
V51™ 16-BIT MICROPROCESSOR

DESCRIPTION

The μ PD70280 (V51) is a 16-bit microprocessor containing a CPU equivalent to the μ PD70116H (V30HL™) and peripheral functions for the PC/XT™ on a single chip.

μ PD70280 employs a μ PD70116H equivalent circuit as the core of the CPU, enabling you to build a high-speed, low-voltage, and power-saving PC/XT-compatible system. In addition, it also contains a memory control unit (MCU) so that DRAM, SRAM, or pseudo SRAM can be easily connected.

The μ PD70270 (V41™) is identical to the μ PD70280, except that it is provided with an 8-bit external data bus.

FEATURES

- o Built-in peripheral units for PC/XT
 - Clock generator
 - Bus cycle generator
 - Bus interface unit
 - Timer/counter unit (equivalent to μ PD71054)
 - DMA control unit (equivalent to μ PD71037)
 - Interrupt control unit (equivalent to μ PD71059)
 - Speaker interface unit
 - Keyboard control unit (PC/XT mode, IBM PS/2™ model 30 mode (selectable))
 - Memory control unit (DRAM, SRAM, or pseudo-SRAM can be directly connected)
 - ROM decoder (for BIOS ROM, expansion ROM)
 - External I/O decoder (six different kinds of I/O devices can be directly controlled)
 - LIM EMS Ver.4.0 support circuit (EMS circuit)
- o μ PD70116H equivalent CPU
- o Operating voltage:
 - 3V (maximum operating frequency 8MHz)
 - 5V (maximum operating frequency 16MHz)
- o Minimum instruction execution time:
 - 125 ns (16MHz, 5V)
 - 250 ns (8MHz, 3V)
- o High-speed multiplication/division instructions:
 - 1.2 to 3.5 μ s (16MHz, 5V)
 - 2.4 to 7.0 μ s (8MHz, 3V)
- o Shadow RAM supported
- o Low power consumption
 - Power consumption at 3V is approximately 1/3 of that of at 5V (at the same frequency)
 - Clock supply can be stopped
 - External bus operation can be disabled
- o Software compatible with 16-bit V series™
- o 160-pin plastic QFP (Quad Flat Package)

The information in this document is subject to change without notice.

APPLICATIONS

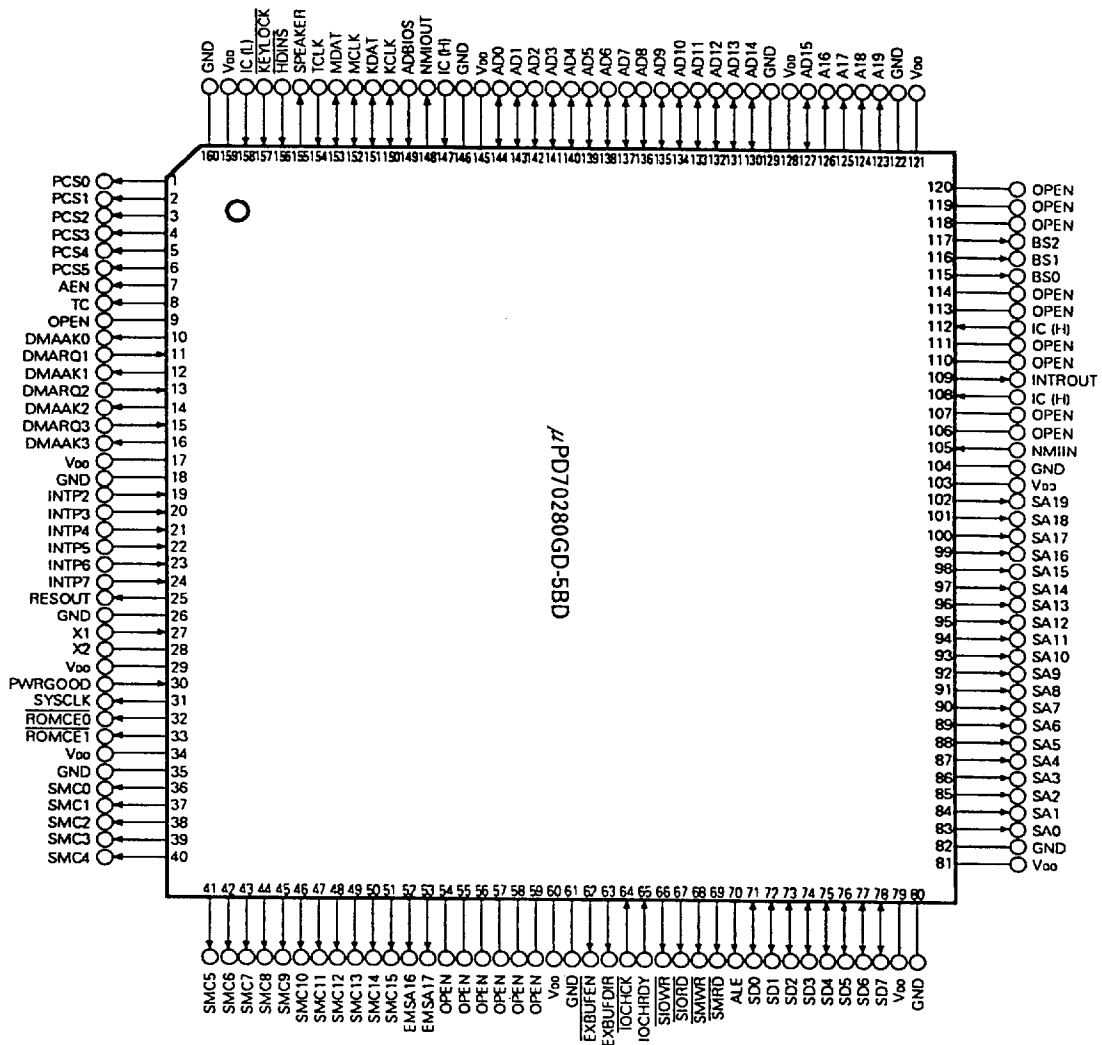
- o OA related devices (portable terminals, personal computers, word processors, electronic typewriters, multi-function telephones, etc.)
- o Control equipment (robotic control, NC control, communication control, etc.)

ORDERING INFORMATION

Part Number	Package	Quality Grade
μPD70280GD-5BD	160-pin plastic QFP (Quad Flat Package)	Standard

Please refer to "Quality Grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specifications of quality grade on the devices and their recommended applications.

PIN CONFIGURATION : 160-PIN PLASTIC QFP (TOP VIEW)

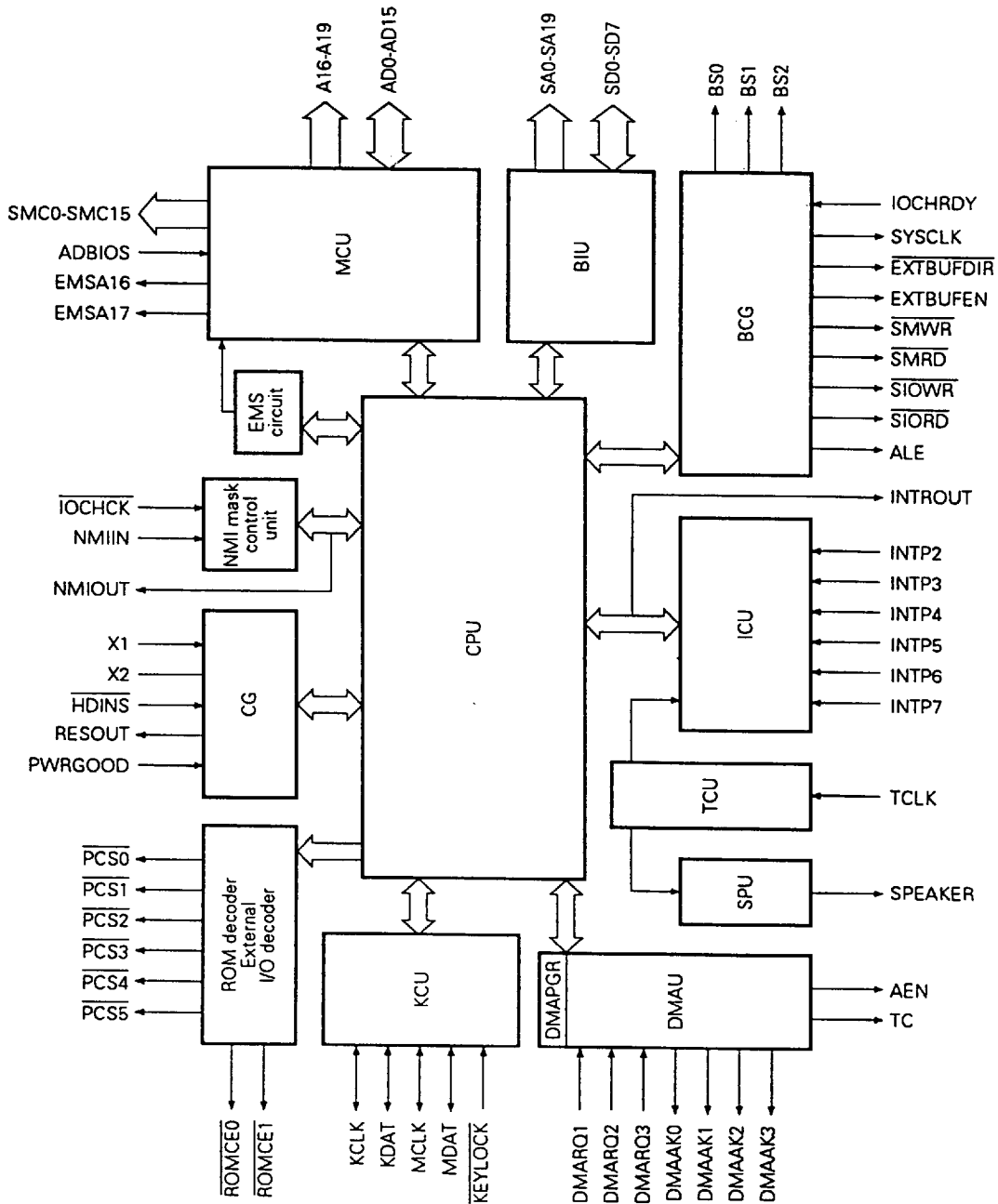


- Note:
1. Connect each IC(H) pin to a VDD through an external pull-up resistor.
 2. Connect each IC(L) pin to a GND through an external pull-down resistor.
 3. Do not connect anything to the OPEN pin.

PIN IDENTIFICATIONS

AD0-AD15	: Address/Data Bus
A16-A19	: Address Bus
ADBIOS	: Address of BIOS
AEN	: Address Enable
ALE	: Address Latch Enable
BS0-BS2	: Bus Status
DMAAK0-DMAAK3	: DMA Acknowledge
DMARQ1-DMARQ3	: DMA Request
EMSA16, EMSA17	: EMS Address
EXTBUFDIR	: External Buffer Direction
EXTBUFEN	: External Buffer Enable
GND	: Ground
HDINS	: Hard Disk Install
IC(H), IC(L)	: Internally Connected
INTP2-INTP7	: Interrupt Request from Peripheral
INTROUT	: Interrupt Request Out
IOCHCK	: I/O Channel Check
IOCHRDY	: I/O Channel Ready
KCLK	: Keyboard Clock
KDAT	: Keyboard Data
KEYLOCK	: Keyboard Lock
MCLK	: Mouse Clock
MDAT	: Mouse Data
NMIIN	: Non-Maskable Interrupt Input
NMIOUT	: Non-Maskable Interrupt Out
OPEN	: Open
PCS0-PCS5	: Programmable Chip Select
PWRGOOD	: Power Good
RESOUT	: Reset Out
ROMCE0, ROMCE1	: ROM Chip Enable
SA0-SA19	: System Address Bus
SD0-SD7	: System Data Bus
SIORD	: System I/O Read Strobe
SIOWR	: System I/O Write Strobe
SMC0-SMC15	: System Memory Control
SMRD	: System Memory Read Strobe
SMWR	: System Memory Write Strobe
SPEAKER	: Speaker
SYSCLK	: System Clock
TC	: Terminal Count
TCLK	: Timer Clock
V _{DD}	: Power
X1, X2	: Crystal

INTERNAL BLOCK DIAGRAM



- | | | | |
|------|---------------------------|--------|--------------------------|
| CPU | : Central Processing Unit | DMAPGR | : DMA Page Register |
| CG | : Clock Generator | ICU | : Interrupt Control Unit |
| BCG | : Bus Cycle Generator | SPU | : Speaker Interface Unit |
| BIU | : Bus Interface Unit | KCU | : Keyboard Control Unit |
| TCU | : Timer/Counter Unit | MCU | : Memory Control Unit |
| DMAU | : DMA Control Unit | | |

FUNCTIONAL OUTLINE

Item	Function	
CPU	Equivalent to the V30HL	
Internal Data Bus Width	16 bits	
External Data Bus Width	16 bits	
Operating Temperature	-40 to +85°C (tentative)	
Power Supply	V _{DD} = 5 V±10% (tentative): maximum frequency of 16MHz V _{DD} = 3 V±10% (tentative): maximum frequency of 8MHz	
Maximum Operating Frequency	V _{DD} = 5V	16MHz (at 32MHz, externally supplied)
	V _{DD} = 3V	8MHz (at 16MHz, externally supplied)
Minimum Instruction Execution Time*1	V _{DD} = 5V	125ns
	V _{DD} = 3V	250ns
Multiply/Division Instruction Execution Time*1	V _{DD} = 5V	1.2-3.5μs
	V _{DD} = 3V	2.4-7.0μs
Memory Space	16 Mbytes (with internal EMS circuit)	
I/O Space	64K bytes (32K words)*2	
Number of Instructions	101	
Number of Registers	12	
Wait Control	0 to 3 wait states can be inserted into the following cycles: • CPU cycle • DMA cycle • Refresh cycle	
Clock Generator (CG)	Maximum input frequency: 32MHz	
Timer/Counter Unit (TCU)	Equivalent to the μPD71054 (all 3 channels are fixed-use)	
Interrupt Control Unit (ICU)	Equivalent to the μPD71059 (vector mode only. 2 of 8 interrupt sources are fixed-use)	
DMA Control Unit (DMAU)	Equivalent to the μPD71037 • Address bus: 20 bits • DMA channel: 4 channels (1 channel is fixed-use)	
Memory Control Unit (MCU)	DRAM, SRAM, or pseudo SRAM can be directly connected	
Keyboard Control Unit (KCU)	PC/XT mode, or PS/2 model 30 mode (selectable)	
External I/O Decoder	CS signal generation for external I/O device (6 lines)	
ROM Decoder	CS signal generation for extended ROM and BIOS ROM	
Interrupt Function	External: 6, internal: 2	
NMI Function	2 input pins (IOCHCK, NMIIN)	
Supply Current with Clock Input Stopped	I _{DD} (MAX.) = 50 μA (tentative)	
Package	160-pin plastic QFP (Quad Flat Package)	
Others	• Bus cycle generator (BCG) • Bus interface unit (BIU) • Speaker interface unit (SPU)	

Note *1 : At maximum operating frequency
*2 : 2904 bytes of 64K-byte I/O space are reserved.

CONTENTS

1.	PIN FUNCTIONS	8
1.1	PIN IDENTIFICATION	8
1.2	SMC PIN FUNCTIONS	10
1.3	PIN STATUS UNDER SPECIFIC CONDITIONS	12
1.4	PROCESSING OF UNUSED PINS	13
2.	MEMORY AND I/O CONFIGURATION	14
2.1	MEMORY MAPPING	14
2.2	I/O SPACE	15
3.	INTERNAL PERIPHERAL UNIT FUNCTIONS	18
3.1	CPU	18
3.2	CG (CLOCK GENERATOR)	18
3.3	BCG (BUS CYCLE GENERATOR)	19
3.4	BIU (BUS INTERFACE UNIT)	19
3.5	SPU (SPEAKER INTERFACE UNIT)	20
3.6	TCU (TIMER/COUNTER UNIT)	21
3.7	ICU (INTERRUPT CONTROL UNIT)	22
3.8	NMI MASK CONTROL UNIT	23
3.9	DMAU (DMA CONTROL UNIT)	23
3.10	DMA PGR (DMA PAGE REGISTER)	24
3.11	KCU (KEYBOARD CONTROL UNIT)	25
3.12	ROM DECODER	25
3.13	EXTERNAL I/O DECODER	25
3.14	MCU (MEMORY CONTROL UNIT)	26
3.15	EMS CIRCUIT	26
4.	INSTRUCTION SET	29
5.	PRELIMINARY ELECTRICAL SPECIFICATIONS	60
5.1	WHEN SUPPLY VOLTAGE $V_{DD} = 5 V \pm 10\%$	60
5.2	WHEN SUPPLY VOLTAGE $V_{DD} = 3 V \pm 10\%$	67
6.	PACKAGE DRAWINGS	78

1. PIN FUNCTIONS

1.1 PIN IDENTIFICATION

Symbol	I/O	Function
X1	Input	When using internal CG : Crystal is connected
X2	-	When using external CG : Inputs external clock
PWRGOOD	Input	System reset input
RESOUT	Output	System reset output
SMC0-SMC15	Output	Address signal and control signal output to system memory.
EMSA16,EMSA17	Output	EMS address bit output
AD0-AD15	3-state I/O	Address signal output/Data input (time division multiplex)
A16-A19	Output	Address signal output
ADBIOS	Input	BIOS ROM connection bus selection
$\overline{\text{ROMCE0}}$	Output	BIOS ROM chip enable output
$\overline{\text{ROMCE1}}$	Output	Extended ROM chip enable output
PCS0-PCS5	Output	I/O chip select output
KCLK	3-state I/O	Channel A keyboard clock input/output
KDAT	3-state I/O	Channel A keyboard data input/output
MCLK	3-state I/O	Channel B keyboard clock input/output (for mouse)
MDAT	3-state I/O	Channel B keyboard data input/output (for mouse)
TCLK	Input	Timer clock input
SPEAKER	Output	Speaker data output
$\overline{\text{HDINS}}$	Input	Hard disk connection status (connected/unconnected) input
$\overline{\text{KEYLOCK}}$	Input	Keyboard lock status (locked/unlocked) input
SYSCLK	Output	SD bus system clock output
SA0-SA19	Output	SD bus address output
SD0-SD7	3-state I/O	SD bus data input/output
ALE	Output	SD bus address latch output
BS0-BS2	Output	Bus status signal output
$\overline{\text{SMWR}}$	Output	Memory write strobe output for memory connected to SD bus.
$\overline{\text{SMRD}}$	Output	Memory read strobe output for memory connected to SD bus.

Symbol	I/O	Function
$\overline{\text{SIOWR}}$	Output	I/O write strobe output for I/O device connected to SD bus.
$\overline{\text{SIORD}}$	Output	I/O read strobe output for I/O device connected to SD bus.
IOCHRDY	Input	SD bus ready input
$\overline{\text{IOCHCK}}$	Input	SD bus parity check output
NMIIN	Input	Non-maskable interrupt request signal input from external source (internally ORed with IOCHCK input)
NMIOUT	Output	Signal output to external device to notify generation of NMI request to internal CPU
INTP2-INTP7	Input	Interrupt request input for ICU
INTROUT	Output	Signal output to external device to notify generation of maskable interrupt request (from ICU) to internal CPU
DMARQ1-DMARQ3	Input	DMA transfer request input for DMAU
DMAAK0-DMAAK3	Output	DMA transfer acknowledge output from DMAU
AEN	Output	DMAU bus control status (control is given /not given) output
TC	Output	Terminal count output from DMAU
$\overline{\text{EXTBUFEN}}$	Output	External data bus buffer control output
EXTBUFDIR	Output	External data bus buffer direction output
V _{DD}	-	Positive power supply
GND	-	Ground
IC(L), IC(H)	-	Internal connection; leave unconnected
OPEN	-	Reserved for function expansion

1.2 SMC PIN FUNCTIONS

The functions for SMC pins (SMC0-SMC15) change, depending on the kind of memory connected.

Table 1-1 lists the SMC pin functions, the maximum number of memory banks, and the maximum memory size for three different DRAM sizes, two different SRAM sizes, and two different pseudo-SRAM sizes.

Table 1-1 SMC Pin Functions (1/2)

(a) Relationship of SMC Pins and Memory

Pin name	Memory	DRAM						SRAM		Pseudo-SRAM	
	Configuration*	256Kx1	256Kx4	1Mx1	1Mx4	4Mx1	4Mx4	32K x 8	128K x 8	32K x 8	128K x 8
	Maximum number of banks	4		3		2		10	10	10	9
	Maximum size	2Mbytes		6Mbytes		16Mbytes		640Kbytes	2.5Mbytes	640Kbytes	2.25Mbytes
SMC0		MA0						CS0			
SMC1		MA1						CS1			
SMC2		MA2						CS2			
SMC3		MA3						CS3			
SMC4		MA4						CS4			
SMC5		MA5						CS5			
SMC6		MA6						CS6			
SMC7		MA7						CS7			
SMC8		MA8						CS8			
SMC9		CAS3	MA9					CS9		RFSH	
SMC10		CAS2			MA10			NC	EMSA14	NC	EMSA14
SMC11		CAS1						NC	EMSA15	NC	EMSA15
SMC12		CAS0						OE \bar{H}		OE \bar{E} /RFSH \bar{H}	OE \bar{H}
SMC13		RASH						OEL		OE \bar{E} /RFSH \bar{L}	OEL
SMC14		RASL						WE \bar{H}			
SMC15		WE						WE \bar{L}			

*: Words x bits

Table 1-1 SMC Pin Functions (2/2)

(b) Outline for each function

Item	Functional outline
MA0-MA9	DRAM address signal. Row address and column address are multiplexed
CAS0-CAS3	DRAM CAS signal. Selects bank
RASL, RASH	DRAM RAS signal. Selects upper memory, lower memory
WE	DRAM write enable signal
CS0-CS9	SRAM, pseudo-SRAM chip select signal
RFSH	Pseudo-SRAM refresh signal
OEL, OEH	SRAM, pseudo-SRAM OE signal
OE/RFSHL, OE/RFSHH	Pseudo-SRAM OE/RFSH signal
WEL, WEH	SRAM, pseudo-SRAM write enable signal
EMSA14, EMSA15	Upper address when using EMS
NC	No connection

1.3 PIN STATUS UNDER SPECIFIC CONDITIONS

Table 1-2 shows the pin statuses in the DMA cycle and bus hold state and on execution of the HALT instruction.

Table 1-2 Pin Status Under Specific Conditions

Pin name	Bus hold state	On execution of HALT instruction
RESOUT	No change	
SMC0-SMC15	Depends on DMA operation	No change (except during DMA operation)
EMSA16, EMSA17		
AD0-AD15		
A16-A19		
ROMCE0		
ROMCE1		
PCS0-PCS5		
SYSCLK	Low level (except during DMA operation)	
SA0-SA19	Depends on DMA operation	No change (except during DMA operation)
SD0-SD7		
ALE	Low level	
SMWR	Depends on DMA operation	No change (except during DMA operation)
SMRD		
SIOWR		
SIORD		
DMAAK0-DMAAK3	Depends on DMARQ	
AEN	High level	
TC	Output at count completion	
EXTBUFEN	Depends on DMA operation	
EXTBUFDIR		

1.4 PROCESSING OF UNUSED PINS

Table 1-3 shows the processing (recommended connections) of pins not used.

Table 1-3 Processing of Unused Pins

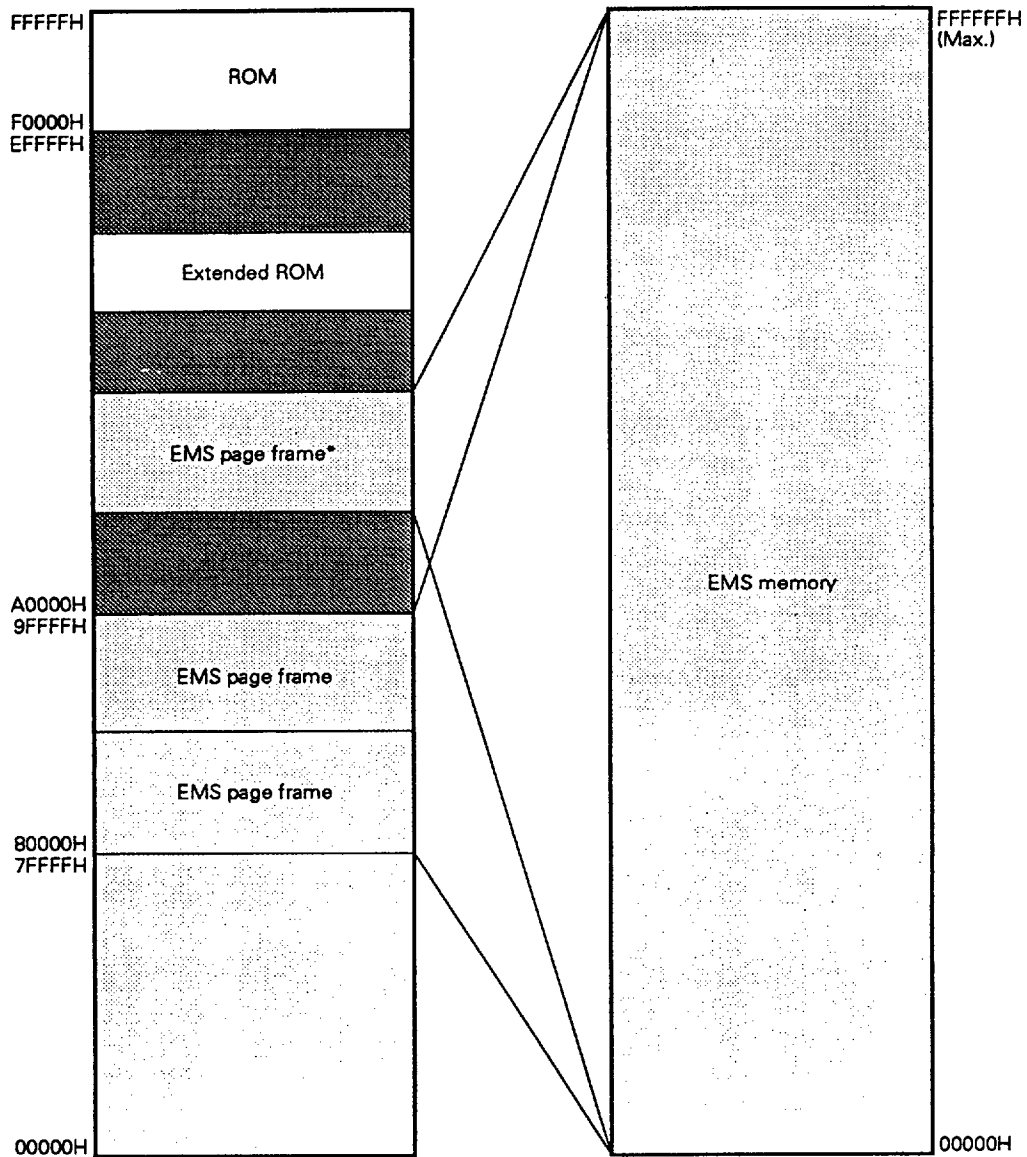
Pin name	I/O	Recommended connections
RESOUT	Output	Open
SMC0-SMC15	Output	
EMSA16, EMSA17	Output	
AD0-AD15	3-state I/O	Connect to V _{DD} through resistor
A16-A19	Output	Open
ROMCE ₀	Output	Open
ROMCE ₁	Output	
PCS0-PCS5	Output	
KCLK	3-state I/O	Connect to V _{DD} or GND through resistor
KDAT	3-state I/O	
MCLK	3-state I/O	
MDAT	3-state I/O	
TCLK	Input	
SPEAKER	Output	Open
SYSCLK	Output	
SA0-SA19	Output	
SD0-SD7	3-state I/O	Connect to V _{DD} or GND through resistor
ALE	Output	Open
BS0-BS2	Output	
SMWR	Output	
SMRD	Output	
SIOWR	Output	
SIORD	Output	
IOCHRDY	Input	Connect to V _{DD} through resistor
IOCHCK	Input	Connect to GND through resistor
NMIIN	Input	
NMIOUT	Output	Open
INTP2-INTP7	Input	Connect to V _{DD} or GND through resistor
INTROUT	Output	Open
DMARQ1-DMARQ3	Input	Connect to GND through resistor
DMAAK0-DMAAK3	Output	Open
AEN	Output	
TC	Output	
EXTBUFEN	Output	
EXTBUFDIR	Output	

2. MEMORY AND I/O CONFIGURATION

2.1 MEMORY MAPPING

The memories connected to the two buses (SD bus, AD bus) of the μPD70280 are managed by the memory control unit (MCU).

Fig. 2-1 shows the mapping for the SD bus and the AD bus.



*: Either addresses C0000H-CFFFFH, D0000H-DFFFFH, or E0000H-EFFFFH can be selected by the program.

Remarks 1 :  : Connected to SD bus
 : Connected to AD bus

2 : The bus used to connect the ROM can be selected by the hardware.

3 : The bus and the address used to connect the extended ROM can be selected by the software.

Fig. 2-1 Memory Mapping

2.2 I/O SPACE

The register for the internal peripheral unit is allocated in the I/O space, and read/written by the input/output instruction. 64K byte (16-bit address) I/O space is provided in the μPD70116H (V30HL) architecture. However, the upper side of the I/O address is not decoded in the IBM PC series, so that only 1K byte (10-bit address) of the I/O space can be distinguished and used.

The μPD70280 contains all the registers necessary to make it compatible with the PC/XT. In addition, extra registers are added to expand its features. The registers compatible with PC/XT registers will be allocated to the same address, in order to maintain the compatibility. However, the extra registers are allocated in the 2-byte I/O area, so as not to eat up the remaining I/O space. 1 byte of this I/O space is also referred to as the index register. Another byte of the I/O space is referred to as the data register. The index register selects the extra registers, and the selected register is accessed through the data register. When accessing the data register, first write the index value to the index register, then write the data to the data register.

Fig. 2-2 shows the I/O map, and Table 2-1 indicates the relationship for index values and the data registers.

Caution: Only the lower 10 bits of the internal I/O address are decoded while the upper 6 bits are not decoded. Therefore, on the software, the same internal I/O operates on two or more addresses.

00A2H-FFFFH	External I/O
00A1H	Interrupt request mask register
00A0H	*
0084H-009FH	External I/O
0081H-0083H	DMA page register
0070H-0080H	External I/O
006FH 006EH	EMS data register
006DH	External I/O
006CH	EMS address register
006BH	External I/O
0066H-006AH	Keyboard interface
0065H 0064H	External I/O
0063H	ICW2 retention register
0060H-0062H	Keyboard interface (PP1A, PP1B, PP1C)
0044H-005FH	External I/O
0040H-0043H	TCU
0028H-003FH	External I/O
0027H	Data register
0026H	Index register
0022H-0025H	External I/O
0021H 0020H	ICU
0010H-001FH	External I/O
0000H-000FH	DMAU

*: Read mode : Interrupt status register
 Write mode : NMI mask register

Fig. 2-2 I/O Map

Table 2-1 Relationship for index Values and Data Registers

Index value	Register name (Data register)
FFH 12H	Unused
11H	BIOS time base register
10H	Shadow RAM control register
0FH	Keyboard trap vector register
0EH	Extended ROM control register
0DH	PCS5 address register
0CH	PCS4 address register
0BH	PCS3 address register
0AH	PCS2 address register
09H	PCS1 address register
08H	PCS0 address register
07H	PCS4, PCS5 control register
06H	PCS0-PCS3 control register
05H	System switch register
04H	DMAU control register
03H	CPU wait state register
02H	BCG control register
01H	Memory control register
00H	Unused

3. INTERNAL PERIPHERAL UNIT FUNCTIONS

3.1 CPU

The CPU is functionally equivalent to the μPD70116H (V30HL). Although its hardware functions are partly changed due to restrictions of use of the buses by other internal peripheral units, all software functions are compatible with that written for the μPD70116H.

3.2 CG (CLOCK GENERATOR)

The μPD70280 has the following two different clocks:

- ① CPUCLK (clock for CPU operation)
- ② SYSCLK (clock for SD bus operation)

CPUCLK is generated by the CG, and the SYSCLK is generated by the BCG.

Fig. 3-1 shows the relationship for the CG and BCG.

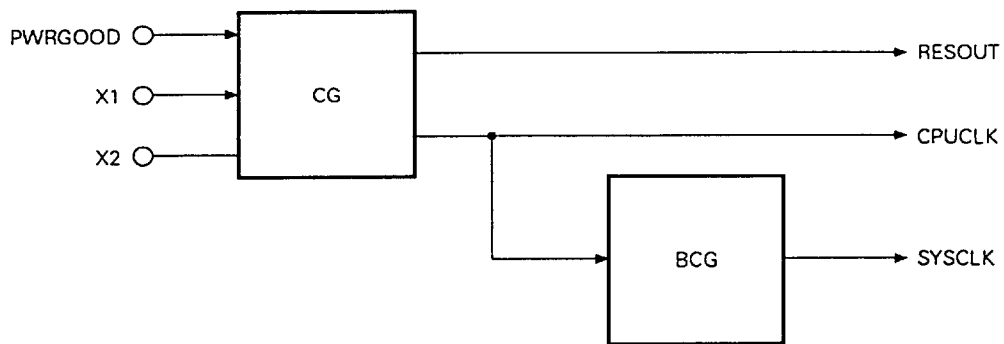


Fig. 3-1 Relationship for CG and BCG

The CG divides the clock input from the X1 and X2 pins by 2 to generate 50% duty CPUCLK. In addition, the CG synchronizes the signal input to the PWRGOOD pin and the CPUCLK to generate the reset signal (RESOUT)*. Fig. 3-2 shows an internal block diagram for the CG.

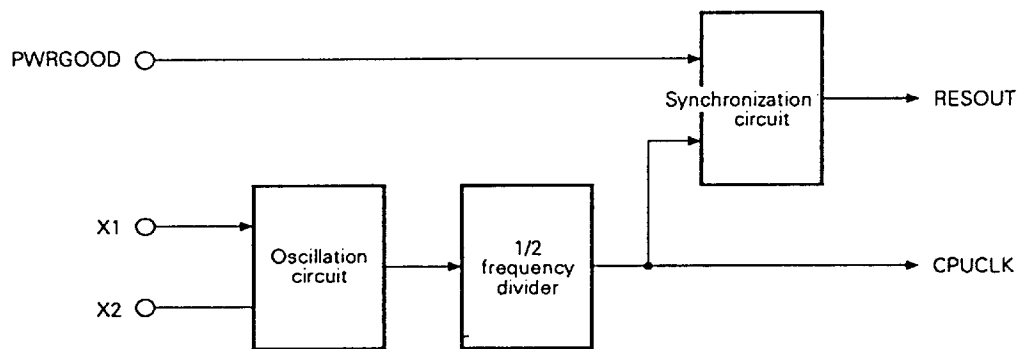


Fig. 3-2 CG Internal Block Diagram

The CG has the following functions:

- Clock oscillation
- CPUCLK generation
- Reset signal (RESOUT) generation

*: The CPU and internal peripheral units are reset when a low-level signal is input to the PWRGOOD pin and when this pin is kept at a low level for 4 clock cycles or longer after the falling edge of the low-level input signal.

3.3 BCG (BUS CYCLE GENERATOR)

The BCG generates SYSCLK, bus control signal, etc., for interfacing the CPU with the internal bus, and SD bus.

Fig. 3-3 shows an internal block diagram for the BCG.

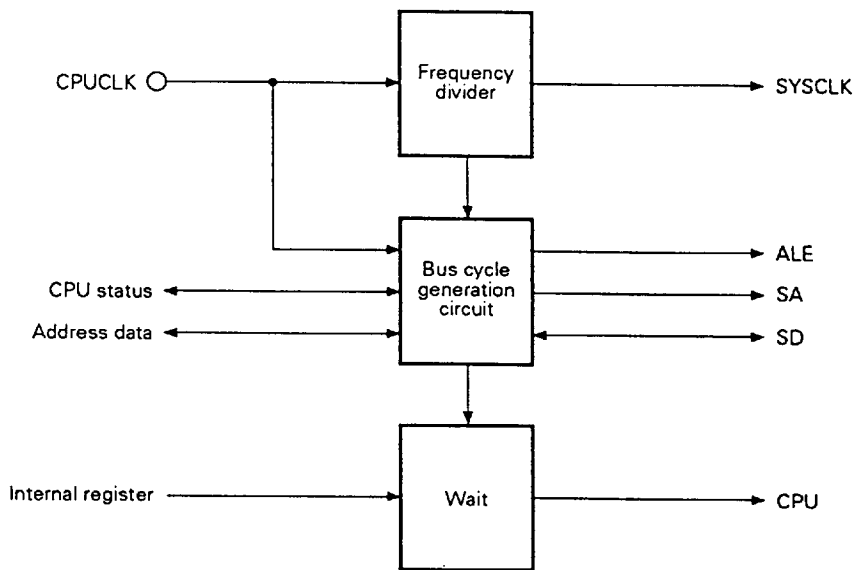


Fig. 3-3 BCG Internal Block Diagram

The BCG has the following functions:

- Generation and division of SYSCLK
- Bus control signal generation
- Wait control

3.4 BIU (BUS INTERFACE UNIT)

The BIU has the following functions:

- Address signal latching
- Address/data separation
- Data sizing control

3.5 SPU (SPEAKER INTERFACE UNIT)

The SPU is the PC/XT compatible speaker interface.

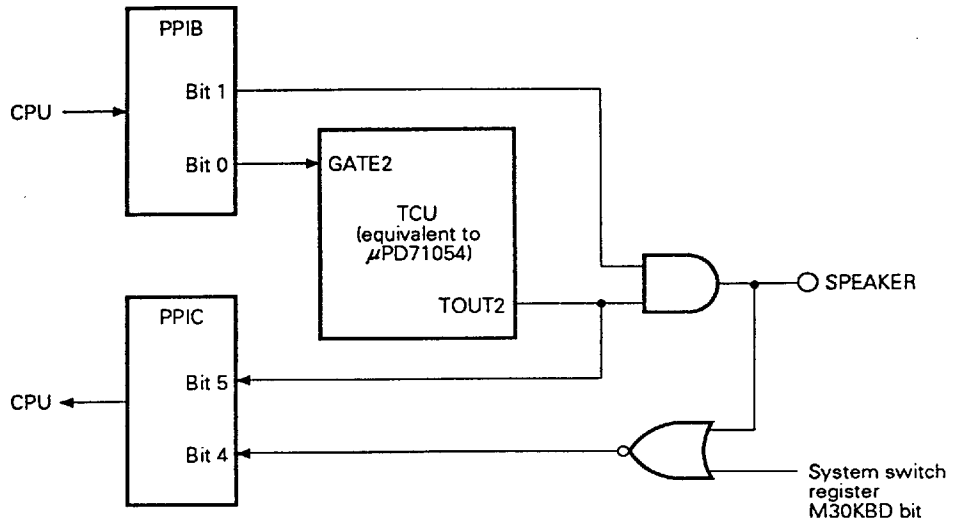


Fig. 3-4 Speaker Interface

3.6 TCU (TIMER/COUNTER UNIT)

The TCU is equivalent to the μPD71054, and has three counters (TCT#0-TCT#2). However, the purposes of these counters are limited, as follows:

- TCT#0 (TOUT0 output) : Can output periodic interrupt request to the CPU.
- TCT#1 (TOUT1 output) : Can output DMA transfer request from the DMAU for refreshing.
- TCT#2 (TOUT2 output) : Can be used as the source for the speaker.

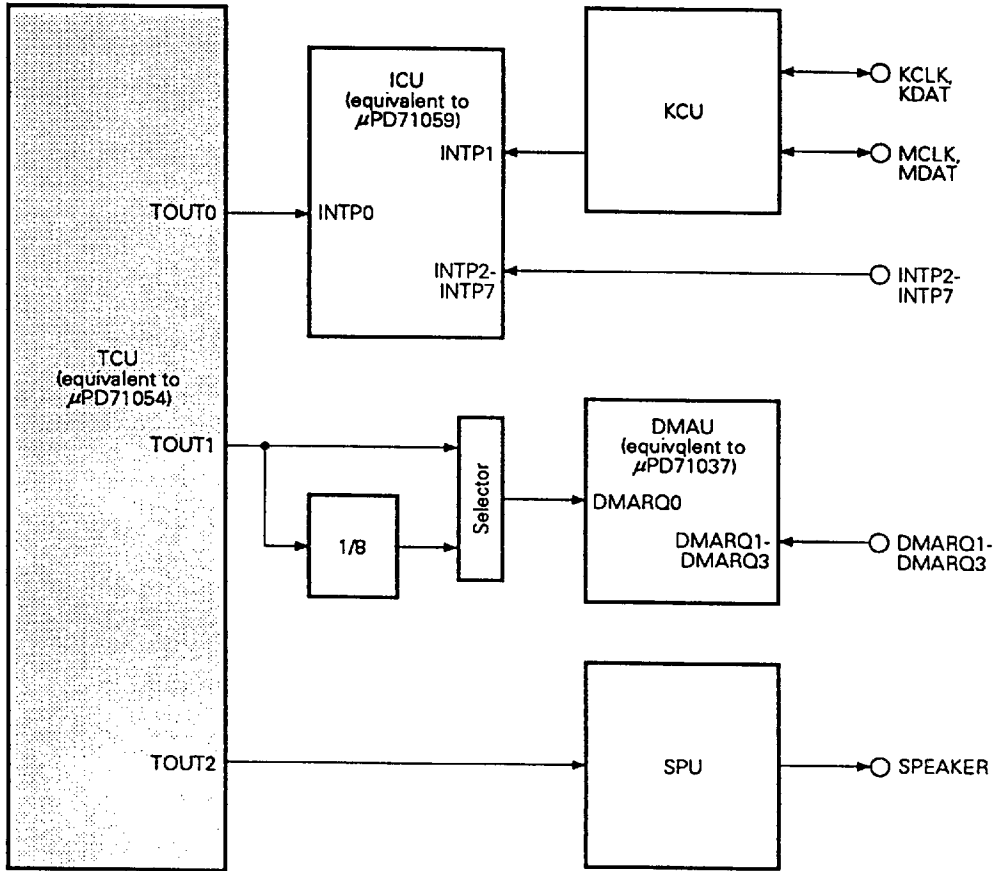


Fig. 3-5 Relationship for TCU and Internal Peripheral Units

3.7 ICU (INTERRUPT CONTROL UNIT)

The ICU arbitrates up to eight interrupt requests (maskable interrupts) generated inside and outside of the μPD70280, and transfers one of them to the CPU. The ICU function is identical to the μPD71059, except that functions, not necessary for the μPD70280, are not provided.

Caution: Cascade connection is not supported by the ICU.

The main features of ICU are as follows:

- Eight interrupt inputs
- Edge or level trigger request input
(However, the input from the internally connected TCU is fixed to the edge trigger)
- Interrupt request can be individually masked
- Programmable interrupt request priority setting
- Polling operation possible

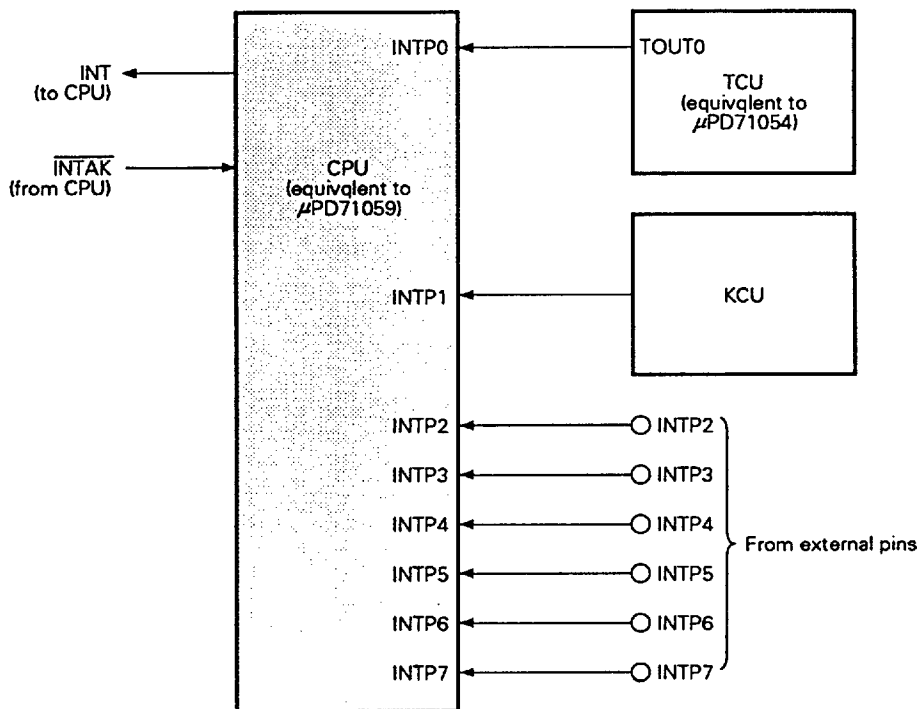


Fig. 3-6 Interrupt Request to ICU

3.8 NMI MASK CONTROL UNIT

There are two conditions that are to be input to the CPU to generate an NMI: IOCHCK and NMIN*. These two conditions are mask-ORed by the NMI mask control unit and input to the CPU.

In addition, mask control functions are also available.

- To allow the CPU to accurately recognize the rising edge of NMI, input a signal to the NMIIN pin in either of the following ways:
 - Input a low-level signal for a duration of three clock cycles or more (CPUCLK) and then input a high-level signal for a duration of 6 clock cycles or more.
 - Input a low-level signal for a duration of 2 clock cycles or more (CPUCLK) and then input a high-level signal for a duration of 5 clock cycles or more while remaining within the setup and hold times of the NMIIN pin.

3.9 DMAU (DMA CONTROL UNIT)

The DMAU has 4 DMA channels, and has the same functions as those of the μPD71037.

Of the four DMA channels for the DMAU, channel 0 is reserved for refreshing, and the refreshing request signal is periodically input from the TCU. In addition, for slow refreshing, DMARQ0 request pulses can be reduced down to 1/8 by the software.

The remaining three channels are released for the SD bus, and can be arbitrarily used.

- Caution 1 :** Channel 0 in the DMAU is the PC/XT architecture, and is reserved for DRAM refreshing. Therefore, even if no DRAM is used as the system RAM, memory-to-memory transfer is not supported.
- 2 :** DMAU cascade connection is not supported.
- 3 :** Channel 0 is used for refreshing. When using other channels, the bus must not be occupied for such a long time that channel 0 refreshing would be interfered with.

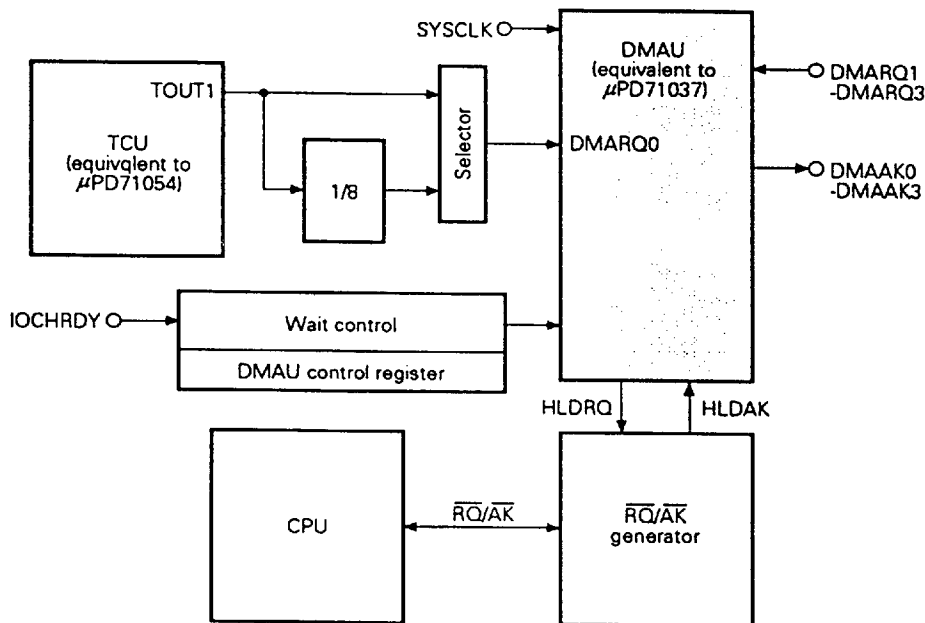


Fig. 3-7 Relationship for DMAU and CPU, and TCU

3.10 DMAPGR (DMA PAGE REGISTER)

The DMA page register sets the expansion DMA addresses (A16-A19) of channels 1 through 3.

During the DMA cycle, the μPD70280 generates 20-bit addresses as shown in Fig. 3-8.

As shown, the lower 16 bits of the address are supplied by the DMAU, while the higher 4 bits are supplied by the page register.

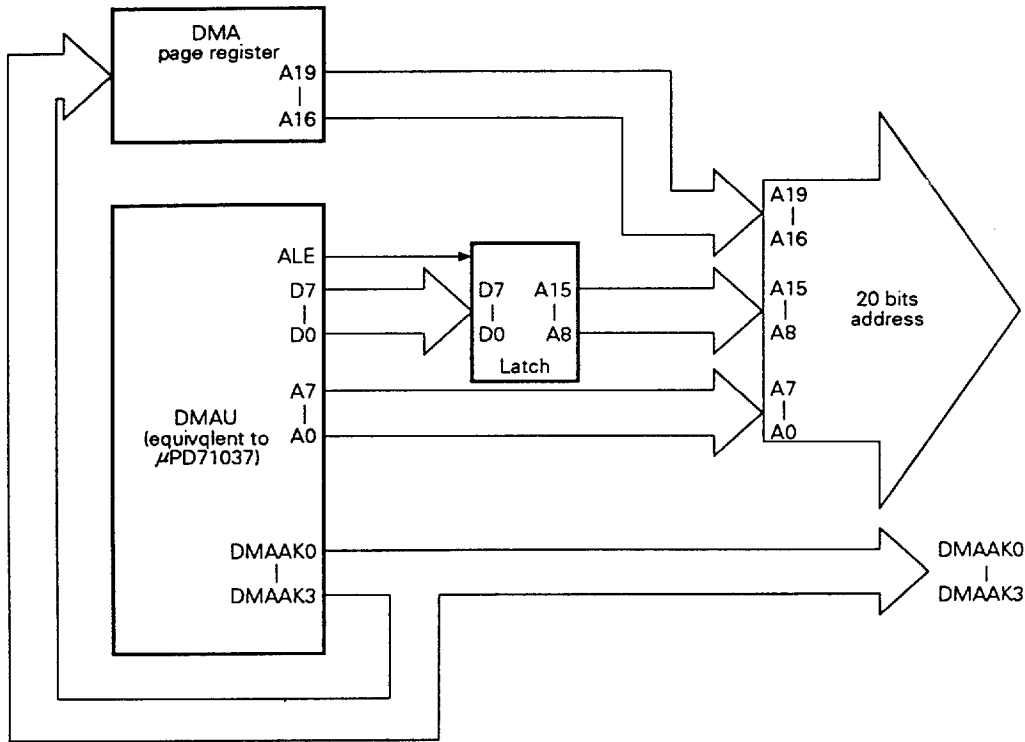


Fig. 3-8 Address Generation during DMA Cycle

3.11 KCU (KEYBOARD CONTROL UNIT)

The PC/XT or PS/2 model 30 key board can be connected to the KCU (set by the system switch register).

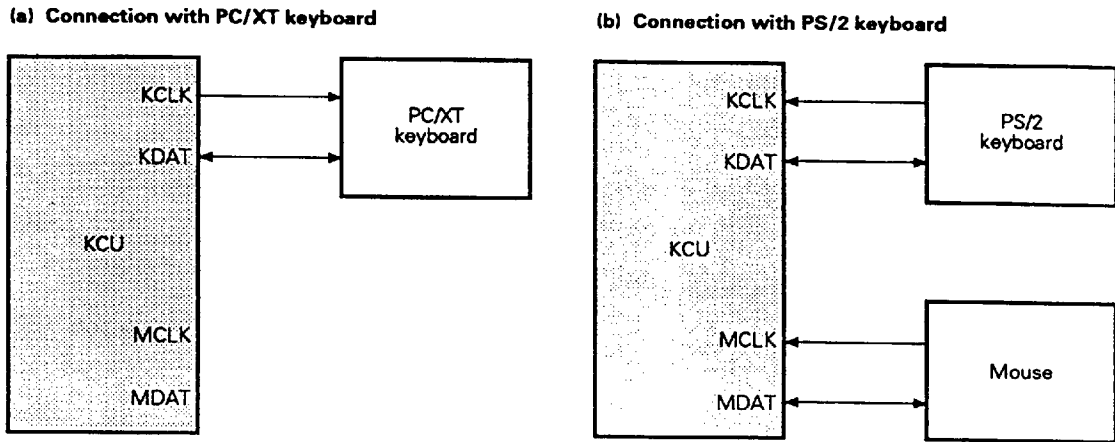


Fig. 3-9 KCU and Keyboard Connection

3.12 ROM DECODER

The ROM decoder generates the following two chip selection signals when connecting ROM:

- $\overline{\text{ROMCE0}}$
This is the chip selection signal for BIOS ROM. It is activated whenever an address from F0000H to FFFFFH is accessed.
- $\overline{\text{ROMCE1}}$
This is for expansion ROM. It may be used to program the start address and capacity of the ROM.

3.13 EXTERNAL I/O DECODER

The external I/O decoder generates six programmable chip selection signals: PCS0 through PCS5.

3.14 MCU (MEMORY CONTROL UNIT)

The μPD70280 can be easily interfaced with the system memory by the MCU. DRAM, SRAM, or pseudo-SRAM can be selected as the system memory. The memory category and the size can be selected by the software. The functions for the memory interface pins (SMC0-SMC15) differ, depending on the memory selected.

Table 3-1 indicates possible memory configurations and their relationship with maximum size.

Table 3-1 Possible Memory Configurations and Maximum Sizes

Memory	Configuration (word × bits)	Maximum size (bytes)
DRAM	256K × 1 / 256K × 4	2M
	1M × 1 / 1M × 4	6M
	4M × 1 / 4M × 4	16M
SRAM	32K × 8	640K
	128K × 8	2.5M
Pseudo-SRAM	32K × 8	640K
	128K × 8	2.25M

Note that the parity generation and check functions, which are provided in the PC/XT, are not supported in the μPD70280.

3.15 EMS CIRCUIT

The μPD70280 can support up to 16 Mbytes of EMS memory (with a DRAM of 4M words × 4 bits).

The EMS memory can use up to 12 EMS page frames.

Of the 12 page frames, eight are mapped (fixed) to within the 128 Kbytes (addresses 80000H-9FFFFH) from the highest address of the 640 Kbyte system memory.

The remaining four page frames may be mapped to any of three 64 Kbyte areas of C0000H-CFFFFH, D0000H-DFFFFH, and E0000H-EFFFFH (set by the memory control register).

The EMS expansion address is set by the EMS address register and EMS data register. The EMS address register specifies a page frame, and the expansion address (higher 10 bits: A14-A23) is written to the EMS data register.

Fig. 3-10 shows the relations between the EMS address register and EMS data register, and Fig. 3-11 shows an example of memory space allocation.

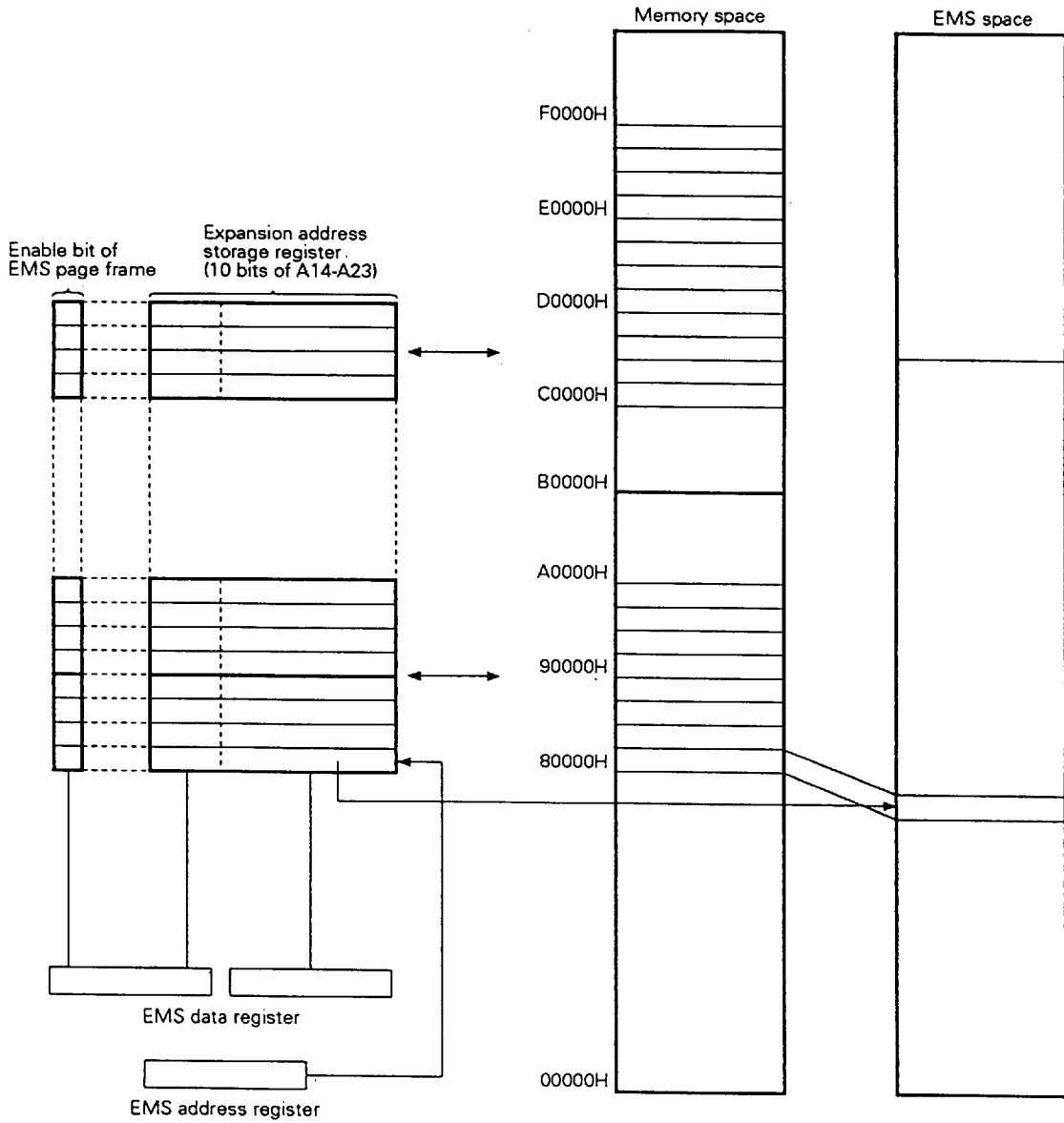


Fig. 3-10 Relations between EMS Address Register and EMS Data Register

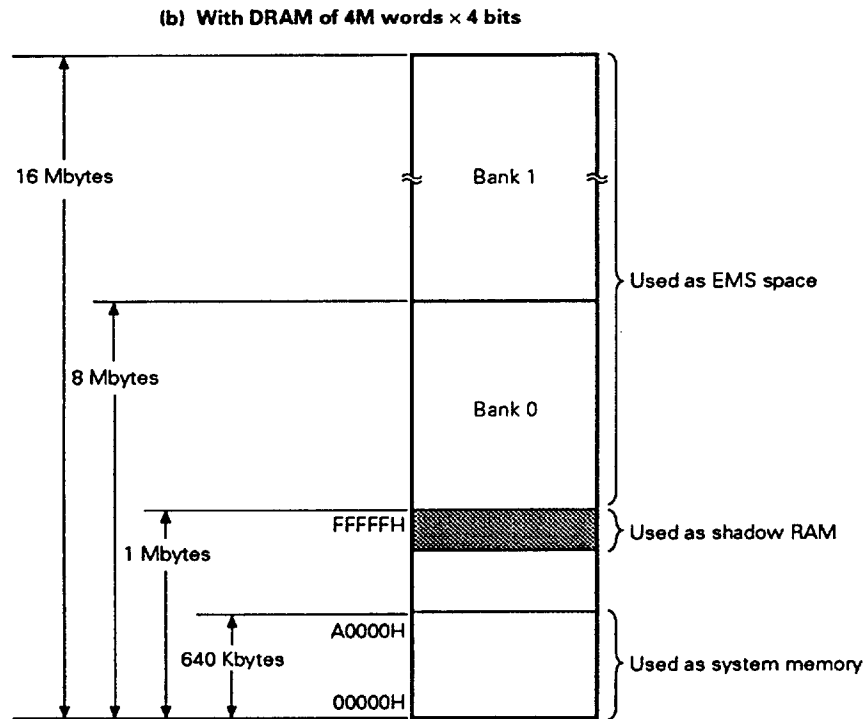
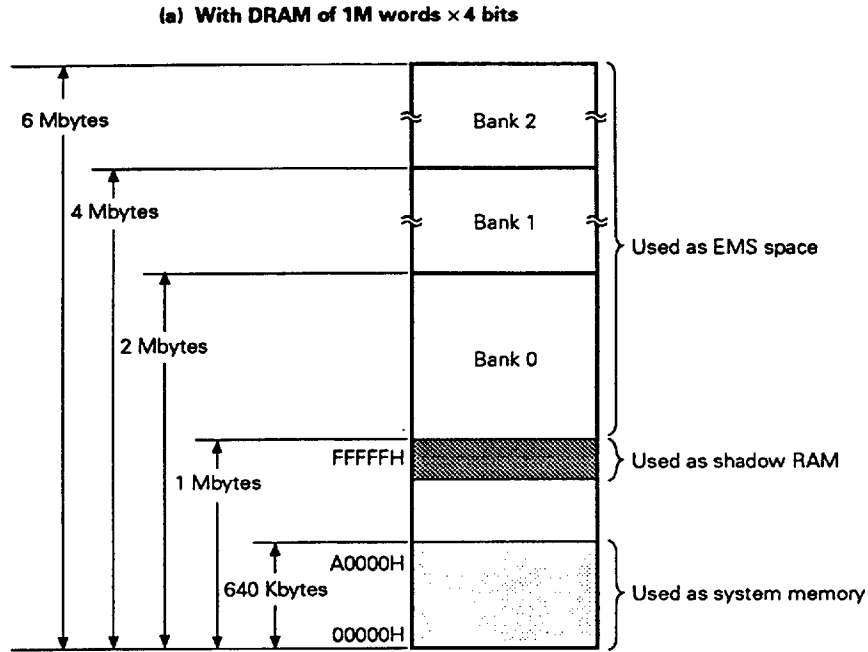


Fig. 3-11 Example of Memory Space Allocation

4. INSTRUCTION SET

Table 4-1 Operand Types

Symbol	Meaning
reg	8/16-bit general-purpose register (Destination register, when instruction uses two 8/16-bit general-purpose registers)
reg'	Source register, when instruction uses two 8/16-bit general-purpose registers
reg8	8-bit general-purpose register (Destination register, when instruction uses two 8-bit registers)
reg8'	Source register, when instruction uses two 8-bit general-purpose registers
reg16	16-bit general-purpose register (Destination register, when instruction uses two 16-bit general-purpose registers)
reg16'	Source register, when instruction uses two 16-bit general-purpose registers
dmem	8/16-bit memory location
mem	8/16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
imm	Constant (0 to FFFFH)
imm3	Constant (0 to 7)
imm4	Constant (0 to FH)
imm8	Constant (0 to FFH)
imm16	Constant (0 to FFFFH)
acc	Register AW or AL
sreg	Segment register
src-table	256-byte translation table
src-block	Block name addressed by register IX
dst-block	Block name addressed by register IY
near-proc	Procedure within the current segment
far-proc	Procedure within a different program segment
near-label	Label within the current segment
short-label	Label between -128 and +127 bytes from the end of the current instruction
far-label	Label within a different program segment
memptr16	Word containing the offset of the memory location within the current program segment to which control is to be transferred.
memptr32	Double word containing the offset and segment base address of the memory location in another segment to which control is to be transferred.
regptr16	16-bit general-purpose register containing the offset of the memory location in another segment to which control is to be transferred.
pop-value	Number of bytes to discard from the stack (0-64K, normally an even number)
fp-op	Immediate data to identify the OP code for the external floating-point operation
R	Register set

Table 4-2 Operation Codes

Symbol	Meaning
W	Word/byte field (0-1)
reg	Register field (000-111)
reg'	Register field (000-111) (Source register when two registers are used)
mem	Memory field (000-111)
mod	Mode field (00-10)
S:W	When S:W=01, data=16 bits; otherwise, data=8 bits When S:W=11, the byte data sign is expanded to 16-bit operand
X,XXX,YYY,ZZZ	Data to identify the OP code for the external floating-point operation

Table 4-3 Symbols Used for Operation

Identifier	Meaning	Identifier	Meaning
AW	Accumulator (16 bits)	V	Overflow flag
AH	Accumulator (high byte)	BRK	Break flag
AL	Accumulator (low byte)	MD	Mode flag
BW	Register BW (16 bits)	(...)	Memory contents indicated by parentheses
CW	Register CW (16 bits)	disp	Displacement (8/16 bits)
CL	Register CW (low byte)	ext-disp8	8-bit displacement sign expanded to 16-bit
DW	Register DW (16 bits)	temp	Temporary register (8/16/32 bits)
BP	Base pointer (16 bits)	TA	Temporary register A (16 bits)
SP	Stack pointer (16 bits)	TB	Temporary register B (16 bits)
PC	Program counter (16 bits)	TC	Temporary register C (16 bits)
PSW	Program status word (16 bits)	tmpcy	Temporary carry flag (1 bit)
IX	Index register (source) (16 bits)	seg	Immediate segment data (16 bits)
IY	Index register (destination) (16 bits)	offset	Immediate offset data (16 bits)
PS	Program segment register (16 bits)	←	Transfer direction
SS	Stack segment register (16 bits)	+	Addition
DS0	Data segment 0 register (16 bits)	-	Subtraction
DS1	Data segment 1 register (16 bits)	x	Multiplication
AC	Auxiliary carry flag	÷	Division
CY	Carry flag	%	Modulo
P	Parity flag	∧	AND
S	Sign flag	∨	OR
Z	Zero flag	⊕	XOR
DIR	Direction flag	xxH	Two-digit hexadecimal value
IE	Interrupt enable flag	xxxxH	Four-digit hexadecimal value

Table 4-4 Flag Operation

Identifier	Meaning
(blank)	No change
0	Cleared to 0
1	Set to 1
x	Set or cleared, according to result
U	Undefined
R	Restored to previous state

Table 4-5 Memory Addressing Mode

mod men	00	01	10
000	BW+IX	BW+IX+disp 8	BW+IX+disp 16
001	BW+IY	BW+IY+disp 8	BW+IY+disp 16
010	BP+IX	BP+IX+disp 8	BP+IX+disp 16
011	BP+IY	BP+IY+disp 8	BP+IY+disp 16
100	IX	IX+disp 8	IX+disp 16
101	IY	IY+disp 8	IY+disp 16
110	DIRECT ADDRESS	BP+disp 8	BP+disp 16
111	BW	BW+disp 8	BW+disp 16

Table 4-6
8/16-Bit General-Purpose Register Selection

reg, reg'	W=0	W=1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

Table 4-7
Segment Register Selection

sreg	
00	DS1
01	PS
10	SS
11	DS0

The instruction set is shown on the following pages in table form.

In the clocks column, for an instruction which can accomplish operation in both byte and word units (with W bit), the left side of the slash (/) indicates the value for byte processing or word processing at an even address. The right side indicates the value for word processing for an odd address.

However, for block transfer related instructions, refer to Table 4-8.

The number of clocks includes the time period for the following processing:

- Decoding
- EA generation
- Operand fetching
- Execution

Assuming that the instruction byte has already been prefetched.

Table 4-8 Number of Clocks for Block Transfer Related Instructions (1/2)

Instruction	Number of clocks			
	Byte processing (W=0)	Word processing (W=1)		
		Odd, odd address	Odd, even address	Even, even address
MOVBK	11+8/rep (11)	11+16/rep (19)	11+12/rep (15)	11+8/rep (11)
CMPBK	7+14/rep (13)	7+22/rep (21)	7+18/rep (17)	7+14/rep (13)

Remarks : Value indicated by parentheses applies only for one processing.

Table 4-8 Number of Clocks for Block Transfer Related Instructions (2/2)

Instruction	Number of clocks		
	Byte processing (W=0)	Word processing (W=1)	
		Odd address	Even address
CMPM	7+10/rep (7)	7+14/rep (11)	7+10/rep (7)
LDM	7+9/rep (7)	7+13/rep (11)	7+9/rep (7)
STM	7+4/rep (7)	7+8/rep (11)	7+4/rep (7)
INM	9+8/rep (10)	9+16/rep (18)	9+8/rep (10)
OUTM	9+8/rep (10)	9+16/rep (18)	9+8/rep (10)

Remarks : Value indicated by parentheses applies only for one processing.

Instruction group	Mnemonic	Operand	Operation code																Number of bytes	Number of clocks	Operation	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				AC	CY	V	P	S	Z
Data transfer instruction	MOV	reg, reg'	1	0	0	0	1	0	1	W	1	1	reg	reg'	2	2	reg→reg'										
		mem, reg	1	0	0	0	1	0	0	W	mod	reg	mem	2-4	9/13	(mem)←reg											
		reg, mem	1	0	0	0	1	0	1	W	mod	reg	mem	2-4	11/15	reg←(mem)											
		mem, imm	1	1	0	0	0	1	1	W	mod	0	0	0	mem	3-6	11/15	(mem)←imm									
		reg, imm	1	0	1	1	W	reg						2-3	4	reg←imm											
		acc, dmem	1	0	1	0	0	0	0	W					3	10/14	When W=0, AL←(dmem)										
		dmem, acc	1	0	1	0	0	0	1	W					3	9/13	When W=1, AH←(dmem+1), AL←(dmem)										
		sreg, reg16	1	0	0	0	1	1	0	1	0	1	1	0	sreg	reg	2	2	sreg←reg16								
		sreg, mem16	1	0	0	0	1	1	0	mod	0	sreg	mem	2-4	11/15	sreg←(mem16)											
		reg16, sreg	1	0	0	0	1	1	0	0	1	1	0	sreg	reg	2	2	reg16←sreg									
		mem16, sreg	1	0	0	0	1	1	0	0	mod	0	sreg	mem	2-4	10/14	(mem16)←sreg										
		DS0, reg16, mem32	1	1	0	0	0	1	0	1	mod	reg	mem	2-4	18/26	reg16←(mem32), DS0←(mem32+2)											
		DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod	reg	mem	2-4	18/26	reg16←(mem32), DS1←(mem32+2)											
AH, PSW	1	0	0	1	1	1	1	1					1	2	AH←S, Z, x, AC, x, P, x, CY												
PSW, AH	1	0	0	1	1	1	0					1	3	S, Z, x, AC, x, P, x, CY←AH													
LDEA	reg16, mem16	1	0	0	0	1	1	0	1	mod	reg	mem	2-4	4	reg16←mem16												

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags											
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S	Z
INS		reg8,reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	3	31-117	16 bit field←AW						
		1 1 reg' reg	1	1	0	0	1	1	1	0	0	1	1	0	0	0	1	/35-133									
EXT		reg8,imm4	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	4	31-117	16-bit field←AW							
		1 1 0 0 0 reg	1	1	0	0	0	0	1	1	1	0	0	1	1	/35-133											
Bit field operation	EXT	reg8,reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	3	26-55	AW←16 bit field							
		1 1 reg' reg	1	1	0	0	1	1	1	0	0	1	1	/34-59													
Bit field operation		reg8,imm4	0	0	0	0	1	1	1	1	0	0	1	1	0	1	1	4	26-55	AW←16 bit field							
		1 1 0 0 0 reg	1	1	0	0	0	0	1	1	1	/34-59															
I/O instruction	IN	acc,imm8	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	2	9/13	When W=0, AL←(imm8) When W=1, AH←(imm8+1), AL←(imm8)							
		acc,DW	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	8/12	When W=0, AL←(DW) When W=1, AH←(DW+1), AL←(DW)							
I/O instruction	OUT	imm8,acc	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	2	8/12	When W=0, (imm8)←AL When W=1, (imm8+1)←AH,(imm8)←AL							
		DW,acc	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1	8/12	When W=0, (DW)←AL When W=1, (DW+1)←AH,(DW)←AL							
Primitive I/O instruction	INM	dst-block, DW	0	1	1	0	1	1	1	0	1	0	1	0	1	0	1	1	See Table 4-8	When W=0, (Y)←(DW) DIR=0: Y←Y+1; DIR = 1: Y←Y-1 When W=1, (Y+1, Y)←(DW+1, DW) DIR=0: Y←Y+2; DIR=1: Y←Y-2							
		DW, src-block	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	See Table 4-8	When W=0, (DW)←(IX) DIR=0: IX←IX+1; DIR=1: IX←IX-1 When W=1, (DW+1,DW)←(IX+1,IX) DIR=0: IX←IX+2; DIR=1: IX←IX-2							

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags									
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P
Addition/Subtraction instruction	ADD	reg,reg'	0	0	0	0	0	1	W	1	1	reg	reg'	2	2	reg←reg+reg'	X	X	X	X	X	X			
		mem,reg	0	0	0	0	0	0	W	mod	reg	mem	mem	2-4	16/24	(mem)←(mem)+reg	X	X	X	X	X	X			
		reg,mem	0	0	0	0	0	1	W	mod	reg	mem	mem	2-4	11/15	reg←reg+(mem)	X	X	X	X	X	X			
		reg,imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg	3-4	4	reg←reg+imm	X	X	X	X	X	X
		mem,imm	1	0	0	0	0	0	S	W	mod	0	0	0	0	mem	3-6	18/26	(mem)←(mem)+imm	X	X	X	X	X	X
		acc,imm	0	0	0	0	0	1	0	W					2-3	4	When W=0, AL←AL+imm When W=1, AW←AW+imm	X	X	X	X	X	X		
	ADDC	reg,reg'	0	0	0	1	0	0	1	W	1	1	reg	reg'	2	2	reg←reg+reg'+CY	X	X	X	X	X	X		
		mem,reg	0	0	0	1	0	0	0	W	mod	reg	mem	mem	2-4	16/24	(mem)←(mem)+reg+CY	X	X	X	X	X	X		
		reg,mem	0	0	0	1	0	0	1	W	mod	reg	mem	mem	2-4	11/15	reg←reg+(mem)+CY	X	X	X	X	X	X		
		reg,imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg	3-4	4	reg←reg+imm+CY	X	X	X	X	X	X
		mem,imm	1	0	0	0	0	0	S	W	mod	0	1	0	0	mem	3-6	18/26	(mem)←(mem)+imm+CY	X	X	X	X	X	X
		acc,imm	0	0	0	1	0	1	0	W					2-3	4	When W=0, AL←AL+imm+CY When W=1, AW←AW+imm+CY	X	X	X	X	X	X		
SUB	reg,reg'	0	0	1	0	1	0	1	W	1	1	reg	reg'	2	2	reg←reg-reg'	X	X	X	X	X	X			
	mem,reg	0	0	1	0	1	0	0	W	mod	reg	mem	mem	2-4	16/24	(mem)←(mem)-reg	X	X	X	X	X	X			
	reg,mem	0	0	1	0	1	0	1	W	mod	reg	mem	mem	2-4	11/15	reg←reg-(mem)	X	X	X	X	X	X			
	reg,imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg	3-4	4	reg←reg-imm	X	X	X	X	X	X	
	mem,imm	1	0	0	0	0	0	S	W	mod	1	0	1	0	mem	3-6	18/26	(mem)←(mem)-imm	X	X	X	X	X	X	
	acc,imm	0	0	1	0	1	0	1	W					2-3	4	When W=0, AL←AL+imm+CY When W=1, AW←AW+imm+CY	X	X	X	X	X	X			
SUBC	reg,reg'	0	0	0	1	0	1	0	W	1	1	reg	reg'	2	2	reg←reg-reg'-CY	X	X	X	X	X	X			
	mem,reg	0	0	0	1	0	0	0	W	mod	reg	mem	mem	2-4	16/24	(mem)←(mem)-reg-CY	X	X	X	X	X	X			
	reg,mem	0	0	0	1	0	1	0	W	mod	reg	mem	mem	2-4	11/15	reg←reg-(mem)-CY	X	X	X	X	X	X			
	reg,imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg	3-4	4	reg←reg-imm-CY	X	X	X	X	X	X	
	mem,imm	1	0	0	0	0	0	S	W	mod	0	1	1	0	mem	3-6	18/26	(mem)←(mem)-imm-CY	X	X	X	X	X	X	
	acc,imm	0	0	0	1	1	0	1	W					2-3	4	When AL←AL-imm-CY When W=1, AW←AW-imm-CY	X	X	X	X	X	X			

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags																		
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S	Z							
BCD operation instruction	ADD4S		0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	2	19xn+7	dst BCD string←dst BCD string + src BCD string*	U	x	U	U	U	x						
	SUB4S		0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	x	2	19xn+7	dst BCD string←dst BCD string - src BCD string*	U	x	U	U	U	x						
	CMP4S		0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	x	2	19xn+7	dst BCD string - src BCD string*	U	x	U	U	U	x						
BCD operation instruction	ROL4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	3	13														
		mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	3-5	28														
			mod 0 0 0 mem	1	1	0	0	0	reg																									
BCD operation instruction	ROR4	reg8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	3	17														
		mem8	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0	3-5	32														
			mod 0 0 0 mem	1	1	0	0	0	reg																									
Increment/decrement instruction	INC	reg8	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	reg	2	2	reg8←reg8+1	x		x	x	x								
		mem	1	1	1	1	1	1	W	mod 0 0 0 mem																								
		reg16	0	1	0	0	0	reg																										
Increment/decrement instruction	DEC	reg8	1	1	1	1	1	1	0	1	1	0	0	1	0	0	1	reg	2	2	reg8←reg8-1	x		x	x	x								
		mem	1	1	1	1	1	1	W	mod 0 0 1 mem																								
		reg16	0	1	0	0	1	reg																										

n : Half of the number of digits of BCD

* : The number of digits in BCD is specified using the CL register. Its value can range from 1 to 254.

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags								
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V
Multiplication instruction	MUL	reg16, (reg16,)* imm8	0	1	1	0	1	1	1	1	1	1	0	reg	reg'	3	28-34	reg16←reg16'ximm8 Product≤16 bits: CY←0,V←0 Product>16 bits: CY←1,V←1	U	x	x	U	U	U
		reg16, mem16, imm8	0	1	1	0	1	0	1	1	1	0	mod	reg mem	3-5	34-40 /38-44	reg16←(mem16)ximm8 Product≤16 bits: CY←0,V←0 Product>16 bits: CY←1,V←1	U	x	x	U	U	U	
		reg16, (reg16,)* imm16	0	1	1	0	1	0	0	1	1	1	0	reg	reg'	4	36-42	reg16←reg16'ximm16 Product≤16 bits: CY←0,V←0 Product>16 bits: CY←1,V←1	U	x	x	U	U	U
		reg16, mem16, imm16	0	1	1	0	1	0	0	1	1	1	0	mod	reg mem	4-6	42-48 /46-52	reg16←(mem16)ximm16 Product≤16 bits: CY←0,V←0 Product>16 bits: CY←1,V←1	U	x	x	U	U	U

* : The second operand can be omitted. When omitted, the same register, specified for the first operand, is assumed to be specified.

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags										
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	reg	AC	CY	V	P
Unsigned division instruction	DIVU	reg8	1	1	1	1	0	1	1	0	1	1	0	1	1	0	reg	2	19	temp←AW When temp+reg8<FFH, AH←temp%reg8, AL←temp+reg8 When temp+reg8>FFH, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U
			1	1	1	1	0	1	1	0	1	1	0	mod	2-4	25	temp←AW When temp+(mem8)<FFH, AH←temp%(mem8), AL←temp+(mem8) When temp+(mem8)>FFH, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U			
		1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	reg	2	25	temp←DW,AW When temp+reg16<FFFFH, DW←temp%reg16, AW←temp+reg16 When temp+reg16>FFFFH, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U
		mem16	1	1	1	1	0	1	1	1	1	1	1	0	mem	2-4	30/34	temp←DW, AW When temp+(mem16)<FFFFH, DW←temp%(mem16), AW←temp+(mem16) When temp+(mem16)>FFFFH, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U		

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags													
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S	Z		
Signed division instruction	DIV	reg8	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	0	2	29-34	temp←AW When temp+reg8>0 and temp+reg8≤7FH or when temp+reg8<0 and temp+reg8>0-7FH-1, AH←temp%reg8, AL←temp+reg8 When temp+reg8>0 and temp+reg8>7FH or when temp+reg8<0 and temp+reg8≤0-7FH-1, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U
		mem8	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	2-4	34-39	temp←AW When temp+(mem8)>0 and temp+(mem8)≤7FH or when temp+(mem8)<0 and temp+(mem8)>0-7FH-1, AH←temp%(mem8), AL←temp+(mem8) When temp+(mem8)>0 and temp+(mem8)>7FH or when temp+(mem8)<0 and temp+(mem8)≤0-7FH-1, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U	
		reg16	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	2	38-43	temp←DW,AW When temp+reg16>0 and temp+reg16≤7FFFH or when temp+reg16<0 and temp+reg16>0-7FFFH-1, DW←temp%reg16,AW←temp+reg16 When temp+reg16>0 and temp+reg16>7FFFH or when temp+reg16<0 and temp+reg16≤0-7FFFH-1, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U
		mem16	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	2-4	43-48 /47-52	temp←DW,AW When temp+(mem16)>0 and temp+(mem16)≤7FFFH or when temp+(mem16)<0 and temp+(mem16)>0-7FFFH-1, DW←temp%(mem16),AW←temp+(mem16) When temp+(mem16)>0 and temp+(mem16)>7FFFH or when temp+(mem16)<0 and temp+(mem16)≤0-7FFFH-1, TA←(001H,000H), TC←(003H,002H) SP←SP-2,(SP+1,SP)←PSW,IE←0,BRK←0 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA	U	U	U	U	U	U	

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags										
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S
BCD coprocessor instruction	ADJBA		0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	7	When AL∧0FH>9 or AC=1, AL←AL+6 AH←AH+1, AC←1, CY←AC, AL←AL∧0FH	x	x	U	U	U	U	U
	ADJAA		0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	3	When AL∧0FH>9 or AC=1, AL←AL+6, CY←CY∨AC, AC←1 When AL>9FH or CY=1, AL←AL+60H, CY←1	x	x	U	x	x	x	x
	ADJBS		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	7	When AL∧0FH>9 or AC=1, AL←AL-6, AH←AH-1, AC←1 CY←AC, AL←AL∧0FH	x	x	U	U	U	U	U
	ADJAS		0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	3	When AL∧0FH>9 or AC=1, AL←AL-6, CY←CY∨AC, AC←1 When AL>9FH or CY=1, AL←AL-60H, CY←1	x	x	U	x	x	x	x
Data conversion	CVTBD		1	1	0	1	0	1	0	0	0	0	1	0	1	0	2	15	AH←AL+0AH, AL←AL%0AH	U	U	U	x	x	x	x
	CVTDB		1	1	0	1	0	1	0	1	0	0	0	1	0	1	2	7	AL←AHx0AH+AL, AH←0	U	U	U	x	x	x	x
	CVTBW		1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	2	When AL<80H, AH←0. Otherwise, AH←FFH							
	CVTWL		1	0	0	1	1	0	0	1	0	0	0	0	0	0	1	4-5	When AW<8000H, DW←0. Otherwise, DW←FFFFH							
Compare instruction	CMP	reg,reg'	0	0	1	1	1	0	1	W	1	1	reg	reg'	2	2	reg-reg	x	x	x	x	x	x	x		
		mem,reg	0	0	1	1	1	0	0	W	mod	reg	mem	2-4	11/15	(mem)-reg	x	x	x	x	x	x	x			
		reg,mem	0	0	1	1	1	0	1	W	mod	reg	mem	2-4	11/15	reg-(mem)	x	x	x	x	x	x	x			
		reg,imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg	3-4	reg-imm	x	x	x	x	x	x	x	
		mem,imm	1	0	0	0	0	0	S	W	mod	1	1	1	1	mem	3-6	(mem)-imm	x	x	x	x	x	x	x	
	acc,imm	0	0	1	1	1	1	0	W			2-3	4	When W=0, AL-imm When W=1, AW-imm	x	x	x	x	x	x	x					

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags									
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P
Complements	NOT	reg	1	1	1	0	1	1	W	1	1	0	1	0	reg	2	reg ← reg								
		mem	1	1	1	0	1	1	W	mod	0	1	0	mem	2-4	(mem) ← (mem)									
	NEG	reg	1	1	1	0	1	1	W	1	1	0	1	1	reg	2	reg ← reg + 1	x	x	x	x	x	x		
		mem	1	1	1	0	1	1	W	mod	0	1	1	mem	2-4	(mem) ← (mem) + 1	x	x	x	x	x	x			
Logical operation instruction	TEST	reg, reg'	1	0	0	0	1	0	W	1	1	reg'	reg	2	reg ∧ reg'	U	0	0	x	x	x				
		mem, reg	1	0	0	0	1	0	W	mod	reg	mem	2-4	(mem) ∧ reg	U	0	0	x	x	x					
		reg, mem	1	0	0	0	1	0	W	mod	reg	mem	2-4	(mem) ∧ reg	U	0	0	x	x	x					
	AND	reg, imm	1	1	1	0	1	1	W	1	1	0	0	reg	3-4	reg ∧ imm	U	0	0	x	x	x			
		mem, imm	1	1	1	0	1	1	W	mod	0	0	0	mem	3-6	(mem) ∧ imm	U	0	0	x	x	x			
		acc, imm	1	0	1	0	1	0	0	W				2-3	When W=0, AL ∧ imm8 When W=1, AW ∧ imm16	U	0	0	x	x	x				
Logical operation instruction	AND	reg, reg'	0	0	1	0	0	1	W	1	1	reg	reg'	2	reg ← reg ∧ reg'	U	0	0	x	x	x				
		mem, reg	0	0	1	0	0	0	W	mod	reg	mem	2-4	(mem) ← (mem) ∧ reg	U	0	0	x	x	x					
		reg, mem	0	0	1	0	0	0	1	W	mod	reg	mem	2-4	reg ← reg ∧ (mem)	U	0	0	x	x	x				
		reg, imm	1	0	0	0	0	0	W	1	1	1	0	0	reg	3-4	reg ← reg ∧ imm	U	0	0	x	x	x		
		mem, imm	1	0	0	0	0	0	W	mod	1	0	0	mem	3-6	(mem) ← (mem) ∧ imm	U	0	0	x	x	x			
	OR	acc, imm	0	0	1	0	0	1	0	W				2-3	When W=0, AL ← AL ∧ imm8 When W=1, AW ← AW ∧ imm16	U	0	0	x	x	x				
		reg, reg'	0	0	0	0	1	0	1	W	1	1	reg	reg'	2	reg ← reg ∨ reg'	U	0	0	x	x	x			
		mem, reg	0	0	0	0	1	0	0	W	mod	reg	mem	2-4	(mem) ← (mem) ∨ reg	U	0	0	x	x	x				
		reg, mem	0	0	0	0	1	0	1	W	mod	reg	mem	2-4	reg ← reg ∨ (mem)	U	0	0	x	x	x				
		reg, imm	1	0	0	0	0	0	W	1	1	0	0	1	reg	3-4	reg ← reg ∨ imm	U	0	0	x	x	x		
Logical operation instruction	OR	mem, imm	1	0	0	0	0	0	W	mod	0	0	1	mem	3-6	(mem) ← (mem) ∨ imm	U	0	0	x	x	x			
		acc, imm	0	0	0	0	1	1	0	W				2-3	When W=0, AL ← AL ∨ imm8 When W=1, AW ← AW ∨ imm16	U	0	0	x	x	x				

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags										
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S
Logical operation instruction	XOR	reg,reg'	0	0	1	1	0	0	1	W	1	1	1	0	0	0	0	0	2	reg←reg ∨ reg'	U	0	0	x	x	x
		mem,reg	0	0	1	1	0	0	0	W	mod	reg	mem	2-4	(mem)←(mem) ∨ reg	U	0	0	x	x	x					
		reg,mem	0	0	1	1	0	0	1	W	mod	reg	mem	2-4	reg←reg ∨ (mem)	U	0	0	x	x	x					
		reg,imm	1	0	0	0	0	0	0	W	1	1	1	1	0	0	0	0	3-4	reg←reg ∨ imm	U	0	0	x	x	x
		mem,imm	1	0	0	0	0	0	0	W	mod	1	1	0	0	0	0	0	3-6	(mem)←(mem) ∨ imm	U	0	0	x	x	x
		acc,imm	0	0	1	1	0	1	0	W								2-3	When W=0, AL←AL ∨ imm8 When W=1, AW←AW ∨ imm16	U	0	0	x	x	x	

Instruction group	Mnemonic	Operand	Operation code		Number of bytes	Number of clocks	Operation	Flags						
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0				AC	CY	V	P	S	Z	
Bit operation instruction	TEST1	reg8,CL	0 0 0 1 0 0 0 0	1 1 0 0 0 reg	3	3	reg8 bit NO.CL=0 : Z←1 reg8 bit NO.CL=1 : Z←0	U	0	0	U	U	X	
		mem8,CL	0 0 0 0	mod 0 0 0 mem	3-5	8	(mem8) bit NO.CL=0 : Z←1 (mem8) bit NO.CL=1 : Z←0	U	0	0	U	U	X	
		reg16,CL	0 0 0 1	1 1 0 0 0 reg	3	3	reg16 bit NO.CL=0 : Z←1 reg16 bit NO.CL=1 : Z←0	U	0	0	U	U	X	
		mem16,CL	0 0 0 1	mod 0 0 0 mem	3-5	8/12	(mem16) bit NO.CL=0 : Z←1 (mem16) bit NO.CL=1 : Z←0	U	0	0	U	U	X	
		reg8,imm3	1 0 0 0	1 1 0 0 0 reg	4	4	reg8 bit NO.imm3=0 : Z←1 reg8 bit NO.imm3=1 : Z←0	U	0	0	U	U	X	
		mem8,imm3	1 0 0 0	mod 0 0 0 mem	4-6	9	(mem8) bit NO.imm3=0 : Z←1 (mem8) bit NO.imm3=1 : Z←0	U	0	0	U	U	X	
		reg16,imm4	1 0 0 1	1 1 0 0 0 reg	4	4	reg16 bit NO.imm4=0 : Z←1 reg16 bit NO.imm4=1 : Z←0	U	0	0	U	U	X	
		mem16,imm4	1 0 0 1	mod 0 0 0 mem	4-6	9/13	(mem16) bit NO.imm4=0 : Z←1 (mem16) bit NO.imm4=1 : Z←0	U	0	0	U	U	X	
		reg8,CL	0 1 1 0	1 1 0 0 0 reg	3	4	reg8 bit NO.CL←reg8 bit NO.CL							
		mem8,CL	0 1 1 0	mod 0 0 0 mem	3-5	13	(mem8) bit NO.CL←(mem8) bit NO.CL							
		reg16,CL	0 1 1 1	1 1 0 0 0 reg	3	4	reg16 bit NO.CL←reg16 bit NO.CL							
		mem16,CL	0 1 1 1	mod 0 0 0 mem	3-5	13/21	(mem16) bit NO.CL←(mem16) bit NO.CL							
		reg8,imm3	1 1 1 0	1 1 0 0 0 reg	4	5	reg8 bit NO.imm3←reg8 bit NO.imm3							
		mem8,imm3	1 1 1 0	mod 0 0 0 mem	4-6	14	(mem8) bit NO.imm3←(mem8) bit NO.imm3							
reg16,imm4	1 1 1 1	1 1 0 0 0 reg	4	5	reg16 bit NO.imm4←reg16 bit NO.imm4									
mem16,imm4	1 1 1 1	mod 0 0 0 mem	4-6	14/22	(mem16) bit NO.imm4←(mem16) bit NO.imm4									

* : 1st byte = 0FH

2nd byte* 3rd byte*

NOT1	CY	1 1 1 1 0 1 0 1	1	2	CY←CY	X
------	----	-----------------	---	---	-------	---

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags											
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S	Z
Bit operation instruction	CLR1	reg8,CL	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	reg	3	5	reg8 bit NO.CL←0						
		mem8,CL	0	0	1	0	mod	0	0	0	0	0	mem		3-5	14	(mem8) bit NO.CL←0										
		reg16,CL	0	0	1	1	1	0	0	0	0	reg		3	5	reg16 bit NO.CL←0											
		mem16,CL	0	0	1	1	mod	0	0	0	0	mem		3-5	14/22	(mem16) bit NO.CL←0											
	SET1	reg8,imm3	1	0	1	0	1	1	0	0	0	reg		4	6	reg8 bit NO.imm3←0											
		mem8,imm3	1	0	1	0	mod	0	0	0	0	mem		4-6	15	(mem8) bit NO.imm3←0											
		reg16,imm4	1	0	1	1	1	0	0	0	0	reg		4	6	reg16 bit NO.imm4←0											
		mem16,imm4	1	0	1	1	mod	0	0	0	0	mem		4-6	15/23	(mem16) bit NO.imm4←0											
		reg8,CL	0	1	0	0	1	1	0	0	0	reg		3	4	reg8 bit NO.CL←1											
		mem8,CL	0	1	0	0	mod	0	0	0	0	mem		3-5	13	(mem8) bit NO.CL←1											
		reg16,CL	0	1	0	1	1	1	0	0	0	reg		3	4	reg16 bit NO.CL←1											
		mem16,CL	0	1	0	1	mod	0	0	0	0	mem		3-5	13/21	(mem16) bit NO.CL←1											
		reg8,imm3	1	1	0	0	1	1	0	0	0	reg		4	5	reg8 bit NO.imm3←1											
		mem8,imm3	1	1	0	0	mod	0	0	0	0	mem		4-6	14	(mem8) bit NO.imm3←1											
reg16,imm4	1	1	0	1	1	1	0	0	0	reg		4	5	reg16 bit NO.imm4←1													
mem16,imm4	1	1	0	1	mod	0	0	0	0	mem		4-6	14/22	(mem16) bit NO.imm4←1													

* : 1st byte = 0FH

2nd byte* 3rd byte*

CLR1	CY	1	1	1	1	0	0	0	0	1	2	CY←0	0
	DIR	1	1	1	1	0	0	0	0	1	2	DIR←0	
SET1	CY	1	1	1	1	0	0	1	1	2	CY←1	1	
	DIR	1	1	1	1	0	1	1	1	2	DIR←1		

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags										
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S
Shift instruction	SHL	reg,1	1	1	0	1	0	0	0	W	1	1	1	0	0	0	reg	2	6	CY←MSB of reg, reg←regx2 When MSB of reg≠CY, V←1 When MSB of reg=CY, V←0	U	x	x	x	x	x
		mem,1	1	1	0	1	0	0	0	W	mod	1	0	0	mem	2-4	16/24	CY←MSB of (mem), (mem)←(mem)x2 When MSB of (mem)≠CY, V←1 When MSB of (mem)=CY, V←0	U	x	x	x	x	x		
		reg,CL	1	1	0	1	0	0	1	W	1	1	1	0	0	reg	2	7+n	temp←CL, repeats following operation, while temp ≠0: CY←MSB of reg, reg←regx2 temp←temp-1	U	x	U	x	x	x	
		mem,CL	1	1	0	1	0	0	1	W	mod	1	0	0	mem	2-4	19/27+n	temp←CL, repeats following operation, while temp≠0: CY←MSB of (mem), (mem)←(mem)x2 temp←temp-1	U	x	U	x	x	x		
	SHR	reg,imm8	1	1	0	0	0	0	0	W	1	1	1	0	0	reg	3	7+n	temp←imm8, repeats following operation, while temp≠0: CY←MSB of reg, reg←regx2 temp←temp-1	U	x	U	x	x	x	
		mem,imm8	1	1	0	0	0	0	0	W	mod	1	0	0	mem	3-5	19/27+n	temp←imm8, repeats following operation, while temp≠0: CY←MSB of (mem), (mem)←(mem)x2 temp←temp-1	U	x	U	x	x	x		
		reg,1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg	2	6	CY←LSB of reg, reg←reg+2 MSB of reg≠next bit of MSB of reg: V←1 MSB of reg=next bit of MSB of reg: V←0	U	x	x	x	x	x	
		mem,1	1	1	0	1	0	0	0	W	mod	1	0	0	mem	2-4	16/24	CY←MSB of (mem), (mem)←(mem)+2 MSB of reg≠next bit of MSB of reg: V←1 MSB of reg=next bit of MSB of reg: V←0	U	x	x	x	x	x		

n : Number of sifs

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags														
			7	6	5	4	3	2	1	0	reg	mem				reg	mem	reg	mem	reg	mem	reg	mem	reg	mem	U	x	CY	V	P
Shift instruction	SHR (cont'd)	reg,CL	1	1	0	1	0	0	1	W	1	1	1	1	0	0	reg	2	7+n	temp←CL, repeats following operation, while temp≠0: CY←LSB of reg, reg←reg+2 temp←temp-1	U	x	U	x	x	x				
		mem,CL	1	1	0	1	0	0	1	W	mod	1	0	0	mem	2-4	19/27+n	temp←CL, repeats following operation, while temp≠0: CY←LSB of (mem), (mem)←(mem)+2 temp←temp-1	U	x	U	x	x	x						
		reg,imm8	1	1	0	0	0	0	W	1	1	1	0	0	reg	3	7+n	temp←imm8, repeats following operation, while temp≠0: CY←LSB of reg, reg←reg+2 temp←temp-1	U	x	U	x	x	x						
		mem,imm8	1	1	0	0	0	0	W	mod	1	0	0	mem	3-5	19/27+n	temp←imm8, repeats following operation, while temp≠0: CY←LSB of (mem), (mem)←(mem)+2 temp←temp-1	U	x	U	x	x	x							
		reg,1	1	1	0	1	0	0	W	1	1	1	1	1	reg	2	6	CY←LSB of reg, reg←reg+2, V←0 MSB of operand is not affected.	U	x	U	x	x	x						
	SHRA	mem,1	1	1	0	1	0	0	W	mod	1	1	1	1	mem	2-4	16/24	CY←LSB of (mem), (mem)←(mem)+2, V←0 MSB of operand is not affected.	U	x	U	x	x	x						
		reg,CL	1	1	0	1	0	0	1	W	1	1	1	1	1	reg	2	7+n	temp←CL, repeats following operation, while temp ≠0: CY←LSB of reg, reg←reg+2 temp←temp-1, MSB of operand is not affected.	U	x	U	x	x	x					
		mem,CL	1	1	0	1	0	0	1	W	mod	1	1	1	1	mem	2-4	19/27+n	temp←CL, repeats following operation, while temp≠0: CY←LSB of (mem), (mem)←(mem)+2 temp←temp-1, MSB of operand is not affected.	U	x	U	x	x	x					
		reg,imm8	1	1	0	0	0	0	W	1	1	1	1	1	reg	3	7+n	temp←imm8, repeats following operation, while temp≠0: CY←LSB of reg, reg←reg+2 temp←temp-1, MSB of operand is not affected.	U	x	U	x	x	x						
		mem,imm8	1	1	0	0	0	0	W	mod	1	1	1	1	mem	3-5	19/27+n	temp←imm8, repeats following operation, while temp≠0: CY←LSB of (mem), (mem)←(mem)+2 temp←temp-1, MSB of operand is not affected.	U	x	U	x	x	x						

n : Number of sifts

Instruction group	Mnemonic	Operand	Operation code																Number of bytes	Number of clocks	Operation	Flags						
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				AC	CY	V	P	S	Z	
Rotate ion instruction	ROL	reg,1	1	1	0	1	0	0	0	W	1	1	0	0	0	0	0	0	reg	2	6	CY←MSB of reg, reg←regx2+CY When MSB of reg≠CY: V←1 When MSB of reg=CX: V←0	x	x				
		mem,1	1	1	0	1	0	0	0	W	mod	0	0	0	0	0	0	mem	2-4	16/24	CY←MSB of (mem), (mem)←(mem)x2+CY When MSB of (mem)≠CY: V←1 When MSB of (mem)=CY: V←0	x	x					
		reg,CL	1	1	0	1	0	0	1	W	1	1	0	0	0	0	0	reg	2	7+n	temp←CL, repeats following operation, while temp≠0: CY←MSB of reg, reg←regx2+CY temp←temp-1	x		U				
		mem,CL	1	1	0	1	0	0	1	W	mod	0	0	0	0	0	0	mem	2-4	19/27+n	temp←CL, repeats following operation, while temp≠0: CY←MSB of (mem), (mem)←(mem)x2+CY temp←temp-1	x		U				
		reg,imm8	1	1	0	0	0	0	0	W	1	1	0	0	0	0	0	reg	3	7+n	temp←imm8, repeats following operation, while temp≠0: CY←MSB of reg, reg←regx2+CY temp←temp-1	x		U				
		mem,imm8	1	1	0	0	0	0	0	W	mod	0	0	0	0	0	0	mem	3-5	19/27+n	temp←imm8, repeats following operation, while temp≠0: CY←MSB of (mem), (mem)←(mem)x2+CY temp←temp-1	x		U				
		reg,1	1	1	0	1	0	0	0	W	1	1	0	0	0	1	0	reg	2	6	CY←LSB of reg←reg+2 MSB of reg←CY MSB of reg ≠ next bit of MSB of reg: V←1 MSB of reg = next bit of MSB of reg: V←0	x	x					
		mem,1	1	1	0	1	0	0	0	W	mod	0	0	0	1	0	1	mem	2-4	16/21	CY←LSB of (mem)←(mem)+2 MSB of (mem)←CY MSB of (mem) ≠ next bit of MSB of reg: V←1 MSB of (mem) = next bit of MSB of reg: V←0	x	x					

n : Number of sifts



Instruction group	Mnemonic	Operand	Operation code							Number of bytes	Number of clocks	Operation	Flags												
			7	6	5	4	3	2	1				0	AC	CY	V	P	S	Z						
Rotate ion instruction	ROR (cont'd)	reg,CL	1	1	0	1	0	0	1	W	1	1	0	0	1	reg	2	7+n	temp←CL, repeats following operation, while CL≠0: CY←LSB of reg, reg←reg+2 MSB of reg←CY temp←temp-1	x	U				
		mem,CL	1	1	0	1	0	0	1	W	mod	0	0	0	1	mem	2-4	19/27+n	temp←CL, repeats following operation, while CL≠0: CY←LSB of (mem), (mem)←(mem)+2 MSB of (mem)←CY temp←temp-1	x	U				
		reg,imm8	1	1	0	0	0	0	0	W	1	1	0	0	1	reg	3	7+n	temp←imm8, repeats following operation, while CL≠0: CY←LSB of reg, reg←reg+2 MSB of reg←CY temp←temp-1	x	U				
		mem,imm8	1	1	0	0	0	0	0	W	mod	0	0	0	1	mem	3-5	19/27+n	temp←imm8, repeats following operation, while CL≠0: CY←LSB of (mem), (mem)←(mem)+2 MSB of (mem)←CY temp←temp-1	x	U				
	ROLC	reg,1	1	1	0	1	0	0	0	W	1	1	0	1	0	reg	2	6	tmpcy←CY, CY←MSB of reg reg←regx2+tmpcy When MSB of reg≠CY: V←1 When MSB of reg=CY: V←0	x	x				
		mem,1	1	1	0	1	0	0	0	W	mod	0	1	0	0	mem	2-4	16/24	tmpcy←CY, CY←MSB of (mem) (mem)←(mem)x2+tmpcy When MSB of (mem)≠CY: V←1 When MSB of (mem)=CY: V←0	x	x				
		reg,CL	1	1	0	1	0	0	1	W	1	1	0	1	0	reg	2	7+n	temp←CL, repeats following operation, while CL≠0: tmpcy←CY, CY←MBS of reg reg←regx2+tmpcy temp←temp-1	x	U				

n : Number of sifs

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags								
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V
Rotate left instruction	ROL (cont'd)	mem,CL	1	1	0	1	0	0	1	W	mod	0	1	0	mem	2-4	19/27+n	temp←CL, repeats following operation, while CL≠0: tmpcy←CY, CY←MSB of (mem) (mem)←(mem)x2+tmpcy temp←temp-1	x	U				
		reg,imm8	1	1	0	0	0	0	W	1	1	0	1	0	reg	3	7+n	temp←imm8, repeats following operation, while CL≠0: tmpcy←CY, CY←MSB of reg reg←regx2+tmpcy temp←temp-1	x	U				
		mem,imm8	1	1	0	0	0	0	W	mod	0	1	0	mem	3-5	19/27+n	temp←imm8, repeats following operation, while CL≠0: tmpcy←CY, CY←MSB of (mem) (mem)←(mem)x2+tmpcy temp←temp-1	x	U					
		reg,1	1	1	0	1	0	0	0	W	1	1	0	1	1	reg	2	6	tmpcy←CY, CY←LSB of reg reg←reg+2 MSB of reg←tmpcy When MSB of reg≠next bit of MSB of: V←1 When MSB of reg=next bit of MSB of reg: V←0	x	x			
Rotate right instruction	ROR	mem,1	1	1	0	1	0	0	0	W	mod	0	1	1	mem	2-4	16/24	tmpcy←CY, CY←LSB of (mem) (mem)←(mem)+2 MSB of (mem)←tmpcy When MSB of (mem)≠next bit of MSB of (mem): V←1 When MSB of (mem)=next bit of MSB of (mem): V←0	x	x				
		reg,CL	1	1	0	1	0	0	1	W	1	1	0	1	1	reg	2	7+n	temp←CL, repeats following operation, while CL≠0: tmpcy←CY, CY←LSB of reg reg←reg+2 MSB of reg←tmpcy temp←temp-1	x	U			

n : Number of sfts

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags								
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V
Rotate ion instruction	RORC (cont'd)	mem,CL	1	1	0	1	0	0	1	W	mod	0	1	1	mem	2-4	19/27+n	temp←CL, repeats following operation, while CL≠0: tmpcy←CY, CY←LSB of (mem) (mem)←(mem)+2 MSB of (mem)←tmpcy temp←temp-1	x	U				
		reg,imm8	1	1	0	0	0	0	1	W	1	1	0	1	reg	3	7+n	temp←imm8, repeats following operation, while CL≠0: tmpcy←CY, CY←LSB of reg reg←reg+2 MSB of reg←tmpcy temp←temp-1	x	U				
		mem,imm8	1	1	0	0	0	0	0	W	mod	0	1	1	mem	3-5	19/27+n	temp←imm8, repeats following operation, while CL≠0: tmpcy←CY, CY←LSB of (mem) (mem)←(mem)+2 MSB of (mem)←tmpcy temp←temp-1	x	U				

n : Number of sifs

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags											
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S	Z
Branch instruction	BR	near-label	1	1	1	0	1	0	0	1									3	13	PC←PC+disp						
		short-label	1	1	1	0	1	0	1	1									2	12	PC←PC+ext-disp8						
		regptr16	1	1	1	1	1	1	1	1	1	1	1	0	0	reg	2	11	PC←regptr16								
		memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	mem	2-4	20/24	PC←(memptr16)									
		far-label	1	1	1	0	1	0	1	0									5	15	PS←seg PC←offset						
		memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	mem	2-4	27/35	PS←(memptr32+2) PC←(memptr32)									
Condition branch instruction	BV	short-label	0	1	1	1	0	0	0									2	14/4	if V=1 PC←PC+ext-disp8							
	BNV	short-label								0	0	0	1					2	14/4	if V=0 PC←PC+ext-disp8							
	BC	short-label								0	0	1	0					2	14/4	if CY=1 PC←PC+ext-disp8							
	BL																										
	BNC	short-label								0	0	1	1					2	14/4	if CY=0 PC←PC+ext-disp8							
	BNL																										
	BE	short-label								0	1	0	0					2	14/4	if Z=1 PC←PC+ext-disp8							
	BZ																										
	BNE	short-label								0	1	0	1					2	14/4	if Z=0 PC←PC+ext-disp8							
	BNZ																										
	BNH	short-label								0	1	1	0					2	14/4	if CY V Z=1 PC←PC+ext-disp8							
	BH	short-label								0	1	1	1					2	14/4	if CY V Z=0 PC←PC+ext-disp8							
	BN	short-label								1	0	0	0					2	14/4	if S=1 PC←PC+ext-disp8							
	BP	short-label								1	0	0	1					2	14/4	if S=0 PC←PC+ext-disp8							
BPE	short-label								1	0	1	0					2	14/4	if P=1 PC←PC+ext-disp8								
BPO	short-label								1	0	1	1					2	14/4	if P=0 PC←PC+ext-disp8								
BLT	short-label								1	1	0	0					2	14/4	if S V V=1 PC←PC+ext-disp8								
BGE	short-label								1	1	0	1					2	14/4	if S V V=0 PC←PC+ext-disp8								
BLE	short-label								1	1	1	0					2	14/4	if (S V V) V Z=1 PC←PC+ext-disp8								
BGT	short-label								1	1	1	1					2	14/4	if (S V V) V Z=0 PC←PC+ext-disp8								

Instruction group	Mnemonic	Operand	Operation code										Number of bytes	Number of clocks	Operation	Flags										
			7	6	5	4	3	2	1	0	7	6				5	4	3	2	1	0	AC	CY	V	P	S
Interrupt instruction	BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	3	38/50	TA←(4n+1,4n), TC←(4n+3,4n+2) n=imm8 SP←SP-2, (SP+1,SP)←PSW, MD←0 MD is write enabled SP←SP-2, (SP+1,SP)←PS, PS←TC SP←SP-2, (SP+1,SP)←PC, PC←TA						
	CHKIND	reg16,mem32	0	1	1	0	0	0	1	0	mod	reg	mem					2-4	53-56 /73-76 18/26	When (mem32)>reg16 or (mem32+2)<reg16, TA←(015H,014H), TC←(017H,016H) SP←SP-2, (SP+1,SP)←PSW, IE←0, BRK←0 SP←SP-2, (SP+1,SP)←PS, PS←TC SP←SP-2, (SP+1,SP)←PC, PC←TA						
	HALT		1	1	1	1	0	1	0	0								1	2	CPU Halt						
	POLL		1	0	0	1	1	0	1	1								1	2+5n	Poll and wait n: Number of POLL pin sampling operations						
	DI		1	1	1	1	1	0	1	0								1	2	IE←0						
CPU control instruction	EI		1	1	1	1	0	1	1								1	2	IE←1							
	BUSLOCK		1	1	1	1	0	0	0								1	2	Bus Lock Prefix							
	FPO1	fp-op	1	1	0	1	1	X	X	X	1	1	Y	Y	Z	Z	Z	2	2	No Operation						
		fp-op,mem	1	1	0	1	1	X	X	X	mod	Y	Y	Y	mem			2-4	11/15	data bus←(mem)						
	FPO2	fp-op	0	1	1	0	0	1	1	X	1	1	Y	Y	Z	Z	Z	2	2	No Operation						
		fp-op,mem	0	1	1	0	0	1	1	X	mod	Y	Y	Y	mem			2-4	11/15	data bus←(mem)						
	NOP		1	0	0	1	0	0	0	0								1	3	No Operation						
*		0	0	1	1	0	1	1	0		sreg	1	1	0			1	2	Segment over-ride prefix							

Remarks: * indicates four kinds; DS0; DS1; PS; and SS.

Instruction group	Mnemonic	Operand	Operation code								Number of bytes	Number of clocks	Operation	Flags								
			7	6	5	4	3	2	1	0				AC	CY	V	P	S	Z			
8 0	RETEM		1	1	1	0	1	1	1	1	1	0	1	2	27/39	PC←(SP+1,SP),PS←(SP+3,SP+2), PSW←(SP+5,SP+4), SP←SP+6, Sets MD to write disabled	R	R	R	R	R	R
	CALLN	imm8	1	1	1	0	1	1	1	1	0	1	1	3	38/58	TA←(4n+1,4n),TC←(4n+3,4n+2) n=imm8 SP←SP-2,(SP+1,SP)←PSW,MD←1 SP←SP-2,(SP+1,SP)←PS,PS←TC SP←SP-2,(SP+1,SP)←PC,PC←TA						

5. PRELIMINARY ELECTRICAL SPECIFICATIONS

5.1 WHEN SUPPLY VOLTAGE $V_{DD} = 5V \pm 10\%$

Absolute Maximum Rating ($T_a = 25^\circ C$)

Parameter	Symbol	Conditions	Rating	Units
Power supply	V_{DD}		-0.5 to +7.0	V
Input voltage	V_i	$V_{DD} = 5V \pm 10\%$	-0.5 to $V_{DD}+0.3$	V
Clock input voltage	V_k		-0.5 to $V_{DD}+1.0$	V
Output voltage	V_o		-0.5 to $V_{DD}+0.3$	V
Operating temperature	T_{opt}		-40 to +85	°C
Storage temperature	T_{stg}		-65 to +150	°C

DC Characteristics ($T_a = -40$ to $+85^\circ C$, $V_{DD} = 5V \pm 10\%$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Units
High-level input voltage	V_{IH1}	*1	2.2		$V_{DD}+0.3$	V
	V_{IH2}	*2	$0.7V_{DD}$			
Low-level input voltage	V_{IL1}	*1	-0.5		+0.8	V
	V_{IL2}	*2			$0.2V_{DD}$	
High-level clock input voltage	V_{KH}		3.9		$V_{DD}+1.0$	V
Low-level clock input voltage	V_{KL}		-0.5		+0.6	V
High-level output voltage	V_{OH}	$I_{OH} = -100\mu A$	$0.7V_{DD}$			V
Low-level output voltage	V_{OL}	$I_{OL} = 1.6mA$			0.4	V
High-level input leakage current	I_{IH}	$V_i = V_{DD}$			10	μA
Low-level input leakage current	I_{IL}	$V_i = 0V$			-10	μA
High-level output leakage current	I_{OH}	$V_o = V_{DD}$			10	μA
Low-level output leakage current	I_{OL}	$V_o = 0V$			-10	μA
Power supply current	I_{DD}	When operating			150	mA
		When clock input stopped	After HALT instruction executed		50	μA
		Other than the above			5	mA

*1 : Each pin of AD0-AD15, SD0-SD7, DMARQ1-DMARQ3, INTP2-INTP7, ADBIOS, HDINS, KEYLOCK, TCLK.

*2 : Pins other than the above

Capacitance ($T_a = 25^\circ C$, $V_{DD} = 0V$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Units
Input capacitance	C_i	$f_c = 1MHz$			15	pF
I/O capacitance	C_{IO}	Other than measurement pins : 0V			15	pF
Output capacitance	C_o				15	pF

AC Characteristics (Ta = -40 to +85°C, V_{DD} = 5 V±10%)

(1) CPU Timing (1/3)

Parameter	Symbol	Conditions	MIN.	MAX.	Units
X1 input cycle	① t _{CYX}		31.25	DC	ns
X1 input width, high	② t _{XXH}		12		ns
X1 input width, low	③ t _{XXL}		12		ns
X1 input rise	④ t _{XR}			5	ns
X1 input fall	⑤ t _{XF}			5	ns
CPUCLK output cycle	⑥ t _{CYCK}		62.5	DC	ns
CPUCLK output width, high	⑦ t _{CKCKH}		t _{CYCK} /2-15		ns
CPUCLK output width, low	⑧ t _{CKCKL}		t _{CYCK} /2-15		ns
CPUCLK output rise	⑨ t _{CKR}			15	ns
CPUCLK output fall	⑩ t _{CKF}			15	ns
CPUCLK output delay (for X1)	⑪ t _{DXCK}			20	ns
SYSCLK output cycle	⑫ t _{CYSK}		62.5	DC	ns
SYSCLK output width, high	⑬ t _{SKSKH}		t _{CYSK} /2-15		ns
SYSCLK output width, low	⑭ t _{SKSKL}	3-time division	2t _{CYSK} -15		ns
		1, 2, 4-time division	t _{CYSK} /2-15		
SYSCLK output rise	⑮ t _{SKR}			15	ns
SYSCLK output fall	⑯ t _{SKF}			15	ns
SYSCLK output delay (for X1)	⑰ t _{DXSK}			20	ns
TCLK input cycle	⑱ t _{CYTK}		62.5	DC	ns
TCLK input width, high	⑲ t _{TKTKH}		30		ns
TCLK input width, low	⑳ t _{TKTKL}		30		ns
TCLK input rise	㉑ t _{TKR}			25	ns
TCLK input fall	㉒ t _{TKF}			25	ns
PWRGOOD set (for CPUCLK↓)	㉓ t _{SPGCK}		20		ns
PWRGOOD hold (for CPUCLK↓)	㉔ t _{HCKPG}		15		ns
CPUCLK↓→RESOUT output delay	㉕ t _{DCKRSO}		5	30	ns
CPUCLK↑→MA address delay	㉖ t _{DCKMA}			25	ns
X1↑→MA address hold	㉗ t _{HXMA}		0		ns
X1↑→MA address delay	㉘ t _{DXMA}			25	ns
CPUCLK↑→MA address hold	㉙ t _{HCKMA}		0		ns
X1↑→WR active delay	㉚ t _{DXWRL}			40	ns
CPUCLK↓→WR inactive delay	㉛ t _{DCKWRH}			35	ns
X1↑→RAS active delay	㉜ t _{DXRASL}			25	ns

Remarks : Figures in the symbol column correspond to figures in the timing chart.

(1) CPU Timing (2/3)

Parameter	Symbol	Conditions	MIN.	MAX.	Units
CPUCLK↓→RAS inactive delay	33 tDCKRASH			25	ns
X1↑→CAS active delay	34 tDXCASL			25	ns
CPUCLK↑→CAS active delay	35 tDCKCASL			25	ns
X1↓→CAS inactive delay	36 tDXCASH			25	ns
CPUCLK↓→CAS inactive delay	37 tDCKCASH		5	25	ns
CPUCLK↓→AD data output delay	38 tDCKD		5	30	ns
CPUCLK↑→AD data floating delay	39 tFCKD		5	30	ns
AD data input set (for CPUCLK↓)	40 tSDCK		20		ns
AD data input hold (for CPUCLK↓)	41 tHCKD		10		ns
CPUCLK↑→BS↓delay	42 tDKBL		0	55	ns
CPUCLK↓→BS↑delay	43 tDKBH		0	55	ns
SYSCLK→MA address delay	44 tDSKMA			25	ns
SYSCLK→MA address hold	45 tHSKMA		0		ns
SYSCLK↑→WR↓delay	46 tDSKWRL			25	ns
SYSCLK↑→WR↑delay	47 tDSKWRH			40	ns
SYSCLK↑→RAS↓delay	48 tDSKRASL			25	ns
SYSCLK↑→RAS↑delay	49 tDSKRASH			40	ns
SYSCLK↑→CAS↓delay	50 tDSKCASL			25	ns
SYSCLK↑→CAS↑delay	51 tDSKCASH			40	ns
AD data set (for SYSCLK↑)	52 tDSKSK		15		ns
AD data hold (for SYSCLK↑)	53 tHSKSD		10		ns
SYSCLK↑→AD data output delay	54 tDSKDO		10		ns
SYSCLK↓→AD data valid output delay	55 tDSKD		15		ns
SYSCLK↑→AD data floating delay	56 tFSKD		10		ns
CPUCLK↓→SA address delay	57 tDCKSA			25	ns
CPUCLK↓→SA address hold	58 tHCKSA		0		ns
CPUCLK→control output delay	59 tDCKCT			25	ns
X1↑→control output delay	60 tDXCT			25	ns
SYSCLK↑→SA address delay	61 tDSKSA			25	ns
SYSCLK↑→SA address hold	62 tHSKSA		0		ns
SYSCLK↓→SA address delay	63 tDSKSA			30	ns
SYSCLK↓→SA address hold	64 tHSKSA		0		ns
SYSCLK↓→PCS delay	65 tDSKCS			35	ns
SYSCLK↓→PCS hold	66 tHSKCS		0		ns
Address set (for ALE↓)	67 tDSAST		tSKSL-10		ns

Remarks : Figures in the symbol column correspond to figures in the timing chart.

(1) CPU Timing (3/3)

Parameter	Symbol	Conditions	MIN.	MAX.	Units
SYSCLK↓→ALE↑delay	68 t _{DSKSTH1}			25	ns
SYSCLK↑→ALE↑delay	69 t _{DSKSTH2}			35	ns
SYSCLK↑→ALE↓delay	70 t _{DSKSTL}			25	ns
SYSCLK→control 1*1 delay	71 t _{DSKCT1}		5	25	ns
SYSCLK→control 2*1 delay	72 t _{DSKCT2}			40	ns
SD data set (for SYSCLK↓)	73 t _{SDSK}		30		ns
SD data hold (for SMRD, SIORD↑)	74 t _{HRSD}		0		ns
SYSCLK↑→SD data output delay	75 t _{DSKSDO}			25	ns
SYSCLK↑→SD data valid output delay	76 t _{DSKSD}			25	ns
SYSCLK↑→SD data floating delay	77 t _{FSKSD}		5	30	ns
IOCHRDY set (for SYSCLK↑)	78 t _{SRYSK}		15		ns
IOCHRDY hold (for SYSCLK↑)	79 t _{HSKRY}		5		ns
DMARQn set (for SYSCLK↓)	80 t _{SDQSK}		15		ns
SYSCLK→AEN output delay	81 t _{DSKAE}		5	55	ns
SYSCLK↓→DMAAKn output delay	82 t _{DSK LDA}		5	55	ns
SYSCLK↑→TC active output delay	83 t _{DSK TCH}			35	ns
SYSCLK↑→TC inactive output delay	84 t _{DSK TCL}			35	ns
TC width, high	85 t _{DTCTC}		t _{SKSL-20}		ns
NMIIN set (for CPUCLK↓)	86 t _{SNICK}		20		ns
NMIIN hold (for CPUCLK↓)	87 t _{CKNI}		20		ns
NMIOUT output delay (for CPUCLK↓)	88 t _{CKNO}			25	ns
IOCHCK→NMIOUT output delay	89 t _{OCINI}			40	ns
IOCHCK set (for CPUCLK)	90 t _{SICK}		20		ns
INTPn width, low	91 t _{PIPL}			80	ns
KCLK input cycle	92 t _{CYKK}		100		ns
KCLK input width, low	93 t _{TKKKL}		45		ns
KCLK input width, high	94 t _{TKKKH}		45		ns
Keyboard data*2 set (for keyboard clock*3 ↑)	95 t _{SKDKK}		15		ns
Keyboard data*2 hold (for keyboard clock*3 ↑)	96 t _{HKKKD}		10		ns
Keyboard data*2 delay (for keyboard clock*3 ↑)	97 t _{DKKD}			35	ns

*1 : SMRD, SMWR, SIORD, and SIOWR signals when other than DMA transfer

*2 : Keyboard data indicate KDAT or MDAT.

*3 : Keyboard clocks indicate KCLK or MCLK.

Remarks 1 : Relationships between keyboard data and keyboard clocks are as follows.

- When checking KDAT, KCLK is used
- When checking MDAT, MCLK is used

2 : Figures in the symbol column correspond to figures in the timing chart.

(2) DRAM Access Timing (Other Than DMA Transfer)

Parameter	Symbol	Conditions	MIN.	MAX.	Units
Random read/write cycle	201 tRC		4(1+n)t _{cyck} -25		ns
RAS access	202 tRAC			2.25(1+n)t _{cyck} -35	ns
CAS access	203 tCAC			1.25(1+n)t _{cyck} -35	ns
Access from column address	204 tAA			1.75(1+n)t _{cyck} -35	ns
Output buffer turn-off delay	205 tOFF			t _{cyck} -25	ns
RAS precharge	206 tRP		1.75t _{cyck} -20		ns
RAS pulse width (when random read/write cycle)	207 tRAS		2.25(1+n)t _{cyck} -25		ns
RAS hold	208 tRSH		(1+n)t _{cyck} -25		ns
CAS pulse width	209 tCAS		1.25(1+n)t _{cyck} -25		ns
CAS hold	210 tCSH		2.25(1+n)t _{cyck} -25		ns
RAS-CAS delay width	211 tRCD			t _{cyck} -25	ns
CAS-RAS precharge	212 tCRP		1.25t _{cyck} -45		ns
CAS precharge	213 tCPN		2.25t _{cyck} -45		ns
Low address set-up	214 tASR		0		ns
Low address hold	215 tRAH		0.5t _{cyck} -15		ns
Column address set-up	216 tASC		0.5t _{cyck} -25		ns
Column address hold	217 tCAH		2.5(1+n)t _{cyck} -25		ns
Column address hold for RAS	218 tAR		3.75(1+n)t _{cyck} -25		ns
Column address delay for RAS	219 tRAD			0.5t _{cyck} -25	ns
Column address read for RAS	220 tRAL		1.75(1+n)t _{cyck} -25		ns
Read command set-up	221 tRCS		2.75t _{cyck} -55		ns
Read command hold for RAS	222 tRRH		1.75t _{cyck} -45		ns
Read command hold	223 tRCH		1.75t _{cyck} -45		ns
Write command hold	224 tWCH		(1+n)t _{cyck} -25		ns
Write command hold for RAS	225 tWCR		2.25(1+n)t _{cyck} -25		ns
Write command pulse width	226 tWP		2.25(1+n)t _{cyck} -40		ns
Data input set-up	227 tDS		t _{cyck} -30		ns
Data input hold	228 tDH		2(1+n)t _{cyck} -25		ns
Data input hold for RAS	229 tDHR		3.25(1+n)t _{cyck} -25		ns
WE command set-up	230 tWCS		1.25t _{cyck} -40		ns

Remarks : Figures in the symbol column correspond to figures in the timing chart.

(3) DRAM Access Timing (DMA Transfer)

Parameter	Symbol	Conditions	MIN.	MAX.	Units
Random read/write cycle	201 t _{RC}		3.5(1+n)t _{CYCK} -25		ns
RAS access	202 t _{RAC}			2(1+n)t _{CYCK} -25	ns
CAS access	203 t _{CAC}			(1+n)t _{CYCK} -35	ns
Access from column address	204 t _{AA}			1.5(1+n)t _{CYCK} -35	ns
Output buffer turn-off delay	205 t _{OFF}			1.5t _{CYCK}	ns
RAS precharge	206 t _{RP}		1.5t _{CYCK}		ns
RAS pulse width (when random read/write cycle)	207 t _{RAS}		2(1+n)t _{CYCK} -25		ns
RAS hold	208 t _{RSH}	read cycle	(1+n)t _{CYCK} -25		ns
		write cycle	t _{CYCK} -25		
CAS pulse width	209 t _{CAS}	read cycle	(1+n)t _{CYCK} -25		ns
		write cycle	t _{CYCK} -25		
CAS hold	210 t _{CSH}		2(1+n)t _{CYCK} -25		ns
RAS-CAS delay width	211 t _{RCD}			t _{CYCK} -25	ns
CAS-RAS precharge	212 t _{CRP}		2t _{CYCK} -40		ns
CAS precharge	213 t _{CPN}		1.5t _{CYCK}		ns
Low address set-up	214 t _{ASR}		t _{CYCK} -25		ns
Low address hold	215 t _{RAH}		0.5t _{CYCK} -15		ns
Column address set-up	216 t _{ASC}		0.5t _{CYCK} -25		ns
Column address hold	217 t _{CAH}		2(1+n)t _{CYCK} -25		ns
Column address hold for RAS	218 t _{AR}		3(1+n)t _{CYCK} -25		ns
Column address delay for RAS	219 t _{RAD}			0.5t _{CYCK} +25	ns
Column address read for RAS	220 t _{RAL}		1.5(1+n)t _{CYCK} -25		ns
Read command set-up	221 t _{RCS}		3t _{CYCK} -40		ns
Read command hold for RAS	222 t _{RAH}		1.5t _{CYCK}		ns
Read command hold	223 t _{RCH}		1.5t _{CYCK}		ns
Write command hold	224 t _{WCH}		t _{CYCK} -25		ns
Write command hold for RAS	225 t _{WCR}		2(1+n)t _{CYCK} -25		ns
Write command pulse width	226 t _{WP}		2(1+n)t _{CYCK} -25		ns
Data input set-up	227 t _{DOS}		(1+n)t _{CYCK} -30		ns
Data input hold	228 t _{DH}		2t _{CYCK} -25		ns
Data input hold for RAS	229 t _{DHR}		3(1+n)t _{CYCK} -25		ns
WE command set-up	230 t _{WCS}		(1+n)t _{CYCK} -25		ns

Remarks : Figures in the symbol column correspond to figures in the timing chart.

(4) DRAM Access Timing (Refresh Cycle)

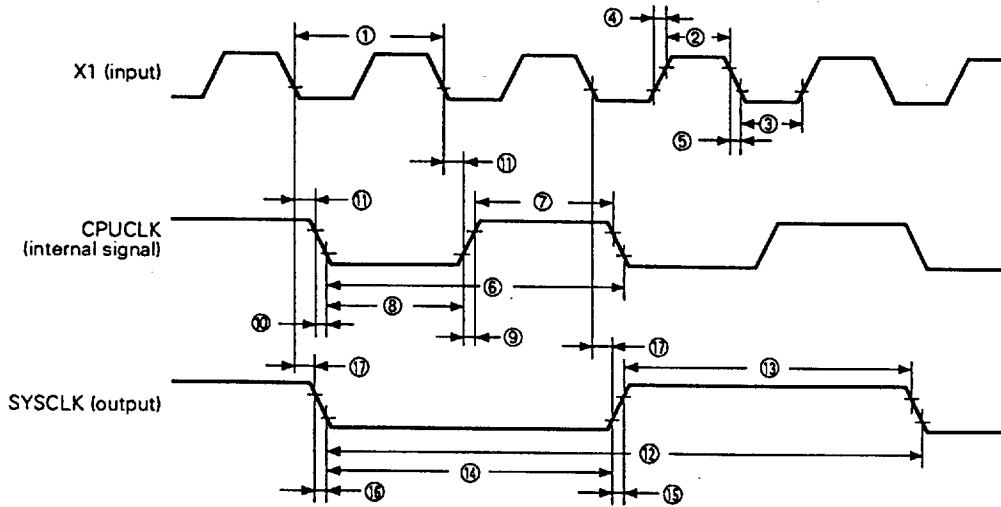
Parameter	Symbol	Conditions	MIN.	MAX.	Units
Random read/write cycle	301 t _{RC}		3.5(1+n)t _{cysk} -25		ns
RAS precharge	302 t _{RP}		2t _{cysk} -40		ns
RAS pulse width (when random read/write cycle time)	303 t _{RAS}		2(1+n)t _{cysk} -25		ns
RAS precharge/CAS hold	304 t _{RPC}		t _{cysk} -40		ns
CAS set-up	305 t _{CSR}		t _{cysk} -25		ns
CAS hold (CAS before RAS refresh)	306 t _{CHR}		2(1+n)t _{cysk} -25		ns

Remarks : Figures in the symbol column correspond to figures in the timing chart.

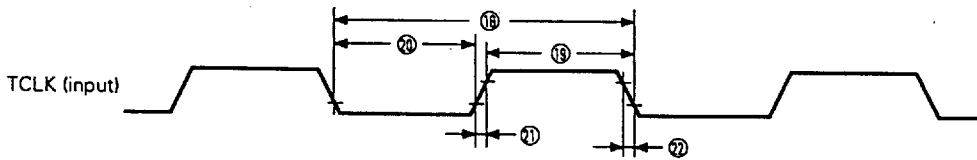
5.2 WHEN SUPPLY VOLTAGE $V_{DD} = 3V \pm 10\%$

Under evaluation

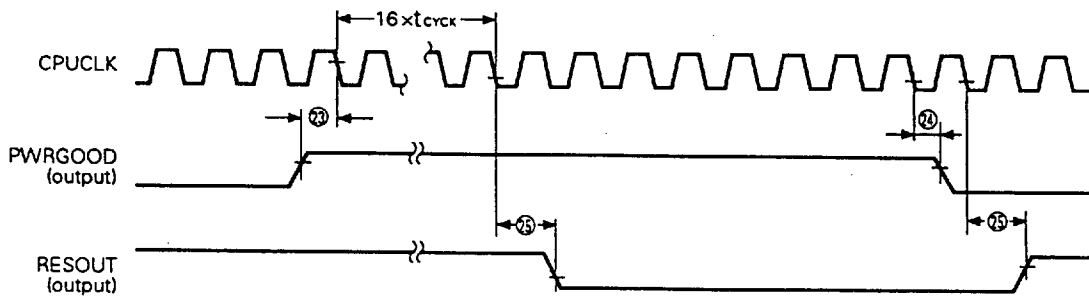
Clock Timing



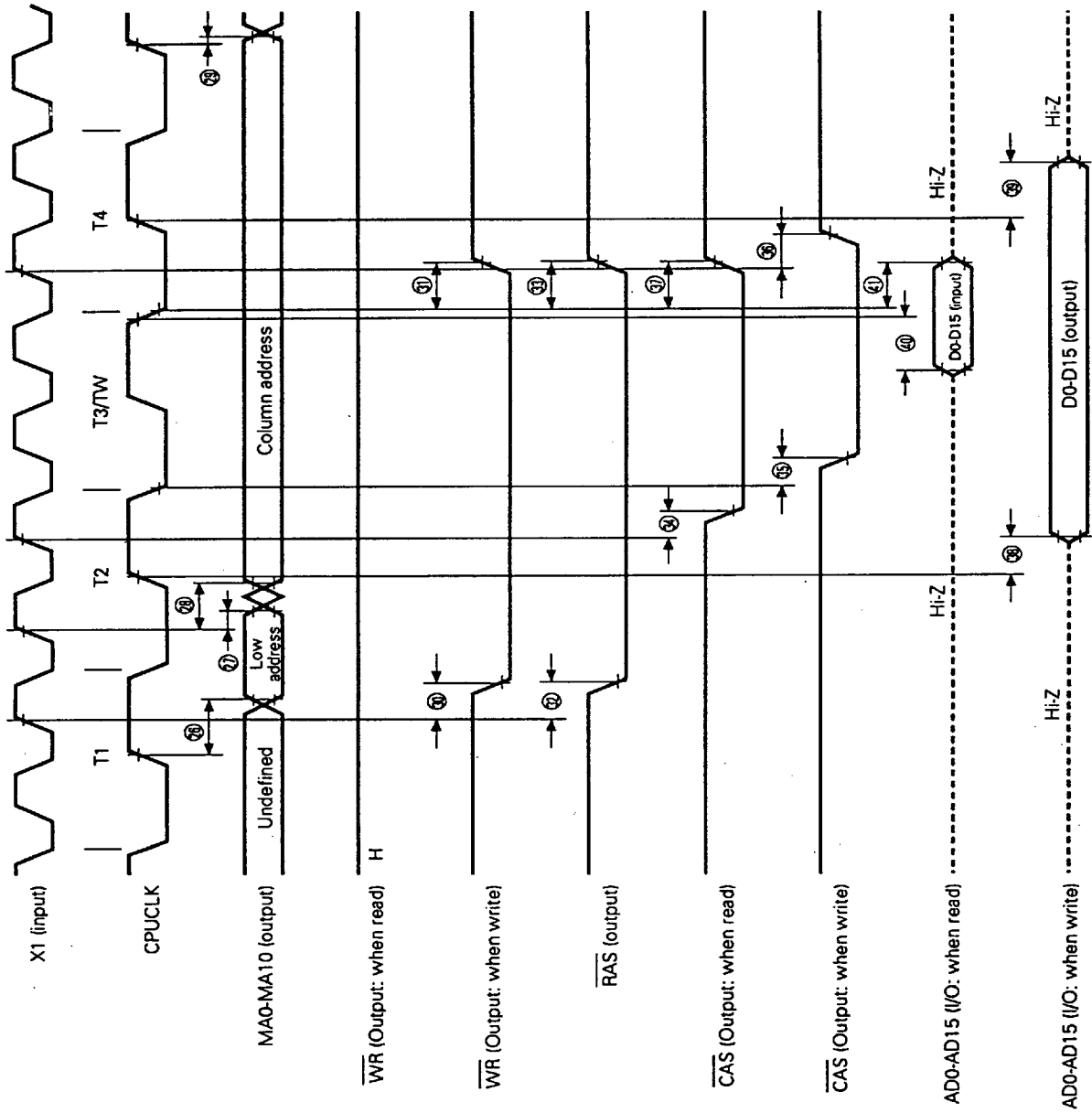
Timer Clock Timing



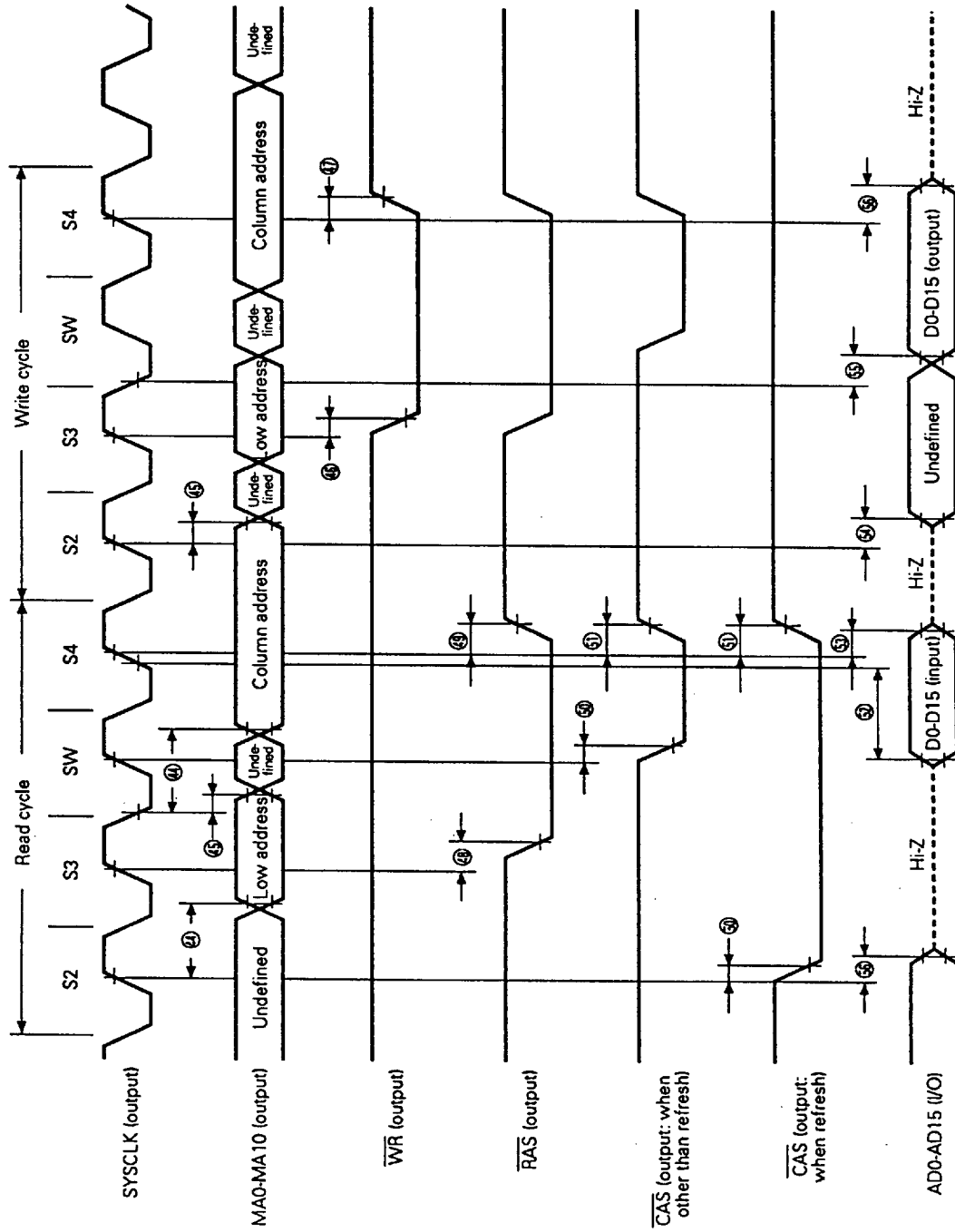
Reset Timing



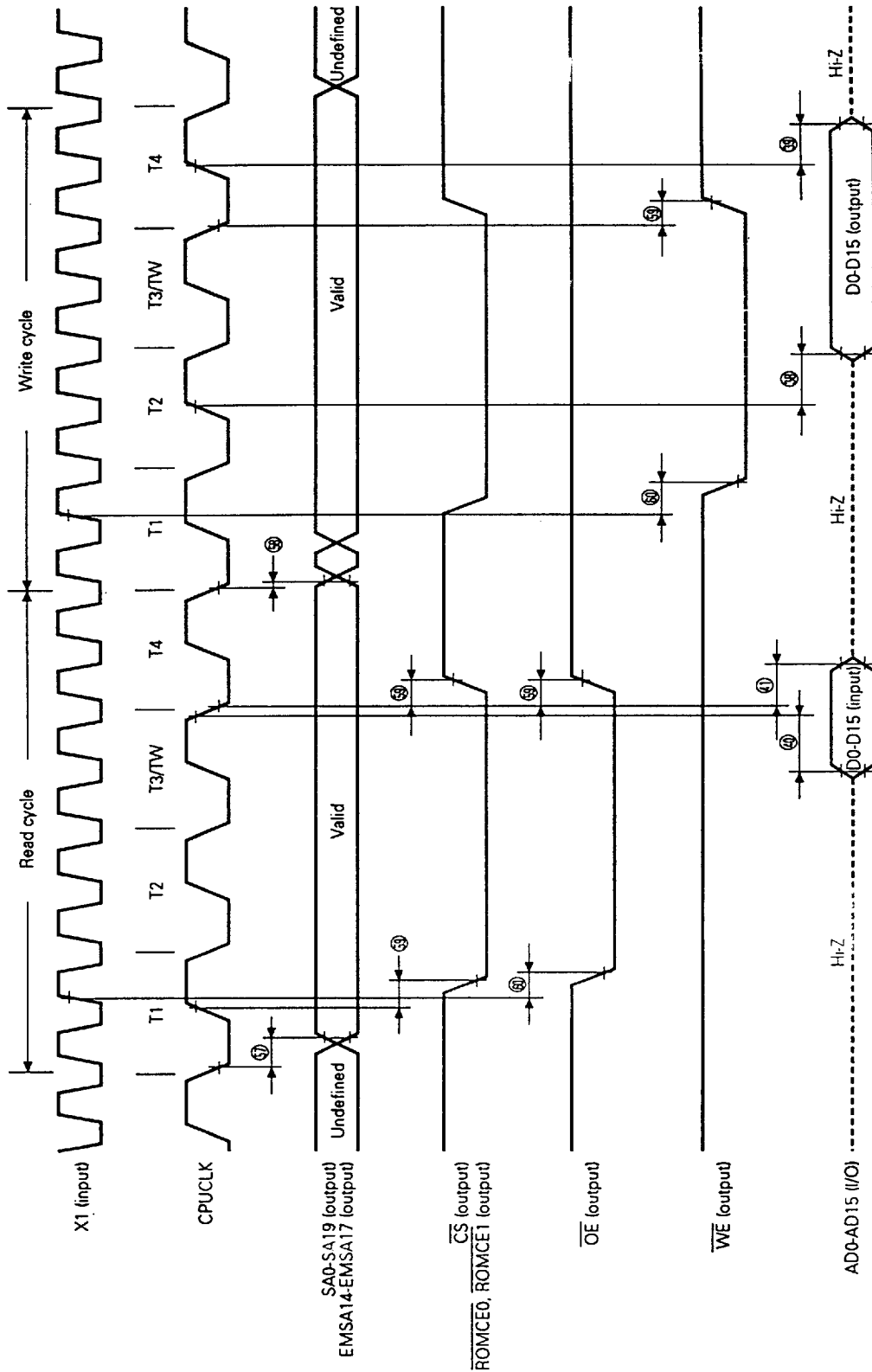
DRAM Access Timing (Other Than DMA Transfer)



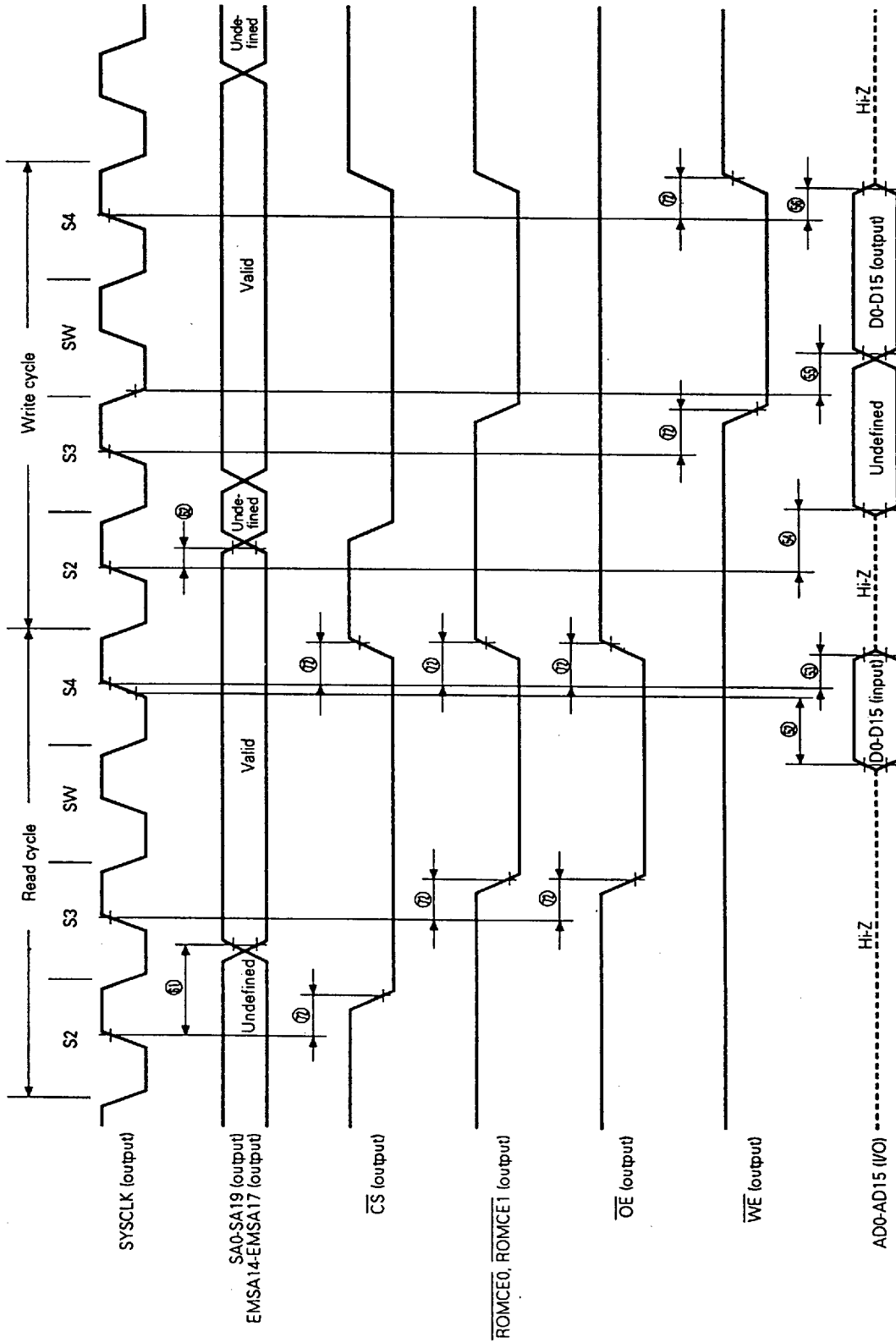
DRAM Access Timing (DMA Transfer)



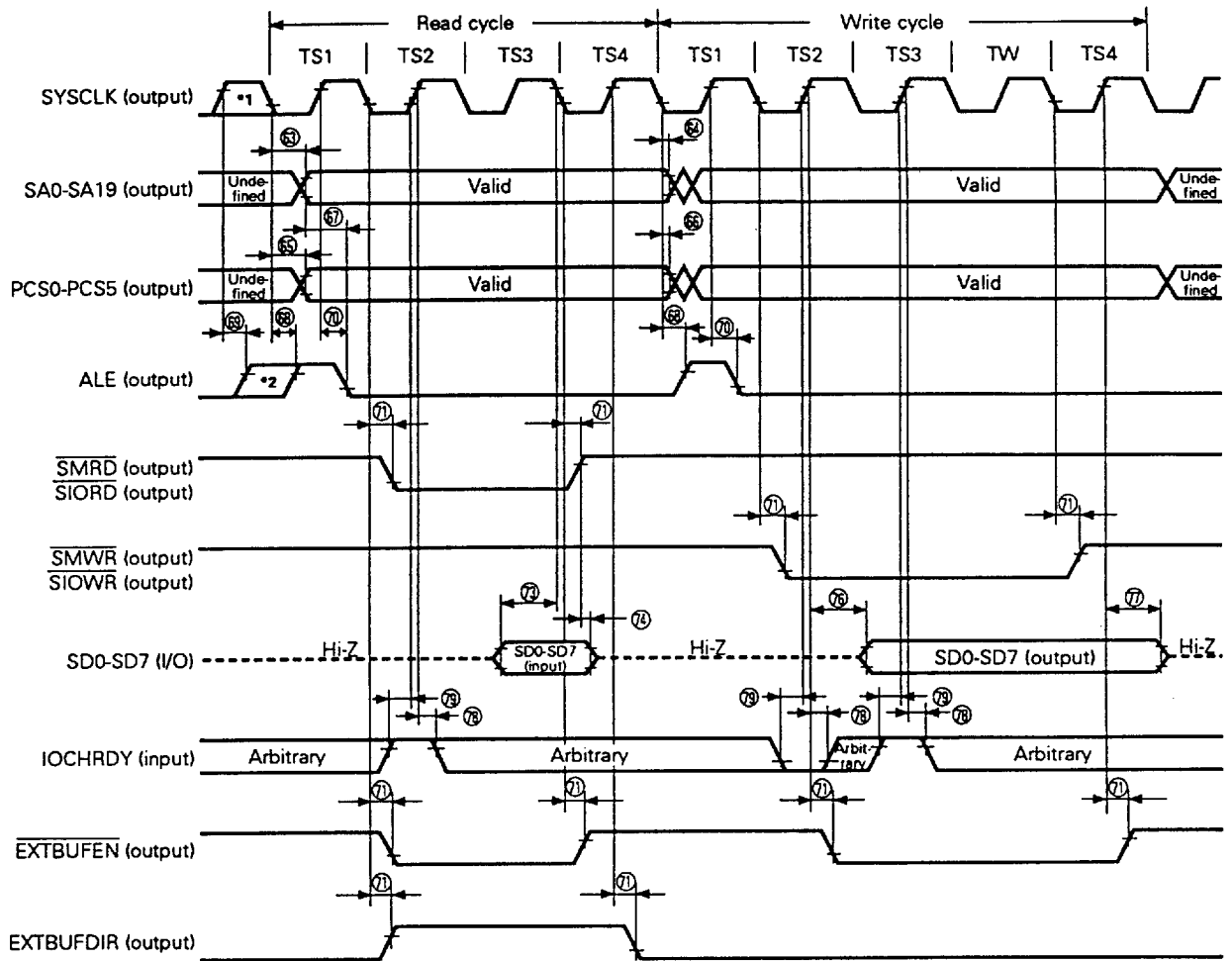
SRAM, Pseudo-SRAM and ROM Access Timing (Other Than DMA Transfer)



SRAM, Pseudo-SRAM and ROM Access Timing (DMA Transfer)

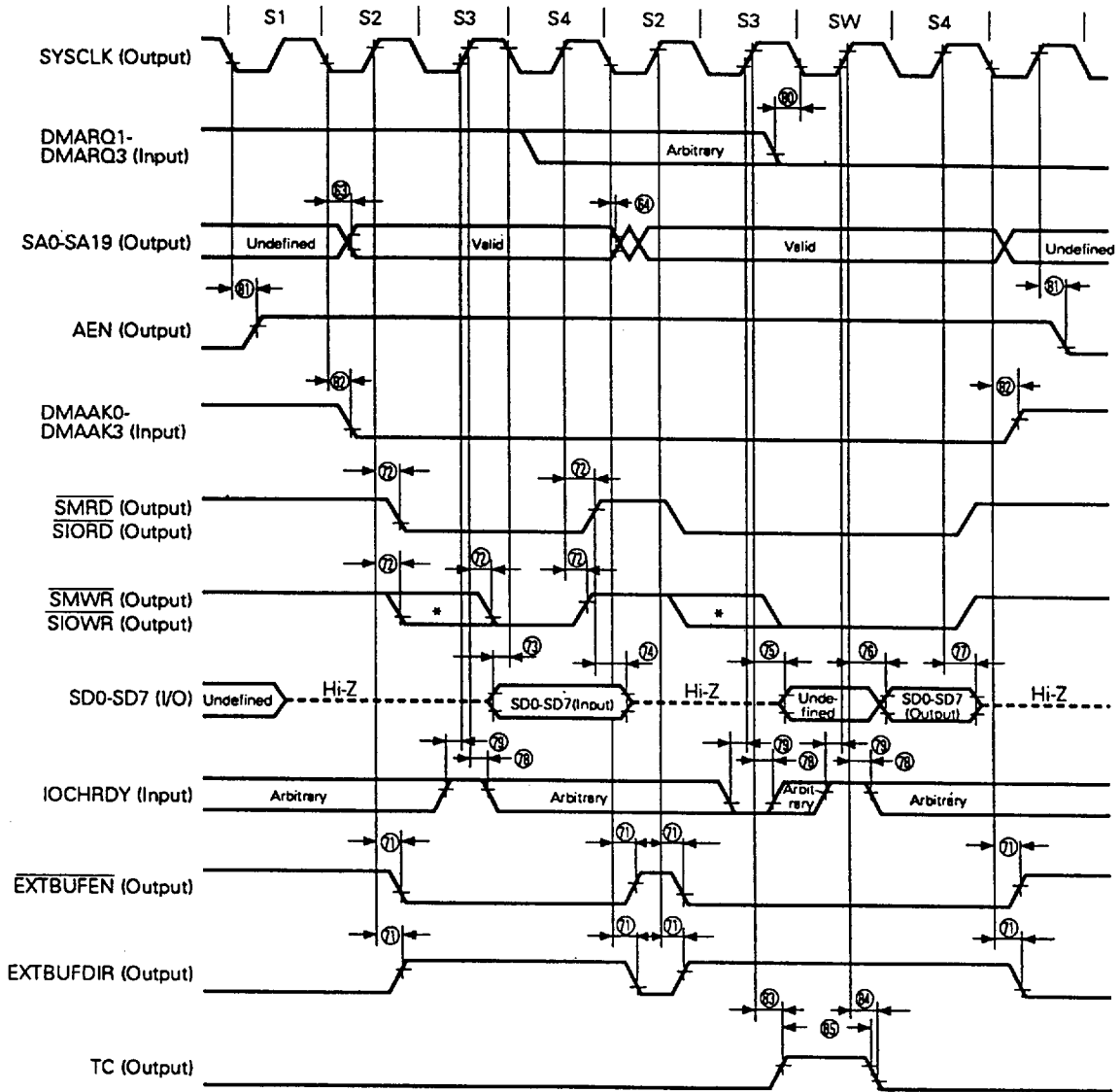


I/O Channel Interface Timing (Other Than DMA Transfer)



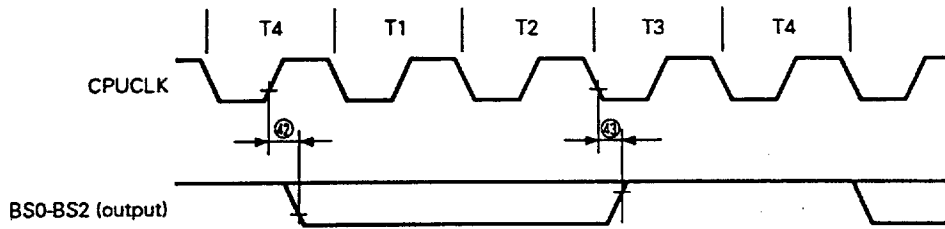
- *1 : High-level only when cycles of SYSCLK are the same as those of CPUCLK.
- *2 : May become high-level half a clock before, unless bus cycles are continuous.

I/O Channel Interface Timing (DMA Transfer)

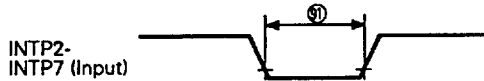


* : Outputs low level when extended write.

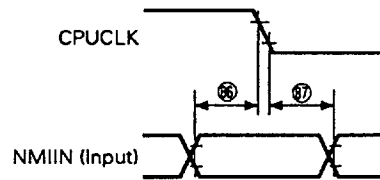
BS0-BS2 Output Timing



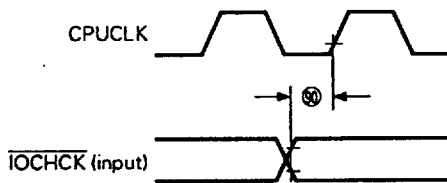
INTP2-INTP7 Input Timing



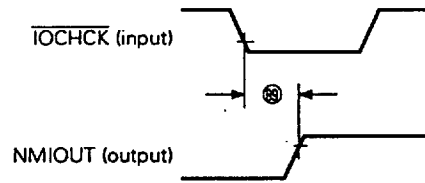
NMIIN Sample Timing



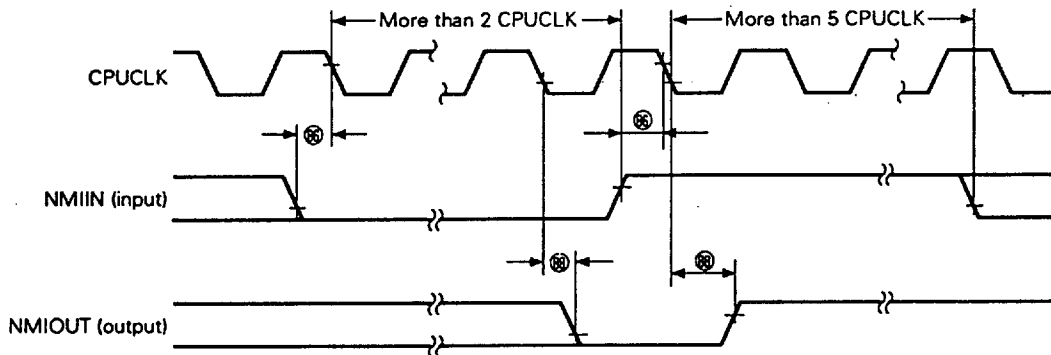
IOCHCK Input Timing



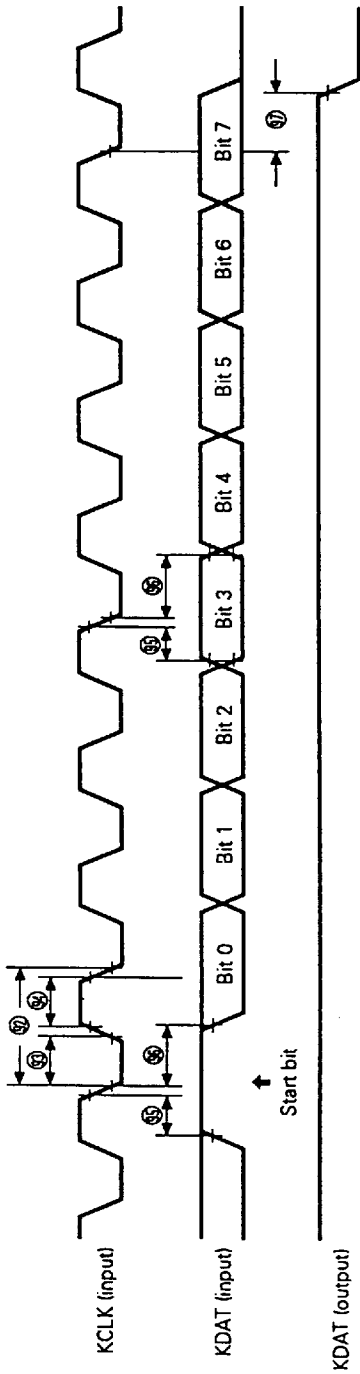
NMIOUT Output Timing (1)



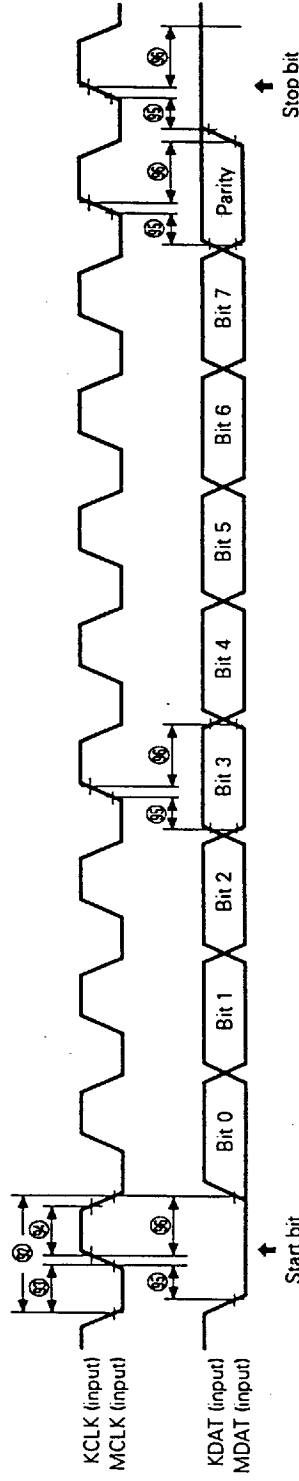
NMIOUT Output Timing (2)



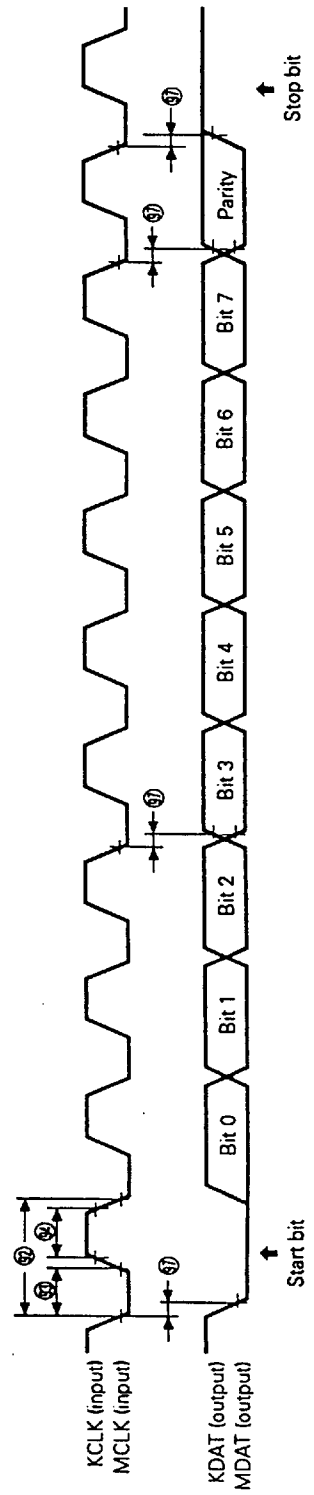
Keyboard Interface (XT Type: Reception)



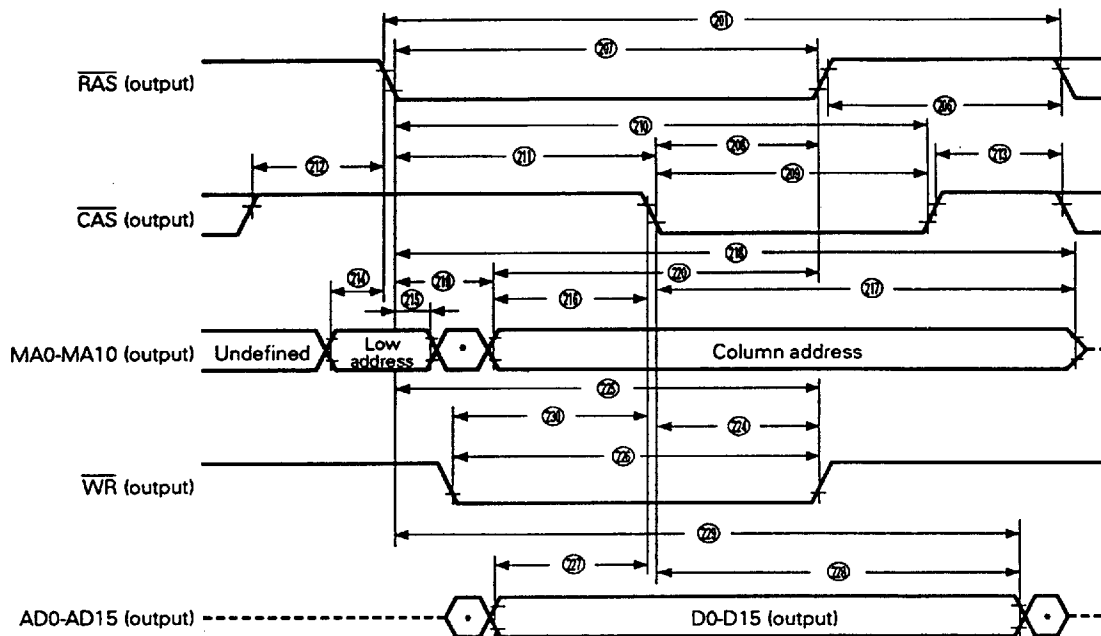
Keyboard Interface (PS/2 Model 30 Type: Reception)



Keyboard Interface (PS/2 Model 30 Type: Transfer)



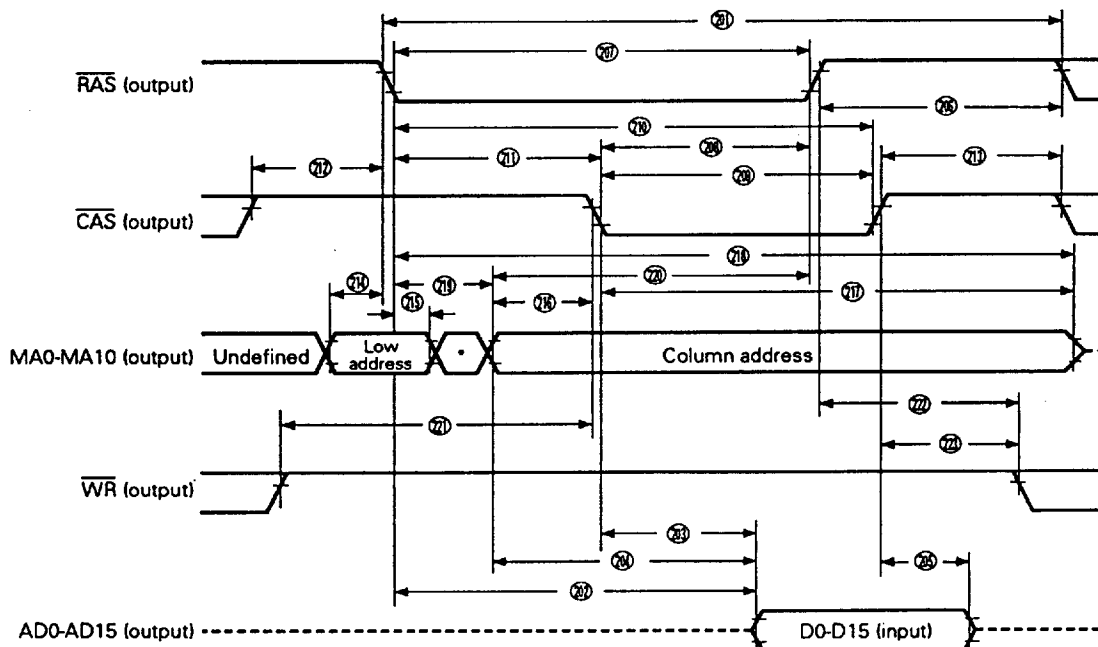
DRAM Write Timing



• : Undefined

Remarks : Broken lines indicate high impedance.

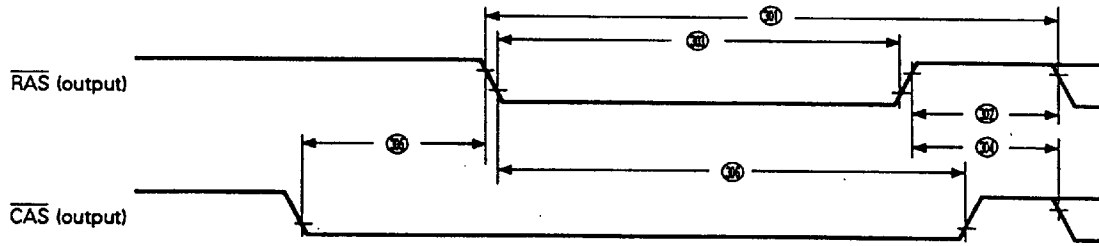
DRAM Read Timing



• : Undefined

Remarks : Broken lines indicate high impedance.

DRAM Refresh Cycle



6. PACKAGE DRAWINGS

160 Pin Plastic QFP (□ 28) (Units: mm)

