Z80180/ Z180™ MPU

ZiLOG

Microprocessor Solutions for Datacommunications and Computer Peripheral Applications

User's Manual

TABLE OF CONTENTS

APPENDICES

November 1989

Z80180
Z180 MPU

FEATURES:

- Operating Frequency to 10 MHz
- On-Chip MMU Supports Extended Address Space
- Two DMA Channels
- On-Chip Wait State Generators
- Two UART Channels
- Two 16-Bit Timer Channels

- On-Chip Interrupt Controller
- On-Chip Clock Oscillator/Generator
- Clocked Serial I/O Port
- Code Compatible with Zilog Z80 CPU
- Extended Instructions

GENERAL DESCRIPTION:

Based on a microcoded execution unit and an advanced CMOS manufacturing technology, the Z80180 is an 8-bit MPU which provides the benefits of reduced system costs and low power operation while offering higher performance and maintaining compatibility with a large base of industry standard software written around the Zilog Z80 CPU.

Higher performance is obtained by virtue of higher operating frequencies, reduced instruction execution times, an enhanced instruction set, and an on-chip memory management unit (MMU) with the capability of addressing up to 1 Mbyte of memory.

Reduced system costs are obtained by incorporating several key system functions on-chip with the CPU. These key functions include I/O devices such as DMA, UART, and timer channels. Also included on-chip are several "glue"

functions such as dynamic RAM refresh control, wait state generators, clock oscillator, and interrupt controller.

Not only does the Z80180 consume a low amount of power during normal operation, but it also provides two operating modes that are designed to drastically reduce the power consumption even further. The SLEEP mode reduces power by placing the CPU into a "stopped" state, thereby consuming less current, while the on-chip I/O device is still operating. The SYSTEM STOP mode places both the CPU and the on-chip peripherals into a "stopped" mode, thereby reducing power consumption even further.

When combined with other CMOS VLSI devices and memories, the Z80180 provides an excellent solution to system applications requiring high performance, and low power operation.
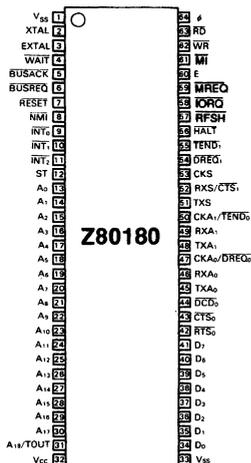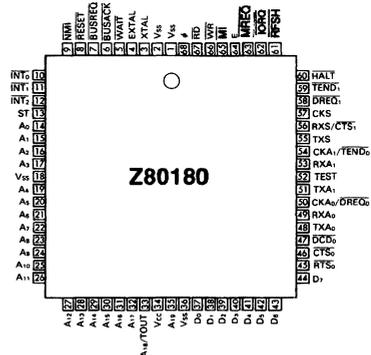

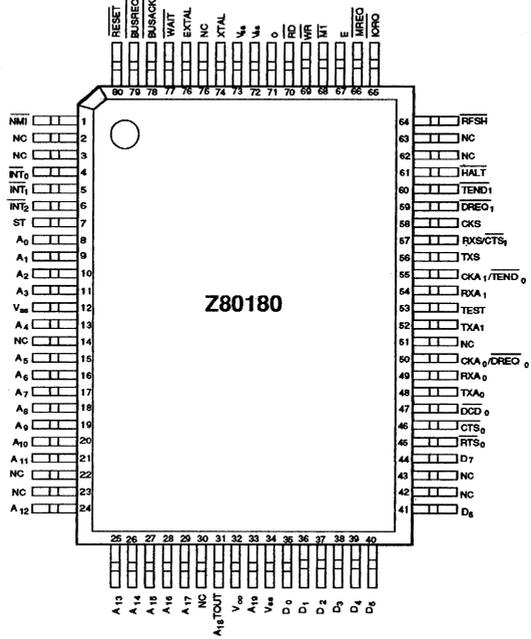Figure 1. 64 Pin DIP

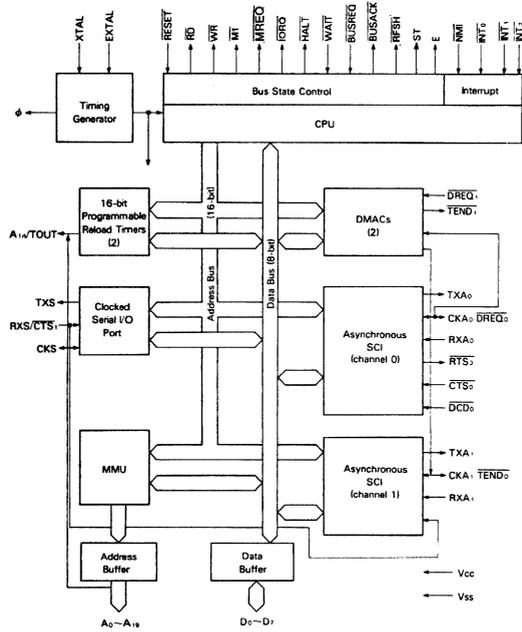Figure 2. 68 Pin PLCC

Figure 3. 80-Pin QFP

Figure 4. Block Diagram

PIN DESCRIPTION:

A_0-A_{19}. *Address Bus (Output, active High, 3-state).* A_0-A_{19} form a 20-bit address bus. The Address Bus provides the address for memory data bus exchanges, up to 1 Mbyte, and I/O data bus exchanges, up to 64K. The address bus enters a high impedance state during reset and external bus acknowledge cycles. Address line A_{18} is multiplexed with the output of PRT channel 1 (TOUT, selected as address output on reset) and address line A_{19} is not available in DIP versions of the Z80180.

$\overline{BUSACK}$. *Bus Acknowledge (Output, active Low).* $\overline{BUSACK}$ indicates the requesting device, the MPU address and data bus, and some control signals, have entered their high impedance state.

$\overline{BUSREQ}$. *Bus Request (Input, active Low).* This input is used by external devices (such as DMA controllers) to request access to the system bus. This request has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle. This signal will stop the CPU from executing further instructions and places the address and data buses, and other control signals, into the high impedance state.

CKA_0, CKA_1. *Asynchronous Clock 0 and 1 (Bidirectional, active High).* These pins are the transmit and receive clocks for the synchronous channels. CKA_0 is multiplexed with $\overline{DREQ_0}$ and CKA_1 is multiplexed with $\overline{TEND_0}$.

CKS. *Serial Clock (Bidirectional, active High).* This line is clock for the CSIO channel.

CLOCK. *System Clock (Output, active High).* The output is used as a reference clock for the MPU and the external system. The frequency of this output is equal to one-half that of the crystal or input clock frequency.

$\overline{CTS_0}$-$\overline{CTS_1}$. *Clear to Send 0 and 1 (Inputs, active Low).* These lines are modem control signals for the ASCI channels. $\overline{CTS1}$ is multiplexed with RXS.

D_0-D_7. *Data Bus (Bidirectional, active High, 3-state).* D_0-D_7 constitute an 8-bit bidirectional data bus, used for the transfer of information to and from I/O and memory devices. The data bus enters the high impedance state during reset and external bus acknowledge cycles.

$\overline{DCD_0}$. *Data Carrier Detect 0 (Input, active Low).* This is a programmable modem control signal for ASCI channel 0.

$\overline{DREQ_0}$, $\overline{DREQ_1}$. DMA Request 0 and 1 (Input, active Low). $\overline{DREQ}$ is used to request a DMA transfer from one of the on-chip DMA channels. The DMA channels monitor these inputs to determine when an external device is ready for a read or write operation. These inputs can be programmed to be either level or edge sensed. $\overline{DREQ_0}$ is multiplexed with CKA_0.

E. *Enable Clock (Output, active High).* Synchronous machine cycle clock output during bus transactions.

EXTAL. *External Clock/Crystal (Input, active High).* Crystal oscillator connection. An external clock can be input to the Z80180 on this pin when a crystal is not used. This input is Schmitt triggered.

$\overline{HALT}$. *Halt/Sleep Status (Output, active Low).* This output is asserted after the CPU has executed either the HALT or SLP instruction, and is waiting for either non-maskable or maskable interrupt before operation can resume. It is also used with the $\overline{M1}$ and ST signals to decode status of the CPU machine cycle.

$\overline{INT_0}$. *Maskable Interrupt Request 0 (Input, active Low).* This signal is generated by external I/O devices. The CPU will honor this request at the end of the current instruction cycle as long as the $\overline{NMI}$ and $\overline{BUSREQ}$ signals are inactive. The CPU acknowledges this interrupt request with an interrupt acknowledge cycle. During this cycle, both the $\overline{M1}$ and $\overline{IORQ}$ signals will become active.

$\overline{INT_1}$, $\overline{INT_2}$. *Maskable Interrupt Requests 1 and 2 (Inputs, active Low).* This signal is generated by external I/O devices. The CPU will honor these requests at the end of the current instruction cycle as long as the $\overline{NMI}$, $\overline{BUSREQ}$, and $\overline{INT_0}$ signals are inactive. The CPU will acknowledge these interrupt requests with an interrupt acknowledge cycle. Unlike the acknowledgement for $\overline{INT_0}$, during this cycle neither the $\overline{M1}$ or $\overline{IORQ}$ signals will become active.

$\overline{IORQ}$. *I/O Request (Output, active Low, 3-state).* $\overline{IORQ}$ indicates that the address bus contains a valid I/O address for an I/O read or I/O write operation. $\overline{IORQ}$ is also generated, along with $\overline{M1}$, during the acknowledgement of the $\overline{INT_0}$ input signal to indicate that an interrupt response vector can be placed onto the data bus. This signal is analogous to the $\overline{IOE}$ signal of the Z64180.

$\overline{M1}$. *Machine Cycle 1 (Output, active Low).* Together with $\overline{MREQ}$, $\overline{M1}$ indicates that the current cycle is the opcode fetch cycle of an instruction execution. Together with $\overline{IORQ}$, $\overline{M1}$ indicates that the current cycle is for an interrupt acknowledge. It is also used with the $\overline{HALT}$ and ST signal to decode status of the CPU machine cycle. This signal is analogous to the $\overline{LIR}$ signal of the Z64180.

$\overline{MREQ}$. *Memory Request (Output, active Low, 3-state).* $\overline{MREQ}$ indicates that the address bus holds a valid address for a memory read or memory write operation. This signal is analogous to the $\overline{ME}$ signal of the Z64180.

$\overline{NMI}$. *Non-maskable Interrupt (Input, negative edge triggered).* $\overline{NMI}$ has a higher priority than $\overline{INT}$ and is always recognized at the end of an instruction, regardless of the state of the interrupt enable flip-flops. This signal forces CPU execution to continue at location 0066H.

$\overline{RD}$. *Read (Output, active Low, 3-state).* $\overline{RD}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O or memory device should use this signal to gate data onto the CPU data bus.

$\overline{RFSH}$. *Refresh (Output, active Low).* Together with $\overline{MREQ}$, $\overline{RFSH}$ indicates that the current CPU machine cycle and the contents of the address bus should be used for refresh of dynamic memories. The low order 8 bits of the address bus (A_7-A_0) contain the refresh address.

This signal is analogous to the $\overline{REF}$ signal of the Z64180.

$\overline{RTS_0}$. *Request to Send 0 (Output, active Low).* This is a programmable modem control signal for ASCI channel 0.

RXA_0, RXA_1. *Receive Data 0 and 1 (Inputs, active High).* These signals are the receive data to the ASCI channels.

RXS. *Clocked Serial Receive Data (Input, active High).* This line is the receiver data for the CSIO channel. RXS is multiplexed with the $\overline{CTS1}$ signal for ASCI channel 1.

ST. *Status (Output, active High).* This signal is used with the $\overline{M1}$ and $\overline{HALT}$ output to decode the status of the CPU machine cycle.

ST	$\overline{HALT}$	$\overline{M1}$	Operation
0	1	0	CPU operation (1st op-code fetch)
1	1	0	CPU operation (2nd op-code and 3rd op-code fetch)
1	1	1	CPU operation (MC except for op-code fetch)
0	X	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (including SYSTEM STOP mode)

NOTE   X: Don't care
MC: Machine cycle

Table 1. Status Summary

$\overline{TEND_0}$, $\overline{TEND_1}$. *Transfer End 0 and 1 (Outputs, active Low).* This output is asserted active during the last write cycle of a DMA operation. It is used to indicate the end of the block transfer. $\overline{TEND_0}$ in multiplexed with CKA_1.

TEST. *Test (Output, not on DIP version).* This pin is for test and should be left open.

TOUT. *Timer Out (Output, active High).* TOUT is the pulse output from PRT channel 1. This line is multiplexed with A_{18} of the address bus.

TXA_0, TXA_1. *Transmit Data 0 and 1 (Outputs, active High).* These signals are the transmitted data from the ASCI channels. Transmitted data changes are with respect to the falling edge of the transmit clock.

TXS. *Clocked Serial Transmit Data (Output, active High).* This line is the transmitted data from the CSIO channel.

$\overline{WAIT}$. *Wait (Input, active Low).* $\overline{WAIT}$ indicates to the MPU that the addressed memory or I/O devices are not ready for a data transfer. This input is used to induce additional clock cycles into the current machine cycle. The $\overline{WAIT}$ input is sampled on the falling edge of T_2 (and subsequent wait states). If the input is sampled low, then additional wait states are inserted until the $\overline{WAIT}$ input is sampled high, at which time execution will continue.

$\overline{WR}$. *Write (Output, active Low, 3-state).* $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored at the addressed I/O or memory location.

XTAL. *Crystal (Input, active High).* Crystal oscillator connection. This pin should be left open if an external clock is used instead of a crystal. The oscillator input is not a TTL level (reference DC characteristics).

Multiplexed pin descriptions

$A_{18}/\overline{TOUT}$ During RESET, this pin is initialized as A_{18} pin. If either TOC1 or TOC0 bit of the Timer Control Register (TCR) is set to 1, TOUT function is selected. If TOC1 and TOC0 bits are cleared to 0, A_{18} function is selected.

$CKA_0/\overline{DREQ_0}$ During RESET, this pin is initialized as CKA_0 pin. If either DM1 or SM1 in DMA Mode Register (DMODE) is set to 1, $\overline{DREQ_0}$ function is always selected.

$CKA_1/\overline{TEND_0}$ During RESET, this pin is initialized as CKA_1 pin. If CKA1D bit in ASCI control register ch 1 (CNTLA1) is set to 1, $\overline{TEND_0}$ function is selected. If CKA1D bit is set to 0, CKA_1 function is selected.

$RXS/\overline{CTS_1}$ During RESET, this pin is initialized as RXS pin. If CTS1E bit in ASCI status register ch1 (STAT1) is set to 1, $\overline{CTS_1}$ function is selected. If CTS1E bit is set to 0, RXS function is selected.

ARCHITECTURE:

The Z80180 combines a high performance CPU core with a variety of system and I/O resources useful in a broad range of applications. The CPU core consists of five functional blocks: clock generator, bus state controller (including dynamic memory refresh), interrupt controller, memory management unit (MMU), and the central processing unit (CPU). The integrated I/O resources make up the remaining four functional blocks: direct memory access (DMA) control (2 channels), asynchronous serial communications interface (ASCI, 2 channels), programmable reload timers (PRT, 2 channels), and a clock serial I/O (CSIO) channel.

Clock Generator. This logic generates the system clock from either an external crystal or clock input. The external clock is divided by two and provided to both internal and external devices.

Bus State Controller. This logic performs all of the status and bus control activity associated with both the CPU and some on-chip peripherals. This includes wait state timing, reset cycles, DRAM refresh, and DMA bus exchanges.

Interrupt Controller. This block monitors and prioritizes the variety of internal and external interrupts and traps to provide the correct responses from the CPU. To remain compatible with the Z80 CPU, three different interrupt modes are supported.

Memory Management Unit. The MMU allows the user to "map" the memory used by the CPU (logically only 64K) into the 1M Byte addressing range supported by the Z80180. The organization of the MMU object code compatibility with the Z80 CPU while offerring access to an extended memory space. This is accomplished by using an effective "common area - banked area" scheme.

Central Processing Unit. The CPU is microcoded to provide a core that is object code compatible with the Z80 CPU. It also provides a superset of the Z80 instruction set, including 8-bit multiply and divide. This core has been enhanced to allow many of the instructions to execute in fewer clock cycles.

DMA Controller. The DMA controller provides high speed transfers between memory and I/O devices. Transfer operations supported are memory to memory, memory to/from I/O, and I/O to I/O. Transfer modes supported are request, burst, and cycle steal. DMA transfers can access the full 1 Mbyte addressing range with a block length up to 64K bytes, and can cross over 64K boundaries.

Asynchronous Serial Communications Interface (ASCI). The ASCI logic provides two individual full-duplex UARTs. Each channel includes a programmable baud rate generator and modem control signals. The ASCI channels can also support a multiprocessor communications format.

Programmable Reload Timer (PRT). This logic consists of two separate channels, each containig a 16-bit counter (timer) and count reload register. The time base for the counters is derived from the system clock (divided by 20) before reaching the counter. PRT channel 1 provides an optional output to allow for waveform generation.

Clocked Serial I/O (CSIO). The CSIO channel provides a half-duplex serial transmitter and receiver. This channel can be used for simple high-speed data connection to another microprocessor or microcomputer.

OPERATION MODES:

The Z80180 can be configured to operate like the 64180. This is accomplished by allowing the user to have control over the $\overline{M1}$, $\overline{IORQ}$, $\overline{WR}$, and $\overline{RD}$ signals. The Operation Mode Control Register (OMCR) determines the $\overline{M1}$ options; the timing of the $\overline{IORQ}$, RD, and $\overline{WR}$ signals; and the RETI operation.
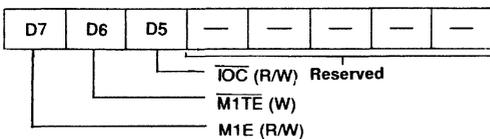
Figure 4. Operation Mode Control Register

$\underline{M1E}$ ($\overline{M1}$ Enable): This bit controls the M1 output and is set to a 1 during reset.

When M1E=1, the $\overline{M1}$ output is asserted LOW during the opcode fetch cycle, the $\overline{INT_0}$ acknowledge cycle, and the first machine cycle of the $\overline{NMI}$ acknowledge. This will also cause the $\overline{M1}$ signal to be active during both fetches of the RETI instruction sequence, which may cause corruption of the external interrupt daisy chain. Hence, this bit should be set to 0 for the Z80180. When M1E=0, the $\overline{M1}$ output is normally inactive and asserted LOW only during the refetch of the RETI instruction sequence and during the $\overline{INT_0}$ acknowledge cycle.

Figure 5. M1 Temporary Enable Timing

$\overline{M1TE}$ (M1 Temporary Enable): This bit controls the temporary assertion of the $\overline{M1}$ signal. It is always read back as

a 1 and is set to 1 during reset. This function is used to "arm" the internal interrupt structure of the Z80PIO. When a control word is written to the Z80PIO to enable interrupts, no enable actually takes place until the PIO sees an active $\overline{M1}$ signal. When $\overline{M1TE}=1$, there is no change in the operation of the $\overline{M1}$ signal and M1E controls its function. When $\overline{M1TE}=0$, the $\overline{M1}$ output will be asserted during the next opcode fetch cycle regardless of the state programmed into the M1E bit. This is only momentary (one time) and the user need not reprogram a 1 to disable the function (See Figure 5).

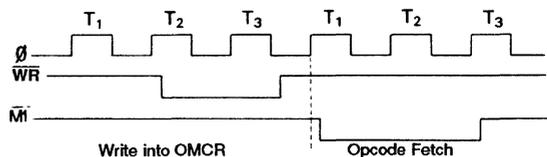$\overline{IOC}$: this bit controls the timing of the $\overline{IORQ}$ and $\overline{RD}$ signals. It is set to 1 by reset.

When $\overline{IOC}=1$, the $\overline{IORQ}$ and $\overline{RD}$ signals function the same as the Z64180.
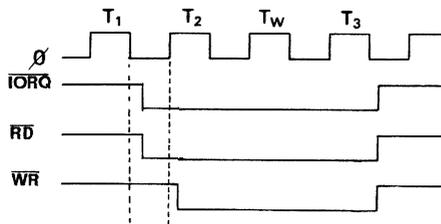
Figure 6. I/O Read and Write Cycles with $\overline{IOC}=1$

When $\overline{IOC}=0$, the timing of the $\overline{IORQ}$ and $\overline{RD}$ signals match the timing required by the Z80 family of peripherals. The $\overline{IORQ}$ and $\overline{RD}$ signals will go active on the rising edge of T2. This allows the Z80180 to satisfy the setup times required by the Z80 peripherals on those two signals.
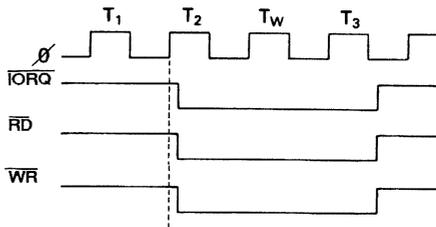
Figure 7. I/O Read and Write Cycles with $\overline{IOC}=0$

For the rest of this manual, it is assumed that M1E=0 and $\overline{IOC}=0$. The user must program the Operation Mode Control Register before the first I/O instruction is executed.

TIMING:

This section explains the Z80180 CPU timing for the following operations:

Instruction (op-code) fetch timing.
Operand and data read/write timing.
I/O read/write timing.
Basic instruction (fetch and execute) timing.
RESET timing.
BUSREQ/BUSACK bus exchange timing.

The basic CPU operation consists of one or more "Machine Cycles" (MC). A machine cycle consists of three system clocks, T1, T2, and T3 while accessing memory or I/O, or it consists of one system clock (T1) during CPU internal operations. The system clock is half the frequency of the Crystal oscillator (e.g., an 8 MHz crystal produces 4 MHz or 250 nsec). For interfacing to slow memory or peripherals, optional wait states (Tw) may be inserted between T2 and T3.

Instruction (op-code) Fetch Timing. Fig. 8 shows the instruction (op-code) fetch timing with no wait states. An opcode fetch cycle is externally indicated when the $\overline{M1}$ output pin is LOW.

In the first half of T_1, the address bus (A_0-A_{19}) is driven

from the contents of the Program Counter (PC). Note that this is the translated address output of the Z80180 on-chip MMU.

In the second half of T_1, the $\overline{MREQ}$ (Memory Request) and $\overline{RD}$ (Read) signals are asserted LOW, enabling the memory.

The op-code on the data bus is latched at the rising edge of T_3 and the bus cycle terminates at the end of T_3.
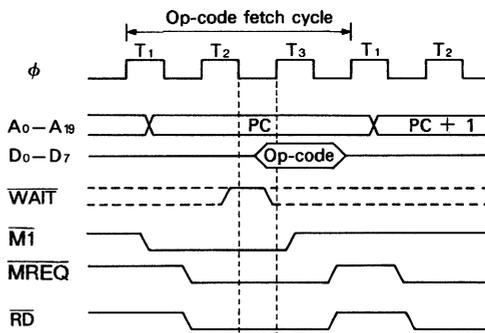
Figure 8. Opcode Fetch timing (Without Wait State)

Fig. 9 illustrates the insertion of wait states (Tw) into the op-code fetch cycle. Wait states (Tw) are controlled by the external $\overline{WAIT}$ input combined with an on-chip programmable wait state generator.

At the falling edge of T_2 the combined $\overline{WAIT}$ input is sampled. If $\overline{WAIT}$ input is asserted LOW, a wait state (Tw) is inserted. The address bus, $\overline{MREQ}$, $\overline{RD}$ and $\overline{M1}$ are held stable during wait states. When the WAIT is sampled inactive HIGH at the falling edge of Tw, the bus cycle enters T_3 and completes at the end of T_3.
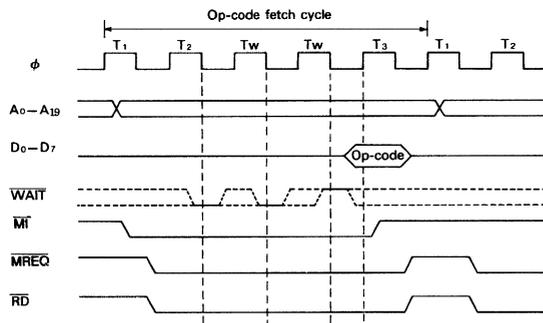
Figure 9. Opcode Fetch Timing (With Wait State)

Operand and Data Read/Write Timing. The instruction operand and data read/write timing differs from op-code fetch timing in two ways. First, the $\overline{M1}$ output is held inactive. Second, the read cycle timing is relaxed by one-half clock cycle since data is latched at the falling edge of T_3.

Instruction operands include immediate data, displacement, and extended addresses, and have the same timing as memory data reads.

During memory write cycles the $\overline{MREQ}$ signal goes active in the second half of T_1. At the end of T_1, the data bus is driven with the write data.

At the start of T_2, the $\overline{WR}$ signal is asserted LOW enabling the memory. $\overline{MREQ}$ and $\overline{WR}$ go inactive in the second half of T3 followed by disabling of the write data on the data bus.

Wait states (Tw) are inserted as previously described for op-code fetch cycles. Fig. 10 illustrates the read/write timing without wait states (Tw), while Fig. 11 illustrates read/write timing with wait states (Tw).
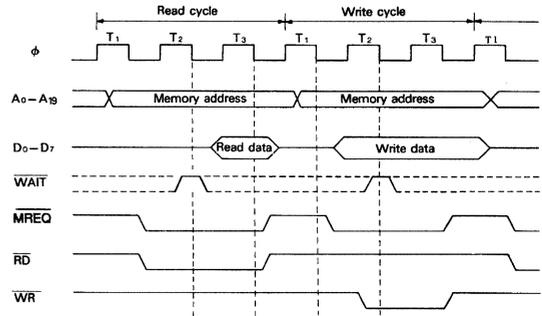
Figure 10. Memory Read/Write Timing (Without Wait State)
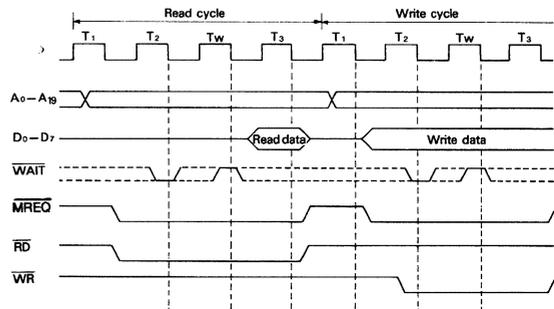
Figure 11. Memory Read/Write Timing (With Wait State)

I/O Read/Write Timing. I/O instructions cause data read/write transfers which differ from memory data transfers in the following three ways:

1. The $\overline{IORQ}$ (I/O Request) signal is asserted LOW instead of the $\overline{MREQ}$ signal.
2. The 16-bit I/O address is not translated by the MMU.
3. A_{16}-A_{19} are held LOW.

At least one wait state (Tw) is always inserted for I/O read and write cycles (except internal I/O cycles).

Fig. 12 shows I/O read/write timing with the automatically inserted wait state (Tw).

NOTE: A_{16}-A_{19} = 0 for I/O cycles
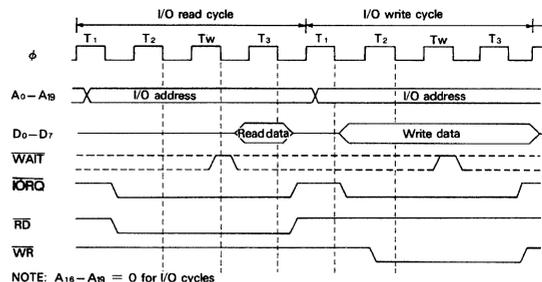
Figure 12. I/O Read/Write Timing

Basic Instruction Timing. An instruction may consist of a number of machine cycles including op-code fetch, operand fetch, and data read/write cycles. An instruction may also include cycles for internal processes which make the bus idle.

NOTE: d = displacement
g = register contents
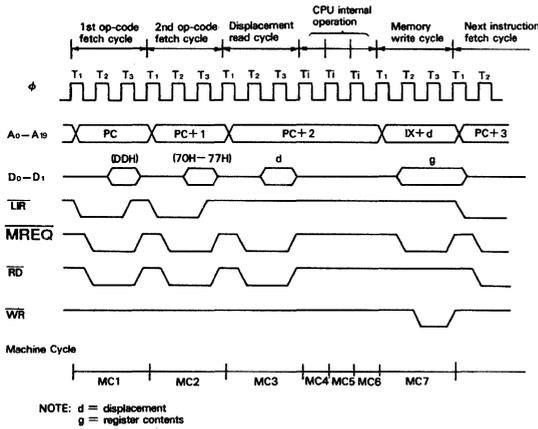
Figure 13. Instruction Timing

The example in Fig. 13 illustrates the bus timing for the data transfer instruction LD (IX+d),g. This instruction moves the contents of a CPU register (g) to the memory location with address computed by adding a signed 8-bit displacement (d) to the contents of an index register (IX).

The instruction cycle starts with the two machine cycles to read the two byte instruction op-code as indicated by $\overline{\text{M1}}$ LOW. Next, the instruction operand (d) is fetched.

The external bus is idle while the CPU computes the effective address. Finally, the computed memory location is written with the contents of the CPU register (g).

RESET Timing. Fig. 14 shows the Z80180 hardware RESET timing. If the $\overline{\text{RESET}}$ pin is LOW for six or more than six clock cycles, processing is terminated and the Z80180 restarts execution from (logical and physical) address 00000H.
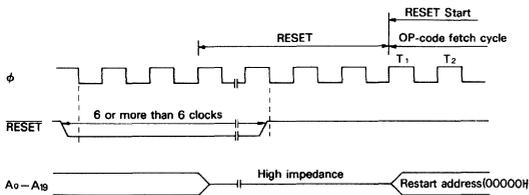
Figure 14. Reset Timing

BUSREQ/BUSACK Bus Exchange Timing. The Z80180 can coordinate the exchange of control, address and data bus ownership with another bus master. The alternate bus master can request the bus release by asserting the $\overline{\text{BUS-}}$

$\overline{\text{REQ}}$ (Bus Request) input LOW. After the Z80180 releases the bus, it relinquishes control to the alternate bus master by asserting the $\overline{\text{BUSACK}}$ (Bus Acknowledge) output LOW.

The bus may be released by the Z80180 at the end of each machine cycle. In this context, a machine cycle consists of a minimum of 3 clock cycles (more if wait states are inserted) for op-code fetch, memory read/write, and I/O read/write cycles. Except for these cases, a machine cycle corresponds to one clock cycle.

When the bus is released, the address (A_0-A_{19}), data (D_0-D_7), and control ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$) signals are placed in the high impedance state.

Note that dynamic RAM refresh is not performed when the Z80180 has released the bus. The alternate bus master must provide dynamic memory refreshing if the bus is released for long periods of time.

Fig. 15 illustrates $\overline{\text{BUSREQ}}$/$\overline{\text{BUSACK}}$ bus exchange during a memory read cycle. Fig. 16 illustrates bus exchange when the bus release is requested during a Z80180 CPU internal operation. $\overline{\text{BUSREQ}}$ is sampled at the falling edge of the system clock prior to T_3, T_i and T_X (BUS RELEASE state). If $\overline{\text{BUSREQ}}$ is asserted LOW at the falling edge of the clock state prior to T_X, another T_X is executed.
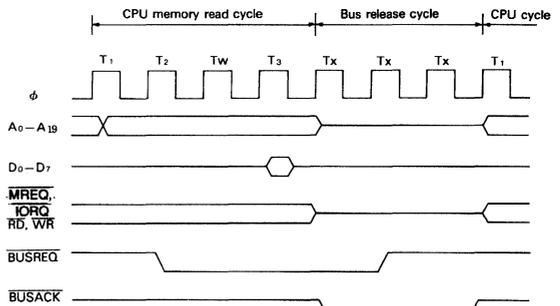
Figure 15. Bus Exchange Timing
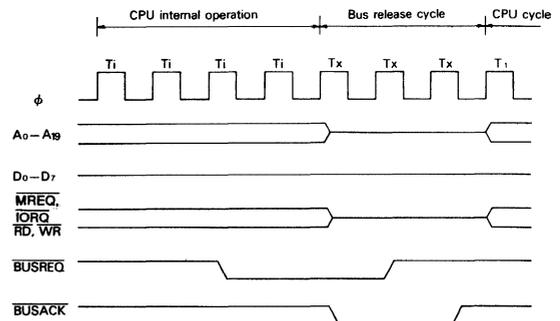
Figure 16. Bus Exchange Timing

WAIT State Generator

To ease interfacing with slow memory and I/O devices, the Z80180 uses wait states (Tw) to extend bus cycle timing. A wait state(s) is inserted based on the combined (logical OR) state of the external $\overline{\text{WAIT}}$ input and an internal programmable wait state (Tw) generator. Wait states (Tw) can be inserted in both CPU execution and DMA transfer cycles.

When the external $\overline{\text{WAIT}}$ input is asserted LOW, wait state(s) (Tw) are inserted between T_2 and T_3 to extend the bus cycle duration. The $\overline{\text{WAIT}}$ input is sampled at the falling edge of the system clock in T_2 or Tw. If the $\overline{\text{WAIT}}$ input is asserted LOW at the falling edge of the system clock in Tw, another Tw is inserted into the bus cycle. Note that $\overline{\text{WAIT}}$ input transitions must meet specified set-up and hold times. This can easily be accomplished by externally synchronizing $\overline{\text{WAIT}}$ input transitions with the rising edge of the system clock.

Dynamic RAM refresh is not performed during wait states (Tw) and thus system designs which use the automatic refresh function must consider the affects of the occurrence and duration of wait states (Tw). Figure 8 shows $\overline{\text{WAIT}}$ timing.
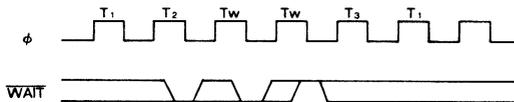
Figure 17. $\overline{\text{WAIT}}$ Timing

Programmable Wait State Insertion. In addition to the $\overline{\text{WAIT}}$ input, wait states (Tw) can also be inserted by program using the Z80180 on-chip wait state generator. Wait state (Tw) timing applies for both CPU execution and on-chip DMAC cycles.

By programming the four significant bits of the DMA/WAIT Control Register (DCNTL) the number of wait states, (Tw) automatically inserted in memory and I/O cycles, can be separately specified. Bits 4 and 5 specify the number of wait states (Tw) inserted for I/O access and bits 6 and 7 specify the number of wait states (Tw) inserted for memory access.

bit	7	6	5	4	
	MWI1	MWI0	IWI1	IWI0	
	R/W	R/W	R/W	R/W	

Figure 18. Memory and I/O Wait State Insertion

The number of wait states (Tw) inserted in a specific cycle is the maximum of the number requested by the WAIT input, and the number automatically generated by the on-chip wait state generator.

Bit 7, 6: MWI1, MWI0, (Memory Wait Insertion)
For CPU and DMAC cycles which access memory (including memory mapped I/O), 0 to 3 wait states may be automatically inserted depending on the programmed value in MWI1 and MWI0.

MWI1	MWI0	The number of wait states
0	0	0
0	1	1
1	0	2
1	1	3

Table 2: Memory Wait States

Bit 5, 4: IWI1, IWI0 (I/O Wait Insertion)
For CPU and DMA cycles which access external I/O (and interrupt acknowledge cycles), 1 to 6 wait states (Tw) may be automatically inserted depending on the programmed value in IWI1 and IWI0.

Note:

(1) For Z80180 internal I/O register access (I/O addresses 0000H-003FH), IWI1 and IWI0 do not determine wait state (Tw) timing. For ASCI, CSI/O and PRT Data Register accesses, 0 to 4 wait states (Tw) will be generated. The number of wait states inserted during access to these registers is a function of internal synchronization requirements and CPU state. All other on-chip I/O register accesses (i.e. MMU, DMAC, ASCI Control Registers, etc.) have 0 wait states inserted and thus require only three clock cycles.

(2) For interrupt acknowledge cycles in which $\overline{\text{M}}$1 is HIGH, such as interrupt vector table read and PC stacking cycle, memory access timing applies.

		the number of wait states				
		For external I/O registers accesses	For internal I/O registers accesses	For $\overline{\text{INT}_0}$ interrupt acknowledge cycles when $\overline{\text{M1}}$ is LOW	For $\overline{\text{INT}_1}$, $\overline{\text{INT}_2}$ and internal interrupts acknowledge cycles (Note (2))	For $\overline{\text{NMI}}$ interrupt acknowledge cycles when $\overline{\text{LIR}}$ is LOW (Note (2))
IWI1	IWI0					
0	0	1		2		
0	1	2	0	4	2	0
1	0	3	(Note (1))	5		
1	1	4		6		

Table 3: Wait State Insertion

9

WAIT input and RESET. During RESET, MWI1, MWI0, IWI1 and IWI0 are all set = 1, selecting the maximum number of wait states (Tw) (3 for memory accesses, 4 for external I/O accesses).

Also, note that the $\overline{\text{WAIT}}$ input is ignored during RESET. For example, if RESET is detected while the Z80180 is in a wait state (Tw), the wait stated cycle in progress will be aborted, and the RESET sequence initiated. Thus, $\overline{\text{RESET}}$ has higher priority than $\overline{\text{WAIT}}$.

HALT and Low Power Operation Modes

The Z80180 can operate in 4 different modes. HALT mode, IOSTOP mode and 2 low power operation modes - SLEEP and SYSTEM STOP. Note that in all operating modes, the basic CPU clock (XTAL, EXTAL) must remain active.

HALT mode. HALT mode is entered by execution of the HALT instruction (op-code = 76H) and has the following characteristics.

(1) The internal CPU clock remains active.

(2) All internal and external interrupts can be received.

(3) Bus exchange ($\overline{\text{BUSREQ}}$ and $\overline{\text{BUSACK}}$) can occur.

(4) Dynamic RAM refresh cycle ($\overline{\text{RFSH}}$) insertion continues at the programmed interval.

(5) I/O operations (ASCI, CSI/O and PRT) continue.

(6) The DMAC can operate.

(7) The $\overline{\text{HALT}}$ output pin is asserted LOW.

(8) The external bus activity consists of repeated "dummy" fetches of the op-code following the HALT instruction.

Essentially, the Z80180 operates normally in HALT mode, except that instruction execution is stopped.

HALT mode can be exited in the following two ways.

RESET Exit from HALT mode. If the $\overline{\text{RESET}}$ input is asserted LOW for at least 6 clock cycles, HALT mode is exited and the normal RESET sequence (restart at address 00000H) is initiated.

Interrupt Exit from HALT mode. When an internal or external interrupt is generated, HALT mode is exited and the normal interrupt response sequence is initiated.

If the interrupt source is masked (individually by enable bit, or globally by IEF1 state), the Z80180 remains in HALT mode. However, $\overline{\text{NMI}}$ interrupt will initiate the normal $\overline{\text{NMI}}$

interrupt response sequence independent of the state of IEF_1.

HALT timing is shown in Fig. 19.
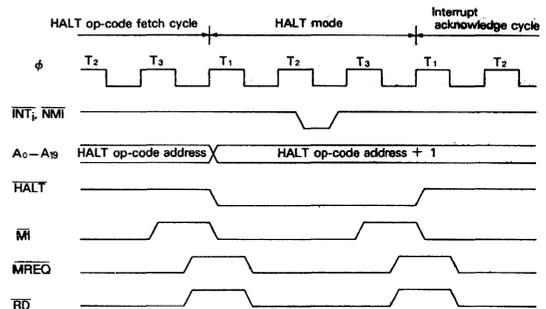
Figure 19. HALT Timing

SLEEP mode. SLEEP mode is entered by execution of the 2 byte SLP instruction. SLEEP mode has the following characteristics.

(1) The internal CPU clock stops, reducing power consumption.

(2) The internal crystal oscillator does not stop.

(3) Internal and external interrupt inputs can be received.

(4) DRAM refresh cycles stop.

(5) I/O operations using on-chip peripherals continue.

(6) The internal DMAC stop.

(7) $\overline{\text{BUSREQ}}$ can be received and acknowledged.

(8) Address outputs go HIGH and all other control signal output become inactive HIGH.

(9) Data Bus, 3-state.

SLEEP mode is exited in one of two ways as shown below.

RESET Exit from SLEEP mode. If the $\overline{RESET}$ input is held LOW for at least 6 clock cycles, it will exit SLEEP mode and begin the normal RESET sequence with execution starting at address (logical and physical) 00000H.

Interrupt Exit from SLEEP mode. The SLEEP mode is exited by detection of an external (NMI, INT_0-INT_2) or internal (ASCI, CSI/O, PRT) interrupt.

In case of $\overline{NMI}$, SLEEP Mode is exited and the CPU begins the normal $\overline{NMI}$ interrupt response sequence.

In the case of all other interrupts, the interrupt response depends on the state of the global interrupt enable flag (IEF_1) and the individual interrupt source enable bit.

If the individual interrupt condition is disabled by the corresponding enable bit, occurrence of that interrupt is ignored and the CPU remains in the SLEEP state.

Assuming the individual interrupt condition is enabled, the response to that interrupt depends on the global interrupt enable flag (IEF_1). If interrupts are globally enabled (IEF_1=1) and an individually enabled interrupt occurs, SLEEP mode is exited and the appropriate normal interrupt response sequence is executed.

If interrupts are globally disabled (IEF_1=0) and an individually enabled interrupt occurs, SLEEP mode is exited and instruction execution begins with the instruction following the SLP instruction. Note that this provides a technique for synchronization with high speed external events without incurring the latency imposed by an interrupt response sequence.

Figure 10 shows SLEEP timing.

IOSTOP mode. IOSTOP mode is entered by setting the IOSTOP bit of the I/O Control Register (ICR) to 1. In this case, on-chip I/O (ASCI, CSI/O, PRT) stops operating. However, the CPU continues to operate. Recovery from IOSTOP mode is by resetting the IOSTOP bit in ICR to 0.

SYSTEM STOP mode. SYSTEM STOP mode is the combination of SLEEP and IOSTOP modes. SYSTEM STOP mode is entered by setting the IOSTOP bit in ICR to 1 followed by execution of the SLP instruction. In this mode, on-chip I/O and CPU stop operating, reducing power consumption. Recovery from SYSTEM STOP mode is the same as recovery from SLEEP mode, noting that internal I/O sources (disabled by IOSTOP) cannot generate a recovery interrupt.
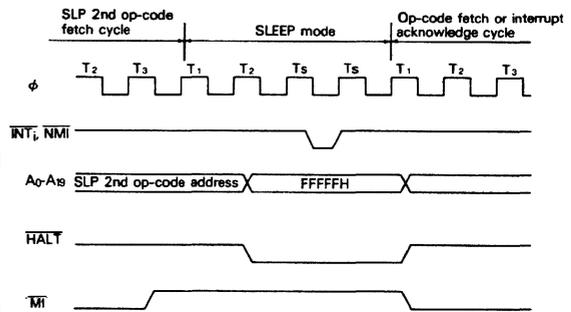
Figure 20. SLEEP Timing

Internal I/O Registers

The Z80180 internal I/O Registers occupy 64 I/O addresses (including reserved addresses). These registers access the internal I/O modules (ASCI, CSI/O, PRT) and control functions (DMAC, DRAM refresh, interrupts, wait state generator, MMU and I/O relocation).

To avoid address conflicts with external I/O, the Z80180 internal I/O addresses can be relocated on 64 byte boundaries within the bottom 256 bytes of the 64K byte I/O address space.

I/O Control Register (ICR). ICR allows relocating of the internal I/O addresses. ICR also controls enabling/disabling of the IOSTOP mode.
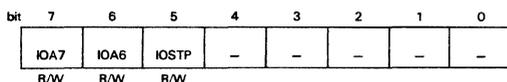
Figure 21 I/O Control Register (ICR : I/O Address = 3FH)

IOA7, 6: I/O Address Relocation (bits 7, 6)
IOA7 and IOA6 relocate internal I/O as shown in Fig. 22. Note that the high-order 8 bits of 16-bit internal I/O addresses are always 0. IOA7 and IOA6 are cleared to 0 during RESET.
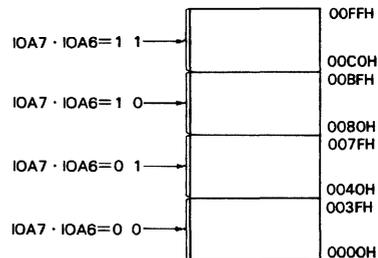
Figure 22. I/O Address Relocation

IOSTP: IOSTOP Mode (bit 5)
IOSTOP mode is enabled when IOSTP is set to 1. Normal

I/O operation resumes when IOSTP is reset to 0. IOSTP is cleared to 0 during RESET.

Internal I/O Registers Address Map

The internal I/O register addresses are shown in Table 4. These addresses are relative to the 64 byte boundary base address specified in ICR.

I/O Addressing Notes. The internal I/O register addresses are located in the I/O address space from 0000H to 00FFH (16-bit I/O addresses). Thus, to access the internal I/O registers (using I/O instructions), the high-order 8 bits of the 16-bit I/O address must be 0.

The conventional I/O instructions (OUT (m), A/ IN A, (m) / OUTI / INI/ etc.) place the contents of a CPU register on the high-order 8 bits of the address bus, and thus may be difficult to use for accessing internal I/O registers.

For efficient internal I/O register access, a number of new instructions have been added, which force the high-order 8 bits of the 16-bit I/O address to 0. These instructions are IN0, OUT0, OTIM, OTIMR, OTDM, OTDMR and TSTIO (see Instruction Set).

When writing to an internal I/O register, the same I/O write occurs on the external bus. However, the duplicate external I/O write cycle will exhibit internal I/O write cycle timing. For example, the $\overline{WAIT}$ input and programmable wait state generator are ignored. Similarly, internal I/O read cycles also cause a duplicate external I/O read cycle - however, the external read data is ignored by the Z80180.

Normally, external I/O addresses should be chosen to avoid overlap with internal I/O addresses to avoid duplicate I/O accesses.

	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCI	ASCI Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCI Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCI Control Register B Ch 0	CNTLB0	XX000010	02H
	ASCI Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCI Status Register Ch 0	STAT0	XX000100	04H
	ASCI Status Register Ch 1	STAT1	XX000101	05H
	ASCI Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCI Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCI Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCI Receive Data Register Ch 1	RDR1	XX001001	09H
CSI/O	CSI/O Control Register	CNTR	XX001010	0AH
	CSI/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch 0L	TMDR0L	XX001100	0CH
	Timer Data Register Ch 0H	TMDR0H	XX001101	0DH
	Reload Register Ch 0L	RLDR0L	XX001110	0EH
	Reload Register Ch 0H	RLDR0H	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001	11H
			$\}$	$\}$
			XX010011	13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
	Reload Register Ch 1L	RLDR1L	XX010110	16H
	Reload Register Ch 1H	RLDR1H	XX010111	17H
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			$\}$	$\}$
			XX011111	1FH

Table 4. I/O Address Map

| | Register | Mnemonic | Address | |
			Binary	Hexadecimal
	DMA Source Address Register Ch 0L	SAR0L	XX100000	20H
	DMA Source Address Register Ch 0H	SAR0H	XX100001	21H
	DMA Source Address Register Ch 0B	SAR0B	XX100010	22H
	DMA Destination Address Register Ch 0L	DAR0L	XX100011	23H
	DMA Destination Address Register Ch 0H	DAR0H	XX100100	24H
	DMA Destination Address Register Ch 0B	DAR0B	XX100101	25H
	DMA Byte Count Register Ch 0L	BCR0L	XX100110	26H
	DMA Byte Count Register Ch 0H	BCR0H	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
DMA	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Reserved		XX101101	2DH
	DMA Byte Count Register Ch 1L	BCR1L	XX101110	2EH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
	DMA/WAIT Control Register	DCNTL	XX110010	32H
	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H
INT	INT/TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H

| | Register | Mnemonic | Address | |
			Binary	Hexadecimal
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
	MMU Common Base Register	CBR	XX111000	38H
MMU	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
	Reserved		XX111011	3BH
I/O			XX111101	3DH
	Operation Mode Control Register	OMCR	XX111110	3EH
	I/O Control Register	ICR	XX111111	3FH

Table 4. I/O Address Map (cont.)

Memory Management Unit (MMU)

The Z80180 has an on-chip MMU which performs the translation of the CPU 64K byte (16-bit addresses 0000H to FFFFH) logical memory address space into a 1024K byte (20-bit addresses 00000H to FFFFFH) physical memory address space. Address translation occurs internally in parallel with other CPU operation.

Logical Address Spaces. The 64K byte CPU logical address space is interpreted by the MMU as consisting of up to three separate logical address areas, Common Area 0, Bank Area, and Common Area 1.

As shown in Fig. 23, a variety of logical memory configurations are possible. The boundaries between the Common and Bank Areas can be programmed with 4K byte resolution.
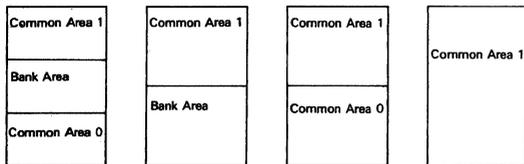
Figure 23. Logical Address Mapping Examples

Logical to Physical Address Translation. Fig. 24 shows an example in which the three logical address space portions are mapped into a 1024K byte physical address space. The important points to note are that Common and Bank Areas can overlap and that Common Area 1 and Bank Area can be freely relocated (on 4K bytes physical address boundaries). Common Area 0 (if it exists) is always based at physical address 00000H.
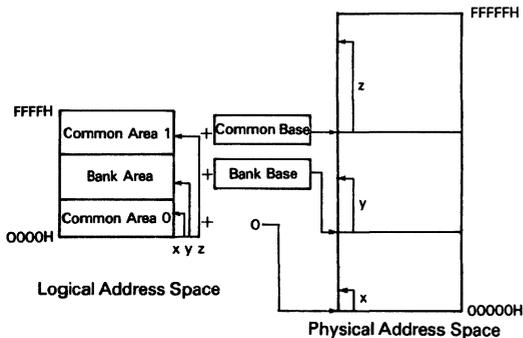
Figure 24. Physical Address Translation

MMU Block Diagram. The MMU block diagram is shown in Fig. 25. The MMU translates internal 16-bit logical addresses to external 20-bit physical addresses.
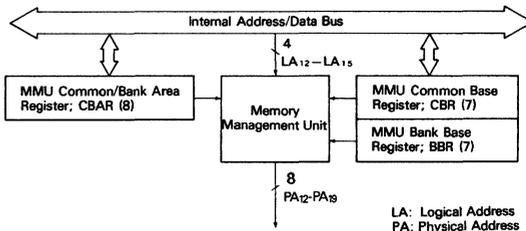
Figure 25. MMU Block Diagram

Whether address translation takes place depends on the type of CPU cycle as follows.

(1) Memory Cycles
Address Translation occurs for all memory access cycles including instruction and operand fetches, memory data reads and writes, hardware interrupt vector fetch, and software interrupt restarts.

(2) I/O Cycles
The MMU is logically bypassed for I/O cycles. The 16-bit logical I/O address space corresponds directly with the 16-bit physical I/O address space. The four high-order bits ($A16$-$A19$) of the physical address are always 0 during I/O cycles.
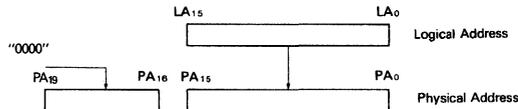
Figure 26. I/O Address Translation

(3) DMA Cycles
When the Z80180 on-chip DMAC is using the external bus, the MMU is physically bypassed. The 20-bit source and destination registers in the DMAC are directly output on the physical address bus (A0-A19).

MMU Registers. Three MMU registers are used to program a specific configuration of logical and physical memory.

(1) MMU Common/Bank Area Register (CBAR)

(2) MMU Common Base Register (CBR)

(3) MMU Bank Base Register (BBR)

CBAR is used to define the logical memory organization, while CBR and BBR are used to relocate logical areas within the 1024k byte physical address space. The resolution for both setting boundaries within the logical space and relocation within the physical space is 4k bytes.

The CAR field of CBAR determines the start address of Common Area 1 (Upper Common) and by default, the end

address of the Bank Area. The BAR field determines the start address of the Bank Area and by default, the end address of Common Area 0 (Lower Common).

The CA and BA fields of CBAR may be freely programmed subject only to the restriction that CA may never be less than BA. Fig. 27 and Fig. 28 show examples of logical memory organizations associated with different values of CA and BA.
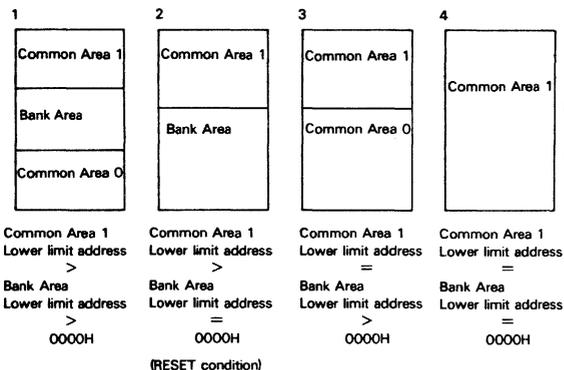
Figure 27. Logical Memory Organization
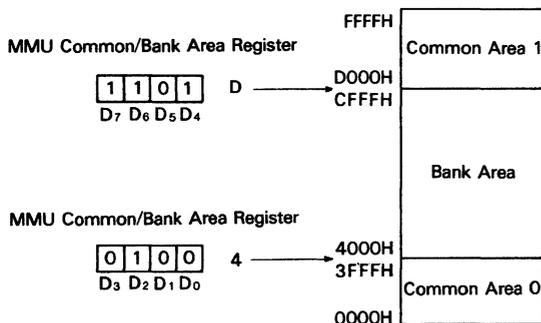
Figure 28. Logical Space Configuration (Example)

MMU Register Description:

MMU Common/Bank Area Register (CBAR). CBAR specifies boundaries within the Z80180 64K byte logical address space for up to three areas; Common Area 0, Bank Area and Common Area 1.

MMU Common/Bank Area Register (CBAR : I/O Address = 3AH)

bit 7	6	5	4	3	2	1	0
CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 29. MMU Common/Bank Area Register (CBAR : I/O Address = 3AH)

CA3-CA0: CA (bits 7-4)
CA specifies the start (low) address (on 4K byte boundaries) for the Common Area 1. This also determines the

last address of the Bank Area. All bits of CA are set to 1 during RESET.

BA-BA0: BA (bits 3-0)
BA specifies the start (low) address (on 4k bytes boundaries) for the Bank Area. This also determines the last address of the Common Area 0. All bits of BA are set to 1 during RESET.

MMU Common Base Register (CBR). CBR specifies the base address (on 4k byte boundaries) used to generate a 20-bit physical address for Common Area 1 accesses. All bits of CBR are reset to 0 during RESET.

bit 7	6	5	4	3	2	1	0
CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 30. MMU Common Base Register (CBR : I/O Address = 38H)

MMU Bank Base Register (BBR). BBR specifies the base address (on 4k byte boundaries) used to generate a 19-bit physical address for Bank Area accesses. All bits of BBR are reset to 0 during RESET.

bit 7	6	5	4	3	2	1	0
BB7	BB6	BB5	BB4	BB3	BB2	BB1	BB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 31. MMU Bank Base Register (BBR : I/O Address = 39H)

Physical address translation. Fig. 17 shows the way in which physical addresses are generated based on the contents of CBAR, CBR and BBR. MMU comparators classify an access by logical area as defined by CBAR. Depending on which of the three potential logical areas (Common Area 1, Bank Area, or Common Area 0) is being accessed, the appropriate 8-bit base address is added to the high-order 4 bits of the logical address, yielding a 20-bit physical address. CBR is associated with Common Area 1 accesses. Common Area 0, if defined, is always based at physical address 00000H.

MMU and RESET. During RESET, all bits of the CA field of CBAR are set to 1 while all bits of the BA field of CBAR, CBR and BBR are reset to 0. The logical 64K byte address space corresponds directly with the first 64K bytes (0000H to FFFFH) of the 1024K byte (00000H to FFFFFH) physical address space. Thus, after RESET, the Z80180 will begin execution at logical and physical address 0.

MMU Register Access Timing. When data is written into CBAR, CBR or BBR, the value will be effective from the cycle immediately following the I/O write cycle which updates these registers.

Care must be taken during MMU programming to insure that CPU program execution is not disrupted. Observe that the next cycle following MMU register programming will

normally be an op-code fetch from the newly translated address. One simple technique is to localize all MMU programming routines in a Common Area that is always enabled.
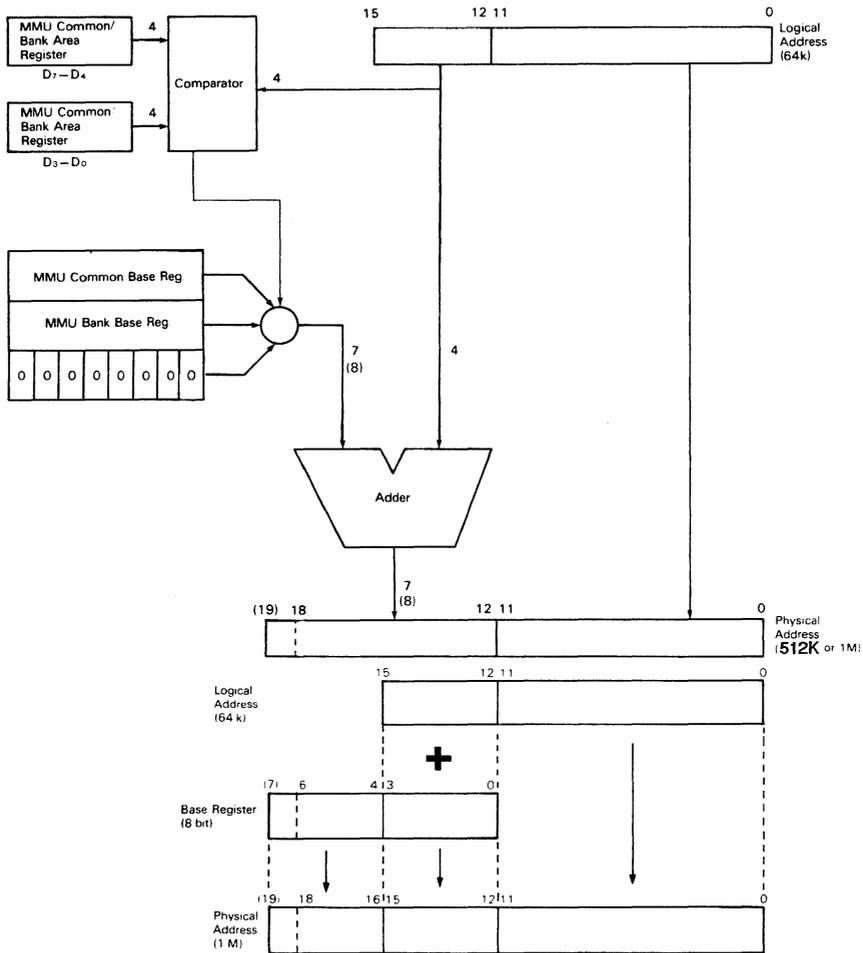
Figure 32. Physical Address Generation

Interrupts

The Z80180 CPU has twelve interrupt sources, 4 external and 8 internal, with fixed priority. (Reference Figure 33.)

This section explains the CPU registers associated with interrupt processing, the TRAP interrupt, interrupt response modes, and the external interrupts. The detailed discussion of internal interrupt generation (except TRAP) is presented in the appropriate hardware section (i.e. PRT, DMAC, ASCI, and CSI/O).

Higher Priority	(1)	TRAP (Undefined Op-code Trap)	Internal Interrupt
	(2)	$\overline{NMI}$ (Non Maskable Interrupt)	
	(3)	$\overline{INT_0}$ (Maskable Interrupt Level 0)	External Interrupt
	(4)	$\overline{INT_1}$ (Maskable Interrupt Level 1)	
	(5)	$\overline{INT_2}$ (Maskable Interrupt Level 2)	
	(6)	Timer 0	
	(7)	Timer 1	
	(8)	DMA channel 0	
	(9)	DMA channel 1	Internal Interrupt
	(10)	Clocked Serial I/O Port	
Lower Priority	(11)	Asynchronous SCI channel 0	
	(12)	Asynchronous SCI channel 1	

Figure 33. Interrupt Sources

16

Interrupt Control Registers and Flags. The Z80180 has three registers and two flags which are associated with interrupt processing.

Function	Name	Access Method
(1) Interrupt Vector High	I	LD A,I and LD I, A instructions
(2) Interrupt Vector Low	IL	I/O instruction (addr=33H)
(3) Interrupt/Trap Control	ITC	I/O instruction (addr=34H)
(4) Interrupt Enable Flag 1,2	IEF_1, IEF_2	EI and DI

Interrupt Vector Register (I)

Mode 2 for $\overline{INT}_0$ external interrupt, $\overline{INT}_1$ and $\overline{INT}_2$ external interrupts, and all internal interrupts (except TRAP) use a programmable vectored technique to determine the address at which interrupt processing starts. In response to the interrupt a 16-bit address is generated. This address accesses a vector table in memory to obtain the address at which execution restarts.

While the method for generation of the least significant byte of the table address differs, all vectored interrupts use the contents of I as the most significant byte of the table address. By programming the contents of I, vector tables can be relocated on 256 byte boundaries throughout the 64K byte logical address space.

Note that I is read/written with the LD A, I and LD I, A instructions rather than I/O (IN, OUT) instructions. I is initialized to 00H during RESET.
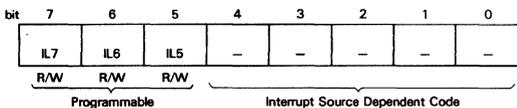
Figure 34. Interrupt Vector Low Register (IL : I/O Address = 33H)

Interrupt Vector Low Register

This register determines the most significant three bits of the low-order byte of the interrupt vector table address for external interrupts INT_1 and INT_2 and all internal interrupts (except TRAP). The five least significant bits are fixed for each specific interrupt source. By programming IL, the vector table can be relocated on 32 byte boundaries. IL is initialized to 00H during RESET.
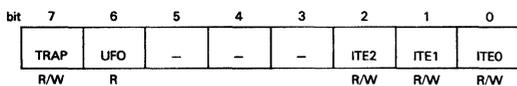
Figure 35. INT/TRAP Control Register (ITC : I/O Address = 34H)

INT/TRAP Control Register (ITC)

ITC is used to handle TRAP interrupts and to enable or disable the external maskable interrupt inputs INT_0, INT_1 and INT_2.

TRAP (bit 7)

This bit is set to 1 when an undefined op-code is fetched. TRAP can be reset under program control by writing it with 0, however, it cannot be written with 1 under program control. TRAP is reset to 0 during RESET.

UFO: Undefined Fetch Object (bit 6)

When a TRAP interrupt occurs the contents of UFO allow determination of the starting address of the undefined instruction. This is necessary since the TRAP may occur on either the second or third byte of the op-code. UFO allows the stacked PC value to be correctly adjusted. If UFO = 0, the first op-code should be interpreted as the stacked PC-1. If UFO = 1, the first op-code address is stacked PC-2. UFO is read-only.

ITE2,1,0: Interrupt Enable 2,1,0 (bits 2-0)

ITE_2, ITE_1 and ITE_0 enable and disable the external interrupt inputs INT_2, INT_1 and INT_0, respectively. If reset to 0, the interrupt is masked. During RESET, ITE_0 is initialized to 1 while ITE_1 and ITE_2 are initialized to 0.

Interrupt Enable Flag 1,2 (IEF1, IEF2)

IEF_1 controls the overall enabling and disabling of all internal and external maskable interrupts (i.e. all interrupts except $\overline{NMI}$ and TRAP.

If IEF_1 = 0, all maskable interrupts are disabled. IEF_1 can be reset to 0 by the DI (Disable Interrupts) instruction and set to 1 by the EI (Enable Interrupts) instruction.

The purpose of IEF_2 is to correctly manage the occurrence of $\overline{NMI}$. During $\overline{NMI}$, the prior interrupt reception state is saved and all maskable interrupts are automatically disabled (IEF_1 copied to IEF_2 and then IEF_1 cleared to 0). At the end of the $\overline{NMI}$ interrupt service routine, execution of the RETN (Return from Non-maskable Interrupt) will automatically restore the interrupt receiving state (by copying IEF2 to IEF1) prior to the occurrence of $\overline{NMI}$.

IEF_2 state can be reflected in the P/V bit of the CPU Status Register by executing LD A, I or LD A, R instructions.

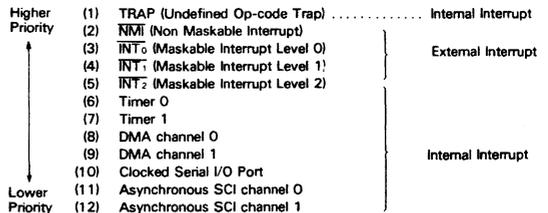| CPU Operation	IEF_1	IEF_2	REMARKS
RESET	0	0	Inhibits the interrupt except $\overline{NMI}$ and TRAP.
$\overline{NMI}$	0	IEF_1	Copies the contents of IEF_1 to IEF_2.
RETN	IEF_2	not affected	Returns from the $\overline{NMI}$ service routine.
Interrupt except $\overline{NMI}$ and TRAP	0	0	Inhibits the interrupt except $\overline{NMI}$ and TRAP.
RETI	not affected	not affected	
TRAP	not affected	not affected	
EI	1	1	
DI	0	0	
LD A, I	not affected	not affected	Transfers the contents of IEF_2 to P/V flag.
LD A, R	not affected	not affected	Transfers the contents of IEF_2 to P/V flag.

Table 5: State of IEF_1 and IEF_2

TRAP Interrupt. The Z80180 generates a non-maskable (not affected by the state of IEF1) TRAP interrupt when an undefined op-code fetch occurs. This feature can be used to increase software reliability, implement an "extended" instruction set, or both. TRAP may occur during op-code fetch cycles and also if an undefined op-code is fetched during the interrupt acknowledge cycle for $\overline{INT}_0$ when Mode 0 is used.

When a TRAP interrupt occurs the Z80180 operates as follows.

(1) The TRAP bit in the Interrupt TRAP/Control (ITC) register is set to 1.

(2) The current PC (Program Counter) value, reflecting location of the undefined op-code, is saved on the stack.

(3) The Z80180 vectors to logical address 0. Note that if logical address 0000H is mapped to physical address 00000H, the vector is the same as for RESET. In this case, testing the TRAP bit in ITC will reveal whether the restart at physical address 00000H was caused by RESET or TRAP.

The state of the UFO (Undefined Fetch Object) bit in ITC allows TRAP manipulation software to correctly "adjust" the stacked PC, depending on whether the second or third byte of the op-code generated the TRAP. If UFO=0, the starting address of the invalid instruction is equal to the stacked PC-1. If UFO=1, the starting address of the invalid instruction is equal to the stacked PC-2.

Note that Bus Release cycle, Refresh cycle, DMA cycle, and $\overline{WAIT}$ cycle cannot be inserted just after TTP state which is inserted for TRAP interrupt sequence.Figure 36 shows TRAP Timing - 2nd Op-code undefined and Figure 37 shows Trap Timing - 3rd Op-code undefined.
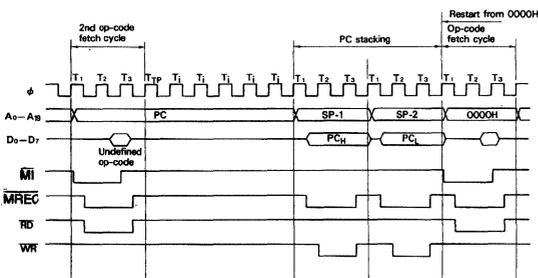
Figure 36. TRAP Timing-2nd Op-Code Undefined
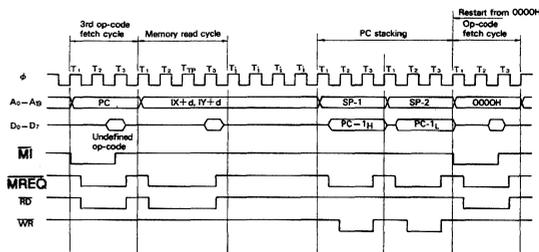
Figure 37. TRAP Timing-3rd Op-Code Undefined

External Interrupts. The Z80180 has four external hardware interrupt inputs.

(1) $\overline{NMI}$ - Non-maskable Interrupt
(2) $\overline{INT}_0$ - Maskable Interrupt Level 0
(3) $\overline{INT}_1$ - Maskable Interrupt Level 1
(4) $\overline{INT}_2$ - Maskable Interrupt Level 2

$\overline{NMI}$, $\overline{INT}_1$ and $\overline{INT}_2$ have fixed interrupt response modes. $\overline{INT}_0$ has 3 different software programmable interrupt response modes - Mode 0, Mode 1 and Mode 2.

$\overline{NMI}$ - Non-Maskable Interrupt. The $\overline{NMI}$ interrupt input is edge sensitive and cannot be masked by software. When $\overline{NMI}$ is detected, the Z80180 operates as follows.

(1) DMAC operation is suspended by the clearing of the DME (DMA Main Enable) bit in DCNTL.

(2) The PC is pushed onto the stack.

(3) The contents of IEF$_1$ are copied to IEF$_2$. This saves the interrupt reception state that existed prior to $\overline{NMI}$.

(4) IEF$_1$ is cleared to 0. This disables all external and internal maskable interrupts (i.e. all interrupts except $\overline{NMI}$ and TRAP).

(5) Execution commences at logical address 0066H.

The last instruction of an $\overline{NMI}$ service routine should be RETN (Return from Non-maskable Interrupt). This restores the stacked PC, allowing the interrupted program to continue. Furthermore, RETN causes IEF$_2$ to be copied to IEF$_1$, restoring the interrupt reception state that existed prior to $\overline{NMI}$.

Note that $\overline{NMI}$, since it can be accepted during Z80180 on-chip DMAC operation, can be used to externally interrupt DMA transfer. The $\overline{NMI}$ service routine can reactivate or abort the DMAC operation as required by the application.

For $\overline{NMI}$, special care must be taken to insure that interrupt inputs do not "overrun" the $\overline{NMI}$ service routine. Unlimited $\overline{NMI}$ inputs without a corresponding number of RETN instructions will eventually cause stack overflow.

Fig. 38 shows the use of $\overline{NMI}$ and RETN while Fig. 39 details $\overline{NMI}$ response timing. $\overline{NMI}$ is edge sensitive and the internally latched $\overline{NMI}$ falling edge is held until it is sampled. If the falling edge of $\overline{NMI}$ is latched before the falling edge of clock state prior to T_3 or T_1 in the last machine cycle, the internally latched $\overline{NMI}$ is sampled at the falling edge of the clock state prior to T_3 or T_1 in the last machine cycle and $\overline{NMI}$ acknowledge cycle begins at the end of the current machine cycle.
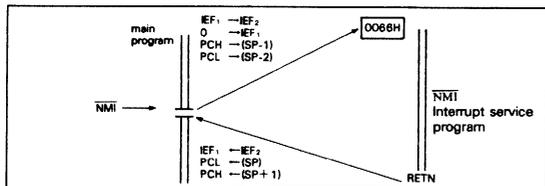
Figure 38. $\overline{NMI}$ Use
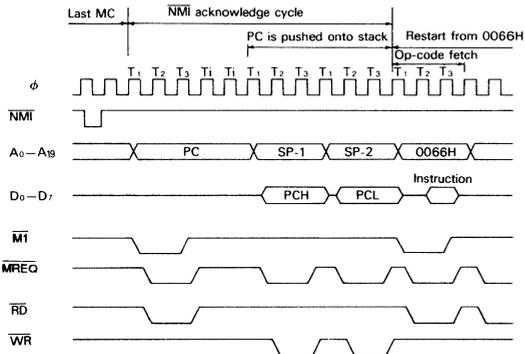
Figure 39. $\overline{NMI}$ Timing

$\overline{INT_0}$ - Maskable Interrupt Level 0

The next highest priority external interrupt after $\overline{NMI}$ is $\overline{INT_0}$. $\overline{INT_0}$ is sampled at the falling edge of the clock state prior to T_3 or T_1 in the last machine cycle. If $\overline{INT_0}$ is asserted LOW at the falling edge of the clock state prior to T_3 or T_1 in the last machine cycle, $\overline{INT_0}$ is accepted. The interrupt is masked if either the IEF1 flag or the ITE0 (Interrupt Enable 0) bit in ITC are reset to 0. Note that after RESET the state is as follows:

(1) IEF1 is 0, so INT_0 is masked.

(2) ITE0 is 1, so INT_0 is enabled by execution of the EI (Enable Interrupts) instruction.

The $\overline{INT_0}$ interrupt is unique in that 3 programmable interrupt response modes are available - Mode 0, Mode 1 and Mode 2. The specific mode is selected with the IM 0, IM 1 and IM 2 (Set Interrupt Mode) instructions. During RESET, the Z80180 is initialized to use Mode 0 for INT0. The 3 interrupt response modes for INT_0 are:

(1) Mode 0 - Instruction fetch from data bus.

(2) Mode 1 - Restart at logical address 0038H.

(3) Mode 2 - Low byte vector table address fetch from data bus.

$\overline{INT_0}$ Mode 0

During the interrupt acknowledge cycle, an instruction is fetched from the data bus (D0-D7) at the rising edge of T3. Often, this instruction is one of the eight single byte RST (RESTART) instructions which stack the PC and restart execution at a fixed logical address. However, multibyte instructions can be processed if the interrupt acknowledging device can provide a multibyte response. Unlike all other interrupts, the PC is not automatically stacked.

Note that TRAP interrupt will occur if an invalid instruction is fetched during Mode 0 interrupt acknowledge. (Reference Figure 40.)
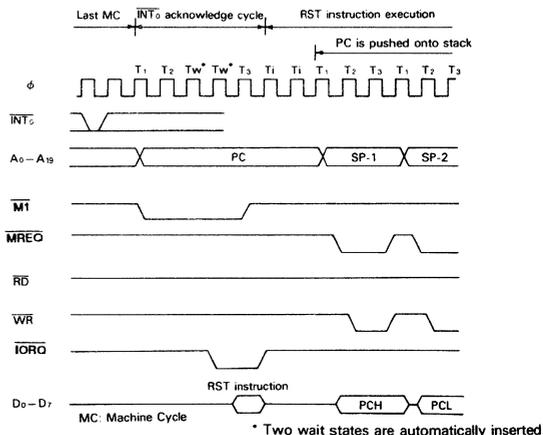
Figure 40. $\overline{INT_0}$ Mode 0 Timing

$\overline{INT_0}$ Mode 1

When $\overline{INT_0}$ is received, the PC is stacked and instruction execution restarts at logical address 0038H. Both IEF1 and IEF2 flags are reset to 0, disabling all maskable interrupts. The interrupt service routine should normally terminate with the EI (Enable Interrupts) instruction followed by the RETI (Return from Interrupt) instruction, to reenable the interrupts. Fig. 41 shows the use of $\overline{INT_0}$ (Mode 1) and RETI for the Mode 1 interrupt sequence.
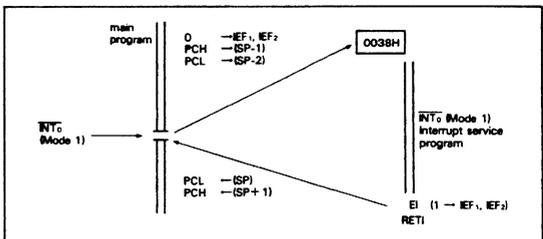
Figure 41. $\overline{INT_0}$ Mode 1 Interrupt Sequence

Fig. 42 shows $\overline{INT}_0$ Mode 1 timing.

$\overline{INT}_0$ Mode 2
This method determines the restart address by reading the contents of a table residing in memory. The vector table consists of up to 128 two-byte restart addresses stored in low byte, high byte order.

The vector table address is located on 256 byte boundaries in the 64K byte logical address space programmed in the 8-bit Interrupt Vector Register (I). Fig. 43 shows the $\overline{INT}_0$ Mode 2 Vector acquisition.
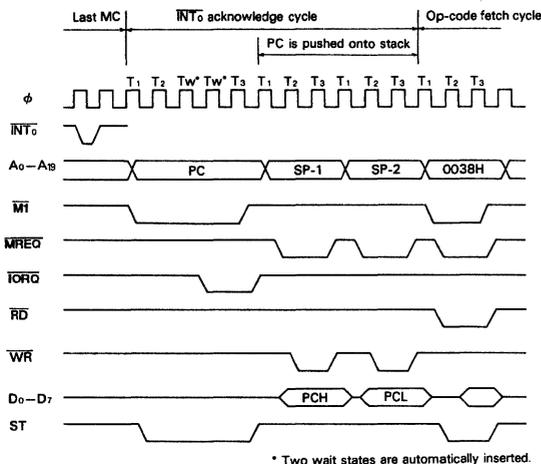
Figure 42. $\overline{INT}_0$ Mode 1 Timing
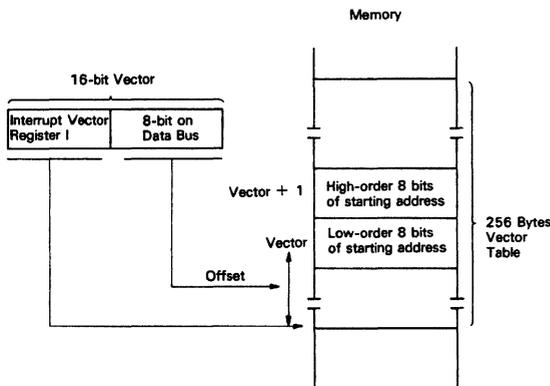
Figure 43. $\overline{INT}_0$ Mode 2 Vector Aquisition

During the $\overline{INT}_0$ Mode 2 acknowledge cycle, the low-order 8 bits of the vector is fetched from the data bus at the rising edge of T3 and the CPU acquires the 16-bit vector.

Next, the PC is stacked. Finally, the 16-bit restart address is fetched from the vector table and execution begins at that address.

Note that external vector acquisition is indicated by both $\overline{M1}$ and $\overline{IORQ}$ LOW. Two wait states (Tw) are automatically inserted for external vector fetch cycles.

During RESET the Interrupt Vector Register (I) is initialized to 00H and, if necessary, should be set to a different value prior to the occurrence of a Mode 2 $\overline{INT}_0$ interrupt. Fig. 44 shows $\overline{INT}_0$ interrupt Mode 2 Timing.
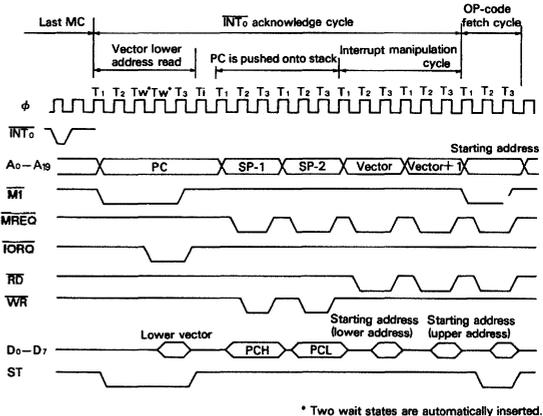
Figure 44. $\overline{INT}_0$ Interrupt Mode 2 Timing

$\overline{INT}_1$, $\overline{INT}_2$
The operation of external interrupts $\overline{INT}_1$ and $\overline{INT}_2$ is a vector mode similar to $\overline{INT}_0$ Mode 2. The difference is that $\overline{INT}_1$ and $\overline{INT}_2$ generate the low-order byte of vector table address using the IL (Interrupt Vector Low) register rather than fetching it from the data bus. This is also the interrupt response sequence used for all internal interrupts (except TRAP).

As shown in Fig. 45, the low-order byte of the vector table address has the most significant 3 bits of the software programmable IL register while the least significant 5 bits are a unique fixed value for each interrupt ($\overline{INT}_1$, $\overline{INT}_2$ and internal) source.
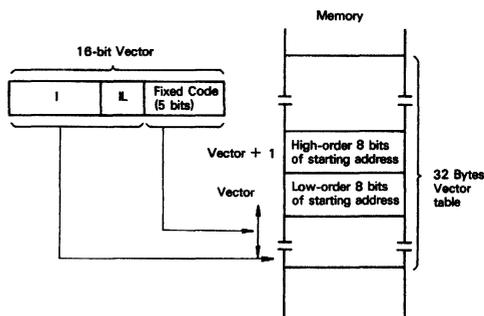
Figure 45. $\overline{INT}_1$, $\overline{INT}_2$ Vector Aquisition

$\overline{INT}_1$ and $\overline{INT}_2$ are globally masked by IEF1 = 0. Each is also individually maskable by respectively clearing the

ITE1 and ITE2 (bits 1,2) of the INT/TRAP control register to 0.
During RESET, IEF1, ITE1 and ITE2 bits are reset to 0.

Internal interrupts. Internal interrupts (except TRAP) use the same vectored response mode as $\overline{INT}_1$ and $\overline{INT}_2$ (Fig. 45). Internal interrupts are globally masked by IEF1 = 0. Individual internal interrupts are enabled/disabled by programming each individual I/O (PRT, DMAC, CSI/O, ASCI) control register. The lower vector of $\overline{INT}_1$, $\overline{INT}_2$ and internal interrupt are summarized in Table 6.

Interrupt Source	Priority	IL			Fixed Code				
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
$\overline{INT}_1$	Highest	•	•	•	0	0	0	0	0
$\overline{INT}_2$		•	•	•	0	0	0	1	0
PRT channel 0		•	•	•	0	0	1	0	0
PRT channel 1		•	•	•	0	0	1	1	0
DMA channel 0		•	•	•	0	1	0	0	0
DMA channel 1		•	•	•	0	1	0	1	0
CSI/O	∘	•	•	•	0	1	1	0	0
ASCI channel 0		•	•	•	0	1	1	1	0
ASCI channel 1	Lowest	•	•	•	1	0	0	0	0

* Programmable

Table 6. Vector Table

Interrupt Acknowledge Cycle Timing
Fig. 46 shows $\overline{INT}_1$, $\overline{INT}_2$, and internal interrupts timing. $\overline{INT}_1$ and $\overline{INT}_2$ are sampled at the falling edge of the clock state prior to T_2 or T_1 in the last machine cycle. If $\overline{INT}_1$ or $\overline{INT}_2$ is asserted LOW at the falling edge of clock state prior to T_3 or T_1 in the last machine cycle, the interrupt request is accepted.

Interrupt Sources During Reset:
Interrupt Vector Register (I). All bits are reset to 0.
Since I = 0 locates the vector tables starting at logical address 0000H, vectored interrupts ($\overline{INT}_0$ Mode 2, $\overline{INT}_1$, $\overline{INT}_2$, and internal interrupts) overlap with fixed restart interrupts like RESET (0), $\overline{NMI}$ (0066H), $\overline{INT}_0$ Mode 1 (0038H) and RST (0000H-0038H). The vector table(s) are built elsewhere in memory and located on 356 byte boundaries by reprogramming I with the LD I, A instruction.
IL Register. Bits 7 - 5 are reset to 0.
The IL Register can be programmed to locate the vector

table for $\overline{INT}_1$, $\overline{INT}_2$ and internal interrupts on 32 byte sub-boundaries within the 256 byte area specified by I.

IEF$_1$, IEF$_2$ Flags. Reset to 0.
Interrupts other than $\overline{NMI}$ and TRAP are disabled.

ITC Register. ITEO set to 1. ITE1, ITE2 reset to 0.
$\overline{INT}_0$ can be enabled by the EI instruction, which sets IEF1 = 1. To enable $\overline{INT}_1$ and $\overline{INT}_2$ also requires that the ITE1 and ITE2 bits be respectively set = 1 by writing to ITC.

I/O Control Registers. Interrupt enable bits reset to 0.
All Z80180 on-chip I/O (PRT, DMAC, CSI/O, ASCI) interrupts are disabled and can be individually enabled by writing to each I/O control register interrupt enable bit.

RETI Instruction Sequence:
When the EDH/4DH sequence is fetched by the Z80180, it is recognized as the RETI instruction sequence. The Z80180 will then refetch the RETI instruction with 4 T-states in the EDH cycle to allow the Z80 peripherals time to decode that cycle (See Figure 47). This allows the internal interrupt structure of the peripheral to properly decode the instruction and behave accordingly.

The M1E bit of the Operation Mode Control Register (OMCR) should be set to ' 0 · so that $\overline{M1}$ signal is active only during the refetch of the RETI instruction sequence. This is the desired operation when Z80 peripherals are connected to the Z80180.
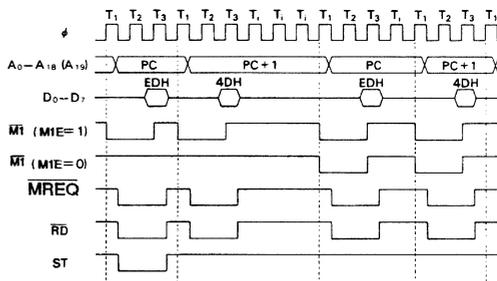
Figure 47. RETI Instruction Sequence

The RETI instruction takes 22 T-states and 10 machine cycles. Table 7 lists the conditions of all the control signals during this sequence for the Z80180.

Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$ IOC=1	IOC=0	$\overline{HALT}$	ST
1	T1-T3	1st Opcode	EDH	0	1	0	1	0	1	1	0
2	T1-T3	2nd Opcode	4DH	0	1	0	1	0	1	1	1
3	TI	don't care	3-state	1	1	1	1	1	1	1	1
4	TI	don't care	3-state	1	1	1	1	1	1	1	1
5	TI	don't care	3-state	1	1	1	1	1	1	1	1
6	T1-T3	1st Opcode	EDH	0	1	0	1	0	0	1	1
7	TI	don't care	3-state	1	1	1	1	1	1	1	1
8	T1-T3	2nd Opcode	4DH	0	1	0	1	0	1	1	1
9	T1-T3	SP	data	0	1	0	1	1	1	1	1
10	T1-T3	SP+1	data	0	1	0	1	1	1	1	1

Table 7. RETI Control Signal States
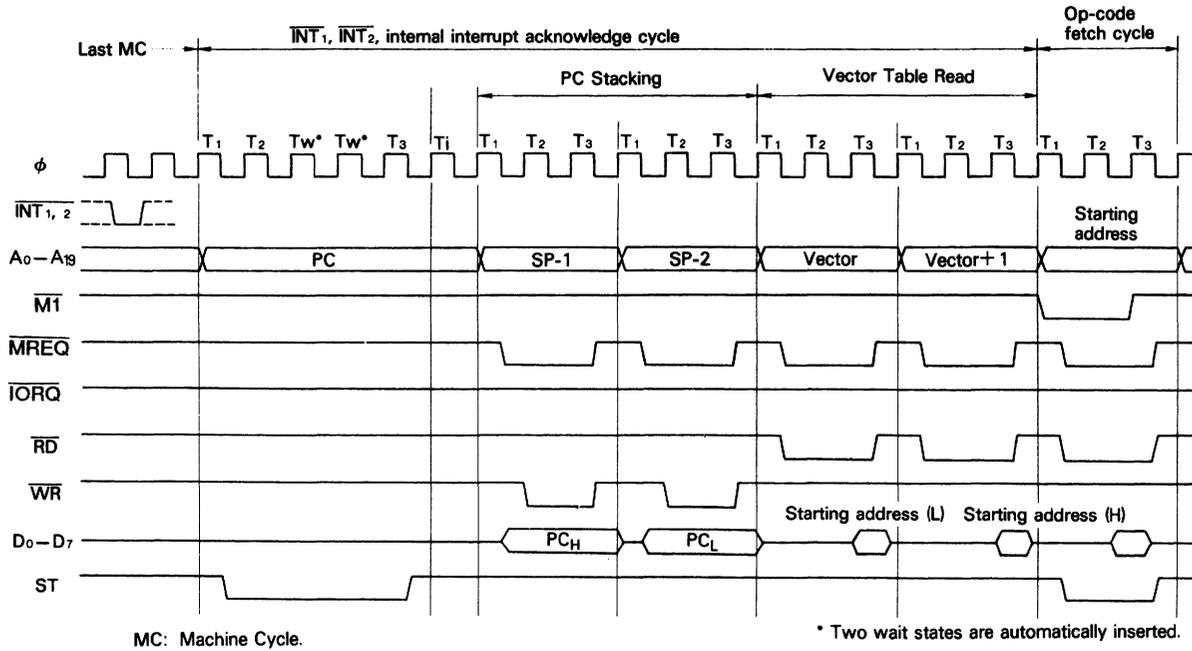
Figure 46. $\overline{INT}_1$, $\overline{INT}_2$, and Internal Interrupts Timing

Dynamic RAM Refresh Control

The Z80180 incorporates a dynamic RAM refresh control circuit including 8-bit refresh address generation and programmable refresh timing. This circuit generates asynchronous refresh cycles inserted at the programmable interval independent of CPU program execution. For systems which do not use dynamic RAM, the refresh function can be disabled.

When the internal refresh controller determines that a refresh cycle should occur, the current instruction is interrupted at the first breakpoint between machine cycles. The refresh cycle is inserted by placing the refresh address on A_0-A_7 and the $\overline{RFSH}$ output is driven LOW.

Refresh cycles may be programmed to be either 2 or 3 clock cycles in duration by programming the REFW (Refresh Wait) bit in the Refresh Control Register (RCR). Note that the external $\overline{WAIT}$ input and the internal wait state generator are not effective during refresh.

Fig. 48 shows the timing of a refresh cycle with a refresh wait (T_{RW}) cycle.

NOTE: * If three refresh cycles are specified, T_{RW}, is inserted.
Otherwise, T_{RW} is not inserted.
MC: Machine Cycle
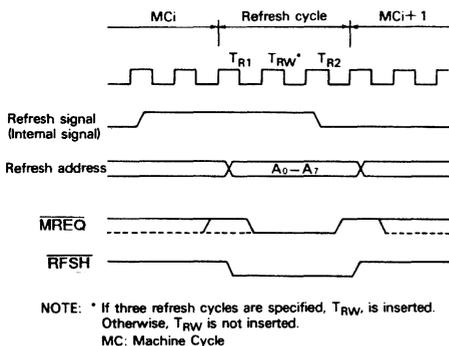
Figure 48. Refresh Cycle Timing

Refresh Control Register (RCR). The RCR specifies the interval and length of refresh cycles, while enabling or disabling the refresh function.

bit 7	6	5	4	3	2	1	0
REFE	REFW	–	–	–	–	CYC1	CYC0
R/W	R/W					R/W	R/W

Figure 49. Refresh Control Register (RCA: I/O Address = 36H)

REFE: Refresh Enable (bit 7)
REFE = 0 disables the refresh controller while REFE = 1 enables refresh cycle insertion. REFE is set to 1 during RESET.

REFW: Refresh Wait (bit 6)
REFW = 0 causes the refresh cycle to be two clocks in duration. REFW = 1 causes the refresh cycle to be three

clocks in duration by adding a refresh wait cycle (TRW). REFW is set to 1 during RESET.

CYC1, 0: Cycle Interval (bit 1, 0)
CYC1 and CYC0 specify the interval (in clock cycles) between refresh cycles. In the case of dynamic RAMs requiring 128 refresh cycles every 2 ms (or 256 cycles in every 4 ms), the required refresh interval is less than or equal to 15.625 μs. Thus, the underlined values indicate the best refresh interval depending on CPU clock frequency. CYC0 and CYC1 are cleared to 0 during RESET. Reference Table 8.

CYC1	CYC0	Insertion interval	Time interval				
			ϕ: 10 MHz	8 MHz	6 MHz	4 MHz	2.5 MHz
0	0	10 states	(1.0 μs)*	(1.25 μs)*	1.66 μs	2.5 μs	4.0 μs
0	1	20 states	(2.0 μs)*	(2.5 μs)*	3.3 μs	5.0 μs	8.0 μs
1	0	40 states	(4.0 μs)*	(5.0 μs)*	6.6 μs	10.0 μs	16.0 μs
1	1	80 states	(8.0 μs)*	(10.0 μs)*	13.3 μs	20.0 μs	32.0 μs

* calculated interval

Table 8. DRAM Refresh Intervals

Refresh Control And Reset. After RESET, based on the initialized value of RCR, refresh cycles will occur with an interval of 10 clock cycles and be 3 clock cycles in duration.

Dynamic Ram Refresh Operation Notes.

(1) Refresh Cycle insertion is stopped when the CPU is in the following states:

(a) During RESET.

(b) When the bus is released in response to $\overline{BUSREQ}$.

(c) During SLEEP mode.

(d) During $\overline{WAIT}$ states.

(2) Refresh cycles are suppressed when the bus is released in response to $\overline{BUSREQ}$. However, the refresh timer continues to operate. Thus, the time at which the first refresh cycle occurs after the Z80180 re-acquires the bus depends on the refresh timer and has no timing relationship with the bus exchange.

(3) Refresh cycles are suppressed during SLEEP mode. If a refresh cycle is requested during SLEEP mode, the refresh cycle request is internally "latched" (until replaced with the next refresh request). The "latched" refresh cycle is inserted at the end of the first machine cycle after SLEEP mode is exited. After this initial cycle, the time at which the next refresh cycle occurs depends on the refresh time and has no timing relationship with the exit from SLEEP mode.

(4) Regarding (2) and (3), the refresh address is incremented by one for each successful refresh cycle, not for

each refresh request. Thus, independent of the number of "missed" refresh requests, each refresh bus cycle will use a refresh address incremented by one from that of the previous refresh bus cycles.

DMA Controller (DMAC)

The Z80180 contains a two-channel DMA (Direct Memory Access) controller which supports high speed data transfer. Both channels (channel 0 and channel 1) have the following capabilities.

Memory Address Space. Memory source and destination addresses can be directly specified anywhere within the 1024K byte physical address space using 20-bit source and destination memory addresses. In addition, memory transfers can arbitrarily cross 64K byte physical address boundaries without CPU intervention.

I/O Address Space. I/O source and destination addresses can be directly specified anywhere within the 64K byte I/O address space (16-bit source and destination I/O addresses).

Transfer Length. Up to 64K bytes are transferred based on a 16-bit byte count register.

DREQ Input. Level and edge sense DREQ input detection are selectable.

TEND Output. Used to indicate DMA completion to external devices.

Transfer Rate. Each byte transfer can occur every 6 clock cycles. Wait states can be inserted in DMA cycles for slow memory or I/O devices. At the system clock (ϕ) = 6 MHz, the DMA transfer rate is as high as 1.0 megabytes/second (no wait states).

There is an additional feature disc for DMA interrupt request by DMA END. Each channel has the following additional specific capabilities.

Channel 0
Memory ↔ memory, memory ↔ I/O, memory ↔ memory mapped I/O transfers.

-Memory address increment, decrement, no-change.
-Burst or cycle steal memory to/from memory transfers.
-DMA to/from both ASCI channels.
-Higher priority than DMAC channel 1.

Channel 1
Memory ↔ I/O transfer.
Memory address increment, decrement

DMAC Registers
Each channel of the DMAC (channel 0, 1) has three registers specifically associated with that channel.

Channel 0
SAR0 - Source Address Register
DAR0 - Destination Address Register
BCR0 - Byte Count Register

Channel 1
MAR1 - Memory Address Register
IAR1 - I/O Address Register
BCR1 - Byte Count Register

The two channels share the following three additional registers in common.

DSTAT - DMA Status Register
DMODE - DMA Mode Register
DCNTL - DMA Control Register

DMAC Block Diagram. Fig. 50 shows the Z64180 DMAC Block Diagram.
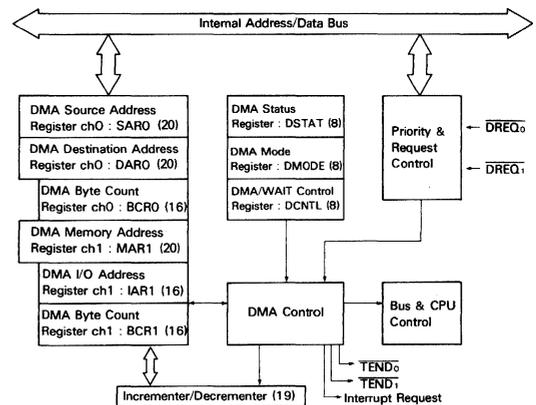
Figure 50. DMAC Block Diagram

DMAC Register Description.

DMA Source Address Register Channel 0 (SAR0: I/O Address = 20H to 22H)
Specifies the physical source address for channel 0 transfers. The register contains 20 bits and can specify up to 1024K byte memory addresses or up to 64K byte I/O addresses. Channel 0 source can be memory, I/O, or memory mapped I/O.

DMA Destination Address Register Channel 0 (DAR0: I/O Address = 23H to 25H)
Specifies the physical destination address for channel 0 transfers. The register contains 20 bits and can specify up

to 1024K byte memory addresses or up to 64K byte I/O addresses. Channel 0 destination can be memory, I/O, or memory mapped I/O.

DMA Byte Count Register Channel 0 (BCRO: I/O Address = 26H to 27H)

Specifies the number of bytes to be transferred. This register contains 16 bits and may specify up to 64K byte transfers. When one byte is transferred, the register is decremented by one. If "n" bytes are transferred, "n" is stored before the DMA operation.

DMA Memory Address Register Channel 1 (MAR1: I/O Address = 28H to 2AH)

Specifies the physical memory address for channel 1 transfers. This may be destination or source memory address. The register contains 20 bits and may specify up to 1024K byte memory address.

DMA I/O Address Register Channel 1 (IAR1: I/O Address = 2BH to 2CH)

Specifies the I/O address for channel 1 transfers. This may be destination or source I/O address. The register contains 16 bits and may specify up to 64K byte I/O addresses.

DMA Byte Count Register Channel 1 (BCR1: I/O Address = 2EH to 2FH)

Specifies the number of bytes to be transferred. This register contains 16 bits and may specify up to 64K byte transfers. When one byte is transferred, the register is decremented by one.

DMA Status Register (DSTAT)

DSTAT is used to enable and disable DMA transfer and DMA termination interrupts. DSTAT also determines DMA transfer status, i.e. completed or in progress.

bit 7	6	5	4	3	2	1	0
DE1	DE0	DWE1	DWE0	DIE1	DIE0	—	DME
R/W	R/W	W	W	R/W	R/W		R
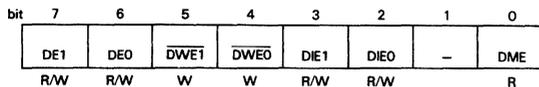
Figure 51. DMA Status Register (DSTAT : I/O Address = 30H)

DE1: DMA Enable Channel 1 (bit 7)

When DE1 = 1 and DME = 1, channel 1 DMA is enabled. When a DMA transfer terminates (BCR1 = 0), DE1 is reset to 0 by the DMAC. When DE1 = 0 and the DMA interrupt is enabled (DIE1 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DE1, DWE1 is written with 0 during the same register write access. Writing DE1 to 0 disables channel 1 DMA, but DMA is restartable. Writing DE1 to 1 enables channel 1 DMA and automatically sets DME (DMA Main Enable) to 1. DE1 is cleared to 0 during RESET.

DE0: DMA Enable Channel 0 (bit 6)

When DE0 = 1 and DME = 1, channel 0 DMA is enabled. When a DMA transfer terminates (BCR0 = 0), DE0 is reset to 0 by the DMAC. When DE0 = 0 and the DMA interrupt is enabled (DIE0 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DE0, DWE0 should be written with 0 during the same register write access. Writing DE0 to 0 disables channel 0 DMA. Writing DE0 to 1 enables channel 0 DMA and automatically sets DME (DMA Main Enable) to 1. DE0 is cleared to 0 during RESET.

DWE1: DE1 Bit Write Enable (bit 5)

When performing any software write to DE1, DWE1 should be written with 0 during the same access. DWE1 write value of 0 is not held and DWE1 is always read as 1.

DWE0: DE0 Bit Write Enable (bit 4)

When performing any software write to DE0, DWE0 should be written with 0 during the same access. DWE0 write value of 0 is not held and DWE0 is always read as 1.

DIE1: DMA Interrupt Enable Channel 1 (bit 3)

When DIE0 is set to 1, the termination channel 1 DMA transfer (indicated when DE1 = 0) causes a CPU interrupt request to be generated. When DIE1 = 0, the channel 1 DMA termination interrupt is disabled. DIE1 is cleared to 0 during RESET.

DIE0: DMA Interrupt Enable Channel 0 (bit 2)

When DIE0 is set to 1, the termination channel 0 of DMA transfer (indicated when DE0 = 0) causes a CPU interrupt request to be generated. When DIE0 = 0, the channel 0 DMA termination interrupt is disabled. DIE0 is cleared to 0 during RESET.

DME: DMA Main Enable (bit 0)

A DMA operation is only enabled when its DE bit (DE0 for channel 0, DE1 for channel 1) and the DME bit are set to 1.

When NMI occurs, DME is reset to 0, thus disabling DMA activity during the NMI interrupt service routine. To restart DMA, DE0 and/or DE1 should be written with 1 (even if the contents are already 1). This automatically sets DME to 1, allowing DMA operations to continue. Note that DME cannot be directly written. It is cleared to 0 by NMI or indirectly set to 1 by setting DE0 and/or DE1 to 1. DME is cleared to 0 during RESET.

DMA Mode Register (DMODE)

DMODE is used to set the addressing and transfer mode for channel 0.

bit 7	6	5	4	3	2	1	0
—	—	DM1	DM0	SM1	SM0	MMOD	—
		R/W	R/W	R/W	R/W	R/W	
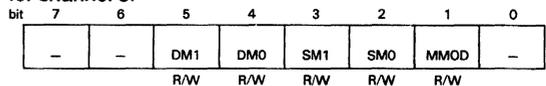
Figure 52. DMA Mode Register (DMODE : I/O Address = 31H)

DM1, DM0: Destination Mode Channel 0 (bits 5,4)

Specifies whether the destination for channel 0 transfers is memory, I/O or memory mapped I/O and the corresponding address modifier. DM1 and DM0 are cleared to 0 during RESET.

DM1	DM0	Memory/I/O	Address Increment/Decrement
0	0	Memory	+1
0	1	Memory	−1
1	0	Memory	fixed
1	1	I/O	fixed

Table 9: Channel 0 Destination

SM1, SM0: Source Mode Channel 0 (bits 3,2)
Specifies whether the source for channel 0 transfers is memory, I/O, or memory mapped I/O and the corresponding address modifier. SM1 and SM0 are cleared to 0 during RESET.

SM1	SM0	Memory/I/O	Address Increment/Decrement
0	0	Memory	+1
0	1	Memory	−1
1	0	Memory	fixed
1	1	I/O	fixed

Table 10: Channel 0 Source

Table 11 shows all DMA transfer mode combinations of DM0, DM1, SM0, SM1. Since I/O to/from I/O transfers are not implemented, 12 combinations are available.

DM1	DM0	SM1	SM0	Transfer Mode	Address Increment/Decrement
0	0	0	0	Memory→Memory	SAR0+1, DAR0+1
0	0	0	1	Memory→Memory	SAR0−1, DAR0+1
0	0	1	0	Memory*→Memory	SAR0 fixed, DAR0+1
0	0	1	1	I/O→Memory	SAR0 fixed, DAR0+1
0	1	0	0	Memory→Memory	SAR0+1, DAR0−1
0	1	0	1	Memory→Memory	SAR0−1, DAR0−1
0	1	1	0	Memory*→Memory	SAR0 fixed, DAR0−1
0	1	1	1	I/O→Memory	SAR0 fixed, DAR0−1
1	0	0	0	Memory→Memory*	SAR0+1, DAR0 fixed
1	0	0	1	Memory→Memory*	SAR0−1, DAR0 fixed
1	0	1	0	reserved	
1	0	1	1	reserved	
1	1	0	0	Memory→I/O	SAR0+1, DAR0 fixed
1	1	0	1	Memory→I/O	SAR0−1, DAR0 fixed
1	1	1	0	reserved	
1	1	1	1	reserved	

* : includes memory mapped I/O

Table 11. Transfer Mode Combinations

MMOD: Memory Mode Channel 0 (bit 1)
When channel 0 is configured for memory to/from memory transfers, the external $\overline{DREQ_0}$ input is not used to control the transfer timing. Instead, two automatic transfer timing modes are selectable - burst (MMOD = 1) and cycle steal (MMOD = 0). For burst memory to/from memory transfers, the DMAC takes control of the bus continuously until the DMA transfer completes (as shown by the byte count register = 0). In cycle steal mode, the CPU is given a cycle for each DMA byte transfer cycle until the transfer is completed.

For channel 0 DMA with I/O source or destination, the $\overline{DREQ_0}$ input times the transfer and thus MMOD is ignored. MMOD is cleared to 0 during RESET.

DMA/WAIT Control Register (DCNTL)
DCNTL controls the insertion of wait states into DMAC (and CPU) accesses of memory or I/O. Also, the DMA request mode for each DREQ ($\overline{DREQ_0}$ and $\overline{DREQ_1}$) input is defined as level or edge sense. DCNTL also sets the DMA transfer mode for channel 1, which is limited to memory to/from I/O transfers.

bit	7	6	5	4	3	2	1	0
	MWI1	MWI0	IWI1	IWI0	DMS1	DMS0	DIM1	DIM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
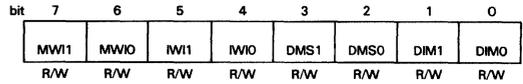
Figure 53. DMA/WAIT Control Register (DCNTL : I/O Address = 32H)

MWI1, MWI0: Memory Wait Insertion (bits 7-6)
Specifies the number of wait states introduced into CPU or DMAC memory access cycles. MWI1 and MWI0 are set to 1 during RESET. See section on Wait State Generator for details.

IWI1, IWI0: I/O Wait Insertion (bits 5-4)
Specifies the number of wait states introduced into CPU or DMAC I/O access cycles. IWI1 and IWI0 are set to 1 during RESET. See section on Wait State Generator for details.

DMS1, DMS0: DMA Request Sense (bits 3-2)
DMS1 and DMS0 specify the DMA request sense for channel 0 ($\overline{DREQ0}$) and channel 1 ($\overline{DREQ1}$), respectively. When reset to 0, the input is level sense. When set to 1, the input is edge sense. DMS1 and DMS0 are cleared to 0 during RESET.

DIM1, DIM0: DMA Channel 1 I/O and Memory Mode (bits 1-0)
Specifies the source/destination and address modifier for channel 1 memory to/from I/O transfer modes. DIM1 and DIM0 are cleared to 0 during RESET.

DIM1	DIM0	Transfer Mode	Address Increment/Decrement
0	0	Memory→I/O	MAR1+1, IAR1 fixed
0	1	Memory→I/O	MAR1−1, IAR1 fixed
1	0	I/O→Memory	IAR1 fixed, MAR1+1
1	1	I/O→Memory	IAR1 fixed, MAR1−1

Table 12: Channel 1 Transfer Mode

DMA Operation
This section discusses the three DMA operation modes for channel 0, memory to/from memory, memory to/from I/O, and memory to/from memory mapped I/O. In addition, the operation of channel 0 DMA with the on-chip ASCI (Asynchronous Serial Communication Interface) as well as Channel 1 DMA are described.

Memory ↔ Memory - Channel 0
For memory to/from memory transfers, the external DREQ0 input is not used for DMA transfer timing. Rather,

the DMA operation is timed in one of two programmable modes - burst or cycle steal. In both modes, the DMA operation automatically proceeds until termination (shown by byte count- BCR0) = 0.

In burst mode, the DMA operation proceeds until termination. In this case, the CPU cannot perform any program execution until the DMA operation is completed.

In cycle steal mode, the DMA and CPU operation are alternated after each DMA byte transfer until the DMA is completed. The sequence...

1 CPU Machine Cycle
DMA Byte Transfer

... is repeated until DMA is completed. Fig. 54. shows cycle steal mode DMA timing.
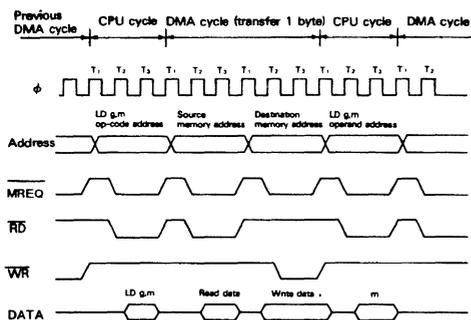
Figure 54. DMA Timing-Cycle Steal Mode

To initiate memory ↔ memory DMA transfer for channel 0, perform the following operations.

(1) Load the memory source and destination address into SAR0 and DAR0.

(2) Specify memory to/from memory mode and address increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.

(3) Load the number of bytes to transfer in BCR0.

(4) Specify burst or cycle steal mode in the MMOD bit of DCNTL.

(5) Program DE0 = 1 (with $\overline{DWE0}$ = 0 in the same access) in DSTAT and the DMA operation starts one machine cycle later. If interrupt occurs at the same time, the DIE0 bit should be set to 1.

Memory ↔ I/O (Memory Mapped I/O) - Channel 0
For memory ↔ I/O (and memory to/from memory mapped I/O) the $\overline{DREQ0}$ input is used to time the DMA transfers. In addition, the $\overline{TEND0}$ (Transfer End) output is used to indicate the last (byte count register BCR0 = 00H) transfer.

The $\overline{DREQ0}$ input can be programmed as level or edge sensitive.

When level sense is programmed, the DMA operation begins when $\overline{DREQ0}$ is sampled LOW. If $\overline{DREQ0}$ is sampled HIGH, after the next DMA byte transfer, control is relinquished to the Z80180 CPU. As shown in Fig. 33, $\overline{DREQ0}$ is sampled at the rising edge of the clock cycle prior to T3, i.e. either T2 or Tw.
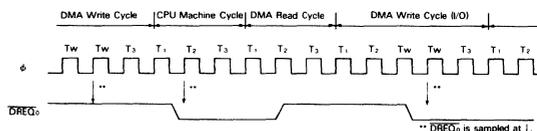
Figure 55. CPU Operation and DMA Operation (DREQ0 is Programmed for Level Sense)

When edge sense is programmed, DMA operation begins at the falling edge of $\overline{DREQ0}$. If another falling edge is detected before the rising edge of the clock prior to T3 during DMA write cycle (i.e. T_2 or Tw), the DMAC continues operating. If an edge is not detected, the CPU is given control after the current byte DMA transfer completes. The CPU will continue operating until a $\overline{DREQ0}$ falling edge is detected before the rising edge of the clock prior to T_3 at which time the DMA operation will (re)start. Fig. 56 shows the edge sense DMA timing.
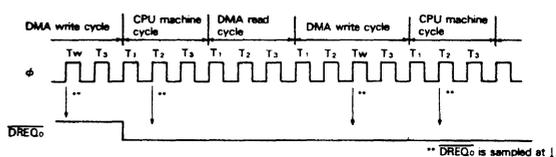
Figure 56. CPU Operation and DMA Operation (DREQ0 is Programmed for Edge Sense)

During the transfers for channel 0, the $\overline{TEND0}$ output goes LOW synchronous with the write cycle of the last (BCR0 = 00H) DMA transfer (Reference Fig. 57).
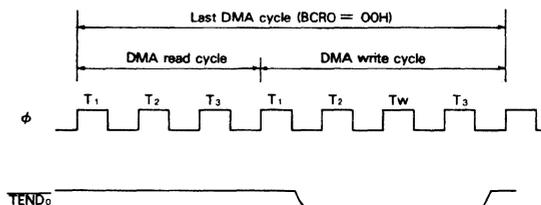
Figure 57. TEND0 Output Timing

The $\overline{DREQ0}$ and $\overline{TEND0}$ pins are programmably multiplexed with the CKA0 and CKA1 ASCI clock input/outputs. However, when DMA channel 0 is programmed for memory ↔ I/O (and memory ↔ memory mapped I/O) transfers, the CKA0/$\overline{DREQ0}$ pin automatically functions as input pin even if it has been programmed as output pin for

CKA0. And the CKA1/$\overline{TEND_0}$ pin functions as output pin for $\overline{TEND_0}$ by setting CKA1D to 1 in CNTLA1.

To initiate memory ↔ I/O (and memory ↔ memory mapped I/O) DMA transfer for channel 0, perform the following operations:

(1) Load the memory and I/O or memory mapped I/O source and destination addresses into SAR0 and DAR0. Note that I/O addresses (not memory mapped I/O) are limited to 16 bits (A_0-A_{15}). Make sure that bits A_{16}, A_{17} and A_{19} are 0 (A_{18} is a don't care) to correctly enable the external $\overline{DREQ_0}$ input.

(2) Specify memory ↔ I/O or memory ↔ memory mapped I/O mode and address increment/decrement in the SM0, SM1, DM0, and DM1 bits of DMODE.

(3) Load the number of bytes to transfer in BCR0.

(4) Specify whether $\overline{DREQ_0}$ is edge or level sense by programming the DMS0 bit of DCNTL.

(5) Enable or disable DMA termination interrupt with the DIE0 bit in DSTAT.

(6) Program DE0 = 1 (with DWE0 = 0 in the same access) in DSTAT and the DMA operation begins under the control of the $\overline{DREQ_0}$ input.

Memory ↔ ASCI - Channel 0

Channel 0 has extra capability to support DMA transfer to/from the on-chip two channel ASCI. In this case, the external $\overline{DREQ_0}$ input is not used for DMA timing. Rather, the ASCI status bits are used to generate an internal $\overline{DREQ_0}$. The TDRE (Transmit Data Register Empty) bit and the RDRF (Receive Data Register Full) bit are used to generate an internal $\overline{DREQ_0}$ for ASCI transmission and reception respectively.

To initiate memory ↔ ASCI DMA transfer, perform the following operations:

(1) Load the source and destination addresses into SAR0 and DAR0. Specify the I/O (ASCI) address as follows:

Bits A_0-A_7 should contain the address of the ASCI channel transmitter or receiver (I/O addresses 6H-9H).

Bits A_8-A_{15} should equal 0.

Bits A_{17}-A_{16} should be set according to Table 13 to enable use of the appropriate ASCI status bit as an internal DMA request.

SAR18	SAR17	SAR16	DMA Transfer Request
X	0	0	$\overline{DREQ_0}$
X	0	1	RDRF (ASCI channel 0)
X	1	0	RDRF (ASCI channel 1)
X	1	1	reserved

X: Don't care

DAR18	DAR17	DAR16	DMA Transfer Request
X	0	0	$\overline{DREQ_0}$
X	0	1	TDRE (ASCI channel 0)
X	1	0	TDRE (ASCI channel 1)
X	1	1	reserved

X: Don't care

Table 13: DMA Request

(2) Specify memory ↔ I/O transfer mode and address increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.

(3) Load the number of bytes to transfer in BCR0.

(4) The DMA request sense mode (DMS0 bit in DCNTL) MUST be specified as "edge sense".'

(5) Enable or disable DMA termination interrupt with the DIE0 bit in DSTAT.

(6) Program DE0 = 1 (with DWE0 = 0 in the same access) in DSTAT and the DMA operation with the ASCI begins under control of the ASCI generated internal DMA request.

The ASCI receiver or transmitter using DMA is initialized to allow the first DMA transfer to begin.

The ASCI receiver must be "empty" as shown by RDRF = 0.

The ASCI transmitter must be "full" as shown by TDRE = 0. Thus, the first byte is written to the ASCI Transmit Data Register under program control. The remaining bytes are transferred using DMA.

Channel 1 DMA. DMAC Channel 1 performs memory to/from I/O transfers. Except for different registers and status/control bits, operation is exactly the same as described for channel 0 memory to/from I/O DMA.

To initiate DMA channel 1 memory ↔ I/O transfer perform the following operations:

(1) Load the memory address (20 bits) into MAR1.

(2) Load the I/O address (16 bits) into IAR1.

(3) Program the source/destination and address increment/decrement mode using the DIM1 and DIM0 bits in DCNTL.

(4) Specify whether $\overline{DREQ1}$ is level or edge sense in the DMS1 bit in DCNTL.

(5) Enable or disable DMA termination interrupt with the DIE1 bit in DSTAT.

(6) Program DE1 = 1 (with $\overline{DWE1}$ = 0 in the same access) in DSTAT and the DMA operation with the external I/O device begins using the external $\overline{DREQ1}$ input and $\overline{TEND}_1$ output.

DMA Bus Timing. When memory (and memory mapped I/O) is specified as a source or destination, $\overline{MREQ}$ goes LOW during the memory access. When I/O is specified as a source or destination, $\overline{IORQ}$ goes LOW during the I/O access.

When I/O (and memory mapped I/O) is specified as a source or destination, the DMA timing is controlled by the external $\overline{DREQ}$ input and the $\overline{TEND}$ output indicates DMA termination. Note that external I/O devices may not overlap addresses with internal I/O and control registers, even using DMA.

For I/O accesses, one wait state is automatically inserted. Additional wait states can be inserted by programming the on-chip wait state generator or using the external $\overline{WAIT}$ input. Note that for memory mapped I/O accesses, this automatic I/O wait state is not inserted.

For memory to memory transfers (channel 0 only), the external $\overline{DREQ}_0$ input is ignored. Automatic DMA timing is programmed as either burst or cycle steal.

When a DMA memory address carry/borrow between bits A15 and A16 of the address bus occurs (when crossing 64K byte boundaries), the minimum bus cycle is extended to 4 clocks by automatic insertion of one internal Ti state.

DMAC Channel Priority. For simultaneous $\overline{DREQ}_0$ and $\overline{DREQ}_1$ requests, channel 0 has priority over channel 1. When channel 0 is performing a memory to/from memory transfer, channel 1 cannot operate until the channel 0 operation has terminated. If channel 1 is operating, channel 0 cannot operate until channel 1 releases control of the bus.

DMAC and BUSREQ, BUSACK. The $\overline{BUSREQ}$ and $\overline{BUSACK}$ inputs allow another bus master to take control of the Z80180 bus. $\overline{BUSREQ}$ and $\overline{BUSACK}$ have priority over the on-chip DMAC and suspends DMAC operation. The DMAC releases the bus to the external bus master at the breakpoint of the DMAC memory or I/O access. Since a single byte DMAC transfer requires a read and a write cycle, it is possible for the DMAC to be suspended after the DMAC read, but before the DMAC write. Hence, when the external master releases the Z80180 bus ($\overline{BUSREQ}$ HIGH), the on-chip DMAC correctly continues the suspended DMA operation.

DMAC Internal Interrupts. Fig. 58 illustrates the internal DMA interrupt request generation circuit.
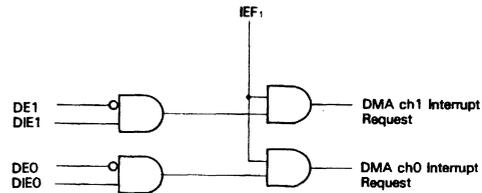
Figure 58. DMA Interrupt Request Generation

DE0 and DE1 are automatically cleared to 0 by the Z80180 at the completion (byte count =0) of a DMA operation for channel 0 and channel 1, respectively. They remain 0 until a 1 is written. Since DE0 and DE1 use level sense, an interrupt occurs if the CPU IEF$_1$ flag is set to 1. Therefore, the DMA termination interrupt service routine should disable further DMA interrupts (by programming the channel DIE bit = 0) before enabling CPU interrupts (i.e. IEF1 is set to 1). After reloading the DMAC address and count registers, the DIE bit can be set to 1 to reenable the channel interrupt, and at the same time DMA can resume by programming the channel DE bit = 1.

DMAC and $\overline{NMI}$. $\overline{NMI}$, unlike all other interrupts, automatically disables DMAC operation by clearing the DME bit of DSTAT. Thus, the $\overline{NMI}$ interrupt service routine responds to time critical events without delay due to DMAC bus usage. Also, $\overline{NMI}$ can be effectively used as an external DMA abort input, recognizing that both channels are suspended by the clearing of DME.

If the falling edge of $\overline{NMI}$ occurs before the falling clock of the state prior to T3 (T2 or Tw) of the DMA write cycle, the DMAC is suspended and the CPU starts the $\overline{NMI}$ response at the end of the current cycle.
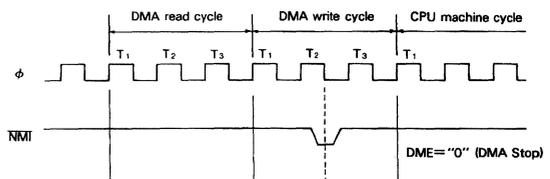
Figure 59. $\overline{NMI}$ and DMA Operation

By setting a channel's DE bit to 1, the channel's operation is restarted and DMA correctly resumes from its suspended point by $\overline{NMI}$. (Reference Fig. 59)

DMAC and RESET. During RESET the bits in DSTAT, DMODE, and DCNTL are initialized as stated in their in-

dividual register descriptions. Any DMA operation in progress is stopped, allowing the CPU to use the bus to perform the RESET sequence. However, the address register (SAR, DAR0, MAR1, IAR1) and byte count register (BCR0, BCR1) contents are not changed during RESET.

Asynchronous Serial Communication Interface (ASCI)

The Z80180 on-chip ASCI has two independent full-duplex channels. Based on full programmability of the following functions, the ASCI directly communicates with a wide variety of standard UARTs (Universal Asynchronous Receiver/Transmitter) including the Z8440 SIO and the Z8530 SCC.

The key functions for ASCI are shown below. Each channel is independently programmable.

-Full-duplex communication.
-7- or 8-bit data length.
-Program controlled 9th data bit for multiprocessor com munication.
-1 or 2 stop bits.
-Odd, even, no parity.
-Parity, overrun, framing error detection.
-Programmable baud rate generator, /16 and /64 modes.
-Speed to 38.4K bits per second (CPU fc = 6.144 MHz).
-Modem control signals - Channel 0 has $\overline{DCD0}$, $\overline{CTS0}$ and $\overline{RTS0}$ Channel 1 has $\overline{CTS1}$.
-Programmable interrupt condition enable and disable.
-Operation with on-chip DMAC.

ASCI Block Diagram. Figure 60 shows the ASCI block diagram.
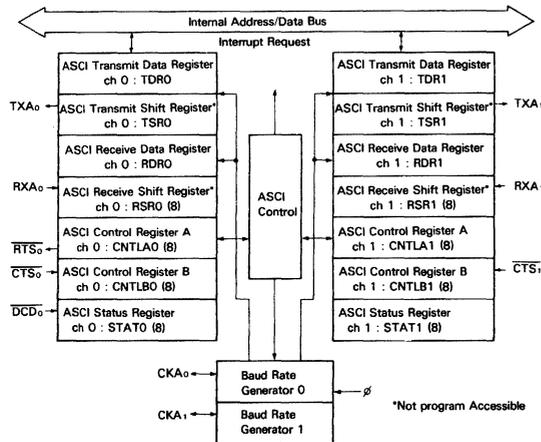
Figure 60. ASCI Block Diagram

ASCI Register Description. The following subparagraphs explaing the various functions of the ASCI registers.

ASCI Transmit Shift Register 0, 1 (TSR0, 1)
When the ASCI Transmit Shift Register receives data from the ASCI Transmit Data Register (TDR), the data is shifted out to the TXA pin. When transmission is completed, the next byte (if available) is automatically loaded from TDR into TSR and the next transmission starts. If no data is available for transmission, TSR idles by outputting a continuous HIGH level. This register is not program accessible.

ASCI Transmit Data Register 0, 1 (TDR0, 1: I/O Address = 06H, 07H)
Data written to the ASCI Transmit Data Register is transferred to the TSR as soon as TSR is empty. Data can be written while TSR is shifting out the previous byte of data. Thus, the ASCI transmitter is double buffered.

Data can be written into and read from the ASCI Transmit Data Register.

If data is read from the ASCI Transmit Data Register, the ASCI data transmit operation will not be affected by this read operation.

ASCI Receive Shift Register 0, 1 (RSR0, 1)
This register receives data shifted in on the RXA pin. When full, data is automatically transferred to the ASCI Receive Data Register (RDR) if it is empty. If RSR is not empty when the next incoming data byte is shifted in, an overrun error occurs. This register is not program accessible.

ASCI Receive Data Register 0, 1 (RDR0, 1: I/O Address = 08H, 09H)
When a complete incoming data byte is assembled in RSR, it is automatically transferred to the RDR if RDR is empty. The next incoming data byte can be shifted into RSR while RDR contains the previous received data byte. Thus, the ASCI receiver is double buffered.

The ASCI Receive Data Register is a read-only-register. However, if RDRF = 0, data can be written into the ASCI Receive Data Register, and the data can be read.

ASCI Status Register 0, 1 (STAT0, 1)
Each channel status register allows interrogation of ASCI communication, error and modem control signal status, and enabling or disabling of ASCI interrupts.

ASCI Status Register 0 (STAT0 : I/O Address = 04H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	$\overline{DCD_0}$	TDRE	TIE
	R	R	R	R	R/W	R	R	R/W

ASCI Status Register 1 (STAT1 : I/O Address = 05H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	CTS1E	TDRE	TIE
	R	R	R	R	R/W	R/W	R	R/W
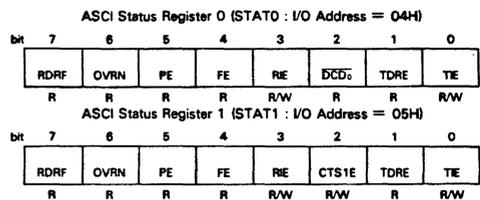
Figure 61. ASCI Status Registers

RDRF: Receive Data Register Full (bit 7)

RDRF is set to 1 when an incoming data byte is loaded into RDR. Note that if a framing or parity error occurs, RDRF is still set and the receive data (which generated the error) is still loaded into RDR. RDRF is cleared to 0 by reading RDR, when the DCD_0 input is HIGH, in IOSTOP mode and during RESET.

OVRN: Overrun Error (bit 6)

OVRN is set to 1 when RDR is full and RSR becomes full. OVRN is cleared to 0 when the EFR bit (Error Flag Reset) of CNTLA is written to 0, when DCD_0 is HIGH, in IOSTOP mode, and during RESET.

PE: Parity Error (bit 5)

PE is set to 1 when a parity error is detected on an incoming data byte and ASCI parity detection is enabled (the MOD1 bit of CNTLA is set to 1). PE is cleared to 0 when the EFR bit (Error Flag Reset) of CNTLA is written to 0, when DCD_0 is HIGH, in IOSTOP mode, and during RESET.

FE: Framing Error (bit 4)

If a receive data byte frame is delimited by an invalid stop bit (i.e. 0, should be 1), FE is set to 1. FE is cleared to 0 when the EFR bit (Error Flag Reset) of CNTLA is written to 0, when DCD_0 is HIGH, in IOSTOP mode, and during RESET.

RIE: Receive Interrupt Enable (bit 3)

RIE should be set to 1 to enable ASCI receive interrupt requests. When RIE is 1, if any of the flags RDRF, OVRN, PE, or FE become set to 1, an interrupt request is generated. For channel 0, an interrupt is also generated by the transition of the external DCD_0 input from LOW to HIGH. RIE is cleared to 0 during RESET.

$\overline{DCD}_0$: Data Carrier Detect (bit 2 STAT0)

Channel 0 has an external $\overline{DCD_0}$ input pin. The $\overline{DCD_0}$ bit is set to 1 when the $\overline{DCD_0}$ input is HIGH. It is cleared to 0 on the first read of STAT0 following the $\overline{DCD_0}$ input transition from HIGH to LOW and during RESET. When $\overline{DCD_0}$ = 1, receiver unit is reset and receiver operation is inhibited.

CTS1E: Channel 1 $\overline{CTS}$ Enable (bit 2 STAT1)

Channel 1 has an external $\overline{CTS1}$ input (pin 52) which is multiplexed with the receive data pin (RXS) for the CSI/O (Clocked Serial I/O Port). Setting CTS1E to 1 selects the $\overline{CTS1}$ function and clearing CTS1E to 0 selects the RXS function.

TDRE: Transmit Data Register Empty (bit 1)

TDRE = 1 indicates that the TDR is empty and the next transmit data byte is written to TDR. After the byte is written to TDR, TDRE is cleared to 0 until the ASCI transfers the byte from TDR to the TSR and then TDRE is again set to 1. TDRE is set to 1 in IOSTOP mode and during RESET. When the external $\overline{CTS}$ input is HIGH, TDRE is reset to 0.

TIE: Transmit Interrupt Enable (bit 0)

TIE should be set to 1 to enable ASCI transmit interrupt re-

quests. If TIE = 1, an interrupt will be requested when TDRE = 1. TIE is cleared to 0 during RESET.

ASCI Control Register A0, 1 (CNTLA0, 1). Each ASCI channel Control Register A configures the major operating modes such as receiver/transmitter enable and disable, data format, and multiprocessor communication mode.

ASCI Control Register A 0 (CNTLA0 : I/O Address = 00H)

bit 7	6	5	4	3	2	1	0
MPE	RE	TE	$\overline{RTS_0}$	MPBR/ EFR	MOD2	MOD1	MOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASCI Control Register A 1 (CNTLA1 : I/O Address = 01H)

bit 7	6	5	4	3	2	1	0
MPE	RE	TE	CKA1D	MPBR/ EFR	MOD2	MOD1	MOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
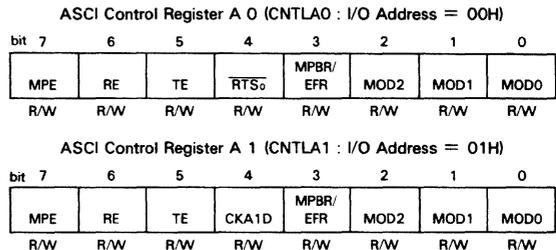
Figure 62. ASCI Control Register A

MPE: Multi-Processor Mode Enable (bit 7)

The ASCI has a multiprocessor communication mode which utilizes an extra data bit for selective communication when a number of processors share a common serial bus. Multiprocessor data format is selected when the MP bit in CNTLB is set to 1. If multiprocessor mode is not selected (MP bit in CNTLB = 0), MPE has no effect. If multiprocessor mode is selected, MPE enables or disables the "wake-up" feature as follows. If MPE is set to 1, only received bytes in which the MPB (multiprocessor bit) = 1 can affect the RDRF and error flags. Effectively, other bytes (with MPB = 0) are "ignored" by the ASCI. If MPE is reset to 0, all bytes, regardless of the state of the MPB data bit, affect the REDR and error flags. MPE is cleared to 0 during RESET.

RE: Receiver Enable (bit 6)

When RE is set to 1, the ASCI receiver is enabled. When RE is reset to 0, the receiver is disabled and any receive operation in progress is interrupted. However, the RDRF and error flags are not reset and the previous contents of RDRF and error flags are held. RE is cleared to 0 in IOSTOP mode during RESET.

TE: Transmitter Enable (bit 5)

When TE is set to 1, the ASCI transmitter is enabled. When TE is reset to 0, the transmitter is disabled and any transmit operation in progress is interrupted. However, the TDRE flag is not reset and the previous contents of TDRE are held. TE is cleared to 0 in IOSTOP mode during RESET.

$\overline{RTS}_0$ - Request to Send Channel 0 (bit 4 in CNTLA0)

When $\overline{RTS_0}$ is reset to 0, the $\overline{RTS_0}$ output pin will go LOW. When $\overline{RTS_0}$ is set to 1, the $\overline{RTS_0}$ output immediately goes HIGH. $\overline{RTS_0}$ is set to 1 during RESET.

CKA1D: CKA1 Clock Disable (bit 4 in CNTLA1)

When CKA1D is set to 1, the multiplexed CKA1/$\overline{TEND_0}$ pin (pin 50) is used for the $\overline{TEND_0}$ function. When CKA1D =

0, the pin is used as CKA1, an external data clock input/output for channel 1. CKA1D is cleared to 0 during RESET.

MPBR/EFR: Multiprocessor Bit Receive/Error Flag Reset (bit 3)

When multiprocessor mode is enabled (MP in CNTLB = 1), MPBR, when read, contains the value of the MPB bit for the last receive operation. When written to 0, the EFR function is selected to reset all error flags (OVRN, FE and PE) to 0. MPBR/EFR is undefined during RESET.

MOD2, 1, 0: ASCI Data Format Mode 2, 1, 0 (bits 2-0)

These bits program the ASCI data format as follows.

MOD2
= 0 → 7 bit data
= 1 → 8 bit data

MOD1
= 0 → No parity
= 1 → Parity enabled

MOD0
= 0 → 1 stop bit
= 1 → 2 stop bits

The data formats available based on all combinations of MOD2, MOD1 and MOD0 are shown in Table 14.

MOD2	MOD1	MOD0	Data Format
0	0	0	Start + 7 bit data + 1 stop
0	0	1	Start + 7 bit data + 2 stop
0	1	0	Start + 7 bit data + parity + 1 stop
0	1	1	Start + 7 bit data + parity + 2 stop
1	0	0	Start + 8 bit data + 1 stop
1	0	1	Start + 8 bit data + 2 stop
1	1	0	Start + 8 bit data + parity + 1 stop
1	1	1	Start + 8 bit data + parity + 2 stop

Table 14. Data Formats

ASCI Control Register B0, 1 (CNTLB0, 1). Each ASCI channel control register B configures multiprocessor mode, parity and baud rate selection.

ASCI Control Register B 0 (CNTLB0 : I/O Address = 02H)
ASCI Control Register B 1 (CNTLB1 : I/O Address = 03H)

bit 7	6	5	4	3	2	1	0
MPBT	MP	CTS/PS	PEO	DR	SS2	SS1	SS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 63. ASCI Control Register B

MPBT: Multiprocessor Bit Transmit (bit 7)

When multiprocessor communication format is selected (MP bit = 1), MPBT is used to specify the MPB data bit for transmission. If MPBT = 1, then MPB =1 is transmitted. If MPBT = 0, then MPB = 0 is transmitted. MPBT state is undefined during and after RESET.

MP: Multiprocessor Mode (bit 6)

When MP is set to 1, the data format is configured for multiprocessor mode based on the MOD2 (number of data bits) and MOD0 (number of stop bits) bits in CNTLA. The format is as follows.

Start bit + 7 or 8 data bits + MPB bit + 1 or 2 stop bits

Note that multiprocessor (MP = 1) format has no provision for parity. If MP = 0, the data format is based on MOD0, MOD1, MOD2, and may include parity. The MP bit is cleared to 0 during RESET.

$\overline{CTS}$/PS: Clear to Send/Prescale (bit 5)

When read, $\overline{CTS}$/PS reflects the state of the external $\overline{CTS}$ input. If the $\overline{CTS}$ input pin is HIGH, $\overline{CTS}$/PS will be read as 1. Note that when the $\overline{CTS}$ input pin is HIGH, the TDRE bit is inhibited (i.e. held at 0). For channel 1, the $\overline{CTS}_1$ input is multiplexed with RXS pin (Clocked Serial Receive Data). Thus, $\overline{CTS}$/PS is only valid when read if the channel 1 CTS1E bit = 1 and the $\overline{CTS}_1$ input pin function is selected. The read data of $\overline{CTS}$/PS is not affected by RESET.

When written, $\overline{CTS}$/PS specifies the baud rate generator prescale factor. If $\overline{CTS}$/PS is set to 1, the system clock is prescaled by 30 while if $\overline{CTS}$/PS is cleared to 0, the system clock is prescaled by 10. $\overline{CTS}$/PS is cleared to 0 during RESET.

PEO: Parity Even Odd (bit 4)

PEO selects even or odd parity. PEO does not affect the enabling/disabling of parity (MOD1 bit of CNTLA). If PEO is cleared to 0, even parity is selected. If PEO is set to 1, odd parity is selected. PEO is cleared to 0 during RESET.

DR: Divide Ratio (bit 3)

DR specifies the divider used to obtain baud rate from the data sampling clock. If DR is reset to 0, divide by 16 is used, while if DR is set to 1, divide by 64 is used. DR is cleared to 0 during RESET.

SS2, 1, 0: Source/Speed Select 2, 1, 0 (bits 2-0)

Specify the data clock source (internal or external) and baud rate prescale factor. SS2, SS1, SS0 are all set to 1 during RESET. Table 15 shows the divide ratio corresponding to SS2, SS1 and SS0.

The external ASCI channel 0 data clock pins are multiplexed with DMA control lines (CKA0/$\overline{DREQ}$ and CKA1/$\overline{TEND}_0$). During RESET, these pins are initialized as ASCI data clock inputs. If SS2, SS1 and SS0 are reprogrammed (any other value than SS2, SS1, SS0 = 1) these pins become ASCI data clock outputs. However, if DMAC channel 0 is configured to perform memory to/from I/O (and memory mapped I/O) transfers the CKA0/$\overline{DREQ}_0$ pin reverts to DMA control signals regardless of SS2, SS1, SS0 programming. Also, if the CKA1D bit in the CNTLA register is set to 1, then the CKA1/$\overline{TEND}_0$ reverts to the DMA Control output function regardless of SS2, SS1 and SS0 programming.

SS2	SS1	SSO	Divide Ratio
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	external clock

Table 15. Divide Ratio

Final data clock rates are based on $\overline{CTS}$/PS (prescale), DR, SS2, SS1, SS0 and the Z80180 system clock frequency (Reference Table 16).

MODEM Control Signals. ASCI channel 0 has CTS_0, DCD_0, and RTS_0 external modem control signals. ASCI channel 1 has a CTS1 modem control signal which is multiplexed with RXS (Clocked Serial Receive Data).

$\overline{CTS}_0$: **Clear to Send 0 (input).** The $\overline{CTS}_0$ input allows external control (start/stop) of ASCI channel 0 transmit operations. When $\overline{CTS}_0$ is HIGH, channel 0 TDRE bit is held at 0 whether or not the TDR0 (Transmit Data Register) is full or empty. When $\overline{CTS}_0$ is LOW, TDRE reflects the state of TDR0. Note that the actual transmit operation is not disabled by $\overline{CTS}_0$ HIGH, only TDRE is inhibited.

$\overline{DCD}_0$: **Data Carrier Detect 0 (input).** The $\overline{DCD}_0$ input allows external control (start/stop) of ASCI channel 0 receive operations. When $\overline{DCD}_0$ is HIGH, channel 0 RDRF bit is held at 0 whether or not the RDR0 (Receive Data Register) is full or empty. The error flags (PE, FE, and OVRN bits) are also held at 0. Even after the $\overline{DCD}_0$ input goes LOW, these bits will not resume normal operation until the status register (STAT0) is read. Note that this first read of STAT0, while enabling normal operation, still indicates the $\overline{DCD}_0$ input is HIGH (DCD0 bit = 1) even though it has gone LOW. Thus, the STAT0 register should be read twice to insure the $\overline{DCD}_0$ bit is reset to 0.

$\overline{RTS}_0$: **Request to Send 0 (output).** $\overline{RTS}_0$ allows the ASCI to control (start/stop) another communication devices transmission (e.g., by connection to that device's $\overline{CTS}$ input). $\overline{RTS}_0$ is essentially a 1 bit output port, having no side effects on other ASCI registers or flags.

$\overline{CTS}$1: **Clear to Send 1 (input).** Channel 1 $\overline{CTS}$1 input is multiplexed with RXS (Clocked Serial Receive Data). The $\overline{CTS}$1 function is selected when the CTS1E bit in STAT1 is set to 1. When enabled, the $\overline{CTS}$1 operation is equivalent to CTS0.

Modem control signal timing is shown in Fig. 64 and Fig. 65.
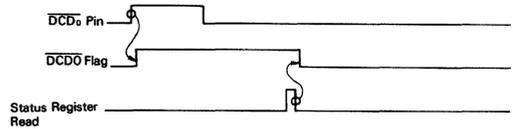
Figure 64. $\overline{DCD0}$ Timing
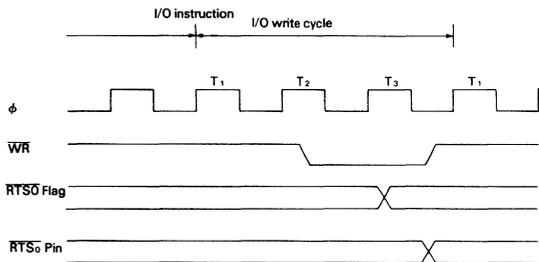
Figure 65. $\overline{RTS0}$ Timing

Fig. 66 shows the ASCI interrupt request generation circuit.
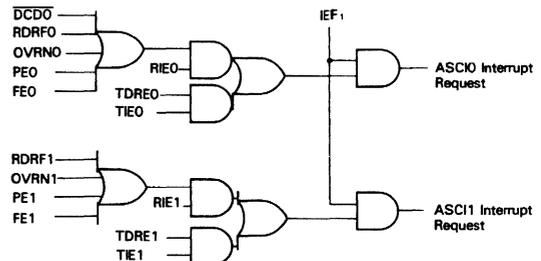
Figure 66. ASCI Interrupt Request Circuit Diagram

ASCI ↔ DMAC Operation. Operation of the ASCI with the on-chip DMAC channel 0 requires that the DMAC be correctly configured to use the ASCI flags as DMA request signals.

ASCI and RESET. During RESET, the ASCI status and control registers are initialized as defined in the individual register descriptions.

Receive and Transmit operations are stopped during RESET. However, the contents of the transmit and receive data registers (TDR and RDR) are not changed by RESET.

ASCI Clock
When in external clock input mode, the external clock is directly input to the sampling rate (÷16/÷64) as shown in Figure 67.
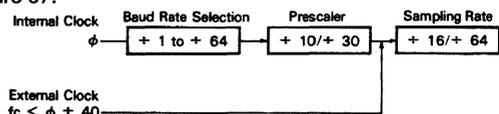
Figure 67.

Prescaler		Sampling Rate		Baud Rate				General Divide Ratio	Baud Rate (Example) (BPS)			CKA	
PS	Divide Ratio	DR	Rate	SS2	SS1	SS0	Divide Ratio		φ=6.144 MHz	φ=4.608 MHz	φ=3.072 MHz	I/O	Clock Frequency
0	φ÷10	0	16	0	0	0	÷1	φ÷160	38400		19200		φ÷10
				0	0	1	2	320	19200		9600		20
				0	1	0	4	640	9600		4800		40
				0	1	1	8	1280	4800		2400	O	80
				1	0	0	16	2560	2400		1200		160
				1	0	1	32	5120	1200		600		320
				1	1	0	64	10240	600		300		640
				1	1	1	—	fc÷16	—	—	—	I	fc
		1	64	0	0	0	÷1	φ÷640	9600		4800		φ÷10
				0	0	1	2	1280	4800		2400		20
				0	1	0	4	2560	2400		1200		40
				0	1	1	8	5120	1200		600	O	80
				1	0	0	16	10240	600		300		160
				1	0	1	32	20480	300		150		320
				1	1	0	64	40960	150		75		640
				1	1	1	—	fc÷64	—	—	—	I	fc
1	φ÷30	0	16	0	0	0	÷1	φ÷480		9600			φ÷30
				0	0	1	2	960		4800			60
				0	1	0	4	1920		2400			120
				0	1	1	8	3840		1200		O	240
				1	0	0	16	7680		600			480
				1	0	1	32	15360		300			960
				1	1	0	64	30720		150			1920
				1	1	1	—	fc÷16	—	—	—	I	fc
		1	64	0	0	0	÷1	φ÷1920		2400			φ÷30
				0	0	1	2	3840		1200			60
				0	1	0	4	7680		600			120
				0	1	1	8	15360		300		O	240
				1	0	0	16	30720		150			480
				1	0	1	32	61440		75			960
				1	1	0	64	122880		37.5			1920
				1	1	1	—	fc÷64	—	—	—	I	fc

Table 16. Baud Rate Selection

Clocked Serial I/O Port (CSI/O)

The Z80180 includes a simple, high speed clock, synchronous serial I/O port. The CSI/O includes transmit/receive (half-duplex), fixed 8-bit data, and internal or external data clock selection. High speed operation (baud rate 200K bits/second at fC = 4 MHz) is provided. The CSI/O is ideal for implementing a multiprocessor communication link between mulitple Z80180s. These secondary devices may typically perform a portion of the system I/O processing, i.e. keyboard scan/decode, LDC interface, etc.

CSI/O Block Diagram. The CSI/O block diagram is shown in Fig. 68. The CSI/O consists of two registers - the Transmit/Receive Data Register (TRDR) and Control Register (CNTR).

CSI/O Transmit/Receive Data Register (TRDR: I/O Address = 0BH). TRDR is used for both CSI/O transmission and reception. Thus, the system design must insure that the constraints of half-duplex operation are met (Transmit and receive operation cannot occur simultaneously). For example, if a CSI/O transmission is attempted while the CSI/O is receiving data, a CSI/O will not work. Also note that TRDR is not buffered. Therefore, attempting to perform a CSI/O transmit while the previous transmit data is still being shifted out causes the shift data to be immediate-

ly updated, thereby corrupting the transmit operation in progress. Similarly, reading TRDR while a transmit or receive is in progress should be avoided.
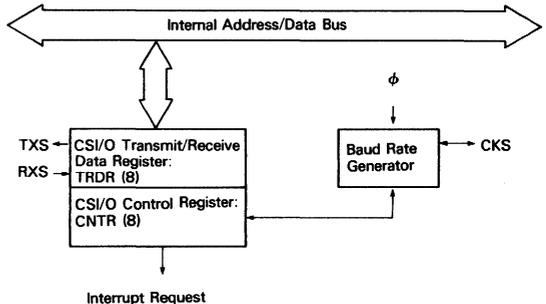
Figure 68. CSI/O Block Diagram

CSI/O Register Description

CSI/O Control/Status Register (CNTR: I/O Address = 0AH). CNTR is used to monitor CSI/O status, enable and disable the CSI/O, enable and disable interrupt generation, and select the data clock speed and source.

bit	7	6	5	4	3	2	1	0
	EF	EIE	RE	TE	—	SS2	SS1	SS0
	R	R/W	R/W	R/W		R/W	R/W	R/W
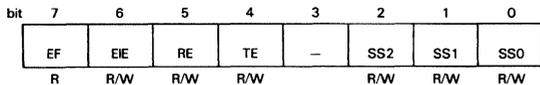
Figure 69. CSI/O Control Register

EF: End Flag (bit 7)

EF is set to 1 by the CSI/O to indicate completion of an 8-bit data transmit or receive operation. If EIE (End Interrupt Enable) bit = 1 when EF is set to 1, a CPU interrupt request is generated. Program access of TRDR only occurs if EF = 1. The CSI/O clears EF to 0 when TRDR is read or written. EF is cleared to 0 during RESET and IOSTOP mode.

EIE: End Interrupt Enable (bit 6)

EIE is set to 1 to enable EF = 1 to generate a CPU interrupt request. The interrupt request is inhibited if EIE is reset to 0. EIE is cleared to 0 during RESET.

RE: Receive Enable (bit 5)

A CSI/O receive operation is started by setting RE to 1. When RE is set to 1, the data clock is enabled. In internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, data is shifted in on the RXS pin in synchronization with the (internal or external) data clock. After receiving 8 bits of data, the CSI/O automatically clears RE to 0, EF is set to 1, and an interrupt (if enabled by EIE = 1) is generated. RE and TE are never both set to 1 at the same time. RE is cleared to 0 during RESET and ISTOP mode.

Note that RXS (pin 52) is multiplexed with CTS1 modem control input of ASCI channel 1. In order to enable the RXS function, the CTS1E bit in CNTA1 should be reset to 0.

Transmit Enable (bit 4)

A CSI/O transmit operation is started by setting TE to 1. When TE is set to 1, the data clock is enabled. When in internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, data is shifted out on the TXS pin synchronous with the (internal or external) data clock. After transmitting 8 bits of data, the CSI/O automatically clears TE to 0, EF is set to 1, and an interrupt (if enabled by EIE = 1) is generated. TE and RE are never both set to 1 at the same time. TE is cleared to 0 during RESET and IOSTOP mode.

SS2, 1, 0: Speed Select 2, 1, 0 (bits 2-0)

SS2, SS1 and SS0 select the CSI/O transmit/receive clock source and speed. SS2, SS1 and SS0 are all set to 1 during RESET. Table 17 shows CSI/O Baud Rate Selection.

SS2	SS1	SS0	Divide Ratio	Baud Rate
0	0	0	÷ 20	(200000)
0	0	1	÷ 40	(100000)
0	1	0	÷ 80	(50000)
0	1	1	÷ 160	(25000)
1	0	0	÷ 320	(12500)
1	0	1	÷ 640	(6250)
1	1	0	÷ 1280	(3125)
1	1	1	external Clock input (less than ÷ 20)	

( ) shows the baud rate (BPS) at ϕ = 4 MHz.

Table 17. CSI/O Baud Rate Selection

After RESET, the CKS pin is configured as an external clock input (SS2, SS1, SS0 = 1). Changing these values causes CKS to become an output pin and the selected clock is output when transmit or receive operations are enabled.

CSI/O Interrupts. The CSI/O interrupt request circuit is shown in Fig. 70.
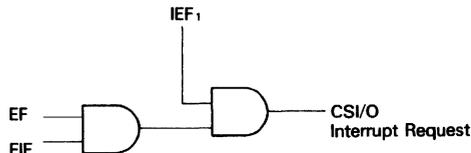
Figure 70. CSI/O Interrupt Request Generation

CSI/O operation. The CSI/O is operated using status polling or interrupt driven algorithms.

Transmit - Polling

1. Poll the TE bit in CNTR until TE = 0.

2. Write the transmit data into TRDR.

3. Set the TE bit in CNTR to 1.

4. Repeat 1 to 3 for each transmit data byte.

Transmit - Interrupts

1. Poll the TE bit in CNTR until TE = 0.

2. Write the first transmit data byte into TRDR.

3. Set the TE and EIE bits in CNTR to 1.

4. When the transmit interrupt occurs, write the next transmit data byte into TRDR.

5. Set the TE bit in CNTR to 1.

6. Repeat 4 to 5 for each transmit data byte.

Receive - Polling

1. Poll the RE bit in CNTR until RE = 0.

2. Set the RE bit in CNTR to 1.

3. Poll the RE bit in CNTR until RE = 0.

4. Read the receive data from TRDR.

5. Repeat 2 to 4 for each receive data byte.

Receive - Interrupts

1. Poll the RE bit in CNTR until RE = 0.

2. Set the RE and EIE bits in CNTR to 1.

3. When the receive interrupt occurs read the receive data from TRDR.

4. Set the RE bit in CNTR to 1.

5. Repeat 3 to 4 for each receive data byte.

CSI/O Operation Timing Notes

(1) Transmitter clocking and receiver sampling timings are different from internal and external clocking modes. Fig. 71 to Fig. 74 show CSI/O Transmit/Receive Timing.

(2) The transmitter and receiver is disabled (TE and RE = 0) when initializing or changing the baud rate.

CSI/O Operation Notes

(1) Disable the transmitter and receiver (TE and RE = 0) before initializing or changing the baud rate. When changing the baud rate after completion of transmission or reception, a delay of at least one bit time is required before baud rate modification.

(2) When RE or TE is cleared to 0 by software, a corresponding receive or transmit operation is immediately terminated. Normally, TE or RE is only cleared to 0 when EF = 1.

(3) Simultaneous transmission and reception is not possible. Thus, TE and RE are not both 1 at the same time.

CSI/O and RESET. During RESET each bit in the CNTR is initialized as defined in the CNTR register description.

CSI/O transmit and receive operations in progress are aborted during RESET. However, the contents of TRDR are not changed.
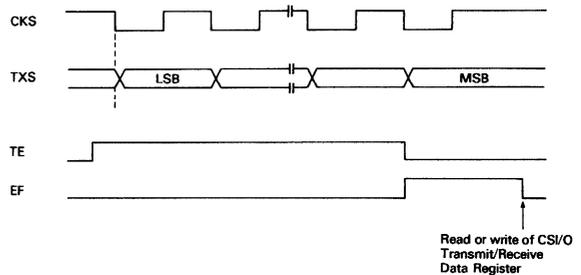
Figure 71. Transmit Timing-Internal Clock
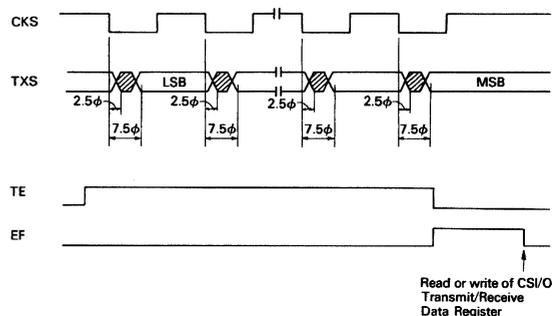
Figure 72. Transmit Timing-External Clock
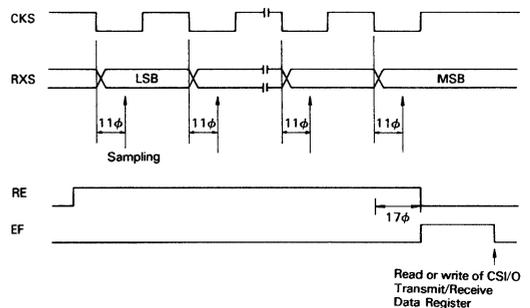
Figure 73. Receive Timing-Internal Clock
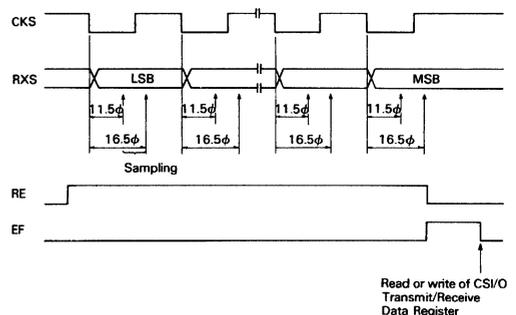
Figure 74. Receive Timing-External Clock

Programmable Reload Timer (PRT)

The Z80180 contains a two channel 16-bit Programmable Reload Timer. Each PRT channel contains a 16-bit down counter and a 16-bit reload register. The down counter is directly read and written and a down counter overflow interrupt can be programmably enabled or disabled. Also, PRT channel 1 has a TOUT output pin (pin 31 - multiplexed with A18) which can be set HIGH, LOW, or toggled. Thus, PRT1 can perform programmable output waveform generation.

PRT block diagram. The PRT block diagram is shown in Fig. 75. The two channels have separate timer data and reload registers and a common status/control register. The PRT input clock for both channels is equal to the system clock divided by 20.
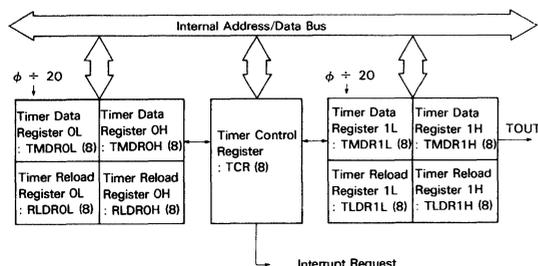
Figure 75. PRT Block Diagram

PRT Register Description. Timer Data Register (TMDR: I/O Address - CH0, ODH, OCH, CH1: 15H, 14H). PRT0 and PRT1 each have 16-bit Timer Data Registers (TMDR). TMDR0 and TMDR1 are each accessed as low and high byte registers (TMDR0H, TMDR0L and TMDR1H, TMDR1L). During RESET, TMDR0 and TMDR1 are set to FFFFH.

TMDR is decremented once every twenty clocks. When TMDR counts down to 0, it is automatically reloaded with the value contained in the Reload Register (RLDR).

TMDR is read and written by software using the following procedures. The read procedure uses a PRT internal temporary storage register to return accurate data without requiring the timer to be stopped. The write procedure requires the PRT to be stopped.

For reading (without stopping the timer), TMDR is read in the order of lower byte - higher byte (TMDRnL, TMDRnH). The lower byte read (TMDRnL) stores the higher byte value in an internal register. The following higher byte read (TMDRnH) accesses this internal register. This procedure insures timer data validity by eliminating the problem of potential 16-bit timer updating between each 8-bit read. Specifically, reading TMDR in higher byte - lower byte order may result in invalid data. Note the implications of TMDR higher byte internal storage for applications which may read only the lower and/or higher bytes. In normal operation all TMDR read routines should access both the lower and higher bytes, in that order. For writing, the TMDR down counting must be inhibited using the TDE (Timer Down Count Enable) bits in the TCR (Timer Control Register). Then, any or both higher and lower bytes or TMDR can be freely written (and read) in any order.

Timer Reload Register (RLDR: I/O Address = CH0, OEH, OFH, CH1: 16H, 17H). PRT0 and PRT1 each have 16-bit Timer Reload Registers (RLDR). RLDR0 and RLDR1 are each accessed as low and high byte registers (RLDR0H, RLDR0L and RLDR1H, RLDR1L). During RESET, RLDR0 and RLDR1 are set to FFFFH.

When the TMDR counts down to 0, it is automatically reloaded with the contents of RLDR.

Timer Control Register (TCR). TCR monitors both channels (PRT0, PRT1) TMDR status. It also controls enabling and disabling of down counting and interrupts along with controlling output pin A18/TOUT for PRT1.

bit	7	6	5	4	3	2	1	0
	TIF1	TIF0	TIE1	TIE0	TOC1	TOC0	TDE1	TDE0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W
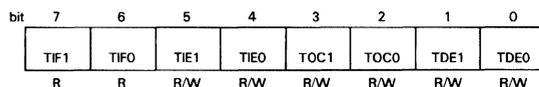
Figure 76. Timer Control Register
(TCR : I/O Address = 10 H)

TIF1: Timer Interrupt Flag 1 (bit 7). When TMDR1 decrements to 0, TIF1 is set to 1. This generates an interrupt request if enabled by TIE1 = 1. TIF1 is reset to 0 when TCR is read and the higher or lower byte of TMDR1 is read. During RESET, TIF1 is cleared to 0.

TIF0: Timer Interrupt Flag 0 (bit 6). When TMDR0 decrements to 0, TIF0 is set to 1. This generates an interrupt request if enabled by TIE0 = 1. TIF0 is reset to 0 when TCR is read and the higher or lower byte of TMDR0 is read. During RESET, TIF0 is cleared to 0.

TIE1: Timer Interrupt Enable 1 (bit 5). When TIE1 is set to 1, TIF1 = 1 generates a CPU interrupt request. When TIE1 is reset to 0, the interrupt request is inhibited. During RESET, TIE1 is cleared to 0.

TIE0: Timer Interrupt Enable 0 (bit 4). When TIE0 is set to 1, TIF0 = 1 generates a CPU interrupt request. When TIE0 is reset to 0, the interrupt request is inhibited. During RESET, TIE0 is cleared to 0.

TOC1, 0: Timer Output Control (bits 3, 2). TOC1 and TOC0 control the output of PRT1 using the multiplexed A18/TOUT pin as shown in Table 18. During RESET, TOC1 and TOC0 are cleared to 0. This selects the address function for A18/TOUT. By programming TOC1 and TOC0, the A18/TOUT pin can be forced HIGH, LOW, or toggled when TMDR1 decrements to 0.

TOC1	TOC0	OUTPUT	
0	0	Inhibited	(A₁₈/TOUT pin is selected as an address output function.)
0	1	toggled*	(A₁₈/TOUT pin is selected as a PRT1 output function.)
1	0	0	
1	1	1	

Table 18: Timer Output Control

TDE1, 0: Timer Down Count Enable (bits 1, 0). TDE1 and TDE0 enable and disable down counting for TMDR1 and TMDR0, respectively. When TDEn (n = 0, 1) is set to 1, down counting is executed for TMDRn. When TDEn is reset to 0, down counting is stopped and TMDRn is freely read or written. TDE1 and TDE0 are cleared to 0 during RESET and TMDRn will not decrement until TDEn is set to 1.

Fig. 77 shows timer initialization, count down, and reload timing. Fig. 78 shows timer output (A18/TOUT) timing.
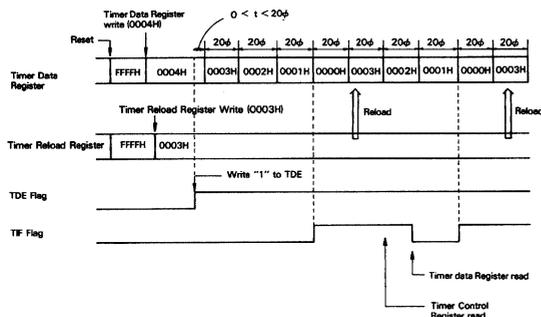
Figure 77. Timer Initialization, Count Down, and Reload Timing
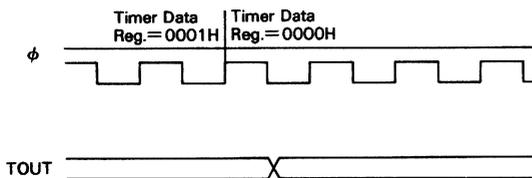
Figure 78. Timer Output Timing

PRT interrupts. The PRT interrupt request circuit is shown in Fig. 79.
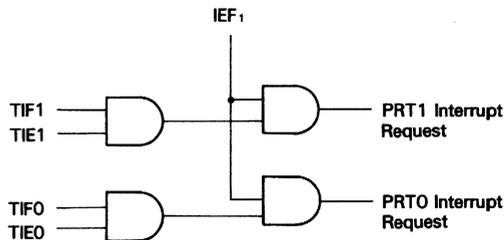
Figure 79. PRT Interrupt Request Generation

PRT and RESET. During RESET, the bits in TCR are initialized as defined in the TCR register description. Down counting is stopped and the TMDR and RLDR registers are initialized to FFFFH. The A18/TOUT pin reverts to the address output function.

PRT Operation Notes.

(1) TMDR data is accurately read without stopping down counting by reading the lower (TMDRnL*) and higher (TMDRnH*) bytes in that order. Also, TMDR is read or written by stopping the down counting.

(2) Care should be taken to insure that a timer reload does not occur during or between lower (RLDRnL*) and higher (RLDRnH*) byte writes. This may be guaranteed by system design/timing or by stopping down counting (with TMDR containing a non-zero value) during the RLDR updating. Similarly, in applications where TMDR is written at each TMDR overflow, the system/software design should guarantee that RLDR can be updated before the next overflow occurs. Otherwise, time base inaccuracy will occur.

Note: *n = 0, 1

(3) During RESET, the multiplexed A18/TOUT pin reverts to the address output. By reprogramming the TOC1 and TOC0 bits, the timer output function for PRT channel 1 is selected. The following shows the initial state of the TOUT pin after TOC1 and TOC0 are programmed to select the PRT channel 1 timer output function.

(i) PRT (channel 1) has not counted down to 0. If the PRT has not counted down to 0 (timed out), the initial state of TOUT depends on the programmed value in TOC1 and TOC0.

Secondary Bus Interface

E clock Output Timing. The Z80180 also has a secondary bus interface that allows it to easily interface with other peripheral families.

These devices require connection with the Z80180 synchronous E clock output. The speed (access time) required for the peripheral devices are determined by the Z80180 clock rate. Table 19, and Figures 80-82 define E clock output timing.

Wait states are inserted in op-code fetch, memory read/write, and I/O read/write cycles which extend the duration of E clock output HIGH. During I/O read/write cycles with no wait states (only occurs during on-chip I/O register accesses), E will not go HIGH.

Condition	Duration of E Clock Output "High"	
Op-code Fetch Cycle Memory Read/Write Cycle	$T_2\uparrow - T_3\downarrow$	$(1.5\phi + n_w \cdot \phi)$
I/O read Cycle	1st $T_w\uparrow - T_3\downarrow$	$(0.5\phi + n_w \cdot \phi)$
I/O Write Cycle	1st $T_w\uparrow - T_3\uparrow$	$(n_w \cdot \phi)$
$\overline{NMI}$ Acknowledge 1st MC	$T_2\uparrow - T_3\downarrow$	(1.5ϕ)
$\overline{INT_0}$ Acknowledge 1st MC	1st $T_w\uparrow - T_3\downarrow$	$(0.5\phi + n_w \cdot \phi)$
BUS RELEASE mode SLEEP mode SYSTEM STOP mode	$\phi\downarrow - \phi\downarrow$	$(2\phi$ or $1\phi)$

NOTE) n_w : the number of wait states
MC : Machine Cycle

Table 19. E Clock Timing in Each Condition
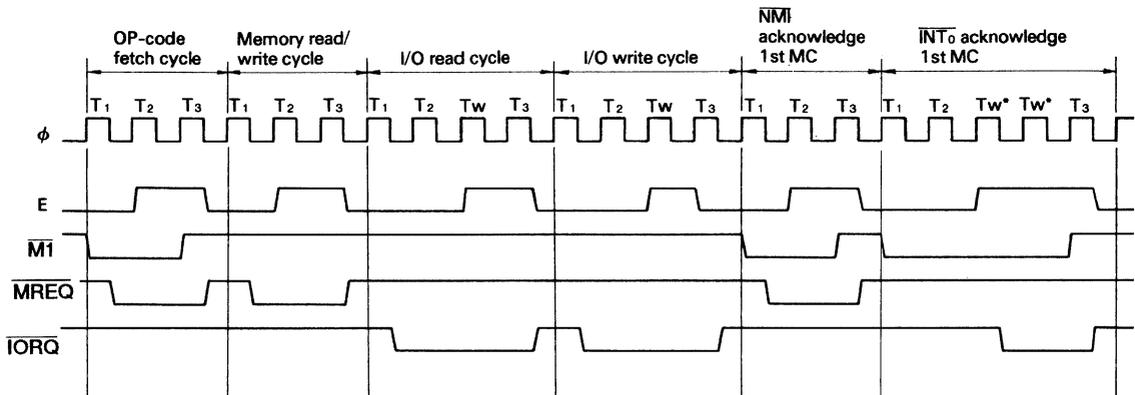
NOTE) MC: Machine Cycle

* Two wait states are automatically inserted.

Figure 80. E Clock Timing (During Read/Write Cycle and Interrupt Acknowledge Cycle)
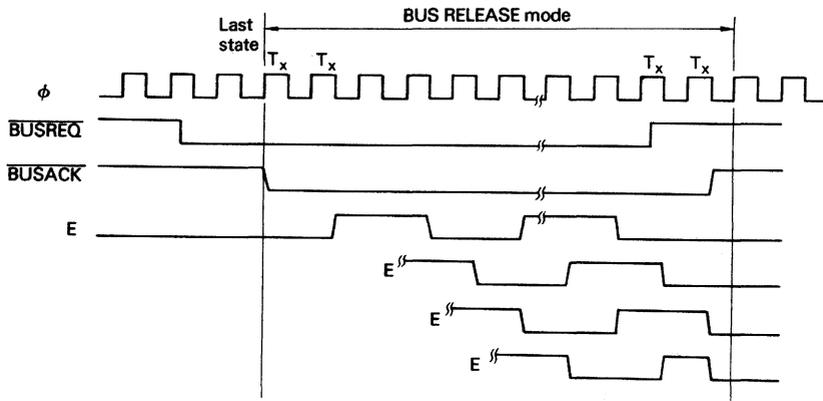
Figure 81. E Clock Timing in BUS RELEASE Mode
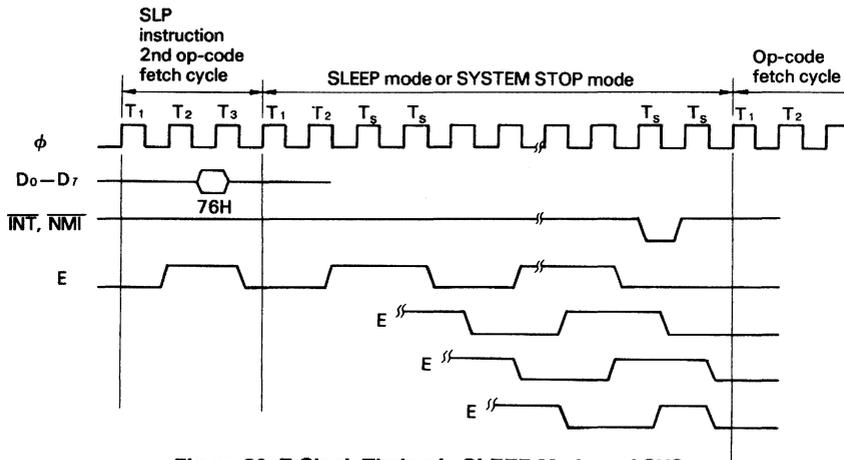
**Figure 82. E Clock Timing in SLEEP Mode and SYS-
TEM STOP Mode**

On-Chip Clock Generator

The Z80180 contains a crystal oscillator and system clock generator. A crystal can be directly connected or an external clock input can be provided. In either case, the system clock is equal to one-half the input clock. For example, a crystal or external clock input of 8 MHz corresponds with a system clock rate of 4 MHz.

The following table shows the AT cut crystal characteristics (Co, Rs) and the load capacitance (CL1, CL2) required for various frequencies of Z80180 operation.

Clock Frequency / Item	4MHz	4MHz < f ≤ 12MHz	12MHz < f ≤ 16MHz
Co	< 7 pF	< 7 pF	< 7 pF
Rs	< 60Ω	< 60Ω	< 60Ω
CL₁, CL₂	10 to 22 pF ± 10%	10 to 22 pF ± 10%	10 to 22 pF ± 10%

Table 20.

If an external clock input is used instead of a crystal, the waveform (twice the clock rate) should exhibit a 50% ± 10% duty cycle. Note that the minimum clock input HIGH voltage level is Vcc-0.6V. The external clock input is connected to the EXTAL pin, while the XTAL pin is left open. Fig. 83 shows external clock interface.
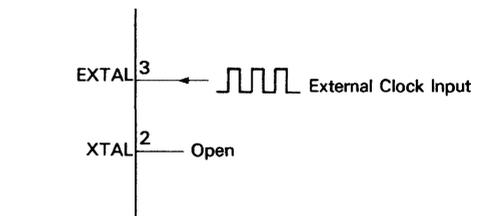
Figure 83. External Clock Interface

Fig. 84 shows the Z80180 clock generator circuit while Fig. 85 and Fig. 86 specify circuit board design rules.
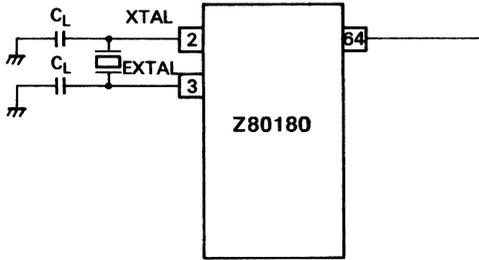
Figure 84. Clock Generator Circuit
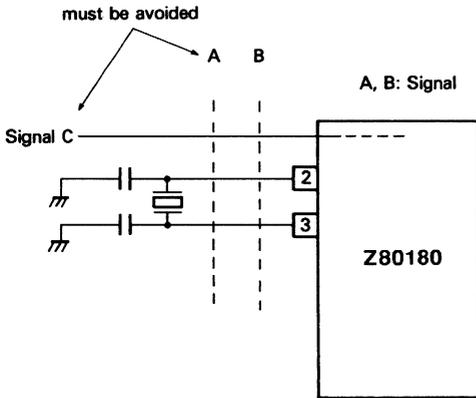
Figure 85. Circuit Board Design Rules

(Top View)
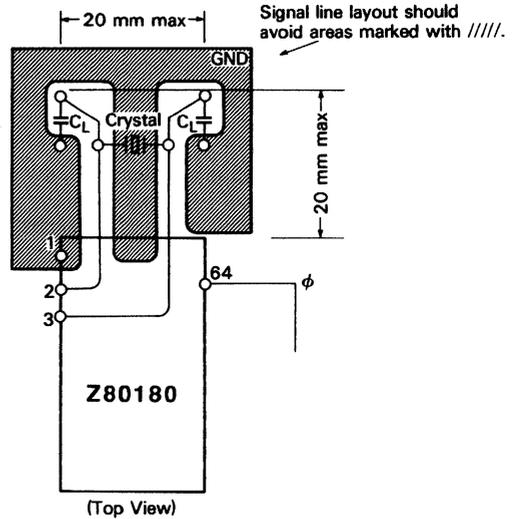
Figure 86. Example of Board Design

Circuit Board design should observe the following.

(1) To prevent induced noise, the crystal and load capacitors should be physically located as close to the LSI as possible.

(2) Signal lines should not run parallel to the clock oscillator inputs. In particular, the clock input circuitry and the system clock output (pin 64) should be separated as much as possible.

(3) Similar to (2), Vcc power lines should be separated from the clock oscillator input circuitry.

(4) Resistivity between XTAL or EXTAL and the other pins should be greater than 10M ohms.

Signal line layout should avoid areas marked with /////.

Miscellaneous

Free Running Counter (I/O Address = 18H)

Read only 8-bit free running counter without control registers and status registers. The contents of the 8-bit free running counter is counted down by one with an interval of 10 clock cycles. The free running counter continues counting down without being affected by the read operation.

If data is written into the free running counter, the interval of DRAM refresh cycle and baud rates for the ASCI and CSI/O are not guaranteed.

In IOSTOP mode, the free running counter continues counting down. It is initialized to FFH during RESET.

SOFTWARE ARCHITECTURE

Instruction Set. The Z80180 is object code compatible with the Z80 CPU, refer to the Z80 CPU Technical Manual or the Z80 Assembly Language Programming Manual for further details.

New Instructions	Operation
SLP	Enter SLEEP mode
MLT	8-bit multiply with 16-bit result
INO g, (m)	Input contents of immediate I/O address
OUT0 (m), g	Output register contents to immediate I/O address
OTIM	Block output - increment
OTIMR	Block output - increment and repeat
OTDM	Block output - decrement
OTDMR	Block output - decrement and repeat
TSTIO m	Non-destructive AND, I/O port, and accumulator
TST g	Non-destructive AND, register, and accumulator
TST m	Non-destructive AND, immediate data and accumulator.
TST (HL)	Non-destructive AND, memory data, and accumulator.

SLP - Sleep. The SLP instruction causes the Z80180 to enter the SLEEP low power consumption mode. See section 2.4 for a complete description of the SLEEP state.

MLT - Multiply. The MLT performs unsigned multiplication on two 8-bit numbers yielding a 16-bit result. MLT may specify BC, DE, HL or SP registers. In all cases, the 8-bit operands are loaded into each half of the 16-bit register and the 16-bit result is returned in that register.

OTIM, OTIMR, OTDM, OTDMR - Block I/O. The contents of memory pointed to by HL is output to the I/O address in (C). The memory address (HL) and I/O address (C) are incremented in OTIM and OTIMR and decremented in OTDM and OTDMR, respectively. The B register is decremented. The OTIMR and OTDMR variants repeat the above sequence until register B is decremented to 0. Since the I/O address (C) is automatically incremented or decremented, these instructions are useful for block I/O (such as Z80180 on-chip I/O) initialization. When I/O is accessed, 00H is output in high-order bits of address automatically.

TSTIO m - Test I/O Port. The contents of the I/O port addressed by C are ANDed with immediately specified 8-bit data and the status flags are updated. The I/O port contents are not written (non-destructive AND). When I/O is accessed, 00H is output in higher bits of address automatically.

TST g - Test Register. The contents of the specified register are ANDed with the accumulator (A) and the status flags are updated. The accumulator and specified register are not changed (non-destructive AND).

TST m - Test Immediate. The contents of the immediately specified 8-bit data are ANDed with the accumulator (A) and the status flags are updated. The accumulator is not changed (non-destructive AND).

TST (HL) - Test Memory. The contents of memory pointed to by HL are ANDed with the accumulator (A) and the status flags are updated. The memory contents and accumulator are not changed (non-destructive AND).

INO g, (m) - Input, Immediate I/O address. The contents of immediately specified 8-bit I/O address are input into the specified register. When I/O is accessed, 00H is output in high-order bits of the address automatically.

OUT0 (m), g - Output, immediate I/O address. The contents of the specified register are output to the immediately specified 8-bit I/O address. When I/O is accessed, 00H is output in high-order bits of the address automatically.

CPU Registers

The Z80180 CPU registers consist of Register Set GR, Register Set GR' and Special Registers.

The Register Set GR consists of 8-bit Accumulator (A), 8-bit Flag Register (F), and three General Purpose Registers (BC, DE, and HL) which may be treated as 16-bit registers (BC, DE, and HL) or as individual 8-bit registers (B, C, D, E, H, and L) depending on the instruction to be executed. The Register Set GR' is alternate register set of Register Set GR and also contains Accumulator (A'), Flag Register (F') and three General Purpose Registers (BC', DE', and HL'). While the alternate Register Set GR' contents are not directly accessible, the contents can be programmably exchanged at high speed with those of Register Set GR.

The Special Registers consist of 8-bit Interrupt Vector Register (I), 8-bit R Counter (R), two 16-bit Index Registers (IX and IY), 16-bit Stack Pointer (SP), and 16-bit Program Counter (PC).

Fig. 87 shows CPU registers configuration.

Register Set GR

Accumulator A	Flag Register F	
B Register	C Register	General
D Register	E Register	Purpose
H Register	L Register	Registers

Register Set GR'

Accumulator A'	Flag Register F'	
B' Register	C' Register	General
D' Register	E' Register	Purpose
H' Register	L' Register	Registers

Special Registers

Interrupt Vector Register I	R Counter R
Index Register	IX
Index Register	IY
Stack Pointer	SP
Program Counter	PC

Figure 87. CPU Registers

Accumulator (A, A')

The Accumulator (A) serves as the primary register used for many arithmetic, logical and I/O instructions.

Flag Registers (F, F')

The flag register stores various status bits (described in the next section) which reflect the results of instruction execution.

General Purpose Registers (BC, BC', DE, DE', HL, HL')

The General Purpose Registers are used for both address and data operation. Depending on instruction, each half (8 bits) of these registers (B, C, D, E, H, and I) may also be used.

Interrupt Vector Register (I)

For interrupts which require a vector table address to be calculated ($\overline{INT}_0$ Mode 2, $\overline{INT}_1$, $\overline{INT}_2$ and internal interrupts), the Interrupt Vector Register (I) provides the most significant byte of the vector table address. I is cleared to 00H during RESET.

R Counter (R)

The least significant seven bits of the R counter (R) serve to count the number of instructions executed by the Z80180. R is incremented for each CPU op-code fetch cycle (each $\overline{M1}$ cycle). R is cleared to 00H during RESET.

Index Registers (IX, and IY)

The Index Registers are used for both address and data operations. For addressing, the contents of a displacement specified in the instruction are added to or subtracted from the Index Register to determine an effective operand address.

Stack Pointer (SP)

The Stack Pointer (SP) contains the memory address based LIFO stack. SP is cleared to 0000H during RESET.

Program Counter (PC)

The Program Counter (PC) contains the address of the instruction to be executed and is automatically updated after each instruction fetch. PC is cleared to 0000H during RESET.

Flag Register (F)

The Flag Register stores the logical state reflecting the results of instruction execution. The contents of the Flag Register are used to control program flow and instruction operation.

bit	7	6	5	4	3	2	1	0
	S	Z	–	H	–	P/V	N	C
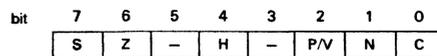
Figure 88. Flag Register (F)

S: Sign (bit 7)

S stores the state of the most significant bit (bit 7) of the result. This is useful for operations with signed numbers in which values with bit 7 = 1 are interpreted as negative.

Z: Zero (bit 6)

Z is set to 1 when instruction execution produces 0 result. Otherwise, Z is reset to 0.

H: Half Carry (bit 4)

H is used by the DAA (Decimal Adjust Accumulator) instruction to reflect borrow or carry from the least significant 4 bits and thereby adjust the results of BCD addition and subtraction.

P/V: Parity/Overflow (bit 2)

P/V serves a dual purpose. For logical operations P/V is set to 1 if the number of 1 bit in the result is even and P/V is reset to 0 if the number of 1 in the result is odd. For two complement arithmetic, P/V is set to 1 if the operation produces a result which is outside the allowable range (+ 127 to - 128 for 8-bit operations, + 32767 to - 32768 for 16-bit operations).

N: negative (bit 1)

N is set to 1 if the last arithmetic instruction was a subtract operation (SUB, DEC, CP, etc.) and N is reset to 0 if the last arithmetic instruction was an addition operation (ADD, INC, etc.).

C: Carry (bit 0)

C is set to 1 when a carry (addition) or borrow (subtraction) from the most significant bit of the result occurs. C is also affected by Accumulator logic operations such as shifts and rotates.

Addressing Modes

The Z80180 instruction set includes eight addressing modes.

Implied Register
Register Direct
Register Indirect
Indexed
Extended
Immediate
Relative
IO

Implied Register (IMP)

Certain op-codes automatically imply register usage, such as the arithmetic operations which inherently reference the Accumulator, index Registers, Stack Pointer and General Purpose Registers.

Register Direct (REG)

Many op-codes contain bit fields specifying registers to be used for the operation. The exact bit field definitions vary depending on instruction as follows.

8-bit Register

g or g' field	Register
0 0 0	B
0 0 1	C
0 1 0	D
0 1 1	E
1 0 0	H
1 0 1	L
1 1 0	—
1 1 1	A

ww field	Register
0 0	B C
0 1	D E
1 0	H L
1 1	S P

xx field	Register
0 0	B C
0 1	D E
1 0	I X
1 1	S P

16-bit Register

zz field	Register
0 0	B C
0 1	D E
1 0	H L
1 1	A F

yy field	Register
0 0	B C
0 1	D E
1 0	I Y
1 1	S P

Suffixed H and L to ww,xx,yy,zz (ex. wwH,IXL) indicate upper and lower 8-bit of the 16-bit register respectively.

Figure 89. Register Direct - Bit Field Definitions

Register Indirect (REG)

The memory operand address is contained in one of the 16-bit General Purpose Registers (BC, DE, and HL).

BC
DE
HL
Operand
Memory

Figure 90. Register Indirect Addressing

Indexed (INDX)

The memory operand address is calculated using the contents of an Index Register (IX or IY) and an 8-bit signed displacement specified in the instruction.

op-code 1
op-code 2
displacement (d)
Sign extended
IX or IY
Operand
Memory
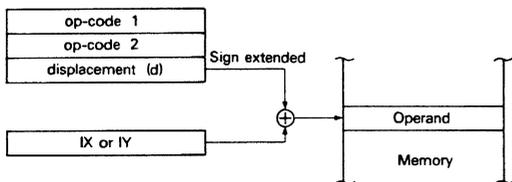
Figure 91. Indexed Addressing

Extended (EXT)

The memory operand address is specified by two bytes contained in the instruction.

op-code
n
m
m   n
Operand
Memory
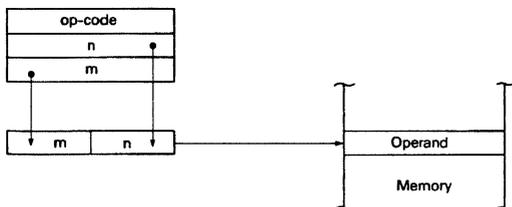
Figure 92. Extended Addressing

Immediate (IMMED)

The memory operands are contained within one or two bytes of the instruction.

op-code
m   } 8-bit operand

op-code
n
m   } 16-bit operand
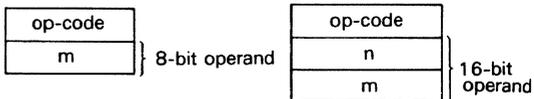
Figure 93. Immediate Addressing

Relative (REL)

Relative addressing mode is only used by the conditional and unconditional branch instructions. The branch displacement (relative to the contents of the program counter) is contained in the instruction.

op-code
displacement (j)   Sign extended

Program Counter (PC)
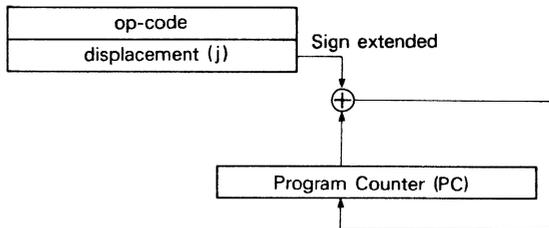
Figure 94. Relative Addressing

IO (IO)

IO addressing mode is used only by I/O instructions. This mode specifies I/O address ($\overline{IORQ} = 0$) and outputs them as follows.

(1) An operand is output to A_0-A_7. The Contents of Accumulator is output to A_8-A_{15}.

(2) The Contents of Register B is output to A_0-A_7. The Contents of Register C is output to A_8-A_{15}.

(3) An operand is output to A_0-A_7. 00H is output to A_8-A_{15}. (useful for internal I/O register access)

(4) The Contents of Register C is output to A_0-A_7. 00H is output to A_8-A_{15}. (useful for internal I/O register access)

ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	$-0.3 \sim +7.0$	V
Input Voltage	V_{in}	$-0.3 \sim V_{CC}+0.3$	V
Operating Temperature	T_{opr}	$0 \sim 70$	°C
Extended Temperature	T_{ext}	$-40 \sim 85$	°C
Storage Temperature	T_{stg}	$-55 \sim +150$	°C

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

DC CHARACTERISTICS
($V_{CC} = 5V + 10\%$, Vss = OV over specified temperature range unless otherwise noted)

Symbol	Item	Condition	min	typ	max	Unit
V_{IH1}	Input "H" Voltage $\overline{RESET}$, EXTAL, $\overline{NMI}$		$V_{CC}-0.6$	–	$V_{CC}+0.3$	V
V_{IH2}	Input "H" Voltage Except $\overline{RESET}$, EXTAL, $\overline{NMI}$		2.0	–	$V_{CC}+0.3$	V
V_{IL1}	Input "L" Voltage $\overline{RESET}$, EXTAL, $\overline{NMI}$		-0.3	–	0.6	V
V_{IL2}	Input "L" Voltage Except $\overline{RESET}$, EXTAL, $\overline{NMI}$		-0.3	–	0.8	V
V_{OH}	Output "H" Voltage All outputs	$I_{OH} = -200\mu A$	2.4	–	–	V
		$I_{OH} = -20\mu A$	$V_{CC}-1.2$	–	–	
V_{OL}	Output "L" Voltage All Outputs	$I_{OL} = 2.2$ mA	–	–	0.45	V
I_{IL}	Input Leakage Current All Inputs Except XTAL, EXTAL	Vin=0.5 $\sim V_{CC}-0.5$	–	–	1.0	μA
I_{TL}	Three State Leakage Current	Vin=0.5 $\sim V_{CC}-0.5$	–	–	1.0	μA
$I_{CC}*$	Power Dissipation* (Normal Operation)	f=6 MHz	–	15	40	mA
		f=8 MHz	–	20	50	
		f=10MHz **	–	25	60	
	Power Dissipation* (SYSTEM STOP mode)	f=6 MHz	–	3.8	12.5	
		f=8 MHz	–	5	15	
		f=10MHz **	–	6.3	17.5	
Cp	Pin Capacitance	Vin=0V, f=1 MHz Ta=25°C	–	–	12	pF

* $V_{IHmin} = V_{CC}-1.0V$, $V_{ILmax} = 0.8V$ (all output terminals are at no load.)
  Vcc = 5.0V

Z80180 AC CHARACTERISTICS

(V$_{CC}$ = 5V ± 10%, V$_{SS}$ = 0V, Ta = 0 to +70°C , unless otherwise noted.)

No.	Symbol	Item	Z80180-6		Z80180-8		Z80180-10		Unit
			min	max	min	max	min	max	
1.	t_{cyc}	Clock Cycle Time	162	2000	125	2000	100	2000	ns
2.	t_{CHW}	Clock "H" Pulse Width	65	—	50	—	40	—	ns
3.	t_{CLW}	Clock "L" Pulse Width	65	—	50	—	· 40	—	ns
4.	t_{cf}	Clock Fall Time	—	15	—	15	—	10	ns
5.	t_{cr}	Clock Rise Time	—	15	—	15	—	10	ns
6.	t_{AD}	Ø ↑ to Address Valid Delay	—	90	—	80	—	70 '	ns
7.	t_{AS}	Address Valid to ($\overline{MREQ}$ ↓ or $\overline{IORQ}$ ↓)	30	—	20	—	10	—	ns
8.	t_{MED1}	Ø ↓ to $\overline{MREQ}$ ↓ Delay	—	60	—	50	—	50	ns
9.	t_{RDD1}	Ø ↓ to $\overline{RD}$ ↓ Delay $\overline{IOC}$= 1	—	60	—	50	—	50	ns
		Ø ↑ to $\overline{RD}$ ↓ Delay $\overline{IOC}$= 0	—	65	—	60	—	55	
10.	t_{M1D1}	Ø ↑ to $\overline{M1}$ ↓ Delay	—	80	—	70	—	60	ns
11.	t_{AH}	Address Hold Time from ($\overline{MREQ}$, $\overline{IOREQ}$, $\overline{RD}$, $\overline{WR}$)	35	—	20	—	10 ·	—	ns
12.	t_{MED2}	Ø ↓ to $\overline{MREQ}$ ↑ Delay	—	60	—	50	—	50	ns
13.	t_{RDD2}	Ø ↓ to $\overline{RD}$ ↑ Delay	—	60	—	50	—	50	ns
14.	t_{M1D2}	Ø ↑ to $\overline{M1}$ ↑ Delay	—	80	—	70*	—	60	ns
15.	t_{DRS}	Data Read Set-up Time	40	—	30	—	25	—	ns
16.	t_{DRH}	Data Read Hold Time	0	—	0	—	0	—	ns
17.	t_{STD1}	Ø ↓ to ST ↓ Delay	—	90	—	70	—	60	ns
18.	t_{STD2}	Ø ↓ to ST ↑ Delay	—	90	—	70	—	60	ns
19.	t_{WS}	$\overline{WAIT}$ Set-up Time to Ø ↓	40	—	40	—	30	—	ns
20.	t_{WH}	$\overline{WAIT}$ Hold Time from Ø ↓	40	—	40	—	30	—	ns
21.	t_{WDZ}	Ø↑ to Data Float Delay	—	95	—	70	—	60	ns
22.	t_{WRD1}	Ø↑ to $\overline{WR}$ ↓ Delay	—	65	—	60	—	50	ns
23.	t_{WDD}	Ø↓ to Write Data Delay Time	—	90	—	80	—	60	ns
24.	t_{WDS}	Write Data Set-up Time to $\overline{WR}$ ↓	40	—	20	—	15	—	ns
25.	t_{WRD2}	Ø↓ to $\overline{WR}$ ↑ Delay	—	80	—	60	—	50	ns
26.	t_{WRP}	$\overline{WR}$ Pulse Width	170	—	130	—	110	—	ns
26a		$\overline{WR}$ Pulse Width (I/O Write Cycle)	332	—	255	—	210	—	ns

	Symbol	Item	Z80180-6		Z80180-8		Z80180-10		Unit
			min	max	min	max	min	max	
27.	t_{WDH}	Write Data Hold Time from $(\overline{WR}\uparrow)$	40	—	15	—	10	—	ns
	t_{IOD1}	$\varnothing\downarrow$ to $\overline{IORQ}\downarrow$ Delay $\overline{IOC}=1$	—	60	—	50	—	50	ns
		$\varnothing\uparrow$ to $\overline{IORQ}\downarrow$ Delay $\overline{IOC}=0$	—	65	—	60	—	55	
29.	t_{IOD2}	$\varnothing\downarrow$ to $\overline{IORQ}\uparrow$ Delay	—	60	—	50	—	50	ns
30.	t_{IOD3}	$\overline{M1}\downarrow$ to $\overline{IORQ}\downarrow$ Delay	340	—	250	—	200	—	ns
31.	t_{INTS}	$\overline{INT}$ Set-up Time to $\varnothing\downarrow$	40	—	40	—	30	—	ns
32.	t_{INTH}	$\overline{INT}$ Hold Time from $\varnothing\downarrow$	40	—	40	—	30	—	ns
33.	t_{NMIW}	$\overline{NMI}$ Pulse Width	120	—	100	—	80	—	ns
34.	t_{BRS}	$\overline{BUSREQ}$ Set-up Time to $\varnothing\downarrow$	40	—	40	—	30	—	ns
35.	t_{BRH}	$\overline{BUSREQ}$ Hold Time from $\varnothing\downarrow$	40	—	40	—	30	—	ns
36.	t_{BAD1}	$\varnothing\uparrow$ to $\overline{BUSACK}\downarrow$ Delay	—	95	—	70	—	60	ns
37.	t_{BAD2}	$\varnothing\downarrow$ to $\overline{BUSACK}\uparrow$ Delay	—	95	—	70	—	60	ns
38.	t_{BZD}	$\varnothing\uparrow$ to Bus Floating Delay Time	—	125	—	90	—	80	ns
39.	t_{MEWH}	$\overline{MREQ}$ Pulse Width (HIGH)	110	—	90	—	70	—	ns
40.	t_{MEWL}	$\overline{MREQ}$ Pulse Width (LOW)	125	—	100	—	80	—	ns
41.	t_{RFD1}	$\overline{\varnothing}\uparrow$ to $\overline{RFSH}\downarrow$ Delay	—	90	—	80	—	60	ns
42.	t_{RFD2}	$\overline{\varnothing}\uparrow$ to $\overline{RFSH}\uparrow$ Delay	—	90	—	80	—	60	ns
43.	t_{HAD1}	$\varnothing\uparrow$ to $\overline{HALT}\downarrow$ Delay	—	90	—	80	—	50	ns
44.	t_{HAD2}	$\varnothing\uparrow$ $\overline{HALT}\uparrow$ Delay	—	90	—	80	—	50	ns
45.	t_{DRQS}	$\overline{DREQi}$ Set-up Time to $\varnothing\uparrow$	40	—	40	—	30	—	ns
46.	t_{DRQH}	$\overline{DREQi}$ Hold Time from $\varnothing\uparrow$	40	—	40	—	30	—	ns
47.	t_{TED1}	$\varnothing\downarrow$ to $\overline{TENDi}\downarrow$ Delay	—	70	—	60	—	50	ns
48.	t_{TED2}	$\varnothing\downarrow$ to $\overline{TENDi}\uparrow$ Delay	—	70	—	60	—	50	ns
49.	t_{ED1}	$\varnothing\uparrow$ to E $\uparrow$ Delay	—	95	—	70	—	60	ns
50.	t_{ED2}	$\varnothing\downarrow$ or $\uparrow$ to E $\downarrow$ Delay	—	95	—	70	—	60	ns
51.	P_{WEH}	E Pulse Width (HIGH)	75	—	65	—	55	—	ns
52.	P_{WEL}	E Pulse Width (LOW)	180	—	130	—	110	—	ns

	Symbol	Item	Z80180-6		Z80180-8		Z80180-10		Unit
			min	max	min	max	min	max	
53.	t_{Er}	Enable Rise Time	—	20	—	20	—	20	ns
54.	t_{Ef}	Enable Fall Time	—	20	—	20	—	20	ns
55.	t_{TOD}	$\varnothing \downarrow$ to Timer Output Delay	—	300	—	200	—	150	ns
56.	t_{STDI}	CSI/O Transmit Data Delay Time (Internal Clock Operation)	—	200	—	200	—	150	ns
57.	t_{STDE}	CSI/O Transmit Data Delay Time (External Clock Operation)	—	7.5tcyc +300	—	7.5tcyc +200	—	7.5tcyc +150	ns
58.	t_{SRSI}	CSI/O Receive Data Set-up Time (Internal Clock Operation)	1	—	1	—	1	—	tcyc
59.	t_{SRHI}	CSI/O Receive Data Hold Time (Internal Clock Operation)	1	—	1	—	1	—	tcyc
60.	t_{SRSE}	CSI/O Receive Data Set-up Time (External Clock Operation)	1	—	1	—	1	—	tcyc
61.	t_{SRHE}	CSI/O Receive Data Hold Time (External Clock Operation)	1	—	1	—	1	—	tcyc
62.	t_{RES}	$\overline{RESET}$ Set-up Time to $\varnothing \downarrow$	120	—	100	—	80	—	ns
63.	t_{REH}	$\overline{RESET}$ Hold Time from $\varnothing \downarrow$	80	—	70	—	50	—	ns
64.	t_{OSC}	Oscillator Stabilization Time	—	20	—	20	—	TBD	ms
65.	t_{EXr}	External Clock Rise Time (EXTAL)	—	25	—	25	—	25	ns
66.	t_{EXf}	External Clock Fall Time (EXTAL)	—	25	—	25	—	25	ns
67.	t_{Rr}	$\overline{RESET}$ Rise Time	—	50	—	50	—	50	ms
68.	t_{Rf}	$\overline{RESET}$ Fall Time	—	50	—	50	—	50	ms
69.	t_{Ir}	Input Rise Time (except EXTAL, $\overline{RESET}$)	—	100	—	100	—	100	ns
70.	t_{If}	Input Fall Time (except EXTAL, $\overline{RESET}$)	—	100	—	100	—	100	ns

TIMING DIAGRAMS
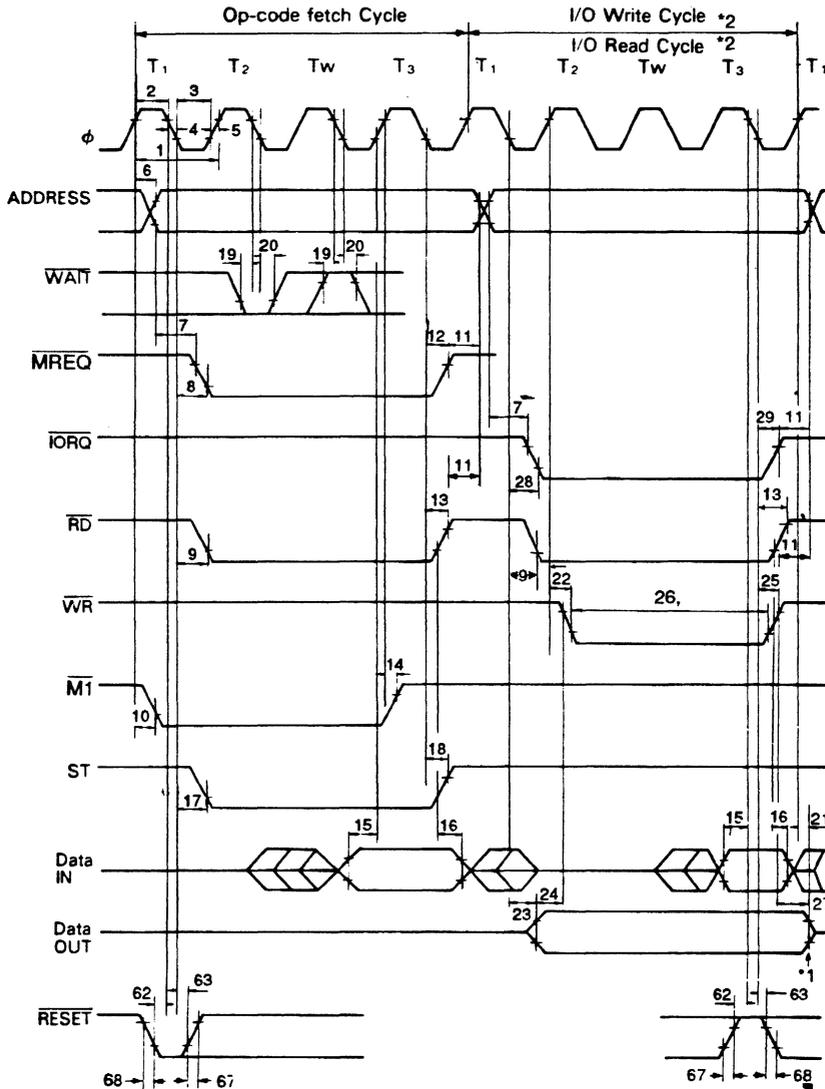
* 1 Output buffer is off at this point.
* 2 Memory Read/Write Cycle timing are the same as I/O Read/Write Cycle
    except there are no automatic wait states (Tw), and $\overline{MREQ}$ is active instead of $\overline{IORQ}$

CPU Timing ⎡Op-code fetch Cycle⎤
Memory Read Cycle
Memory Write Cycle
I/O Write Cycle
⎣I/O Read Cycle ⎦

*1 during $\overline{INT_0}$ acknowledge cycle
*2 during refresh cycle
*3 Output buffer is off at this point.
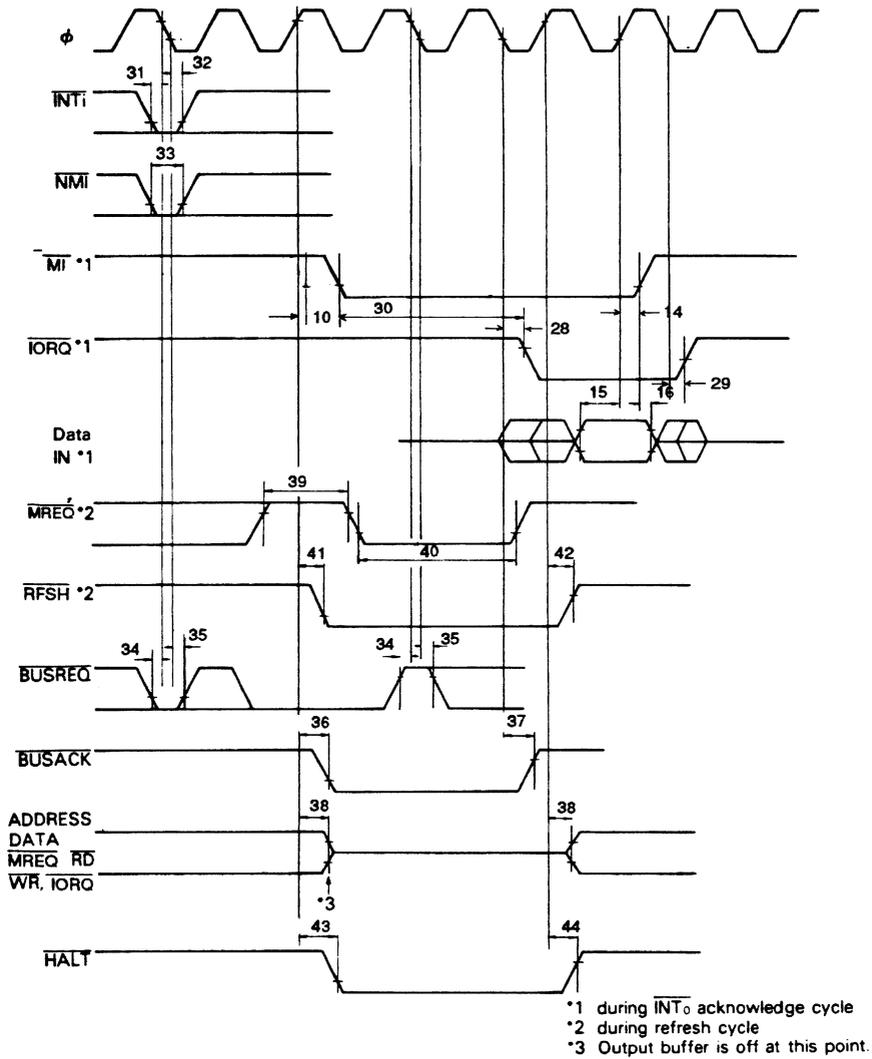
CPU Timing
{
$\overline{INT_0}$ Acknowledge cycle
Refresh Cycle
BUS RELEASE Mode
HALT Mode
SLEEP Mode
SYSTEM STOP Mode
}
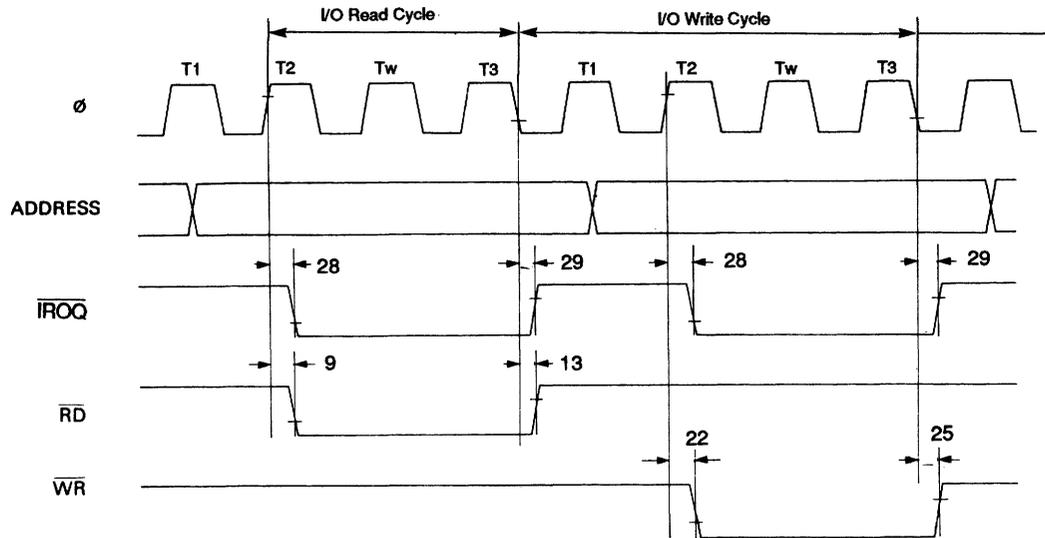
CPU Timing ($\overline{IOC}=0$)

$\left\{ \begin{array}{l} \text{I/O Read Cycle} \\ \text{I/O Write Cycle} \end{array} \right\}$
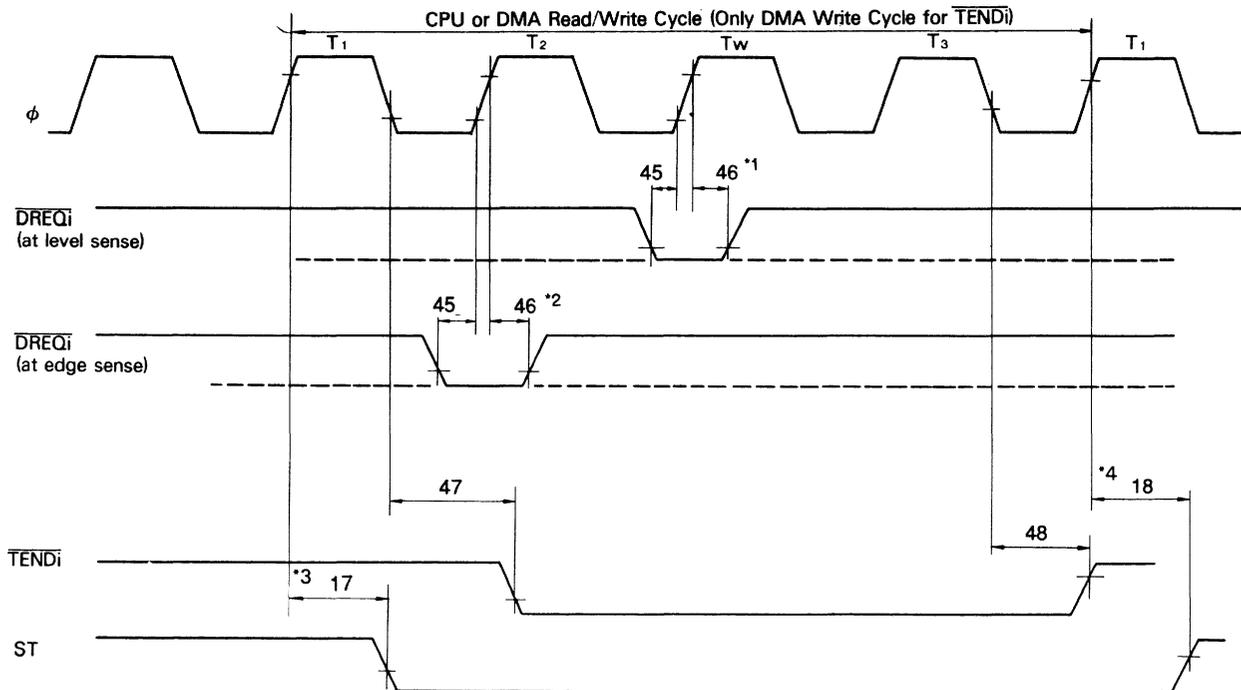
DMA Control Signals

*1 t_{DRQS} and t_{DRQH} are specified for the rising edge of clock followed by T_3.
*2 t_{DRQS} and t_{DRQH} are specified for the rising edge of clock.
*3 DMA cycle starts.
*4 CPU cycle starts.

E Clock Timing (Memory Read/Write Cycle)
(I/O Read/Write Cycle)

E Clock Timing (BUS RELEASE Mode)
(SLEEP Mode)
(SYSTEM STOP Mode)

T₂   T_W   T₃   T₁   T₂

φ

E
Example
(I/O read
→ Op-code fetch)

50
52
49
49
50
54
53
51
54

E
(I/O Write)

53
54
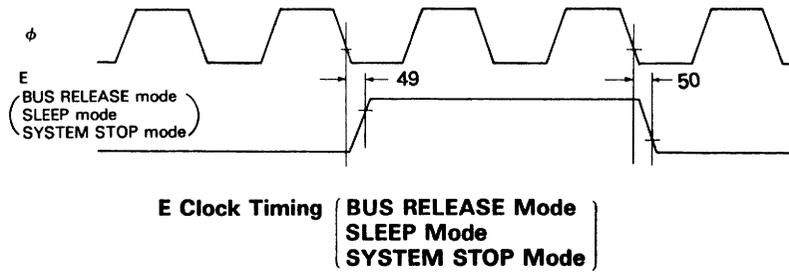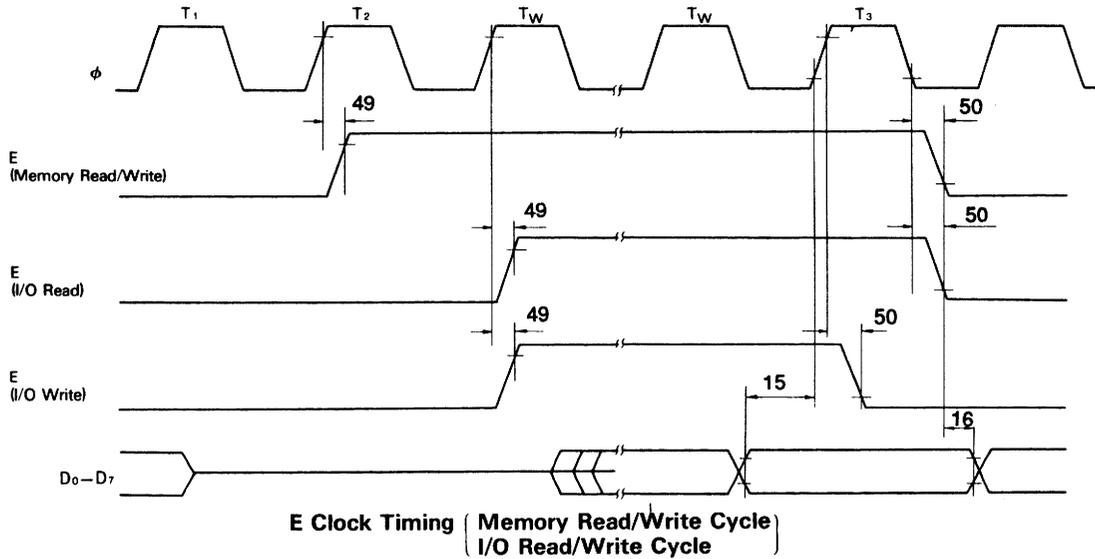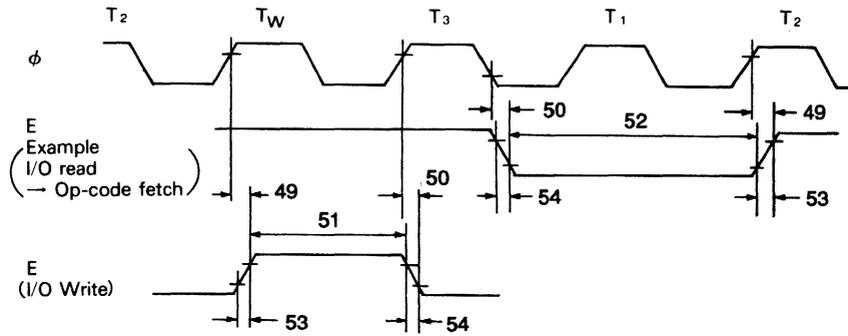
E Clock Timing $\left(\begin{array}{l}\text{Minimum timing example}\\ \text{of } P_{WEL} \text{ and } P_{WEH}\end{array}\right)$

φ

Timer Data
Reg.=0000H

A₁₈/TOUT

55
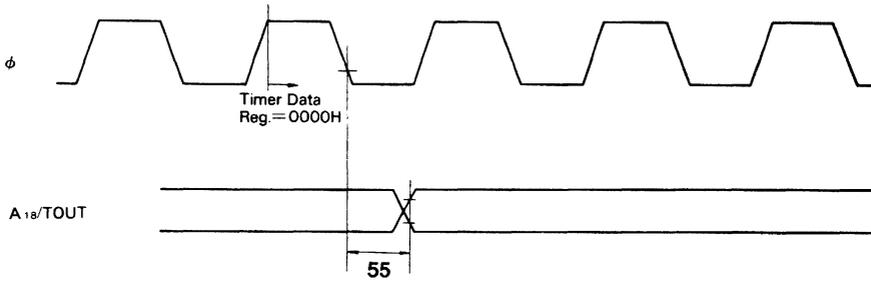
Timer Output Timing
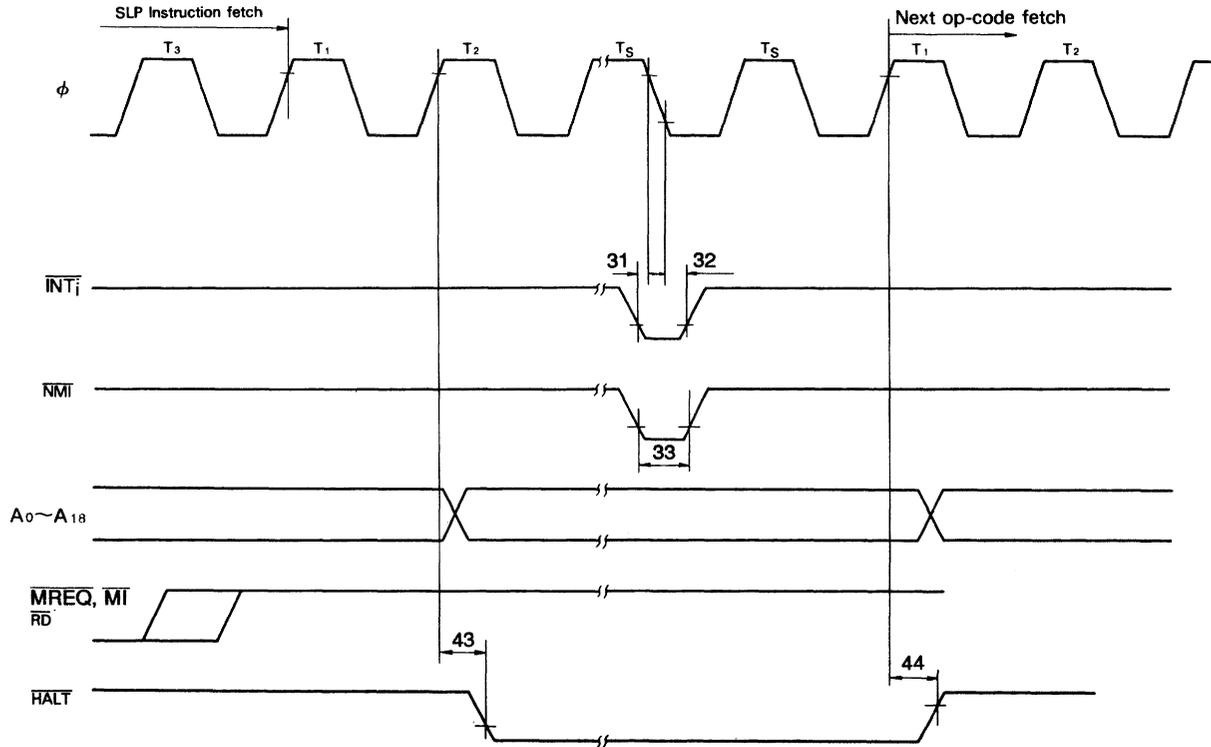
SLP Execution Cycle
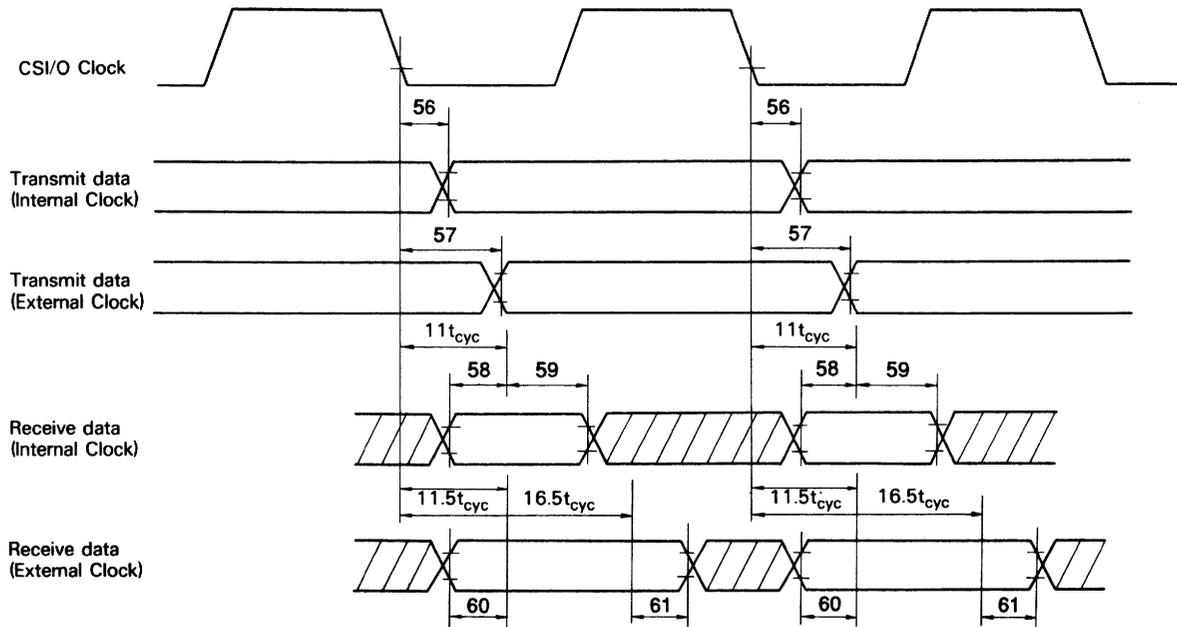
CSI/O Receive/Transmit Timing
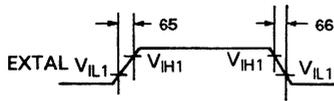
External Clock Rise Time and Fall Time
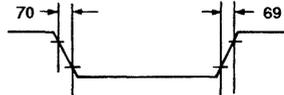
Input Rise Time and Fall Time (Except EXTAL, RESET)

STANDARD TEST CONDITIONS:

The DC Characteristics and Capacitance sections above apply to the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows in to the referenced pin.

All AC parameters assume a load capaitance of 100 pF. Add 10 ns delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for the address and control lines. AC timing measurements are referenced to 1.5 volts (except for CLOCK, which is referenced to the 10% and 90% points).

The Ordering Information section lists temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.
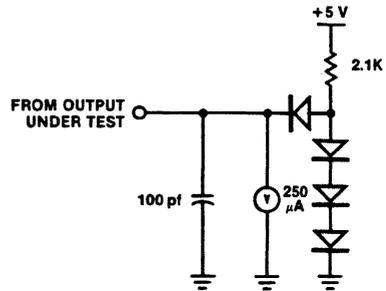
APPENDICES

A. Instruction Set

The following explains the symbols in instruction set.

1. Register

g, g', ww,xx, yy, and zz specify a register to be used. g and g' specify an 8-bit register. ww, xx, yy, and zz specify a pair of 16-bit registers. The following tables show the correspondence between symbols and registers.

g,g'	Reg.		ww	Reg.		xx	Reg.		yy	Reg.		zz	Reg.
000	B		00	BC		00	BC		00	BC		00	BC
001	C		01	DE		01	DE		01	DE		01	DE
010	D		10	HL		10	IX		10	IY		10	HL
011	E		11	SP		11	SP		11	SP		11	AF
100	H												
101	L												
111	A												

NOTE: Suffixed H and L to ww, xx, yy, zz (ex. wwH, IXL) indicate upper and lower 8-bit of the 16-bit register respectively.

2. Bit

b specifies a bit to be manipulated in the bit manipulation instruction. The following table shows the correspondence between b and bits.

b	Bit
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

3. Condition

f specifies the condition in program control instructions. The following shows the correspondence between f and conditions.

f	Condition	
000	NZ	non zero
001	Z	zero
010	NC	non carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign plus
111	M	sign minus

4. Restart Address

v specifies a restart address. The following table shows the correspondence between v and restart addresses.

v	Address
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

5. Flag

The following symbols show the flag conditions.

.	:	not affected
↑	:	affected
x	:	undefined
S	:	set to 1
R	:	reset to 0
P	:	parity
V	:	overflow

6. Miscellaneous

( )M	:	data in the memory address
( )I	:	data in the I/o address
m or n	:	8-bit data
mn	:	16-bit data
r	:	8-bit register
R	:	16-bit register
b.( )M	:	acontent of bit b in the memory address
b.gr	:	a content of bit b in the register gr
d or j	:	8-bit signed displacement
S	:	source addressing
D	:	destination addressing mode
.	:	AND operation
+	:	OR operation
⊕	:	EXCLUSIVE OR operation
**	:	added new instructions to Z80

1 . Data Manipulation Instructions

(1) Arithmetic and Logical Instructions (8-bit)

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
ADD	ADD A,g	10 000 g				S		D		1	4	Ar+gr→Ar	1	1	1	V	R	1
	ADD A,(HL)	10 000 110					S	D		1	6	Ar+(HL)$_m$→Ar	1	1	1	V	R	1
	ADD A,m	11 000 110 ⟨ m ⟩	S					D		2	6	Ar+m→Ar	1	1	1	V	R	1
	ADD A,(IX+d)	11 011 101 10 000 110 ⟨ d ⟩			S			D		3	14	Ar+(IX+d)$_m$→Ar	1	1	1	V	R	1
	ADD A,(IY+d)	11 111 101 10 000 110 ⟨ d ⟩			S			D		3	14	Ar+(IY+d)$_m$→Ar	1	1	1	V	R	1
ADC	ADC A,g	10 001 g				S		D		1	4	Ar+gr+c→Ar	1	1	1	V	R	1
	ADC A,(HL)	10 001 110					S	D		1	6	Ar+(HL)$_m$+c→Ar	1	1	1	V	R	1
	ADC A,m	11 001 110 ⟨ m ⟩	S					D		2	6	Ar+m+c→Ar	1	1	1	V	R	1
	ADC A,(IX+d)	11 011 101 10 001 110 ⟨ d ⟩			S			D		3	14	Ar+(IX+d)$_m$+c→Ar	1	1	1	V	R	1
	ADC A,(IY+d)	11 111 101 10 001 110 ⟨ d ⟩			S			D		3	14	Ar+(IY+d)$_m$+c→Ar	1	1	1	V	R	1
AND	AND g	10 100 g				S		D		1	4	Ar·gr→Ar	1	1	S	P	R	R
	AND (HL)	10 100 110					S	D		1	6	Ar·(HL)$_m$→Ar	1	1	S	P	R	R
	AND m	11 100 110 ⟨ m ⟩	S					D		2	6	Ar·m→Ar	1	1	S	P	R	R
	AND (IX+d)	11 011 101 10 100 110 ⟨ d ⟩			S			D		3	14	Ar·(IX+d)$_m$→Ar	1	1	S	P	R	R
	AND (IY+d)	11 111 101 10 100 110 ⟨ d ⟩			S			D		3	14	Ar·(IY+d)$_m$→Ar	1	1	S	P	R	R
Compare	CP g	10 111 g				S		D		1	4	Ar−gr	1	1	1	V	S	1
	CP (HL)	10 111 110					S	D		1	6	Ar−(HL)$_m$	1	1	1	V	S	1
	CP m	11 111 110 ⟨ m ⟩	S					D		2	6	Ar−m	1	1	1	V	S	1
	CP (IX+d)	11 011 101 10 111 110 ⟨ d ⟩			S			D		3	14	Ar−(IX+d)$_m$	1	1	1	V	S	1
	CP (IY+d)	11 111 101 10 111 110 ⟨ d ⟩			S			D		3	14	Ar−(IY+d)$_m$	1	1	1	V	S	1
COMPLE-MENT	CPL	00 101 111						S/D		1	3	$\overline{Ar}$→Ar	·	·	S	·	S	·
DEC	DEC g	00 g 101				S/D				1	4	gr−1→gr	1	1	1	V	S	·
	DEC (HL)	00 110 101					S/D			1	10	(HL)$_m$−1→(HL)$_m$	1	1	1	V	S	·
	DEC (IX+d)	11 011 101 00 110 101 ⟨ d ⟩			S/D					3	18	(IX+d)$_m$−1→(IX+d)$_m$	1	1	1	V	S	·
	DEC (IY+d)	11 111 101 00 110 101 ⟨ d ⟩			S/D					3	18	(IY+d)$_m$−1→(IY+d)$_m$	1	1	'1	V	S	·
INC	INC g	00 g 100				S/D				1	4	gr+1→gr	1	1	1	V	R	·
	INC (HL)	00 110 100					S/D			1	10	(HL)$_m$+1→(HL)$_m$	1	1	1	V	R	·
	INC (IX+d)	11 011 101 00 110 100 ⟨ d ⟩			S/D					3	18	(IX+d)$_m$+1→(IX+d)$_m$	1	1	1	V	R	·
	INC (IY+d)	11 111 101 00 110 100 ⟨ d ⟩			S/D					3	18	(IY+d)$_m$+1→(IY+d)$_m$	1	1	1	V	R	·

(to be continued)

Operation name	MNEMONICS	OP code	Addressing IMMED	EXT	IND	REG	REGI	IMP	REL	Bytes	States	Operation	Flag 7 S	6 Z	4 H	2 P/V	1 N	0 C
MULT	MLT ww **	11 101 101 / 01 ww1 100			S/D					2	17	wwHr×wwLr→ww₁	•	•	•	•	•	•
NEGATE	NEG	11 101 101 / 01 000 100						S/D		2	6	0−Ar→Ar	1	1	1	V	S	1
OR	OR g	10 110 g				S		D		1	4	Ar+gr→Ar	1	1	R	P	R	R
	OR (HL)	10 110 110					S	D		1	6	Ar+(HL)ₘ→Ar	1	1	R	P	R	R
	OR m	11 110 110 / ‹ m ›	S					D		2	6	Ar+m→Ar	1	1	R	P	R	R
	OR (IX+d)	11 011 101 / 10 110 110 / ‹ d ›			S			D		3	14	Ar+(IX+d)ₘ→Ar	1	1	R	P	R	R
	OR (IY+d)	11 111 101 / 10 110 110 / ‹ d ›			S			D		3	14	Ar+(IY+d)ₘ→Ar	1	1	R	P	R	R
SUB	SUB g	10 010 g				S		D		1	4	Ar−gr→Ar	1	1	1	V	S	1
	SUB (HL)	10 010 110					S	D		1	6	Ar−(HL)ₘ→Ar	1	1	1	V	S	1
	SUB m	11 010 110 / ‹ m ›	S					D		2	6	Ar−m→Ar	1	1	1	V	S	1
	SUB (IX+d)	11 011 101 / 10 010 110 / ‹ d ›			S			D		3	14	Ar−(IX+d)ₘ→Ar	1	1	1	V	S	1
	SUB (IY+d)	11 111 101 / 10 010 110 / ‹ d ›			S			D		3	14	Ar−(IY+d)ₘ→Ar	1	1	1	V	S	1
SUBC	SBC A,g	10 011 g				S		D		1	4	Ar−gr−c→Ar	1	1	1	V	S	1
	SBC A,(HL)	10 011 110					S	D		1	6	Ar−(HL)ₘ−c→Ar	1	1	1	V	S	1
	SBC A,m	11 011 110 / ‹ m ›	S					D		2	6	Ar−m−c→Ar	1	1	1	V	S	1
	SBC A,(IX+d)	11 011 101 / 10 011 110 / ‹ d ›			S			D		3	14	Ar−(IX+d)ₘ−c→Ar	1	1	1	V	S	1
	SBC A,(IY+d)	11 111 101 / 10 011 110 / ‹ d ›			S			D		3	14	Ar−(IY+d)ₘ−c→Ar	1	1	1	V	S	1
TEST	TST g **	11 101 101 / 00 g 100				S				2	7	Ar·gr	1	1	S	P	R	R
	TST (HL) **	11 101 101 / 00 110 100					S			2	10	Ar·(HL)ₘ	1	1	S	P	R	R
	TST m **	11 101 101 / 01 100 100 / ‹ m ›	S							3	9	Ar·m	1	1	S	P	R	R
XOR	XOR g	10 101 g				S		D		1	4	Ar⊕gr→Ar	1	1	R	P	R	R
	XOR (HL)	10 101 110					S	D		1	6	Ar⊕(HL)ₘ→Ar	1	1	R	P	R	R
	XOR m	11 101 110 / ‹ m ›	S					D		2	6	Ar⊕m→Ar	1	1	R	P	R	R
	XOR (IX+d)	11 011 101 / 10 101 110 / ‹ d ›			S			D		3	14	Ar⊕(IX+d)ₘ→Ar	1	1	R	P	R	R
	XOR (IY+d)	11 111 101 / 10 101 110 / ‹ d ›			S			D		3	14	Ar⊕(IY+d)ₘ→Ar	1	1	R	P	R	R

(2) Rotate and Shift Instructions

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
Rotate and Shift Data	RLA	00 010 111						S/D		1	3		·	·	R	·	R	I
	RL g	11 001 011 00 010 g				S/D				2	7		I	I	R	P	R	I
	RL (HL)	11 001 011 00 010 110					S/D			2	13		I	I	R	P	R	I
	RL (IX+d)	11 011 101 11 001 011 ⟨ d ⟩ 00 010 110			S/D					4	19		I	I	R	P	R	I
	RL (IY+d)	11 111 101 11 001 011 ⟨ d ⟩ 00 010 110			S/D					4	19		I	I	R	P	R	I
	RLCA	00 000 111						S/D		1	3		·	·	R	·	R	I
	RLC g	11 001 011 00 000 g				S/D				2	7		I	I	R	P	R	I
	RLC (HL)	11 001 011 00 000 110					S/D			2	13		I	I	R	P	R	I
	RLC (IX+d)	11 011 101 11 001 011 ⟨ d ⟩ 00 000 110			S/D					4	19		I	I	R	P	R	I
	RLC (IY+d)	11 111 101 11 001 011 ⟨ d ⟩ 00 000 110			S/D					4	19		I	I	R	P	R	I
	RLD	11 101 101 01 101 111						S/D		2	16		I	I	R	P	R	·
	RRA	00 011 111						S/D		1	3		·	·	R	·	R	I
	RR g	11 001 011 00 011 g				S/D				2	7		I	I	R	P	R	I
	RR (HL)	11 001 011 00 011 110					S/D			2	13		I	I	R	P	R	I
	RR (IX+d)	11 011 101 11 001 011 ⟨ d ⟩ 00 011 110			S/D					4	19		I	I	R	P	R	I
	RR (IY+d)	11 111 101 11 001 011 ⟨ d ⟩ 00 011 110			S/D					4	19		I	I	R	P	R	I
	RRCA	00 001 111						S/D		1	3		·	·	R	·	R	I
	RRC g	11 001 011 00 001 g				S/D				2	7		I	I	R	P	R	I
	RRC (HL)	11 001 011 00 001 110					S/D			2	13		I	I	R	P	R	I
	RRC (IX+d)	11 011 101 11 001 011 ⟨ d ⟩ 00 001 110			S/D					4	19		I	I	R	P	R	I
	RRC (IY+d)	11 111 101 11 001 011 ⟨ d ⟩ 00 001 110			S/D					4	19		I	I	R	P	R	I

(to be continued)

Operation name	MNEMONICS	OP code	Addressing IMMED	EXT	IND	REG	REGI	IMP	REL	Bytes	States	Operation	Flag 7 S	6 Z	4 H	2 P/V	1 N	0 C
Rotate and Shift Data	RRD	11 101 101 / 01 100 111						S/D		2	16		I	I	R	P	R	·
	SLA g	11 001 011 / 00 100 g				S/D				2	7		I	I	R	P	R	I
	SLA (HL)	11 001 011 / 00 100 110					S/D			2	13		I	I	R	P	R	I
	SLA (IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 00 100 110			S/D					4	19		I	I	R	P	R	I
	SLA (IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 00 100 110			S/D					4	19		I	I	R	P	R	I
	SRA g	11 001 011 / 00 101 g				S/D				2	7		I	I	R	P	R	I
	SRA (HL)	11 001 011 / 00 101 110					S/D			2	13		I	I	R	P	R	I
	SRA (IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 00 101 110			S/D					4	19		I	I	R	P	R	I
	SRA (IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 00 101 110			S/D					4	19		I	I	R	P	R	I
	SRL g	11 001 011 / 00 111 g				S/D				2	7		I	I	R	P	R	I
	SRL (HL)	11 001 011 / 00 111 110					S/D			2	3		I	I	R	P	R	I
	SRL (IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 00 111 110			S/D					4	19		I	I	R	P	R	I
	SRL (IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 00 111 110			S/D					4	19		I	I	R	P	R	I
Bit Set	SET b,g	11 001 011 / 11 b g				S/D				2	7	$1 \to b \cdot g_r$	·	·	·	·	·	·
	SET b,(HL)	11 001 011 / 11 b 110					S/D			2	13	$1 \to b \cdot (HL)_M$	·	·	·	·	·	·
	SET b,(IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 11 b 110			S/D					4	19	$1 \to b \cdot (IX+d)_M$	·	·	·	·	·	·
	SET b,(IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 11 b 110			S/D					4	19	$1 \to b \cdot (IY+d)_M$	·	·	·	·	·	·
Bit Reset	RES b,g	11 001 011 / 10 b g				S/D				2	7	$0 \to b \cdot g_r$	·	·	·	·	·	·
	RES b,(HL)	11 001 011 / 10 b 110					S/D			2	13	$0 \to b \cdot (HL)_M$	·	·	·	·	·	·
	RES b,(IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 10 b 110			S/D					4	19	$0 \to b \cdot (IX+d)_M$	·	·	·	·	·	·
	RES b,(IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 10 b 110			S/D					4	19	$0 \to b \cdot (IY+d)_M$	·	·	·	·	·	·
Bit Test	BIT b,g	11 001 011 / 01 b g				S				2	6	$\overline{b \cdot g_r} \to z$	X	I	S	X	R	·
	BIT b,(HL)	11 001 011 / 01 b 110					S			2	9	$\overline{b \cdot (HL)_M} \to z$	X	I	S	X	R	·
	BIT b,(IX+d)	11 011 101 / 11 001 011 / ⟨ d ⟩ / 01 b 110			S					4	15	$\overline{b \cdot (IX+d)_M} \to z$	X	I	S	X	R	·
	BIT b,(IY+d)	11 111 101 / 11 001 011 / ⟨ d ⟩ / 01 b 110			S					4	15	$\overline{b \cdot (IY+d)_M} \to z$	X	I	S	X	R	·

(4) Arithmetic Instructions (16-bit)

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
ADD	ADD HL,ww	00 ww1 001				S		D		1	7	$HL_R+ww_R \to HL_R$	·	·	X	·	R	1
	ADD IX,xx	11 011 101 / 00 xx1 001				S		D		2	10	$IX_R+xx_R \to IX_R$	·	·	X	·	R	1
	ADD IY,yy	11 111 101 / 00 yy1 001				S		D		2	10	$IY_R+yy_R \to IY_R$	·	·	X	·	R	1
ADC	ADC HL,ww	11 101 101 / 01 ww1 010				S		D		2	10	$HL_R+ww_R+c \to HL_R$	1	1	X	V	R	1
DEC	DEC ww	00 ww1 011				S/D				1	4	$ww_R-1 \to ww_R$	·	·	·	·	·	·
	DEC IX	11 011 101 / 00 101 011						S/D		2	7	$IX_R-1 \to IX_R$	·	·	·	·	·	·
	DEC IY	11 111 101 / 00 101 011						S/D		2	7	$IY_R-1 \to IY_R$	·	·	·	·	·	·
INC	INC ww	00 ww0 011				S/D				1	4	$ww_R+1 \to ww_R$	·	·	·	·	·	·
	INC IX	11 011 101 / 00 100 011						S/D		2	7	$IX_R+1 \to IX_R$	·	·	·	·	·	·
	INC IY	11 111 101 / 00 100 011						S/D		2	7	$IY_R+1 \to IY_R$	·	·	·	·	·	·
SBC	SBC HL,ww	11 101 101 / 01 ww0 010				S		D		2	10	$HL_R-ww_R-c \to HL_R$	1	1	X	V	S	1

2. Data Transfer Instructions

(1) 8-Bit Load

Operation name	MNEMONICS	OP code	IMMED	EXT	IND	REG	REGI	IMP	REL	Bytes	States	Operation	S (7)	Z (6)	H (4)	P/V (2)	N (1)	C (0)
Load 8-bit Data	LD A,I	11 101 101 / 01 010 111						S/D		2	6	$I_r \rightarrow A_r$	↕	↕	R	IEF	R	·
	LD A,R	11 101 101 / 01 011 111						S/D		2	6	$R_r \rightarrow A_r$	↕	↕	R	IEF	R	·
	LD A,(BC)	00 001 010					S	D		1	6	$(BC)_m \rightarrow A_r$ ①	·	·	·	·	·	·
	LD A,(DE)	00 011 010					S	D		1	6	$(DE)_m \rightarrow A_r$	·	·	·	·	·	·
	LD A,(mn)	00 111 010 / (n) / (m)		S				D		3	12	$(mn)_m \rightarrow A_r$	·	·	·	·	·	·
	LD I,A	11 101 101 / 01 000 111						S/D		2	6	$A_r \rightarrow I_r$	·	·	·	·	·	·
	LD R,A	11 101 101 / 01 001 111						S/D		2	6	$A_r \rightarrow R_r$	·	·	·	·	·	·
	LD (BC),A	00 000 010					D	S		1	7	$A_r \rightarrow (BC)_m$	·	·	·	·	·	·
	LD (DE),A	00 010 010					D	S		1	7	$A_r \rightarrow (DE)_m$	·	·	·	·	·	·
	LD (mn),A	00 110 010 / (n) / (m)		D				S		3	13	$A_r \rightarrow (mn)_m$	·	·	·	·	·	·
	LD g,g'	01 g g'				S/D				1	4	$g_r{}' \rightarrow g_r$	·	·	·	·	·	·
	LD g,(HL)	01 g 110				D	S			1	6	$(HL)_m \rightarrow g_r$	·	·	·	·	·	·
	LD g,m	00 g 110 / (m)	S			D				2	6	$m \rightarrow g_r$	·	·	·	·	·	·
	LD g,(IX+d)	11 011 101 / 01 g 110 / (d)			S	D				3	14	$(IX+d)_m \rightarrow g_r$	·	·	·	·	·	·
	LD g,(IY+d)	11 111 101 / 01 g 110 / (d)			S	D				3	14	$(IY+d)_m \rightarrow g_r$	·	·	·	·	·	·
	LD (HL),m	00 110 110 / (m)	S				D			2	9	$m \rightarrow (HL)_m$	·	·	·	·	·	·
	LD (IX+d),m	11 011 101 / 00 110 110 / (d) / (m)	S		D					4	15	$m \rightarrow (IX+d)_m$	·	·	·	·	·	·
	LD (IY+d),m	11 111 101 / 00 110 110 / (d) / (m)	S		D					4	15	$m \rightarrow (IY+d)_m$	·	·	·	·	·	·
	LD (HL),g	01 110 g				S	D			1	7	$g_r \rightarrow (HL)_m$	·	·	·	·	·	·
	LD (IX+d),g	11 011 101 / 01 110 g / (d)			D	S				3	15	$g_r \rightarrow (IX+d)_m$	·	·	·	·	·	·
	LD (IY+d),g	11 111 101 / 01 110 g / (d)			D	S				3	15	$g_r \rightarrow (IY+d)_m$	·	·	·	·	·	·

① In the case of R1 and Z Mask, interrupts are not sampled at the end of LD A, I or LD A, R.

(2) 16-Bit Load

Operation name	MNEMONICS	OP code	IMMED	EXT	IND	REG	REGI	IMP	REL	Bytes	States	Operation	7 S	6 Z	4 H	2 P/V	1 N	0 C
Load 16-bit Data	LD ww,mn	00 ww0 001 ⟨ n ⟩ ⟨ m ⟩	S			D				3	9	$mn \to ww_H$	·	·	·	·	·	·
	LD IX,mn	11 011 101 00 100 001 ⟨ n ⟩ ⟨ m ⟩	S					D		4	12	$mn \to IX_H$	·	·	·	·	·	·
	LD IY,mn	11 111 101 00 100 001 ⟨ n ⟩ ⟨ m ⟩	S					D		4	12	$mn \to IY_H$	·	·	·	·	·	·
	LD SP,HL	11 111 001						S/D		1	4	$HL_H \to SP_H$	·	·	·	·	·	·
	LD SP,IX	11 011 101 11 111 001						S/D		2	7	$IX_H \to SP_H$	·	·	·	·	·	·
	LD SP,IY	11 111 101 11 111 001						S/D		2	7	$IY_H \to SP_H$	·	·	·	·	·	·
	LD ww,(mn)	11 101 101 01 ww1 011 ⟨ n ⟩ ⟨ m ⟩		S		D				4	18	$(mn+1)_M \to wwHr$ $(mn)_M \to wwLr$	·	·	·	·	·	·
	LD HL,(mn)	00 101 010 ⟨ n ⟩ ⟨ m ⟩		S		D				3	15	$(mn+1)_M \to Hr$ $(mn)_M \to Lr$	·	·	·	·	·	·
	LD IX,(mn)	11 011 101 00 101 010 ⟨ n ⟩ ⟨ m ⟩		S		D				4	18	$(mn+1)_M \to IXHr$ $(mn)_M \to IXLr$	·	·	·	·	·	·
	LD IY,(mn)	11 111 101 00 101 010 ⟨ n ⟩ ⟨ m ⟩		S		D				4	18	$(mn+1)_M \to IYHr$ $(mn)_M \to IYLr$	·	·	·	·	·	·
	LD (mn),ww	11 101 101 01 ww0 011 ⟨ n ⟩ ⟨ m ⟩		D		S				4	19	$wwHr \to (mn+1)_M$ $wwLr \to (mn)_M$	·	·	·	·	·	·
	LD (mn),HL	00 100 010 ⟨ n ⟩ ⟨ m ⟩		D				S		3	16	$Hr \to (mn+1)_M$ $Lr \to (mn)_M$	·	·	·	·	·	·
	LD (mn),IX	11 011 101 00 100 010 ⟨ n ⟩ ⟨ m ⟩		D				S		4	19	$IXHr \to (mn+1)_M$ $IXLr \to (mn)_M$	·	·	·	·	·	·
	LD (mn),IY	11 111 101 00 100 010 ⟨ n ⟩ ⟨ m ⟩		D				S		4	19	$IYHr \to (mn+1)_M$ $IYLr \to (mn)_M$	·	·	·	·	·	·

(3) Block Transfer

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Block Transfer Search Data	CPD	11 101 101 10 101 001					S	S		2	12	Ar−(HL)$_H$ BC$_R$−1→BC$_R$ HL$_R$−1→HL$_R$	③ I	② I	I	② I	S	·
	CPDR	11 101 101 10 111 001					S	S		2	14 12	BC$_R$≠0 Ar≠(HL)$_H$ BC$_R$=0 or Ar=(HL)$_H$ Q {Ar−(HL)$_H$ / BC$_R$−1→BC$_R$ / HL$_R$−1→HL$_R$} Repeat Q until Ar=(HL)$_H$ or BC$_R$=0	③ I	② I	I	② I	S	·
	CPI	11 101 101 10 100 001					S	S		2	12	Ar−(HL)$_H$ BC$_R$−1→BC$_R$ HL$_R$+1→HL$_R$	③ I	② I	I	② I	S	·
	CPIR	11 101 101 10 110 001					S	S		2	14 12	BC$_R$≠0 Ar≠(HL)$_H$ BC$_R$=0 or Ar=(HL)$_H$ Q {Ar−(HL)$_H$ / BC$_R$−1→BC$_R$ / HL$_R$+1→HL$_R$} Repeat Q until Ar=(HL)$_H$ or BC$_R$=0	③ I	② I	I	② I	S	·
	LDD	11 101 101 10 101 000					S/D			2	12	(HL)$_H$→(DE)$_H$ BC$_R$−1→BC$_R$ DE$_R$−1→DE$_R$ HL$_R$−1→HL$_R$	·	·	R	② I	R	·
	LDDR	11 101 101 10 111 000					S/D			2	14 (BC$_R$≠0) 12 (BC$_R$=0)	(HL)$_H$→(DE)$_H$ Q {BC$_R$−1→BC$_R$ / DE$_R$−1→DE$_R$ / HL$_R$−1→HL$_R$} Repeat Q until BC$_R$=0	·	·	R	② R	R	·
	LDI	11 101 101 10 100 000					S/D			2	12	(HL)$_H$→(DE)$_H$ BC$_R$−1→BC$_R$ DE$_R$+1→DE$_R$ HL$_R$+1→HL$_R$	·	·	R	② I	R	·
	LDIR	11 101 101 10 110 000					S/D			2	14 (BC$_R$≠0) 12 (BC$_R$=0)	(HL)$_H$→(DE)$_H$ Q {BC$_R$−1→BC$_R$ / DE$_R$+1→DE$_R$ / HL$_R$+1→HL$_R$} Repeat Q until BC$_R$=0	·	·	R	② R	R	·

② P/V=0 : BC$_R$−1=0
　P/V=1 : BC$_R$−1≠0
③ Z=1 : Ar=(HL)$_H$
　Z=0 : Ar≠(HL)$_H$

(4) Stack and Exchange

Operation name	MNEMONICS	OP code	IMMED	EXT	IND	REG	REGI	IMP	REL	Bytes	States	Operation		Flag 7 S	6 Z	4 H	2 P/V	1 N	0 C
PUSH	PUSH zz	11 zz0 101				S		D		1	11	zzLr→(SP−2)ₘ		·	·	·	·	·	·
												zzHr→(SP−1)ₘ							
												SPₓ−2→SPₓ							
	PUSH IX	11 011 101						S/D		2	14	IXLr→(SP−2)ₘ		·	·	·	·	·	·
		11 100 101										IXHr→(SP−1)ₘ							
												SPₓ−2→SPₓ							
	PUSH IY	11 111 101						S/D		2	14	IYLr→(SP−2)ₘ		·	·	·	·	·	·
		11 100 101										IYHr→(SP−1)ₘ							
												SPₓ−2→SPₓ							
POP	POP zz	11 zz0 001				D		S		1	9	(SP+1)ₘ→zzHr ④		·	·	·	·	·	·
												(SP)ₘ→zzLr							
												SPₓ+2→SPₓ							
	POP IX	11 011 101						S/D		2	12	(SP+1)ₘ→IXHr		·	·	·	·	·	·
		11 100 001										(SP)ₘ→IXLr							
												SPₓ+2→SPₓ							
	POP IY	11 111 101						S/D		2	12	(SP+1)ₘ→IYHr		·	·	·	·	·	·
		11 100 001										(SP)ₘ→IYLr							
												SPₓ+2→SPₓ							
Exchange	EX AF,AF'	00 001 000						S/D		1	4	AFₐ↔AFₐ'		·	·	·	·	·	·
	EX DE,HL	11 101 011						S/D		1	3	DEₐ↔HLₐ		·	·	·	·	·	·
	EXX	11 011 001						S/D		1	3	BCₐ↔BCₐ'		·	·	·	·	·	·
												DEₐ↔DEₐ'							
												HLₐ↔HLₐ'							
	EX (SP),HL	11 100 011						S/D		1	16	Hr↔(SP+1)ₘ		·	·	·	·	·	·
												Lr↔(SP)ₘ							
	EX (SP),IX	11 011 101						S/D		2	19	IXHr↔(SP+1)ₘ		·	·	·	·	·	·
		11 100 011										IXLr↔(SP)ₘ							
	EX (SP),IY	11 111 101						S/D		2	19	IYHr↔(SP+1)ₘ		·	·	·	·	·	·
		11 100 011										IYLr↔(SP)ₘ							

④ In the case of POP AF, Flag is written a current contents of the stack.

3 . Program Control Instructions

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
Call	CALL mn	11 001 101 < n > < m >		D						3	16	PCHr→(SP−1)ᴍ PCLr→(SP−2)ᴍ mn→PCₐ SPₛ−2→SPₛ	·	·	·	·	·	·
	CALL f,mn	11 f 100 < n > < m >		D						3	6 (f : false) 16 (f : true)	continue : f is false CALL mn : f is true	·	·	·	·	·	·
Jump	DJNZ j	00 010 000 < j-2 >							D	2 2	9 (Br≠0) 7 (Br=0)	Br−1→Br continue : Br=0 PCₐ+j→PCₐ : Br≠0	·	·	·	·	·	·
	JP f,mn	11 f 010 < n > < m >		D						3 3	6 (f : false) 9 (f : true)	mn→PCₐ : f is true continue : f is false	·	·	·	·	·	·
	JP mn	11 000 011 < n > < m >		D						3	9	mn→PCₐ	·	·	·	·	·	·
	JP (HL)	11 101 001				D				1	3	HLₐ→PCₐ	·	·	·	·	·	·
	JP (IX)	11 011 101 11 101 001				D				2	6	IXₐ→PCₐ	·	·	·	·	·	·
	JP (IY)	11 111 101 11 101 001				D				2	6	IYₐ→PCₐ	·	·	·	·	·	·
	JR j	00 011 000 < j-2 >							D	2	8	PCₐ+j→PCₐ	·	·	·	·	·	·
	JR C,j	00 111 000 < j-2 >							D	2 2	6 8	continue : C=0 PCₐ+j→PCₐ : C=1	·	·	·	·	·	·
	JR NC,j	00 110 000 < j-2 >							D	2 2	6 8	continue : C=1 PCₐ+j→PCₐ : C=0	·	·	·	·	·	·
	JR Z,j	00 101 000 < j-2 >							D	2 2	6 8	continue : Z=0 PCₐ+j→PCₐ : Z=1	·	·	·	·	·	·
	JR NZ,j	00 100 000 < j-2 >							D	2 2	6 8	continue : Z=1 PCₐ+j→PCₐ : Z=0	·	·	·	·	·	·
Return	RET	11 001 001						D		1	9	(SP)ᴍ→PCLr (SP+1)ᴍ→PCHr SPₛ+2→SPₛ	·	·	·	·	·	·
	RET f	11 f 000						D		1 1	5 (f : false) 10 (f : true)	continue : f is false RET : f is true	·	·	·	·	·	·
	RETI	11 101 101 01 001 101						D		2	12 (R0, R1) 22 (Z)	(SP)ᴍ→PCLr (SP+1)ᴍ→PCHr SPₛ+2→SPₛ	·	·	·	·	·	·
	RETN	11 101 101 01 000 101						D		2	12	(SP)ᴍ→PCLr (SP+1)ᴍ→PCHr SPₛ+2→SPₛ IEFₜ→IEFᵢ	·	·	·	·	·	·
Restart	RST v	11 v 111							D	1	11	PCHr→(SP−1)ᴍ PCLr→(SP−2)ᴍ 0→PCHr v→PCLr SPₛ−2→SPₛ	·	·	·	·	·	·

4 . I/O Instructions

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
INPUT	IN A,(m)	11 011 011 ⟨ m ⟩						D	S	2	9	(Am)₁→Ar m→A₀~A₇ Ar→A₈~A₁₅	·	·	·	·	·	·
	IN g,(C)	11 101 101 01 g 000				D			S	2	9	(BC)₁→gr g=110 : Only the flags will change. Cr→A₀~A₇ Br→A₈~A₁₅	I	I	R	P	R	·
	IN0 g,(m) **	11 101 101 00 g 000 ⟨ m ⟩				D			S	3	12	(00m)₁→gr g=110 : Only the flags will change. m→A₀~A₇ 00→A₈~A₁₅	I	I	R	P	R	·
	IND	11 101 101 10 101 010					D		S	2	12	(BC)₁→(HL)ₘ HL₍₁₆₎−1→HL₍₁₆₎ Br−1→Br Cr→A₀~A₇ Br→A₈~A₁₅	X	⑤ I	X	X	⑥ I	X
	INDR	11 101 101 10 111 010					D		S	2	14(Br≠0) 12(Br=0)	Q ⎰(BC)₁→(HL)ₘ ⎱HL₍₁₆₎−1→HL₍₁₆₎ ⎰Br−1→Br Repeat Q until Br=0 Cr→A₀~A₇ Br→A₈~A₁₅	X	S	X	X	⑥ I	X
	INI	11 101 101 10 100 010					D		S	2	12	(BC)₁→(HL)ₘ HL₍₁₆₎+1→HL₍₁₆₎ Br−1→Br Cr→A₀~A₇ Br→A₈~A₁₅	X	⑤ I	X	X	⑥ I	X
	INIR	11 101 101 10 110 010					D		S	2	14(Br≠0) 12(Br=0)	Q ⎰(BC)₁→(HL)ₘ ⎱HL₍₁₆₎+1→HL₍₁₆₎ ⎰Br−1→Br Repeat Q until Br=0 Cr→A₀~A₇ Br→A₈~A₁₅	X	S	X	X	⑥ I	X

⑤ Z=1 : Br−1=0
 Z=0 : Br−1≠0
⑥ N=1 : MSB of Data=1
 N=0 : MSB of Data=0

(to be continued)

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
OUTPUT	OUT (m),A	11 010 011 〈 m 〉						S	D	2	10	Ar→(Am)₁ m→A₀~A₇ Ar→A₈~A₁₅	·	·	·	·	·	·
	OUT (C),g	11 101 101 01 g 001				S			D	2	10	gr→(BC)₁ Cr→A₀~A₇ Br→A₈~A₁₅	·	·	·	·	·	·
	OUT0 (m),g **	11 101 101 00 g 001 〈 m 〉				S			D	3	13	gr→(00m)₁ m→A₀~A₇ 00→A₈~A₁₅	·	·	·	·	·	·
	OTDM **	11 101 101 10 001 011					S		D	2	14	(HL)₁₆→(00C)₁ HL₁₆−1→HL₁₆ Cr−1→Cr Br−1→Br Cr→A₀~A₇ 00→A₈~A₁₅	⑤ I	I	I	P	⑥ I	I
	OTDMR **	11 101 101 10 011 011					S		D	2	16(Br≠0) 14(Br=0)	Q {(HL)₁₆→(00C)₁ HL₁₆−1→HL₁₆ Cr−1→Cr Br−1→Br} Repeat Q until Br=0 Cr→A₀~A₇ 00→A₈~A₁₅	R	S	R	S	⑥ I	R
	OTDR	11 101 101 10 111 011					S		D	2	14(Br≠0) 12(Br=0)	Q {(HL)₁₆→(BC)₁ HL₁₆−1→HL₁₆ Br−1→Br} Repeat Q until Br=0 Cr→A₀~A₇ Br→A₈~A₁₅	X	S	X	X	⑥ I	X
	OUTI	11 101 101 10 100 011					S		D	2	12	(HL)₁₆→(BC)₁ HL₁₆+1→HL₁₆ Br−1→Br Cr→A₀~A₇ Br→A₈~A₁₅	X	⑤ I	X	X	⑥ I	X
	OTIR	11 101 101 10 110 011					S		D	2	14(Br≠0) 12(Br=0)	Q {(HL)₁₆→(BC)₁ HL₁₆+1→HL₁₆ Br−1→Br} Repeat Q until Br=0 Cr→A₀~A₇ Br→A₈~A₁₅	X	S	X	X	⑥ I	X
	TSTIO m **	11 101 101 01 110 100 〈 m 〉	S						S	3	12	(00C)₁ · m Cr→A₀~A₇ 00→A₈~A₁₅	I	I	S	P	R	R
	OTIM **	11 101 101 10 000 011					S		D	2	14	(HL)₁₆→(00C)₁ HL₁₆+1→HL₁₆ Cr+1→Cr Br−1→Br Cr→A₀~A₇ 00→A₈~A₁₅	⑤ I	I	I	P	⑥ I	I
	OTIMR **	11 101 101 10 010 011					S		D	2	16(Br≠0) 14(Br=0)	Q {(HL)₁₆→(00C)₁ HL₁₆+1→HL₁₆ Cr+1→Cr Br−1→Br} Repeat Q until Br=0 Cr→A₀~A₇ 00→A₈~A₁₅	R	S	R	S	⑥ I	R
	OUTD	11 101 101 10 101 011					S		D	2	12	(HL)₁₆→(BC)₁ HL₁₆−1→HL₁₆ Br−1→Br Cr→A₀~A₇ Br→A₈~A₁₅	X	⑤ I	X	X	⑥ I	X

⑤ Z=1 : Br−1=0
   Z=0 : Br−1≠0
⑥ N=1 : MSB of Data=1
   N=0 : MSB of Data=0

5 . Special Control Instructions

Operation name	MNEMONICS	OP code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	IND	REG	REGI	IMP	REL				7 S	6 Z	4 H	2 P/V	1 N	0 C
Special Function	DAA	00 100 111						S/D		1	4	Decimal Adjust Accumulator	I	I	I	P	·	I
Carry Control	CCF	00 111 111								1	3	C̄→C	·	·	R	·	R	I
	SCF	00 110 111								1	3	1→C	·	·	R	·	R	S
CPU Control	DI	11 110 011								1	3	0→IEF₁, 0→IEF₂ ⑦	·	·	·	·	·	·
	EI	11 111 011								1	3	1→IEF₁, 1→IEF₂ ⑦	·	·	·	·	·	·
	HALT	01 110 110								1	3	CPU halted	·	·	·	·	·	·
	IM 0	11 101 101 01 000 110								2	6	Interrupt mode 0	·	·	·	·	·	·
	IM 1	11 101 101 01 010 110								2	6	Interrupt mode 1	·	·	·	·	·	·
	IM 2	11 101 101 01 011 110								2	6	Interrupt mode 2	·	·	·	·	·	·
	NOP	00 000 000								1	3	No operation	·	·	·	·	·	·
	SLP **	11 101 101 01 110 110								2	8	Sleep	·	·	·	·	·	·

⑦ Interrupts are not sampled at the end of DI or EI.

B. Instruction Summary in Alphabetical Order

•• : Added new instructions to Z80

MNEMONICS	Bytes	Machine Cycles	States
ADC A,m	2	2	6
ADC A,g	1	2	4
ADC A, (HL)	1	2	6
ADC A, (IX+d)	3	6	14
ADC A, (IY+d)	3	6	14
ADD A,m	2	2	6
ADD A,g	1	2	4
ADD A, (HL)	1	2	6
ADD A, (IX+d)	3	6	14
ADD A, (IY+d)	3	6	14
ADC HL,ww	2	6	10
ADD HL,ww	1	5	7
ADD IX,xx	2	6	10
ADD IY,yy	2	6	10
AND m	2	2	6
AND g	1	2	4
AND (HL)	1	2	6
AND (IX+d)	3	6	14
AND (IY+d)	3	6	14
BIT b, (HL)	2	3	9
BIT b, (IX+d)	4	5	15
BIT b, (IY+d)	4	5	15
BIT b,g	2	2	6
CALL f,mn	3	2	6
			(If condition is false)
	3	6	16
			(If condition is true)

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
CALL mn	3	6	16
CCF	1	1	3
CPD	2	6	12
CPDR	2	8	14
			(If $BC_R \neq 0$ and $Ar \neq (HL)_M$)
	2	6	12
			(If $BC_R = 0$ or $Ar = (HL)_M$)
CP (HL)	1	2	6
CPI	2	6	12
CPIR	2	8	14
			(If $BC_R \neq 0$ and $Ar \neq (HL)_M$)
	2	6	12
			(If $BC_R = 0$ or $Ar = (HL)_M$)
CP (IX+d)	3	6	14
CP (IY+d)	3	6	14
CPL	1	1	3
CP m	2	2	6
CP g	1	2	4
DAA	1	2	4
DEC (HL)	1	4	10
DEC IX	2	3	7
DEC IY	2	3	7
DEC (IX+d)	3	8	18
DEC (IY+d)	3	8	18
DEC g	1	2	4
DEC ww	1	2	4
DI	1	1	3

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
DJNZ j	2	5	9 (If Br≠0)
	2	3	7 (If Br=0)
EI	1	1	3
EX AF,AF'	1	2	4
EX DE,HL	1	1	3
EX (SP),HL	1	6	16
EX (SP),IX	2	7	19
EX (SP),IY	2	7	19
EXX	1	1	3
HALT	1	1	3
IM 0	2	2	6
IM 1	2	2	6
IM 2	2	2	6
INC g	1	2	4
INC (HL)	1	4	10
INC (IX+d)	3	8	18
INC (IY+d)	3	8	18
INC ww	1	2	4
INC IX	2	3	7
INC IY	2	3	7
IN A,(m)	2	3	9
IN g,(C)	2	3	9
INI	2	4	12
INIR	2	6	14 (If Br≠0)
	2	4	12 (If Br=0)
IND	2	4	12
INDR	2	6	14 (If Br≠0)

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
INDR	2	4	12 (If Br=0)
INO g,(m)**	3	4	12
JP f,mn	3	2	6
			(If f is false)
	3	3	9
			(If f is true)
JP (HL)	1	1	3
JP (IX)	2	2	6
JP (IY)	2	2	6
JP mn	3	3	9
JR j	2	4	8
JR C,j	2	2	6
			(If condition is false)
	2	4	8
			(If condition is true)
JR NC,j	2	2	6
			(If condition is false)
	2	4	8
			(If condition is true)
JR Z,j	2	2	6
			(If condition is false)
	2	4	8
			(If condition is true)
JR NZ,j	2	2	6
			(If condition is false)
	2	4	8
			(If condition is true)

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
LD A, (BC)	1	2	6
LD A, (DE)	1	2	6
LD A,I	2	2	6
LD A, (mn)	3	4	12
LD A,R	2	2	6
LD (BC),A	1	3	7
LDD	2	4	12
LD (DE),A	1	3	7
LD ww,mn	3	3	9
LD ww,(mn)	4	6	18
LDDR	2	6	14 (If $BC_R \neq 0$)
	2	4	12 (If $BC_R = 0$)
LD (HL),m	2	3	9
LD HL,(mn)	3	5	15
LD (HL),g	1	3	7
LDI	2	4	12
LD I,A	2	2	6
LDIR	2	6	14 (If $BC_R \neq 0$)
	2	4	12 (If $BC_R = 0$)
LD IX,mn	4	4	12
LD IX,(mn)	4	6	18
LD (IX+d),m	4	5	15
LD (IX+d),g	3	7	15
LD IY,mn	4	4	12
LD IY,(mn)	4	6	18
LD (IY+d),m	4	5	15
LD (IY+d),g	3	7	15

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
LD (mn),A	3	5	13
LD (mn),ww	4	7	19
LD (mn),HL	3	6	16
LD (mn),IX	4	7	19
LD (mn),IY	4	7	19
LD R,A	2	2	6
LD g,(HL)	1	2	6
LD g,(IX+d)	3	6	14
LD g,(IY+d)	3	6	14
LD g,m	2	2	6
LD g,g'	1	2	4
LD SP,HL	1	2	4
LD SP,IX	2	3	7
LD SP,IY	2	3	7
MLT ww**	2	13	17
NEG	2	2	6
NOP	1	1	3
OR (HL)	1	2	6
OR (IX+d)	3	6	14
OR (IY+d)	3	6	14
OR m	2	2	6
OR g	1	2	4
OTDM**	2	6	14
OTDMR**	2	8	16 (If Br≠0)
	2	6	14 (If Br=0)
OTDR	2	6	14 (If Br≠0)
	2	4	12 (If Br=0)

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
OTIM**	2	6	14
OTIMR**	2	8	16 (If Br≠0)
	2	6	14 (If Br=0)
OTIR	2	6	14 (If Br≠0)
	2	4	12 (If Br=0)
OUTD	2	4	12
OUTI	2	4	12
OUT (m),A	2	4	10
OUT (C),g	2	4	10
OUTO (m),g **	3	5	13
POP IX	2	4	12
POP IY	2	4	12
POP zz	1	3	9
PUSH IX	2	6	14
PUSH IY	2	6	14
PUSH zz	1	5	11
RES b,(HL)	2	5	13
RES b,(IX+d)	4	7	19
RES b,(IY+d)	4	7	19
RES b,g	2	3	7
RET	1	3	9
RET f	1	3	5
			(If condition is false)
	1	4	10
			(If condition is true)
RETI	2	4 (R0, R1)	12 (R0, R1)
		10 (Z)	22 (Z)
RETN	2	4	12

MNEMONICS	Bytes	Machine Cycles	States
RLA	1	1	3
RLCA	1	1	3
RLC (HL)	2	5	13
RLC (IX+d)	4	7	19
RLC (IY+d)	4	7	19
RLC g	2	3	7
RLD	2	8	16
RL (HL)	2	5	13
RL (IX+d)	4	7	19
RL (IY+d)	4	7	19
RL g	2	3	7
RRA	1	1	3
RRCA	1	1	3
RRC (HL)	2	5	13
RRC (IX+d)	4	7	19
RRC (IY+d)	4	7	19
RRC g	2	3	7
RRD	2	8	16
RR (HL)	2	5	13
RR (IX+d)	4	7	19
RR (IY+d)	4	7	19
RR g	2	3	7
RST v	1	5	11
SBC A,(HL)	1	2	6
SBC A,(IX+d)	3	6	14
SBC A,(IY+d)	3	6	14
SBC A,m	2	2	6

(to be continued)

MNEMONICS	Bytes	Machine Cycles	States
SBC A,g	1	2	4
SBC HL,ww	2	6	10
SCF	1	1	3
SET b,(HL)	2	5	13
SET b,(IX+d) ·	4	7	19
SET b,(IY+d)	4	7	19
SET b,g	2	3	7
SLA (HL)	2	5	13
SLA (IX+d)	4	7	19
SLA (IY+d)	4	7	19
SLA g	2	3	7
SLP**	2	2	8
SRA (HL)	2	5	13
SRA (IX+d)	4	7	19
SRA (IY+d)	4	7	19
SRA g	2	3	7
SRL (HL)	2	5	13
SRL (IX+d)	4	7	19
SRL (IY+d)	4	7	19
SRL g	2	3	7
SUB (HL)	1	2	6
SUB (IX+d)	3	6	14
SUB (IY+d)	3	6	14
SUB m	2	2	6
SUB g	1	2	4
**TSTIO m	3	4	12
**TST g	2	3	7

MNEMONICS	Bytes	Machine Cycles	States
TST m**	3	3	9
TST (HL)**	2	4	10
XOR (HL)	1	2	6
XOR (IX+d)	3	6	14
XOR (IY+d)	3	6	14
XOR m	2	2	6
XOR g	1	2	4

C. Op-code Map

Table 1 1st op-code map
Instruction format : <u>XX</u>

ww (LO=ALL)					g (LO=0~7)								LO=0~7				
BC	DE	HL	SP										BC	DE	HL	AF	zz
				B	D	H	(HL)	B	D	H	(HL)		NZ	NC	PO	P	f
													00H	10H	20H	30H	v

Main grid (HI=ALL, S):

HI→			0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
	LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
B	0000	0	NOP	DJNZ j	JR NZ,j	JR NC,j									RET f				0
C	0001	1	LD ww, mn							NOTE1)					POP zz				1
D	0010	2	LD (ww), A		LD(mn) ,HL	LD(mn) ,A									JP f, mn				2
E	0011	3	INC ww				LD g, s				ADD A	SUB s	AND s	OR s	JP mn	OUT(m) ,A	EX(SP) ,HL	DI	3
H	0100	4	INC g			NOTE1)					,s				CALL f, mn				4
L	0101	5	DEC g			NOTE1)									PUSH zz				5
(HL)	0110	6	LD g, m			NOTE1)	NOTE2)			HALT	NOTE2)	NOTE2)	NOTE2)	NOTE2)	ADD A,m	SUB m	AND m	OR m	6
A	0111	7	RLCA	RLA	DAA	SCF									RST v				7
B	1000	8	EXAF,AF'	JR j	JR Z,j	JR C,j									RET f				8
C	1001	9	ADD HL, ww												RET	EXX	JP (HL)	LD SP, HL	9
D	1010	A	LD A, (ww)		LD HL, (mn)	LD A, (mn)									JP f, mn				A
E	1011	B	DEC ww				LD g, s				ADC A	SBC A	XOR s	CP s	Table2	IN A, (m)	EX DE,HL	EI	B
H	1100	C	INC g								,s	,s			CALL f, mn				C
L	1101	D	DEC g												CALL mn	NOTE3)	Table3	NOTE3)	D
(HL)	1110	E	LD g, m				NOTE2)				NOTE2)	NOTE2)	NOTE2)	NOTE2)	ADC A,m	SBC A,m	XOR m	CP m	E
A	1111	F	RRCA	RRA	CPL	CCF									RST v				F

Bottom header:

			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
			C	E	L	A	C	E	L	A					Z	C	PE	M	f
			g (LO=8~F)												08H	18H	28H	38H	v
															LO=8~F				

83

NOTE1) (HL) replaces g.
   2) (HL) replaces s.
   3) If DDH is supplemented as 1st op-code for the instructions which have HL or (HL) as an operand in Table 1, the instructions are executed replacing HL with IX and (HL) with (IX+d).

   ex.   22H : LD (mn), HL
          DDH 22H : LD (mn), IX

   If FDH is supplemented as 1st op-code for the instructions which have HL or (HL) as an operand in Table 1, the instructions are executed replacing HL with IY and (HL) with (IY+d).

   ex.   34H : INC (HL)
          FDH 34H : INC (IY+d)

   However, JP (HL) and EX DE, HL are exception and note the followings.

   If DDH is supplemented as 1st op-code for JP (HL), (IX) replaces (HL) as operand and JP (IX) is executed.

   If FDH is supplemented as 1st op-code for JP (HL), (IY) replaces (HL) as operand and JP (IY) is executed.

   Even if DDH or FDH is supplemented as 1st op-code for EX DE, HL, HL is not replaced and the instruction is regarded as illegal instruction.

Table 2 2nd op-code map
Instruction format : **CB XX**

		HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
			b (LO=0~7)				0	2	4	6	0	2	4	6	0	2	4	6	
	LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
B	0000	0																	0
C	0001	1																	1
D	0010	2																	2
E	0011	3																	3
H	0100	4	RLC g	RL g	SLA g		BIT b,g				RES b,g				SET b,g				4
L	0101	5																	5
(HL)	0110	6	NOTE1)	NOTE1)	NOTE1)		NOTE 1)				NOTE 1)				NOTE 1)				6
A	0111	7																	7
B	1000	8																	8
C	1001	9																	9
D	1010	A																	A
E	1011	B																	B
H	1100	C	RRC g	RR g	SRA g	SRL g	BIT b,g				RES b,g				SET b,g				C
L	1101	D																	D
(HL)	1110	E	NOTE1)	NOTE1)	NOTE1)	NOTE1)	NOTE 1)				NOTE 1)				NOTE 1)				E
A	1111	F																	F
			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
							1	3	5	7	1	3	5	7	1	3	5	7	
							b (LO=8~F)												

(HI = ALL) g

NOTE 1) If DDH is supplemented as 1st op-code for the instructions which have (HL) as operand in Table 2, the instructions are executed replacing (HL) with (IX+d).

If FDH is supplemented as 1st op-code for the instructions which have (HL) as operand in Table 2, the instructions are executed replacing (HL) with (IY+d).

Table 3 2nd op-code map

Instruction format : ED XX

					ww (LO=ALL)											
					BC	DE	HL	SP								
	g (LO=0~7)															
	B	D	H		B	D	H									
HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000 0	INO g, (m)				IN g, (C)						LDI	LDIR				
0001 1	OUTO (m),g				OUT (C),g						CPI	CPIR				
0010 2					SBC HL,ww						INI	INIR				
0011 3					LD (mn),ww				OTIM	OTIMR	OUTI	OTIR				
0100 4	TST g			TST (HL)	NEG		TST m	TSTIO m								
0101 5					RETN											
0110 6					IM 0	IM 1		SLP								
0111 7					LD I,A	LD A,I	RRD									
1000 8	INO g, (m)				IN g, (C)						LDD	LDDR				
1001 9	OUTO (m),g				OUT (C),g						CPD	CPDR				
1010 A					ADC HL,ww						IND	INDR				
1011 B					LD ww, (mn)				OTDM	OTDMR	OUTD	OTDR				
1100 C	TST g				MLT ww											
1101 D					RETI											
1110 E					IM 2											
1111 F					LD R,A	LD A,R	RLD									
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	C	E	L	A	C	E	L	A								
	g (LO=8~F)															

D. Bus and Control Signal Condition in each Machine Cycle

* (ADDRESS) : invalid
Z (DATA) : high impedance.
** : added new instructions to Z80

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
ADD HL,ww	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂ ~MC₅	TiTiTiTi	*	Z	1	1	1	1	1	1	1
ADD IX,xx ADD IY,yy	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃ ~MC₆	TiTiTiTi	*	Z	1	1	1	1	1	1	1
ADC HL,ww SBC HL,ww	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃ ~MC₆	TiTiTiTi	*	Z	1	1	1	1	1	1	1
ADD A,g ADC A,g SUB g SBC A,g AND g OR g XOR g CP g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	*	Z	1	1	1	1	1	1	1
ADD A,m ADC A,m SUB m SBC A,m AND m OR m XOR m CP m	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	m	0	1	0	1	1	1	1
ADD A, (HL) ADC A, (HL) SUB (HL) SBC A, (HL) AND (HL) OR (HL) XOR (HL) CP (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
ADD A, (IX+d) ADD A, (IY+d) ADC A, (IX+d) ADC A, (IY+d) SUB (IX+d) SUB (IY+d) SBC A, (IX+d)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{MI}$	$\overline{HALT}$	ST
SBC A, (IY+d) AND (IX+d)	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
AND (IY+d) OR (IX+d) OR (IY+d) XOR (IX+d) XOR (IY+d)	MC₄ ~MC₅	TiTi	•	Z	1	1	1	1	1	1	1
CP (IX+d) CP (IY+d)	MC₆	T₁T₂T₃	IX+d IY+d	DATA	0	1	0	1	1	1	1
BIT b,g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
BIT b, (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
BIT b, (IX+d) BIT b, (IY+d)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	3rd op-code Address	3rd op-code	0	1	0	1	0	1	1
	MC₅	T₁T₂T₃	IX+d IY+d	DATA	0	1	0	1	1	1	1
CALL mn	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₄	Ti	•	Z	1	1	1	1	1	1	1
	MC₅	T₁T₂T₃	SP−1	PCH	1	0	0	1	1	1	1
	MC₆	T₁T₂T₃	SP−2	PCL	1	0	0	1	1	1	1
CALL f,mn (If condition is false)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
CALL f,mn (If condition is true)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₄	Ti	•	Z	1	1	1	1	1	1	1
	MC₅	T₁T₂T₃	SP−1	PCH	1	0	0	1	1	1	1
	MC₆	T₁T₂T₃	SP−2	PCL	1	0	0	1	1	1	1
CCF	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
CPI CPD	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄ ~MC₆	TiTiTi	•	Z	1	1	1	1	1	1	1
CPIR CPDR (If BC$_R$≠0 and Ar≠(HL)$_M$)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄ ~MC_d	TiTiTiTiTi	•	Z	1	1	1	1	1	1	1
CPIR CPDR (If BC$_R$=0 or Ar=(HL)$_M$)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄ ~MC₆	TiTiTi	•	Z	1	1	1	1	1	1	1
CPL	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
DAA	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
DI •1	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0

•1 Interrupt request is not sampled.

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
DJNZ j (If Br≠0)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	Ti *2	•	Z	1	1	1	1	1	1	1
	MC3	T1T2T3	1st operand Address	j-2	0	1	0	1	1	1	1
	MC4 ~MC5	TiTi	•	Z	1	1	1	1	1	1	1
DJNZ j (If Br=0)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	Ti *1	•	Z	1	1	1	1	1	1	1
	MC3	T1T2T3	1st operand Address	j-2	0	1	0	1	1	1	1
EI *3	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
EX DE, HL EXX	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
EX AF, AF'	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	Ti	•	Z	1	1	1	1	1	1	1
EX (SP), HL	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	SP	DATA	0	1	0	1	1	1	1
	MC3	T1T2T3	SP+1	DATA	0	1	0	1	1	1	1
	MC4	Ti	•	Z	1	1	1	1	1	1	1
	MC5	T1T2T3	SP+1	H	1	0	0	1	1	1	1
	MC6	T1T2T3	SP	L	1	0	0	1	1	1	1
EX (SP),IX EX (SP),IY	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	SP	DATA	0	1	0	1	1	1	1
	MC4	T1T2T3	SP+1	DATA	0	1	0	1	1	1	1
	MC5	Ti	•	Z	1	1	1	1	1	1	1

(to be continued)

*2 DMA, REFRESH, or BUS RELEASE cannot be executed after this state. (Request is ignored)
*3 Interrupt request is not sampled.

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
EX (SP), IX EX (SP), IY	MC₆	T₁T₂T₃	SP+1	IXH IYH	1	0	0	1	1	1	1
	MC₇	T₁T₂T₃	SP	IXL IYL	1	0	0	1	1	1	1
HALT	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	—	—	Next op-code Address	Next op-code	0	1	0	1	0	0	0
IM 0 IM 1 IM 2	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
INC g DEC g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
INC (HL) DEC (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	HL	DATA	1	0	0	1	1	1	1
INC (IX+d) INC (IY+d) DEC (IX+d) DEC (IY+d)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄ ~MC₅	TiTi	•	Z	1	1	1	1	1	1	1
	MC₆	T₁T₂T₃	IX+d IY+d	DATA	0	1	0	1	1	1	1
	MC₇	Ti	•	Z	1	1	1	1	1	1	1
	MC₈	T₁T₂T₃	IX+d IY+d	DATA	1	0	0	1	1	1	1
INC ww DEC ww	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
INC IX INC IY DEC IX DEC IY	MC₁	T₁T₂T₃	1st op-code Address	1st op-dode	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
IN A,(m)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	1st operand Address	m	0	1	0	1	1	1	1
	MC3	T1T2T3	m to A0~A7 A to A8~A15	DATA	0	1	1	0	1	1	1
IN g,(C)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	BC	DATA	0	1	1	0	1	1	1
INO g,(m)**	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	1st operand Address	m	0	1	0	1	1	1	1
	MC4	T1T2T3	m to A0~A7 00H to A8~A15	DATA	0	1	1	0	1	1	1
INI IND	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	BC	DATA	0	1	1	0	1	1	1
	MC4	T1T2T3	HL	DATA	1	0	0	1	1	1	1
INIR INDR (If Br≠0)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	BC	DATA	0	1	1	0	1	1	1
	MC4	T1T2T3	HL	DATA	1	0	0	1	1	1	1
	MC5 ~MC6	TiTi	*	Z	1	1	1	1	1	1	1
INIR INDR (If Br=0)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	BC	DATA	0	1	1	0	1	1	1
	MC4	T1T2T3	HL	DATA	1	0	0	1	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$ ·	$\overline{HALT}$	ST
JP mn	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
JP f,mn (If f is false)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
JP f,mn (If f is true)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
JP (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
JP (IX) JP (IY)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
JR j	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	j-2	0	1	0	1	1	1	1
	MC₃ ~MC₄	TiTi	•	Z	1	1	1	1	1	1	1
JR C,j JR NC,j JR Z,j JR NZ,j (If condition is false)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	j-2	0	1	0	1	1	1	1
JR C,j JR NC,j JR Z,j JR NZ,j (If condition is true)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	j-2	0	1	0	1	1	1	1
	MC₃ ~MC₄	TiTi	•	Z	1	1	1	1	1	1	1
LD g,g'	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
LD g,m	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	m	0	1	0	1	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
LD g, (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
LD g, (IX+d) LD g, (IY+d)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄ ~MC₅	TiTi	•	Z	1	1	1	1	1	1	1
	MC₆	T₁T₂T₃	IX+d IY+d	DATA	0	1	0	1	1	1	1
LD (HL),g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
	MC₃	T₁T₂T₃	HL	g	1	0	0	1	1	1	1
LD (IX+d),g LD (IY+d),g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄ ~MC₆	TiTiTi	•	Z	1	1	1	1	1	1	1
	MC₇	T₁T₂T₃	IX+d IY+d	g	1	0	0	1	1	1	1
LD (HL),m	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	m	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	HL	DATA	1	0	0	1	1	1	1
LD (IX+d),m LD (IY+d),m	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	IX+d IY+d	DATA	1	0	0	1	1	1	1
LD A, (BC) LD A, (DE)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
LD A, (BC) LD A, (DE)	MC2	T1T2T3	BC DE	DATA	0	1	0	1	1	1	1
LD A,(mn)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	1st operand Address	n	0	1	0	1	1	1	1
	MC3	T1T2T3	2nd operand Address	m	0	1	0	1	1	1	1
	MC4	T1T2T3	mn	DATA	0	1	0	1	1	1	1
LD (BC),A LD (DE),A	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	Ti	•	Z	1	1	1	1	1	1	1
	MC3	T1T2T3	BC DE	A	1	0	0	1	1	1	1
LD (mn),A	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	1st operand Address	n	0	1	0	1	1	1	1
	MC3	T1T2T3	2nd operand Address	m	0	1	0	1	1	1	1
	MC4	Ti	•	Z	1	1	1	1	1	1	1
	MC5	T1T2T3	mn	A	1	0	0	1	1	1	1

LD A,I ⎱ *4
LD A,R ⎰
LD I,A
LD R,A
	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0

	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
LD ww, mn	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	1st operand Address	n	0	1	0	1	1	1	1
	MC3	T1T2T3	2nd operand Address	m	0	1	0	1	1	1	1
LD IX,mn LD IY,mn	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	1st operand Address	n	0	1	0	1	1	1	1
	MC4	T1T2T3	2nd operand Address	m	0	1	0	1	1	1	1
LD HL, (mn)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	1st operand Address	n	0	1	0	1	1	1	1

*4 In the case of R1 and Z MASK, interrupt request is not sampled.

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	R̄D̄	W̄R̄	M̄R̄ĒQ̄	ĪŌR̄Q̄	M̄1̄	H̄ĀL̄T̄	ST
LD HL, (mn)	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	mn	DATA	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	mn+1	DATA	0	1	0	1	1	1	1
LD ww,(mn)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	mn	DATA	0	1	0	1	1	1	1
	MC₆	T₁T₂T₃	mn+1	DATA	0	1	0	1	1	1	1
LD IX,(mn) LD IY,(mn)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	mn	DATA	0	1	0	1	1	1	1
	MC₆	T₁T₂T₃	mn+1	DATA	0	1	0	1	1	1	1
LD (mn),HL	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₄	Ti	·	Z	1	1	1	1	1	1	1
	MC₅	T₁T₂T₃	mn	L	1	0	0	1	1	1	1
	MC₅	T₁T₂T₃	mn+1	H	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
LD (mn),ww	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₅	Ti	•	Z	1	1	1	1	1	1	1
	MC₆	T₁T₂T₃	mn	wwL	1	0	0	1	1	1	1
	MC₇	T₁T₂T₃	mn+1	wwH	1	0	0	1	1	1	1
LD (mn),IX LD (mn),IY	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	n	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	2nd operand Address	m	0	1	0	1	1	1	1
	MC₅	Ti	•	Z	1	1	1	1	1	1	1
	MC₆	T₁T₂T₃	mn	IXL IYL	1	0	0	1	1	1	1
	MC₇	T₁T₂T₃	mn+1	IXH IYH	1	0	0	1	1	1	1
LD SP, HL	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
LD SP,IX LD SP,IY	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
LDI LDD	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	DE	DATA	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
LDIR LDDR (If $BC_R \neq 0$)	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	$T_1T_2T_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	$T_1T_2T_3$	HL	DATA	0	1	0	1	1	1	1
	MC4	$T_1T_2T_3$	DE	DATA	1	0	0	1	1	1	1
	MC5 ~MC6	TiTi	•	Z	1	1	1	1	1	1	1
LDIR LDDR (If $BC_R = 0$)	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	$T_1T_2T_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	$T_1T_2T_3$	HL	DATA	0	1	0	1	1	1	1
	MC4	$T_1T_2T_3$	DE	DATA	1	0	0	1	1	1	1
MILT ww**	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	$T_1T_2T_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3 ~MC13	TiTiTiTi TiTiTiTi TiTiTi	•	Z	1	1	1	1	1	1	1
NEG	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	$T_1T_2T_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
NOP	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
OUT (m),A	MC1	$T_1T_2T_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	$T_1T_2T_3$	1st operand Address	m	0	1	0	1	1	1	1
	MC3	Ti	•	Z	1	1	1	1	1	1	1
	MC4	$T_1T_2T_3$	m to A_0~A_7 A to A_8~A_{15}	A	1	0	1	0	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
OUT (C),g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	BC	g	1	0	1	0	1	1	1
OUT0 (m),g**	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	m	0	1	0	1	1	1	1
	MC₄	Ti	•	Z	1	1	1	1	1	1	1
	MC₅	T₁T₂T₃	m to A₀～A₇ 00H to A₈～A₁₅	g	1	0	1	0	1	1	1
OTIM** OTDM**	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	C to A₀～A₇ 00H to A₈～A₁₅	DATA	1	0	1	0	1	1	1
	MC₆	Ti	•	Z	1	1	1	1	1	1	1
OTIMR** OTDMR** (If Br≠0)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	C to A₀～A₇ 00H to A₈～A₁₅	DATA	1	0	1	0	1	1	1
	MC₆ ～MC₁	TiTiTi	•	Z	1	1	1	1	1	1	1

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
OTIMR** OTDMR** (If Br=0)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₅	T₁T₂T₃	C to A₀~A₇ 00H to A₄~A₁₅	DATA	1	0	1	0	1	1	1
	MC₆	Ti	•	Z	1	1	1	1	1	1	1
OUTI OUTD	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	BC	DATA	1	0	1	0	1	1	1
OTIR OTDR (If Br≠0)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	BC	DATA	1	0	1	0	1	1	1
	MC₅ ~MC₆	TiTi	•	Z	1	1	1	1	1	1	1
OTIR OTDR (If Br=0)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	BC	DATA	1	0	1	0	1	1	1
POP zz	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	SP	DATA	0	1	0	1	1	1	1
	MC₃	T₁T₂T₃	SP+1	DATA	0	1	0	1	1	1	1
POP IX POP IY	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
POP IX POP IY	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	T$_1$T$_2$T$_3$	SP	DATA	0	1	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	SP+1	DATA	0	1	0	1	1	1	1
PUSH zz	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$ ~MC$_3$	TiTi	•	Z	1	1	1	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	SP−1	zzH	1	0	0	1	1	1	1
	MC$_5$	T$_1$T$_2$T$_3$	SP−2	zzL	1	0	0	1	1	1	1
PUSH IX PUSH IY	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$ ~MC$_4$	TiTi	•	Z	1	1	1	1	1	1	1
	MC$_5$	T$_1$T$_2$T$_3$	SP−1	IXH IYH	1	0	0	1	1	1	1
	MC$_6$	T$_1$T$_2$T$_3$	SP−2	IXL IYL	1	0	0	1	1	1	1
RET	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	SP	DATA	0	1	0	1	1	1	1
	MC$_3$	T$_1$T$_2$T$_3$	SP+1	DATA	0	1	0	1	1	1	1
RET f (If condition is false)	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$ ~MC$_3$	TiTi	•	Z	1	1	1	1	1	1	1
RET f (If condition is true)	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	Ti	•	Z	1	1	1	1	1	1	1
	MC$_3$	T$_1$T$_2$T$_3$	SP	DATA	0	1	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	SP+1	DATA	0	1	0	1	1	1	1
RETI (R0, R1) RETN	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	T$_1$T$_2$T$_3$	SP	DATA	0	1	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	SP+1	DATA	0	1	0	1	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
RETI (Z)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0·5 / 1	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0·5 / 1	1	1
	MC3 ~MC5	TiTiTi	•	Z	1	1	1	1	1·5 / 1	1	1
	MC6	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0·5 / 0	1	1
	MC7	Ti	•	Z	1	1	1	1	1·5 / 1	1	1
	MC8	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0·5 / 0	1	1
	MC9	T1T2T3	SP	data	0	1	0	1	1·5 / 1	1	1
	MC10	T1T2T3	SP+1	data	0	1	0	1	1·5 / 1	1	1
RLCA RLA RRCA RRA	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
RLC g RL g RRC g RR g SLA g SRA g SRL g	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	Ti	•	Z	1	1	1	1	1	1	1
RLC (HL) RL (HL) RRC (HL) RR (HL) SLA (HL) SRA (HL) SRL (HL)	MC1	T1T2T3	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC2	T1T2T3	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC3	T1T2T3	HL	DATA	0	1	0	1	1	1	1
	MC4	Ti	•	Z	1	1	1	1	1	1	1
	MC5	T1T2T3	HL	DATA	1	0	0	1	1	1	1

·5 The upper and lower data show the state of $\overline{LIR}$ when $\overline{IOC}=1$ and $\overline{IOC}=0$ respectively.

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
RLC (IX+d) RLC (IY+d) RL (IX+d) RL (IY+d) RRC (IX+d) RRC (IY+d) RR (IX+d) RR (IY+d) SLA (IX+d) SLA (IY+d) SRA (IX+d) SRA (IY+d) SRL (IX+d) SRL (IY+d)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	1st operand Address	d	0	1	0	1	1	1	1
	MC₄	T₁T₂T₃	3rd op-code Address	3rd op-code	0	1	0	1	0	1	1
	MC₅	T₁T₂T₃	IX+d IY+d	DATA	0	1	0	1	1	1	1
	MC₆	Ti	•	Z	1	1	1	1	1	1	1
	MC₇	T₁T₂T₃	IX+d IY+d	DATA	1	0	0	1	1	1	1
RLD RRD	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄ ~MC₇	TiTiTiTi	•	Z	1	1	1	1	1	1	1
	MC₈	T₁T₂T₃	HL	DATA	1	0	0	1	1	1	1
RST v	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂ ~MC₃	TiTi	•	Z	1	1	1	1	1	1	1
	MC₄	T₁T₂T₃	SP−1	PCH	1	0	0	1	1	1	1
	MC₅	T₁T₂T₃	SP−2	PCL	1	0	0	1	1	1	1
SCF	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
SET b,g RES b,g	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂T₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	Ti	•	Z	1	1	1	1	1	1	1
SET b, (HL) RES b, (HL)	MC₁	T₁T₂T₃	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC₂	T₁T₂ ₃	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC₃	T₁T₂T₃	HL	DATA	0	1	0	1	1	1	1
	MC₄	Ti	•	Z	1	1	1	1	1	1	1
	MC₅	T₁T₂T₃	HL	DATA	1	0	0	1	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
SET b, (IX + d) SET b, (IY + d) RES b, (IX + d) RES b, (IY + d)	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	T$_1$T$_2$T$_3$	1st operand Address	d	0	1	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	3rd op-code Address	3rd op-code	0	1	0	1	0	1	1
	MC$_5$	T$_1$T$_2$T$_3$	IX + d IY + d	DATA	0	1	0	1	1	1	1
	MC$_6$	Ti	•	Z	1	1	1	1	1	1	1
	MC$_7$	T$_1$T$_2$T$_3$	IX + d IY + d	DATA	1	0	0	1	1	1	1
SLP**	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	—	—	7FFFFH	Z	1	1	1	1	1	0	1
TSTIO m**	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	T$_1$T$_2$T$_3$	1st operand Address	m	0	1	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	C to A$_0$~A$_7$ 00H to A$_8$~A$_{15}$	DATA	0	1	1	0	1	1	1
TST g**	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	Ti	•	Z	1	1	1	1	1	1	1
TST m**	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	T$_1$T$_2$T$_3$	1st operand Address	m	0	1	0	1	1	1	1
TST (HL)**	MC$_1$	T$_1$T$_2$T$_3$	1st op-code Address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	T$_1$T$_2$T$_3$	2nd op-code Address	2nd op-code	0	1	0	1	0	1	1
	MC$_3$	Ti	•	Z	1	1	1	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	HL	DATA	0	1	0	1	1	1	1

INTERRUPT

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
$\overline{NMI}$	MC₁	$T_1T_2T_3$	Next op-code Address (PC)		0	1	0	1	0	1	0
	MC₂ ~MC₃	TiTi	•	Z	1	1	1	1	1	1	1
	MC₄	$T_1T_2T_3$	SP−1	PCH	1	0	0	1	1	1	1
	MC₅	$T_1T_2T_3$	SP−2	PCL	1	0	0	1	1	1	1
$\overline{INT_0}$ MODE 0 (RST INSERTED)	MC₁	$T_1T_2T_W$ T_WT_3	Next op-code Address (PC)	1st op-code	1	1	1	0	0	1	0
	MC₂ ~MC₃	TiTi	•	Z	1	1	1	1	1	1	1
	MC₄	$T_1T_2T_3$	SP−1	PCH	1	0	0	1	1	1	1
	MC₅	$T_1T_2T_3$	SP−2	PCL	1	0	0	1	1	1	1
$\overline{INT_0}$ MODE 0 (CALL INSERTED)	MC₁	$T_1T_2T_W$ T_WT_3	Next op-code Address (PC)	1st op-code	1	1	1	0	0	1	0
	MC₂	$T_1T_2T_3$	PC	n	0	1	0	1	1	1	1
	MC₃	$T_1T_2T_3$	PC+1	m	0	1	0	1	1	1	1
	MC₄	Ti	•	Z	1	1	1	1	1	1	1
	MC₅	$T_1T_2T_3$	SP−1	PC+2(H)	1	0	0	1	1	1	1
	MC₆	$T_1T_2T_3$	SP−2	PC+2(L)	1	0	0	1	1	1	1
$\overline{INT_0}$ MODE 1	MC₁	$T_1T_2T_W$ T_WT_3	Next op-code Address (PC)		1	1	1	0	0	1	0
	MC₂	$T_1T_2T_3$	SP−1	PCH	1	0	0	1	1	1	1
	MC₃	$T_1T_2T_3$	SP−2	PCL	1	0	0	1	1	1	1
$\overline{INT_0}$ MODE 2	MC₁	$T_1T_2T_W$ T_WT_3	Next op-code Address (PC)	Vector	1	1	1	0	0	1	0
	MC₂	Ti	•	Z	1	1	1	1	1	1	1
	MC₃	$T_1T_2T_3$	SP−1	PCH	1	0	0	1	1	1	1
	MC₄	$T_1T_2T_3$	SP−2	PCL	1	0	0	1	1	1	1
	MC₅	$T_1T_2T_3$	I, Vector	DATA	0	1	0	1	1	1	1
	MC₆	$T_1T_2T_3$	I, Vector+1	DATA	0	1	0	1	1	1	1

(to be continued)

Instruction	Machine Cycle	States	ADDRESS	DATA	$\overline{RD}$	$\overline{WR}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{M1}$	$\overline{HALT}$	ST
$\overline{INT_1}$ $\overline{INT_2}$ Internal Interrupts	MC$_1$	T$_1$T$_2$T$_W$ T$_W$T$_3$	Next op-code Address (PC)		1	1	1	1	1	1	0
	MC$_2$	Ti	•	Z	1	1	1	1	1	1	1
	MC$_3$	T$_1$T$_2$T$_3$	SP−1	PCH	1	0	0	1	1	1	1
	MC$_4$	T$_1$T$_2$T$_3$	SP−2	PCL	1	0	0	1	1	1	1
	MC$_5$	T$_1$T$_2$T$_3$	I, Vector	DATA	0	1	0	1	1	1	1
	MC$_6$	T$_1$T$_2$T$_3$	I, Vector+1	DATA	0	1	0	1	1	1	1

Request \ Current Status	Normal Operation (CPU mode) (IOSTOP mode)	WAIT State	Refresh Cycle	Interrupt Acknowledge Cycle	DMA Cycle	BUS RELEASE mode	SLEEP mode	SYSTEM STOP mode
$\overline{\text{WAIT}}$	Acceptable	Acceptable	Not acceptable	Acceptable	Acceptable	Not acceptable	Not acceptable	Not acceptable
Refresh Request (Request of Refresh by the on-chip Refresh Controller)	Refresh cycle begins at the end of MC.	Not acceptable	Not acceptable	Refresh cycle begins at the end of MC.	Refresh cycle begins at the end of MC.	Not acceptable	Not acceptable	Not acceptable
$\overline{\text{DREQ}}_0$ $\overline{\text{DREQ}}_1$	DMA cycle begins at the end of MC.	DMA cycle begins at the end of MC.	Acceptable * Refresh cycle precedes. DMA cycle begins at the end of one MC.	Acceptable DMA cycle begins at the end of MC.	Acceptable Refer to "2.9 DMA Controller" for details.	Acceptable * After BUS RELEASE cycle, DMA cycle begins at the end of one MC.	Not acceptable	Not acceptable
$\overline{\text{BUSREQ}}$	Bus is released at the end of MC.	Not acceptable	Not acceptable	Bus is released at the end of MC.	Bus is released at the end of MC.	Continue BUS RELEASE mode.	Acceptable	Acceptable
Interrupt — $\overline{\text{INT}}_0$, $\overline{\text{INT}}_1$, $\overline{\text{INT}}_2$	Accepted after executing the current instruction.	Accepted after executing the current instruction	Not acceptable	Not acceptable	Not acceptable	Not acceptable	Acceptable Return from SLEEP mode to normal operation.	Acceptable Return from SYSTEM STOP mode to normal operation.
Interrupt — Internal I/O Interrupt	↑	↑	↑	↑	↑	↑	↑	Not acceptable
Interrupt — $\overline{\text{NMI}}$	↑	↑	↑	Not acceptable Interrupt acknowledge cycle precedes. NMI is accepted after executing the next instruction.	Acceptable DMA cycle stops.	↑	↑	Acceptable Return from SYSTEM STOP mode to normal operation.

NOTE) * : not acceptable when DMA Request is in level sense.
↑ : same as the above
MC : Machine Cycle

E-2. Request Priority

The Z80180 has the following three types of requests.

Type 1.
    To be accepted in specified state ....WAIT

Type 2.
    To be accepted in each machine cycle....................Refresh Req.
        DMA Req.
        Bus Req.

Type 3.
    To be accepted in each instruction .. ..Interrrupt Req.

Type 1, Type 2, and Type 3 requests priority is shown
as follows.
    highest priority Type 1 > Type 2> Type 3 lowest priority
Each request priority in Type 2 is shown as follows.
    highest priority Bus Req. > Refresh Req. > DMA Req.
    lowest priority

(NOTE) If Bus Req. and Refresh Req. occurs simul—
taneously, Bus Req. is accepted but Refresh Req. is
cleared.

E-3. Operation Mode Transition
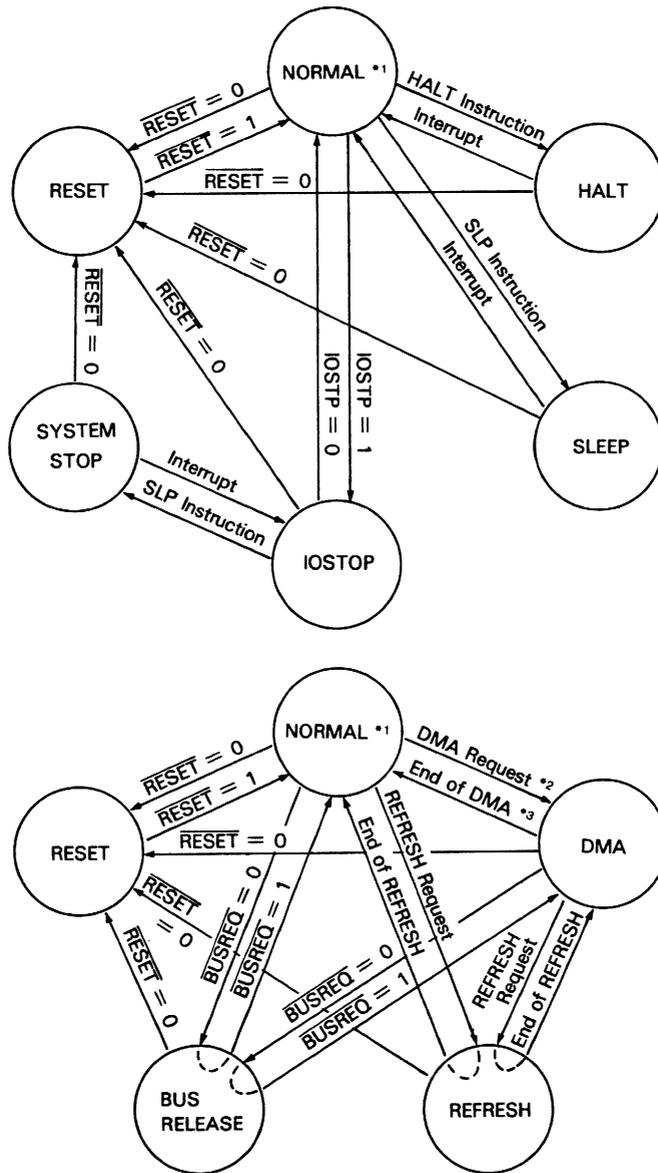
NOTE) *1 NORMAL: CPU executes instructions normally in NORMAL mode.
*2 DMA request: DMA is requested in the following cases.
(1) $\overline{DREQ_0}$, $\overline{DREQ_1}$ = 0 (memory to/from (memory mapped) I/O DMA transfer)
(2) DE0 = 1 (memory to/from memory DMA transfer)
*3 DMA end: DMA ends in the following cases.
(1) $\overline{DREQ_0}$, $\overline{DREQ_1}$ = 1 (memory to/from (memory mapped) I/O DMA transfer)
(2) BCR0, BCR1 = 0000H (all DMA transfers)
(3) $\overline{NMI}$ = 0 (all DMA transfers)

Other operation mode transitions

The following operation mode transitions are also possible.

1. HALT ⇄ ⎰DMA   ⎱
            ⎨REFRESH   ⎬
            ⎱BUS RELEASE⎰

    IOSTOP ⇄ ⎰DMA   ⎱
               ⎨REFRESH   ⎬
               ⎱BUS RELEASE⎰

2. SLEEP ⇄ BUS RELEASE

    SYSTEM STOP ⇄ BUS RELEASE
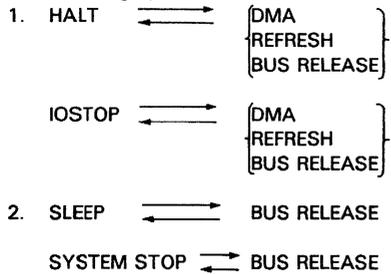
F-1. Status Signals

The following table shows pin outputs in each operating mode.

Mode		$\overline{M1}$	$\overline{MREQ}$	$\overline{IORQ}$	$\overline{RD}$	$\overline{WR}$	$\overline{REF}$	$\overline{HALT}$	$\overline{BUSACK}$	ST	Address BUS	Data BUS
CPU operation	Op-code Fetch (1st op-code)	0	0	1	0	1	1	1	1	0	A	IN
	Op-code Fetch (except 1st op-code)	0	0	1	0	1	1	1	1	1	A	IN
	Memory Read	1	0	1	0	1	1	1	1	1	A	IN
	Memory Write	1	0	1	1	0	1	1	1	1	A	OUT
	I/O Read	1	1	0	0	1	1	1	1	1	A	IN
	I/O Write	1	1	0	1	0	1	1	1	1	A	OUT
	Internal Operation	1	1	1	1	1	1	1	1	1	A	IN
Refresh		1	0	1	1	1	0	1	1	•	A	IN
Interrupt Acknowledge Cycle (1st machine cycle)	$\overline{NMI}$	0	0	1	0	1	1	1	1	0	A	IN
	$\overline{INT_0}$	0	1	0	1	1	1	1	1	0	A	IN
	$\overline{INT_1}$, $\overline{INT_2}$ & Internal Interrupts	1	1	1	1	1	1	1	1	0	A	IN
BUS RELEASE		1	Z	Z	Z	Z	1	1	0	•	Z	IN
HALT		0	0	1	0	1	1	0	1	0	A	IN
SLEEP		1	1	1	1	1	1	0	1	1	1	IN
Internal DMA	Memory Read	1	0	1	0	1	1	•	1	0	A	IN
	Memory Write	1	0	1	1	0	1	•	1	0	A	OUT
	I/O Read	1	1	0	0	1	1	•	1	0	A	IN
	I/O Write	1	1	0	1	0	1	•	1	0	A	OUT
RESET		1	1	1	1	1	1	1	1	1	Z	IN

NOTE) 1 : HIGH
     0 : LOW
     A : Programmable
     Z : High Impedance
     IN : Input
     OUT : Output
     • : Invalid

F-2. Pin Status during RESET and Low Power Operation Modes

Symbol	Pin function	Pin status in each operation mode			
		RESET	SLEEP	IOSTOP	SYSTEM STOP
$\overline{\text{WAIT}}$	–	IN (N)	IN (N)	IN (A)	IN (N)
$\overline{\text{BUSACK}}$	–	1	OUT	OUT	OUT
$\overline{\text{BUSREQ}}$	–	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{\text{RESET}}$	–	0	IN (A)	IN (A)	IN (A)
$\overline{\text{NMI}}$	–	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{\text{INT}_0}$	–	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{\text{INT}_1}$	–	IN (N)	IN (A)	IN (A)	IN (A)
$\overline{\text{INT}_2}$	–	IN (N)	IN (A)	IN (A)	IN (A)
ST	–	1	1	OUT	1
$A_0 \sim A_{17}, A_{19}$	–	Z	1	A	1
A_1 / TOUT	A_{11}	Z	1	A	1
	TOUT	Z	OUT	H	H
$D_0 \sim D_7$	–	Z	Z	A	Z
$\overline{\text{RTS}_0}$	–	1	H	OUT	H
$\overline{\text{CTS}_0}$	–	IN (N)	IN (A)	IN (N)	IN (N)
$\overline{\text{DCD}_0}$	–	IN (N)	IN (A)	IN (N)	IN (N)
TXA_0	–	1	OUT	H	H
RXA_0	–	IN (N)	IN (A)	IN (N)	IN (N)
$\text{CKA}_0 / \overline{\text{DREQ}_0}$	CKA_0 (internal clock mode)	Z	OUT	Z	Z
	CKA_0 (external clock mode)	Z	IN (A)	IN (N)	IN (N)
	$\overline{\text{DREQ}_0}$	Z	IN (N)	IN (A)	IN (N)
TXA_1	–	1	OUT	H	H
RXA_1	–	IN (N)	IN (A)	IN (N)	IN (N)
$\text{CKA}_1 / \overline{\text{TEND}_0}$	CKA_1 (internal clock mode)	Z	OUT	Z	Z
	CKA_1 (external clock mode)	Z	IN (A)	IN (N)	IN (N)
	$\overline{\text{TEND}_0}$	Z	1	OUT	1
TXS	–	1	OUT	H	H
$\text{RXS} / \overline{\text{CTS}_1}$	RXS	IN (N)	IN (A)	IN (N)	IN (N)
	$\overline{\text{CTS}_1}$	IN (N)	IN (A)	IN (N)	IN (N)
CKS	CKS (internal clock mode)	Z	OUT	1	1
	CKS (external clock mode)	Z	IN (A)	Z	Z
$\overline{\text{DREQ}_1}$	–	IN (N)	IN (N)	IN (A)	IN (N)
$\overline{\text{TEND}_1}$	–	1	1	OUT	1
$\overline{\text{HALT}}$	–	1	0	OUT	0
$\overline{\text{RFSH}}$	–	1	1	OUT	1
$\overline{\text{IORQ}}$	–	1	1	OUT	1
$\overline{\text{MREQ}}$	–	1	1	OUT	1
E	–	0	E clock output	—	—
$\overline{\text{M1}}$	–	1	1	OUT	1
$\overline{\text{WR}}$	–	1	1	OUT	1
$\overline{\text{RD}}$	–	1	1	OUT	1
ϕ	–	ϕ clock output	—	—	—

1: HIGH　　0: LOW　　A: Programmable　　Z: High Impedance
IN (A): Input (Active)　　IN (N): Input (Not active)　　OUT: Output
H: Holds the previous state
—: same as the left

G. Internal I/O Registers

By programming IOA7 and IOA6 in the I/O control register, internal I/O register addresses are relocatable within ranges from 0000H to 00FFH in the I/O address space.

REGISTER	MNEMONICS	ADDRESS	REMARKS
ASCI Control Register A Channel 0 : CNTLA0		0 0	

CNTLA0 (Address 0 0)

bit	MPE	RE	TE	RTS0̄	MPBR/EFR	MOD2	MOD1	MOD0
during RESET	0	0	0	1	invalid	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- MODO/MOD2, MOD1, MOD0 — MODE Selection
- MPBR/EFR — Multi Processor Bit Receive/Error Flag Reset
- RTS0̄ — Request To Send
- TE — Transmit Enable
- RE — Receive Enable
- MPE — Multi Processor Enable

CNTLA1 (Address 0 1) — ASCI Control Register A Channel 1

bit	MPE	RE	TE	CKA1D	MPBR/EFR	MOD2	MOD1	MOD0
during RESET	0	0	0	1	invalid	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- MOD2, MOD1, MOD0 — MODE Selection
- MPBR/EFR — Multi Processor Bit Receive/Error Flag Reset
- CKA1D — CKA1 Disable
- TE — Transmit Enable
- RE — Receive Enable
- MPE — Multi Processor Enable

MOD2, 1, 0
```
0 0 0   Start + 7 bit Data + 1 Stop
0 0 1   Start + 7 bit Data + 2 Stop
0 1 0   Start + 7 bit Data + Parity + 1 Stop
0 1 1   Start + 7 bit Data + Parity + 2 Stop
1 0 0   Start + 8 bit Data + 1 Stop
1 0 1   Start + 8 bit Data + 2 Stop
1 1 0   Start + 8 bit Data + Parity + 1 Stop
1 1 1   Start + 8 bit Data + Parity + 2 Stop
```

CNTLB0 (Address 0 2) — ASCI Control Register B Channel 0

bit	MPBT	MP	CTS̄/PS	PEO	DR	SS2	SS1	SS0
during RESET	invalid	0	•	0	0	1	1	1
R/W	R/W	R/W	R/W	R/W.	R/W	R/W	R/W	R/W

- SS2, SS1, SS0 — Clock Source and Speed Select
- DR — Divide Ratio
- PEO — Parity Even or Odd
- CTS̄/PS — Clear To Send/Prescale
- MP — Multi Processor
- MPBT — Multi Processor Bit Transmit

- CTS̄ : Depending on the condition of CTS̄ Pin.
- PS : Cleared to 0.

REGISTER	MNEMONICS	ADDRESS	REMARKS
ASCI Control Register B Channel 1 : CNTLB1		0 3	

ASCI Control Register B Channel 1 : CNTLB1 (Address 0 3)

bit	MPBT	MP	$\overline{\text{CTS}}$/PS	PEO	DR	SS2	SS1	SS0
during RESET	invalid	0	0	0	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- SS0, SS1, SS2: Clock Source and Speed Select
- DR: Divide Ratio
- PEO: Parity Even or Odd
- $\overline{\text{CTS}}$/PS: Clear To Send/Prescale
- MP: Multi Processor
- MPBT: Multi Processor Bit Transmit

General divide ratio	PS=0 (divide ratio=10)		PS=1 (divide ratio=30)	
SS2,1,0	DR=0 (×16)	DR=1 (×64)	DR=0 (×16)	DR=1 (×64)
0 0 0	$\phi\div$ 160	$\phi\div$ 640	$\phi\div$ 480	$\phi\div$ 1920
0 0 1	÷ 320	÷ 1280	÷ 960	÷ 3840
0 1 0	÷ 640	÷ 2560	÷ 1920	÷ 7680
0 1 1	÷ 1280	÷ 5120	÷ 3840	÷ 15360
1 0 0	÷ 2560	÷ 10240	÷ 7680	÷ 30720
1 0 1	÷ 5120	÷ 20480	÷ 15360	÷ 61440
1 1 0	÷ 10240	÷ 40960	÷ 30720	÷ 122880
1 1 1	External clock (frequency $< \phi \div 40$)			

ASCI Status Register Channel 0 : STAT0 (Address 0 4)

bit	RDRF	OVRN	PE	FE	RIE	$\overline{\text{DCD0}}$	TDRE	TIE
during RESET	0	0	0	0	0	•	••	0
R/W	R	R	R	R	R/W	R	R	R/W

- TIE: Transmit Interrupt Enable
- TDRE: Transmit Data Register Empty
- $\overline{\text{DCD0}}$: Data Carrier Detect
- RIE: Receive Interrupt Enable
- FE: Framing Error
- PE: Parity Error
- OVRN: Over Run Error
- RDRF: Receive Data Register Full

* $\overline{\text{DCD0}}$: Depending on the condition of $\overline{\text{DCD0}}$ Pin.

••	$\overline{\text{CTS0}}$ Pin	TDRE
	L	1
	H	0

ASCI Status Register Channel 1 : STAT1 (Address 0 5)

bit	RDRF	OVRN	PE	FE	RIE	CTS1E	TDRE	TIE
during RESET	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R/W	R/W	R	R/W

- TIE: Transmit Interrupt Enable
- TDRE: Transmit Data Register Empty
- CTS1E: CTS1 Enable
- RIE: Receive Interrupt Enable
- FE: Framing Error
- PE: Parity Error
- OVRN: Over Run Error
- RDRF: Receive Data Register Full

(to be continued)

REGISTER	MNEMONICS	ADDRESS	REMARKS
ASCI Transmit Data Register Channel 0	: TDR0	0 6	
ASCI Transmit Data Register Channel 1	: TDR1	0 7	
ASCI Receive Data Register Channel 0	: TSR0	0 8	
ASCI Receive Data Register Channel 1	: TSR1	0 9	
CSI/O Control Register	: CNTR	0 A	
CSI/O Transmit/Receive Data Register	: TRDR	0 B	
Timer Data Register Channel 0L	: TMDR0L	0 C	
Timer Data Register Channel 0H	: TMDR0H	0 D	
Timer Reload Register Channel 0L	: RLDR0L	0 E	
Timer Reload Register Channel 0H	: RLDR0H	0 F	
Timer Control Register	: TCR	1 0	

CSI/O Control Register (CNTR) — address 0 A:

bit	EF	EIE	RE	TE	—	SS2	SS1	SS0
during RESET	0	0	0	0	1	1	1	1
R/W	R	R/W	R/W	R/W		R/W	R/W	R/W

- Speed Select
- Transmit Enable
- Receive Enable
- End Interrupt Enable
- End Flag

SS2,1,0	Baud Rate	SS2,1,0	Baud Rate
000	$\phi \div 20$	100	$\phi \div 320$
001	$\div 40$	101	$\div 640$
010	$\div 80$	110	$\div 1280$
011	$\div 160$	111	External (frequency $< \div 20$)

Timer Control Register (TCR) — address 1 0:

bit	TIF1	TIF0	TIE1	TIE0	TOC1	TOC0	TDE1	TDE0
during RESET	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Timer Down Count Enable 1,0
- Timer Output Control 1,0
- Timer Interrupt Enable 1,0
- Timer Interrupt Flag 1,0

TOC1,0	A_{18}/TOUT
00	Inhibited
01	Toggle
10	0
11	1

REGISTER	MNEMONICS	ADDRESS	REMARKS
Timer Data Register Channel 1L : TMDR1L		1 4	
Timer Data Register Channel 1H : TMDR1H		1 5	
Timer Reload Register Channel 1L : RLDR1L		1 6	
Timer Reload Register Channel 1H : RLDR1H		1 7	
Free Running Counter : FRC		1 8	read only
DMA Source Address Register Channel 0L : SAR0L		2 0	
DMA Source Address Register Channel 0H : SAR0H		2 1	
DMA Source Address Register Channel 0B : SAR0B		2 2	Bits 0-2 (3) are used for SAR0B.
DMA Destination Address Register Channel 0L : DAR0L		2 3	
DMA Destination Address Register Channel 0H : DAR0H		2 4	
DMA Destination Address Register Channel 0B : DAR0B		2 5	Bits 0-2 (3) are used for DAR0B.
DMA Byte Count Register Channel 0L : BCR0L		2 6	
DMA Byte Count Register Channel 0H : BCR0H		2 7	
DMA Memory Address Register Channel 1L : MAR1L		2 8	
DMA Memory Address Register Channel 1H : MAR1H		2 9	
DMA Memory Address Register Channel 1B : MAR1B		2 A	Bits 0-2 (3) are used for MAR1B.
DMA I/O Address Register Channel 1L : IAR1L		2 B	
DMA I/O Address Register Channel 1H : IAR1H		2 C	

Remarks for address 2 2 (SAR0B):

A_{19}^*, A_{18},	A_{17},	A_{16}		DMA Transfer Request
X	X	0	0	$\overline{DREQ_0}$ (external)
X	X	0	1	RDR0 (ASCI0)
X	X	1	0	RDR1 (ASCI1)
X	X	1	1	Not Used

Remarks for address 2 5 (DAR0B):

A_{19}^*, A_{18},	A_{17},	A_{16}		DMA Transfer Request
X	X	0	0	$\overline{DREQ_0}$ (external)
X	X	0	1	TDR0 (ASCI0)
X	X	1	0	TDR1 (ASCI1)
X	X	1	1	Not Used

(to be continued)

* In the R1 and Z Mask, these DMAC registers are expanded from 4 bits to 3 bits in the package version of CP-68 and FP-80.

REGISTER	MNEMONICS	ADDRESS	REMARKS
DMA Byte Count Register Channel 1L	: BCR1L	2 E	
DMA Byte Count Register Channel 1H	: BCR1H	2 F	
DMA Status Register	: DSTAT	3 0	
DMA Mode Register	: DMODE	3 1	

DSTAT (DMA Status Register):

bit	DE1	DE0	$\overline{DWE1}$	$\overline{DWE0}$	DIE1	DIE0	—	DME
during RESET	0	0	1	1	0	0	1	0
R/W	R/W	R/W	W	W	R/W	R/W		R

- DME — DMA Master Enable
- DIE1,0 — DMA Interrupt Enable 1,0
- DWE1,0 — DMA Enable Bit Write Enable 1,0
- DE1,0 — DMA Enable ch 1,0

DMODE (DMA Mode Register):

bit	—	—	DM1	DM0	SM1	SM0	MMOD	—
during RESET	1	1	0	0	0	0	0	1
R/W			R/W	R/W	R/W	R/W	R/W	

- MMOD — Memory MODE Select
- SM1,0 — Ch 0 Source Mode 1,0
- DM1,0 — Ch 0 Destination Mode 1,0

DM1, 0	Destination	Address
0 0	M	DAR0+1
0 1	M	DAR0−1
1 0	M	DAR0 fixed
1 1	I/O	DAR0 fixed

SM1, 0	Source	Address
0 0	M	SAR0+1
0 1	M	SAR0−1
1 0	M	SAR0 fixed
1 1	I/O	SAR0 fixed

MMOD	Mode
0	Cycle Steal Mode
1	Burst Mode

(to be continued)

REGISTER	MNEMONICS	ADDRESS	REMARKS
DMA/WAIT Control Register : DCNTL		3 2	

bit	MWI1	MWI0	IWI1	IWI0	DMS1	DMS0	DIM1	DIM0
during RESET	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMA Ch 1 I/O Memory Mode Select
- $\overline{DREQi}$ Select, i = 1,0
- I/O Wait Insertion
- Memory Wait Insertion

MWI1,0	The number of wait states	IWI1,0	The number of wait states
0 0	0	0 0	0
0 1	1	0 1	2
1 0	2	1 0	3
1 1	3	1 1	4

DMSi	Sense
1	Edge sense
0	Level sense

DIM1,0	Transfer Mode	Address Increment/Decrement	
0 0	M→I/O	MAR1+1	IAR1 fixed
0 1	M→I/O	MAR1−1	IAR1 fixed
1 0	I/O→M	IAR1 fixed	MAR1+1
1 1	I/O→M	IAR1 fixed	MAR1−1

REGISTER	MNEMONICS	ADDRESS
Interrupt Vector Low Register : IL		3 3

bit	IL7	IL6	IL5	—	—	—	—	—
during RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W					

- Interrupt Vector Low

REGISTER	MNEMONICS	ADDRESS
INT/TRAP Control Register : ITC		3 4

bit	TRAP	UFO	—	—	—	ITE2	ITE1	ITE0
during RESET	0	0	1	1	1	0	0	1
R/W	R/W	R				R/W	R/W	R/W

- $\overline{INT}$ Enable 2,1,0
- Undefined Fetch Object
- TRAP

REGISTER	MNEMONICS	ADDRESS
Refresh Control Register : RCR		3 6

bit	REFE	REFW	—	—	—	—	CYC1	CYC0
during RESET	1	1	1	1	1	1	0	0
R/W	R/W	R/W					R/W	R/W

- Cycle Select
- Refresh Wait State
- Refresh Enable

CYC1,0	Interval of Refresh Cycle
0 0	10 States
0 1	20
1 0	40
1 1	80

(to be continued)

REGISTER	MNEMONICS	ADDRESS	REMARKS								
MMU Common Base Register	: CBR	3 8	bit	CB7*	CB6	CB5	CB4	CB3	CB2	CB1	CB0
			during RESET	0	0	0	0	0	0	0	0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

└ MMU Common Base Register

MMU Bank Base Register	: BBR	3 9	bit	BB7*	BB6	BB5	BB4	BB3	BB2	BB1	BB0
			during RESET	0	0	0	0	0	0	0	0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

└── MMU Bank Base Register

MMU Common/Bank Area Register	: CBAR	3 A	bit	CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
			during RESET	1	1	1	1	0	0	0	0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

└ MMU Bank Area Register

└ MMU Common Area Register

Operation Mode Control Register	: OMCR	3 E	bit	MIE	$\overline{\text{MITE}}$	IOC	–	–	–	–	–
			during RESET	1	1	1	1	1	1	1	1
			R/W	R/W	W	R/W					

└ I/O Compatibility

└ $\overline{\text{M1}}$ Temporary Enable

└ $\overline{\text{M1}}$ Enable

I/O Control Register	: ICR	3 F	bit	IOA7	IOA6	IOSTP	–	–	–	–	–
			during RESET	0	0	0	1	1	1	1	1
			R/W	R/W	R/W	R/W					

└ I/O Stop

└ I/O Address

These MMU registers are expanded from 7 bits to 8 bits in the PLCC package

ORDERING INFORMATION

Codes

PACKAGE
P = Plastic Dip
V = Plastic Chip Carrier
F = Quad Flat Pack

TEMPERATURE
S = 0 C to +70 C
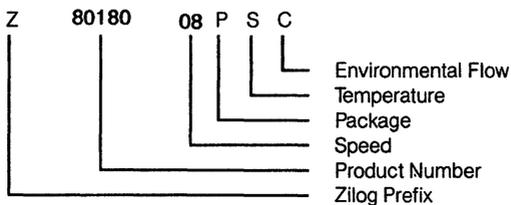E = -40°C to +100°C

SPEED
06 = 6MHz
08 = 8MHz
10 = 10MHz

ENVIRONMENTAL
C = Plastic Standard

Example:
Z8018008PSC is an 80180 8 MHz, Plastic DIP, 0 C to
70 C, Plastic Standard Flow.

```
Z    80180    08 P S C
                    └──── Environmental Flow
                  └────── Temperature
                └──────── Package
              └────────── Speed
        └──────────────── Product Number
    └──────────────────── Zilog Prefix
```

ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA
Agoura 818-707-2160
Campbell.. 408-370-8120
Irvine 714-453-9701

COLORADO
Boulder 303-494-2905

FLORIDA
Largo 813-585-2533

GEORGIA
Norcross 404-448-9370

ILLINOIS
Schaumburg 708-517-8080

MINNESOTA
Minneapolis 612-944-0737

NEW HAMPSHIRE
Nashua... 603-888-8590

OHIO
Independence 216-447-1480

OREGON
Portland 503-274-6250

PENNSYLVANIA
Horsham 215-784-0805

TEXAS
Dallas 214-987-9987

WASHINGTON
Seattle 206-523-3591

INTERNATIONAL SALES OFFICES

CANADA
Toronto 416-673-0634

GERMANY
Munich 49-8967-2045
Sömmerda 49-3634-23906

JAPAN
Tokyo 81-3-3587-0528

HONG KONG
Kowloon 852-7238979

KOREA
Seoul 82-2-577-3272

SINGAPORE
Singapore 65-2357155

TAIWAN
Taipei 886-2-741-3125

UNITED KINGDOM
Maidenhead 44-628-392-00

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-8056

�ù ZiLAG *INQUIRY CARD*

Thank you for your interest in Zilog products. To receive prompt attention and information, please fill out this reply card and mail it to Zilog.

Please send me more information on Zilog products in the following areas:

❏ General-Purpose ❏ Mass Storage ❏ Low Voltage Controllers ❏ Wireless Communications

❏ Multimedia ❏ Fax/Modem Control ❏ Datacommunications ❏ Speech Processing/Synthesis

The primary product/industry in which my company is involved is: _____.

My specific application is: _____.

My need is: ❏ Immediate ❏ Within 6-12 months ❏ Long-term ❏ For information only

Name _____ **Title** _____

Company _____

Address _____

City _____ **ST** _____ **ZIP** _____

Telephone (____)_____, **Ext.** _____

Fax (____)_____

Please tell us where you heard about Zilog:

❏ Colleague's recommendation ❏ Direct sales call ❏ Advertisement (where?_____)

❏ Prior experience with company ❏ Distributor or rep ❏ Conference Presentation (where?_____)

Fold Here

▌▌ ▌ ▌▌

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 35 CAMPBELL CA

POSTAGE WILL BE PAID BY ADDRESSEE

ZILOG INC
MAIL STOP C1-1
210 E HACIENDA AVE
CAMPBELL CA 95008-9694

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES