Outputs serial output enable signal (active low) to enable serial data output from SO.
SORQ	23	36	O	Outputs serial output request signal (active high). It is set to "1" when output data are set in SO register and cleared to "0" when on completion of output.

SCK	18	28	I	Inputs serial data clock with which serial data input/output is synchronized.

2.3 Pin Functions for Writing/Reading uPD77P25's Internal UVEPROM

Table 2-4 Pin Functions for Writing/Reading UVEPROM

Pin Name	DIP Pin No.	Pin Name for normal operation	Function
A0	27	A0	Input address (viewed from external device) for writing/reading UVEPROM (instruction ROM and data ROM).
A1	24	$\overline{WR}$	
A2	23	SORQ	
A3	22	SO	
A4	21	SI	
A5	20	$\overline{SOEN}$	
A6	19	$\overline{SIEN}$	
A7	18	SCK	
A8	17	INT	
A9	15	CLK	
A10	5	P1	
A11	4	P0	
A12	3	DRQ	
A13	2	$\overline{DACK}$	
D0 - D7	6-13	D0 - D7	Inputs/outputs data for UVEPROM (instruction ROM and data ROM)
$\overline{CE}$	26	$\overline{CS}$	UVEPROM write strobe signal (active low)
$\overline{OE}$	25	$\overline{RD}$	UVEPROM read strobe signal (active Low)
V_{PP}	1	V_{PP}	Power pin for writing UVEPROM Apply +12.5V for writing and +5V for reading.
V_{DD}	28	V_{DD}	Power pin Apply +6V for writing and +5V for reading.
GND	14	GND	Ground pin

-	16	RST	Sets UVEPROM write or read mode. Mode is set when +12.0V is applied.

CHAPTER 3 HARDWARE ARCHITECTURE

This chapter describes the operations and functions of each uPD77C25 circuit block.

3.1 Instruction ROM Peripheral Circuit

The uPD77C25 has an on-chip, 2 K word x 24 bit instruction ROM. This instruction ROM is a mask ROM in the uPD77C25 and a UVEPROM in the uPD77P25.

Fig. 3-1 shows the peripheral circuit of the instruction ROM.

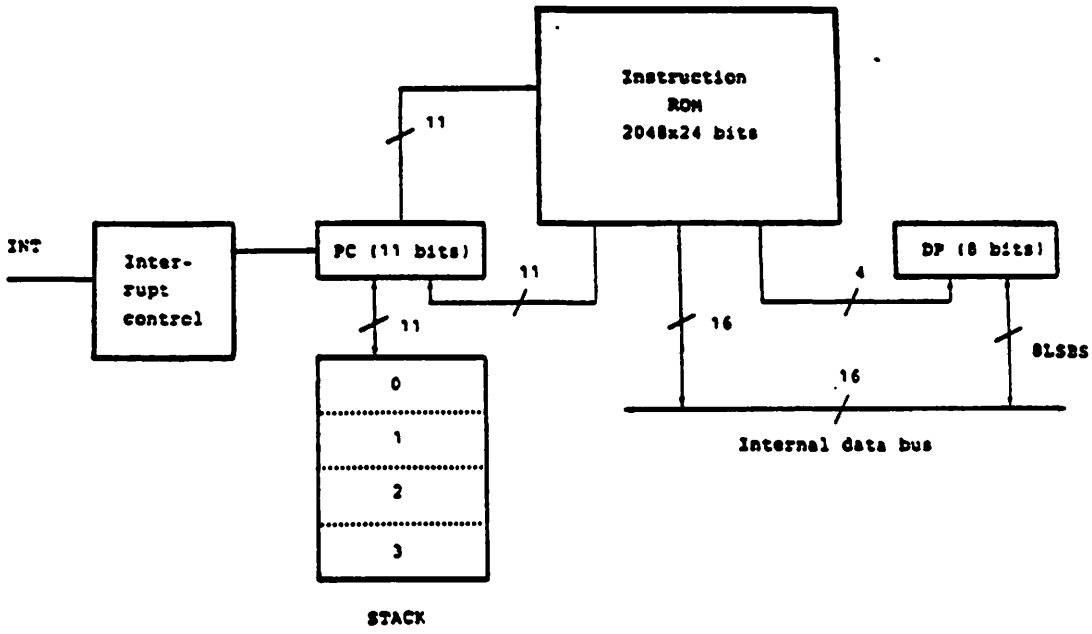


Fig. 3-1 Instruction ROM Peripheral Circuit

3.1.1 Instruction ROM

This ROM stores the program of the uPD77C25 and has a capacity of 2 K words x 24 bits. The address to be accessed is specified by Program Counter (PC).

3.1.2 Program counter (PC)

This is a 11-bit binary counter which specifies an address of the instruction ROM. PC functions as follows:

(1) Usually, the contents of PC are incremented by one each time an instruction is fetched.
(2) When any of the following branch instructions is executed, the NA (Next Address) field value of the instruction is input to the PC.
   - Unconditional jump instruction (JMP instruction)
   - Conditional jump instruction (if the condition is met)
   - Subroutine call instruction (CALL instruction)
(3) When the RT (return) instruction is executed, the return address saved in the stack is input to the PC.
(4) When an interrupt request is input to the INT pin with the interrupt enabled, the interrupt address (100H) is input to the PC.
(5) The reset input clears the PC contents to 000H and the program execution starts from address 0.

3.1.3 Stack

The stack has 11 bit x 4 level, LIFO (Last-In Fast-Out) configuration and stores the return address when the subroutine call instruction is executed or an interrupt is generated. The return address is read out from the stack and input to the PC when the return instruction is executed.
Up to four levels of nesting, including an interrupt, are allowed.
If nesting is attempted at more than four levels, return addresses will be discarded, starting from the one stored first.

3.2 Data ROM Peripheral Circuit

The uPD77C25 is provided with an on-chip the 1 K word x 16 bit data ROM.

The data ROM is a mask ROM in the uPD77C25 and a UVEPROM in the uPD77P25.

Fig. 3-2 shows the peripheral circuit of the data ROM.

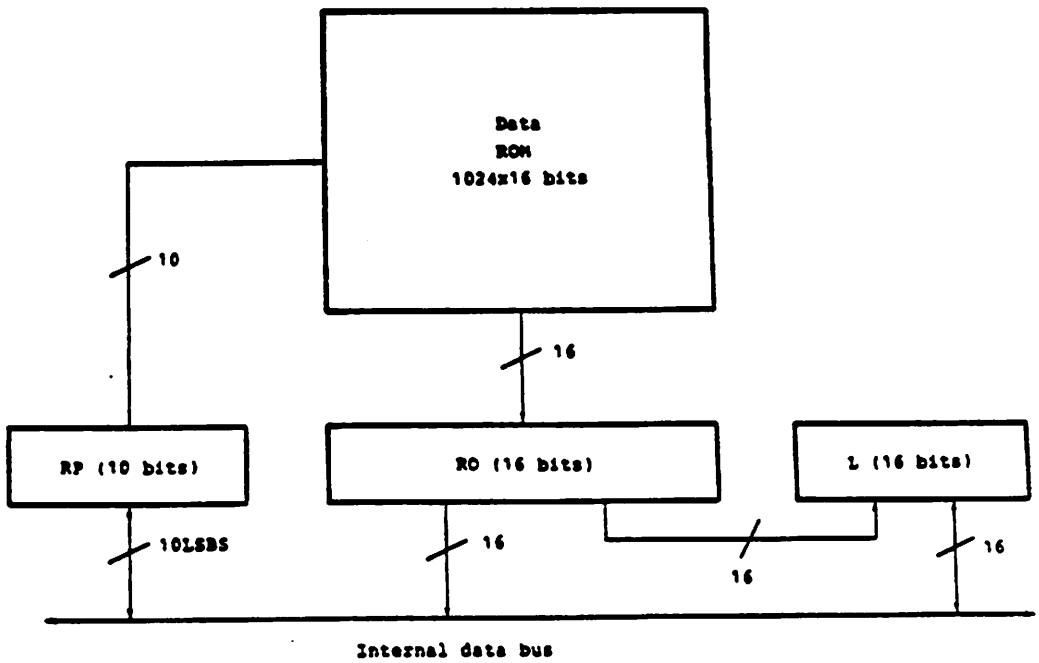


Fig. 3-2 Data ROM Peripheral Circuit

3.2.1 Data ROM

The data ROM stores various fixed data of the uPD77C25 and has a capacity of 1 K words x 16 bits. The address to be accessed is specified by ROM Pointer (RP) and is output to the data bus and the input register L of the multiplier through ROM Output Buffer (RO).

3.2.2 ROM Pointer (RP)

The ROM Pointer is a 10-bit register which specifies a data ROM address to be accessed. It is connected to the lower 10 bits of the internal data bus and has the following functions.

(1) When data is to be input from the internal data bus to RP, only the lower 10 bits of the data bus are valid and the upper 6 bits are ignored.

(2) When data is to be output from RP to the internal data bus, the contents of RP are output to the lower 10 bits of the data bus; "0s" are output to the upper 6 bits.
(3) The contents of RP can be decremented by specifying the RPDCR bit of an instruction. However, the decrementing operation is ignored when data is transferred to RP by the same instruction.
(4) The RP value resulting from the decrementing operation is valid from the next and subsequent instructions.

3.2.3 ROM output buffer (RO)

The ROM output buffer is a 16 bits register which holds the data output by the data ROM. The contents of this register are directly output to the internal data bus and the input register L of the multiplier.

3.3 RAM Peripheral Circuit

The uPD77C25 has an on-chip 256 word x 16 bit RAM.
Fig. 3-3 shows the RAM peripheral circuit.

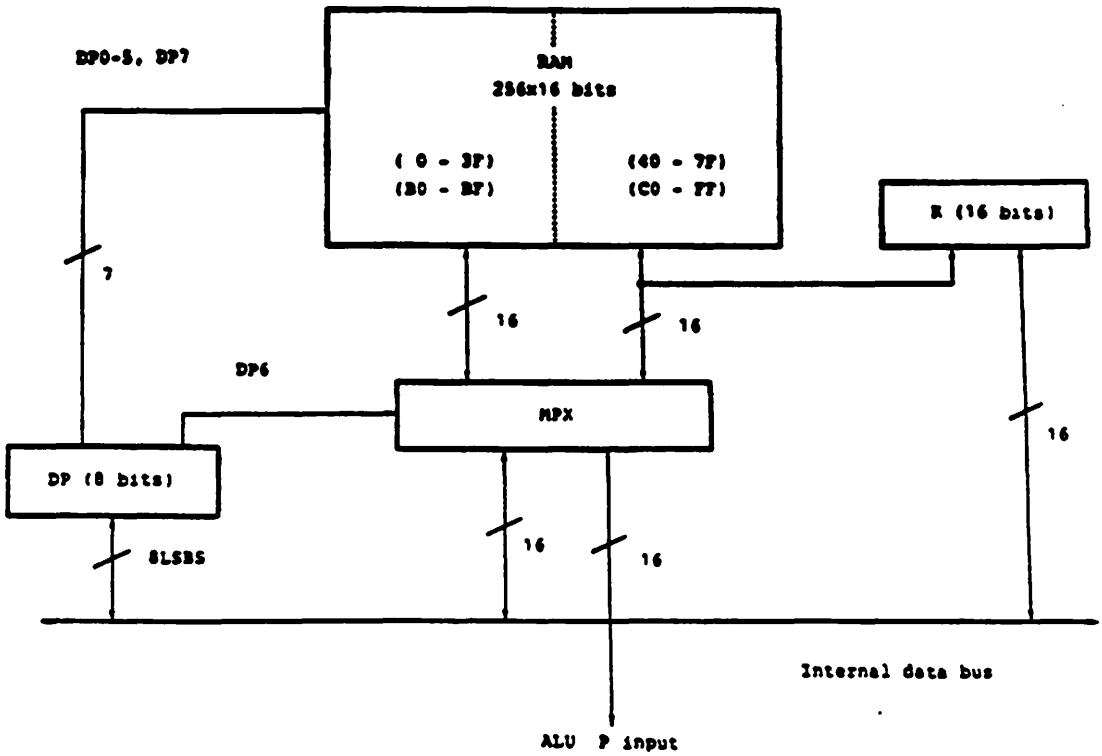


Fig. 3-3 RAM Peripheral Circuit

3.3.1 RAM

The RAM stores the data of the uPD77C25 and is configured of 256 words x 16 bits. The address to be accessed is specified by Data Pointer (DP).
In addition to transferring data to and from the internal data bus, the RAM can directly output data to the P input of ALU. Moreover, it can also directly output to the K register the address data with DP6 replaced with "1".
For example, the contents of address 40H can be output to the K register while the contents of address 0H are being read through the internal data bus when DP=0.

3.3.2 Data pointer (DP)

The data pointer is an 8-bit register which specifies a RAM address and is divided into two sections: the higher 4 bits

(DPH) and the lower 4 bits (DPL). It is connected to the lower 8 bits of the internal data bus and has the following functions:

(1) When data is input from the internal data bus to DP, only the lower 8 bits of the data bus are valid and the upper 8 bits are ignored.

(2) When data is output from DP to the internal data bus, the contents of DP are output to the lower 8 bits of the data bus: "0s" are output to the upper 8 bits.

(3) The upper 4 bits of DP (DPH) can be modified when exclusively ORed with the 4 bits of the DPH.M field in an instruction.

(4) The contents of the lower 4 bits of DP (DPL) can be incremented, decremented, or cleared depending on the specification of the DPL field in an instruction. The carry and borrow generating from the lower 4 bits to the upper 4 bits will be ignored.

(5) Although operations (3) and (4) above can be performed at the same time, they will be ignored if data is to be transferred to DP with the same instruction.

(6) The DP value resulting from the data transfer operation or operations (3) and (4) is valid from the next and subsequent instructions.

3.4 Multiplier

This is a parallel multiplier using secondary Booth's algorithm. It performs multiplication of 16 bits by 16 bits (->31 bit) in one instruction cycle.

Fig. 3-4 shows the peripheral circuit of the multiplier.

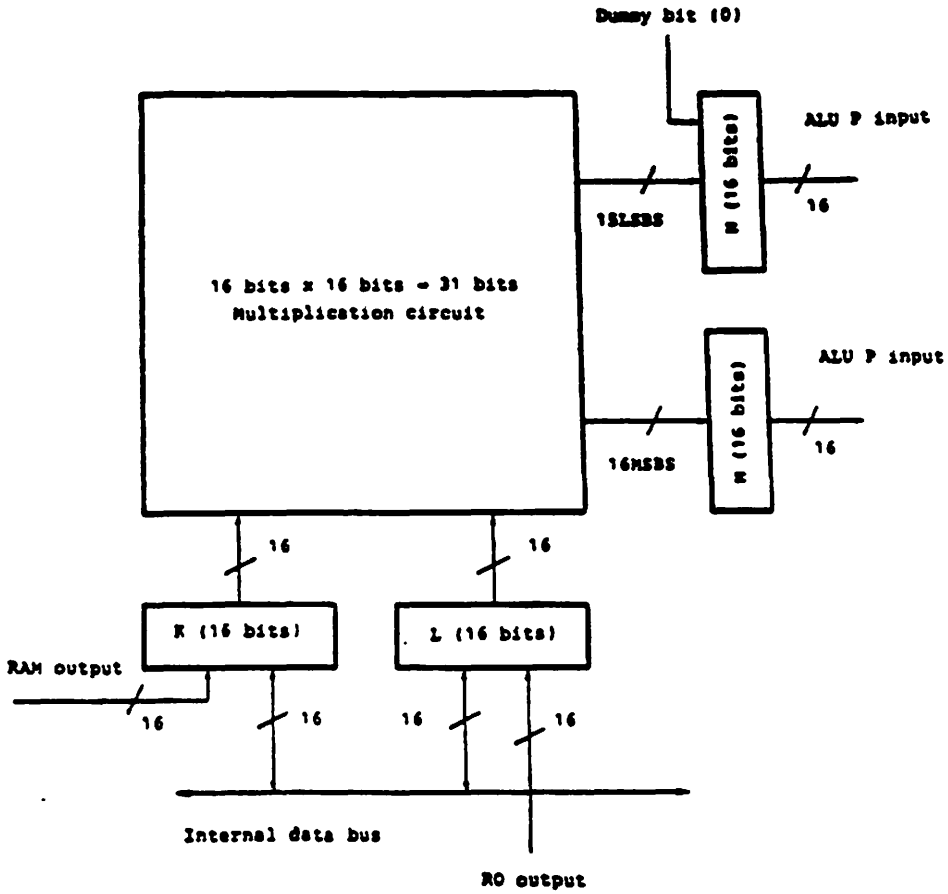


Fig. 3-4 Multiplier Peripheral Circuit

3.4.1 Multiplier

The multiplier multiplies the 2's complement of the 16-bit data stored in the K and L registers in each instruction cycle. As a result, a sign bit and 30-bit data are obtained. The multiplier then stores the sign bit and the upper 15 bits of the 30-bit data in the M register. The lower 15 bits are stored in the upper 15 bits of the N register. ("0" is stored in the LSB.) Since the multiplier overflows if the maximum negative value, 8000H, is

input to both the K and L registers, it outputs 80000000H as an exceptional process.

3.4.2 K and L registers

These are 16-bit registers which hold the data input from the multiplier. The K register is connected to the internal data bus and the output of RAM. It can not only directly input the data from RAM but also transfer data to and from the internal data bus. The L register is connected to the internal data bus and the output of RO. It can directly input the data from the data ROM. In addition, the L register can transfer data to and from the internal data bus.

3.4.3 M and N registers

These are 16-bit registers which hold the results of the multiplication performed by the multiplier. The upper 16 bits of the multiplication results (i.e., sign bit plus the upper 15 bits) are stored in the M register. The lower 15 bits are stored in the N register ("0" is stored in the LSB). Both the M and N registers are connected to the P input of the ALU.

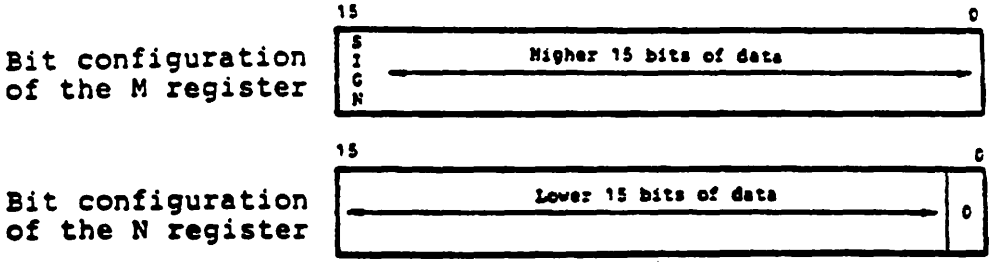


Fig. 3-5 Bit Configuration of M and N Registers

3.5 ALU Peripheral Circuit

The ALU peripheral circuit consists of an ALU that performs numeric operations, except multiplication; logical operations; two accumulators that hold results of an operation; and others. Fig. 3-6 shows the ALU peripheral circuit.

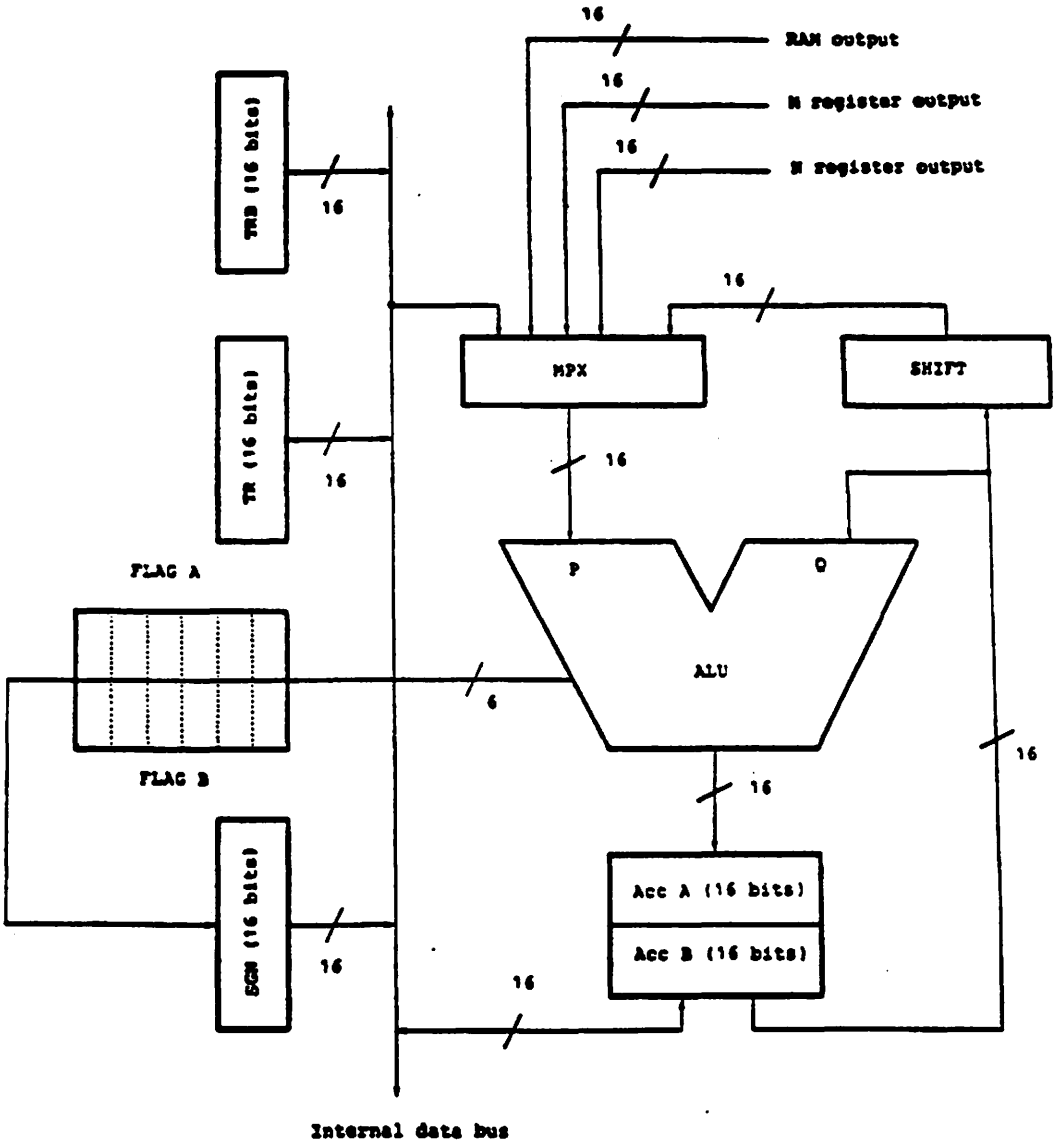


Fig. 3-6 ALU Peripheral Circuit

3.5.1 Arithmetic Logic Unit (ALU)

The ALU performs various arithmetic operations of 16-bit data on two inputs: P and Q.

As the P input, the internal data bus, RAM, M register, or N register can be specified by the P-SELECT field of an instruction. The output from the shift register is automatically input to the P input when the shift operation is performed. However, 0000H is automatically input during the accumulator increment, decrement, and complement operations.

Either AccA or AccB can be specified for the Q input by the ASL field of an instruction. The result of ALU operation is output to the specified Acc, affecting the corresponding flags.

Table 3-1 List of ALU Operations

Mnemonic	ALU field				Operation
	D19	D18	D17	D16	
NOP	0	0	0	0	No operation
OR	0	0	0	1	OR (Acc)←(Acc)∨(P)
AND	0	0	1	0	AND (Acc)←(Acc)∧(P)
XOR	0	0	1	1	Exclusive OR (Acc)←(Acc)∀(P)
SUB	0	1	0	0	Substract (Acc)←(Acc)-(P)
ADD	0	1	0	1	Add (Acc)←(Acc)+(P)
SBB	0	1	1	0	Substract with Borrow (Acc)←(Acc)-(P)-(C)
ADC	0	1	1	1	Add with Carry (Acc)←(Acc)+(P)+(C)
DEC	1	0	0	0	Decrement Acc (Acc)←(Acc)-1
INC	1	0	0	1	Increment Acc (Acc)←(Acc)+1
CMP	1	0	1	0	Complement Acc (Acc)←$\overline{(Acc)}$
SHR1	1	0	1	1	1-bit R-Shift
SHL1	1	1	0	0	1-bit L-Shift
SHL2	1	1	0	1	2-bit L-Shift
SHL4	1	1	1	0	4-bit L-Shift
XCHG	1	1	1	1	8-bit Exchange

3.5.2 Shift register

This is a register which shifts 16-bit data input from AccA and AccB. The shifting is performed in the following modes.

- 1-bit shift right
- 1-bit shift left
- 2-bit shift left
- 4-bit shift left
- 8-bit exchange

3.5.3 Accumulators AccA and AccB

Both AccA and AccB are 16-bit accumulators which store the results of the arithmetic operation performed by the ALU. They are connected to the output of the ALU and the internal data bus. The ASL field of an instruction specifies which accumulator is to be used to store the data output from the ALU. However, if data transfer to an Acc specified by the ASL field of the same instruction as that specifics the Acc, the output of ALU is ignored.

3.5.4 Flag registers FLAG A and B

FLAG A and FLAG B are both 6-bit registers that hold the result of the ALU operation. FLAG A operates when AccA is selected, while FLAB B operates when AccB is selected. Both FLAG A and FLAG B are cleared to "0s" by reset input.

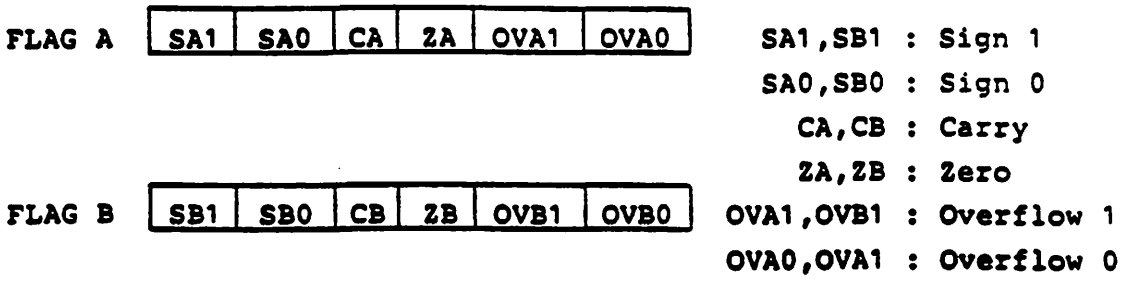


Fig. 3-7 Configuration of FLAG A and FLAG B

Table 3-2 Functions of FLAG

Contents of FLAG	Function
SA1, SB1 (Sign 1)	When an arithmetic operation is performed, if the previous operation resulted in OVA1(OVB1)="0", store the sign bit resulting from the new operation (same as SA0, SB0). If OVA1(OVB1)="1", the previous SA1(SB1) value is retained.
SA0, SB0 (Sign 0)	Store the sign bit (MSB) of the result of the operation.
CA, CB (Carry)	Store the carry or borrow resulting from the executed operation.
ZA, ZB (Zero)	Set to "1" if the result of the executed operation is "0"; otherwise, cleared to "0".
OVA0, OVB0 (Overflow 0)	Set to "1" if an overflow in positive or negative direction has occurred as a result of the executed operation; otherwise, cleared to "0".
OVA1, OVB1 (Overflow 1)	Set to "1" if an overflow has occurred the odd number of times as a result of the 3 operations executed, and set to "0" if it has occurred the even number of times. However, when the overflow has occurred in order of 1, 0, 1 with SA1(SB1)=SA0(SB0), set to "1". If SA1(SB1)≠SA0(SB0), cleared to "0".

Table 3-3 Effect of ALU Operation on Flags

	Selected FLAG						Unselected FLAG					
	S1	S0	C	Z	OV1	OV0	S1	S0	C	Z	OV1	OV0
NOP	●	●	●	●	●	●	●	●	●	●	●	●
OR	×	↕	0	↕	0	0	●	●	●	●	●	●
AND	×	↕	0	↕	0	0	●	●	●	●	●	●
XOR	×	↕	0	↕	0	0	●	●	●	●	●	●
SUB	↕	↕	↕	↕	↕	↕	●	●	●	●	●	●
ADD	↕	↕	↕	↕	↕	↕	●	●	●	●	●	●
SBB	↕	↕	↕	↕	↕	↕	●	●	—	●	●	●
ADC	↕	↕	↕	↕	↕	↕	●	●	—	●	●	●
DEC	↕	↕	↕	↕	↕	↕	●	●	●	●	●	●
INC	↕	↕	↕	↕	↕	↕	●	●	●	●	●	●
CMP	×	↕	0	↕	0	0	●	●	●	●	●	●
SHR1	×	↕	↕	↕	0	0	●	●	●	●	●	●
SHL1	×	↕	↕	↕	0	0	●	●	—	●	●	●
SHL2	×	↕	0	↕	0	0	●	●	●	●	●	●
SHL4	×	↕	0	↕	0	0	●	●	●	●	●	●
XCHG	×	↕	0	↕	0	0	●	●	●	●	●	●

— : Affects the result of operation
↕ : Affected by the result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains the previous state
× : Undefined

3.5.5 Temporary registers TR and TRB

TR and TRB are 16-bit general-purpose registers which can be used to temporarily latch data. They are connected to the internal data bus.

3.5.6 Sign register SGN

SGN is a register used to compensate the overflow. It is set to 8000H if the SA1 flag is "0" and to 7FFFH if the flag is "1". Therefore, the overflow compensation can be performed with one instruction by simply transferring the contents of the SGN register, without a test instruction.

3.5.7 Overflow processing

The μPD77C25 is provided with the OVA1 (OVB1), SA1 (SB1), and SGN register to effectively process overflow that may occur during an arithmetic operation.

When arithmetic operations are continuously performed on 2's complement data and if an overflow has occurred the even number of times, the eventual result may sometimes be correct.

The OVA1 (OVB1) flag indicates whether the results of the three operations performed immediately before are correct. The SA1 (SB1) flag indicates the direction in which the overflow (positive or negative) has, if any, occurred. The SGN register generates compensation data (the positive maximum value or negative maximum value) when an overflow occurs.

Example 1: When overflow occurs odd number of times

```
        x=7FFFH (+32767)
        y=0002H     (+2)
        z=00FEH   (+254)
        w=0A00H  (+2560)
                          Assuming the previous OVA1=0
    x+y
        7FFFH (+32767)
        0002H     (+2)
        8001H (-32767)  -> OVA0=1, SA0=1, SA1=1, OVA1=1
    x+y+z
        8001H (-32767)
        00FEH   (+254)
        80FFH (-32513)  -> OVA0=0, SA0=1, SA1=1, OVA1=0
    x+y+z+w
        80FFH (-32513)
        0A00H  (+2560)
        8AFFH (-29953)  -> OVA0=0, SA0=1, SA1=1
```

Since OVA0 has been set the odd number of times, it causes OVA1 to become 1 and thus, data 7FFFH that compensates for the normal value (+35583) is set to SGN.

Example 2: When overflow occurs even number of times

```
        x=7010H (+28688)
        y=1FFFH  (+8191)
        z=8001H (-32767)
        w=0F00H  (+3840)
                          Assuming the previous OVA1=0
    x+y
        7010H (+28688)
        1FFFH  (+8191)
        900FH (-28657)  -> OVA0=1, SA0=1, SA1=1, OVA1=1
    x+y+z
        900FH (-28657)
        8001H (-32767)
        1010H  (+4112)  -> OVA0=1, SA0=0, SA1=1, OVA1=0
    x+y+z+w
        1010H  (+4112)
        0F00H  (+3840)
        1F10H  (+7952)  -> OVA0=0, SA0=0, SA1=0
```

Since OVA0 has been set the even number of times, it causes OVA1 to become 0 and thus, the normal value (+7952) results.

Example 3: When overflows occur in order of 1, 0, and 1.
(a) Example of SA0 (SB0)=SA1 (SB1)

```
        x=7FFFH (+32767)
        y=0FFEH  (+4094)
        z=7F00H (+32512)
        w=7F04H (+32516)
                          Assuming the previous OA1=0
    x+y
        7FFFH (+32767)
        0FFEH  (+4094)
        8FFDH (-28675)  -> OVA0=1, SA0=1, SA1=1, OVA1=1
```

```
x+y+z
    8FFDH (-28675)
    7F00H (+32512)
    0EFDH  (+3837)  -> OVA0=0. SA0=0. SA1=1. OVA1=1
x+y+z+w
    0EFDH  (+3837)
    7F04H (+32516)
    8E01H (-29183)  -> OVA0=1. SA0=1. SA1=1
```

Since OVA0 has been set in the order of 1. 0. and 1. and SA0 is equal to SA1. OVA1 becomes 1. Thus. the data 7FFFH (+32767) that compensates for the normal value is set to SGN.

(b) Example of SA0 (SB0)=SA1 (SB1)

```
  x=7FFFH (+32767)
  y=0002H     (+2)
  z=00FEH   (+254)
  w=8001H (-32767)
                    Assuming the previous OVA1=0
x+y
    7FFFH (+32767)
    0002H     (+2)
    8001H (-32767)  -> OVA0=1. SA0=1. SA1=1. OVA1=1
x+y+z
    8001H (-32767)
    00FEH   (+254)
    80FFH (-32513)  -> OVA0=0. SA0=1. SA1=1. OVA1=1
x+y+z+w
    80FFH (-32513)
    8001H (-32767)
    0100H   (+256)  -> OVA0=1. SA0=0. SA1=1
```

Since OVA0 has been set in the order of 1. 0. and 1. and SA0 is not equal to SA1. OVA1 becomes 1. Thus. the normal value (+256) results.

3.6 System Control

3.6.1 Hardware reset

A pulse width of wider than four periods of system clock is required for the RST signal.

(1) Internal operations by hardware reset

- PC, SR register, FLAG A, and FLAG B are cleared to "0".
- SI ACK flag and SO ACK flag are cleared to "0".

(2) States of pins upon hardware reset

Table 3-4 Pin State during Hardware Reset

Pin name	State
DRQ	Low level
P0, P1	Low level
D0 to D7	Depend on $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, and $\overline{DACK}$.
SO	High impedance
SORQ	Low level

3.6.2 Interrupt control

The uPD77C25 has interrupt input signal INT that can be disabled by software. The interrupt functions are as follows:

(1) An interrupt is detected at the rising edge of the INT signal when the EI bit of the SR register is set to "1".

(2) The EI bit is set by transferring data to the SR register with a specific bit set to "1". The high level width of the INT signal must be wider than eight periods of system clocks.

(3) The EI bit is automatically cleared to "0" when INT is detected.

(4) To detect the next INT, the EI bit must be set to "1" and the INT signal must be cleared to "0". The low level width of the INT signal must be wider than eight periods of system clocks.

(5) The interrupt address is 100H.

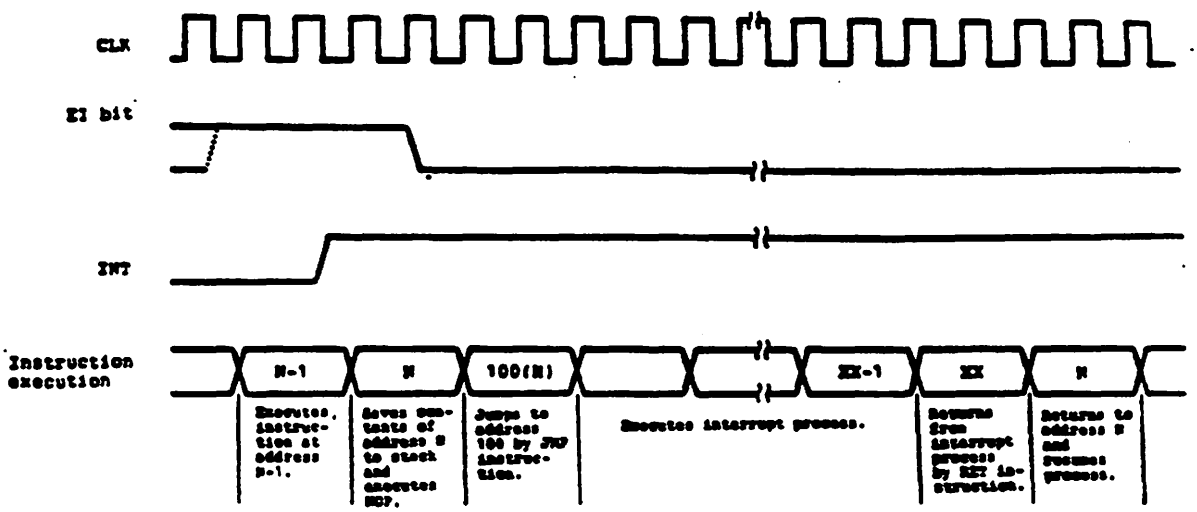
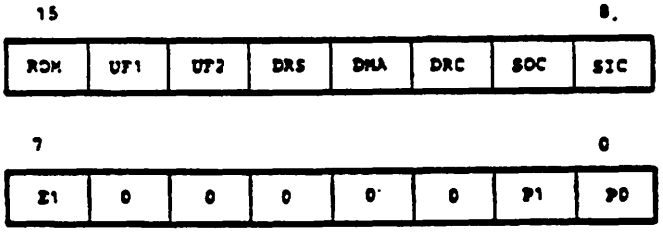


Fig. 3-8 Interrupt Timing

3.6.3 Status register

The status register is a register which hold the status the uPD77C25 needs to transfer data to/from external devices. It consists of 11 bits of statuses and is internally handled as a 16-bit register. The upper 8 bits can be read from an external device through D0 to D7 pins.



Can be read out from D0 to D7 pins.

Fig. 3-9 Configuration of Status Register

Table 3-5 Functions of Status Register

Bit name	Read/Write	Function
P0, P1	Read/Write	Correspond to output ports P0 and P1 and to which values input to these bits are directly output.
EI	Read/Write	Enables/disables interrupt. Interrupt is enabled when this bit is set to "1" and disabled if it is cleared to "0". When interrupt is recognized, this bit is automatically cleared to "0", disabling subsequent interrupts.
SIC	Read/Write	Specifies length of serial input data. When this bit is "0", 16-bit length is specified; otherwise, 8-bit length is specified.
SOC	Read/Write	Specifies length of serial output data. When this bit is "0", 16-bit length is specified; otherwise, 8-bit length is specified.
DRC	Read/Write	Specifies length of data transfer to and from host CPU. When this bit is "0", 16-bit length is specified; otherwise, 8-bit length is specified.
DMA	Read/Write	Specifies mode to transfer data to and from host CPU. When this bit is "0", non-DMA mode is specified; otherwise, DMA mode is specified.
DRS	Read	Indicates data transfer status of DR register. When this bit is "1", data transfer is in progress; when it is "0", transfer is terminated. When DRC bit is "1", DRS bit is always "0".
UF0,UF1	Read/Write	Flag bits which can be freely used by user.

RQM	Read	Indicates that uPD77C25 is requesting host CPU for data write/read. This bit is set to "1" when DR register is internally read/written and cleared to "0" when data read/write is externally performed. Instructions which can read DR register while RQM is "1" are also provided.

3.7 Host CPU Interface

The uPD77C25 can transfer data to and from the host CPU via pins D0 through D7. The data is input to and output from the host CPU through the internal DR register (16 bits). Since the DR register is connected to the internal data bus, it can also transfer the data to/from other internal function blocks sections. Fig. 3-10 shows the block diagram of the host CPU interface.

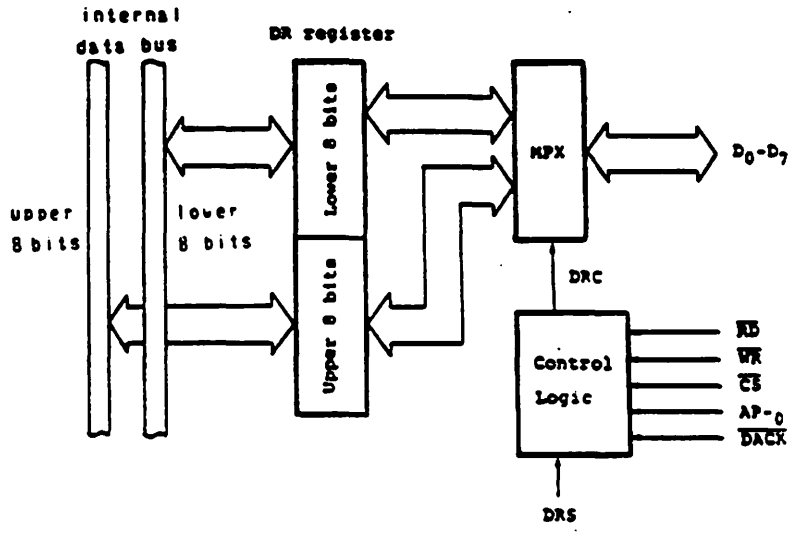


Fig. 3-10 Host CPU Interface

3.7.1 Read/Write control

The following operations are performed depending on the status of $\overline{CS}$, A0, $\overline{RD}$, $\overline{WR}$, and $\overline{DACK}$.

Table 3-6 Read/Write Operation

$\overline{CS}$	A0	$\overline{DACK}$	$\overline{WR}$	$\overline{RD}$	Function
1	X	1	X	X	Internal operation is not affected.
X	X	1	1	1	
0	0	1	0	1	Data on D0 to D7 are latched to DR register.
X	X	0	0	1	
0	0	1	1	0	Data of DR register are output to D0 to D7.
X	X	0	1	0	
0	1	1	1	0	Upper 8 bits of SR register are output to D0 to D7.
0	1	1	0	1	Inhibited.
0	X	1	0	0	
X	X	0	0	0	
X	X	0	1	1	

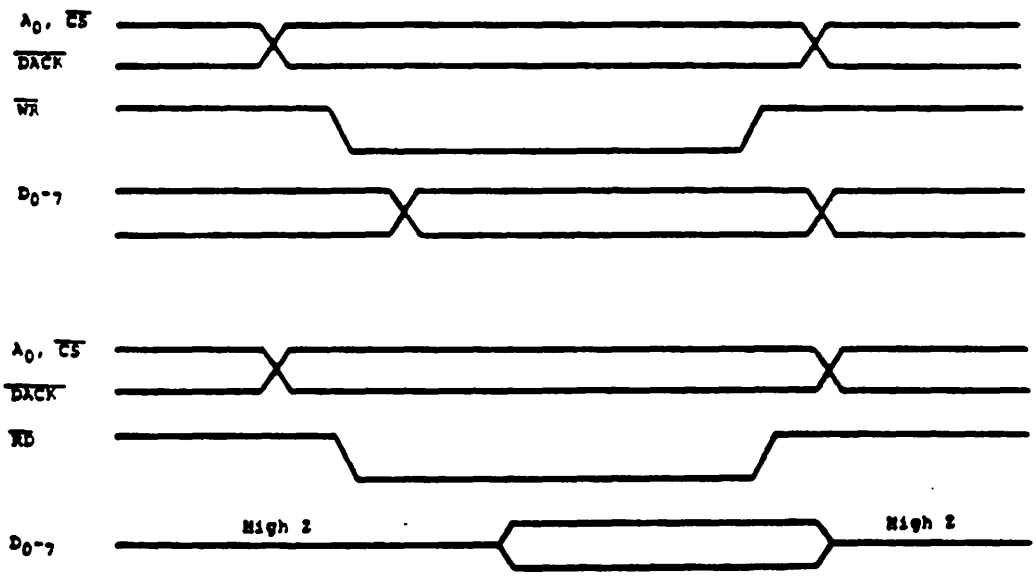


Fig. 3-11 Read/Write Timing

3.7.2 Bit length for data transfer

The bit length for data transfer can be set to be 8 bits or 16 bits by the DRC bit of the status register. When the DRC bit is "0", 16-bit length is specified for data transfer; when it is "1", the 8-bit data length is specified.

When the 8-bit mode is specified, only the lower 8 bits of the DR register (16 bits long) are valid. The upper 8 bits are undefined. When the 16-bit mode is specified, the data written by the host CPU are stored in the DR register in the sequence of the lower 8 bits and then the upper 8 bits. The host CPU reads the data in the sequence of the lower 8 bits and the upper 8 bits. The DRS bit of the status register is set to "1" while the lower 8 bits of the DR register being accessed and cleared to "0" while both the upper and lower 8 bits are being accessed. In the 8-bit mode, the DRS bit is always cleared to "0".

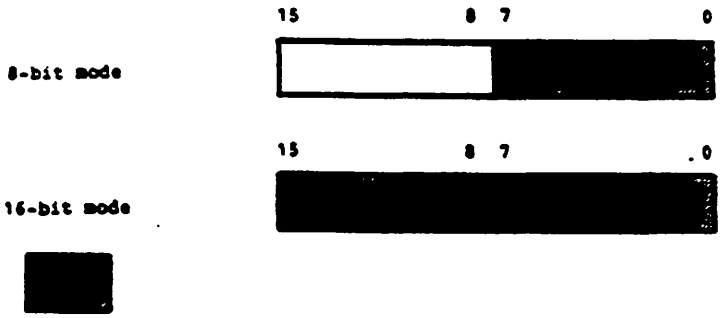


Fig. 3-12 Data Transfer Format

The shaded areas indicate the locations of input/output data and the numbers in these areas indicate the input/output sequence.

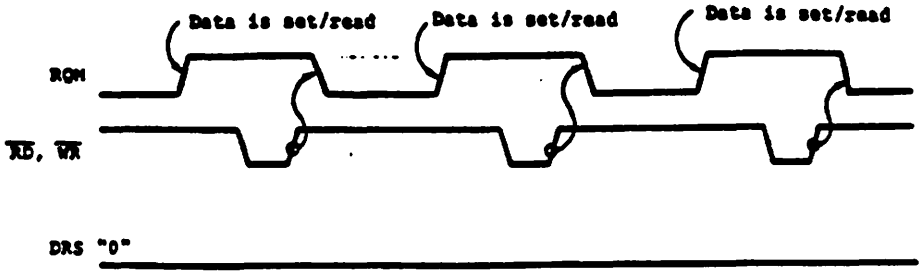


Fig. 3-13 Host CPU Read/Write Timing (8-bit mode)

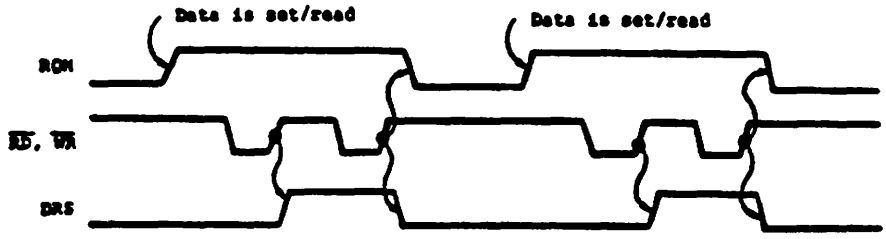


Fig. 3-14 Host CPU Read/Write Timing (16-bit mode)

3.7.3 DMA transfer mode

When the DMA bit of the status register is set to "1", the uPD77C25 transfers data to/from the host CPU in the DMA mode. In the DMA mode, data are transferred through two pins: DRQ and $\overline{\text{DACK}}$. The DRQ pin outputs a DMA request signal. This signal is set to "1" when data is internally read from or written to the DR register; it is cleared to "0" at the falling edge of $\overline{\text{DACK}}$. In the 16-bit mode, however, the DMA request signal remains "1", until the falling edge of $\overline{\text{DACK}}$, to transfer the upper 8 bits. Instructions which can read data out from the DR register while DRQ is "1" are also provided. The $\overline{\text{DACK}}$ pin inputs "0" when the DMA transfer is enabled. At this time, the DACK pin functions in the same manner as when $\overline{\text{CS}}$ is "0" and A0 is "0".

The RQM bit in the status register functions as usual even in the DMA mode.

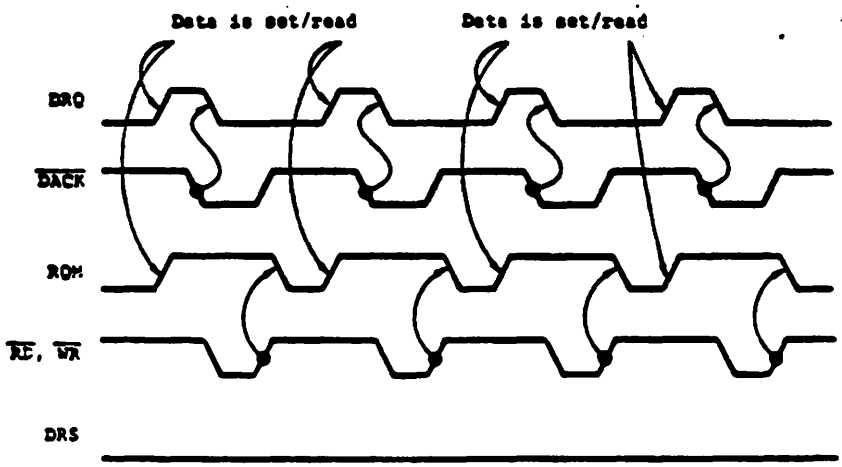


Fig. 3-15 DMA Mode Read/Write Timing (8-bit Mode)

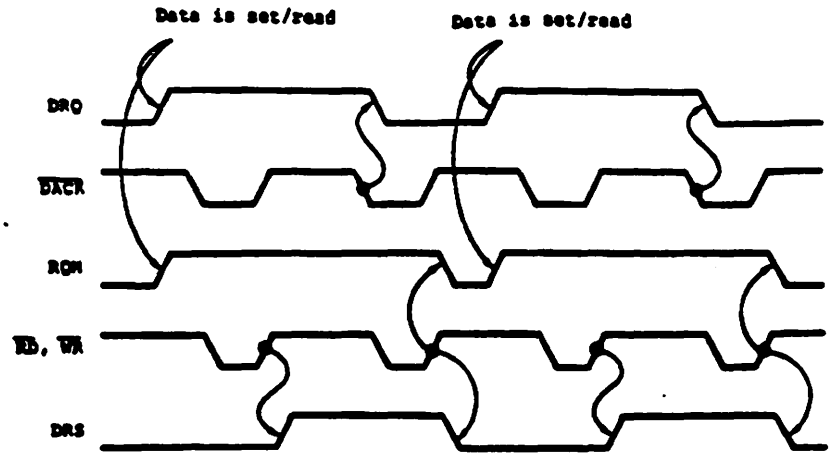


Fig. 3-16 DMA Mode Read/Write Timing (16-bit Mode)

3.8 Serial Interface

The uPD77C25 is provided with a serial interface to transfer signal data to and from external devices. The serial data from an external device is input to the SI pin and transferred to the internal data bus through the SI register. The serial data to an external device is output to the SO pin. from the internal data bus through the SO register. Fig. 3-17 shows the block diagram of the serial interface.

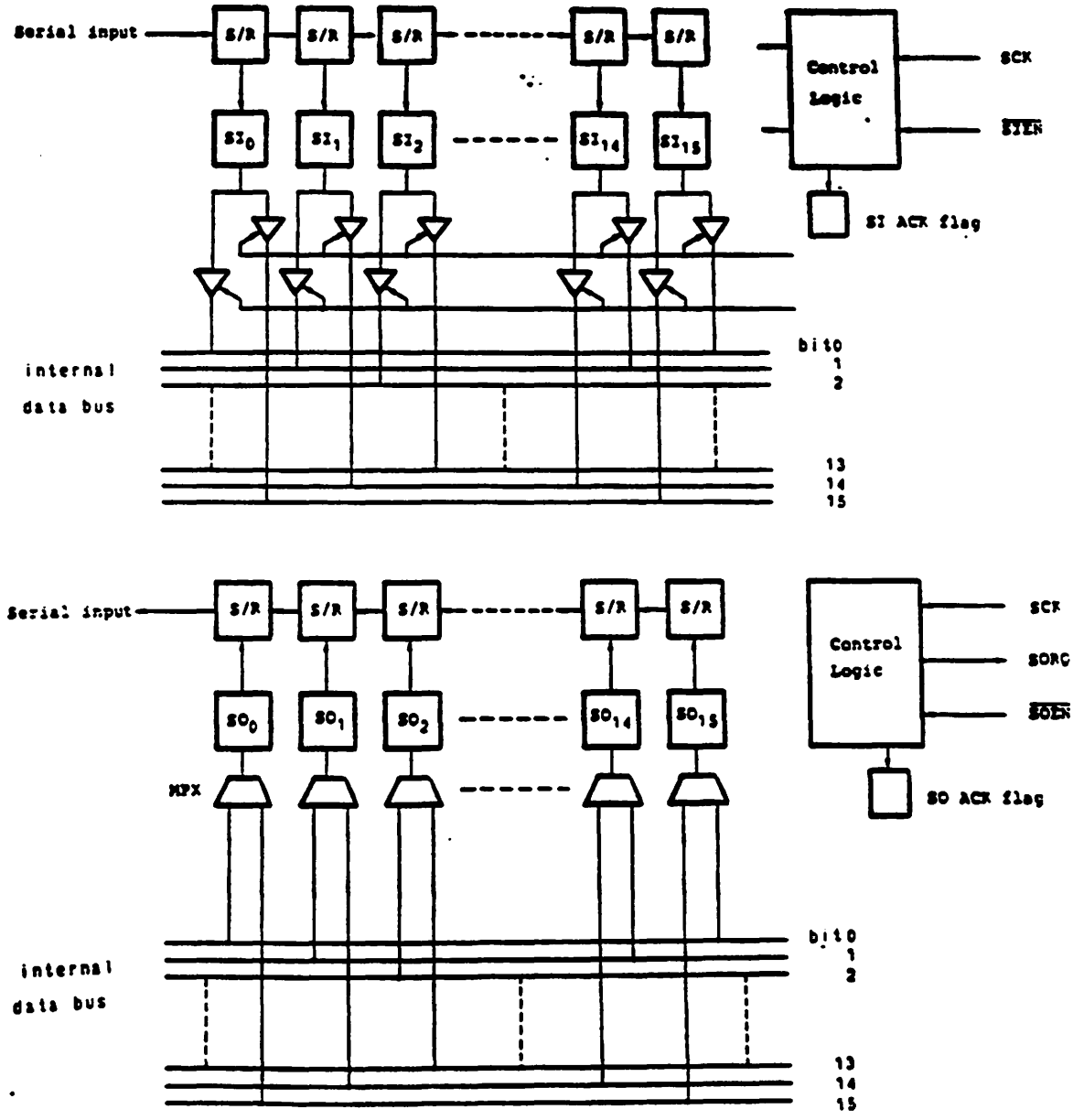


Fig. 3-17 Block Diagram of Serial Interface

3.8.1 Serial data length

The length of the serial data can be set by either the SIC or SOC bit of the status register. When the bit is "0", 16-bit data length is specified; when it is "1", the data length is specified to be 8 bits. The SIC bit sets the length of the serial input and the SOC bit sets the serial output data length.

3.8.2 Serial input timing

(1) Serial data input (SI pin) is enabled by setting $\overline{\text{SIEN}}$ to "0".

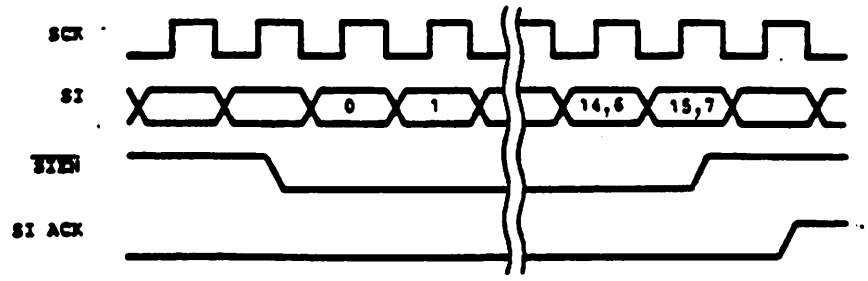
(2) The serial input data are sequentially taken into the shift register at the rising edge of serial clock SCK only when $\overline{\text{SIEN}}$ is "0".

(3) When the data have been taken by the length specified by the SIC bit of the status register, the data are transferred from the shift register to SI, setting the SI ACK flag.
If the 8-bit data length is specified, "0s" are transferred to the remaining 8 bits of the SI register.

(4) If $\overline{\text{SIEN}}$ is set to "1" before the shift register is filled with the specified number of bits, the current status of the shift register is retained until $\overline{\text{SIEN}}$ is cleared to "0". The data input is resumed when $\overline{\text{SIEN}}$ has become "0".

(5) If $\overline{\text{SIEN}}$ remains "0", the next data can be consecutively input to the shift register.

(6) If the subsequent data input is terminated before transferring the preceding data from the SI register to other register, the preceding data is lost.



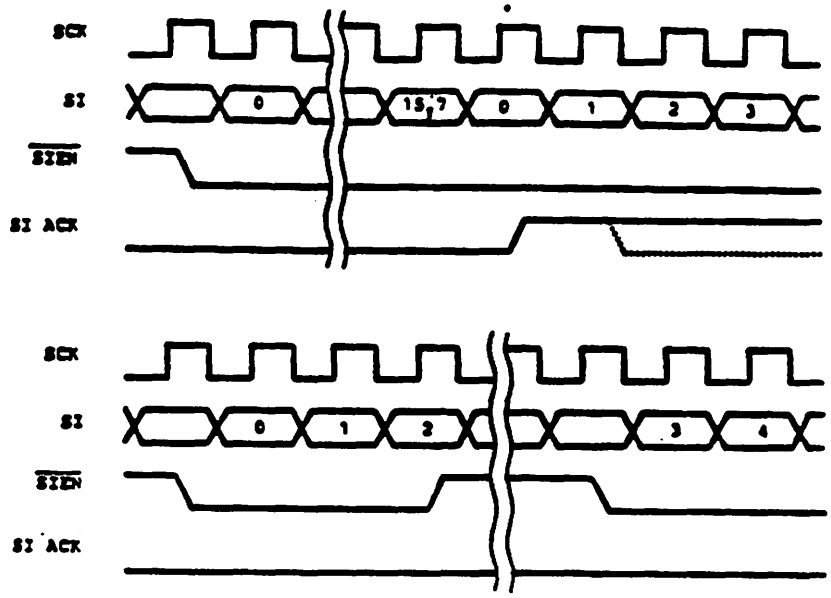


Fig. 3-18 Serial Input Timing

3.8.3 Internal transfer of serial input data

(1) Upon completion of data input to the SI register, the SI ACK flag is set to "1", indicating that valid data have been stored in the SI register. The status of the SI ACK flag can be tested by the JNSIAK or JSIAK instruction.
The JNSIAK instruction causes program execution to jump to a specified address when the SI ACK flag is cleared to "0", while the JSIAK instruction does the same when the flag is set to "1".

(2) The SI register is a 16-bit register whose contents can be read to the internal data bus by an instruction. When they are read out, the SI ACK flag is cleared to "0".

(3) To read out the data of the SI register to the internal data bus, two instruction are available. One transfers the bit first sent to the SI pin to a high bit of the data bus. The other transfers it to a low bit.

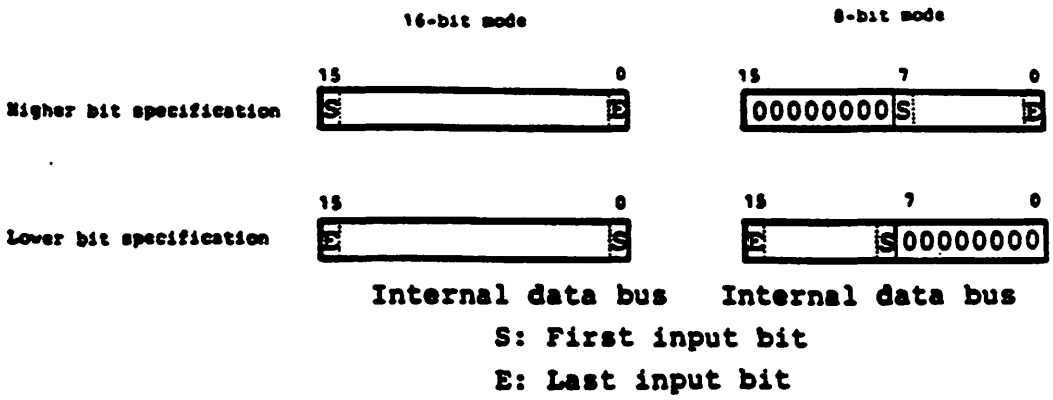


Fig. 3-19 Reading Out SI Register Contents to Internal Data Bus

3.8.4 Serial output timing

(1) When data is written into the SO register by an instruction, the SO ACK flag is set to "1".

(2) The data written into the SI register is transferred to the shift register data input is enabled (if other data is not being sent) and the SO ACK flag is cleared to "0". Then the SORQ pin is set to "1" in synchronization with the falling edge of SCK. If data input to the shift register is disabled, the uPD77C25 waits until it is enabled.

(3) When data is set to the shift register (that is, SORQ=1), the serial output is enabled by clearing $\overline{\text{SOEN}}$ to "0", causing the SO pin to become active and the data to be output in synchronization with the falling edge of SCK.

(4) When the data has been output by the length specified by the SOC bit of the status register (8/16 bits), and if data to be transferred is input to the SO register, the data is transferred to the shift register; otherwise, SORQ is cleared to "0" and the SO pin enter the high-impedance status.

(5) If $\overline{\text{SOEN}}$ is set to "1" before the shift register is filled with the specified number of bits, the SO pin enters the high-impedance status. The data is retained until $\overline{\text{SOEN}}$ is

cleared to "0" again. After $\overline{\text{SOEN}}$ has been cleared to "0", the SO pin is made active and the data output is resumed.

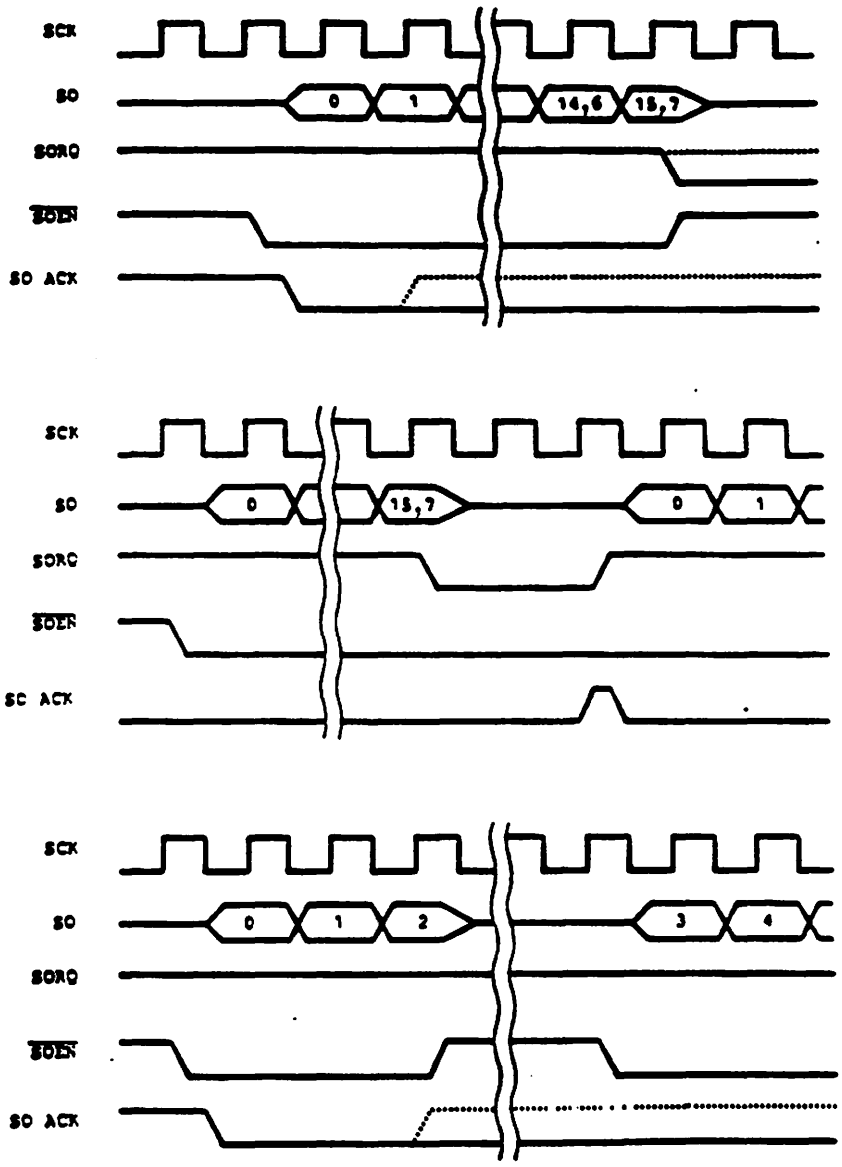


Fig. 3-20 Serial Output Timing

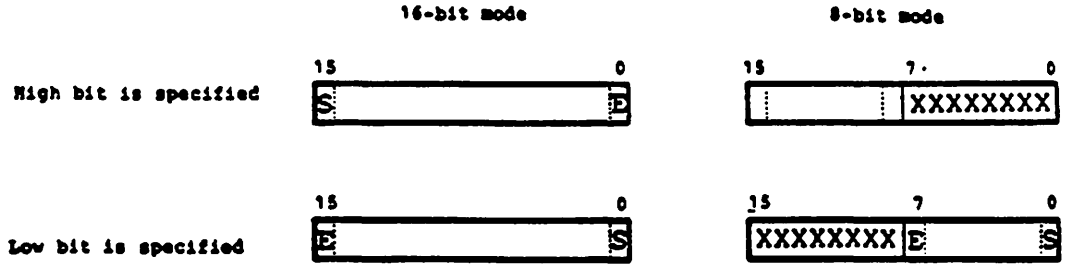
3.8.5 Internal transfer of serial output data

(1) When valid data is not set in the SO register, the SO ACK flag is cleared to "0", indicating that data can be written to the SO register. The status of the SI ACK flag can be tested by the JNSOAK or JSOAK instruction. The JNSOAK instruction causes program execution to jump to the specified address when the SO ACK flag is cleared to "0", while the JSOAK instruction does the same but when the flag is set to "1".

(2) The SO register is a 16-bit register to which data can be written from the internal data bus by an instruction. When the data is written, the SO ACK flag is set to "1".

(3) When the data written into the SO register are transferred to the shift register, the SO ACK flag is cleared to "0", thereby enabling the next data to be written.

(4) Two instructions are provided for writing data into the SO register from the internal data bus. One sends the high bit of the data bus to the SO pin first. The other sends the low bit of the data bus to the SO pin first.



Internal data bus Internal data bus

S: First output bit

E: Last output bit

X: Don't Care

Fig. 3-21 Data Write into SO Register from Internal Data Bus

3.9 uPD77P25 UVEPROM Interface

Both the instruction ROM and data ROM of the uPD77P25 are an ultraviolet-light erasable UVEPROM. This section describes how to write reading data to/from the UVEPROM.

3.9.1 Input/output data format

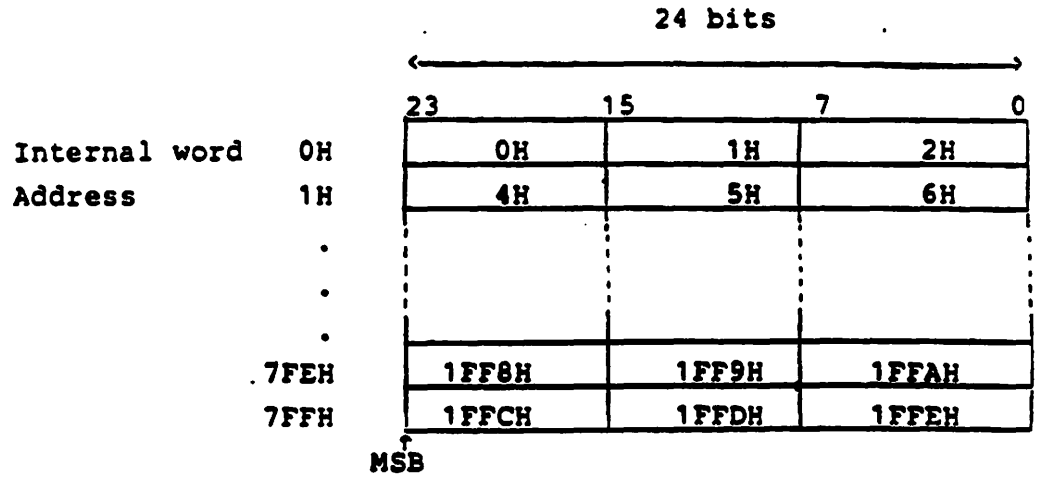
One word of the instruction ROM consists of 24 bits, while 16 bits make up one data ROM word.

Data are written to or read from these UVEPROMs in units of bytes or 8 bits. Therefore, special addresses are assigned to the UVEPROMs.

Fig. 3-22 shows the address assignment. Addresses 0H through 1FFFH are assigned to the 2 K-word instruction ROM. The following addresses, 2000H through 27FFH are assigned to the 1 K-word data ROM. Since the instruction ROM is configured on a 1-word-for-24-bit basis, one dummy byte address is provided per word.

For example, data in word address 0H of the instruction ROM is equivalent to three bytes of byte addresses 0H to 2H. 3H is a dummy address.

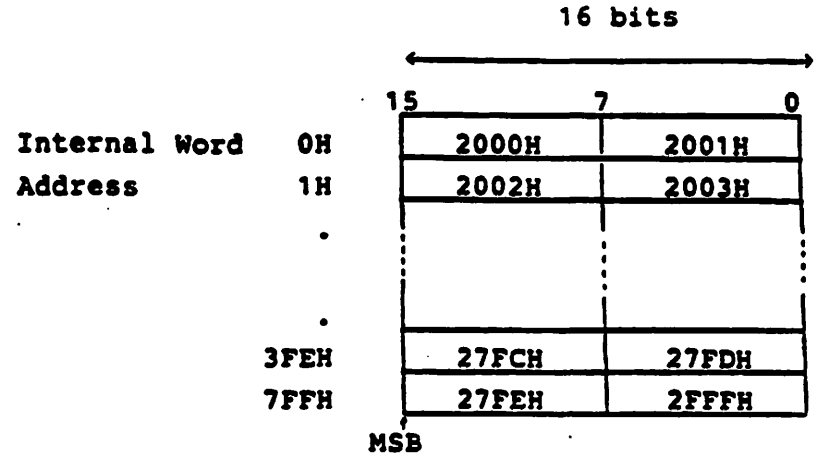
(a) Instruction ROM (1 word=24 bits)



Note: Numeric values within the boxes are byte addresses of the instruction ROM viewed from an external device.

Fig. 3-22 Memory Map of uPD77P25 On-Chip UVEPROM

(b) Data ROM (1 word=16 bits)



Note: Numeric values within the boxes are byte addresses of the data ROM viewed from an external device.

Fig. 3-22 Memory Map of the uPD77P25 On-Chip UVEPROM (Continued)

3.9.2 Erasing data

The data in the uPD77P25's UVEPROMs can be erased them by exposing them to a light with a wavelength shorter than 400nm. All data in the UVEPROMs are set to "1s" after the erasure. Note that, if the uPD77P25 is exposed to the direct sunlight or fluorescent light for a long time, the data might be erased. To prevent this, the UVEPROM window must be masked with a cover or film for shielding from the ultraviolet light.

Usually, the UVEPROMs are erased exposed to the ultraviolet light with a wavelength of 254nm. The total light quantity required to completely erase the written data is 15WS/cm^2 (UV intensity x erase time) that is equivalent to exposure to a UV lamp with a wavelength of 12000uW/cm^2 for about 15 to 20 minutes. However, a longer erasing time may be required due to such factors as the life of the UV lamp and stains on the window of the package.

The uPD77P25 must be positioned within one inch away from the UV lamp.

3.9.3 Procedure to write data

To write data, the μPD77P25 UVEPROMs must be first erased as described in 3.9.2. Then perform the writing operation observing the following procedure.

(1) Apply +12.0V to RST (pin 16), +6V to VDD, and +12.5V to VPP. This causes the UVEPROMs to enter write mode.

(2) Specify the desired ROM byte address from address input pins A0 to A13.

(3) Write the data on the data bus (D0 to D7) by applying "0" to $\overline{CE}$ while $\overline{OE}$ is "1". (program mode).

(4) Output the written data to the data bus (D0 to D7) by applying "0" to $\overline{OE}$ while $\overline{CE}$ is "1" (program verify mode).

(5) Repeat steps (2) through (4) 25 times maximum until the data is properly written to the specified address.

(6) After verifying that the data has been properly written, apply additional pulses by setting $\overline{OE}$ to "1" (clear $\overline{CE}$ to "0". The pulse width of it depends on the number of repetitions in (3) and (4)).

The above procedure completes writing one byte of data.

In case the data will not be properly written even after steps (2) to (4) have been repeated more than 25 times, it means that the μPD77C25 is defective.

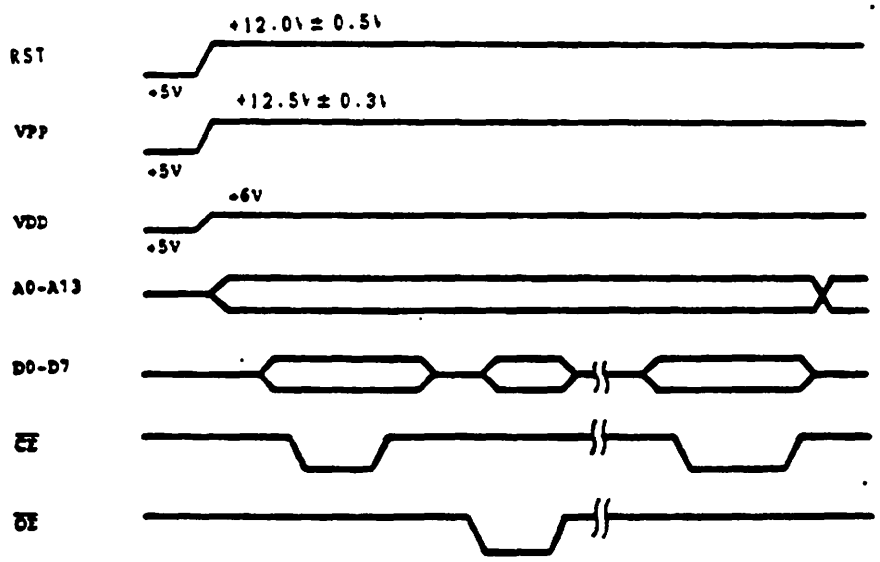


Fig. 3-23 On-Chip UVEPROM Write Timing

3.9.4 Procedure to read data

(1) Apply +12.0V to RST (pin 16), +5V to VDD, and +5V to VPP. This causes the UVEPROMs to enter read mode.

(2) Specify the desired ROM byte address from the address input pins A0 to A13.

(3) Data will be output to the data bus (D0 to D7) by cleaning $\overline{OE}$ and $\overline{CE}$ to "0".

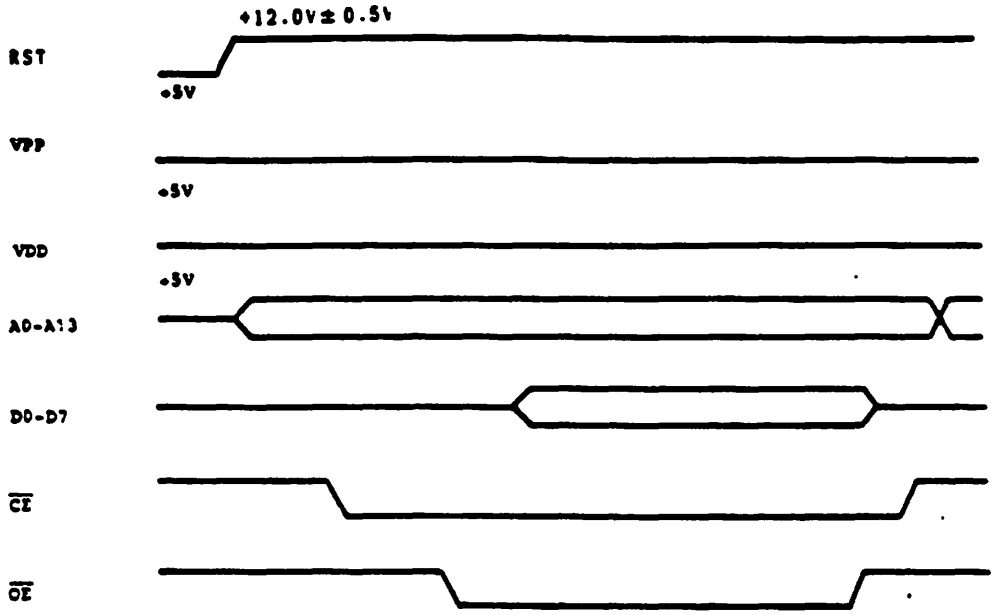


Fig. 3-24 On-Chip UVEPROM Read Timing

CHAPTER 4 INSTRUCTIONS

The uPD77C25 operates according to the external square wave applied to the CLK pin. This square wave is internally divided two to generate two phases of clocks as shown in Fig. 4-1 for internal operation.

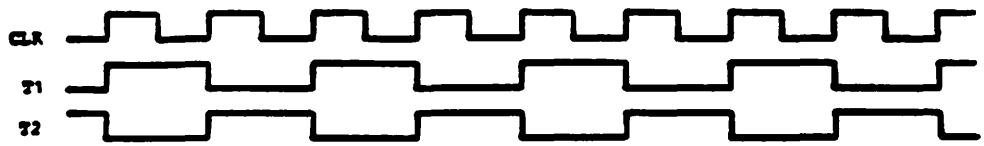


Fig. 4-1 Internal Clock Timing

4.1 Instruction Execution Timing

The instruction is executed as follows. Fig. 4-2 shows the timing chart.

(1) An instruction is executed within two T2's.

(2) Data is read from and written to RAM and registers and is read from the data ROM are performed at T1.

(3) ALU performs an operation in T1 and T2 and the output result is latched to an Acc in one T2.

(4) The input data to the multiplier is set in one at T1. At the same time, a multiplication is performed and the its result is output at the next T1.

(5) An ALU operation and multiplication by the multiplier can be simultaneously performed in one instruction cycle. A typical example is as follows:

Example: When the contents of RAM are input to the H and L registers while the contents of the M register are are being stored in ALU's input P and those of Acc. to U, the result of the ALU operation is output to the Acc and the result of the multiplier, to the M and N registers.

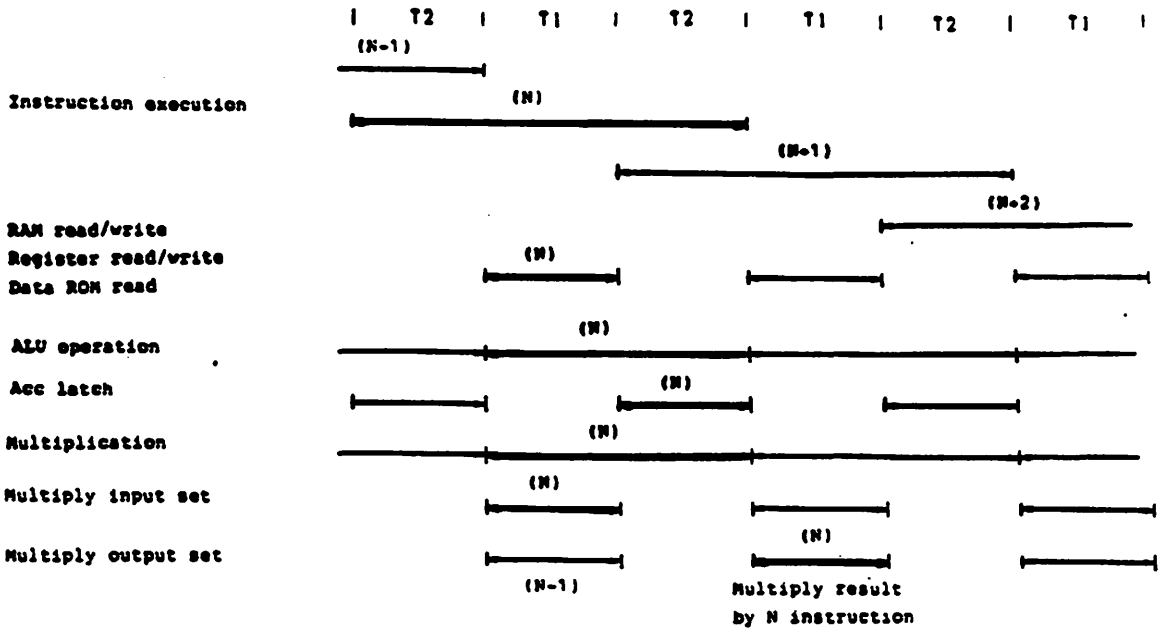


Fig. 4-2 Instruction Execution Timing

4.2 Outline of Instructions

All instructions of the uPD77C25 are one word instruction; one instruction is composed of 24 bits. The instructions are divided into the following four types:

(1) OP (Operation) instruction

This Type of instruction is used to perform operations such as ordinary arithmetic operations and data transfer.

(2) RT (Return) instruction

This instruction returns program execution from a sub-routine and interrupt process to the main routine. At the same time, the same operation as the OP instruction performs can be specified.

(3) JP (Jump) instruction

This instruction causes unconditional or conditional jump of program execution, or calls a subroutine.

(4) LD (Load) instruction

This loads 16-bit immediate data to the specified register.

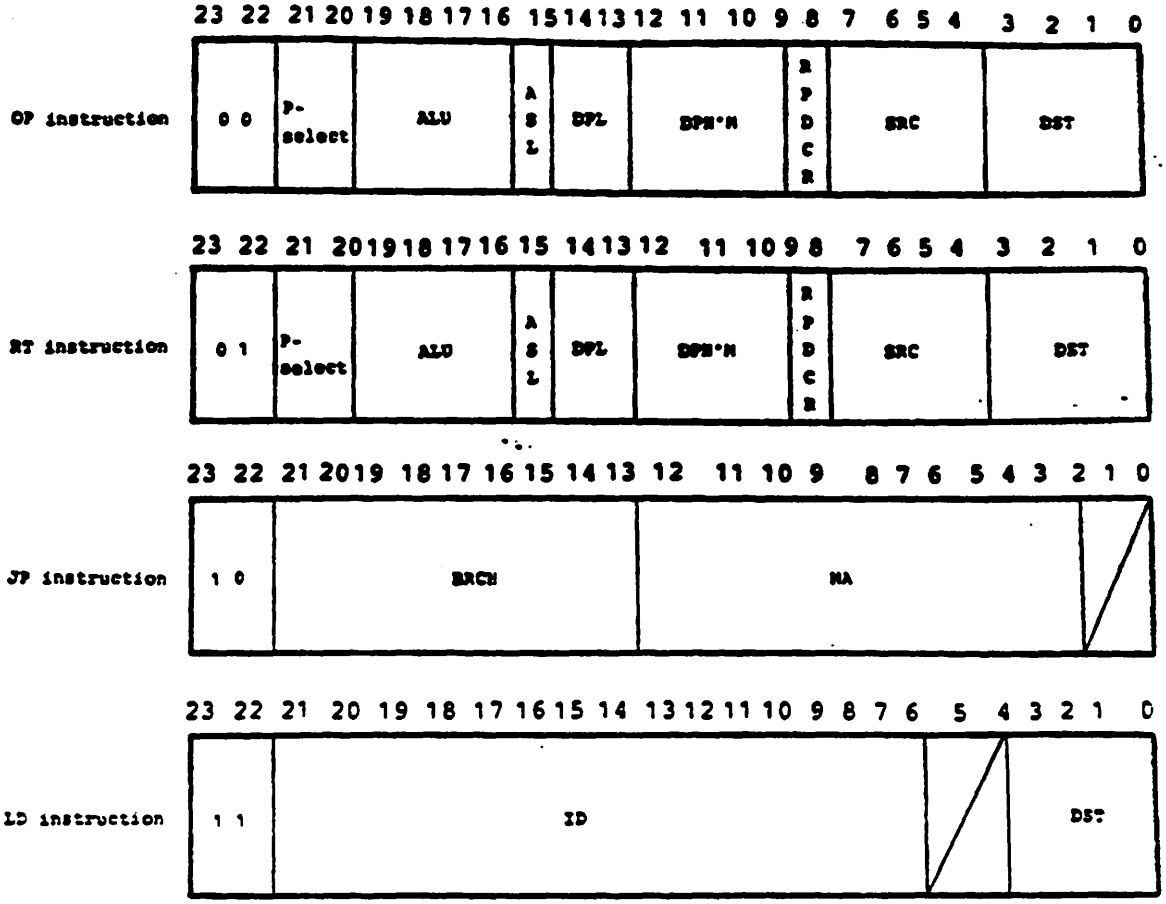


Fig. 4-3 Internal Configuration of Instruction

4.3 OP Instruction

The OP instruction is used to perform ordinary arithmetic operations and to transfer data. It consists of eight fields as shown in Fig. 4-4 and has the functions listed in Table 4-1.

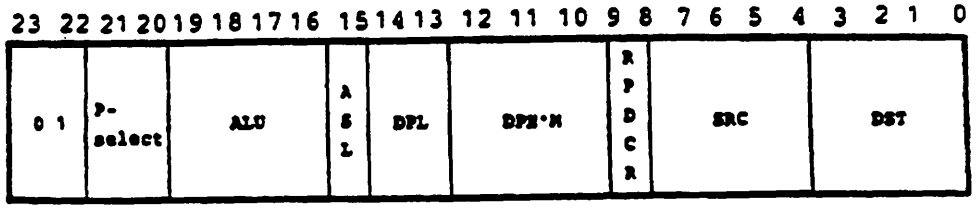


Fig. 4-4 Configuration of OP instruction

Table 4-1 Fields of OP Instruction

Field	Function
P-SELECT	Selects P input of ALU and specifies RAM, internal data bus, M register, or N register.
ALU	Specifies ALU operation. 16 operations are available.
ASL	Selects Q input of ALU and FLAG and specifies AccA and FLAGA or AccB and FLAGB. Unary operation is performed on Acc selected by this field and result is stored in the same Acc. Binary operation is performed between data specified by ASL and P-SELECT fields and the result is stored in the Acc specified by ASL field.
DPL	Specifies operation of lower 4 bits of DP (Data Pointer).
DPH,M	Modifies changes value of the upper 4 bits of DP (Data Pointer).
RPDCR	Specifies decrement operation of RP (ROM pointer).
SRC	Specifies source register for transfer operation. 16 registers can be specified.
DST	Specifies destination register for transfer operation. 16 registers can be specified.

4.3.1 Contents of P-SELECT field

The P-SELECT field specifies data to be input to the P input of ALU when ALU performs binary operation.

The contents of RAM, internal data bus, M register, or N register can be specified as the data for the operation.

Table 4-2 Contents of P-SELECT Field

Mnemonic	P-SELECT field		P input data
	D21	D20	
RAM	0	0	RAM
IDB	0	1	Internal data bus
M	1	0	M register
N	1	1	N register

4.3.2 Contents of ASL bit

The ASL bit specifies an Acc to be used for an operation and FLAG to store the status of the operation. The result of the operation is ignored if the Acc specified by the ASL bit is specified by the same instruction as a destination register for transfer operation.

Table 4-3 Contents of ASL bit

Mnemonic	ASL bit	Selection for Acc and FLAG
	D15	
ACCA	0	AccA, FLAGA
ACCB	1	AccB, FLAGB

4.3.3 Contents of ALU field

The ALU field specifies the operation to be performed by ALU. Sixteen operations which are listed in Table 4-4 can be specified. The detailed description of each operation is given on the following pages.

Table 4-4 A list of the Contents of ALU Field

Mnemonic	ALU field				Description of operation
	D19	D18	D17	D16	
NOP	0	0	0	0	No operation
OR	0	0	0	1	OR (Acc)←(Acc)∨(P)
AND	0	0	1	0	AND (Acc)←(Acc)∧(P)
XOR	0	0	1	1	Exclusive OR (Acc)←(Acc)⊻(P)
SUB	0	1	0	0	Subtract (Acc)←(Acc)-(P)
ADD	0	1	0	1	Add (Acc)←(Acc)+(P)
SBB	0	1	1	0	Subtract with Borrow (Acc)←(Acc)-(P)-(C)
ADC	0	1	1	1	Add with Carry (Acc)←(Acc)+(P)+(C)
DEC	1	0	0	0	Decrement Acc (Acc)←(Acc)-1
INC	1	0	0	1	Increment Acc (Acc)←(Acc)+1
CMP	1	0	1	0	Complement Acc(Acc)←$\overline{(Acc)}$
SHR1	1	0	1	1	1-bit R-Shift
SHL1	1	1	0	0	1-bit L-Shift
SHL2	1	1	0	1	2-bit L-Shift
SHL4	1	1	1	0	4-bit L-Shift
XCHG	1	1	1	1	8-bit Exchange

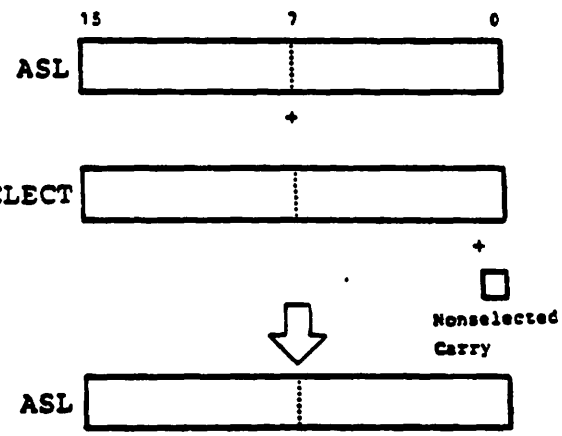
Contents of ALU Field:

D19	D18	D17	D16
0	1	1	1

Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Add data with a carry to the Acc.



Flag Operations:

FLAG specified by field

S1	S0	C	Z	OV1	OV0
↕	↕	↕	↕	↕	↕

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	—	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

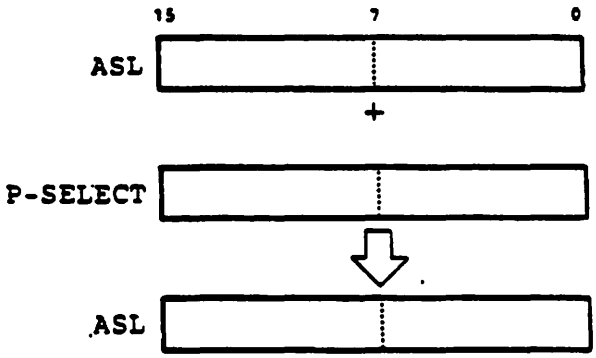
Contents of ALU Field:

D19	D18	D17	D16
0	1	0	1

Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one is specified by the DST field is specified, the result of the operation is ignored.

Operation:
Add data to the Acc.



Flag Operations:

ASL field specified FLAG

S1	S0	C	Z	OV1	OV0
↕	↕	↕	↕	↕	↕

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

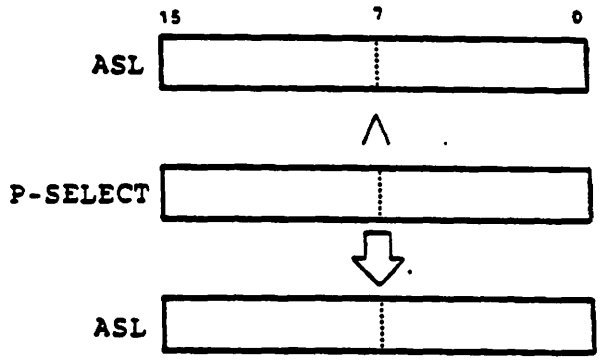
Contents of ALU Field:

D19	D18	D17	D16
0	0	1	0

Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

Storage location of operation result:-
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Logical AND each bit to the destination operand.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
x	↕	0	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

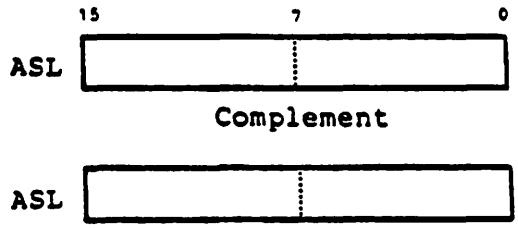
Contents of ALU Field:

D19	D18	D17	D16
1	0	1	0

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:
The result of the operation stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Inverts each bit of the Acc (1's complement).



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
x	↕	0	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

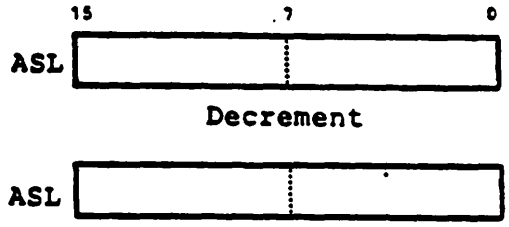
Contents of ALU Field:

D19	D18	D17	D16
1	0	0	0

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:-
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one is specified by the DST field is specified, the result of the operation is ignored.

Operation:
Decrements the Acc (subtract 1).



Flag Operations:

FLAG specified by ASL field

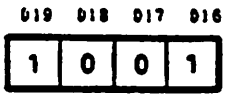
S1	S0	C	Z	OV1	OV0
↕	↕	↕	↕	↕	↕

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

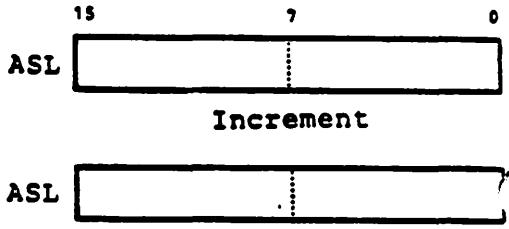
Contents of ALU Field:



Data subject to operation:
Acc specified by the ASL field.

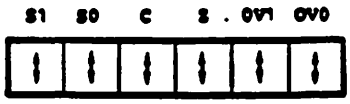
Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Increment the Acc (adds 1).

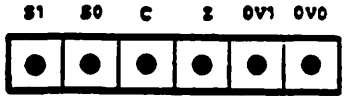


Flag Operations:

FLAG specified by ASL field



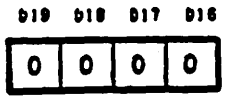
FLAG not selected



← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

NOP No Operation NOP

Contents of ALU Field:



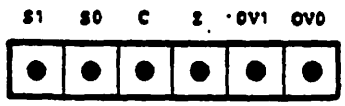
Data subject to operation:
None.

Storage location of operation result:
None.

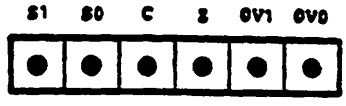
Operation:
Does not perform operation.

Flag Operations:

FLAG specified by ASL field



FLAG not selected



← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

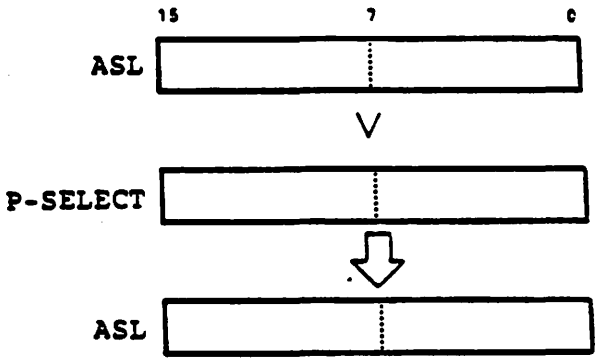
Contents of ALU Field:

D19	D18	D17	D16
0	0	0	1

Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
ORs each bit of the destination operand.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
x	↕	0	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

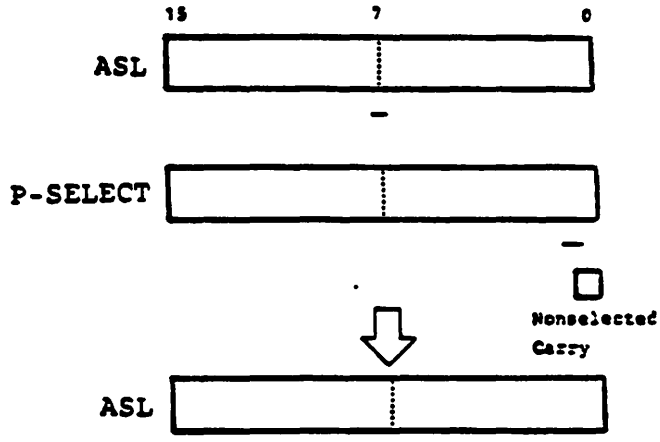
Contents of ALU Field:

D19	D18	D17	D16
0	1	1	0

Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Subtracts data with borrow from the Acc.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
↕	↕	↕	↕	↕	↕

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	—	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

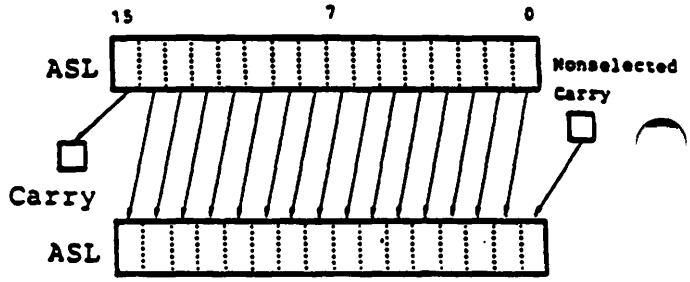
Contents of ALU Field:

D19	D18	D17	D16
1	1	0	0

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Shifts the contents of the Acc 1 bit to the left. The carry not selected is loaded into LSB of the shift result and MSB before shift is loaded into the carry flag.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
X	↕	↕	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	—	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

SHL2 2 bit Left-Shift

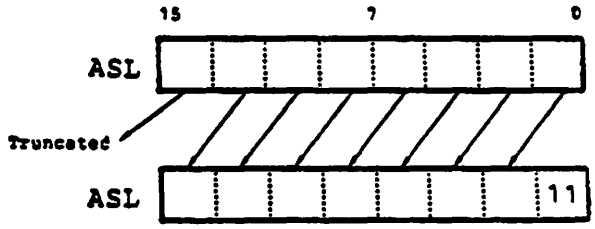
Contents of ALU Field:

I19	I18	I17	I16
1	1	0	1

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Shifts the contents of the Acc 2 bits to the left. 1s are loaded into the lower 2 bits of the shift result and the upper 2 bits before shift are truncated.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
x	↕	0	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OVC
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

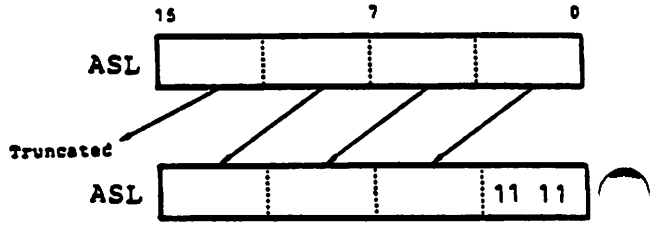
Contents of ALU Field:

D19	D18	D17	D16
1	1	1	0

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc is as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Shifts the contents of the Acc 4 bit to the left. 1s are loaded into the lower 4 bits of the shift result and the upper 4 bits before shift are truncated.



Flag Operations:

FLAG specified by ASL field

S1	S0	C	Z	OV1	OV0
×	↕	0	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
●	●	●	●	●	●

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
× : Undefined

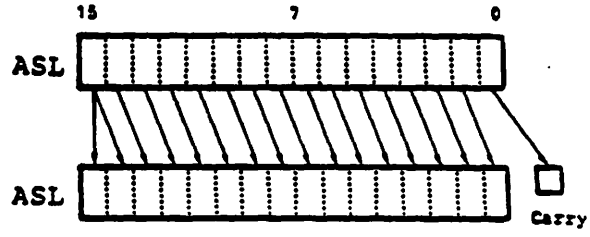
Contents of ALU Field:

D19	D18	D17	D16
1	0	1	1

Data subject to operation:
Acc specified by the ASL field.

Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Shifts the contents of the Acc 1 bit to the right. The MSB before shift is loaded into MSB of the shift result and LSB before shift is loaded into the carry flag.



Flag Operations:

FLAG specified by ASL field

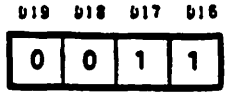
S1	S0	C	Z	OV1	OV0
X	↕	↕	↕	0	0

FLAG not selected

S1	S0	C	Z	OV1	OV0
•	•	•	•	•	•

← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
• : Retains previous status
x : Undefined

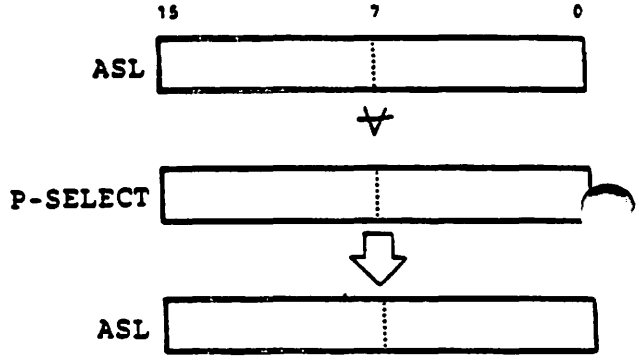
Contents of ALU Field:



Data subject to operation:
RAM, M register, N register, and internal data bus specified by the P-SELECT field and Acc specified by the ASL field.

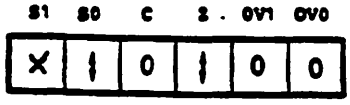
Storage location of operation result:
The result of the operation is stored in the Acc specified by the ASL field. If the same Acc as the one specified by the DST field is specified, the result of the operation is ignored.

Operation:
Exclusively ORs each bit of the destination operand.

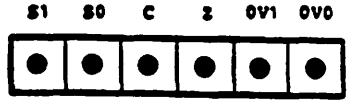


Flag Operations:

FLAG specified by ASL field



FLAG not selected



← : Affects result of operation
↕ : Affected by result of operation
0 : Cleared to 0
1 : Set to 1
● : Retains previous status
x : Undefined

4.3.4 Contents of DPL field

This field specifies the operation of the lower 4 bits of DP (Data Pointer). The new value of DPL specified by this field becomes valid from the next instruction. If data writing to DP is specified by the same instruction, the specification by this field is ignored.

Table 4-5 The contents of DPL field

Mnemonic	DPL field		Operation
	D14	D13	
DPNOP	0	0	No Operation
DPINC	0	1	Increment DPL
DPDEC	1	0	Decrement DPL
DPCLR	1	1	Clear DPL

4.3.5 Contents of DPH.M field

The value of this field is modified exclusively ORed with the value of the upper 4 bits in DP (Data Pointer). The result is input to DPH. The modified value of DPH valid from the next instruction. If data writing to DP is specified by the same instruction, the specification by this field is ignored.

Table 4-6 Contents of DP_HM Field

Mnemonic	DP_HM field				Qualification
	D12	D11	D10	D9	
M0	0	0	0	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 0 0 0)
M1	0	0	0	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 0 0 1)
M2	0	0	1	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 0 1 0)
M3	0	0	1	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 0 1 1)
M4	0	1	0	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 1 0 0)
M5	0	1	0	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 1 0 1)
M6	0	1	1	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 1 1 0)
M7	0	1	1	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(0 1 1 1)
M8	1	0	0	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 0 0 0)
M9	1	0	0	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 0 0 1)
MA	1	0	1	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 0 1 0)
MB	1	0	1	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 0 1 1)
MC	1	1	0	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 1 0 0)
MD	1	1	0	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 1 0 1)
ME	1	1	1	0	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 1 1 0)
MF	1	1	1	1	(DP7 DP6 DP5 DP4)←(DP7 DP6 DP5 DP4) ∀(1 1 1 1)

4.3.6 Contents of RPDCR bit

This field specifies whether to decrement the contents of RP (ROM Pointer). The value of RP changed by this field becomes valid from the next instruction. If data writing to RP is specified by the same instruction, the specification by this field is ignored.

Table 4-7 Contents of the RPDCR Bit

Mnemonic	RPDCR Bit D8	Operation
RPNOP	0	No operation
RPDEC	1	Decrement DPL

4.3.7 Contents of SRC field

This field specified the source register for a transfer operation. The contents of the register specified are output to the internal data bus. When AccA, AccB, RP, or DP is specified and if an operation or contents manipulation (DPL, DPHM, or RPDCR) is specified by the same instruction, the value before executing the instruction is output.

Table 4-8 Contents of SRC field

Mnemonic	SRC field				Specified register
	D7	D6	D5	D4	
NON#1 TRB	0	0	0	0	TRB register
A	0	0	0	1	AccA register
B	0	0	1	0	AccB register
TR	0	0	1	1	TR register
DP	0	1	0	0	DP register
RP	0	1	0	1	RP register
RO	0	1	1	0	RO register
SGN	0	1	1	1	SGN register
DR	1	0	0	0	DR register
DRNF	1	0	0	1	DR register#2
SP	1	0	1	0	SP register

SIM	1	0	1	1	SI register (1st ->MSB)*3
SIL	1	1	0	0	SI register (1st ->LSB)*4
K	1	1	0	1	K register
L	1	1	1	0	L register
MEM	1	1	1	1	RAM

*1 The contents of the TRB register are also output if NON is specified.

*2 Although the contents of the DR register are output to the internal data bus, the ROM flag is not set. Neither is the DRQ flag in DMA mode.

*3 With the 16-bit data, the serial data input first is output to the MSB of the internal data bus; the data input last is output to the LSB.

*4 With the 16-bit data, the serial data input first is output to the LSB of the internal data bus; the data input last is output to the MSB.

4.3.8 Contents of DST field

This field specifies the destination register for a transfer operation. The data output to the internal data bus is written to the specified register. When AccA, AccB, RP, or DP is specified and if an operation or contents manipulation (DPL, DPH,M, or RPDCR) is specified by the same instruction, the execution result is ignored.

If KLR or LKM is specified in the DST field, the K or L register cannot be specified by the SRC field. If RAM is specified by the SRC field, RAM cannot be specified by the DST field. (Transfer from RAM to RAM cannot be performed.)

Table 4-9 Contents of DST Field

Mnemonic	DST field				Specified register
	D3	D2	D1	D0	
@NON	0	0	0	0	No specified register
@A	0	0	0	1	AccA register
@B	0	0	1	0	AccB register
@TR	0	0	1	1	TR register
@DP	0	1	0	0	DP register
@RP	0	1	0	1	RP register
@DR	0	1	1	0	DR register
@SR	0	1	1	1	SR register
@SOL	1	0	0	0	SO register (LSB ->1st)#1
@SOM	1	0	0	1	SO register (MSB ->1st)#2
@K	1	0	1	0	K register
@KLR	1	0	1	1	KLR#3
@KLM	1	1	0	0	KLM#4
@L	1	1	0	1	L register
@TRB	1	1	1	0	TRB register
@MEM	1	1	1	1	RAM

#1 With 16-bit data, the serial output is sequentially performed from the LSB bit of the internal data bus.

#2 WITH 16-bit data, the serial output is sequentially performed from the MSB of the internal data bus.

#3 The data on the internal bus and the output from the RO register (ROM) are set to the K and L registers, respectively.

#4 The data on the internal bus and the contents of RAM (DP7, "1", DP5, DP4, DP3, DP2, DP1, and DP0) specified by DP6=("1") are set to the L and K registers, respectively.

4.4 RT Instruction

This instruction returns program execution from a subroutine or interrupt process. At the same time, the same operation and transfer as those specified by the OP instruction can be specified. As shown in Fig. 4-5, this instruction consists of eight fields, like the OP instruction, and restores the return address saved to the stack to PC after performing the same operation as the OP instruction.

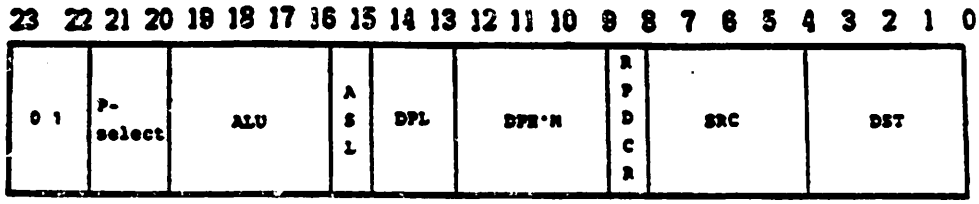


Fig. 4-5 Configuration of RT instruction

4.5 JP Instruction

The JP instruction causes program execution of jump unconditionally or conditionally. It also calls a subroutine.
As shown in Fig. 4-6, this instruction consists of two fields which specify an instruction and a jump address.

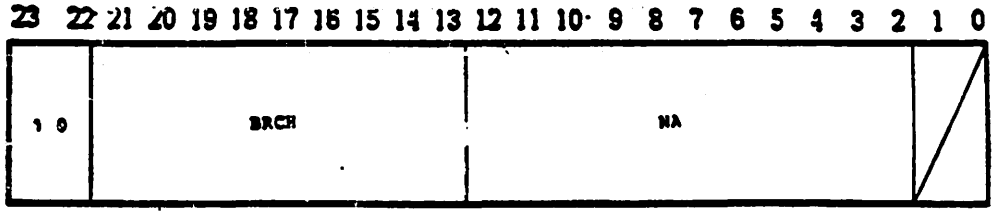


Fig. 4-6 Configuration of JP instruction

Table 4-10 Fields in JP instruction

Field	Function
BRCH	Specifies type of jump instruction and condition of conditional jump instruction.
NA	Specifies jump address.

4.5.1 Contents of BRCH field

This field specifies a type of jump instruction or the condition of a conditional jump instruction. If the condition is met in a conditional jump instruction, or unconditional jump instruction, the address specified by the NA field is transferred to PC. In a subroutine call instruction, an address where this instruction is stored plus one will be transferred to the stack and the address specified by the NA field is transferred to PC.

Table 4-11 Contents of BRCH Field

Mnemonic	BRCH Field									Condition
	D 21	D 20	D 19	D 18	D 17	D 16	D 15	D 14	D 13	
JMP	1	0	0	0	0	0	0	0	0	Unconditional jump
CALL	1	0	1	0	0	0	0	0	0	Unconditional call
JNCA	0	1	0	0	0	0	0	0	0	CA=0
JCA	0	1	0	0	0	0	0	1	0	CA=1
JNCB	0	1	0	0	0	0	1	0	0	CB=0
JCB	0	1	0	0	0	0	1	1	0	CB=1
JNZA	0	1	0	0	0	1	0	0	0	ZA=0
JZA	0	1	0	0	0	1	0	1	0	ZA=1
JNZB	0	1	0	0	0	1	1	0	0	ZB=0
JZB	0	1	0	0	0	1	1	1	0	ZB=1
JNO\AO	0	1	0	0	1	0	0	0	0	O\AO=0
JO\AO	0	1	0	0	1	0	0	1	0	O\AO=1
JNO\BO	0	1	0	0	1	0	1	0	0	O\BO=0
JO\BO	0	1	0	0	1	0	1	1	0	O\BO=1
JNO\A1	0	1	0	0	1	1	0	0	0	O\A1=0
JO\A1	0	1	0	0	1	1	0	1	0	O\A1=1
JNO\B1	0	1	0	0	1	1	1	0	0	O\B1=0

IQ\B1	0	1	0	0	1	1	1	1	0	Q\B1=1
INSA0	0	1	0	1	0	0	0	0	0	SA0=0
ISA0	0	1	0	1	0	0	0	1	0	SA0=1
INSB0	0	1	0	1	0	0	1	0	0	SB0=0
ISB0	0	1	0	1	0	0	1	1	0	SB0=1
INSA1	0	1	0	1	0	1	0	0	0	SA1=0
ISA1	0	1	0	1	0	1	0	1	0	SA1=1
INSB1	0	1	0	1	0	1	1	0	0	SB1=0
ISB1	0	1	0	1	0	1	1	1	0	SB1=1
IDPL0	0	1	0	1	1	0	0	0	0	DPL=0
IDPLN0	0	1	0	1	1	0	0	0	1	DPL≠0
IDPLF	0	1	0	1	1	0	0	1	0	DPL=F (HEX)
IDPLNF	0	1	0	1	1	0	0	1	1	DPL≠F (HEX)
INS1AK	0	1	0	1	1	0	1	0	0	S1 ACK=0
IS1AK	0	1	0	1	1	0	1	1	0	S1 ACK=1
INS0AK	0	1	0	1	1	1	0	0	0	S0 ACK=0
IS0AK	0	1	0	1	1	1	0	1	0	S0 ACK=1
INRQM	0	1	0	1	1	1	1	0	0	RQM=0
IRQM	0	1	0	1	1	1	1	1	0	RQM=1

4.6 LD Instruction

This is an immediate data load instruction which loads 16-bit data to the specified register. As shown in Fig. 4-7, this instruction is configured of two fields. The 16-bit data specified by the ID-field is loaded to the register specified by the DST field through the internal data bus. This DST field is same as the DST field in the OP instruction.

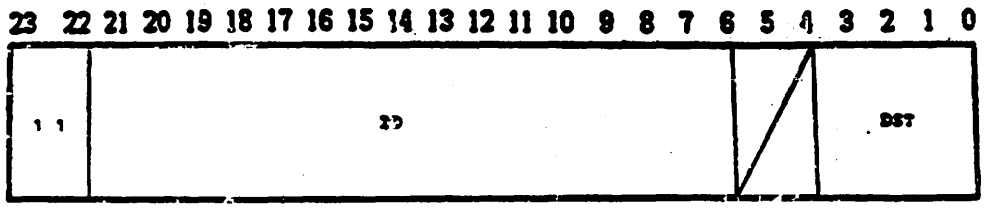


Fig. 4-7 Configuration of LD Instruction

CHAPTER 5 PROGRAM EXAMPLE

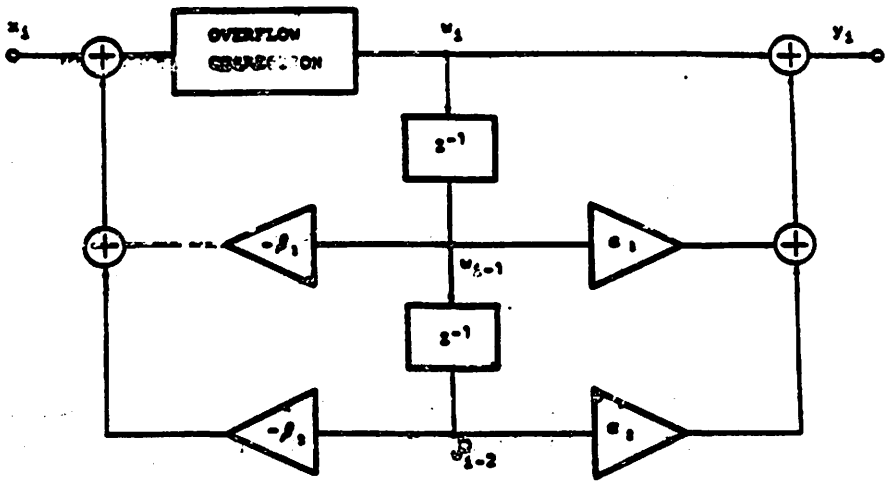
5.1 Biquad Digital Filter

The program for the Biquad Filter in Fig. 5-1 is shown in Table 5-1. As can be seen from the table, this program performs the Biquad Filter having four coefficients with only nine instructions.

Table 5-1 Biquad filter Program

Assumption: Range of coefficient $-2 \le -\beta_1, -\beta_2 < 2$
Initial conditions ACCA=x_{-1}, OVA0=0, OVA1=0,
RAM output=y_{-1},
ROM output=$(-\beta_1/2)$

INST STEP	ALU OPERATION	REGISTER SET	NEXT ADDRESS	INSTRUCTION
1	ACCA ← ACCA − RAM $/x_0/$ $/w_{0-0}/$	K ← RAM . L ← ROM $/w_{0-0}/$ $/-\beta_1/2/$	RAM address modify ROM address decrement	OP MOV . SUB MI RPDEC ORLR.MRM ACCA.IDB
2	ACCA ← ACCA + M $/x_0-w_{0-0}/$ $/(-\beta_1/2)w_{0-0}/$	K ← RAM . L ← ROM $/w_{0-0}/$ $/-\beta_2/$	ROM address decrement	OP MOV . ADD RPDEC ORLR.MRM .ACCA.M
3	ACCA ← ACCA + M $/x_0-\beta_2 \cdot w_{0-0}/$ $/-\beta_2 \cdot w_{0-0}/$	K ← RAM . L ← ROM $/w_{0-0}/$ $/x_0/$	ROM address decrement	OP MOV ADD RPDEC ORLR.MRM ACCA.M
4	JP TO 6 IF OVA1 = 0			JNOVA1 X
5	ACCA ← SGN /1·max or −max/			OP MOV OA.SGN
6	ACCA ← ACCA + M $/x_0/$ $/x_0 \cdot w_{0-0}/$	TR ← ACCA(OLD) $/x_0/$	RAM address modify	X: OP MOV ADD MI OTR,A ACCA.M
7	ACCA ← ACCA + RAM $/x_0(=c_0 \cdot w_{0-0}$ $/w_{0-0}/$	K ← RAM . L ← ROM $/w_{0-0}/$ $/x_0-1/$	ROM address decrement	OP MOV ADD RPDEC ORLR.MRM ACCA.IDB
8	ACCA ← ACCA + M $/x_0 \cdot w_{0-0} \cdot (a_0 \cdot w_{0-0}/(a_0-1)w_{0-0}/$	RAM ← TR $/x_0/$	RAM address modify	OP MOV ADD MI OMRM.TR ACCA.M
9		RAM ← K $/w_{0-0}/$		OP MOV RPINC MI RET OMRM.K



$$y_i = w_i + a_1 \cdot w_{i-1} + a_2 \cdot w_{i-2}$$

$$w_i = x_i - \beta_1 \cdot w_{i-1} - \beta_2 \cdot w_{i-2}$$

Fig. 5-1 Biquad Digital Filter

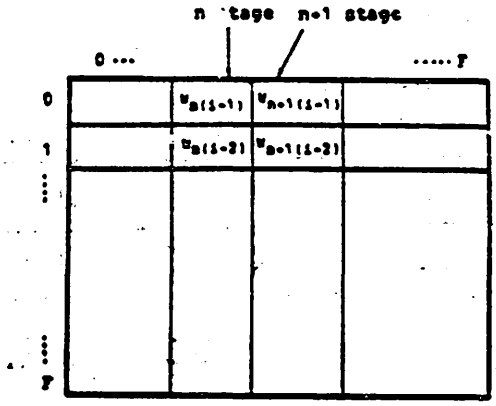


Fig. 5-2 RAM Map

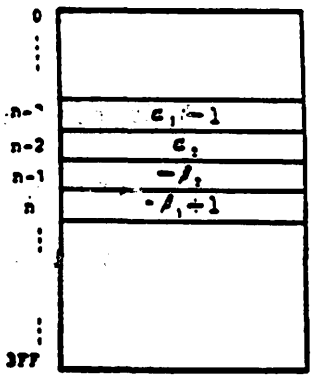


Fig. 5-3 Data ROM Map

CHAPTER 6 SYSTEM CONFIGURATION EXAMPLES

The uPD77C25 is provided with an 8-bit data bus that transfers as signals for interfacing with the host CPU. The signal processor thus offers a wide range of applications. Having ROM and RAM, the uPD77C25 can also be used as a single-chip CPU.

6.1 Application as I/O processor

When using the uPD77C25 as an I/O processor, its 8-bit data bus is utilized. This method is useful for configuring PBR (Push Button Receiver), etc. Fig. 6-1 shows a system configuration example when the uPD77C25 is used as an I/O processor.

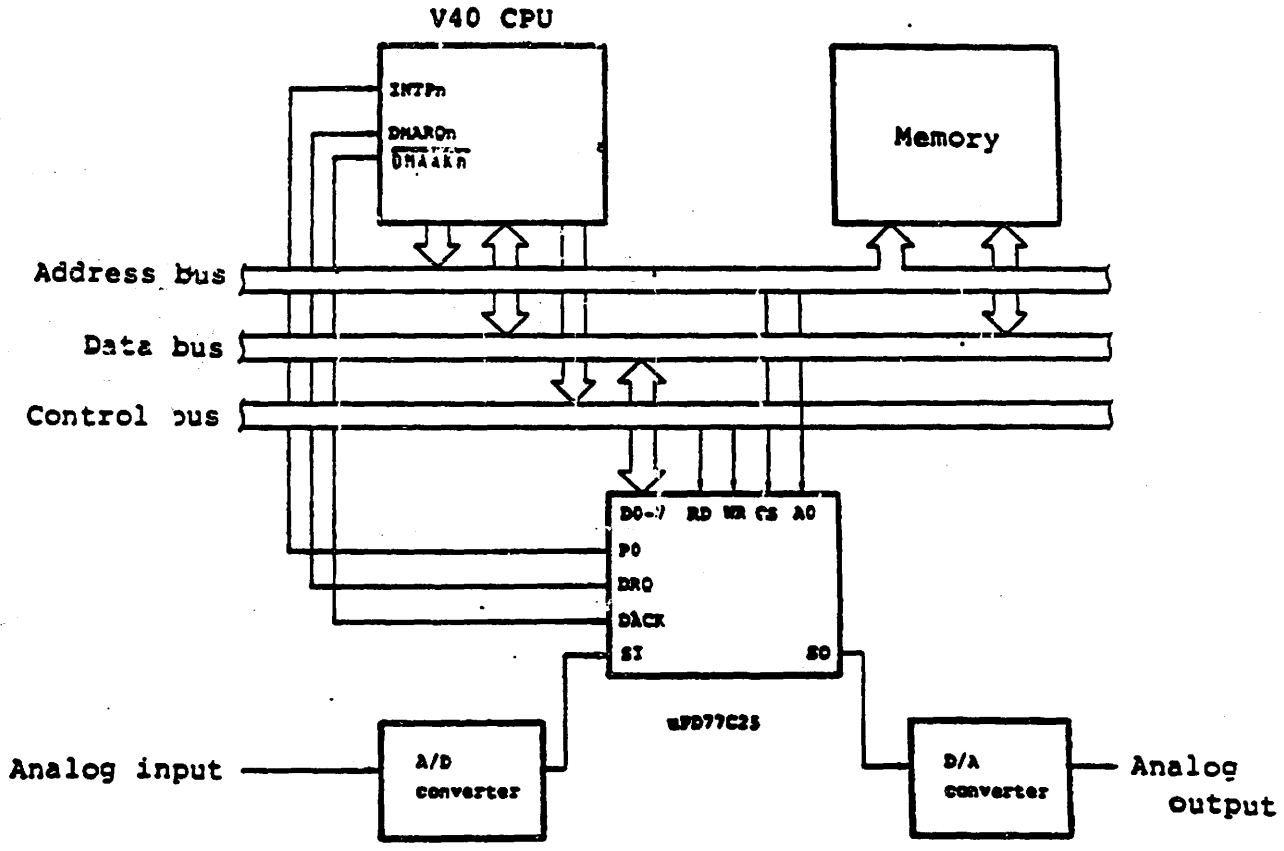


Fig. 6-1 Application Example As I/O Processor (Reference Diagram)

6.2.1 Application as single-chip CPU

When used as a single-chip CPU, the uPD77C25 is suitable for configuring a small-scale system in which a digital filter is made up with an A/D and a D/A converter connected to the input and output. Fig. 6-2 shows an example of two-chip system configuration. It is also possible to combine this application example with the application example as an I/O processor illustrated in Fig. 6-1.

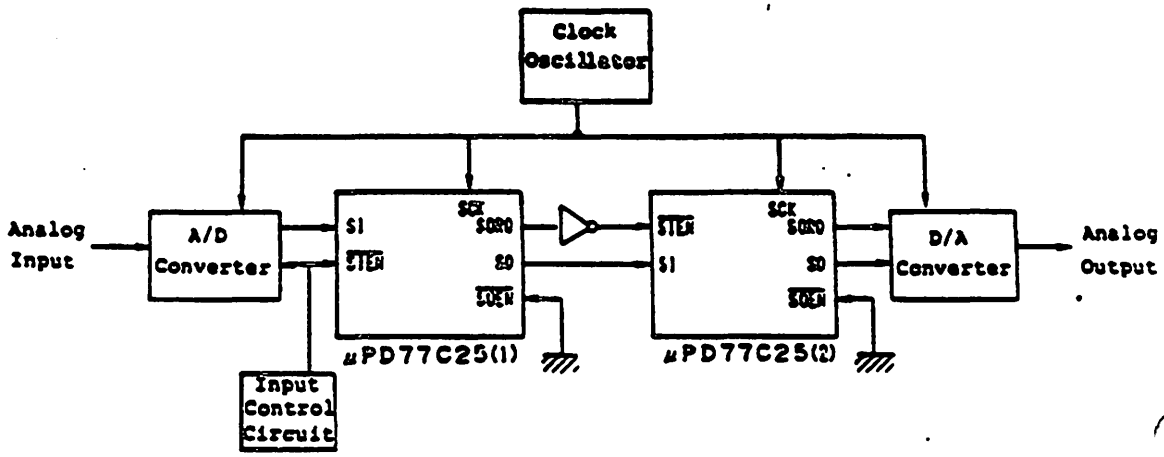


Fig. 6-2 Application Example As Single-Chip Processor (Reference Diagram)

REGIONAL SALES AND ENGINEERING SUPPORT OFFICES

NORTHEAST
Twenty Burlington Mall Road, Suite 449
Burlington, MA 01803
TEL 617-272-1774 TWX 710-348-6515

6 Neshaminy Interplex, Suite 205
Trevose, PA 19047
TEL 215-638-8989 TWX 510-224-6090

SOUTHEAST
Radice Corporate Center
600 Corporate Drive, Suite 412
Fort Lauderdale, FL 33334
TEL 305-776-0682 TWX 759839

MIDWEST
1500 West Shure Drive, Suite 250
Arlington Heights, IL 60005
TEL 312-577-8630 TWX 910-687-1492

SOUTHCENTRAL
16475A Dallas Parkway, Suite 380
Dallas, TX 75248
TEL 214-931-0641 EZLINK 62901906

SOUTHWEST
200 East Sandpointe, Bldg. 8 Suite 480
Santa Ana, CA 92707
TEL 714-546-0501 TWX 759845

NORTHWEST
10080 North Wolfe Road, SW3 Suite 360
Cupertino, CA 95014
TEL 408-973-0850 TLX 5[illegible]7

DISTRICT OFFICES

200 Broadhollow Road, Suite 302
Route 110
Melville, NY 11747
TEL 516-423-3300 TWX 510-224-6090

2 Jefferson Street, Suite 103
Poughkeepsie, NY 12601
TEL 914-452-4747 TWX 510-248-0066

200 Perinton Hills Office Park
Fairport, NY 14450
TEL 716-425-4590 TWX 510-100-8949

5720 Peachtree Parkway, Suite 120
Norcross, GA 30092
TEL 404-447-4409 TWX 910-997-0150

29200 Southfield Road, Suite 208
Southfield, MI 48076
TEL 313-559-4242 TWX 810-224-4625

340 East Big Beaver Road, Suite 210
Troy, MI 48083
TEL 313-680-0506 TWX 810-224-4625

Busch Corporate Center
6480 Busch Boulevard, Suite 121
Columbus, OH 43229
TEL 614-4[illegible]-1778 TWX 510-101-1771

8080 Cedar Avenue South, Suite 229
Bloomington, MN 55420
TEL 612-854-4443 TWX 910-307-0726

DISTRICT OFFICES (cont)

1804 Santa Rosa Road, Suite 245
Richmond, VA 23288
TEL 804-282-8303 TWX 510-601-5015

Echelon Building 2
9430 Research Boulevard, Suite 330
Austin, TX 78759
TEL 512-346-9280 EZLINK 62811181

6150 Canoga Avenue, Suite 112
Woodland Hills, CA 91367
TEL 818-716-1535 TWX 559210

Lincoln Center Building
10300 S. W. Greenburg Road, Suite 540
Portland, OR 97223
TEL 503-245-1600

5445 DTC Parkway, Suite 218
Englewood, CO 30111
TEL 303-694-0041 TWX 510-600-5666

NATICK TECHNOLOGY CENTER
1 Natick Executive Park
Natick, MA 01760
TEL 617-655-3833 TWX 710-386-2110

NEC
NEC Electronics Inc.
CORPORATE HEADQUARTERS
401 Ellis Street
P.O. Box 7241
Mountain View, CA 94039
TEL 415-960-6000
TWX 910-379-3985

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

NECEL-002080-0387
STOCK NO. 800364