PRELIMINARY DATA SHEET

NEC
ELECTRON DEVICE

MOS INTEGRATED CIRCUIT
μPD70270

V41™ 16/8-BIT MICROPROCESSOR

DESCRIPTION

The μPD70270 (V41) is a 16/8-bit microprocessor containing a CPU equivalent to the μPD70108H (V20HL™) and peripheral functions for the PC/XT™ on a single chip.

μPD70270 employs a μPD70108H 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 μPD70280 (V51™) is identical to the μPD70270, except that it is provided with an 16-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 μPD71054)
  - • DMA control unit (equivalent to μPD71037)
  - • Interrupt control unit (equivalent to μPD71059)
  - • 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 μPD70108H 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 3.5μs (16MHz, 5V)
  - 2.4 to 7.0μs (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)

C

Document No. ID-8408
(O. D. No. ID)
Date Published July 1992 P
Printed in Japan

APPLICATIONS

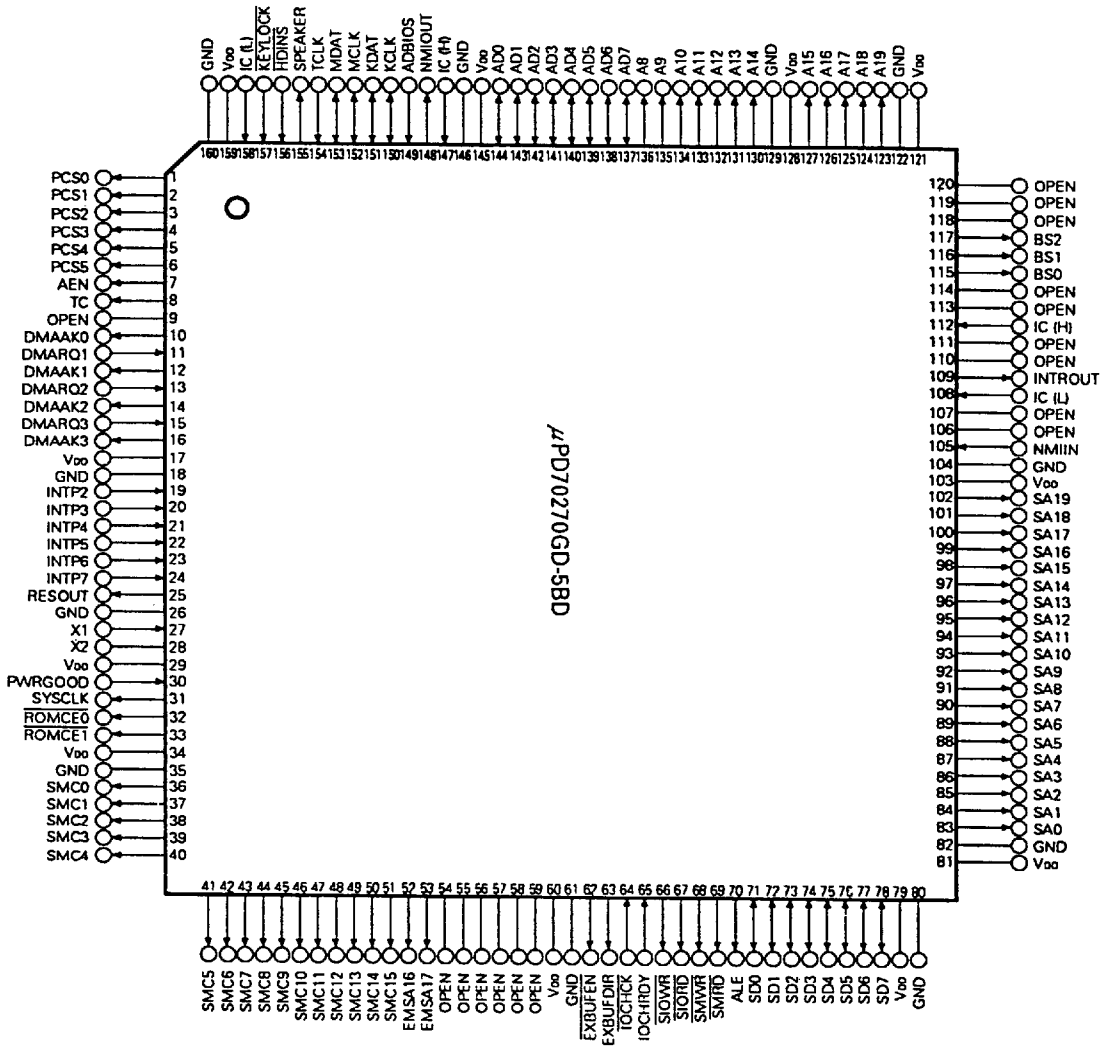
- o OA related devices (portable terminals, personal computers, word processors, electronic typewriters, multi-function telephones, etc.)
- o Control equipment (robotic control, NC control, communication control, etc.)

ORDERING INFORMATION

Part Number	Package	Quality Grade
μPD70270GD-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)

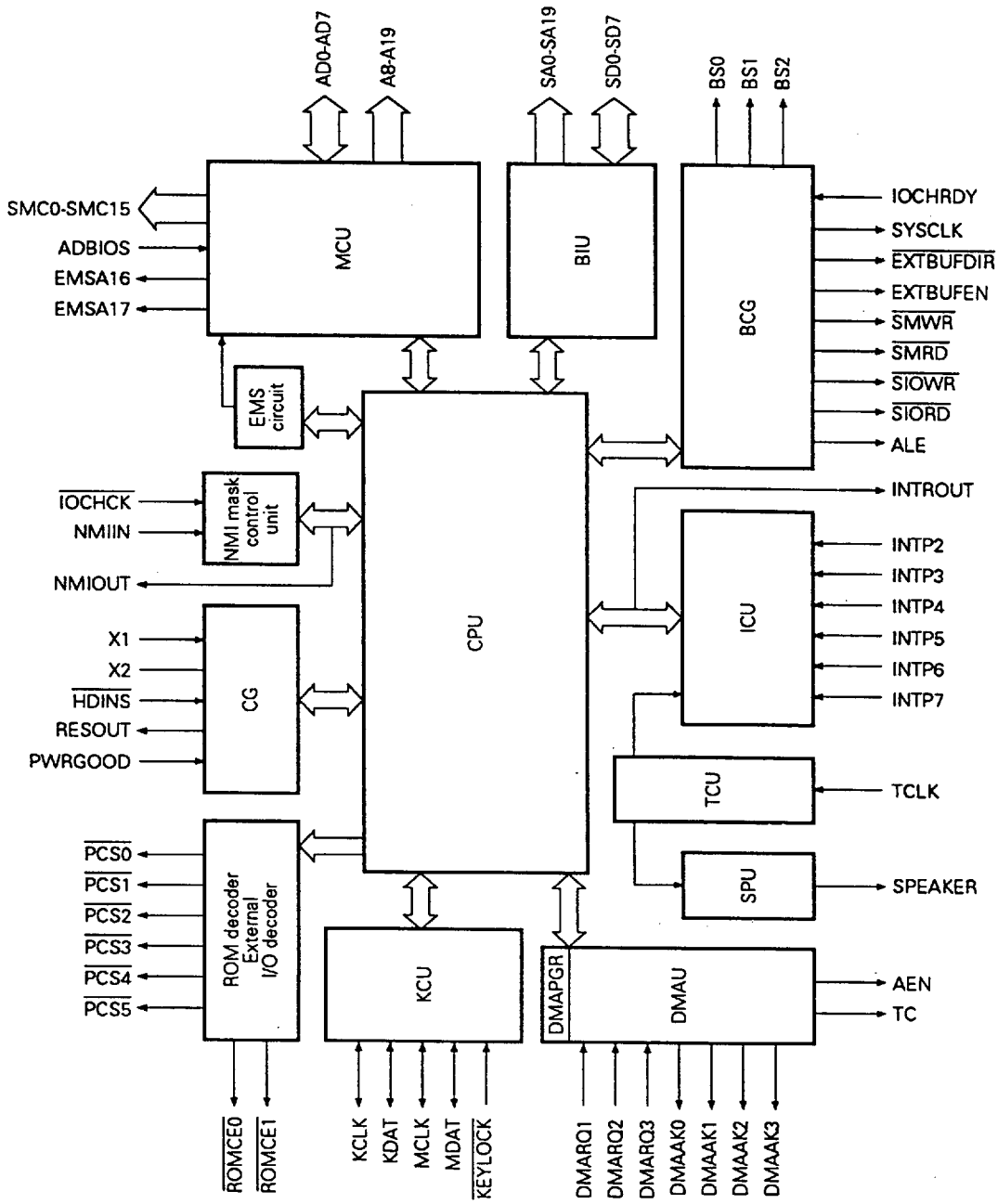


Note: 1. Fix each IC(H) pin to high level through an external pull-up resistor.
2. Fix each IC(L) pin to low level through an external pull-down resistor.
3. Do not connect anything to the OPEN pin.

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
$\overline{\text{EXTBUFEN}}$	:	External Buffer Enable
GND	:	Ground
$\overline{\text{HDINS}}$	:	Hard Disk Install
IC(H), IC(L)	:	Internally Connected
INTP2-INTP7	:	Interrupt Request from Peripheral
INTROUT	:	Interrupt Request Out
$\overline{\text{IOCHCK}}$	:	I/O Channel Check
IOCHRDY	:	I/O Channel Ready
KCLK	:	Keyboard Clock
KDAT	:	Keyboard Data
$\overline{\text{KEYLOCK}}$	:	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
$\overline{\text{ROMCE0}}$, $\overline{\text{ROMCE1}}$	:	ROM Chip Enable
SA0-SA19	:	System Address Bus
SD0-SD7	:	System Data Bus
$\overline{\text{SIORD}}$	:	System I/O Read Strobe
$\overline{\text{SIOWR}}$	:	System I/O Write Strobe
SMC0-SMC15	:	System Memory Control
$\overline{\text{SMRD}}$	:	System Memory Read Strobe
$\overline{\text{SMWR}}$	:	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 V20HL
Internal Data Bus Width		16 bits
External Data Bus Width		8 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		$\overline{CS}$ signal generation for external I/O device (6 lines)
ROM Decoder		$\overline{CS}$ signal generation for extended ROM and BIOS ROM
Interrupt Function		External: 6, internal: 2
NMI Function		2 input pins ($\overline{IOCHCK}$, 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

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. (Refer to 1.2 SMC Functions)
EMSA16,EMSA17	Output	EMS address bit output
AD0-AD7	3-state I/O	Address signal output/Data input (time division multiplex)
A8-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 $\overline{\text{IOCHCK}}$ 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*	256K×1	256K×4	1M×1	1M×4	4M×1	4M×4	32K × 8	128K × 8	32K × 8	128K × 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						$\overline{CS0}$			
SMC1	MA1						$\overline{CS1}$			
SMC2	MA2						$\overline{CS2}$			
SMC3	MA3						$\overline{CS3}$			
SMC4	MA4						$\overline{CS4}$			
SMC5	MA5						$\overline{CS5}$			
SMC6	MA6						$\overline{CS6}$			
SMC7	MA7						$\overline{CS7}$			
SMC8	MA8						$\overline{CS8}$			
SMC9	$\overline{CAS3}$		MA9				$\overline{CS9}$			$\overline{RFSH}$
SMC10	$\overline{CAS2}$				MA10		NC	EMSA14	NC	EMSA14
SMC11	$\overline{CAS1}$						NC	EMSA15	NC	EMSA15
SMC12	$\overline{CAS0}$						$\overline{OEH}$		$\overline{OE/RFSHH}$	$\overline{OEH}$
SMC13	$\overline{RASH}$						$\overline{OEL}$		$\overline{OE/RFSHL}$	$\overline{OEL}$
SMC14	$\overline{RASL}$						$\overline{WEH}$			
SMC15	$\overline{WE}$						$\overline{WEL}$			

*: Words × 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
$\overline{CAS0}$-$\overline{CAS3}$	DRAM $\overline{CAS}$ signal. Selects bank
$\overline{RASL}$, $\overline{RASH}$	DRAM $\overline{RAS}$ signal. Selects upper memory, lower memory
$\overline{WE}$	DRAM write enable signal
$\overline{CS0}$-$\overline{CS9}$	SRAM, pseudo-SRAM chip select signal
$\overline{RFSH}$	Pseudo-SRAM refresh signal
$\overline{OEL}$, $\overline{OEH}$	SRAM, pseudo-SRAM $\overline{OE}$ signal
$\overline{OE}$/$\overline{RFSHL}$, OE/$\overline{RFSHH}$	Pseudo-SRAM $\overline{OE}$/$\overline{RFSH}$ signal
$\overline{WEL}$, $\overline{WEH}$	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-AD7		
A8-A19		
$\overline{\text{ROMCE0}}$		
$\overline{\text{ROMCE1}}$		
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	
$\overline{\text{SMWR}}$	Depends on DMA operation	No change (except during DMA operation)
$\overline{\text{SMRD}}$		
$\overline{\text{SIOWR}}$		
$\overline{\text{SIORD}}$		
DMAAK0-DMAAK3	Depends on DMARQ	
AEN	High level	
TC	Output at count completion	
$\overline{\text{EXTBUFEN}}$	Depends on DMA operation	
$\overline{\text{EXTBUFDIR}}$		

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

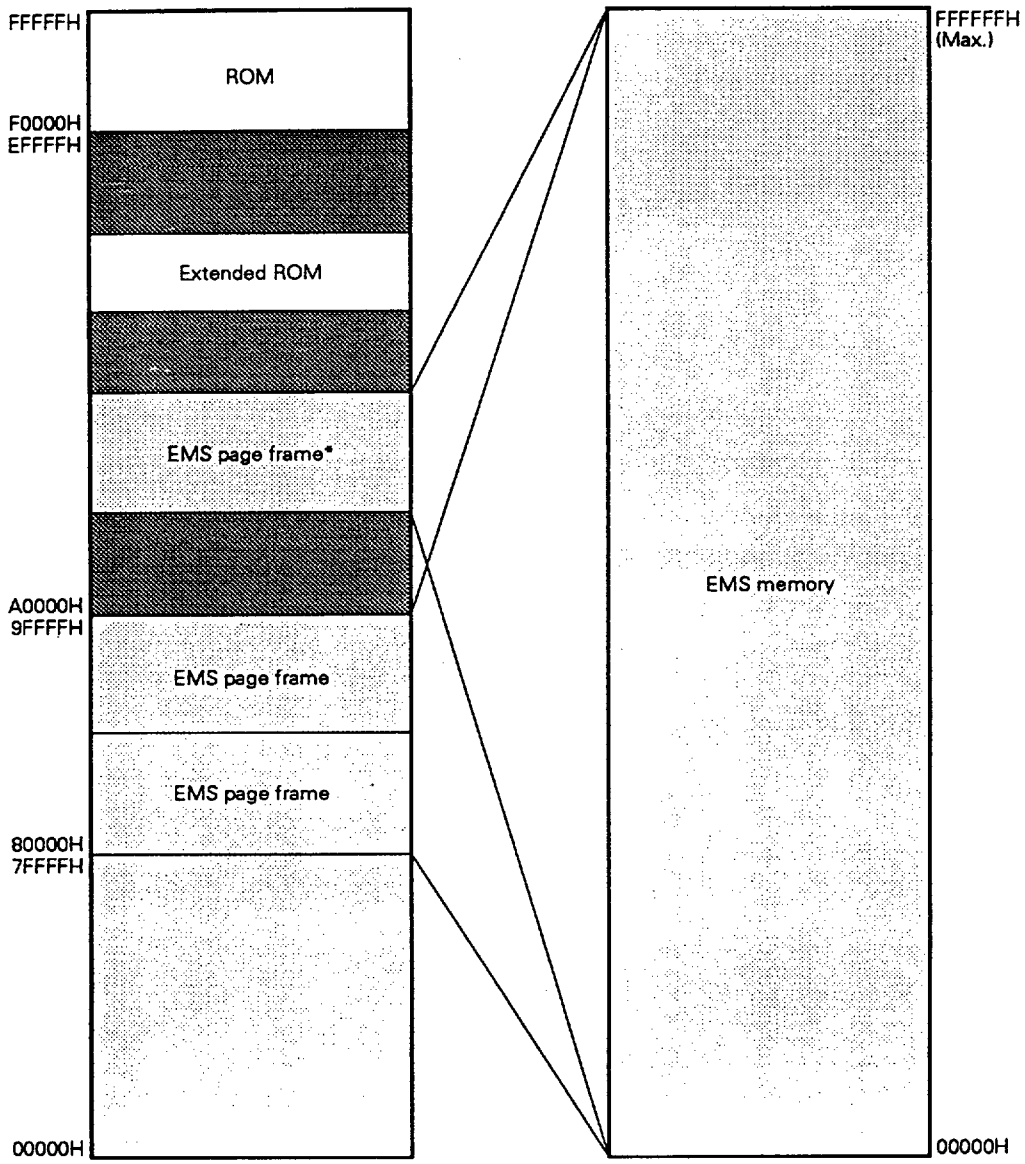
Pin name	I/O	Recommended connections
RESOUT	Output	Open
SMC0-SMC15	Output	
EMSA16, EMSA17	Output	
AD0-AD7	3-state I/O	Connect to VDD through resistor
A8-A19	Output	Open
$\overline{\text{ROMCE0}}$	Output	Open
$\overline{\text{ROMCE1}}$	Output	
PCS0-PCS5	Output	
KCLK	3-state I/O	
KDAT	3-state I/O	Connect to VDD or GND through resister
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 VDD or GND through resister
ALE	Output	Open
BS0-BS2	Output	
$\overline{\text{SMWR}}$	Output	
$\overline{\text{SMRD}}$	Output	
$\overline{\text{SIOWR}}$	Output	
$\overline{\text{SIORD}}$	Output	
IOCHRDY	Input	Connect to VDD through resistor
$\overline{\text{IOCHCK}}$	Input	
NMIIN	Input	Connect to GND through resister
NMIOUT	Output	Open
INTP2-INTP7	Input	Connect to VDD or GND through resister
INTROUT	Output	Open
DMARQ1-DMARQ3	Input	Connect to GND through resister
DMAAK0-DMAAK3	Output	Open
AEN	Output	
TC	Output	
$\overline{\text{EXTBUFEN}}$	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 μPD70270 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

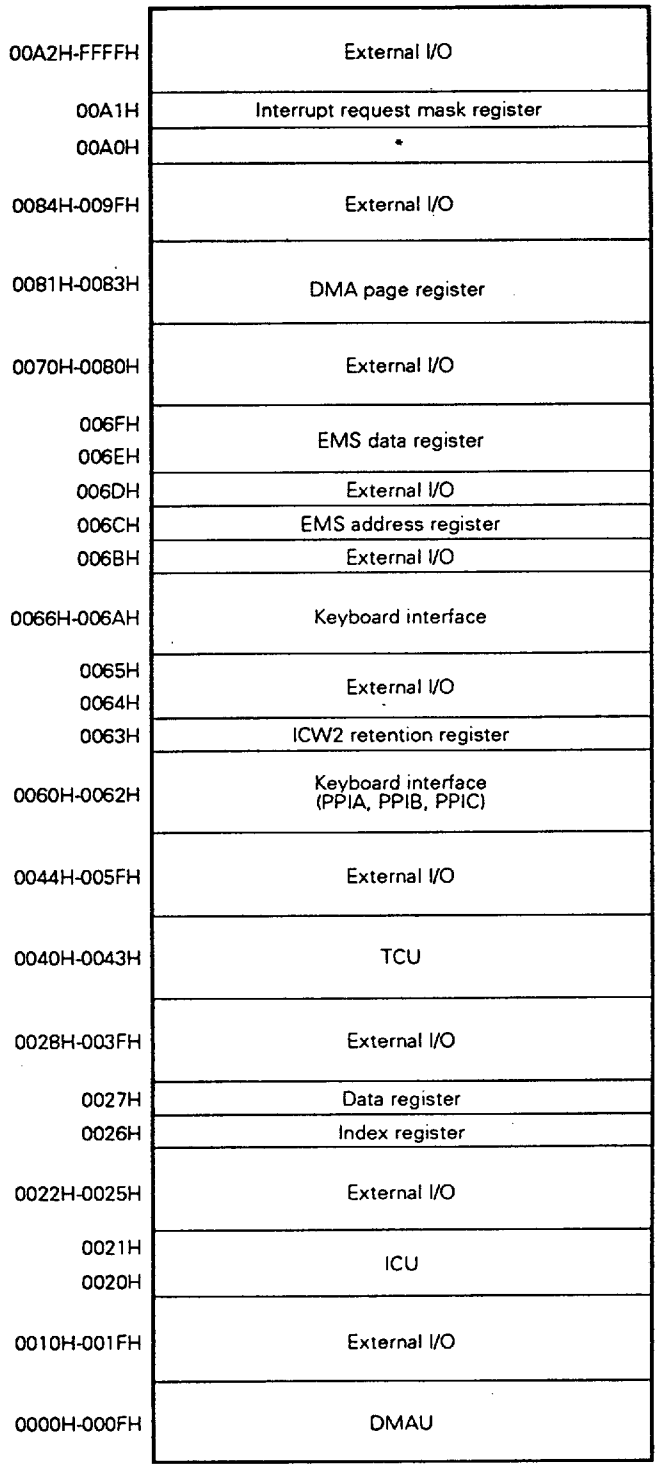
2.2 I/O SPACE

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 μPD70108H (V20HL) 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 μPD70270 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.



*: 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 I 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 μPD70108H (V20HL). 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 μPD70108H.

3.2 CG (CLOCK GENERATOR)

The μPD70270 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.

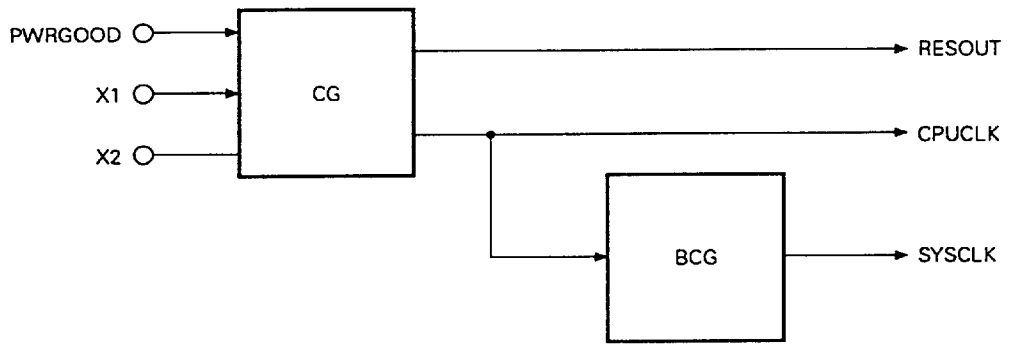


Fig. 3-1 Relationship for CG and BCG

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.

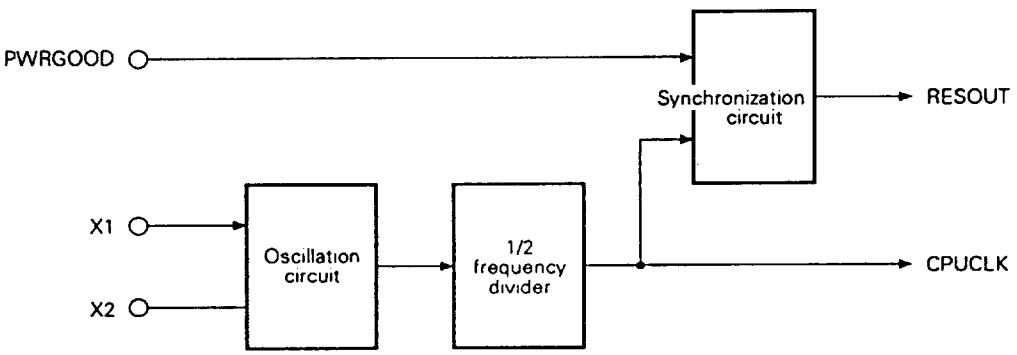


Fig. 3-2 CG Internal Block Diagram

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.

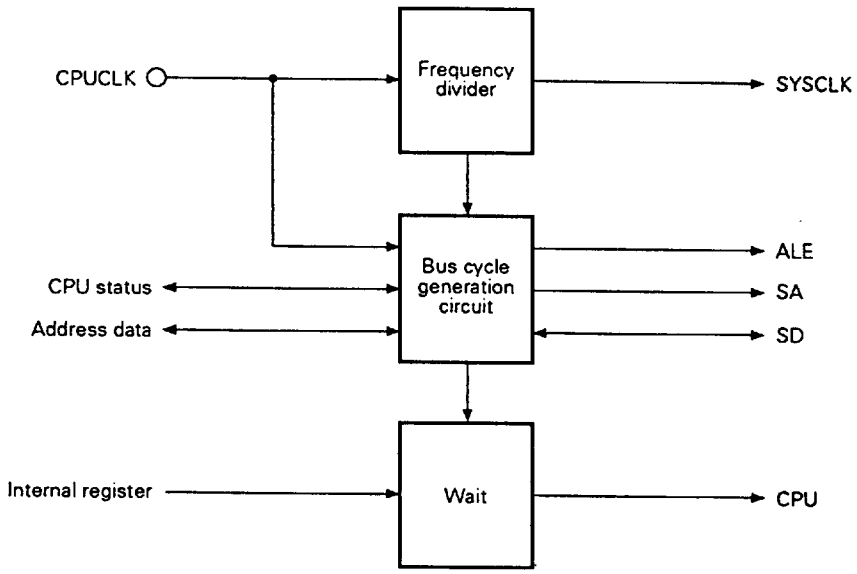


Fig. 3-3 BCG Internal Block Diagram

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.

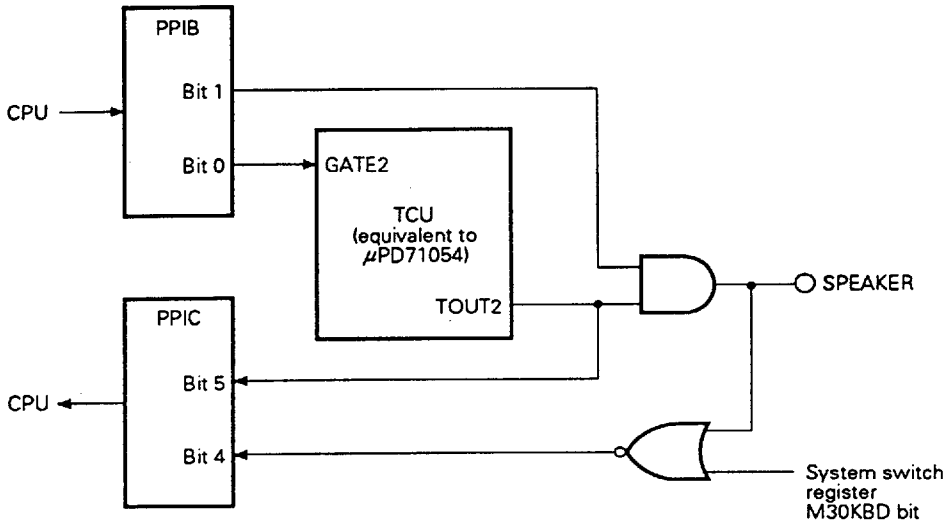


Fig. 3-4 Speaker Interface

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.

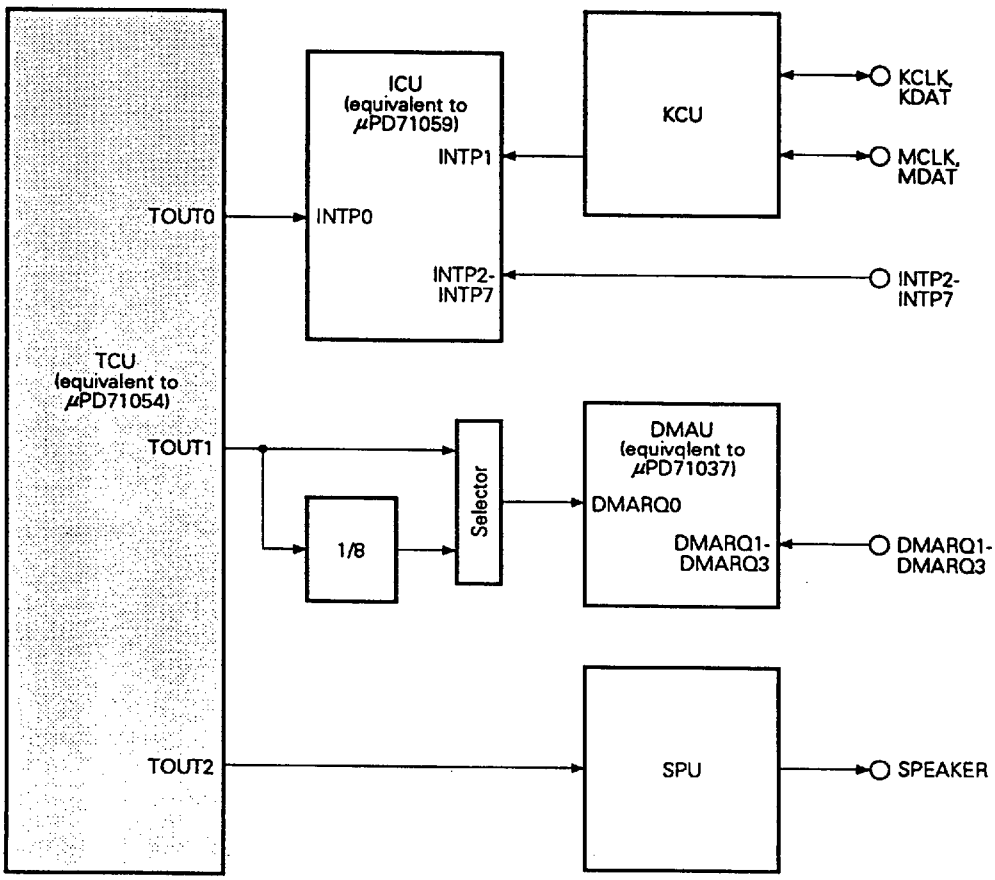


Fig. 3-5 Relationship for TCU and Internal Peripheral Units

3.7 ICU (INTERRUPT CONTROL UNIT)

The ICU arbitrates up to eight interrupt requests (maskable interrupts) generated inside and outside of the μPD70270, and transfers one of them to the CPU. The ICU function is identical to the μPD71059, except that functions, not necessary for the μPD70270, 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

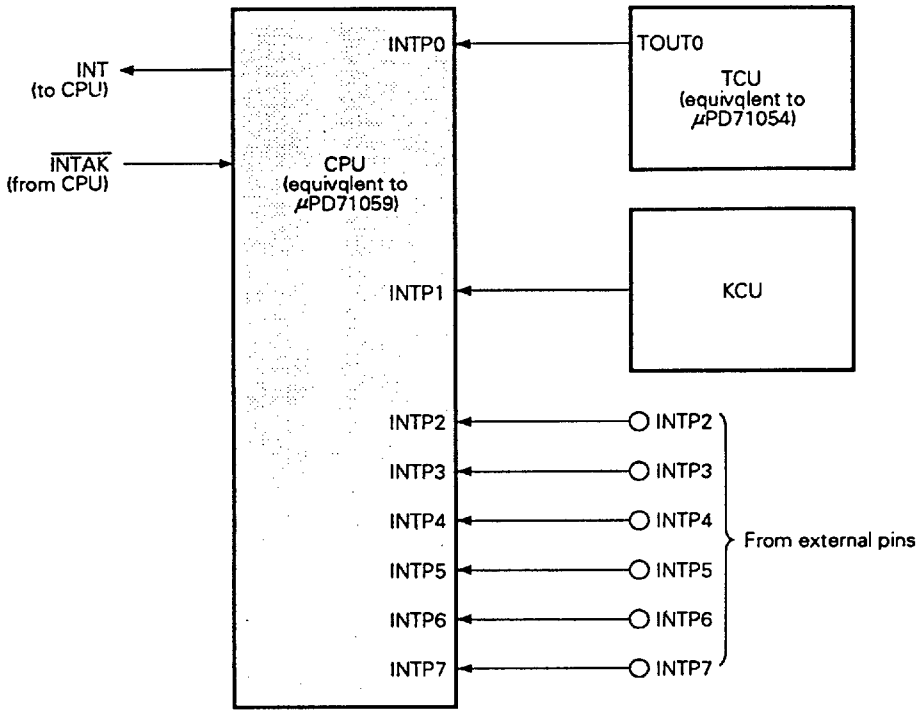


Fig. 3-6 Interrupt Request to ICU

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.

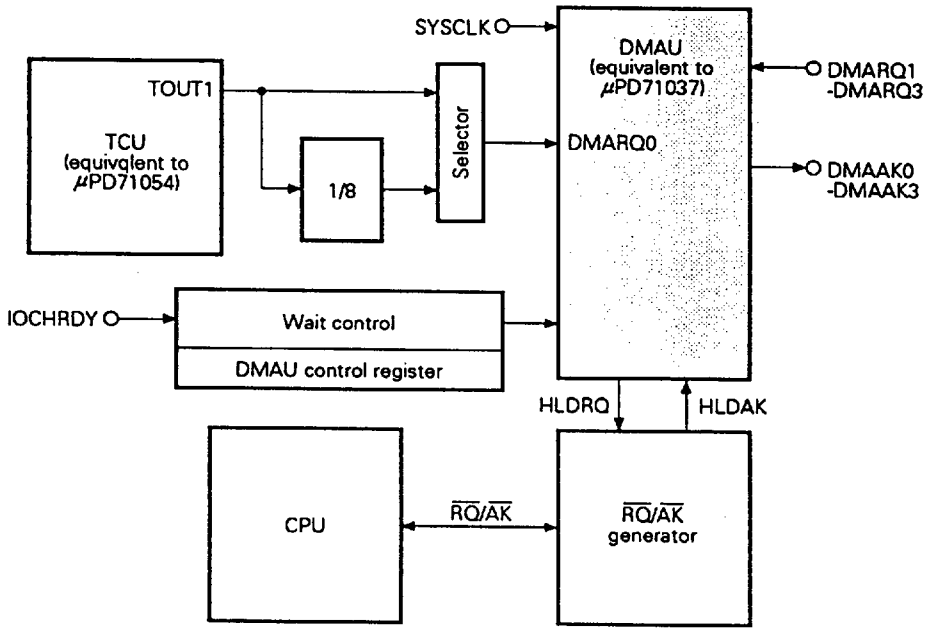


Fig. 3-7 Relationship for DMAU and CPU, and TCU

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 μPD70270 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.

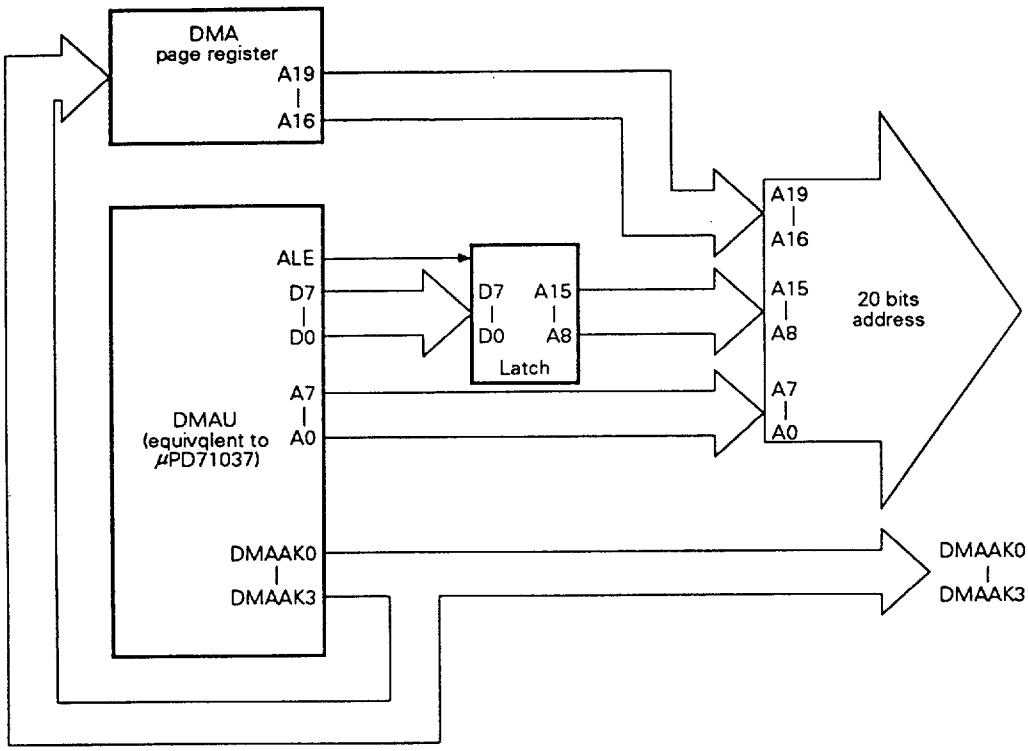


Fig. 3-8 Address Generation during DMA Cycle

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).

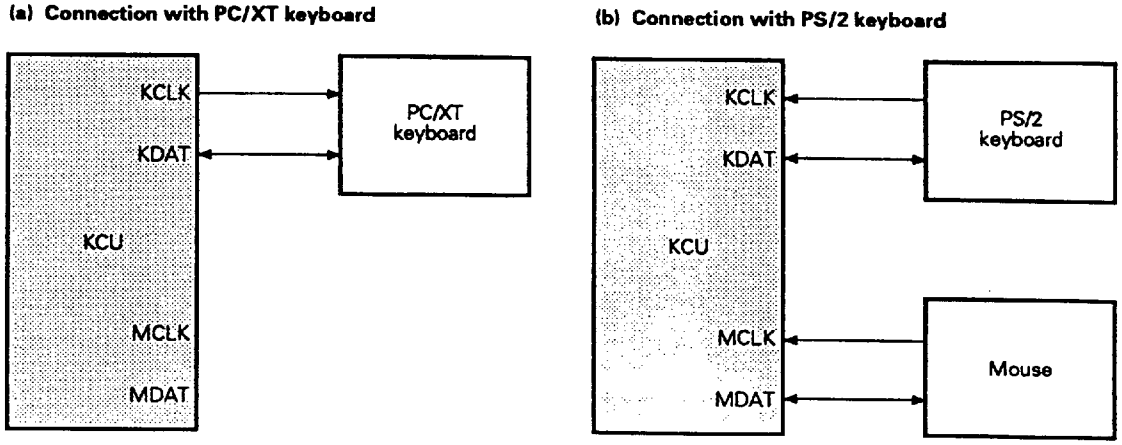


Fig. 3-9 KCU and Keyboard Connection

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 μPD70270 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 μPD70270.

3.15 EMS CIRCUIT

The μPD70270 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, D00000H-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.

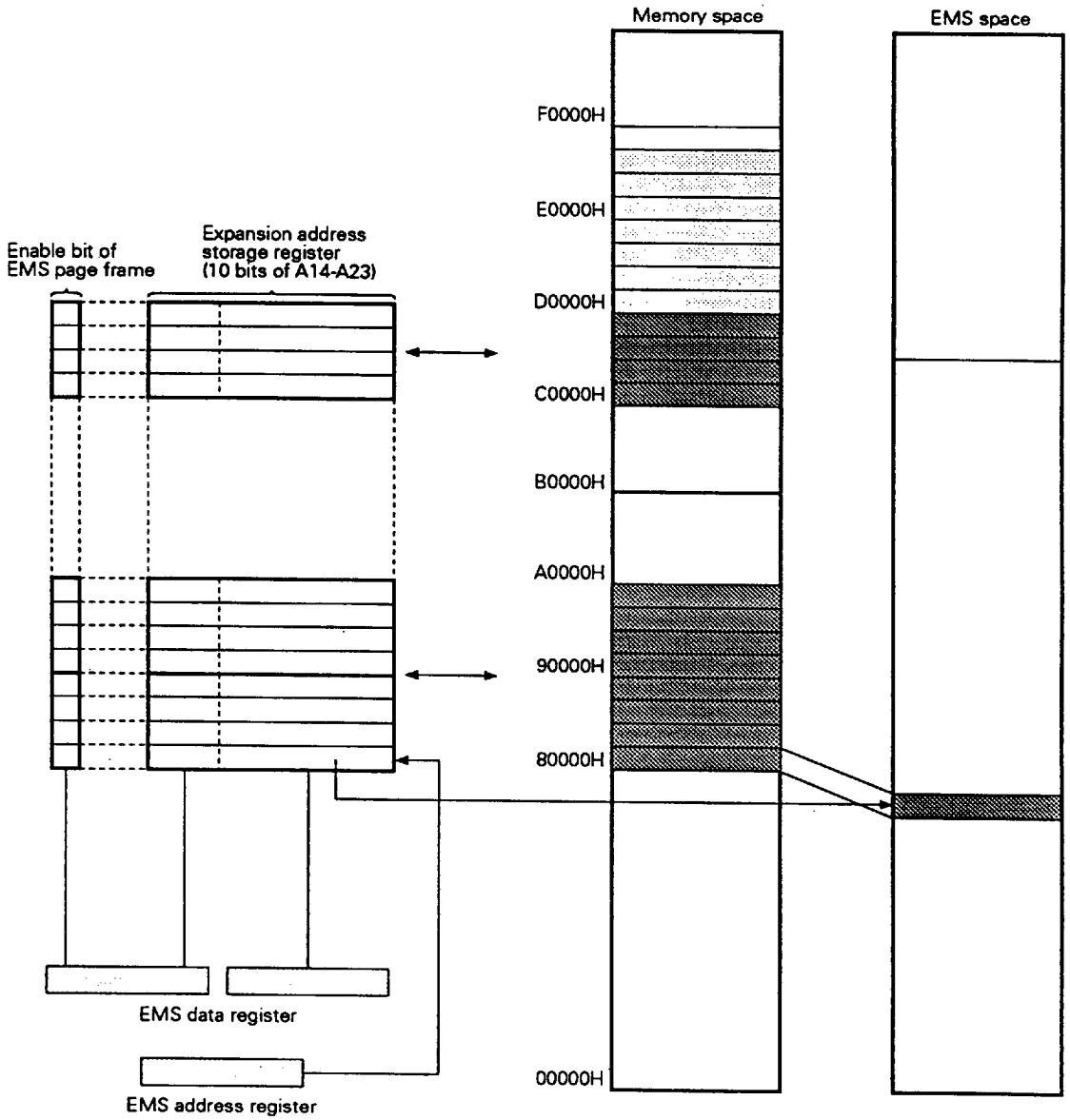
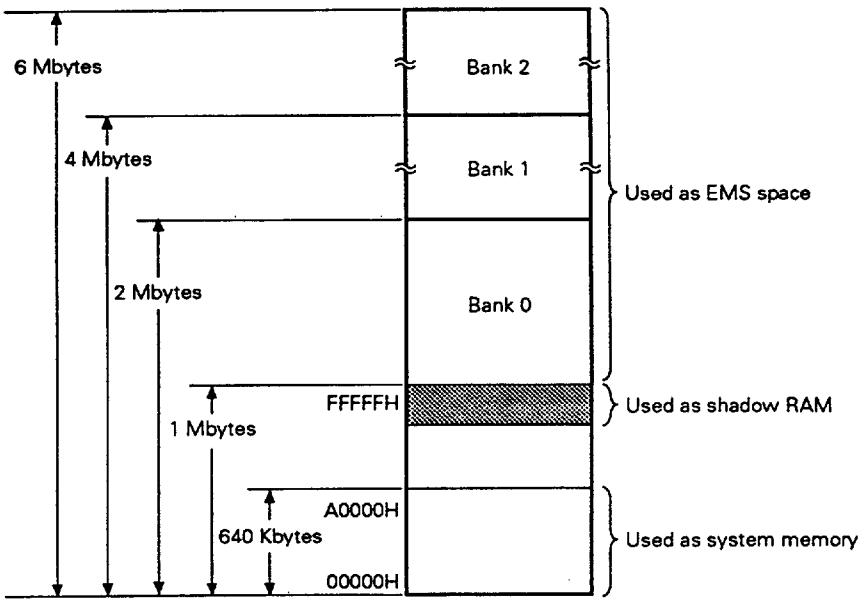


Fig. 3-10 Relations between EMS Address Register and EMS Data Register

(a) With DRAM of 1M words × 4 bits



(b) With DRAM of 4M words × 4 bits

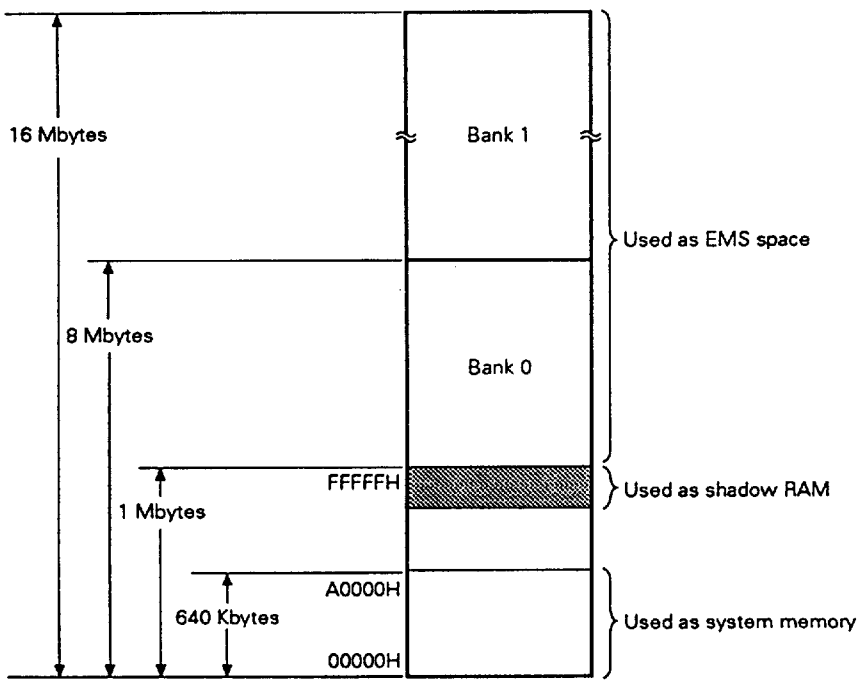


Fig. 3-11 Example of Memory Space Allocation

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 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)	×	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

Instruction	Number of clocks	
	Byte processing (W=0)	Word processing (W=1)
MOVBK	11+8/rep (11)	11+16/rep (19)
CMPBK	7+14/rep (13)	7+22/rep (21)
CMPM	7+10/rep (7)	7+14/rep (11)
LDM	7+9/rep (7)	7+13/rep (11)
STM	7+4/rep (7)	7+8/rep (11)
INM	9+8/rep (10)	9+16/rep (18)
OUTM	9+8/rep (10)	9+16/rep (18)

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

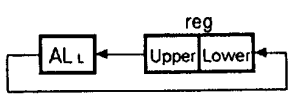
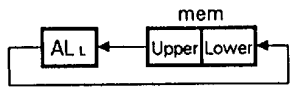
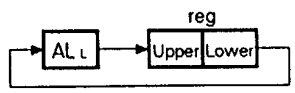
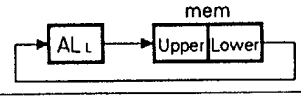
Instruction group	Mnemonic	Operand	Operation code														Number of bytes	Number of clocks	Operation	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				AC	CY	V	P	S	Z
Data transfer instruction	MOV	reg, reg'	1	0	0	0	1	0	1	W	1	1	reg		reg'				2	2	reg→reg'						
		mem, reg	1	0	0	0	1	0	0	W	mod		reg		mem				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) When W=1, AH←(dmem+1), AL←(dmem)						
		dmem,acc	1	0	1	0	0	0	1	W									3	9/13	When W=0, (dmem)←AL When W=1, (dmem+1)←AH, (dmem)←AL						
		sreg, reg16	1	0	0	0	1	1	1	0	1	1	0	sreg		reg			2	2	sreg←reg16 sreg: SS, DS0, DS1						
		sreg,mem16	1	0	0	0	1	1	1	0	mod	0	sreg		mem				2-4	11/15	sreg←(mem16) sreg: SS, DS0, DS1						
		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	26	reg16←(mem32), DS0←(mem32+2)						
		DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod		reg		mem				2-4	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	1	0									1	3	S, Z, x, AC, x, P, x, CY←AH	x	x		x	x	x
	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
Data transfer instruction	TRANS	src-table	1	1	0	1	0	1	1	1									1	9	AL←(BW+AL)						
	XCH	reg, reg'	1	0	0	0	0	1	1	W	1	1	reg			reg'			2	3	reg↔reg'						
		mem, reg reg, mem	1	0	0	0	0	1	1	W	mod		reg			mem			2-4	16/24	(mem)↔reg						
		AW, reg16 reg16, AW	1	0	0	1	0	reg											1	3	AW↔reg16						
Repeat prefix	REPC		0	1	1	0	0	1	0	1									1	2	While CW≠0, executes the primitive block transfer of the successive bytes, and decrements (-1) CW. If any interrupt has been on hold, the interrupt is processed. Exits from loop, when CY≠1.						
	REPNC		0	1	1	0	0	1	0	0									1	2	Same as above. Exits from loop, when CY≠0.						
	REP REPE REPZ		1	1	1	1	0	0	1	1									1	2	While CW≠0, executes the primitive block transfer of the successive bytes, and decrements (-1) CW. If any interrupt has been on hold, the interrupt is processed. Exits from loop, when the primitive block transfer instruction is CMPBK or CMPM and Z≠1.						
	REPNE REPNZ		1	1	1	1	0	0	1	0									1	2	Same as above. Exits from loop, when Z≠0.						

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
Primitive block transfer instruction	MOVBK	dst-block, src-block	1	0	1	0	0	1	0	W									1	See Table 4-8	When W=0, (IY)←(IX) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX-1, IY←IY-1 When W=1, (IY+1,IY)←(IX+1,IX) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX-2, IY←IY-2						
	CMPBK	src-block, dst-block	1	0	1	0	0	1	1	W									1	See Table 4-8	When W=0, (IX) - (IY) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX-1, IY←IY-1 When W=1, (IX+1,IX)-(IY+1,IY) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX-2, IY←IY-2	×	×	×	×	×	×
	CMPM	dst-block	1	0	1	0	1	1	1	W									1	See Table 4-8	When W=0, AL-(IY) DIR=0: IY←IY+1; DIR=1: IY←IY-1 When W=1, AW-(IY+1, IY) DIR=0: IY←IY+2; DIR=1: IY←IY-2	×	×	×	×	×	×
	LDM	src-block	1	0	1	0	1	1	0	W									1	See Table 4-8	When W=0, AL←(IX) DIR=0: IX←IX+1; DIR=1: IX←IX-1 When W=1, AW←(IX+1, IX) DIR=0: IX+2; DIR=1: IX←IX-2						
	STM	dst-block	1	0	1	0	1	0	1	W									1	See Table 4-8	When W=0, (IY)←AL DIR=0: IY←IY+1; DIR=1: IY←IY-1 When W=1, (IY+1, IY)←AW DIR=0: IY←IY+2; DIR=1: IY←IY-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	S	Z
Bit field operation instruction	INS	reg8,reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	3	35-133	16 bit field←AW						
			1	1	reg'			reg																			
		reg8,imm4	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	4	35-133	16-bit field←AW						
			1	1	0	0	0	reg																			
	EXT	reg8,reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	3	34-59	AW←16 bit field						
			1	1	reg'			reg																			
		reg8,imm4	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	4	34-59	AW←16 bit field						
			1	1	0	0	0	reg																			
I/O instruction	IN	acc,imm8	1	1	1	0	0	1	0	W									2	9/13	When W=0, AL←(imm8) When W=1, AH←(imm8+1), AL←(imm8)						
		acc,DW	1	1	1	0	1	1	0	W									1	8/12	When W=0, AL←(DW) When W=1, AH←(DW+1), AL←(DW)						
	OUT	imm8,acc	1	1	1	0	0	1	1	W									2	8/12	When W=0, (imm8)←AL When W=1, (imm8+1)←AH,(imm8)←AL						
		DW,acc	1	1	1	0	1	1	1	W									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	0	W									1	See **Table 4-8**	When W=0, (IY)←(DW) DIR=0: IY←IY+1; DIR = 1: IY←IY-1						
																					When W=1, (IY+1, IY)←(DW+1, DW) DIR=0: IY←IY+2; DIR=1: IY←IY-2						
	OUTM	DW, src-block	0	1	1	0	1	1	1	W									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	S	Z
Addition/Subtraction instruction	ADD	reg,reg'	0	0	0	0	0	0	1	W	1	1	reg				reg'		2	2	reg←reg+reg'	×	×	×	×	×	×
		mem,reg	0	0	0	0	0	0	0	W	mod		reg			mem			2-4	16/24	(mem)←(mem)+reg	×	×	×	×	×	×
		reg,mem	0	0	0	0	0	0	1	W	mod		reg			mem			2-4	11/15	reg←reg+(mem)	×	×	×	×	×	×
		reg,imm	1	0	0	0	0	0	S	W	1	1	0	0	0	reg			3-4	4	reg←reg+imm	×	×	×	×	×	×
		mem,imm	1	0	0	0	0	0	S	W	mod	0	0	0	mem				3-6	18/26	(mem)←(mem)+imm	×	×	×	×	×	×
		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	×	×	×	×	×	×
	ADDC	reg,reg'	0	0	0	1	0	0	1	W	1	1	reg				reg'		2	2	reg←reg+reg'+CY	×	×	×	×	×	×
		mem,reg	0	0	0	1	0	0	0	W	mod		reg			mem			2-4	16/24	(mem)←(mem)+reg+CY	×	×	×	×	×	×
		reg,mem	0	0	0	1	0	0	1	W	mod		reg			mem			2-4	11/15	reg←reg+(mem)+CY	×	×	×	×	×	×
		reg,imm	1	0	0	0	0	0	S	W	1	1	0	1	0	reg			3-4	4	reg←reg+imm+CY	×	×	×	×	×	×
		mem,imm	1	0	0	0	0	0	S	W	mod	0	1	0	mem				3-6	18/26	(mem)←(mem)+imm+CY	×	×	×	×	×	×
		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	×	×	×	×	×	×
	SUB	reg,reg'	0	0	1	0	1	0	1	W	1	1	reg				reg'		2	2	reg←reg-reg'	×	×	×	×	×	×
		mem,reg	0	0	1	0	1	0	0	W	mod		reg			mem			2-4	16/24	(mem)←(mem)-reg	×	×	×	×	×	×
		reg,mem	0	0	1	0	1	0	1	W	mod		reg			mem			2-4	11/15	reg←reg-(mem)	×	×	×	×	×	×
		reg,imm	1	0	0	0	0	0	S	W	1	1	1	0	1	reg			3-4	4	reg←reg-imm	×	×	×	×	×	×
		mem,imm	1	0	0	0	0	0	S	W	mod	1	0	1	mem				3-6	18/26	(mem)←(mem)-imm	×	×	×	×	×	×
		acc,imm	0	0	1	0	1	1	0	W									2-3	4	When W=0, AL←AL+imm+CY When W=1, AW←AW+imm+CY	×	×	×	×	×	×
	SUBC	reg,reg'	0	0	0	1	1	0	1	W	1	1	reg				reg'		2	2	reg←reg-reg'-CY	×	×	×	×	×	×
		mem,reg	0	0	0	1	1	0	0	W	mod		reg			mem			2-4	16/24	(mem)←(mem)-reg-CY	×	×	×	×	×	×
		reg,mem	0	0	0	1	1	0	1	W	mod		reg			mem			2-4	11/15	reg←reg-(mem)-CY	×	×	×	×	×	×
		reg,imm	1	0	0	0	0	0	S	W	1	1	0	1	1	reg			3-4	4	reg←reg-imm-CY	×	×	×	×	×	×
		mem,imm	1	0	0	0	0	0	S	W	mod	0	1	1	mem				3-6	18/26	(mem)←(mem)-imm-CY	×	×	×	×	×	×
		acc,imm	0	0	0	1	1	1	0	W									2-3	4	When AL←AL-imm-CY When W=1, AW←AW-imm-CY	×	×	×	×	×	×

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	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	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	2	19xn+7	dst BCD string - src BCD string*	U x U U U x
	ROL4	reg8	0 0 0 0 1 1 1 1 / 1 1 0 0 0 reg	3	13		
		mem8	0 0 0 0 1 1 1 1 / mod 0 0 0 mem	3-5	28		
	ROR4	reg8	0 0 0 0 1 1 1 1 / 1 1 0 0 0 reg	3	17		
		mem8	0 0 0 0 1 1 1 1 / mod 0 0 0 mem	3-5	32		
Increment/decrement instruction	INC	reg8	1 1 1 1 1 1 1 0 / 1 1 0 0 0 reg	2	2	reg8←reg8+1	x x x x x
		mem	1 1 1 1 1 1 1 W / mod 0 0 0 mem	2-4	16/24	(mem)←(mem)+1	x x x x x
		reg16	0 1 0 0 0 reg	1	2	reg16←reg16+1	x x x x x
	DEC	reg8	1 1 1 1 1 1 1 0 / 1 1 0 0 1 reg	2	2	reg8←reg8-1	x x x x x
		mem	1 1 1 1 1 1 1 W / mod 0 0 1 mem	2-4	16/24	(mem)←(mem)-1	x x x x x
		reg16	0 1 0 0 1 reg	1	2	reg16←reg16-1	x x x x x

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	P	S	Z
Multiplication instruction	MULU	reg8	1	1	1	1	0	1	1	0	1	1	1	0	0	reg			2	21-22	AW←ALxreg8 AH=0: CY←0,V←0 AH≠0: CY←1,V←1	U	x	x	U	U	U
		mem8	1	1	1	1	0	1	1	0	mod1	0	0	mem					2-4	27-28	AW←ALx(mem8) AH=0: CY←0,V←0 AH≠0: CY←1,V←1	U	x	x	U	U	U
		reg16	1	1	1	1	0	1	1	1	1	1	1	0	0	reg			2	29-30	DW, AW←AWxreg16 DW=0: CY←0,V←0 DW≠0: CY←1,V←1	U	x	x	U	U	U
		mem16	1	1	1	1	0	1	1	1	mod1	0	0	mem					2-4	39-40	DW,AW←AWx(mem16) DW=0: CY←0,V←0 DW≠0: CY←1,V←1	U	x	x	U	U	U
	MUL	reg8	1	1	1	1	0	1	1	0	1	1	1	0	1	reg			2	33-39	AW←ALxreg8 AH = Sign extension for AL: CY←0,V←0 AH ≠ Sign extension for AL: CY←1,V←1	U	x	x	U	U	U
		mem8	1	1	1	1	0	1	1	0	mod1	0	1	mem					2-4	39-45	AW←ALx(mem8) AH = Sign extension for AL: CY←0,V←0 AH ≠ Sign extension for AL: CY←1,V←1	U	x	x	U	U	U
		reg16	1	1	1	1	0	1	1	1	1	1	1	0	1	reg			2	41-47	DW,AW←AWxreg16 DW = Sign extension for AW: CY←0,V←0 DW ≠ Sign extension for AW: CY←1,V←1	U	x	x	U	U	U
		mem16	1	1	1	1	0	1	1	1	mod1	0	1	mem					2-4	51-57	DW,AW←AWx(mem16) DW = Sign extension for AW: CY←0,V←0 DW ≠ Sign extension for AW: CY←1,V←1	U	x	x	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
Multiplication instruction	MUL (cont'd)	reg16, (reg16',)* imm8	0	1	1	0	1	0	1	1	1	1	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	mod		reg			mem			3-5	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	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	mod		reg			mem			4-6	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				AC	CY	V	P	S	Z
Unsigned division instruction	DIVU	reg8	1 1 1 1 0 1 1 0	1 1 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
		mem8	1 1 1 1 0 1 1 0	mod 1 1 0 mem	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
		reg16	1 1 1 1 0 1 1 1	1 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	mod 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 1 1 1 reg	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 / mod 1 1 1 mem	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 reg	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 / mod 1 1 1 mem	2-4	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	Z
BCD coprocessor instruction	ADJBA		0	0	1	1	0	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	×	×	U	U	U	U
	ADJ4A		0	0	1	0	0	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	×	×	U	×	×	×
	ADJBS		0	0	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	×	×	U	U	U	U
	ADJ4S		0	0	1	0	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	×	×	U	×	×	×
Data conversion	CVTBD		1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	2	15	AH←AL+0AH, AL←AL%0AH	U	U	U	×	×	×
	CVTDB		1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	2	7	AL←AHxOAH+AL, AH←0	U	U	U	×	×	×
	CVTBW		1	0	0	1	1	0	0	0									1	2	When AL<80H, AH←0. Otherwise, AH←FFH						
	CVTWL		1	0	0	1	1	0	0	1									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	×	×	×	×	×	×
		mem,reg	0	0	1	1	1	0	0	W	mod		reg			mem			2-4	11/15	(mem)-reg	×	×	×	×	×	×
		reg,mem	0	0	1	1	1	0	1	W	mod		reg			mem			2-4	11/15	reg-(mem)	×	×	×	×	×	×
		reg,imm	1	0	0	0	0	0	S	W	1	1	1	1	1	reg			3-4	4	reg-imm	×	×	×	×	×	×
		mem,imm	1	0	0	0	0	0	S	W	mod		1	1	1	mem			3-6	13/17	(mem)-imm	×	×	×	×	×	×
		acc,imm	0	0	1	1	1	1	0	W									2-3	4	When W=0, AL-imm When W=1, AW-imm	×	×	×	×	×	×

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
Complements	NOT	reg	1 1 1 1 0 1 1 W / 1 1 0 1 0 reg	2	2	reg←$\overline{reg}$	
		mem	1 1 1 1 0 1 1 W / mod 0 1 0 mem	2-4	16/24	(mem)←$\overline{(mem)}$	
	NEG	reg	1 1 1 1 0 1 1 W / 1 1 0 1 1 reg	2	2	reg←$\overline{reg}$+1	× × × × × ×
		mem	1 1 1 1 0 1 1 W / mod 0 1 1 mem	2-4	16/24	(mem)←$\overline{(mem)}$+1	× × × × × ×
Logical operation instruction	TEST	reg,reg'	1 0 0 0 0 1 0 W / 1 1 reg' reg	2	2	reg ∧ reg'	U 0 0 × × ×
		mem,reg reg,mem	1 0 0 0 0 1 0 W / mod reg mem	2-4	10/14	(mem) ∧ reg	U 0 0 × × ×
		reg,imm	1 1 1 1 0 1 1 W / 1 1 0 0 0 reg	3-4	4	reg ∧ imm	U 0 0 × × ×
		mem,imm	1 1 1 1 0 1 1 W / mod 0 0 0 mem	3-6	11/15	(mem) ∧ imm	U 0 0 × × ×
		acc,imm	1 0 1 0 1 0 0 W	2-3	4	When W=0, AL ∧ imm8 When W=1, AW ∧ imm16	U 0 0 × × ×
	AND	reg,reg'	0 0 1 0 0 0 1 W / 1 1 reg reg'	2	2	reg←reg ∧ reg'	U 0 0 × × ×
		mem,reg	0 0 1 0 0 0 0 W / mod reg mem	2-4	16/24	(mem)←(mem) ∧ reg	U 0 0 × × ×
		reg,mem	0 0 1 0 0 0 1 W / mod reg mem	2-4	11/15	reg←reg ∧ (mem)	U 0 0 × × ×
		reg,imm	1 0 0 0 0 0 0 W / 1 1 1 0 0 reg	3-4	4	reg←reg ∧ imm	U 0 0 × × ×
		mem,imm	1 0 0 0 0 0 0 W / mod 1 0 0 mem	3-6	18/26	(mem)←(mem) ∧ imm	U 0 0 × × ×
		acc,imm	0 0 1 0 0 1 0 W	2-3	4	When W=0, AL←AL ∧ imm8 When W=1, AW←AW imm16	U 0 0 × × ×
	OR	reg,reg'	0 0 0 0 1 0 1 W / 1 1 reg reg'	2	2	reg←reg reg'	U 0 0 × × ×
		mem,reg	0 0 0 0 1 0 0 W / mod reg mem	2-4	16/24	(mem)←(mem) reg	U 0 0 × × ×
		reg,mem	0 0 0 0 1 0 1 W / mod reg mem	2-4	11/15	reg←reg (mem)	U 0 0 × × ×
		reg,imm	1 0 0 0 0 0 0 W / 1 1 0 0 1 reg	3-4	4	reg←reg imm	U 0 0 × × ×
		mem,imm	1 0 0 0 0 0 0 W / mod 0 0 1 mem	3-6	18/26	(mem)←(mem) imm	U 0 0 × × ×
		acc,imm	0 0 0 0 1 1 0 W	2-3	4	When W=0, AL←AL imm8 When W=1, AW←AW imm16	U 0 0 × × ×

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
Logical operation instruction	XOR	reg,reg'	0	0	1	1	0	0	1	W	1	1	reg			reg'			2	2	reg←reg ∀ reg'	U	0	0	×	×	×
		mem,reg	0	0	1	1	0	0	0	W	mod		reg			mem			2-4	16/24	(mem)←(mem) ∀ reg	U	0	0	×	×	×
		reg,mem	0	0	1	1	0	0	1	W	mod		reg			mem			2-4	11/15	reg←reg ∀ (mem)	U	0	0	×	×	×
		reg,imm	1	0	0	0	0	0	0	W	1	1	1	1	0	reg			3-4	4	reg←reg ∀ imm	U	0	0	×	×	×
		mem,imm	1	0	0	0	0	0	0	W	mod	1	1	0	mem				3-6	18/26	(mem)←(mem) ∀ imm	U	0	0	×	×	×
		acc,imm	0	0	1	1	0	1	0	W									2-3	4	When W=0, AL←AL ∀ imm8 When W=1, AW←AW ∀ imm16	U	0	0	×	×	×

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	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	13	(mem16) bit NO.imm4=0 : Z←1 (mem16) bit NO.imm4=1 : Z←0	U 0 0 U U x
	NOT1	reg8,CL	0 1 1 0 / 1 1 0 0 0 reg	3	4	reg8 bit NO.CL←$\overline{\text{reg8 bit NO.CL}}$	
		mem8,CL	0 1 1 0 / mod 0 0 0 mem	3-5	13	(mem8) bit NO.CL←$\overline{\text{(mem8) bit NO.CL}}$	
		reg16,CL	0 1 1 1 / 1 1 0 0 0 reg	3	4	reg16 bit NO.CL←$\overline{\text{reg16 bit NO.CL}}$	
		mem16,CL	0 1 1 1 / mod 0 0 0 mem	3-5	21	(mem16) bit NO.CL←$\overline{\text{(mem16) bit NO.CL}}$	
		reg8,imm3	1 1 1 0 / 1 1 0 0 0 reg	4	5	reg8 bit NO.imm3←$\overline{\text{reg8 bit NO.imm3}}$	
		mem8,imm3	1 1 1 0 / mod 0 0 0 mem	4-6	14	(mem8) bit NO.imm3←$\overline{\text{(mem8) bit NO.imm3}}$	
		reg16,imm4	1 1 1 1 / 1 1 0 0 0 reg	4	5	reg16 bit NO.imm4←$\overline{\text{reg16 bit NO.imm4}}$	
		mem16,imm4	1 1 1 1 / mod 0 0 0 mem	4-6	22	(mem16) bit NO.imm4←$\overline{\text{(mem16) bit NO.imm4}}$	
			2nd byte* / 3rd byte*			✱ : 1st byte = 0FH	
	NOT1	CY	1 1 1 1 0 1 0 1	1	2	CY←$\overline{\text{CY}}$	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	1	0	0	0	reg			3	5	reg8 bit NO.CL←0						
		mem8,CL					0	0	1	0	mod	0	0	0	mem				3-5	14	(mem8) bit NO.CL←0						
		reg16,CL					0	0	1	1	1	1	0	0	0	reg			3	5	reg16 bit NO.CL←0						
		mem16,CL					0	0	1	1	mod	0	0	0	mem				3-5	22	(mem16) bit NO.CL←0						
		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	mem				4-6	15	(mem8) bit NO.imm3←0						
		reg16,imm4					1	0	1	1	1	1	0	0	0	reg			4	6	reg16 bit NO.imm4←0						
		mem16,imm4					1	0	1	1	mod	0	0	0	mem				4-6	23	(mem16) bit NO.imm4←0						
	SET1	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	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	mem				3-5	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	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	mem				4-6	22	(mem16) bit NO.imm4←1						

2nd byte* 3rd byte*

* : 1st byte = 0FH

	CLR1	CY	1	1	1	1	1	0	0	0									1	2	CY←0		0				
		DIR	1	1	1	1	1	1	0	0									1	2	DIR←0						
	SET1	CY	1	1	1	1	1	0	0	1									1	2	CY←1		1				
		DIR	1	1	1	1	1	1	0	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	Z
Shift instruction	SHL	reg,1	1	1	0	1	0	0	0	W	1	1	1	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
		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
	SHR	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: CY, V←0	U	x	x	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
Shift instruction	SHR (cont'd)	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←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	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	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
	SHRA	reg,1	1	1	0	1	0	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
		mem,1	1	1	0	1	0	0	0	W	mod	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	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	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	0	W	mod	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	reg			2	6	CY←MSB of reg, reg←regx2+CY 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	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	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	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	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	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			
	ROR	reg,1	1	1	0	1	0	0	0	W	1	1	0	0	1	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	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	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	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	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	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 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	ROLC (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 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 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
	RORC	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
		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 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	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	0	W	1	1	0	1	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 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
Subroutine control instruction	CALL	near-proc	1	1	1	0	1	0	0	0									3	20	SP←SP-2, (SP+1,SP)←PC PC←PC+disp						
		regptr16	1	1	1	1	1	1	1	1	1	1	0	1	0	reg			2	18	SP←SP-2, (SP+1,SP)←PC PC←regptr16						
		memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	mem				2-4	31	TA←(memptr16) SP←SP-2, (SP+1,SP)←PC, PC←TA						
		far-proc	1	0	0	1	1	0	1	0									5	29	SP←SP-2, (SP+1,SP)←PS, PS←seg SP←SP-2, (SP+1,SP)←PC, PC←offset						
		memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	mem				2-4	47	TA←(memptr32), TB←(memptr32+2) SP←SP-2, (SP+1,SP)←PS, PS←TB SP←SP-2, (SP+1,SP)←PC, PC←TA						
	RET		1	1	0	0	0	0	1	1									1	19	PC←(SP+1,SP) SP←SP+2						
		pop-value	1	1	0	0	0	0	1	0									3	24	PC←(SP+1,SP) SP←SP+2, SP←SP+pop-value						
			1	1	0	0	1	0	1	1									1	29	PC←(SP+1,SP) PS←(SP+3,SP+2) SP←SP+4						
		pop-value	1	1	0	0	1	0	1	0									3	32	PC←(SP+1,SP) PS←(SP+3,SP+2) SP←SP+4, SP←SP+pop-value						

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
Stack operation instruction	PUSH	mem16	1 1 1 1 1 1 1 1	mod 1 1 0 mem	2-4	26	SP←SP-2 (SP+1,SP)←(mem16)						
		reg16	0 1 0 1 0 reg		1	12	SP←SP-2 (SP+1,SP)←reg16						
		sreg	0 0 0 sreg 1 1 0		1	12	SP←SP-2 (SP+1,SP)←sreg						
		PSW	1 0 0 1 1 1 0 0		1	12	SP←SP-2 (SP+1,SP)←PSW						
		R	0 1 1 0 0 0 0 0		1	67	Push registers on the stack						
		imm	0 1 1 0 1 0 S 0		2-3	11 or 12	SP←SP-2 (SP+1,SP)←imm sign expansion when S=1						
	POP	mem16	1 0 0 0 1 1 1 1	mod 0 0 0 mem	2-4	25	(mem16)←(SP+1,SP) SP←SP+2						
		reg16	0 1 0 1 1 reg		1	12	reg16←(SP+1,SP) SP←SP+2						
		sreg	0 0 0 sreg 1 1 1		1	12	sreg←(SP+1,SP) sreg: SS,DS0,DS1 SP←SP+2						
		PSW	1 0 0 1 1 1 0 1		1	12	PSW←(SP+1,SP) SP←SP+2	R	R	R	R	R	R
		R	0 1 1 0 0 0 0 1		1	75	Pop registers from the stack						
	PREPARE	imm16,imm8	1 1 0 0 1 0 0 0		4	*	Prepare New Stack Frame						
	DISPOSE		1 1 0 0 1 0 0 1		1	10	Dispose of Stack Frame						

*: When imm8 = 0, 16

When imm8 ≥ 1, 21+16 (imm8-1)

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 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	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	35	PS←(memptr32+2) PC←(memptr32)		
Condition branch instruction	BV	short-label	0 1 1 1 0 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 BL	short-label	0 0 1 0	2	14/4	if CY=1	PC←PC+ext-disp8	
	BNC BNL	short-label	0 0 1 1	2	14/4	if CY=0	PC←PC+ext-disp8	
	BE BZ	short-label	0 1 0 0	2	14/4	if Z=1	PC←PC+ext-disp8	
	BNE BNZ	short-label	0 1 0 1	2	14/4	if Z=0	PC←PC+ext-disp8	
	BNH	short-label	0 1 1 0	2	14/4	if CY ∨ Z=1	PC←PC+ext-disp8	
	BH	short-label	0 1 1 1	2	14/4	if CY ∨ 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=1	PC←PC+ext-disp8	
	BGE	short-label	1 1 0 1	2	14/4	if S ∀ V=0	PC←PC+ext-disp8	
	BLE	short-label	1 1 1 0	2	14/4	if (S ∀ V) ∨ Z=1	PC←PC+ext-disp8	
	BGT	short-label	1 1 1 1	2	14/4	if (S ∀ 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 Z
Condition branch instruction	DBNZNE	short-label	1 1 1 0 0 0 0 0	2	14/5	CW=CW-1 PC←PC+ext-disp8 if Z=0 and CW≠0	
	DBNZE	short-label	0 0 0 1	2	14/5	CW=CW-1 PC←PC+ext-disp8 if Z=1 and CW≠0	
	DBNZ	short-label	0 0 1 0	2	13/5	CW=CW-1 PC←PC+ext-disp8 if CW≠0	
	BCWZ	short-label	0 0 1 1	2	13/5	if CW=0 PC←PC+ext-disp8	
Interrupt instruction	BRK	3	1 1 0 0 1 1 0 0	1	50	TA←(00DH,00CH), TC←(00FH,00EH) 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	
		imm8 (≠3)	1 1 0 0 1 1 0 1	2	50	TA←(4n+1,4n), TC←(4n+3,4n+2) n=imm8 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	
	BRKV		1 1 0 0 1 1 1 0	1	52/3	When V=1, TA←(011H,010H), TC←(013H,012H) 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	
	RETI		1 1 0 0 1 1 1 1	1	39	PC←(SP+1,SP), PS←(SP+3,SP+2), PSW←(SP+5,SP+4), SP←SP+6	R R R R R R

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
Interrupt instruction	BRKEM	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	3	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	73-76 /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						
CPU control instruction	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 $\overline{\text{POLL}}$ pin sampling operations						
	DI		1	1	1	1	1	0	1	0									1	2	IE←0						
	EI		1	1	1	1	1	0	1	1									1	2	IE←1						
	BUSLOCK		1	1	1	1	0	0	0	0									1	2	Bus Lock Prefix						
	FPO1	fp-op	1	1	0	1	1	X	X	X	1	1	Y	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	15	data bus←(mem)						
	FP02	fp-op	0	1	1	0	0	1	1	X	1	1	Y	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	15	data bus←(mem)						
	NOP		1	0	0	1	0	0	0	0									1	3	No Operation						
	*		0	0	1	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	7	6	5	4	3	2	1	0				AC	CY	V	P	S	Z
80	RETEM		1	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	2	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
80	CALLN	imm8	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	3	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} = 5 V±10%

Absolute Maximum Rating (T_a = 25°C)

Parameter	Symbol	Conditions	Rating	Units
Power supply	V_{DD}		−0.5 to +7.0	V
Input voltage	V_I	V_{DD} = 5 V±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°C, V_{DD} = 5 V+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} = −100μA		0.7V_{DD}			V
Low-level output voltage	V_{OL}	I_{OL} = 1.6mA				0.4	V
High-level input leakage current	I_{LIH}	V_I = V_{DD}				10	μA
Low-level input leakage current	I_{LIL}	V_I = 0V				−10	μA
High-level output leakage current	I_{LOH}	V_O = V_{DD}				10	μA
Low-level output leakage current	I_{LOL}	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-AD7, SD0-SD7, DMARQ1-DMARQ3, INTP2-INTP7, ADBIOS, HDINS, KEYLOCK, TCLK.

*2 : Pins other than the above

Capacitance (T_a = 25°C, V_{DD} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Units
Input capacitance	C_I	fc=1MHz			15	pF
I/O capacitance	C_{IO}	Other than measurement pins : 0 V			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↑→$\overline{WR}$ active delay	30	t_{DXWRL}			40	ns
CPUCLK↓→$\overline{WR}$ inactive delay	31	t_{DCKWRH}			35	ns
X1↑→$\overline{RAS}$ 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↓→$\overline{RAS}$ inactive delay	33	$t_{DCKRASH}$			25	ns
X1↑→$\overline{CAS}$ active delay	34	t_{DXCASL}			25	ns
CPUCLK↑→$\overline{CAS}$ active delay	35	$t_{DCKCASL}$			25	ns
X1↓→$\overline{CAS}$ inactive delay	36	t_{DXCASH}			25	ns
CPUCLK↓→$\overline{CAS}$ inactive delay	37	$t_{DCKCASH}$		5	25	ns
CPUCLK↓→AD data output delay	38	t_{DCKD}		5	30	ns
CPUCLK↑→AD data floating delay	39	t_{FCKD}		5	30	ns
AD data input set (for CPUCLK↓)	40	t_{SDCK}		20		ns
AD data input hold (for CPUCLK↓)	41	t_{HCKD}		10		ns
CPUCLK↑→BS↓delay	42	t_{DKBL}		0	55	ns
CPUCLK↓→BS↑delay	43	t_{DKBH}		0	55	ns
SYSCLK→MA address delay	44	t_{DSKMA}			25	ns
SYSCLK→MA address hold	45	t_{HSKMA}		0		ns
SYSCLK↑→$\overline{WR}$↓delay	46	t_{DSKWRL}			25	ns
SYSCLK↑→$\overline{WR}$↑delay	47	t_{DSKWRH}			40	ns
SYSCLK↑→$\overline{RAS}$↓delay	48	$t_{DSKRASL}$			25	ns
SYSCLK↑→$\overline{RAS}$↑delay	49	$t_{DSKRASH}$			40	ns
SYSCLK↑→$\overline{CAS}$↓delay	50	$t_{DSKCASL}$			25	ns
SYSCLK↑→$\overline{CAS}$↑delay	51	$t_{DSKCASH}$			40	ns
AD data set (for SYSCLK↑)	52	t_{SDSK}		15		ns
AD data hold (for SYSCLK↑)	53	t_{HSKD}		10		ns
SYSCLK↑→AD data output delay	54	t_{DSKDO}		10		ns
SYSCLK↓→AD data valid output delay	55	t_{DSKD}		15		ns
SYSCLK↑→AD data floating delay	56	t_{FSKD}		10		ns
CPUCLK↓→SA address delay	57	t_{DCKSA}			25	ns
CPUCLK↓→SA address hold	58	t_{HCKSA}		0		ns
CPUCLK→control output delay	59	t_{DCKCT}			25	ns
X1↑→control output delay	60	t_{DXCT}			25	ns
SYSCLK↑→SA address delay	61	t_{DSKSA}			25	ns
SYSCLK↑→SA address hold	62	t_{HSKSA}		0		ns
SYSCLK↓→SA address delay	63	t_{DSKSA}			30	ns
SYSCLK↓→SA address hold	64	t_{HSKSA}		0		ns
SYSCLK↓→PCS delay	65	t_{DSKCS}			35	ns
SYSCLK↓→PCS hold	66	t_{HSKCS}		0		ns
Address set (for ALE↓)	67	t_{DSAST}		t_{SKSKL}-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_{DSKDT2}			40	ns
SD data set (for SYSCLK↓)	73	t_{SSDSK}		30		ns
SD data hold (for $\overline{SMRD}$, $\overline{SIORD}$↑)	74	t_{HRDSD}		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↓→$\overline{DMAAK}$n output delay	82	t_{DSKLDA}		5	55	ns
SYSCLK↑→TC active output delay	83	t_{DSKTCH}			35	ns
SYSCLK↑→TC inactive output delay	84	t_{DSKTCL}			35	ns
TC width, high	85	t_{DTCTC}		t_{SKSKL}-20		ns
NMIIN set (for CPUCLK↓)	86	t_{SNICK}		20		ns
NMIIN hold (for CPUCLK↓)	87	t_{HCKNI}		20		ns
NMIOUT output delay (for CPUCLK↓)	88	t_{DCKNO}			25	ns
$\overline{IOCHCK}$→NMIOUT output delay	89	t_{DCINI}			40	ns
$\overline{IOCHCK}$ set (for CPUCLK)	90	t_{SICK}		20		ns
INTPn width, low	91	t_{IPIPL}			80	ns
KCLK input cycle	92	t_{CYKK}		100		ns
KCLK input width, low	93	t_{TKKKKL}		45		ns
KCLK input width, high	94	t_{TKKKKH}		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_{DKKKD}			35	ns

*1 : $\overline{SMRD}$, $\overline{SMWR}$, $\overline{SIORD}$, and $\overline{SIOWR}$ 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)	t_{RC}		4(1+n)t_{CYCK}-25		ns
$\overline{RAS}$ access	(202)	t_{RAC}			2.25(1+n)t_{CYCK}-35	ns
$\overline{CAS}$ access	(203)	t_{CAC}			1.25(1+n)t_{CYCK}-35	ns
Access from column address	(204)	t_{AA}			1.75(1+n)t_{CYCK}-35	ns
Output buffer turn-off delay	(205)	t_{OFF}			t_{CYCK}-25	ns
$\overline{RAS}$ precharge	(206)	t_{RP}		1.75t_{CYCK}-20		ns
$\overline{RAS}$ pulse width (when random read/write cycle)	(207)	t_{RAS}		2.25(1+n)t_{CYCK}-25		ns
$\overline{RAS}$ hold	(208)	t_{RSH}		(1+n)t_{CYCK}-25		ns
$\overline{CAS}$ pulse width	(209)	t_{CAS}		1.25(1+n)t_{CYCK}-25		ns
$\overline{CAS}$ hold	(210)	t_{CSH}		2.25(1+n)t_{CYCK}-25		ns
$\overline{RAS}$-$\overline{CAS}$ delay width	(211)	t_{RCD}			t_{CYCK}-25	ns
$\overline{CAS}$-$\overline{RAS}$ precharge	(212)	t_{CRP}		1.25t_{CYCK}-45		ns
$\overline{CAS}$ precharge	(213)	t_{CPN}		2.25t_{CYCK}-45		ns
Low address set-up	(214)	t_{ASR}		0		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}		2.5(1+n)t_{CYCK}-25		ns
Column address hold for $\overline{RAS}$	(218)	t_{AR}		3.75(1+n)t_{CYCK}-25		ns
Column address delay for $\overline{RAS}$	(219)	t_{RAD}			0.5t_{CYCK}-25	ns
Column address read for $\overline{RAS}$	(220)	t_{RAL}		1.75(1+n)t_{CYCK}-25		ns
Read command set-up	(221)	t_{RCS}		2.75t_{CYCK}-55		ns
Read command hold for $\overline{RAS}$	(222)	t_{RRH}		1.75t_{CYCK}-45		ns
Read command hold	(223)	t_{RCH}		1.75t_{CYCK}-45		ns
Write command hold	(224)	t_{WCH}		(1+n)t_{CYCK}-25		ns
Write command hold for $\overline{RAS}$	(225)	t_{WCR}		2.25(1+n)t_{CYCK}-25		ns
Write command pulse width	(226)	t_{WP}		2.25(1+n)t_{CYCK}-40		ns
Data input set-up	(227)	t_{DS}		t_{CYCK}-30		ns
Data input hold	(228)	t_{DH}		2(1+n)t_{CYCK}-25		ns
Data input hold for $\overline{RAS}$	(229)	t_{DHR}		3.25(1+n)t_{CYCK}-25		ns
$\overline{WE}$ command set-up	(230)	t_{WCS}		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}		$3.5(1+n)t_{CYSK}-25$		ns
$\overline{RAS}$ access	202	t_{RAC}			$2(1+n)t_{CYSK}-25$	ns
$\overline{CAS}$ access	203	t_{CAC}			$(1+n)t_{CYSK}-35$	ns
Access from column address	204	t_{AA}			$1.5(1+n)t_{CYSK}-35$	ns
Output buffer turn-off delay	205	t_{OFF}			$1.5t_{CYSK}$	ns
$\overline{RAS}$ precharge	206	t_{RP}		$1.5t_{CYSK}$		ns
$\overline{RAS}$ pulse width (when random read/write cycle)	207	t_{RAS}		$2(1+n)t_{CYSK}-25$		ns
$\overline{RAS}$ hold	208	t_{RSH}	read cycle	$(1+n)t_{CYSK}-25$		ns
			write cycle	$t_{CYSK}-25$		
$\overline{CAS}$ pulse width	209	t_{CAS}	read cycle	$(1+n)t_{CYSK}-25$		ns
			write cycle	$t_{CYSK}-25$		
$\overline{CAS}$ hold	210	t_{CSH}		$2(1+n)t_{CYSK}-25$		ns
$\overline{RAS}$-$\overline{CAS}$ delay width	211	t_{RCD}			$t_{CYSK}-25$	ns
$\overline{CAS}$-$\overline{RAS}$ precharge	212	t_{CRP}		$2t_{CYSK}-40$		ns
$\overline{CAS}$ precharge	213	t_{CPN}		$1.5t_{CYSK}$		ns
Low address set-up	214	t_{ASR}		$t_{CYCK}-25$		ns
Low address hold	215	t_{RAH}		$0.5t_{CYSK}-15$		ns
Column address set-up	216	t_{ASC}		$0.5t_{CYSK}-25$		ns
Column address hold	217	t_{CAH}		$2(1+n)t_{CYSK}-25$		ns
Column address hold for $\overline{RAS}$	218	t_{AR}		$3(1+n)t_{CYSK}-25$		ns
Column address delay for $\overline{RAS}$	219	t_{RAD}			$0.5t_{CYSK}+25$	ns
Column address read for $\overline{RAS}$	220	t_{RAL}		$1.5(1+n)t_{CYSK}-25$		ns
Read command set-up	221	t_{RCS}		$3t_{CYSK}-40$		ns
Read command hold for $\overline{RAS}$	222	t_{RRH}		$1.5t_{CYSK}$		ns
Read command hold	223	t_{RCH}		$1.5t_{CYSK}$		ns
Write command hold	224	t_{WCH}		$t_{CYSK}-25$		ns
Write command hold for $\overline{RAS}$	225	t_{WCR}		$2(1+n)t_{CYSK}-25$		ns
Write command pulse width	226	t_{WP}		$2(1+n)t_{CYSK}-25$		ns
Data input set-up	227	t_{DS}		$(1+n)t_{CYSK}-30$		ns
Data input hold	228	t_{DH}		$2t_{CYSK}-25$		ns
Data input hold for $\overline{RAS}$	229	t_{DHR}		$3(1+n)t_{CYSK}-25$		ns
$\overline{WE}$ command set-up	230	t_{WCS}		$(1+n)t_{CYSK}-25$		ns

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

(4) DRAM Access Timing (Refresh Cycle)

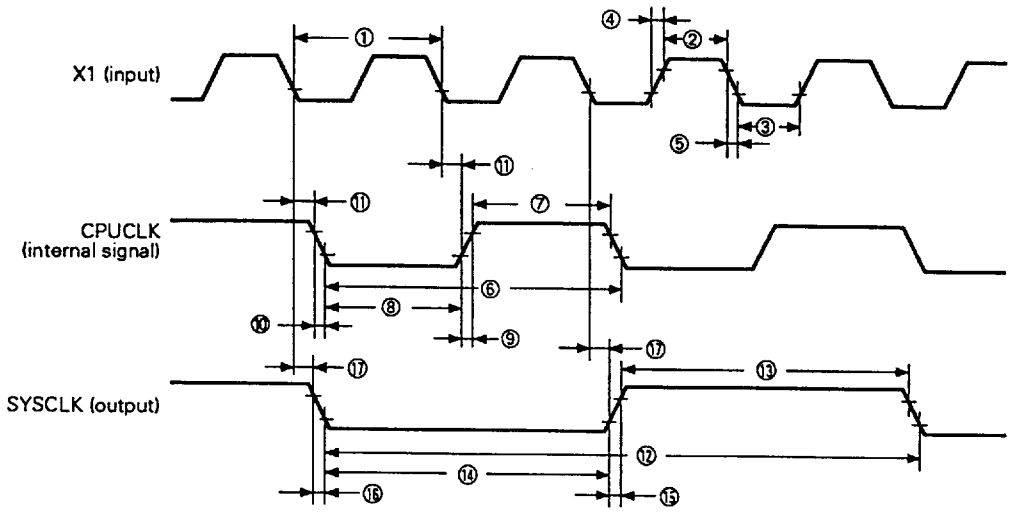
Parameter	Symbol		Conditions	MIN.	MAX.	Units
Random read/write cycle	(301)	t_{RC}		3.5(1+n)t_{CYSK}-25		ns
$\overline{RAS}$ precharge	(302)	t_{RP}		2t_{CYSK}-40		ns
$\overline{RAS}$ pulse width (when random read/write cycle time)	(303)	t_{RAS}		2(1+n)t_{CYSK}-25		ns
$\overline{RAS}$ precharge/$\overline{CAS}$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

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

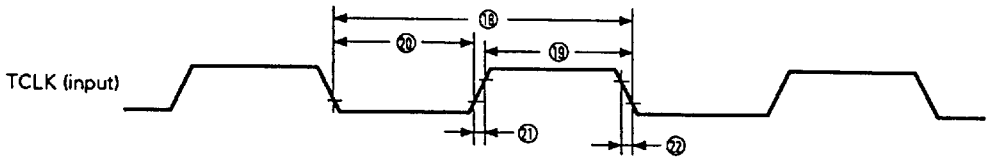
5.2 WHEN SUPPLY VOLTAGE V_{DD} = 3 V±10%

Under evaluation

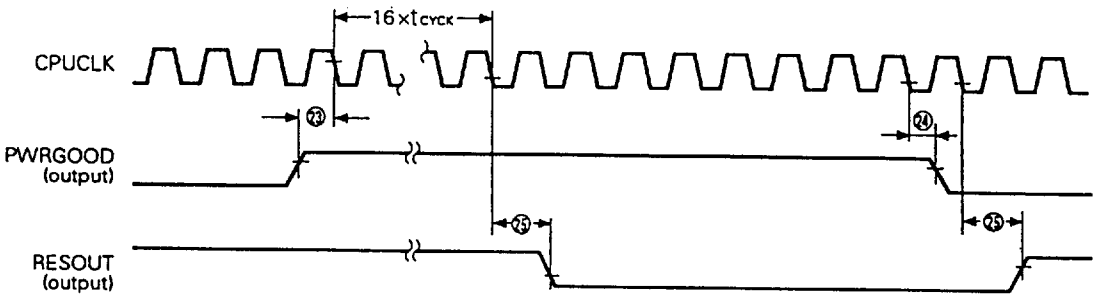
Clock Timing



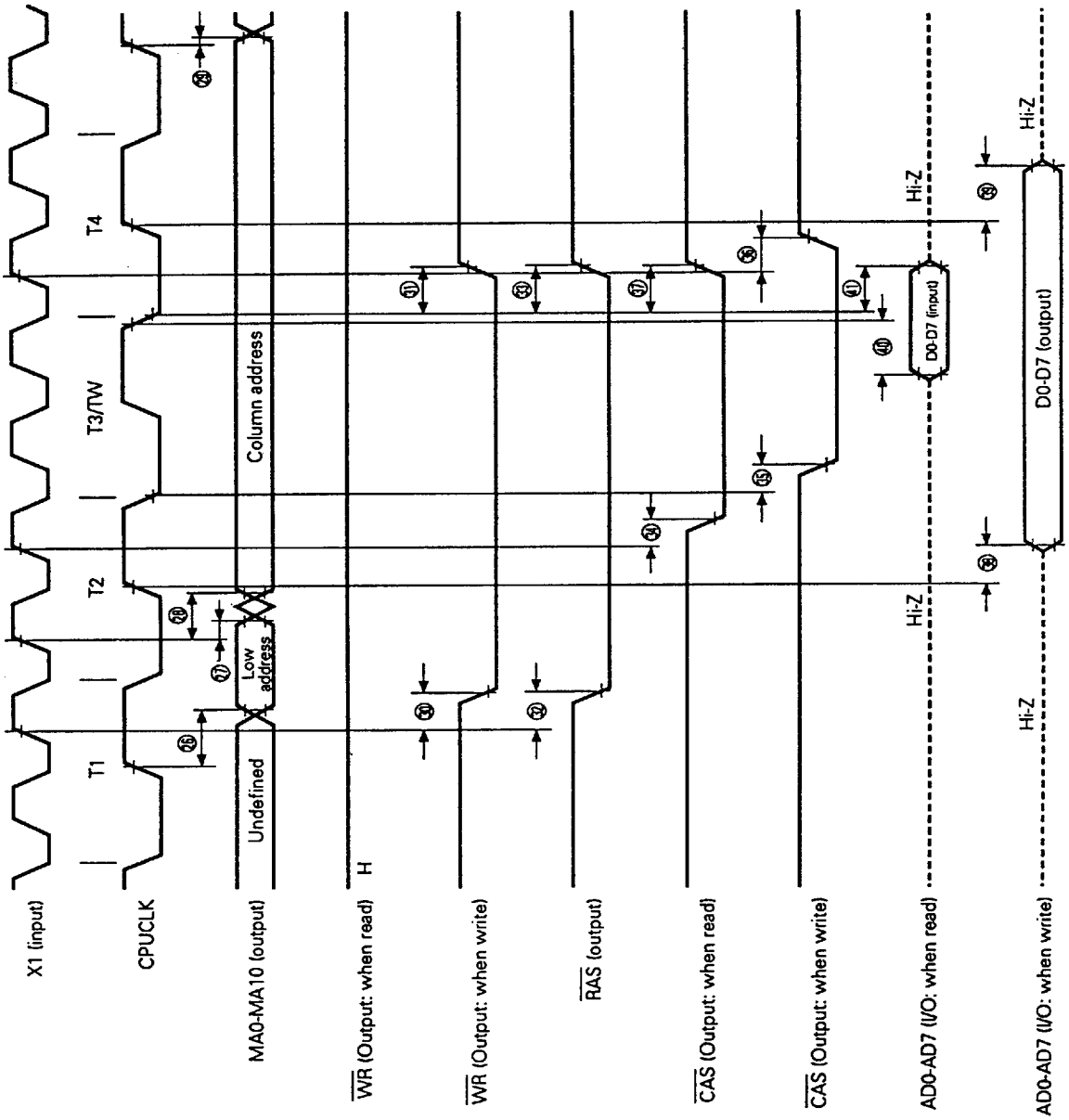
Timer Clock Timing



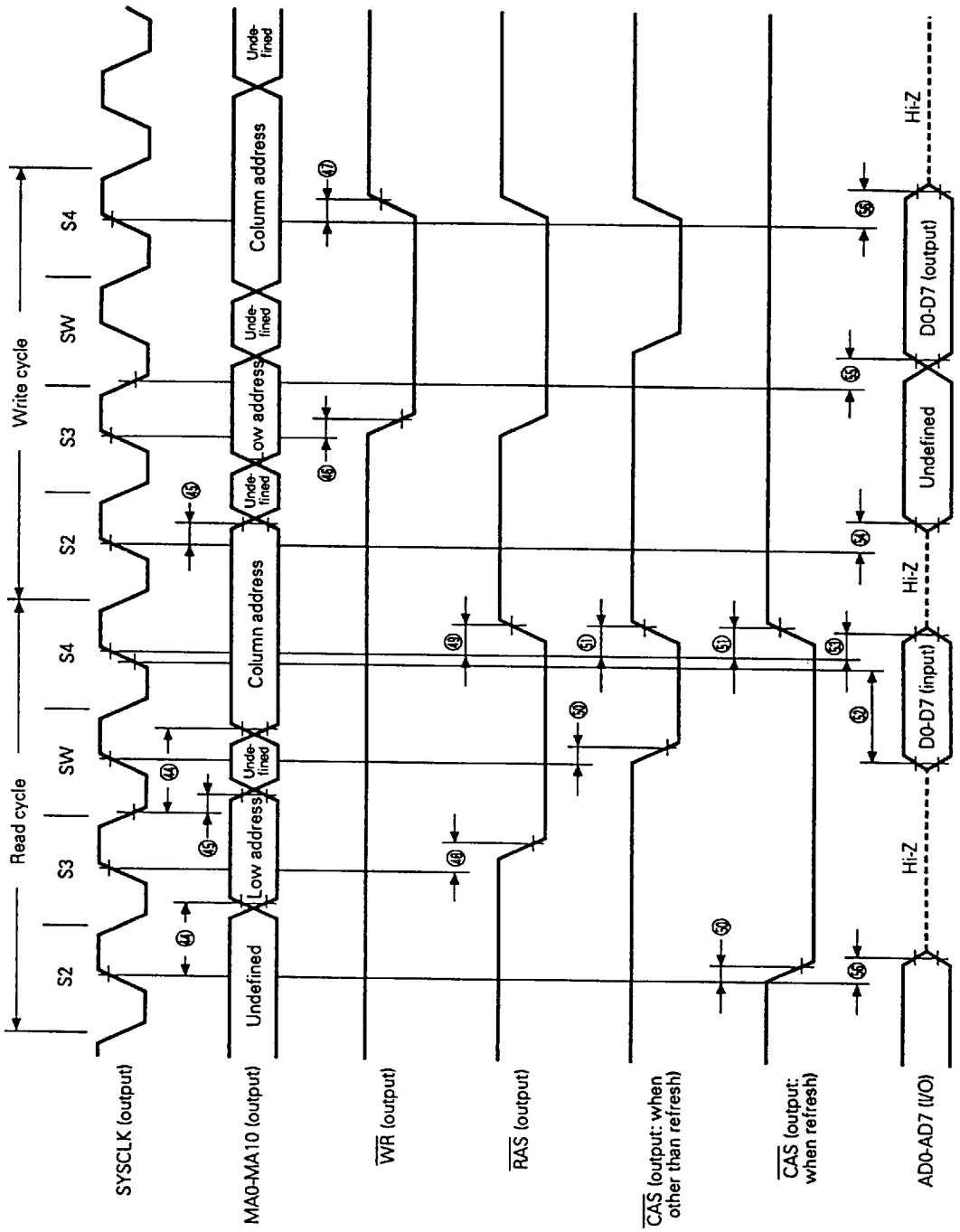
Reset Timing



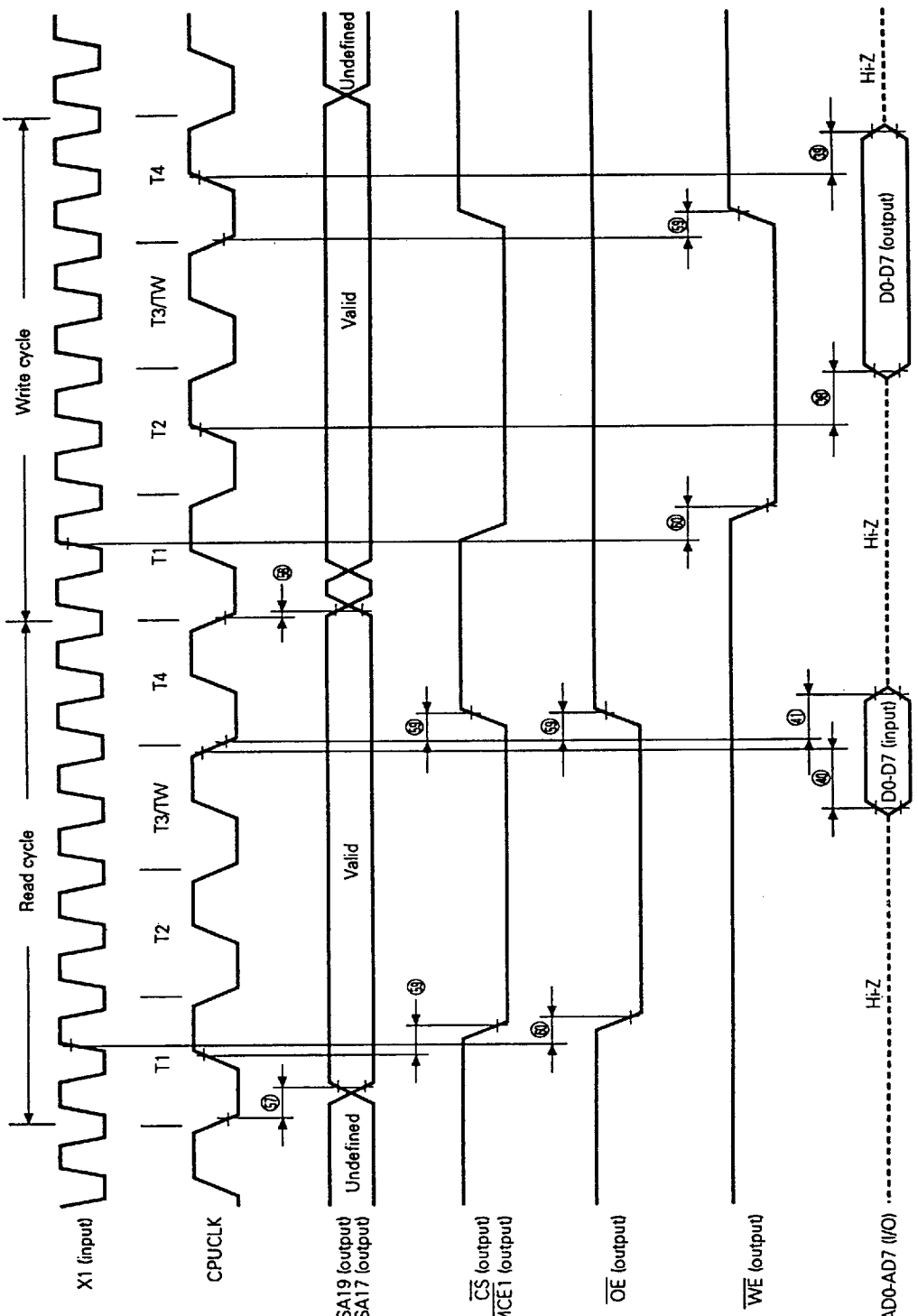
DRAM Access Timing (Other Than DMA Transfer)



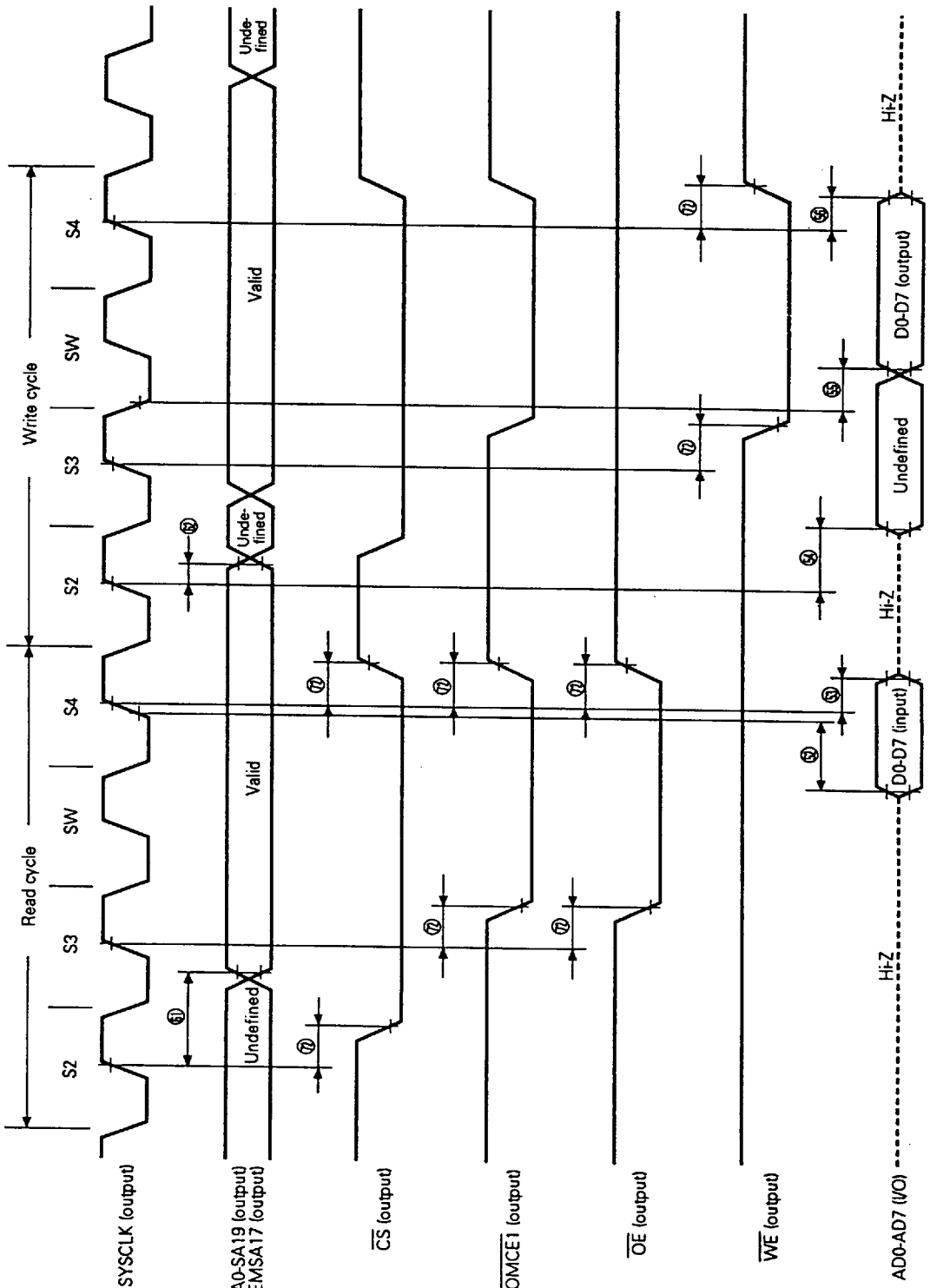
DRAM Access Timing (DMA Transfer)



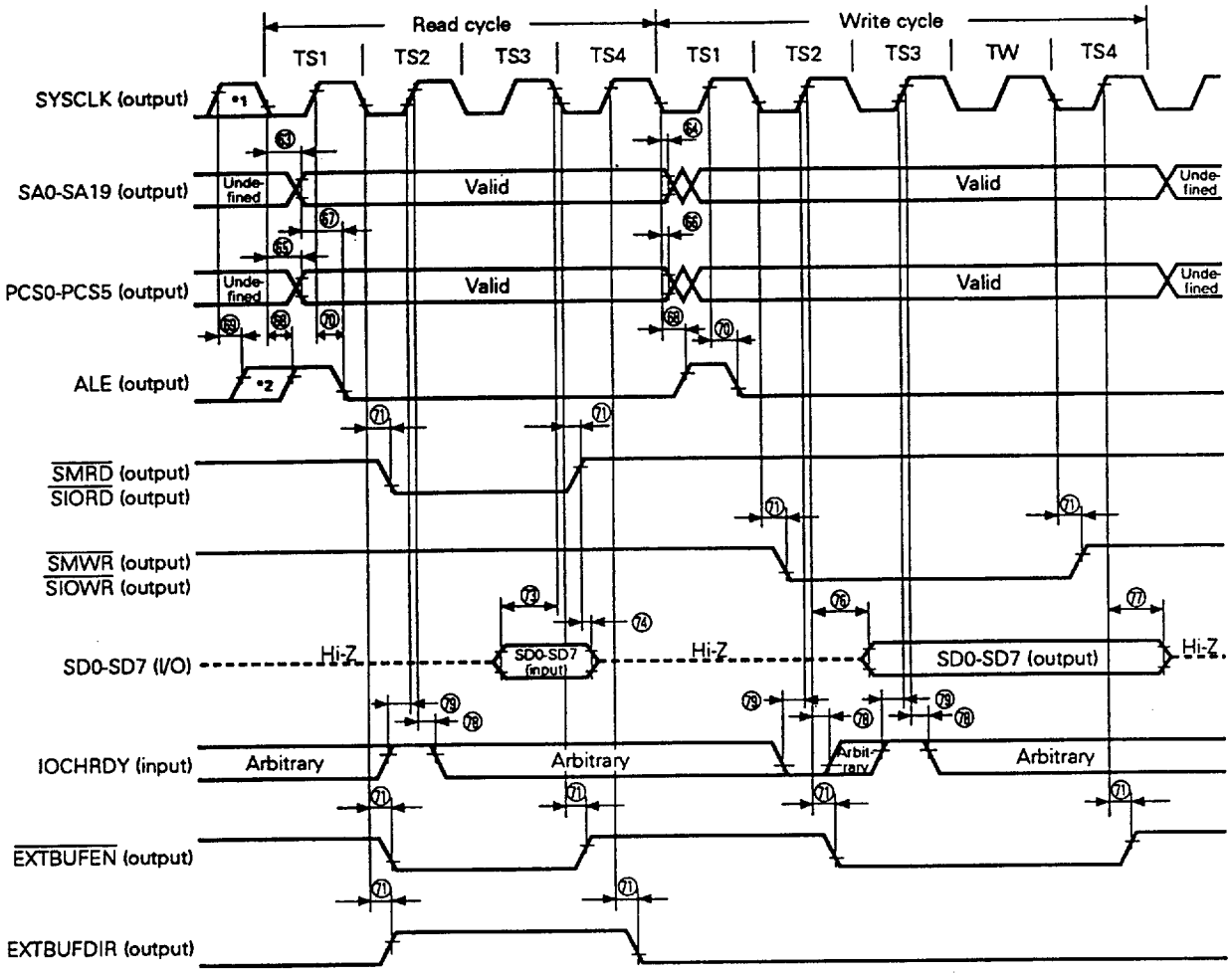
SRAM, Pseudo-SRAM and ROM Access Timing (Other Than DMA Transfer)



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



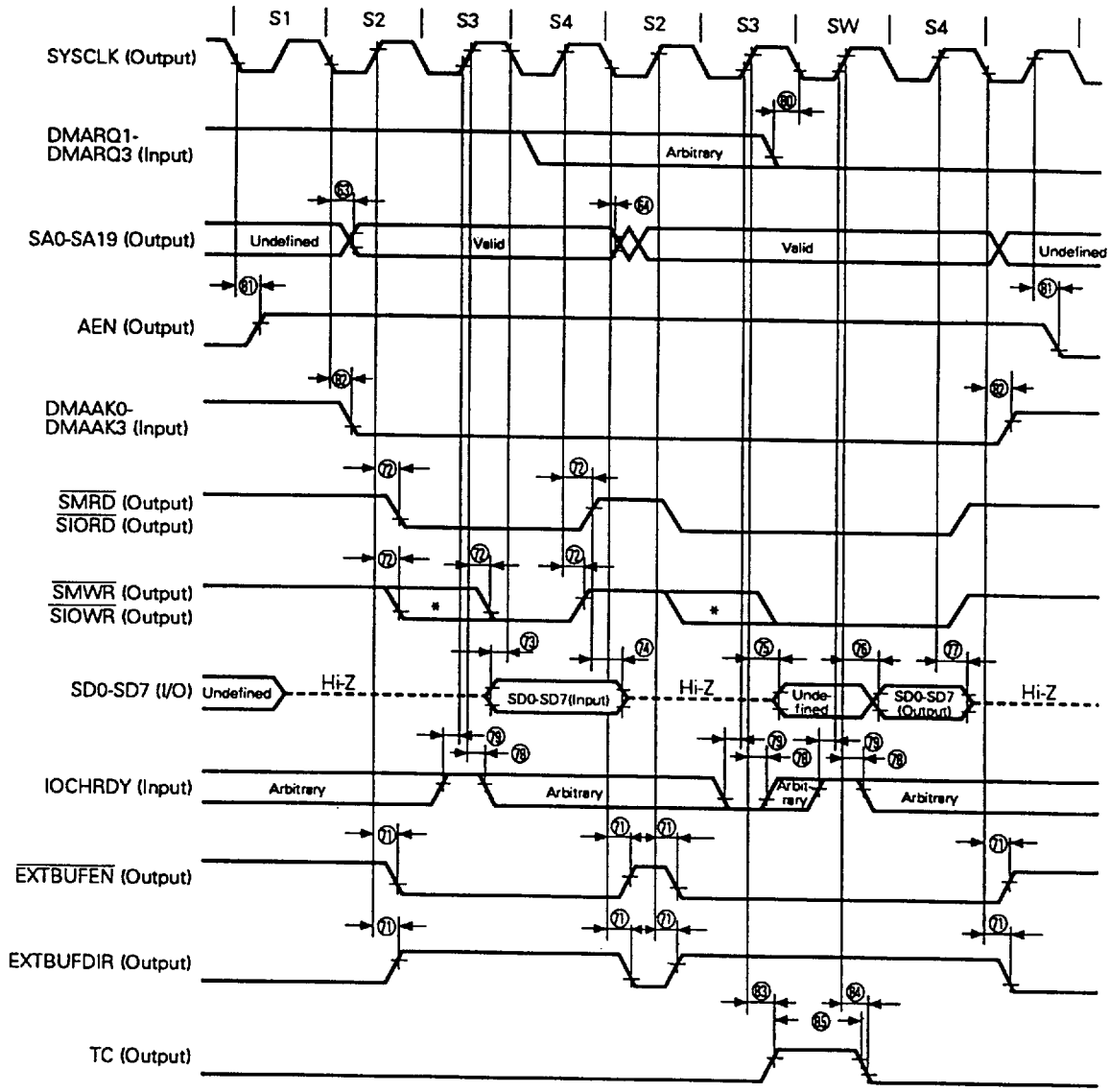
I/O Channel Interface Timing (Other Than DMA Transfer)



*1 : High-level only when cycles of SYSCLK are the same as those of CPUCLK.

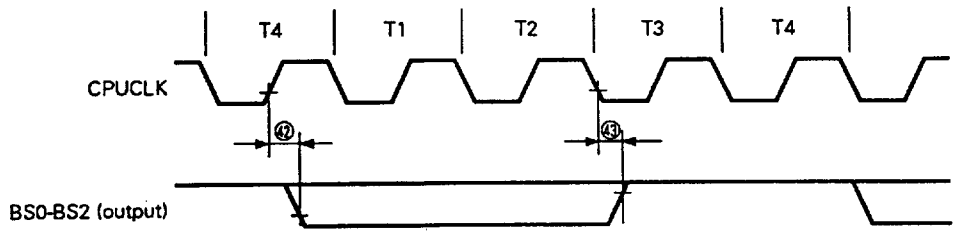
*2 : May become high-level half a clock before, unless bus cycles are continuous.

I/O Channel Interface Timing (DMA Transfer)

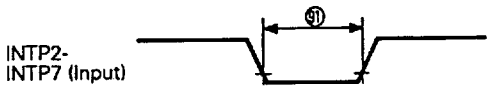


* : Outputs low level when extended write.

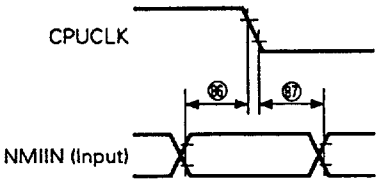
BS0-BS2 Output Timing



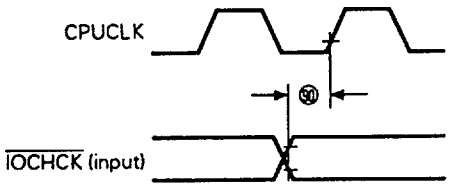
INTP2-INTP7 Input Timing



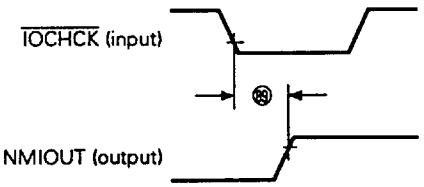
NMIIN Sumple Timing



$\overline{\text{IOCHCK}}$ Input Timing



NMIOUT Output Timing (1)



NMIOUT Output Timing (2)