TMPZ84C013AT-10
TLCS-Z80 MICROPROCESSOR

1. OUTLINE AND FEATURES

The TMPZ84C013A is a high-performance CMOS 8-bit microprocessor incorporating the counter timer circuit (CTC), the serial I/O port (SIO), the clock generator/controller (CGC), and the watchdog timer (WDT) around the TLCS-Z80 MPU. This microprocessor inherits the basic architecture of the TLCS-Z80 series, allowing the user to utilize the software resources and development tools accumulated so far.

The TMPZ84C013A is based on the new CMOS process and housed in a standard 84-pin PLCC package, greatly contributing to system miniaturization and power saving.

The TMPZ84C013A incorporates the high-performance serial I/O port, the counter timer circuit which can be used as the baud rate generator, and the watchdog timer indispensable for control applications, offering the user a wide range of system applications such as the communication controllers including a communication adaptor and the various other controllers which need miniaturization.

Features

- Built-in TLCS-Z80 series MPU, CTC, SIO, CGC and watchdog timer features.
- High speed operation (10MHz operation)
- Built-in clock generator (CGC:Clock Generator Controller)
- Built-in standby capability (with the controller built in) provides 4 operation modes:

  Run mode (Normal operation)
  Idle-1 mode (Only clock oscillation goes on.)
  Idle-2 mode (Wake-up by CTC enabled.)
  Stop mode (Clock oscillation stopped; standby state)

- Wide operational voltage range (5V ± 10% : 10MHz VERSION) supported.
- Wide operating temperature range (−40°C to +70°C : 10MHz VERSION)
- Low power dissipation

  In operation: (RUN mode) 40mA TYP. at 10 MHz
  In idle: (IDLE-1 mode) 2.5mA TYP. at 10 MHz
  (IDLE-2 mode) 19mA TYP. at 10 MHz
  In standby: (STOP mode) 500nA TYP.

- Built-in TLCS-Z80 series SIO capability

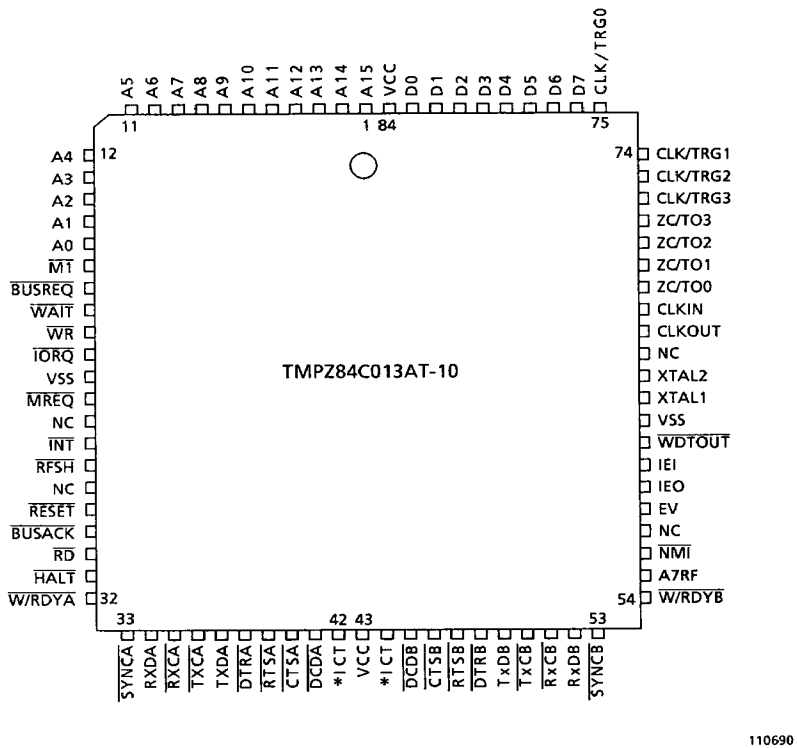
- A pair of independent full duplex channels supports the asynchronous as well as synchronous byte-oriented (monosync and bisync) and bit-oriented HDLC and CCITT-X. 25 protocols.
- Built-in CRC generation and check capability.
- Data transfer rates of up to 2000 K bits/sec (10 MHz).
- Built-in TLCS-Z80 series CTC capability

  Four independent built-in channels.
  The timer or counter modes can be set .
  Also available as the SIO baud rate generator.
- Built-in watchdog timer
- Programmed daisy-chain interrupt control
- Built-in dynamic RAM refresh controller
- TTL/CMOS compatible
- Housed in compact standard 84-pin PLCC package
- The Toshiba real-time emulator (RTE80) and the commercially available Z80 ICE can be used (the TMPZ84C013A used as the evaluator).
- The Toshiba evaluator board installed.

note : Z80 is a trade mark of Zilog Inc.

2. PIN ASSIGNMENT AND FUNCTIONS

2.1 Pin Assignments (Top View)



Note : The ICT pin is the test pin. Do not make any external connection to this pin.

Figure 2.1.1 Pin Assignment

2.2 (A) Pin Functions

(1/5)

Pin	Q'ty (Number)	Type	Function
D0-D7	8 (76-83)	Input / output 3-state	The 8-bit bi-directional data bus.
A0-A15	16 (1-16)	Output 3-state	The 16-bit address bus. These pins specify memory and I/O port addresses. During a refresh cycle, the refresh address is output to the low-order 7 bits and A7RF.
$\overline{M1}$	1 (7)	Output 3-state	The Machine Cycle 1 signal. In an operation code fetch cycle, this pin goes "0" with the $\overline{MREQ}$ signal. At the execution of a 2-byte operation code, this pin goes "0" for each operation code fetch. In a maskable interrupt acknowledge cycle, this pin goes "0" with the $\overline{IORQ}$ signal. When the EV signal is applied, this pin is put in the high-impedance state.
$\overline{RD}$	1 (30)	Output 3-state	The Read signal. It indicates that the MPU is ready for accepting data from memory or I/O device. The data from the addressed memory or I/O devices is gated by this signal onto the MPU data bus. When the $\overline{BUSREQ}$ signal is applied, this pin is put in the high-impedance state.
$\overline{WR}$	1 (20)	Output 3-state	The Write signal. This signal is output when the data to be stored in the addressed memory or I/O device is on the data bus. When the $\overline{BUSREQ}$ signal is applied, this pin is put in the high-impedance state.
$\overline{MREQ}$	1 (23)	Output 3-state	The Memory Request signal. When the execution address for memory access is on the address bus, this pin goes "0". During a memory refresh cycle, this pin also goes "0" with $\overline{RFSH}$ signal.
$\overline{IORQ}$	1 (21)	Output 3-state	The Input/Output Request signal. This pin goes "0" when the address for an I/O read or write operation is on the low-order 8 bits (A0 through A7) of the address bus. The $\overline{IORQ}$ signal is also output with the $\overline{M1}$ signal at interrupt acknowledge to tell an I/O device that the interrupt response vector can be placed on the data bus. Note that the interrupt priority among the TMPZ84C013A CTC, and SIO is selected by a program.
IEO	1 (59)	Output	The Interrupt Enable Output signal. In the daisy chain interrupt control, this signal controls the interrupt from the peripheral LSIs connected next to the TMPZ84C013A. The IEO pin goes "1" only when the IEI pin is "1" and the MPU is not servicing an interrupt from the built-in peripheral LSI.

190589

(2/5)

Pin	Q'ty (Number)	Type	Function
XTAL1 XTAL2	2 (63) (64)	Input Output	The crystal oscillator connection. Connects an oscillator having the oscillation frequency 2 times as high as the system clock (CLKOUT) frequency.
CLKIN	1 (67)	Input	The Single-phase Clock Input. When the clock input is placed in the DC state (continued "1" or "0" level), this pin stops operating and holds the state of that time. Normally, this pin is connected with the CLKOUT pin. However, to operate the system with the external clock, connect the external clock to the CLKIN pin.
CLKOUT	1 (66)	Output	The Single-phase Clock Output. When a Halt instruction is executed in the Stop or Idle-1 mode, the CLKOUT output is retained at "0". In the Run and Idle-2 modes, the clock is kept output. This pin provides the clock to other peripheral ICs.
$\overline{\text{RESET}}$	1 (28)	Input	The Reset signal input. This signal resets the internal states of the TMPZ84C013A. This signal is also used to return from the standby state in the Stop or Idle mode.
$\overline{\text{INT}}$	1 (25)	Input	The Maskable Interrupt signal. An interrupt is caused by the internal CTC, SIO or the peripheral LSI. An interrupt is acknowledged when the interrupt enable flip-flop (IFF) is set to "1" by software. The $\overline{\text{INT}}$ pin is normally wire-ORed and requires an external pullup resistor for these applications. This signal is also used to return from the stand-by state in the Stop or Idle mode.
$\overline{\text{WAIT}}$	1 (19)	Input	The Wait Request signal. This signal indicates the MPU that the addressed memory or I/O device is not ready for data transfer. As long as this signal is "0", the MPU is in the Wait state.
$\overline{\text{BUSREQ}}$	1 (18)	Input	The bus Request signal. The $\overline{\text{BUSREQ}}$ signal forces the MPU address bus, data bus, and control signals $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ to be placed in the high-impedance state. This signal is normally Wire-ORed and requires an external pullup resistor for these applications.
$\overline{\text{BUSACK}}$	1 (29)	Output	The Bus Acknowledge signal. In response to the $\overline{\text{BUSREQ}}$ signal, the $\overline{\text{BUSACK}}$ signal indicates to the requesting peripheral LSI that the MPU address bus, data bus, and control signals $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ have been put in the high-impedance state.

190589

(3/5)

Pin	Q'ty (Number)	Type	Function
$\overline{HALT}$	1 (31)	Output 3-state	The Halt signal. This pin goes "0" when the MPU has executed a Halt instruction and is in the Halt state. It is put in the high-impedance state when the EV signal is applied.
$\overline{RFSH}$	1 (26)	Output	The refresh signal. When the dynamic memory refresh address is on the low-order 8 bits of the address bus, this signal goes "0" . At the same time, the $\overline{MREQ}$ signal also goes active ("0"). This pin is put in the high-impedance state when the EV signal is applied.
CLK/TRG0 ~ CLK/TRG3	4 (72-75)	Input	The external clock/timer trigger. These 4 CLK/TRG pins correspond to 4 channels. In the counter mode, the down counter is decremented by 1 and in the timer mode, the timer is activated at each active edge (a rising or falling edge) of the signal which are input by these pins. It can be selected by program whether the active edge is a rising or falling edge.
ZC/TO0 ~ZC/TO3	4 (68-71)	Output	The Zero Count/Timer Out signal. In either the Timer mode, or counter mode, pulses are output from these pins when the down counter has reached zero.
IEI	1 (60)	Input	The Interrupt Enable Input signal. This signal is used with the IEO to form a priority daisy chain when there is more than one interrupt-driven peripheral LSI.
$\overline{NMI}$	1 (56)	Input	The Non-maskable Interrupt Request signal. This interrupt request has a higher priority than the maskable interrupt and is not dependent on the interrupt enable flip-flop (IFF) state. This signal is also used to return from the stand-by state in the Stop or Idle mode.
EV	1 (58)	Input	The Evaluator signal. When this signal is active, the $\overline{M1}$, $\overline{HALT}$, and $\overline{RFSH}$ pins are put in the high-impedance state. In using the TMPZ84C013A as an evaluator chip, the MPU is electrically disconnected (put in the high-impedance state) after one machine cycle is executed with the EV signal being "1" and the $\overline{BUSREQ}$ signal being "0", and follows the instructions from other MPU (such as the MPU of ICE). The signals of the disconnected MPU are A0 through A15, D0 through D7, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, $\overline{WR}$, $\overline{M1}$, $\overline{HALT}$, and $\overline{RFSH}$. $\overline{BUSACK}$ needs to be disconnected by an externally connected circuit. The evaluator board is available to use the TMPZ84C013A as an evaluator chip.

190589

(4/5)

Pin	Q'ty (Number)	Type	Function
A7RF	1 (55)	Output	The 1-bit auxiliary address bus. This pin outputs the same signal as the bit 7 (A7) of the address bus. However, during a refresh cycle, this pin outputs the address which is the most significant bit of the 8-bit refresh address signal linked to the low-order 7 bits of the address bus.
$\overline{W/RDYA}$ $\overline{W/RDYB}$	2 (32, 54)	Output	The Wait/Ready signal A and the Wait/Ready signal B. These signals can be used as the Wait or Ready depending on SIO programming. When these signals are programmed as "Wait", they go active at "0" to indicate to the MPU that the addressed memory or I/O devices are not ready for data transfer, requesting the MPU to wait. When these signals are programmed as "Ready", they go active at "0" to determine when a peripheral device associated with a DMA port is ready for a read or write data. The DMA is requested to transfer data.
$\overline{SYNCA}$ $\overline{SYNCB}$	2 (33, 53)	Input / output	The Synchronization signal. In the asynchronous receive mode, these signals act as the $\overline{CTS}$ and $\overline{DCD}$ signals. In the external sync mode, these signals act as inputs and in the internal sync mode, they act as outputs.
RXDA RXDB	2 (34, 52)	Input	The Serial Receive Data signal.
$\overline{RXCA}$ $\overline{RXCB}$	2 (35, 51)	Input	The Receive Clock signal. In the asynchronous mode, the Receive Clocks may be 1, 16, 32 or 64 times the data transfer rate.
$\overline{TXCA}$ $\overline{TXCB}$	2 (36, 50)	Input	The Transmitter Clock signal. In the asynchronous mode, the Transmitter Clocks may be 1, 16, 32, or 64 times the data transfer rate.
TXDA TXDB	2 (37, 49)	Output	The serial transmit data signal.
$\overline{DTRA}$ $\overline{DTRB}$	2 (38, 48)	Output	The Data Terminal Ready signal. These signals indicate whether the data terminal is ready to receive serial data or not. When it is ready, these signals go active to enable the transmitter of the terminal. When it is not ready, these signals go inactive to disable the transfer from the terminal.
$\overline{RTSA}$ $\overline{RTSB}$	2 (39, 47)	Output	The Request to Send signal. These pins are "0" when transmitting serial data. That is, to transmit data, these signals are made active to enable their receivers.

190589

(5/5)

Pin	Q'ty (Number)	Type	Function
$\overline{CTSA}$ $\overline{CTSB}$	2 (40, 46)	Input	The Clear To Send signal. When these pins are "0", the modem having transmitted these signals is ready to receive serial data. When it is ready, these signals go active to enable the transmitter of the terminal. When it is not ready, these signals go inactive to disable the transfer from the terminal.
$\overline{DCDA}$ $\overline{DCDB}$	2 (41, 45)	Input	The Data Carrier Detect signal. When these pins are "0" the receive of serial data can be enabled. That is, to transmit data, these signals are made active to enable their receivers.
ICT	2 (42, 44)	Output	The test pins. To be used in the open state.
$\overline{WDTOUT}$	1 (61)	Output	The Watchdog Timer signal. The output pulse width depends on the externally connected pin.
VCC	2 (43, 84)		The power supply (+ 5 V) pins.
VSS	2 (22, 62)		The ground (0 V) pins.

190589

2.2 (B) TMPZ84C013A Internal I/O Address Map

Internal I/O	Channel	I/O Address
CTC (Counter Timer)	ch 0	# 10
	ch 1	# 11
	ch 2	# 12
	ch 3	# 13
SIO (Serial I/O)	ch A Send/Receive buffer	# 18
	ch A Command/Status Register	# 19
	ch B Send/Receive buffer	# 1A
	ch B Command/Status Register	# 1B
Watch Dog Timer Stand-by mode Register	WDTER, WDTPR, HALTMR	# F0
Watch Dog Timer command Register	Clear command ($4E_H$) disable command ($B1_H$)	# F1
Daisy-chaine interrupt precedence Register	Only use bit2~bit0	# F4

190589

3. OPERATIONAL DESCRIPTION

3.1 Block Diagram and Operational Outline

3.1.1 Block Diagram

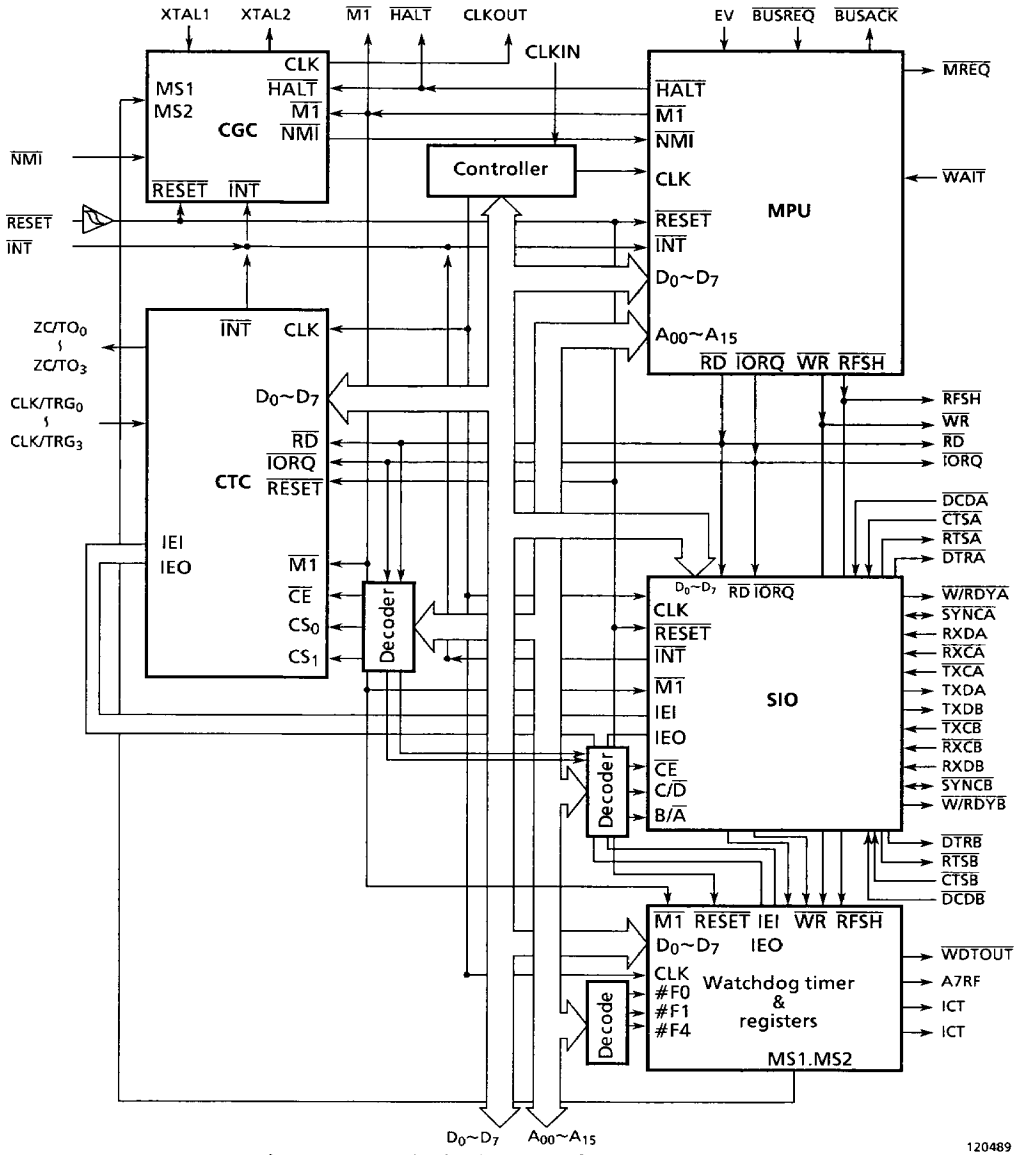


Figure 3.1.1 Block Diagram of TMPZ84C013A

3.1.2 Operational Outline

The TMPZ84C013A largely consists of a processor (MPU), a counter/timer circuit (CTC), a serial input/output controller (SIO), a watchdog timer (WDT), and a clock generator/controller (CGC).

- The MPU provides all the capabilities and pins of the Toshiba TLCS-Z80 MPU (TMPZ84C00A) to play the role of the TLCS-Z80 microprocessor perfectly.
- The CTC provides the capabilities of the Toshiba TLCS-Z80 CTC (TMPZ84C30A) and has the pins required to perform the necessary operations as a TLCS-Z80 peripheral LSI. The four independent timer channels are I/O-addressed internally.
- The SIO provides the capabilities of the Toshiba TLCS-Z80 SIO (TMPZ84C43A) and has the pins required to perform the necessary operations as a TLCS-Z80 peripheral LSI. The two independent serial channels are I/O-addressed internally.
- The WDT incorporates one-channel watchdog timer and the read/write-enabled watchdog timer control registers indispensable for control applications. The WDT also has the register to determine interrupt priorities, allowing the user to set the daisy-chain interrupt priorities by program. Additionally, the WDT has the IEI and IEO pins required to process the daisy- chain interrupts caused by the peripheral LSIs to be added both inside and outside the TMPZ84C013A.
- The CGC provides the four operation modes to control the entire TMPZ84C013A chip; the Run, Idle-1, Idle-2, and Stop modes. They are program-selectable. This chip has two clock pins: CLKOUT to supply clock from the CGC and CLKIN to get clock from the outside. Therefore, the TMPZ84C013A can be operated on the clock supplied from the outside at the CLKIN pin without using the CGC. The following briefly describes the four operation modes of the CGC with the CLKOUT and CLKIN pins connected:
- In the Run mode, the clock generated by the CGC is supplied to the TMPZ84C013A and peripheral LSIs to perform the normal programmed microcomputer operations.
- In the Idle-1 mode, clock oscillation is going on but the clock is not supplied to the TMPZ84C013A and peripheral LSI, thereby saving the system power consumption and shortening the time required for system restart.

- In the Idle-2 mode, clock oscillation is performed and the clock is output from the CLKOUT pin. The clock is supplied only to the CTC in the TMPZ84C013A, permitting a wake-up operation by the CTC. Like the Idle-1 mode, the Idle-2 mode saves the system power consumption and shortens the time required for system restart.
- In the Stop mode, clock oscillation is not performed and the system operation can be stopped completely. In this mode, the system can be restarted with the internal data retained with an extremely low power consumption level unique to the CMOS technology.

Note that these modes can be set only when the MPU has executed a HALT instruction.

Additionally, the TMPZ84C013A has also the EV pin which is used with the $\overline{\text{BUSREQ}}$ pin to put the MPU in the high-impedance state for electrical disconnection, thus functioning as an evaluator chip. That is, the MPU in the TMPZ84C013A is electrically disconnected by these two pins to implement the emulation by the signal from the in-circuit emulator (ICE). For emulation, one machine cycle is performed on the MPU in the TMPZ84C013A with EV being "1" and the $\overline{\text{BUSREQ}}$ being "0" then the emulation is performed as instructed by the MPU. The MPU signals to be disconnected are A0 through A15, D0 through D7, $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, $\overline{\text{M1}}$, $\overline{\text{HALT}}$, and $\overline{\text{RFSH}}$, $\overline{\text{BUSACK}}$ needs to be disconnected by an externally connected circuit.

3.2 MPU Operations

This subsection describes the system configuration, functions and the basic operations of the MPU of the TMPZ84C013A.

3.2.1 Block Diagram

Figure 3.2.1 shows the block diagram of the MPU.

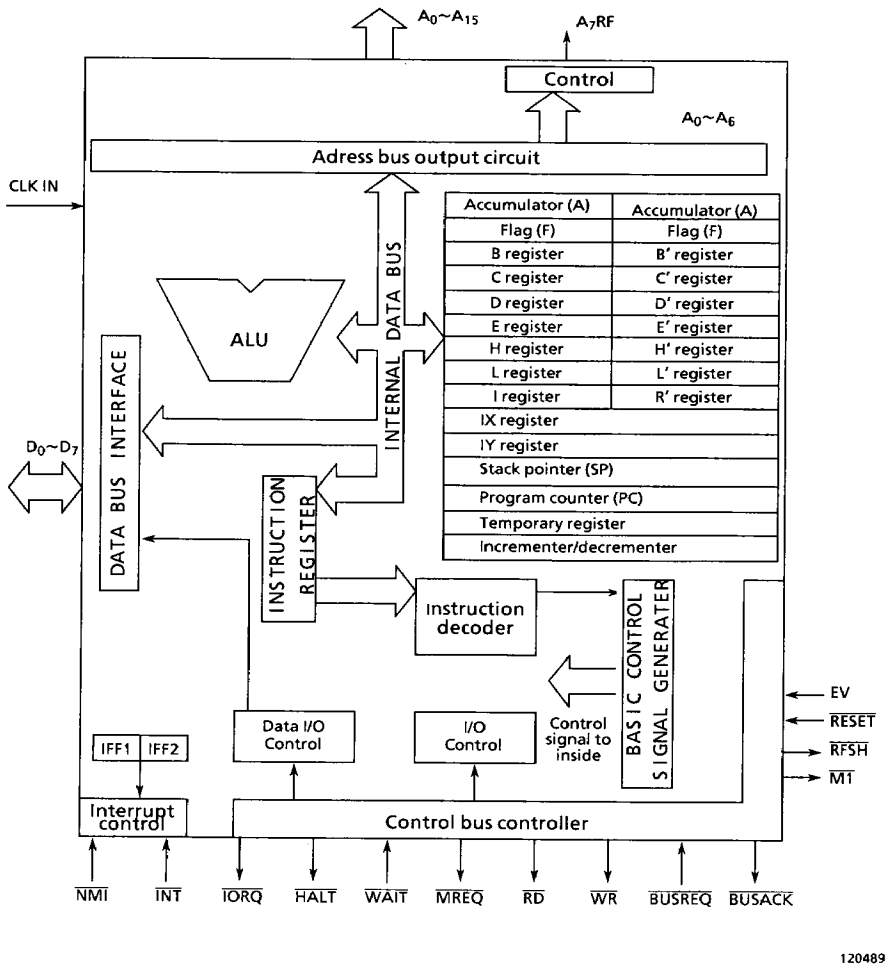


Figure 3.2.1 MPU Block Diagram

3.2.2 MPU System Configuration

The MPU has the configuration shown in Figure 3.2.1. The address signal is put on the address bus via the address buffer. The data bus is controlled for input or output by the data bus interface. Both the address and data buses are put in the high-impedance state by the $\overline{\text{BUSREQ}}$ signal input to make them available for other peripheral LSIs. The Opcode read from memory via the data bus is written to the instruction register. This Opcode is decoded by the instruction decoder. According to the result of the decoding, control signals are sent to the relevant devices. Receiving these control signals, the ALU performs various arithmetic operations. The register array temporarily hold the information required to perform operation.

The following describes the MPU's main components and functions which the user must understand to operate the TMPZ84C013A:

[1] Internal Register Groups

The configuration of the internal register groups is as follows:

(1) Main registers

A, F, B, C, D, E, H, L

(2) Alternate registers

A', F', B', C', D', E', H', L'

(3) Special purpose registers

I, R, IX, IY, SP, PC

Figure 3.2.3 shows the configuration of the internal register groups. The register groups, each being of a static RAM, consists of eighteen 8-bit registers and four 16-bit registers. The following describes the function of each register:

(1) Main registers (A, F, B, C, D, E, H, L)

(a) Accumulator (A)

The accumulator is an 8-bit register used for arithmetic and data transfer operations.

(b) Flag register (F) (see Figure 3.2.2)

The flag register is an 8-bit register to hold the result of each arithmetic operation. Actually, the 6 of the 8 bits are set ("1")/reset ("0") according to the condition specified by an instruction.

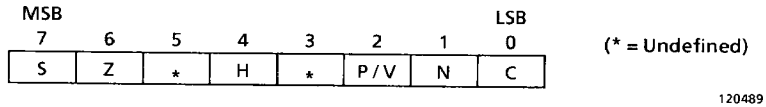


Figure 3.2.2 Flag Register Configuration

The following 4 bits are directly available to the programmer for setting the jump, call and return instruction conditions:

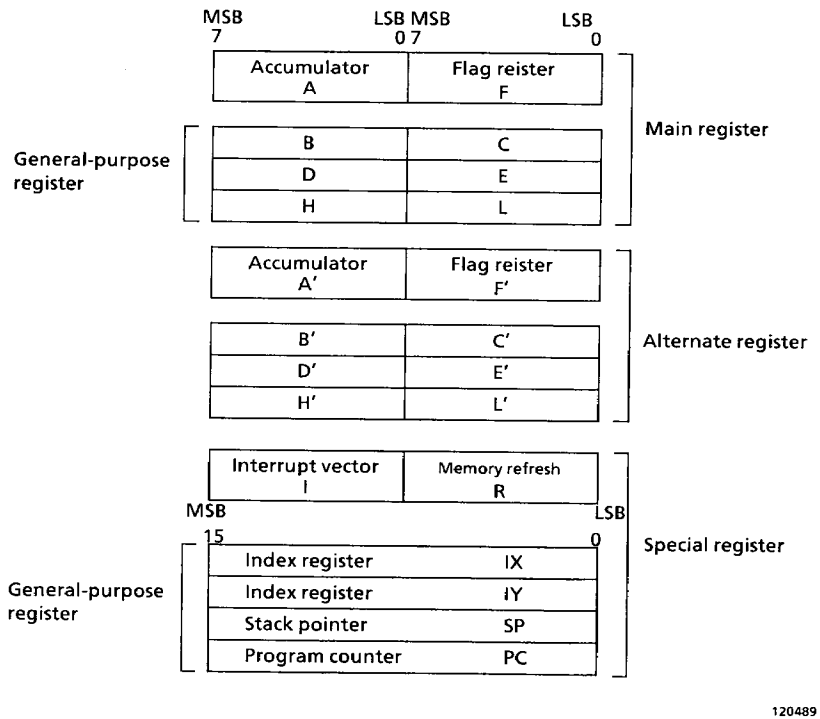


Figure 3.2.3 Flag Register Configuration

- Sign flag (S)

  When the result of an operation is negative, the S flag is set to "1" . Actually, the content of bit 7 of accumulator is stored in this flag.

- Zero flag (Z)

  When all bits turn out to be "0" s after operation, the Z flag is set to "1" . Otherwise, it is set to "0". With a block search instruction (CPI, CPIR, CPD or CPDR), the Z flag is set to "1" if the source data and the accumulator data match. With a block I/O instruction (INI, IND, OUTI or OUTD), the Z flag is set to "1" if the content of the B register used as the byte counter is "0" at the end of comparison.

- Parity/overflow flag (P/V)

  This flag has two functions. One is the parity flag (P) that indicates the result of a logical operation (AND A, B etc.). The P flag is set to "1" if the parity is even as a result of the operation on signed values by two's complement. It is reset to "0" if the parity is odd. With a block search instruction (CPI, CPIR, CPD or CPDR) and a block transfer instruction (LDI or LDD), the P flag indicates the state of the byte counter (register pair B and C). It is set to "1" if the byte counter is not "0" and reset to "0" when the byte counter becomes "0" (at the end of comparison or data transfer). The content of the interrupt enable flip-flop (IFF) is saved to the P flag when the contents of the R register or I register are transferred to the accumulator.

  The other use of the P/V flag is the overflow flag (V) that indicates whether an overflow has occurred or not as a result of an arithmetic operation. The V flag is set to "1" when the value in the accumulator gets out of a range of the maximum value +127 and the minimum value −128 and therefore cannot be correctly represented as a two's complement notation.

  Whether the P/V flag operates as the P flag or V flag is determined by the type of the instruction executed.

- Carry flag (C)

  The C flag is set to "1" if a carry occurs from bit 7 of the accumulator or a borrow occurs as a result of an operation.

  The following two flags are not available to the programmer for the test and set ("1")/reset ("0") purposes. They are internally used by the MPU for BCD arithmetic operations.

- Half carry flag (H)

  The H flag is used for holding the carry or borrow from the low-order 4 bits of a BCD operation result. When a DAA instruction (decimal adjust) is executed, the MPU automatically uses the H flag to adjust the result of a decimal addition or subtraction.

- Add/subtract flag (N)

  In BCD operation, algorithm is different between addition and subtraction. The N flag indicates whether the executed operation is addition or subtraction.

  For change of the flag state depending on the instruction, see 3.2.4 "TMPZ84C013A Instruction Set".

(c) General-purpose registers (B, C, D, E, H, L)

General-purpose registers consist of 8 bits each. They are used as 16-bit register pairs (BC, DE, HL) as well as independent 8-bit registers to supplement the accumulator. The B register and the register pair BC are used as counters when a block I/O, block transfer, or search instruction is executed. The register pair HL has various memory addressing features as compared with the register pairs BC and DE.

(2) Alternate registers (A', F', B', C', D', E', H', L')

The configuration of the alternate registers is exactly the same as that of the main registers. There is no instruction that handles the alternate registers directly. The data in the alternate registers are processed by moving them into the main registers by means of exchange instructions as shown below:

EX AF, AF' (A↔A', F↔F')

EXX (B↔B', C↔C', D↔D', E↔E', H↔H', L↔L')

When a hign-speed interrupt response has been requested within the system, these instruction can be used to quickly move the contents of the accumulator, flag registers, and general-purpose registers into the corresponding registers. This eliminates the need for transferring the register contents to/from the external stack during execution of the interrupt handling routine, thereby shortening the interrupt servicing time greatly.

(3) Special purpose registers (I, R, IX, IY, SP, PC)

(a) Interrupt page address register (I)

The TMPZ84C013A provides two kinds of interrupts :maskable interrupt (INT) and non-maskable interrupt (NMI). The maskable interrupt provides three modes (0, 1, and 2) in which the interrupt is handled. These modes can be selected by instructions IM0, IM1, and IM2 respectively. In Mode 2, any memory location can be called indirectly depending on the interrupt. For this purpose, the I register stores the high-order 8 bits of the indirect address. The low-order 8 bits are supplied from the interrupting peripheral LSI. This scheme permits calling the interrupt handling routine from any memory location in an extremely short access time. For the details of interrupts, see [4] "Interrupt Capability".

(b) Memory refresh register (R)

The R register is used as the memory refresh counter when the dynamic RAM is used for memory. This permits using of the dynamic memory in the same manner as the static memory. This 8-bit register is automatically incremented for each instruction fetch. While the MPU decodes and executes the fetched instruction, the contents of the R register are synchronized with the refresh signal to place the low-order 7 bits and A7RF on the address bus. This operation is all performed by the MPU and, therefore, dose not need a special processing by program. The MPU operation is not delayed by this operation. During refresh, the contents of the I register are placed on the high-order 8 bits of the address bus.

(c) Index registers (IX, IY)

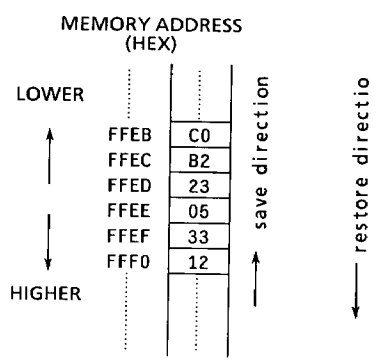
The two independent index registers IX and IY hold the 16-bit base address when used in the index addressing mode. In this addressing mode, the memory address obtained by adding the contents of an index register to the displacement value (for example, LD IX+40H) is specified. This mode is convenient for using data tables. Also these registers can be used separately for memory addressing and data retaining registers.

(d) Stack pointer (SP)

The stack pointer is a 16-bit register to provide the start address information in the stack area in the external RAM. The content of the stack pointer is decremented at the execution of a call instruction or PUSH instruction or interrupt handling and is incremented at the execution of a return instruction or POP instruction. At the execution of a call instruction or interrupt handling, the current content of the program counter is saved into the stack. At the execution of a return instruction, the content is restored from the stack to the program counter. These operations are all performed by the MPU automatically. However, the other registers are not saved or restored automatically. For the storing of the contents of these registers, an exchange instruction (EX or EXX) for alternate register, a PUSH or a POP instructions must be used. When a PUSH instruction is executed, the contents of the specified register are saved into the stack. When a POP instruction is executed, the contents of the stack are moved to the specified register.

These data are restored on a last-in, first-out basis. Use of the stack permits processing of multiple-level interrupts, deep subroutine nestings, and various data manipulation very easily. The stack pointer is not initialized in the hardware approach. Therefore, it is required to allocate the stack area in RAM to specify initialization (at the highest address of the stack area) in the initial program.

(ex)



MEMORY ADDRESS (HEX)	INSTRUCTION		The contents of the SP before the instruction is accepted.
1230	CALL	1500H	FFF1
1500	PUSH	AF	FFEF ; A=05, F=23
1501	PUSH	BC	FFED ; B=B2, C=C0
1600	POP	BC	FFEB
1601	POP	AF	FFED
1602	RET		FFEF

The foregoing example shows the stack pointer and stack operations in which the instructions starting with the CALL at address 1230H and ending with the RET at address 1602H have been executed. However, it is assumed that there is no instruction or interrupt other than shown above that uses the stack during the execution. When the value of the stack pointer before executing the CALL instruction at address 1230H indicates address FFF1H, address 1233H is stored at addresses FFF0H and FFEFH because the CALL instruction consists of 3 bytes, then the stack pointer is decremented. Similarly, the data are saved or restored sequentially according to the instructions. These stack and stack pointer operations are all performed automatically.

(e) Program counter (PC)

The program counter holds, in 16 bits, the memory address of the instruction to be executed next. The MPU fetches the instruction from the memory location indicated by the program counter. When the content of the program counter is put on the address bus, the program counter is incremented automatically. However, it is not incremented with a jump instruction, a call instruction, or interrupt processing. Instead, the specified new address is set on it. With a return instruction, the content restored from the stack is set on the program counter.

These operations are all performed automatically and therefore, no care is required for programming.

[2] Halt Capability

When a HALT instruction has been executed, the MPU is put in the halt state. The halt capability can be used to halt the MPU against the external interrupts, thereby reducing the power dissipation. In the halt state the states of MPU's internal registers are retained. The halt state is cleared by reset or when an interrupt is accepted. For the details of halt operation, see [3] "Basic Timing".

(1) Halt operation

When a HALT instruction has been executed, the MPU sets the $\overline{\text{HALT}}$ signal to "0" to indicate that the MPU is going to be put in the halt state. Actually, the MPU in the halt state automatically continues executing NOP instructions if there is the system clock input. However, the program counter is not incremented. This keeps the refresh signal generated when the dynamic memory is used. During halt, the MPU's internal states are retained. The TMPZ84C013A contains the clock generator/controller, easily implementing the clock input control for these halt operations.

(2) Releasing the halt state

The halt state is cleared by accepting an interrupt (the $\overline{INT}$ or $\overline{NMI}$ signal input) or by reset (the $\overline{RESET}$ signal input). When an interrupt is accepted, the halt state is cleared and the interrupt handling routine is executed. However, a maskable interrupt (INT) cannot be accepted unless the interrupt enable flip-flop (IFF) is set.

Note that when the halt state is cleared by the $\overline{RESET}$ signal, the MPU is reset and the program counter is set to "0".

[3] RESET Signal

Holding the $\overline{RESET}$ pin at the low level ("0") under the following conditions, the MPU's internal states are reset:

(1) The supply voltage level is within the operational voltage range.

(2) System clock stabilization.

(3) Holding the $\overline{RESET}$ signal at the low level ("0") for at least 3 full clock cycles. When the $\overline{RESET}$ signal goes high ("1"), the MPU starts executing instructions from address 0000H after at least 2T state dummy cycles.

When reset, the MPU performs the following processing:

(a) Program counter

0000H is set.

(b) Interrupt

The interrupt enable flip-flop (IFF) is reset to "0" to disable the maskable interrupt. For the maskable interrupt processing, mode 0 is specified.

(c) Control output

All control outputs are made inactive ("1"). Therefore, the halt state is also cleared.

(d) Interrupt page address register (I register)

The content of the R register becomes 00H.

(e) Refresh register (R register)

The content of the R register becomes 00H.

The contents of the registers other than above and the external memory do not change.

Therefore, they must be initialized as required.

[4] Interrupt Capability

The interrupt capability is used to suspend the execution of the current program and execute the processing of the requested peripheral LSI. Normally, this interrupt processing routine contains the data exchange and transfer of status and control information between the MPU and the peripheral LSI. When this routine has been completed, the MPU returns to the state before the interrupt has been accepted.

The TMPZ84C013A provides the non-maskable interrupt (NMI) and maskable interrupt (INT) capabilities which are detected by the $\overline{NMI}$ and $\overline{INT}$ interrupt request signals, respectively. A non-maskable interrupt, when caused by a peripheral LSI, is accepted unconditionally. This interrupt is used to support critical functions such as the protection of the system from unpredictable happening including power failure. A maskable interrupt can be enabled or disabled by program. For example, if the timer is used and, therefore, an interrupt is not desired, the system can be programmed to disable the interrupt. Table 3.2.1 lists the processing by interrupt source.

(1) Interrupt enable/disable

A non-maskable interrupt cannot be disabled by program, while a maskable interrupt can be enabled or disabled by program. The MPU has the interrupt enable flip-flop (IFF). A maskable interrupt can be enabled or disabled by setting this flip-flop to "1" (set) or "0" (reset) through an EI instruction (enable) or a DI instruction (disable) in program. Actually, the IFF consists of two flip-flops IFF1 and IFF2. IFF1 is used to select between the enable and disable of a maskable interrupt. IFF2 holds the state of IFF1 before a maskable interrupt has been accepted. Both IFF1 and IFF2 are reset to "0" when any of the following conditions occurs, disabling an interrupt:

- MPU reset
- Execution of DI instruction
- Acceptance of maskable interrupt

Both IFF1 and IFF2 are set to "1" when the the following condition occurs, enabling an interrupt:

- Execution of EI instruction

Actually, the waiting maskable interrupt request is accepted after the execution of the instruction that follows the EI instruction.
This delay by one instruction is caused by accepting an interrupt after completion of the execution of a return instruction if the instruction following the EI instruction is a return instruction.

In the above operation, the contents of IFF1 and IFF2 are the same.

Table 3.2.1 Processing by Interrupt Source

Interrupt Source	Priority	Programmed condition		Vector address	Interrupt return instruction
Non-maskable interrupt (the falling edge of $\overline{NMI}$)	1	None		Address 66H	RETN
Maskable interrupt ($\overline{INT}$ becomes "0" at instruction's last clock)	2	IFF = 1	Mode 0	Instruction from peripheral LSI. Normally, CALL or RST instruction.	(Note) RETI
			Mode 1	Address 38H.	
			Mode 2	The address indicated by the data table (memory) at the address specified by I register (high-order 8 bits) and data from peripheral LSI (low-order 8 bits, LSB = "0").	

120489

Note : Mode 0 applies when the instruction from peripheral LSI is CALL or RST instruction.

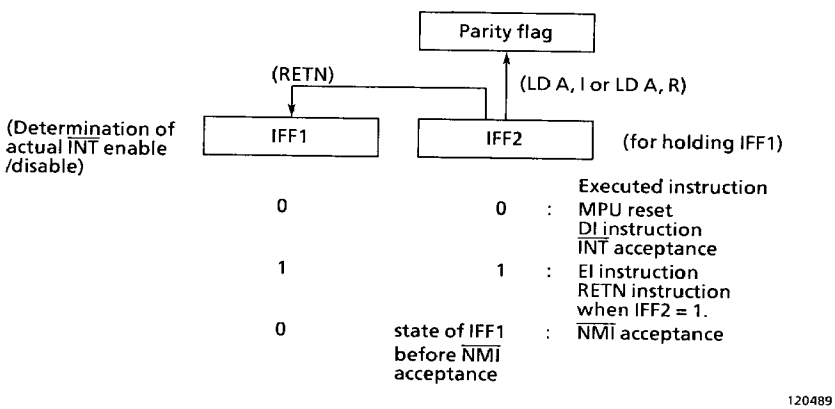


Figure 3.2.4 Interrupt Enable Flip-Flop (IFF)

When a non-maskable interrupt has been accepted, IFF1 is reset to "0" (interrupt disable) until an EI or RETN instruction is executed, so as to prevent from accepting the next interrupt. For this purpose, the state (interrupt enable/disable) of IFF1 immediately before non-maskable interrupt acceptance must be stored. This state is copied into IFF2 upon acceptance of a non-maskable interrupt. The content of IFF2 is copied into the parity flag at the execution of the following instructions, so that the copied data can be tested or stored:

- The load instruction (LD A, I) to load the contents of the I register into the accumulator.
- The load instruction (LD A, R) to load the contents of the R register into the accumulator.

When the return instruction (RETN) from the non-maskable interrupt is executed, the contents of the current IFF2 are copied back to IFF1. If an operation which changes the contents of IFF2 (due to the execution of EI or DI instruction, for example) has not been performed during interrupt handling, IFF1 automatically returns to the state immediately before the interrupt acceptance. Table 3.2.1 lists the states of IFF1 and IFF2 after execution of interrupt-related instructions.

Table 3.2.2 State of IFF1 and IFF2

Operation sequence	IFF1	IFF2	Remarks
MPU reset	0	0	
EI	1	1	
NMI acceptance	0	1	
LD A, I	*	*	Parity flag←IFF2
RETN	1	1	IFF1←IFF2
LD A, R	*	*	Parity flag←IFF2
INT acceptance	0	0	
RETI	*	*	
EI	1	1	
NMI acceptance	0	1	
DI	0	0	
RETN	*	*	

Note : * = no change. 120489

(2) Interrupt processing

With a non-maskable interrupt, the internal NMI flip-flop is set to "1" on the falling edge of the interrupt signal, $\overline{NMI}$. The state of this flip-flop is sampled on the rising edge of the last clock of each instruction to accept an interrupt. A maskable interrupt is accepted if the interrupt signal $\overline{INT}$ is low ("0") on the rising edge of the last clock of each instruction and the interrupt enable state (IFF = 1 and $\overline{BUREQ}$ signal = inactive ("1")) is on. The following is the processing to be performed after a non-maskable interrupt and a maskable interrupt are accepted:

(a) Non-maskable interrupt (NMI)

When a non-maskable interrupt has been accepted, the MPU performs the following processing:

1 The internal NMI flip-flop is reset to "0".

2 IFF1 is reset to "0", disabling the maskable interrupt.

The contents of the IFF1 immediately before the interrupt acceptance are copied into the IFF2.

3 The content of the current program counter is saved into the stack.

4 The instructions starting from non-maskable interrupt vector address 66H are executed.

The non-maskable interrupt processing program terminates after executing the RETN instruction. This return instruction performs the followings:

1 The contents of the current IFF2 are copied into IFF1.

2 The contents of the program counter are restored from the stack.

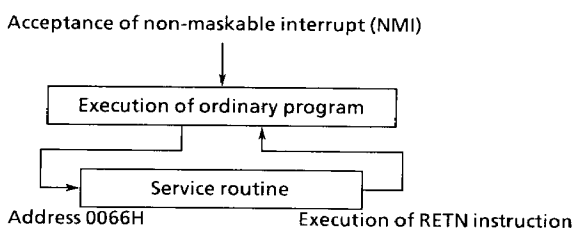


Figure 3.2.5 Non-Maskable Interrupt Processing

(b) Maskable interrupt (INT)

When a maskable interrupt has been accepted, the MPU performs the following processings:

1 Both IFF1 and IFF2 are reset to "0", disabling the maskable interrupts.

2 The content of the current program counter is saved into the stack.

3 A maskable interrupt is serviced in one of the three modes 0, 1 and 2. A mode is selected by executing the instruction IM0, IM1or IM2 before the interrupt is serviced. The instructions are executed starting from the vector address corresponding to the selected mode.

- Mode 0

In mode 0, the interrupting peripheral LSI puts a restart instruction (RST) or a call instruction (CALL) on the data bus and the MPU executes the interrupt service routine according to that instruction. At reset, this mode is automatically set.

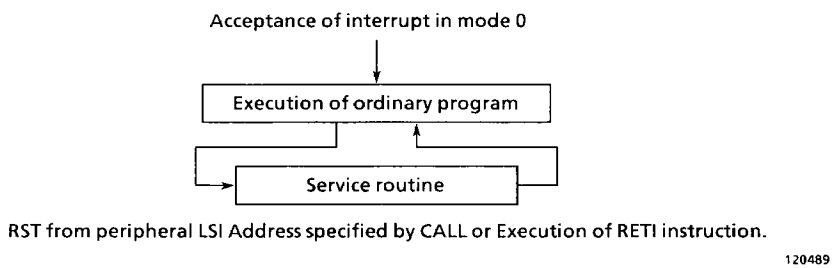


Figure 3.2.6 Interrupt Processing in Mode 0

- Mode 1

When an interrupt is accepted in mode 1, restart is performed from address 0038H. Therefore, the service routine for this interrupt is programmed from the address 0038H .

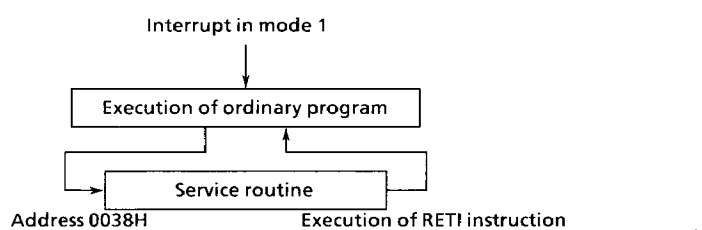


Figure 3.2.7 Interrupt Processing in Mode 1

- Mode 2

The interrupt processing in mode 2 requires a 16-bit pointer consisting of the high-order 8 bits of the I register and the low-order 8 bits (with the LSB = "0") of the data fetched from the interrupting CTC or TLCS-Z80 family peripheral LSI. Therefore, the necessary value must be loaded in the I register beforehand. This pointer is used to specify the memory address in the table. The contents of the specified address and the next address provide the start address of the service routine. Therefore, use of this mode requires the table of the service routine's start address (16 bits) to be set at appropriate location under software control . This location can be anywhere in memory. The LSB of the table pointer is set to "0" because a 2-byte data is needed to specify the service routine start address in 16 bits and start that address from an even-number address. In the table, the start address begins with the low-order byte followed by the high-order byte as shown in Fig. 3.2.8.

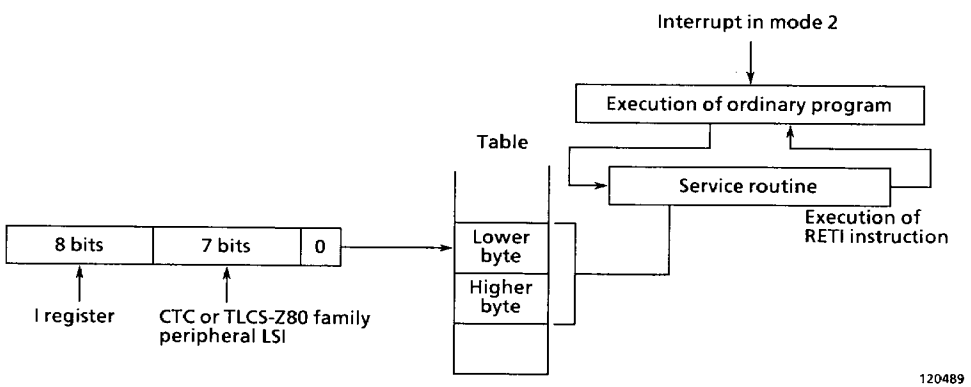


Figure 3.2.8 Interrupt Processing in Mode 2

Mode 2 is used in the daisy chain interrupt processing using the CTC and TLCS-Z80 family LSI. The CTC and TLCS-Z80 family peripheral LSIs all contain the interrupt priority controller in daisy chain structure. In this interrupt structure, the interrupt request signals are connected one after another and given priorities for processing when two or more maskable interrupt requests occur at a time. Only the interrupt vector from the peripheral LSI having the highest priority is put on the data bus. By receiving the interrupt vector in mode 2, the processing for that peripheral LSI can be performed. When an interrupt requested by a peripheral LSI having a priority higher than that of the current peripheral LSI during the execution of the interrupt processing routine, the higher priority interrupt can be enabled by the EI instruction to form an interrupt nesting.

The maskable interrupt processing program terminates by executing an RETI instruction. This return instruction performs the following processings:

- Restores the content of the program counter from the stack.
- Notifies the requesting peripheral LSI of the termination of interrupt processing.

3.2.3 MPU Status Transition Diagram and Basic Timing

The following describes the MPU status transition and the basic timing of each MPU operation.

[1] Instruction Cycle

Each TMPZ84C013A instruction is executed by combining the basic operations of memory read/write, input/output, bus request/acknowledge, and interrupt. These basic operations are performed synchronizing with the system clock (the CLK signal).

One clock period is called a state (T). The smallest unit of each basic operation is called a machine cycle (M). Each instruction consists of 1 to 6 machine cycles and each machine cycle consists of 3 to 6 clock states basically. However, the number of clock states in a machine cycle can be increased by the $\overline{WAIT}$ signal described later on. Figure 3.2.9 shows an example of the basic timing of a 3-machine-cycle instruction.

The first machine cycle (M1) of each instruction is the cycle in which the Opcode of the instruction to be executed next is read (this is called the Opcode fetch cycle). The Opcode fetch cycle basically consists of 4 to 6 clock states. In the machine cycle that follows the Opcode fetch cycle, data is transferred between the MPU and the memory or peripheral LSIs. This operation basically consists of 3 to 5 clock states.

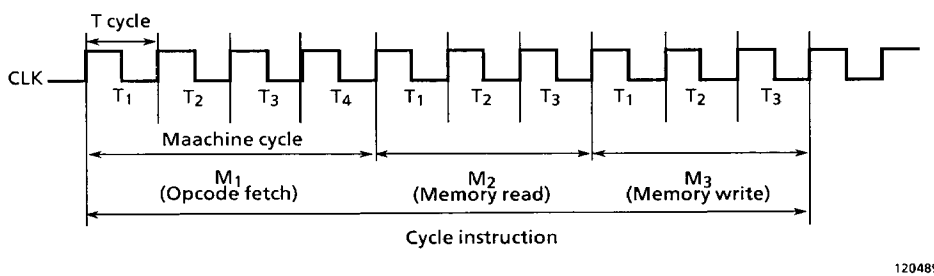


Figure 3.2.9 Example of MPU Basic Timing (3-Machine-Cycle Instruction)

[2] Status Transition Diagram

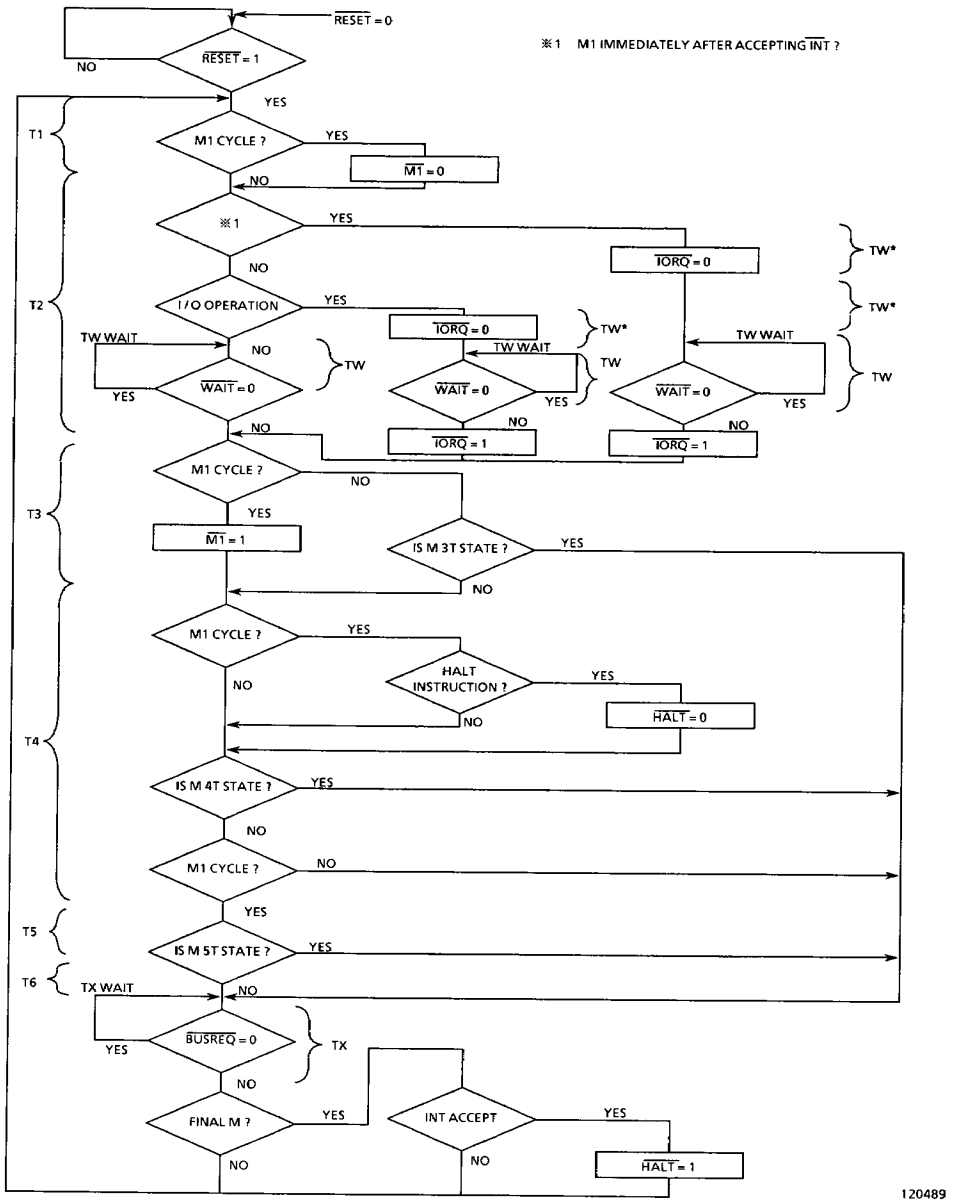


Figure 3.2.10 Status Transition Diagram

[3] Basic Timing

(1) Opcode fetch cycle (M1)

In the Opcode fetch cycle, MPU fetches an Opcode in the machine-language codes in memory. This is also called the M1 cycle because it is the first machine cycle to execute each instruction.

Figure 3.2.12 shows the basic timing of a basic Opcode fetch cycle.

In clock state T1, the content of the program counter is put on the address bus. The $\overline{M1}$ signal goes "0", indicating to the MPU that this is the Opcode fetch cycle. At the same time, $\overline{MREQ}$ and $\overline{RD}$ signals go "0". When the $\overline{MREQ}$ signal goes "0", the address signal has already been stabilized. Therefore, this signal can be used for the memory chip enable signal. The $\overline{RD}$ signal indicates that the MPU is ready to accept the data from memory. By these signals, the MPU accesses memory to fetch the Opcode in the instruction register. The MPU samples the $\overline{WAIT}$ signal on the falling edge of clock state T2. If the $\overline{WAIT}$ signal is "0" on the falling edge of clock state T2 and the following wait state (TW), the next state becomes clock state TW. Figure 3.2.13 shows the delay state of the Opcode fetch cycle caused by the $\overline{WAIT}$ signal.

The data (Opcode) on the data bus is fetched on the rising edge of clock state T3 then, the $\overline{MREQ}$, $\overline{RD}$, and $\overline{M1}$ signals go "1". In clock state T3, a memory refresh address is put on the 8 bits consisting of the low-order 7 bits of the address bus and the A7RF corresponding to bit 8 and the $\overline{RFSH}$ signal goes "0" and the $\overline{MREQ}$ signal goes "0" again. This signal indicates that the memory refresh cycle is on. At this time, the contents of the I register are put on the high-order 8 bits of the address bus and the 7 bits of the R register contents and the A7RF signal corresponding to bit 8 are put on the low-order 8 bits of the address bus. By using the $\overline{RFSH}$ and $\overline{MREQ}$ signals, memory refresh is performed in clock state T3 and T4. However, the $\overline{RD}$ signal remains "1" because the contents of the memory refresh address are not put on the data bus.

The address bus of 8 bits consisting of the low-order 7 bits of address (A6 through A0) and the A7RF are used as the 8-bits refresh address. That is, when A7RF is used for the refresh address, signals "00H" through "FFH" are output. In cycles other than the refresh cycle, the signal equivalent to A7RF are output. However, at reset, the signals to be output are uncertain. Figure 3.2.11 shows the refresh timing.

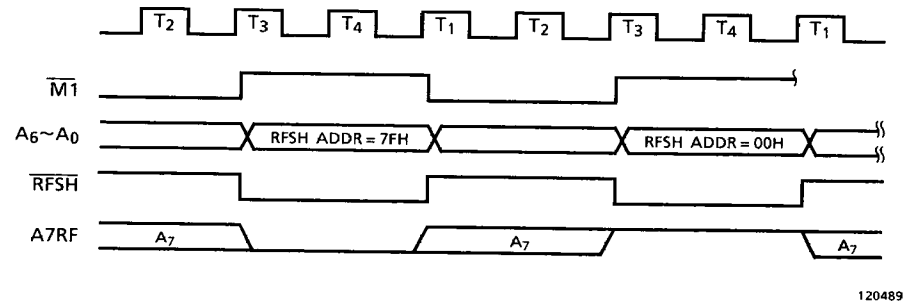


Figure 3.2.11 Refresh Timing

In clock state T4, the $\overline{MREQ}$ signal returns to "1". The refresh address is kept output until the rising edge of the clock state T1 in the next machine cycle, keeping the $\overline{RFSH}$ signal set to "0". The cycle delay state caused by setting the $\overline{WAIT}$ signal to "0" is the same in the memory read/write, input/output, and maskable interrupt acknowledge cycles. The diagram of the cycle delay state caused by the $\overline{WAIT}$ signal set to "0" is omitted in the following description.

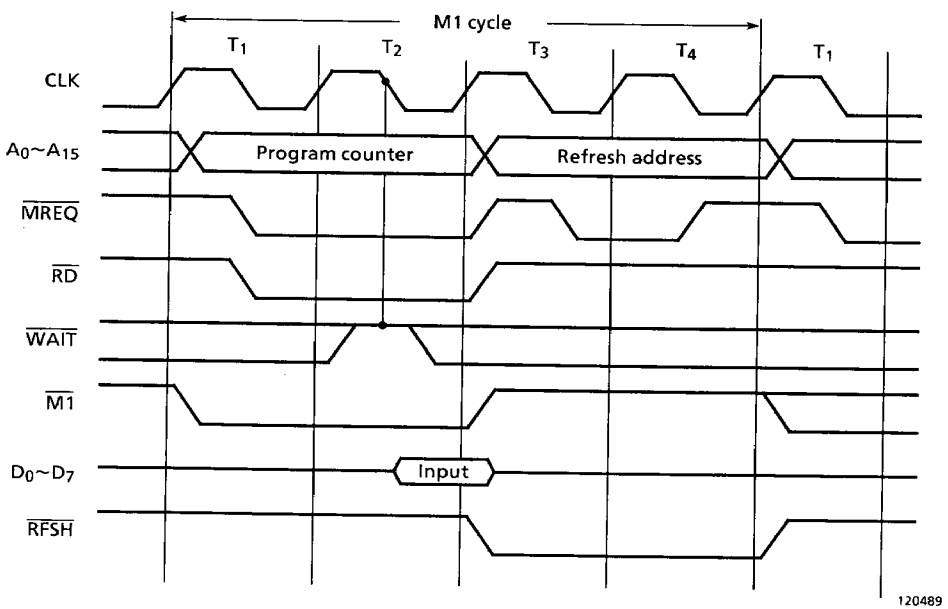


Figure 3.2.12 Opcode Fetch Timing

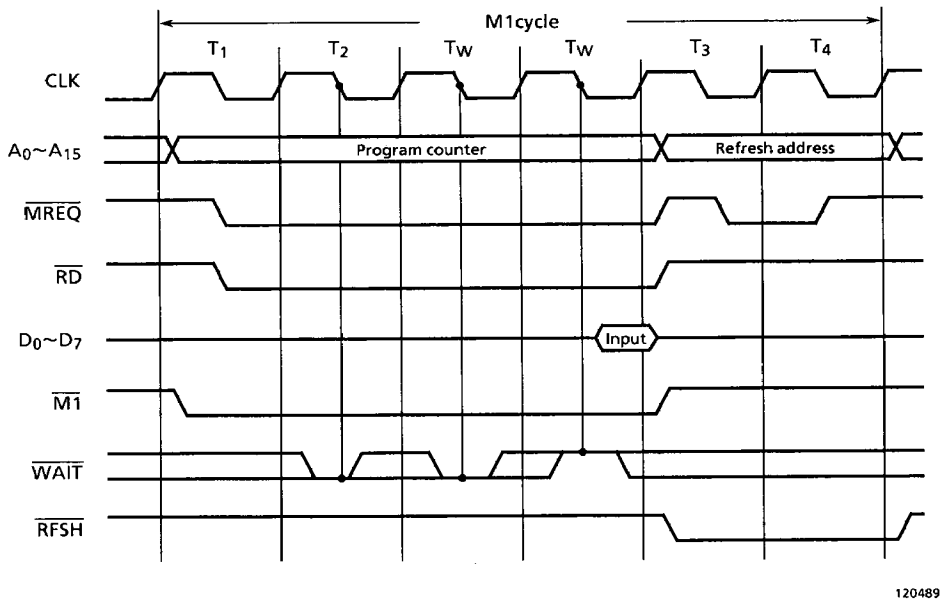


Figure 3.2.13 Opcode Fetch Timing Including Wait State

(2) Memory read/write operations

Figure 3.2.14 shows the basic timing of memory read/write operations (except for the Opcode fetch cycle) in the same diagram for convenience.

In each operation, the memory address signal to read/write data on the address bus is output in clock state T1. The operation in which the $\overline{WAIT}$ signal is sampled in clock state T2 and the following TW state is the same as the Opcode fetch cycle.

In memory read, memory data is put on the data bus by the address, $\overline{MREQ}$, and $\overline{RD}$ signals. The MPU reads this data.

In memory write, the memory address signal is put on the address bus then the $\overline{MREQ}$ signal is set to "0" to put the write data onto the data bus. When the data bus has been stabilized, the $\overline{WR}$ signal is output in clock state T2. The $\overline{WR}$ signal can be used as the memory write signal.

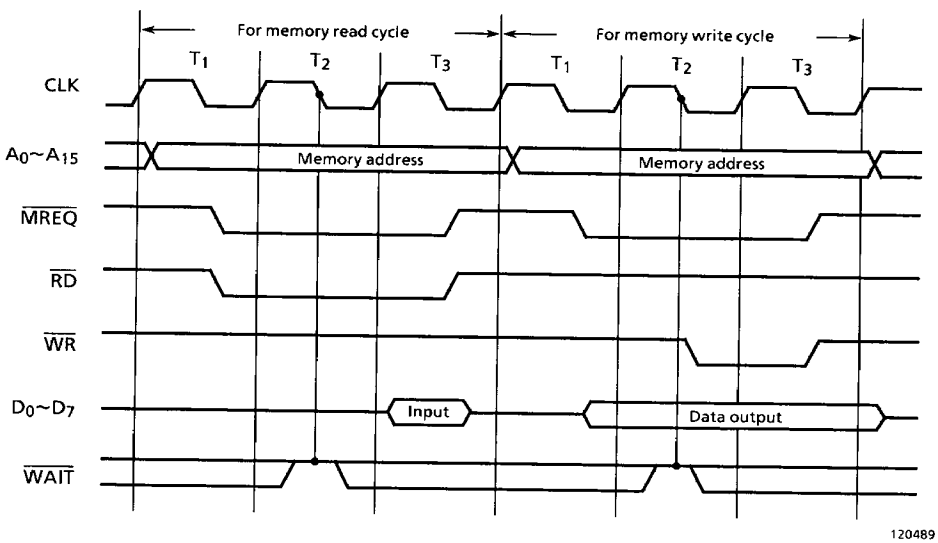


Figure 3.2.14 Memory Read/Write Cycle Timing

(3) Input/output operations

Figure 3.2.15 shows the basic timing of input/output operations. The feature of the I/O operation timing is that, regardless of the state of the $\overline{WAIT}$ signal in clock state T2, the I/O cycle automatically goes in the wait state (TW*) after clock T2. The $\overline{WAIT}$ signal is sampled on the falling edge of TW*. If the $\overline{WAIT}$ signal is "0" on the falling edges of TW* and the following clock state, the I/O operation enters into clock state TW*. Clock state TW* is inserted because the $\overline{IORQ}$ signal goes "0" in clock state T2, so that it is too late to sample the $\overline{WAIT}$ signal after decoding the I/O port address. In each of input and output operations, the I/O port address is put on the low-order 8 bits of the address bus in clock state T1. On the high-order 8 bits, the contents of the accumulator or B register are output. In clock state T2, the $\overline{IORQ}$ signal goes "0" instead of the $\overline{MREQ}$ signal. The $\overline{IORQ}$ signal can be used as the chip enable signal for a peripheral LSI.

In an input operation, the contents of the input port are read onto the data bus by the address, $\overline{IORQ}$, or $\overline{RD}$ signals. The MPU reads this data.

In an output operation, the output port address and the output data are respectively put on the address bus and data bus in clock state Tl, then the $\overline{IORQ}$ and $\overline{WR}$ signals go "0" in clock state T2. The $\overline{WR}$ signal can be used as the output port write signal.

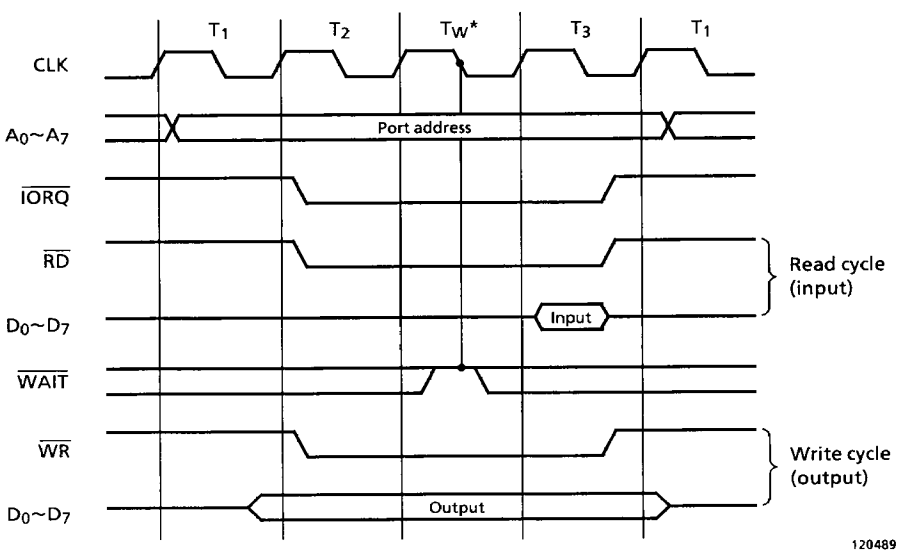


Figure 3.2.15 I/O Operating Timing

(4) Bus request and bus acknowledge operations

Figure 3.2.16 shows the basic timing of bus request and bus acknowledge operations.

The address bus (A0 through A15), data bus (D0 through D7), $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ signals controlled by the MPU can be put in the high-impedance state (floating) to electrically disconnect them from the MPU. This operation, after sampling the $\overline{BUSREQ}$ signal on the rising edge of the last clock of each machine cycle, starts on the rising edge of the next clock if this signal is found "0".

Subsequently, these buses are controlled by external peripheral LSIs. For example, data can be directly transferred between memory and these peripheral LSIs. This state is cleared if the $\overline{BUSREQ}$ signal is found "1" after sampling it on the rising edge of each subsequent clock state (TX), and enters into the next machine cycle. During the floating state, the $\overline{BUSACK}$ signal goes "0" to indicate it to the peripheral LSIs.

In this state, however, no memory refresh is performed and, therefore, the $\overline{RFSH}$ signal is set to "1". Hence, to maintain this state for a long time with a system using dynamic memory, memory refresh must be performed by the external controller.

Note that, in the floating state, neither maskable interrupt (INT) nor non-maskable interrupt (NMI) can be accepted.

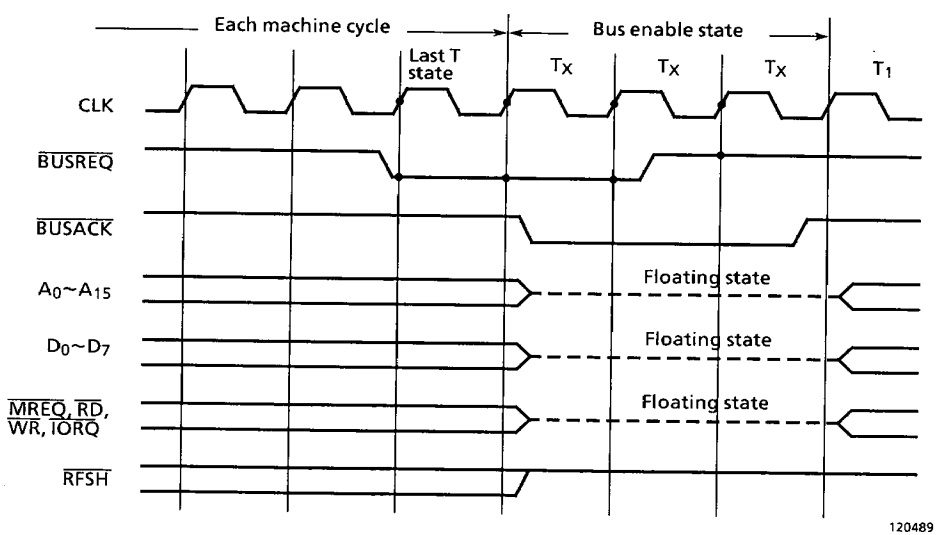


Figure 3.2.16 Bus Request and Bus Acknowledge Timing

(5) Maskable interrupt acknowledge operation

Figure 3.2.17 shows the basic timing of the maskable interrupt acknowledge.

The MPU samples the maskable interrupt request signal ($\overline{INT}$) on the rising edge of the last clock of each instruction execution. If the $\overline{INT}$ signal is found "0", a maskable interrupt is accepted except in the following cases:

- The interrupt enable flip-flop is reset to "0".
- The $\overline{BUSREQ}$ signal is "0".

When a maskable interrupt has been accepted, a special Opcode fetch cycle is generated. In this cycle, 2 clock states of wait state (TW*) is automatically inserted after the clock state T2. The $\overline{WAIT}$ signal is sampled on the falling edges of the second clock state TW* and the following clock state TW and, if the $\overline{WAIT}$ signal is found "0", the instruction cycle enters in the next clock state TW. In this Opcode fetch cycle, the $\overline{IORQ}$ signal goes "0" in the first TW* state instead of the $\overline{MREQ}$ signal while, in a normal Opcode fetch cycle, the $\overline{MREQ}$ signal goes "0" in clock state T1. This indicates to the maskable interrupt requesting LSI that the 8-bit interrupt vector can be put on the data bus. The MPU reads this data to perform interrupt processing. Therefore, the contents of the program counter put on the address bus are not used. Unlike an ordinary I/O operation, the $\overline{RD}$ signal does not go "0".

In clock state T3, the memory refresh address signal is put on the address bus for memory refresh like normal Opcode fetch cycle and the $\overline{RFSH}$ signal goes "0". In the subsequent machine cycles (M2 and M3) , the contents of the current program counter are saved into the stack. In machine cycles M4 and M5, the contents of the I register (the high-order 8 bits) and the contents of the address indicated by the address of the vector (the low-order 8 bits) from the CTC and the peripheral LSI are fetched in the program counter.

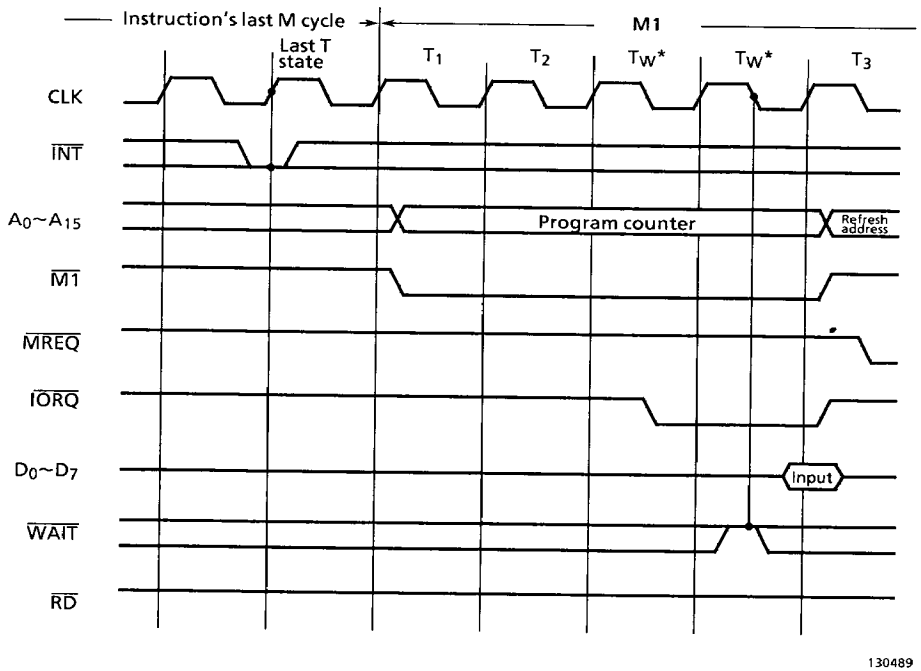


Figure 3.2.17 Maskable Interrupt Acknowledge Timing

(6) Non-maskable interrupt acknowledge operation

Figure 3.2.18 shows the basic timing of non-maskable interrupt acknowledge.

When the non-maskable interrupt request signal ($\overline{NMI}$) goes low, the internal non-maskable flip-flop is set to "1". The $\overline{NMI}$ signal is detected in any timing of each instruction. However, the internal NMI flip-flop is sampled on the rising edge of the last clock of each instruction. Therefore, the $\overline{NMI}$ signal should go low by the last clock state of an instruction.

The Opcode fetch cycle for non-maskable interrupt request acknowledge is generally the same as the ordinary Opcode fetch cycle. However, the Opcode on the data bus at the time is ignored. The contents of the current program counter are saved into the stack in the subsequent machine cycles (M2 and M3) . In the following machine cycle, the operation jumps to address 0066H, the non-maskable interrupt vector address. The machine cycles after these depend on the contents of the fetched Opcode.

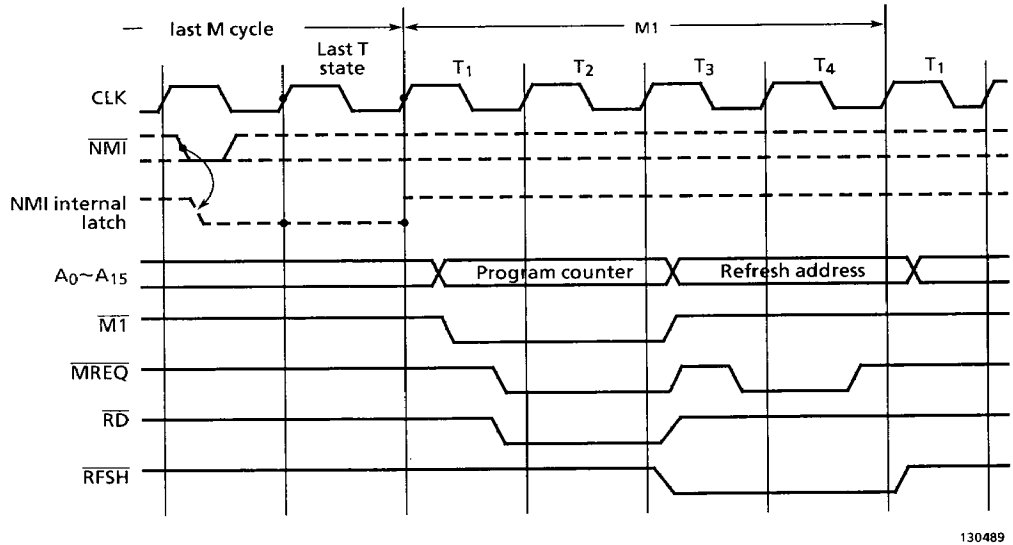


Figure 3.2.18 Non-Maaskable Inpterrupt Acknowledge Timing

(7) Halt operation

When a HALT instruction is fetched in the Opcode fetch cycle, the MPU sets the $\overline{\text{HALT}}$ signal to "0" synchronized with the falling edge of clock state T4 to indicate it to the peripheral LSI and stops operating. If the system clock is kept supplied in the halt state, the MPU continues executing NOP instructions. This is done to output refresh signals when the dynamic memory is used. The NOP instruction execution cycle is the same as the ordinary Opcode fetch cycle except the data on the data bus are ignored.

The halt state is cleared when an interrupt is accepted or the $\overline{\text{RESET}}$ signal is set to "0" to reset the MPU. Figure 3.2.19 shows the halt state clear operation by interrupt acknowledge. An interrupt is sampled on the rising edge of the last clock (clock state T4) of the NOP instruction. A maskable interrupt can be accepted when the $\overline{\text{INT}}$ signal is "0". A non-maskable interrupt is accepted when the internal NMI flip-flop which is set on the falling edge of the $\overline{\text{NMI}}$ signal is set at "1". However, it is required that the interrupt enable flip-flop is set to "1" for a maskable interrupt to be accepted. The interrupt processing for the accepted interrupt starts from the next cycle.

However, when the supply of the system clock from the CGC has been stopped by the power down operation, it is required to restart the supply of the system clock and input the $\overline{INT}$ signal until the execution of one instruction is completed or the $\overline{RESET}$ signal until 3 clocks are output. Figure 3.2.20 shows the timing of clearing the halt state caused by power down.

For the reset operation, see (8) "Reset operation". Note that the $\overline{INT}$ and $\overline{NMI}$ signals are shown on the same diagram in Figures 3.2.19 and 3.2.20 for convenience.

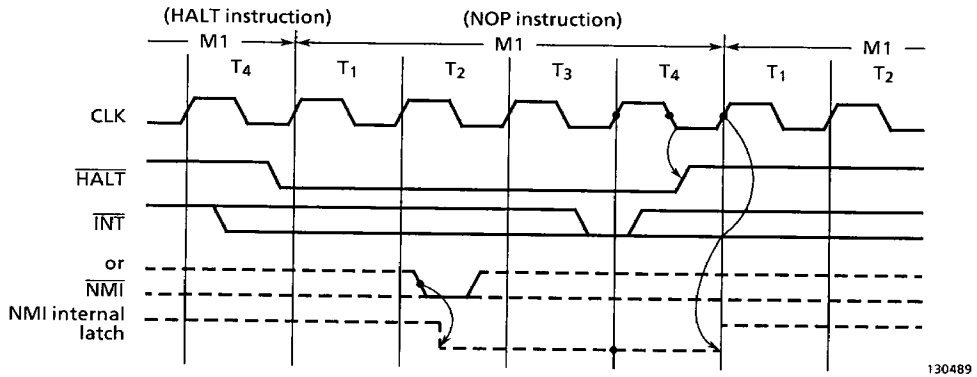


Figure 3.2.19 Timing of Clearing Halt State Caused by Interrupt Acknowledge

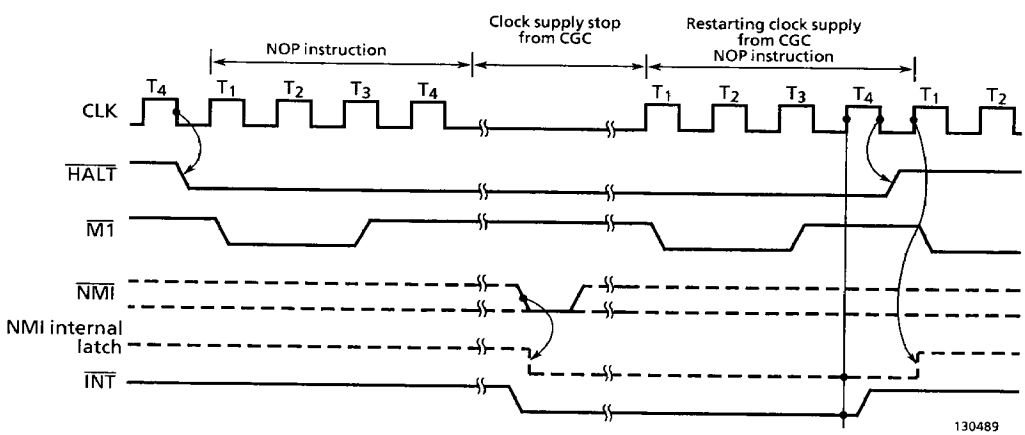


Figure 3.2.20 Timing of Clearing Halt State Caused by Power Down

(8) Reset operation

Figure 3.2.21 shows the basic timing of reset operation.

To reset the MPU, the $\overline{RESET}$ signal must be kept at "0" for at least 3 clocks. When the $\overline{RESET}$ signal goes "1", instruction execution starts from address 0000H after a dummy cycle of at least 2 clock states.

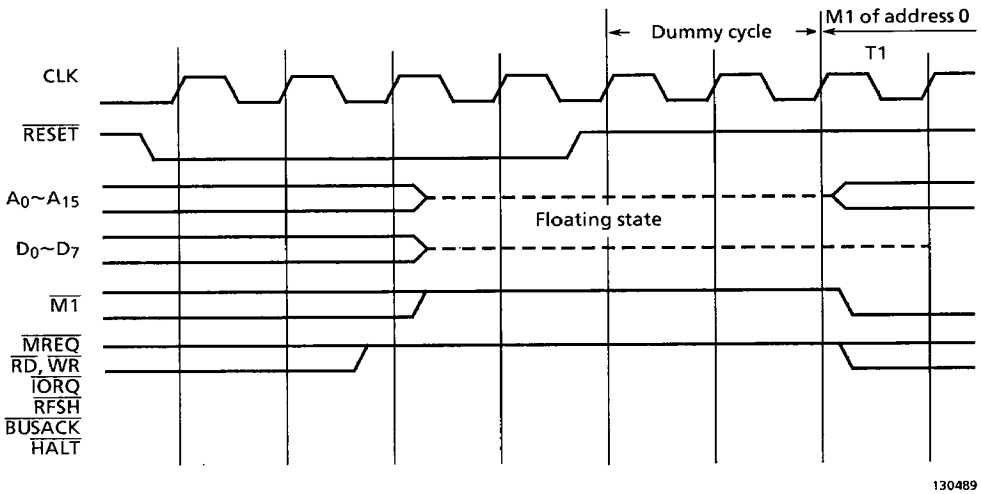


Figure 3.2.21 Reset Timing

To clear the power down state by using the $\overline{RESET}$ signal, the $\overline{RESET}$ signal must be input until 3 clocks or more are supplied by restarting the supply of the system clock from the CGC.

(9) Evaluation operation

Each of the MPU signals (A0 through A15, D0 through D7, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, $\overline{WR}$, $\overline{HALT}$, $\overline{M1}$, and $\overline{RFSH}$) can be put in the high-impedance state by EV and $\overline{BUSREQ}$ signals to electrically disconnect them from the MPU.

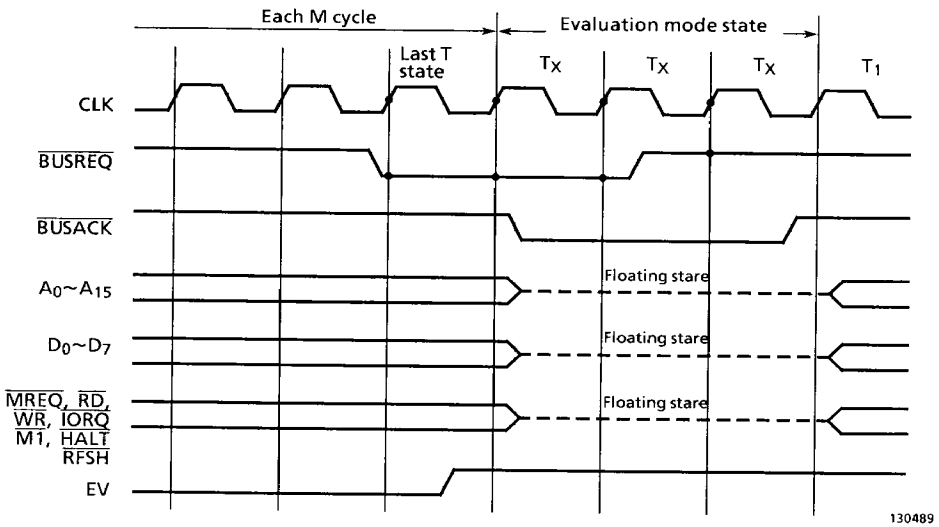


Figure 3.2.22 Evaluation Timing

Figure 3.2.23 shows the block diagram of the TMPZ84C013A operating as an evaluator in the evaluation mode.

The operations controlled by signals from the external MPU in the evaluation mode are the same as those of each device constituting the TMPZ84C013A. (However, for the watchdog timer operations, see "WDT Operational Description" because the watchdog timer is of random logic configuration.)

For the electrical characteristics and timing of each device, see "Inactive State".

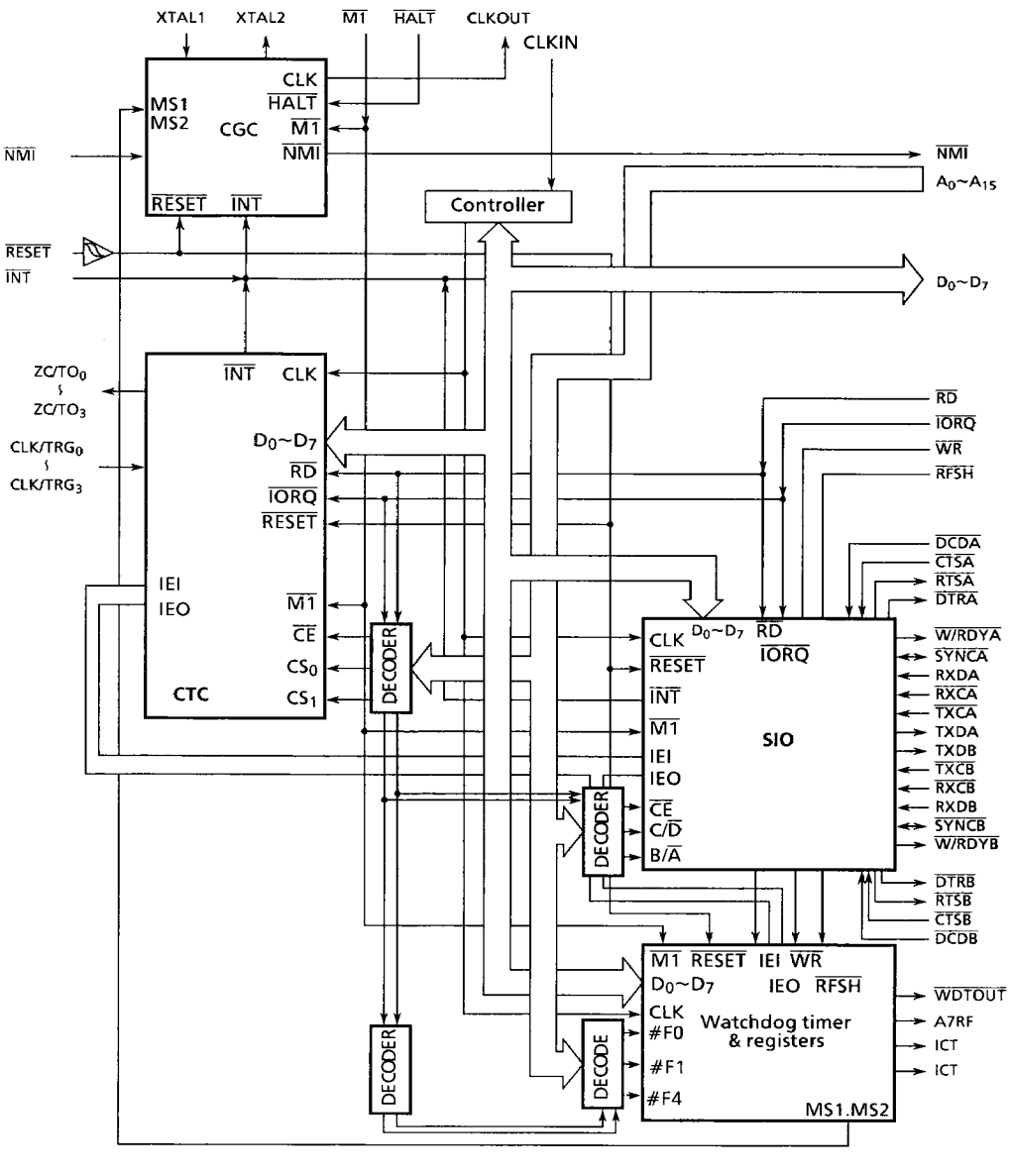


Figure 3.2.23 Block Diagram of the TMPZ84C013A Functioning As Evaluator

3.2.4 TMPZ84C013A Instruction Set

This subsection lists the TMPZ84C013A instruction codes and their functions. The table below lists the symbols and abbreviations used to describe the instruction set. The symbols which require special attention are described in the locations in which they appear.

- Symbols (1/2)

Classification	Symbol	Meaning	
Register	r, g	Register	B, C, D, E, H, L, A,
	t	Register pair	BC, DE, HL
		Stack pointer	SP
	q	Register pair	BC, DE, HL, AF
	p	Register pair	BC, DE
		Index register	IX
		Stack pointer	SP
	s	Register pair	BC, DE
		Index register	IY
		Stack pointer	SP
	t_H	Higher register of register pair (B, D, H)	
		Higher 8 bits of stack pointer (SP)	
	q_H	Higher register of register pair (B, D, H, A)	
	IX_H	Higher 8 bits of index register IX	
	IY_H	Higher 8 bits of index register IY	
	PC_H	Higher 8 bits of program counter (PC)	
	t_L	Lower register of register pair (C, E, L)	
		Lower 8 bits of stack pointer (SP)	
	q_L	Lower register of register pair (C, E, L, F)	
	IX_L	Lower 8 bits of index register IX	
	IY_L	Lower 8 bits of index register IY	
	PC_L	Lower 8 bits of program counter (PC)	
	rb	Bit b (0-7) of register (B, C, D, E, H, L, A)	

120489

- Symbols (2/2)

Classification	Symbol	Meaning
Memory	mn	Memory address represented in 16 bits. m indicates higher 8 bits and n, lower 8 bits.
	$(HL)_b$	Bit b (0-7) of the contents of the memory address indicated by register pair HL.
	$(IX+d)_b$	Bit b (0-7) of the contents of the memory address indicated by the value obtained by adding 8-bit data d to the content of index register IX.
	$(IY+d)_b$	Bit b (0-7) of the contents of the memory address indicated by the value obtained by adding 8-bit data d to the content of index register IY.
Flag change symbol	0	Reset to "0" by operation.
	1	Set to "1" by operation.
	-	No change
	*	Affected by operation
	X	Undefined
	P	Handled as parity flag. P = 0: odd parity P = 1: even parity
	V	Handled as overflow flag. V = 0: No overflow V = 1: Overflow
Operator	←	Transfer
	↔	Exchange
	+	Add
	−	Subtract
	∧	Logical and between bits.
	∨	Logical or between bits.
	⊕	Exclusive or between bits
Others	IFF	Interrupt enable flip-flop
	CY	Carry flag
	Z	Zero flag

120489

TMPZ84C013A Instruction Set (1/9)

ITEM/ CLASSIFICATION	Assembler mnemonic	Object code Binary (76 543 210)	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CYCLES	No. OF STATES
8-BIT DATA LOAD	LD r,g	01 rrr ggg	40+r×8+g	r←g	–	–	X	–	X	–	–	–	1	4
	LD r,n	00 rrr 110 nn nnn nnn	06+r×8 n	r←n	–	–	X	–	X	–	–	–	2	7
	LD r,(HL)	01 rrr 110	46+r×8	r←(HL)	–	–	X	–	X	–	–	–	2	7
	LD r,(IX+d)	11 011 101 01 rrr 110 dd ddd ddd	DD 46+r×8 d	r←(IX+d)	–	–	X	–	X	–	–	–	5	19
	LD r,(IY+d)	11 111 101 01 rrr 110 dd ddd ddd	FD 46+r×8 d	r←(IY+d)	–	–	X	–	X	–	–	–	5	19
	LD (HL),r	01 110 rrr	70+r	(HL)←r	–	–	X	–	X	–	–	–	2	7
	LD (IX+d),r	11 011 101 01 110 rrr dd ddd ddd	DD 70+r d	(IX+d)←r	–	–	X	–	X	–	–	–	5	19
	LD (IY+d),r	11 111 101 01 110 rrr dd ddd ddd	FD 70+r d	(IY+d)←r	–	–	X	–	X	–	–	–	5	19
	LD (HL),n	00 110 110 nn nnn nnn	36 n	(HL)←n	–	–	X	–	X	–	–	–	3	10
	LD (IX+d), n	11 011 101 00 110 110 dd ddd ddd nn nnn nnn	DD 36 d n	(IX+d)←n	–	–	X	–	X	–	–	–	5	19
	LD (IY+d),n	11 111 101 00 110 110 dd ddd ddd nn nnn nnn	FD 36 d n	(IY+d)←n	–	–	X	–	X	–	–	–	5	19
	LD A,(BC)	00 001 010	0A	A←(BC)	–	–	X	–	X	–	–	–	2	7
	LD A,(DE)	00 011 010	1A	A←(DE)	–	–	X	–	X	–	–	–	2	7
	LD A,(mn)	00 111 010 nn nnn nnn mm mmm mmm	3A n m	A←(mn)	–	–	X	–	X	–	–	–	4	13
	LD (BC),A	00 000 010	02	(BC)←A	–	–	X	–	X	–	–	–	2	7
	LD (DE),A	00 010 010	12	(DE)←A	–	–	X	–	X	–	–	–	2	7
	LD (mn),A	00 110 010 nn nnn nnn mm mmm mmm	32 n m	(mn)←A	–	–	X	–	X	–	–	–	4	13
	LD A,I	11 101 101 01 010 111	ED 57	A←I	*	*	X	0	X	IFF	0	–	2	9
	LD A,R	11 101 101 01 011 111	ED 5F	A←R	*	*	X	0	X	IFF	0	–	2	9
	LD I,A	11 101 101 01 000 111	ED 47	I←A	–	–	X	–	X	–	–	–	2	9
	LD R,A	11 101 101 01 001 111	ED 4F	R←A	–	–	X	–	X	–	–	–	2	9
16-BIT DATA LOAD	LD t,mn	00 tt0 001 nn nnn nnn mm mmm mmm	01+t×10 n m	t←mn	–	–	X	–	X	–	–	–	3	10
	LD IX,mn	11 011 101 00 100 001 nn nnn nnn mm mmm mmm	DD 21 n m	IX←mn	–	–	X	–	X	–	–	–	4	14

r	rrr
g	ggg
B	000
C	001
D	010
E	011
H	100
L	101
A	111

t	tt
BC	00
DE	01
HL	10
SP	11

Note : r,g means any of the registers A, B, C, D, E, H, L.
IFF in "Flag" column indicates that the content of the interrupt enable flip-flop is copied into the P/V flag.

120489

TMPZ84C013A Instruction Set (2/9)

ITEM/ CLASSI -FICA- TION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CY- CLES	No. OF STA- TES
16-BIT DATA LOAD	LD IY,mn	11 111 101 00 100 001 nn nnn nnn mm mmm mmm	FD 21 n m	IY←mn	–	–	X	–	X	–	–	–	4	14
	LD HL,(mn)	00 101 010 nn nnn nnn mm mmm mmm	2A n m	H←(mn+1) L←(mn)	–	–	X	–	X	–	–	–	5	16
	LD t,(mn)	11 101 101 01 tt1 011 nn nnn nnn mm mmm mmm	ED 4B+t×10 n m	t_H←(mn+1) t_L←(mn)	–	–	X	–	X	–	–	–	6	20
	LD IX,(mn)	11 011 101 00 101 010 nn nnn nnn mm mmm mmm	DD 2A n m	IX_H←(mn+1) IX_L←(mn)	–	–	X	–	X	–	–	–	6	20
	LD IY,(mn)	11 111 101 00 101 010 nn nnn nnn mm mmm mmm	FD 2A n m	IY_H←(mn+1) IY_L←(mn)	–	–	X	–	X	–	–	–	6	20
	LD (mn),HL	00 100 010 nn nnn nnn mm mmm mmm	22 n m	(mn+1)←H (mn)←L	–	–	X	–	X	–	–	–	5	16
	LD (mn),t	11 101 101 01 tt0 011 nn nnn nnn mm mmm mmm	ED 43+t×10 n m	(mn+1)←t_H (mn)←t_L	–	–	X	–	X	–	–	–	6	20
	LD (mn),IX	11 011 101 00 100 010 nn nnn nnn mm mmm mmm	DD 22 n m	(mn+1)←IX_H (mn)←IX_L	–	–	X	–	X	–	–	–	6	20
	LD (mn),IY	11 111 101 00 100 010 nn nnn nnn mm mmm mmm	FD 22 n m	(mn+1)←IY_H (mn)←IY_L	–	–	X	–	X	–	–	–	6	20
	LD SP,HL	11 111 001	F9	SP←HL	–	–	X	–	X	–	–	–	1	6
	LD SP,IX	11 011 101 11 111 001	DD F9	SP←IX	–	–	X	–	X	–	–	–	2	10
	LD SP,IY	11 111 101 11 111 001	FD F9	SP←IY	–	–	X	–	X	–	–	–	2	10
	PUSH q	11 qq0 101	C5+q×10	(SP−2)←q_L,(SP−1)←q_H, SP←SP−2	–	–	X	–	X	–	–	–	3	11
	PUSH IX	11 011 101 11 100 101	DD E5	(SP−2)←IX_L,(SP−1)←IX_H SP←SP−2	–	–	X	–	X	–	–	–	4	15
	PUSH IY	11 111 101 11 100 101	FD E5	(SP−2)←IY_L,(SP−1)←IY_H SP←SP−2	–	–	X	–	X	–	–	–	4	15
	POP q	11 qq0 001	C1+q×10	q_H←(SP+1),q_L←(SP), SP←SP+2	–	–	X	–	X	–	–	–	3	10
	POP IX	11 011 101 11 100 001	DD E1	IX_H←(SP+1),IX_L←(SP) SP←SP+2	–	–	X	–	X	–	–	–	4	14
	POP IY	11 111 101 11 100 001	FD E1	IY_H←(SP+1),IY_L←(SP) SP←SP+2	–	–	X	–	X	–	–	–	4	14
*1	EX DE,HL	11 101 011	EB	DE↔HL	–	–	X	–	X	–	–	–	1	4
	EX AF,AF'	00 001 000	08	AF↔AF'	–	–	X	–	X	–	–	–	1	4
	EXX	11 011 001	D9	BC↔BC',DE↔DE',HL↔HL'	–	–	X	–	X	–	–	–	1	4

t	tt
BC	00
DE	01
HL	10
SP	11

q	qq
BC	00
DE	01
HL	10
AF	11

Note : t is any of the register pairs BC, DE, HL, SP.
q is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively. (Ex) $BC_L = C$, $AF_H = A$.

*1 : EXCHANGE

120489

TMPZ84C013A Instruction Set (3/9)

ITEM/ CLASSI -FICA- TION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CY- CLES	No. OF STA- TES	
EXCHANGE	EX (SP),HL	11 100 011	E3	H↔(SP+1),L↔(SP)	–	–	X	–	X	–	–	–	5	19	
	EX (SP),IX	11 011 101 11 100 011	DD E3	IX_H↔(SP+1) IX_L↔(SP)	–	–	X	–	X	–	–	–	6	23	
	EX (SP),IY	11 111 101 11 100 011	FD E3	IY_H↔(SP+1) IY_L↔(SP)	–	–	X	–	X	–	–	–	6	23	
BLOCK TRANSFER BLOCK SEARCH	LDI	11 101 101 10 100 000	ED A0	(DE)←(HL),DE←DE+1 HL←HL+1,BC←BC−1	–	–	X	0	X	*M	0	–	4	16	
	LDIR	11 101 101 10 110 000	ED B0	(DE)←(HL),DE←DE+1 HL←HL+1,BC←BC−1 Repeat until BC=0	–	–	X	0	X	0	0	–	5 4	21 16	←[BC< >0] ←[BC=0]
	LDD	11 101 101 10 101 000	ED A8	(DE)←(HL),DE←DE−1 HL←HL−1,BC←BC−1	–	–	X	0	X	*M	0	–	4	16	
	LDDR	11 101 101 10 111 000	ED B8	(DE)←(HL),DE←DE−1 HL←HL−1,BC←BC−1 Repeat until BC=0	–	–	X	0	X	0	0	–	5 4	21 16	←[BC< >0] ←[BC=0]
	CPI	11 101 101 10 100 001	ED A1	A−(HL) HL←HL+1,BC←BC−1	*	*N	X	*	X	*M	1	–	4	16	
	CPIR	11 101 101 10 110 001	ED B1	A−(HL),HL←HL+1,BC←BC−1 Repeat until A=(HL) or BC=0	*	*N	X	*	X	*M	1	–	5 4	21 16	←[BC< >0 & A< > (HL)] ↖[BC=0 or A= (HL)]
	CPD	11 101 101 10 101 001	ED A9	A−(HL) HL←HL−1,BC←BC−1	*	*N	X	*	X	*M	1	–	4	16	
	CPDR	11 101 101 10 111 001	ED B9	A−(HL),HL←HL−1,BC←BC−1 Repeat until A=(HL) or BC=0	*	*N	X	*	X	*M	1	–	5 4	21 16	←[BC< >0 & A< > (HL)] ↖[BC=0 or A= (HL)]
8-BIT ARITHMETIC AND LOGICAL	ADD A,r	10 000 rrr	80+r	A←A+r	*	*	X	*	X	V	0	*	1	4	
	ADD A,n	11 000 110 nn nnn nnn	C6 n	A←A+n	*	*	X	*	X	V	0	*	2	7	
	ADD A,(HL)	10 000 110	86	A←A+(HL)	*	*	X	*	X	V	0	*	2	7	
	ADD A,(IX+d)	11 011 101 10 000 110 dd ddd ddd	DD 86 d	A←A+(IX+d)	*	*	X	*	X	V	0	*	5	19	
	ADD A,(IY+d)	11 111 101 10 000 110 dd ddd ddd	FD 86 d	A←A+(IY+d)	*	*	X	*	X	V	0	*	5	19	
	ADC A,r	10 001 rrr	88+r	A←A+r+CY	*	*	X	*	X	V	0	*	1	4	
	ADC A,n	11 001 110 nn nnn nnn	CE n	A←A+n+CY	*	*	X	*	X	V	0	*	2	7	
	ADC A,(HL)	10 001 110	8E	A←A+(HL)+CY	*	*	X	*	X	V	0	*	2	7	
	ADC A,(IX+d)	11 011 101 10 001 110 dd ddd ddd	DD 8E d	A←A+(IX+d)+CY	*	*	X	*	X	V	0	*	5	19	
	ADC A,(IY+d)	11 111 101 10 001 110 dd ddd ddd	FD 8E d	A←A+(IY+d)+CY	*	*	X	*	X	V	0	*	5	19	
	SUB r	10 010 rrr	90+r	A←A−r	*	*	X	*	X	V	1	*	1	4	
	SUB n	11 010 110 nn nnn nnn	D6 n	A←A−n	*	*	X	*	X	V	1	*	2	7	
	SUB (HL)	10 010 110	96	A←A−(HL)	*	*	X	*	X	V	1	*	2	7	
	SUB (IX+d)	11 011 101 10 010 110 dd ddd ddd	DD 96 d	A←A−(IX+d)	*	*	X	*	X	V	1	*	5	19	
	SUB (IY+d)	11 111 101 10 010 110 dd ddd ddd	FD 96 d	A←A−(IY+d)	*	*	X	*	X	V	1	*	5	19	

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Note : *M P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1.
*N Z flag is 1 if A = (HL), otherwise Z = 0.
[ ]indicates the total condition of the number of cycles and states indicated by arrow.
r means any of the registers A, B, C, D, E, H, L.

120489

TMPZ84C013A Instruction Set (4/9)

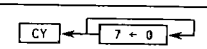
ITEM/ CLASSI -FICA- TION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CY- CLES	No. OF STA- TES
8-BIT ARITHMETIC AND LOGICAL	SBC A,r	10 011 rrr	98+r	A←A−r−CY	*	*	X	*	X	V	1	*	1	4
	SBC A,n	11 011 110 nn nnn nnn	DE n	A←A−n−CY	*	*	X	*	X	V	1	*	2	7
	SBC A,(HL)	10 011 110	9E	A←A−(HL)−CY	*	*	X	*	X	V	1	*	2	7
	SBC A,(IX+d)	11 011 101 10 011 110 dd ddd ddd	DD 9E d	A←A−(IX+d)−CY	*	*	X	*	X	V	1	*	5	19
	SBC A,(IY+d)	11 111 101 10 011 110 dd ddd ddd	FD 9E d	A←A−(IY+d)−CY	*	*	X	*	X	V	1	*	5	19
	AND r	10 100 rrr	A0+r	A←A∧r	*	*	X	1	X	P	0	0	1	4
	AND n	11 100 110 nn nnn nnn	E6 n	A←A∧n	*	*	X	1	X	P	0	0	2	7
	AND (HL)	10 100 110	A6	A←A∧(HL)	*	*	X	1	X	P	0	0	2	7
	AND (IX+d)	11 011 101 10 100 110 dd ddd ddd	DD A6 d	A←A∧(IX+d)	*	*	X	1	X	P	0	0	5	19
	AND (IY+d)	11 111 101 10 100 110 dd ddd ddd	FD A6 d	A←A∧(IY+d)	*	*	X	1	X	P	0	0	5	19
	OR r	10 110 rrr	B0+r	A←A∨r	*	*	X	0	X	P	0	0	1	4
	OR n	11 110 110 nn nnn nnn	F6 n	A←A∨n	*	*	X	0	X	P	0	0	2	7
	OR (HL)	10 110 110	B6	A←A∨(HL)	*	*	X	0	X	P	0	0	2	7
	OR (IX+d)	11 011 101 10 110 110 dd ddd ddd	DD B6 d	A←A∨(IX+d)	*	*	X	0	X	P	0	0	5	19
	OR (IY+d)	11 111 101 10 110 110 dd ddd ddd	FD B6 d	A←A∨(IY+d)	*	*	X	0	X	P	0	0	5	19
	XOR r	10 101 rrr	A8+r	A←A⊻r	*	*	X	0	X	P	0	0	1	4
	XOR n	11 101 110 nn nnn nnn	EE n	A←A⊻n	*	*	X	0	X	P	0	0	2	7
	XOR (HL)	10 101 110	AE	A←A⊻(HL)	*	*	X	0	X	P	0	0	2	7
	XOR (IX+d)	11 011 101 10 101 110 dd ddd ddd	DD AE d	A←A⊻(IX+d)	*	*	X	0	X	P	0	0	5	19
	XOR (IY+d)	11 111 101 10 101 110 dd ddd ddd	FD AE d	A←A⊻(IY+d)	*	*	X	0	X	P	0	0	5	19
	CP r	10 111 rrr	B8+r	A−r	*	*	X	*	X	V	1	*	1	4
	CP n	11 111 110 nn nnn nnn	FE n	A−n	*	*	X	*	X	V	1	*	2	7
	CP (HL)	10 111 110	BE	A−(HL)	*	*	X	*	X	V	1	*	2	7
	CP (IX+d)	11 011 101 10 111 110 dd ddd ddd	DD BE d	A−(IX+d)	*	*	X	*	X	V	1	*	5	19
	CP (IY+d)	11 111 101 10 111 110 dd ddd ddd	FD BE d	A−(IY+d)	*	*	X	*	X	V	1	*	5	19
	INC r	00 rrr 100	04+r×8	r←r+1	*	*	X	*	X	V	0	−	1	4
	INC (HL)	00 110 100	34	(HL)←(HL)+1	*	*	X	*	X	V	0	−	3	11
	INC (IX+d)	11 011 101 00 110 100 dd ddd ddd	DD 34 d	(IX+d)←(IX+d)+1	*	*	X	*	X	V	0	−	6	23

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Note : r means any of the registers A, B, C, D, E, H, L.

120489

TMPZ84C013A Instruction Set (5/9)

ITEM/CLASSIFICATION	Assembler mnemonic	Object code Binary	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CY-CLES	No. OF STATES
8-BIT ARITHMETIC AND LOGICAL	INC (IY+d)	11 111 101 00 110 100 dd ddd ddd	FD 34 d	(IY+d)←(IY+d)+1	•	•	X	•	X	V	0	–	6	23
	DEC r	00 rrr 101	05+r×8	r←r−1	•	•	X	•	X	V	1	–	1	4
	DEC (HL)	00 110 101	35	(HL)←(HL)−1	•	•	X	•	X	V	1	–	3	11
	DEC (IX+d)	11 011 101 00 110 101 dd ddd ddd	DD 35 d	(IX+d)←(IX+d)−1	•	•	X	•	X	V	1	–	6	23
	DEC (IY+d)	11 111 101 00 110 101 dd ddd ddd	FD 35 d	(IY+d)←(IY+d)−1	•	•	X	•	X	V	1	–	6	23
GENELAL-PURPOSE ARITHMETIC AND MPU CONTROL	DAA	00 100 111	27	Decimal adjust accumulator	•	•	X	•	X	P	–	•	1	4
	CPL	00 101 111	2F	A←$\overline{A}$	–	–	X	1	X	–	1	–	1	4
	NEG	11 101 101 01 000 100	ED 44	A←0−A	•	•	X	•	X	V	1	•	2	8
	CCF	00 111 111	3F	CY←$\overline{CY}$	–	–	X	X	X	–	0	•	1	4
	SCF	00 110 111	37	CY←1	–	–	X	0	X	–	0	1	1	4
	NOP	00 000 000	00	no operation	–	–	X	–	X	–	–	–	1	4
	HALT	01 110 110	76	MPU Halted	–	–	X	–	X	–	–	–	1	4
	DI	11 110 011	F3	IFF←0	–	–	X	–	X	–	–	–	1	4
	EI	11 111 011	FB	IFF←1	–	–	X	–	X	–	–	–	1	4
	IM 0	11 101 101 01 000 110	ED 46	Set interrupt mode 0	–	–	X	–	X	–	–	–	2	8
	IM 1	11 101 101 01 010 110	ED 56	Set interrupt mode 1	–	–	X	–	X	–	–	–	2	8
	IM 2	11 101 101 01 011 110	ED 5E	Set interrupt mode 2	–	–	X	–	X	–	–	–	2	8
16-BIT ARITHMETIC	ADD HL,t	00 tt1 001	09+t×10	HL←HL+t	–	–	X	X	X	–	0	•	3	11
	ADC HL,t	11 101 101 01 tt1 010	ED 4A+t×10	HL←HL+t+CY	•	•	X	X	X	V	0	•	4	15
	SBC HL,t	11 101 101 01 tt0 010	ED 42+t×10	HL←HL−t−CY	•	•	X	X	X	V	1	•	4	15
	ADD IX,p	11 011 101 00 pp1 001	DD 09+p×10	IX←IX+p	–	–	X	X	X	–	0	•	4	15
	ADD IY,s	11 111 101 00 ss1 001	FD 09+s×10	IY←IY+s	–	–	X	X	X	–	0	•	4	15
	INC t	00 tt0 011	03+t×10	t←t+1	–	–	X	–	X	–	–	–	1	6
	INC IX	11 011 101 00 100 011	DD 23	IX←IX+1	–	–	X	–	X	–	–	–	2	10
	INC IY	11 111 101 00 100 011	FD 23	IY←IY+1	–	–	X	–	X	–	–	–	2	10
	DEC t	00 tt1 011	0B+t×10	t←t−1	–	–	X	–	X	–	–	–	1	6
	DEC IX	11 011 101 00 101 011	DD 2B	IX←IX−1	–	–	X	–	X	–	–	–	2	10
	DEC IY	11 111 101 00 101 011	FD 2B	IY←IY−1	–	–	X	–	X	–	–	–	2	10
ROTATE	RLCA	00 000 111	07		–	–	X	0	X	–	0	•	1	4

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

t	tt
BC	00
DE	01
HL	10
SP	11

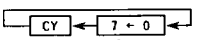
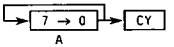
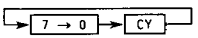
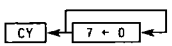
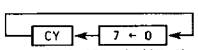
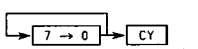
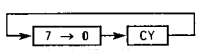
p	pp
BC	00
DE	01
IX	10
SP	11

s	ss
BC	00
DE	01
IY	10
SP	11

Note : ss is any of the register pairs BC, DE, HL, SP. PP is any of the register pairs BC, DE, IX, SP.
rr is any of the register pairs BC, DE, IY, SP.

120489

TMPZ84C013A Instruction Set (6/9)

ITEM/ CLASSIFICATION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CYCLES	No. OF STATES
ROTATE SHIFT	RLA	00 010 111	17		–	–	X	0	X	–	0	*	1	4
	RRCA	00 001 111	0F		–	–	X	0	X	–	0	*	1	4
	RRA	00 011 111	1F		–	–	X	0	X	–	0	*	1	4
	RLC r	11 001 011 00 000 rrr	CB 00+r		*	*	X	0	X	P	0	*	2	8
	RLC (HL)	11 001 011 00 000 110	CB 06		*	*	X	0	X	P	0	*	4	15
	RLC (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 000 110	DD CB d 06		*	*	X	0	X	P	0	*	6	23
	RLC (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 000 110	FD CB d 06		*	*	X	0	X	P	0	*	6	23
	RL r	11 001 011 00 010 rrr	CB 10+r		*	*	X	0	X	P	0	*	2	8
	RL (HL)	11 001 011 00 010 110	CB 16		*	*	X	0	X	P	0	*	4	15
	RL (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 010 110	DD CB d 16		*	*	X	0	X	P	0	*	6	23
	RL (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 010 110	FD CB d 16		*	*	X	0	X	P	0	*	6	23
	RRC r	11 001 011 00 001 rrr	CB 08+r		*	*	X	0	X	P	0	*	2	8
	RRC (HL)	11 001 011 00 001 110	CB 0E		*	*	X	0	X	P	0	*	4	15
	RRC (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 001 110	DD CB d 0E		*	*	X	0	X	P	0	*	6	23
	RRC (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 001 110	FD CB d 0E		*	*	X	0	X	P	0	*	6	23
	RR r	11 001 011 00 011 rrr	CB 18+r		*	*	X	0	X	P	0	*	2	8
	RR (HL)	11 001 011 00 011 110	CB 1E		*	*	X	0	X	P	0	*	4	15
	RR (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 011 110	DD CB d 1E		*	*	X	0	X	P	0	*	6	23

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Note : r means any of the registers A, B, C, D, E, H, L.

120489

TMPZ84C013A Instruction Set (7/9)

ITEM/ CLASSI-FICATION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z	H	P/V	N	C	No. OF CYCLES	No. OF STATES
ROTATE SHIFT	RR (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 011 110	FD CB d 1E	7 → 0 → CY (IY+d)	*	*	X	0	X	P	0	*	6	23
	SLA r	11 001 011 00 100 rrr	CB 20+r	CY ← 7 ← 0 ← 0 r,(HL),(IX+d),(IY+d)	*	*	X	0	X	P	0	*	2	8
	SLA (HL)	11 001 011 00 100 110	CB 26		*	*	X	0	X	P	0	*	4	15
	SLA (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 100 110	DD CB d 26		*	*	X	0	X	P	0	*	6	23
	SLA (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 100 110	FD CB d 26		*	*	X	0	X	P	0	*	6	23
	SRA r	11 001 011 00 101 rrr	CB 28+r	7 → 0 → CY r,(HL),(IX+d),(IY+d)	*	*	X	0	X	P	0	*	2	8
	SRA (HL)	11 001 011 00 101 110	CB 2E		*	*	X	0	X	P	0	*	4	15
	SRA (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 101 110	DD CB d 2E		*	*	X	0	X	P	0	*	6	23
	SRA (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 101 110	FD CB d 2E		*	*	X	0	X	P	0	*	6	23
	SRL r	11 001 011 00 111 rrr	CB 38+r	0 → 7 → 0 → CY r,(HL),(IX+d),(IY+d)	*	*	X	0	X	P	0	*	2	8
	SRL (HL)	11 001 011 00 111 110	CB 3E		*	*	X	0	X	P	0	*	4	15
	SRL (IX+d)	11 011 101 11 001 011 dd ddd ddd 00 111 110	DD CB d 3E		*	*	X	0	X	P	0	*	6	23
	SRL (IY+d)	11 111 101 11 001 011 dd ddd ddd 00 111 110	FD CB d 3E		*	*	X	0	X	P	0	*	6	23
	RLD	11 101 101 01 101 111	ED 6F	A 7 4 3 0 ⇄ 7 4 3 0 (HL)	*	*	X	0	X	P	0	–	5	18
	RRD	11 101 101 01 100 111	ED 67	A 7 4 3 0 ⇄ 7 4 3 0 (HL)	*	*	X	0	X	P	0	–	5	18
BIT SET RESET AND TEST	BIT b,r	11 001 011 01 bbb rrr	CB 40+b×8+r	$Z \leftarrow \overline{r_b}$	X	*	X	1	X	X	0	–	2	8
	BIT b,(HL)	11 001 011 01 bbb 110	CB 46+b×8	$Z \leftarrow \overline{(HL)_b}$	X	*	X	1	X	X	0	–	3	12

*1

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

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

Note : *1 : Rotate digit left and right between the accmulator and location (HL).
The content of the upper half of the accumulator is unaffected.
The notation $(HL)_b$ indicates bit $_b$ (0 to 7) within the contents of the HL register pair.
The notation r_b indicates bit $_b$ (0 to 7) within the r register.

120489

TMPZ84C013A Instruction Set (8/9)

ITEM/ CLASSI -FICA- TION	Assembler mnemonic	Object code Binary 76 543 210	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CY- CLES	No. OF STA- TES
BIT SET RESET AND TEST	BIT b,(IX+d)	11 011 101	DD		X	*	X	1	X	X	0	–	5	20
		11 001 011	CB	$Z \leftarrow \overline{(IX+d)_b}$										
		dd ddd ddd	d											
		01 bbb 110	46+b×8											
	BIT b,(IY+d)	11 111 101	FD		X	*	X	1	X	X	0	–	5	20
		11 001 011	CB	$Z \leftarrow \overline{(IY+d)_b}$										
		dd ddd ddd	d											
		01 bbb 110	46+b×8											
	SET b,r	11 001 011	CB	$r_b \leftarrow 1$	–	–	X	–	X	–	–	–	2	8
		11 bbb rrr	C0+b×8+r											
	SET b,(HL)	11 001 011	CB	$(HL)_b \leftarrow 1$	–	–	X	–	X	–	–	–	4	15
		11 bbb 110	C6+b×8											
	SET b,(IX+d)	11 011 101	DD	$(IX+d)_b \leftarrow 1$	–	–	X	–	X	–	–	–	6	23
		11 001 011	CB											
		dd ddd ddd	d											
		11 bbb 110	C6+b×8											
	SET b,(IY+d)	11 111 101	FD	$(IY+d)_b \leftarrow 1$	–	–	X	–	X	–	–	–	6	23
		11 001 011	CB											
		dd ddd ddd	d											
		11 bbb 110	C6+b×8											
	RES b,r	11 001 011	CB	$r_b \leftarrow 0$	–	–	X	–	X	–	–	–	2	8
		10 bbb rrr	80+b×8+r											
	RES b,(HL)	11 001 011	CB	$(HL)_b \leftarrow 0$	–	–	X	–	X	–	–	–	4	15
		10 bbb 110	86+b×8											
	RES b,(IX+d)	11 011 101	DD	$(IX+d)_b \leftarrow 0$	–	–	X	–	X	–	–	–	6	23
		11 001 011	CB											
		dd ddd ddd	d											
		10 bbb 110	86+b×8											
	RES b,(IY+d)	11 111 101	FD	$(IY+d)_b \leftarrow 0$	–	–	X	–	X	–	–	–	6	23
		11 001 011	CB											
		dd ddd ddd	d											
		10 bbb 110	86+b×8											
JUMP	JP mn	11 000 011	C3	PC←mn	–	–	X	–	X	–	–	–	3	10
		nn nnn nnn	n											
		mm mmm mmm	m											
	JP c,mn	11 ccc 010	C2+c×8	PC←mn	–	–	X	–	X	–	–	–	3	10
		nn nnn nnn	n	(Only when condition is met)										
		mm mmm mmm	m											
	JR $+e	00 011 000	18	PC←$+e	–	–	X	–	X	–	–	–	3	12
		aa aaa aaa	a											
	JR C,$+e	00 111 000	38	If C=0, continue	–	–	X	–	X	–	–	–	2	7
		aa aaa aaa	a	If C=1, PC←$+e	–	–	X	–	X	–	–	–	3	12
	JR NC,$+e	00 110 000	30	If C=0, PC←$+e	–	–	X	–	X	–	–	–	3	12
		aa aaa aaa	a	If C=1, continue	–	–	X	–	X	–	–	–	2	7
	JR Z,$+e	00 101 000	28	If Z=0, continue	–	–	X	–	X	–	–	–	2	7
		aa aaa aaa	a	If Z=1, PC←$+e	–	–	X	–	X	–	–	–	3	12
	JR NZ,$+e	00 100 000	20	If Z=0, PC←$+e	–	–	X	–	X	–	–	–	3	12
		aa aaa aaa	a	If Z=1, continue	–	–	X	–	X	–	–	–	2	7
	DJNZ $+e	00 010 000	10	B←B−1, If B=0, continue	–	–	X	–	X	–	–	–	2	8
		aa aaa aaa	a	B←B−1, If B<>0, continue	–	–	X	–	X	–	–	–	3	13
	JP (HL)	11 101 001	E9	PC←HL	–	–	X	–	X	–	–	–	1	4

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

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

e represents the extension in the relative addressing mode, a = e − 2. e is a signed two's complement number in the range of −126 ≤ e ≤ 129

Note :
- a = e − 2 in the Opcode provides an effective address of PC + e as PC is incremented by 2 before the addition of e.
- $ indicates the reference to the location counter value of the current segment.
- The notation $(HL)_b$, $(IX + d)_b$ indicates bit $_b$ (0 to 7) within the contents of the register pair.
- The notation r_b indicates bit $_b$ (0 to 7) within the r register.
- a = e-2 in the op-code provides effective address of PC + e as PC is incremented by 2 prior to the addition of e.

c	ccc	Condition
NZ	000	Non-Zero
Z	001	Zero
NC	010	No-carry
C	011	Carry
PO	100	Odd Parity
PE	101	Even Parity
P	110	Sign Positive
M	111	Sign Negative

270489

TMPZ84C013A Instruction Set (9/9)

ITEM/CLASSIFICATION	Assembler mnemonic	Object code Binary (76 543 210)	Object code Hex	Function	Flag S	Z		H		P/V	N	C	No. OF CYCLES	No. OF STATES
JUMP	JP (IX)	11 011 101 11 101 001	DD E9	PC←(IX)	–	–	X	–	X	–	–	–	2	8
	JP (IY)	11 111 101 11 101 001	FD E9	PC←(IY)	–	–	X	–	X	–	–	–	2	8
CALL AND RETURN	CALL mn	11 001 101 nn nnn nnn mm mmm mmm	CD n m	(SP−1)←PC_H,(SP−2)←PC_L PC←mn SP←SP−2	–	–	X	–	X	–	–	–	5	17
	CALL c,mn	11 ccc 100 nn nnn nnn mm mmm mmm	C4+cX8 n m	If condition c is met, same as CALL mn.	–	–	X	–	X	–	–	–	5	17
				If condition c is not met,continue	–	–	X	–	X	–	–	–	3	10
	RET	11 001 001	C9	PC_L←(SP), PC_H←(SP+1) SP←SP+2	–	–	X	–	X	–	–	–	3	10
	RET c	11 ccc 000	C0+cX8	If condition c is met, same as RET.	–	–	X	–	X	–	–	–	3	11
				If condition c is not met,continue	–	–	X	–	X	–	–	–	1	5
	RETI	11 101 101 01 001 101	ED 4D	Return from interrupt Processing routine	–	–	X	–	X	–	–	–	4	14
	RETN	11 101 101 01 000 101	ED 45	Return from non-maskable interrupt Processing routine	–	–	X	–	X	–	–	–	4	14
	RST j	11 kkk 111	C7+kX8	(SP−1)←PC_H,(SP−2)←PC_L PC_H←0,PC_L←j,SP←SP−2	–	–	X	–	X	–	–	–	3	11
INPUT AND OUTPUT	IN A,(n)	11 011 011 nn nnn nnn	DB n	A←(n) n→A0~A7,A→A8~A15	–	–	X	–	X	–	–	–	3	11
	IN r,(C)	11 101 101 01 rrr 000	ED 40+rX8	r←(C) If r=110, only the flags will be affected.	*	*	X	*	X	P	0	–	3	12
	INI	11 101 101 10 100 010	ED A2	(HL)←(C),B←B−1,HL←HL+1	X	*M	X	X	X	X	1	X	4	16
	INIR	11 101 101 10 110 010	ED B2	(HL)←(C),B←B−1,HL←HL+1 Repeat until B=0	X	1	X	X	X	X	1	X	5 4	21 16
	IND	11 101 101 10 101 010	ED AA	(HL)←(C),B←B−1,HL←HL−1	X	*M	X	X	X	X	1	X	4	16
	INDR	11 101 101 10 111 010	ED BA	(HL)←(C),B←B−1,HL←HL−1 Repeat until B=0	X	1	X	X	X	X	1	X	5 4	21 16
	OUT (n),A	11 010 011 nn nnn nnn	D3 n	(n)←A n→A0~A7,A→A8~A15	–	–	X	–	X	–	–	–	3	11
	OUT (C),r	11 101 101 01 rrr 001	ED 41+rX8	(C)←r	–	–	X	–	X	–	–	–	3	12
	OUTI	11 101 101 10 100 011	ED A3	(C)←(HL),B←B−1,HL←HL+1	X	*M	X	X	X	X	1	X	4	16
	OTIR	11 101 101 10 110 011	ED B3	(C)←(HL),B←B−1,HL←HL+1 Repeat until B=0	X	1	X	X	X	X	1	X	5 4	21 16
	OUTD	11 101 101 10 101 011	ED AB	(C)←(HL),B←B−1,HL←HL−1	X	*M	X	X	X	X	1	X	4	16
	OTDR	11 101 101 10 111 010	ED BB	(C)←(HL),B←B−1,HL←HL−1 Repeat until B=0	X	1	X	X	X	X	1	X	5 4	21 16

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

r	rrr
B	000
C	001
D	010
E	011
H	100
L	101
A	111

←[B<>0]
←[B=0] *1

←[B<>0]
←[B=0]

←B[<>0]
←[B=0] *1

←[B<>0]
←[B=0]

Note :
- *M If the result of B − 1 is zero, the Z flag is set, otherwise it is reset.
- A0 through A15 indicate the address bus.
- [ ] indicates the total condition of the number of cycles and states indicated by arrow.

*1 C→A0~A7
B→A8~A15

c	ccc	Condition
NZ	000	Non-Zero
Z	001	Zero
NC	010	No-Carry
C	011	Carry
PO	100	Odd Parity
PE	101	Even Parity
P	110	Sign Positive
M	111	Sign negative

120489

TMPZ84C013A Instruction Map (1/7)

MPU Instruction Table (I)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC, mn	LD (BC), A	INC BC	INC B	DEC B	LD B, n	RLCA	EX AF,AF	ADD HC,BC	LD A, (BC)	DEC BC	INC C	DEC C	LD C, n	RRCA
1	DJNZ e	LD DE, mn	LD (DE), A	INC DE	INC D	DEC D	LD D,N	RLA	JE e	ADD HL,DE	LD A, (DE)	DEC DE	INC E	DEC E	LD E, n	RRA
2	JR NZ, e	LD HL, mn	LD (mn),HL	INC HL	INC D	DEC H	LD H, n	DAA	JR Z e	ADD HL,HL	LD HL,(mn)	DEC HL	INC L	DEC L	LD L, n	CPL
3	JR NC, e	LD SP, mn	LD (mn),A	INC SP	INC (HL)	DEC (HL)	LD (HL), n	SCF	JR C e	ADD HL,SP	LD A,(mn)	DEC SP	INC A	DEC A	LD A, n	CCF
4	LD B, B	LD B, C	LD B, D	LD B, E	LD B, H	LD B, L	LD B, (HL)	LD B, A	LD C, B	LD C, C	LD C, D	LD C, E	LD C, H	LD C, L	LD C, (HL)	LD C, A
5	LD D, B	LD D, C	LD D, D	LD D, E	LD D, H	LD D, L	LD D, (HL)	LD D, A	LD E, B	LD E, C	LD E, D	LD E, E	LD E, H	LD E, L	LD E, (HL)	LD E, A
6	LD H, B	LD H, C	LD H, D	LD H, E	LD H, H	LD H, L	LD H, (HL)	LD H, A	LD L, B	LD L, C	LD L, D	LD L, E	LD L, H	LD L, L	LD L, (HL)	LD L, A
7	LD (HL), B	LD (HL), C	LD (HL), D	LD (HL), E	LD (HL), H	LD (HL), L	HALT	LD (HL), A	LD A, B	LD A, C	LD A, D	LD A, E	LD A, H	LD A, L	LD A, (HL)	LD A, A
8	ADD A, B	ADD A, C	ADD A, D	ADD A, E	ADD A, H	ADD A, L	ADD A, (HL)	ADD A, A	ADC A, B	ADC A, C	ADC A, D	ADC A, E	ADC A, H	ADC A, L	ADC A, (HL)	ADC A, A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A, B	SBC A, C	SBC A, D	SBC A, E	SBC A, H	SBC A, L	SBC A, (HL)	SBC A, A
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	XOR A
C	RET NZ	POP BC	JP NZ, mn	JP mn	CALL NZ, mn	PUSH BC	ADD A, n	RST 00H	RET C	RET	JP Z, mn	①	CALL Z, mn	CALL mn	ADC A, n	RST 08H
D	RET NC	POP DE	JP NC, mn	OUT (n), A	CALL NC, mn	PUSH DE	SUB n	RST 10H	RET C	EXX	JP C, mn	IN A, (n)	CALL C, mn	③	SBC A, n	RST 18H
E	RET PO	POP HL	JP PO, mn	EX (SP),HL	CALL PO, mn	PUSH HL	AND n	RST 20H	RET PE	JP (HL)	JP PE, mn	EX DE,HL	CALL PI, mn	②	XOR n	RST 28H
F	RET P	POP AF	JP P, mn	DI	CALL P, mn	PUSH AF	OR n	RST 30H	RET M	LD SP, HL	JP M, mn	EI	CALL M, mn	④	CP n	RST 38H

120489

Note ①~④: Multi-Opcode Instructions (ref. Table (II)~(VII))

TMPZ84C013A Instruction Map (2/7)

① Byte 1 "CB"

Instruction Table (II) (Byte 2 of 2-byte Opcode)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	BIT 0, B	BIT 0, C	BIT 0, D	BIT 0, E	BIT 0, H	BIT 0, L	BIT 0,(HL)	BIT 0, A	BIT 1, B	BIT 1, C	BIT 1, D	BIT 1, E	BIT 1, H	BIT 1, L	BIT 1, (HL)	BIT 1, A
5	BIT 2, B	BIT 2, C	BIT 2, D	BIT 2, E	BITT 2, H	BIT 2, L	BIT 2, (HL)	BIT 2, A	BIT 3, B	BIT 3, C	BIT 3, D	BIT 3, E	BIT 3, H	BIT 3, L	BIT 3, (HL)	BIT 3, A
6	BIT 4, B	BIT 4, C	BIT 4, D	BIT 4, E	BIT 4, H	BIT 4, L	BIT 4, (HL)	BIT 4, A	BIT 5, B	BIT 5, C	BIT 5, D	BIT 5, E	BIT 5, H	BIT 5, L	BIT 5, (HL)	BIT 5, A
7	BIT 6, B	BIT 6, C	BIT 6, D	BIT 6, E	BIT 6, H	BIT 6, L	BIT 6, (HL)	BIT 6, A	BIT 7, B	BIT 7, C	BIT 7, D	BIT 7, E	BIT 7, H	BIT 7, L	BIT 7, (HL)	BIT 7, A
8	RES 0, B	RES 0, C	RES 0, D	RES 0, E	RES 0, H	RES 0, L	RES 0, (HL)	RES 0, A	RES 1, B	RES 1, C	RES 1, D	RES 1, E	RES 1, H	RES 1, L	RES 1, (HL)	RES 1, A
9	RES 2, B	RES 2, C	RES 2, D	RES 2, E	RES 2, H	RES 2, L	RES 2, (HL)	RES 2, A	RES 3, B	RES 3, C	RES 3, D	RES 3, E	RES 3, H	RES 3, L	RES 3, (HL)	RES 3, A
A	RES 4, B	RES 4, C	RES 4, D	RES 4, E	RES 4, H	RES 4, L	RES 4, (HL)	RES 4, A	RES 5, B	RES 5, C	RES 5, D	RES 5, E	RES 5, H	RES 5, L	RES 5, (HL)	RES 5, A
B	RES 6, B	RES 6, C	RES 6, D	RES 6, E	RES 6, H	RES 6, L	RES 6, (HL)	RES 6, A	RES 7, B	RES 7, C	RES 7, D	RES 7, E	RES 7, H	RES 7, L	RES 7, (HL)	RES 7, A
C	SET 0, B	SET 0, C	SET 0, D	SET 0, E	SET 0, H	SET 0, L	SET 0, (HL)	SET 0, A	SET 1, B	SET 1, C	SET 1, D	SET 1, E	SET 1, H	SET 1, L	SET 1, (HL)	SET 1, A
D	SET 2, B	SET 2, C	SET 2, D	SET 2, E	SET 2, H	SET 2, L	SET 2, (HL)	SET 2, A	SET 3, B	SET 3, C	SET 3, D	SET 3, E	SET 3, H	SET 3, L	SET 3, (HL)	SET 3, A
E	SET 4, B	SET 4, C	SET 4, D	SET 4, E	SET 4, H	SET 4, L	SET 4, (HL)	SET 4, A	SET 5, B	SET 5, C	SET 5, D	SET 5, E	SET 5, H	SET 5, L	SET 5, (HL)	SET 5, A
F	SET 6, B	SET 6, C	SET 6, D	SET 6, E	SET 6, H	SET 6, L	SET 6, (HL)	SET 6, A	SET 7, B	SET 7, C	SET 7, D	SET 7, E	SET 7, H	SET 7, L	SET 7, (HL)	SET 7, A

120489

TMPZ84C013A Instruction Map (3/7)

② Byte 1 "ED"

Instruction Table (III) (Byte 2 of 2-byte Opcode)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4	IN B, (C)	OUT (C), B	SBC HL,BC	LD (mn),BC	NEG	RETN	IM0	LD I, A	IN C, (C)	OUT (C), C	ADC HL,BC	LD BC,(mn)		RETI		LD R, A
5	IN D, (C)	OUT (C), D	SBC HL,DE	LD (mn),DE			IM1	LD A, I	IN E, (C)	OUT (C), E	ADC HL,DE	LD DE,(mn)			IM2	LD A, R
6	IN H, (C)	OUT (C), H	SBC HL,HL	LD (mn),HL				RRD	IN L, (C)	OUT (C), L	ADC HL,HL	LD HL,(mn)				RLD
7			SBC HL,SP	LD (mn),SP					IN A, (C)	OUT (C), A	ADC HL,SP	LD SP,(mn)				
8																
9																
A	LDI	CPI	INI	OUTI					LDD	CPD	IND	OUTD				
B	LDIR	CPIR	INIR	OTIR					LDDR	CPDR	INDR	OTDR				
C																
D																
E																
F																

120489

TMPZ84C013A Instruction Map (4/7)

③ Byte 1 "DD"

Instruction Table (Ⅳ) (Byte 2 of 2-byte Opcode)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0										ADD IX,BC						
1										ADD IX,DE						
2		LD IX, mn	LD (mn), IX	INC IX						ADD IX, IX	LD IX,(mn)	DEC IX				
3					INC (IX + d)	DEC (IX + d)	LD (IX + d) , n			ADD IX, SP						
4							LD B, (IX + d)								LD C, (IX + d)	
5							LD D, (IX + d)								LD E, (IX + d)	
6							LD H, (IX + d)								LD L, (IX + d)	
7	LD (IX + d) , B	LD (IX + d) , C	LD (IX + d) , D	LD (IX + d) , E	LD (IX + d) , H	LD (IX + d) , L		LD (IX + d) , A							LD A, (IX + d)	
8							ADD (IX + d)								ADC 1, (IX + d)	
9							SUB (IX + d)								SBC A, (IX + d)	
8							AND (IX + d)								XOR (IX + d)	
9							OR (IX + d)								CP (IX + d)	
C												⑤				
D																
E		POP IX		EX (SP), IX		PUSH IX				JP (IX)						
F										LD SP, IX						

120489

Note ⑤: Special 2 byte Opcode Instructions (ref. Table (Ⅵ))

TMPZ84C013A Instruction Map (5/7)

④ Byte 1 "FD"

Instruction Table (Ⅴ) (Byte 2 of 2-byte Opcode)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0										ADD IY,BC						
1										ADD IY,DE						
2		LD IY, nn	ID (mn), IY	INC IY						ADD IY, IY	LD IY, (mn)	DEC IY				
3					INC (IY + d)	DEC (IY + d)	LD (IY + d), n			ADD IY, SP						
4							LD B, (IY + d)								LD C, (IY + d)	
5							LD D, (IY + d)								LD E, (IY + d)	
6							LD H, (IY + d)								LD L, (IY + d)	
7	LD (IY + d), B	LD (IY + d), C	LD (IY + d), D	LD (IY + d), E	LD (IY + d), H	LD (IY + d), L		LD (IY + d), A							LD A, (IY + d)	
8							ADD (IY + d)								ADDC A, (IY + d)	
9							SUB (IY + d)								SUBC A, (IY + d)	
8							AND (IY + d)								XOR (IY + d)	
9							OR (IY + d)								CP (IY + d)	
C												⑥				
D																
E		POP IY		EX (SP), IY		PUSH IY				JP (IY)						
F										LD SP, IY						

120489

Note ⑥: Special 2 byte Opcode Instructions (ref. Table (Ⅶ))

TMPZ84C013A Instruction Map (6/7)

⑤ Byte 1 "DD"

Byte 2 "CB"

Instruction Table (VI) (Special case of 2-byte Opcode : Byte 3)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0							RLC (IX + d)								RRC (IX + d)	
1							RL (IX + d)								RR (IX + d)	
2							SLA (IX + d)								SRA (IX + d)	
3															SRL (IX + d)	
4							BIT 0, (IX + d)								BIT 1, (IX + d)	
5							BIT 2, (IX + d)								BIT 3, (IX + d)	
6							BIT 4, (IX + d)								BIT 5, (IX + d)	
7							BIT 6, (IX + d)								BIT 7, (IX + d)	
8							RES 0, (IX + d)								RES 1, (IX + d)	
9							RES 2, (IX + d)								RES 3, (IX + d)	
A							RES 4, (IX + d)								RES 5, (IX + d)	
B							RES 6, (IX + d)								RES 7, (IX + d)	
C							SET 0, (IX + d)								SET 1, (IX + d)	
D							SET 2, (IX + d)								SET 3, (IX + d)	
E							SET 4, (IX + d)								SET 5, (IX + d)	
F							SET 6, (IX + d)								SET 7, (IX + d)	

120489

TMPZ84C013A Instruction Map (7/7)

⑥ Byte 1 "FD"

Byte 2 "CB"

Instruction Table (VII) (Special case of 2-byte Opcode : Byte 3)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0							RLC (IY + d)								RRC (IY + d)	
1							RL (IY + d)								RR (IY + d)	
2							SLA (IY + d)								SRA (IY + d)	
3															SRL (IY + d)	
4							BIT 0, (IY + d)								BIT 1, (IY + d)	
5							BIT 2, (IY + d)								BIT 3, (IY + d)	
6							BIT 4, (IY + d)								BIT 5, (IY + d)	
7							BIT 6, (IY + d)								BIT 7, (IY + d)	
8							RES 0, (IY + d)								RES 1, (IY + d)	
9							RES 2, (IY + d)								RES 3, (IY + d)	
A							RES 4, (IY + d)								RES 5, (IY + d)	
B							RES 6, (IY + d)								RES 7, (IY + d)	
C							SET 0, (IY + d)								SET 1, (IY + d)	
D							SET 2, (IY + d)								SET 3, (IY + d)	
E							SET 4, (IY + d)								SET 5, (IY + d)	
F							SET 6, (IY + d)								SET 7, (IY + d)	

120489

3.3. CGC Operations

This subsection describes the system configuration, functions, and basic operations of the clock generator/controller (CGC).

3.3.1 Block Diagram

Figure 3.3.1 shows the block diagram of CGC.

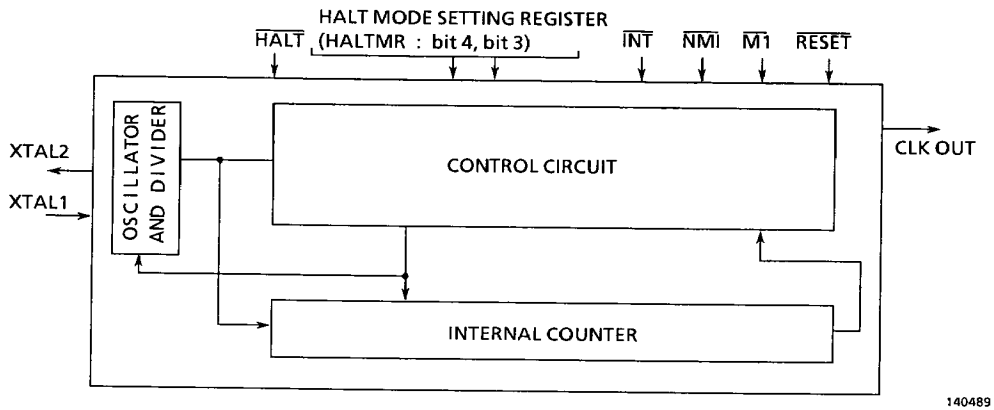


Figure 3.3.1 Block Diagram

3.3.2 CGC System Configuration

The internal configuration of the CGC is shown in Figure 3.3.1. The waveform from the external oscillator oscillated by the internal oscillator and divided by the divider is converted into the square wave for clock. The clock is controlled by the control circuit and the counter to be sent to the outside the CGC. The following describes the CGC's main components and their functions.

(1) Clock Generation

(2) Operation Modes

[1] Clock Generation

The CGC contains an oscillation circuit. By connecting oscillator to external pins (XTAL1 and XTAL2), the required clock can be generated easily. The CGC provides the clock whose frequency is 1/2 of the osicillation frequency. Figure 3.3.2 shows an example of oscillator connection.

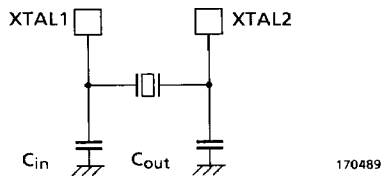


Figure 3.3.2 (a) Example of Crystal Connection

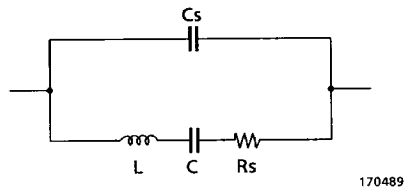


Figure 3.3.2 (b) Oscillator Equivalent Circuit

(1) For the quartz crystal oscillator, use the MR8000-C20 (oscillation frequency 8 MHz) or MR12000-C20 (oscillation frequency 12 MHz) manufactured by Tokyo Denpa Company Ltd., or the equivalent;

Product No.	Holder Type	Frequency (MHz)	Cin (pF)	Cout (pF)	Quartz Crystal Parameter (Typ.)			Drive Level (mW)	Condition Load Capacitance (pF)
					C_1 (pF)	C_0 (pF)	R_1 (Ω)		
MR8000-C20	HC-49 -U (TR-49)	8	22	33	—	4.00	30.0	—	—
MR8000-C14		8	20	20	0.0189	3.87	6.0	0.5	12.67
MR12000-C20		12	33	33	—	4.00	25.0	—	—
MR12000-C14		12	20	20	0.0190	3.81	6.9	0.5	12.55
MR16000-C14		16	20	20	0.0197	4.00	5.7	0.5	12.20
MR20000-C14		20	20	20	—	4.00	25.0	0.5	14.00

170890

(2) For the ceramic resonater, use the CSA8.00MT, CST8.00MT (oscillation frequency 8 MHz) or CSA12.00MT, CST12.00MT (osillation frequency 12 MHz) manufactured by Murata MFG Co., Ltd.

Product No.	Frequency (MHz)	Cin (pF)	Cout (pF)
CSA8.00MT	8	30	30
CST8.00MT	8		
CSA12.0MT	12	30	30
CST12.0MT	12		
CSA20.00MX040	20	5	5

170890

Note : The CST8.00MT and CST12.0MT need no outer capacitance.

[2] Operation Modes

The CGC has the capability to control 4 operation modes;Run, Idle 1, Idle 2, and Stop. Any one of them can be selected through the mode setting register (#F0:bit4, bit3:HALTMR). These modes become valid when the MPU executes a HALT instruction. Fetching a HALT instruction, the MPU sets the $\overline{HALT}$ signal to "0", indicating that it has been put in the halt state. After the execution of the HALT instruction, the CGC performs the operation in the specified mode. Table 3.3.1 shows the operations in each mode.

Table 3.3.1 CGC Operation Modes

Halt mode setting register (#F0:HALTMR)		Operational Mode	Description
Bit 4	Bit 3		
0	0	Idle 1 Mode	Only the internal oscillator operates, stopping the supply of clock outside. The clock output (CLKOUT) is held at "0".
0	1	Idle 2 Mode	The internal oscillator continues operating with the supply of clock outside (CLKOUT) continued. When pins CLKOUT and CLKIN are connected, only the supply of clock (CLKOUT) to the CTC is continued.
1	0	Stop Mode	All internal operations are stopped. The clock output (CLKOUT) is held at "0".
1	1	Run Mode	The supply of clock outside is continued.

140489

The restart from the clock stop state in Idle 1, Idle 2 (these two modes are referred to as Idle mode hereafter), or Stop mode is performed by reset ($\overline{RESET}$ signal) or acknowledge of maskable interrupt ($\overline{INT}$ signal) or non-maskable interrupt ($\overline{NMI}$ signal).

[3] Warm-up Time for Restart (from Stop mode)

Releasing the halt state by interrupt acknowledge, the MPU begins executing interrupt processing. Therefore, when restarting the clock by the $\overline{NMI}$ or $\overline{INT}$ restart signal in the Stop mode, the oscillation must be fully stabilized before supplied outside. The CGC provides, by means of the internal counter, the warm-up time enough for the clock to stabilize frequency. The warm-up ends on the rising edge of the internal counter output dividing the oscillation frequency to start clock output. The warm-up time is equal to the time derived by dividing the frequency of the externally attached oscillator by 2^{14}.

Figure 3.3.3 shows the block diagram of the internal counter. Table 3.3.2 shows the relationship between the oscillation frequency and the warm-up time.

In the restart by the $\overline{RESET}$ signal, no warm-up is performed for the quick operation at power-on. Therefore, expand the width of the $\overline{RESET}$ signal adequately to provide the warm-up time.

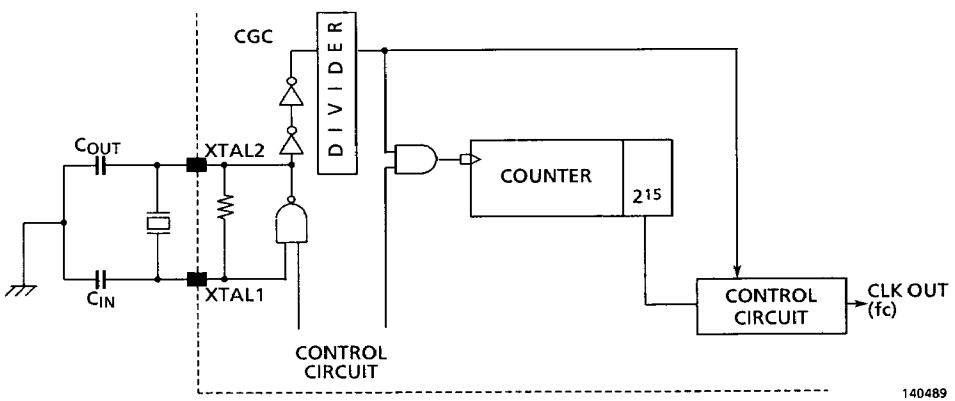


Figure 3.3.3 Block Diagram of Internal Counter

* fc = fxTAL/2

Table 3.3.2 Warm-up Time

Counter output	Warm-up Time		
		fXTAL = 12MHz	fXTAL = 8MHz
2^{15}	2^{14}/fc	2.7 ms	4 ms

140489

3.3.3 CGC Status Transition Diagram and Basic Timing

The following describes the status transition and basic timing to be provided when the CGC operates.

[1] Status Transition Diagram

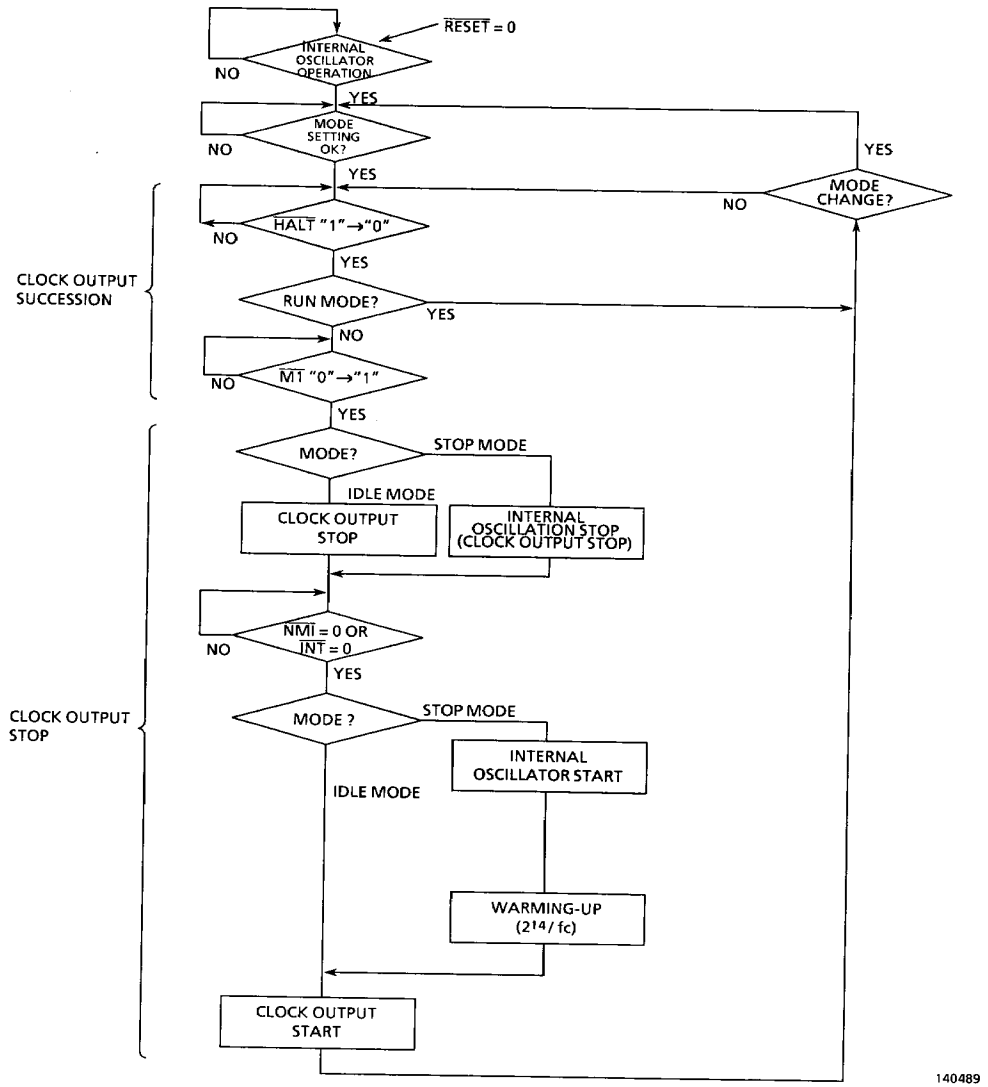


Figure 3.3.4 Status Transition Diagram

[2] Basic Timing

The following describes the CGC basic timing when the CGC clock output pin (CLKOUT) and clock input pin (CLKIN) are connected.

(1) Operation at execution of HALT instruction

The following describes the basic timing in each mode to be provided when the MPU executes a HALT instruction. The MPU sets the $\overline{\text{HALT}}$ signal to "0" synchronized with the falling edge of clock state T4 of the HALT instruction Opcode fetch cycle (M1). This signal indicates to the CGC that the MPU is going to enter into the halt states.

(a) Run mode (#F0:bit 4 = 1, bit 3 = 1:HALTMR)

Figure 3.3.5 shows the basic timing in the Run mode. In the Run mode, the CGC continues supplying the clock to the outside even when the MPU is in the halt state. Therefore, the MPU continues executing NOPs during the halt state. The systems which need memory address refresh use this mode.

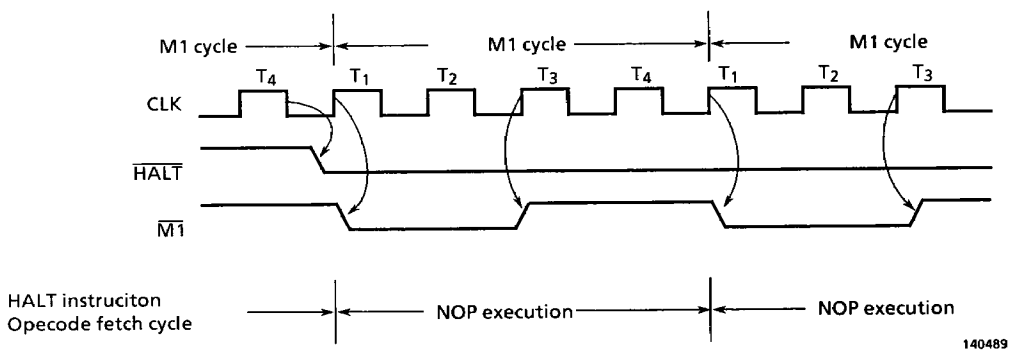


Figure 3.3.5 Basic Timing in Ruin Mode

(b) Idle 1 mode (#F0:bit 4=0, bit 3=0:HALTMR), idle 2 mode (#F0:bit 4=0, bit 3=0:HALTMR), and Stop mode (#F0:bit 4=1, bit 3=0)

Figure 3.3.6 shows the basic timing in the Idle modes and Stop mode. In these modes, the clock output is stopped with clock state T4 being "0" by the $\overline{\text{HALT}}$ signal and the $\overline{\text{M1}}$ signal which follows the HALT instruction.

However, in the Stop mode, the CGC's internal oscillator also stops.

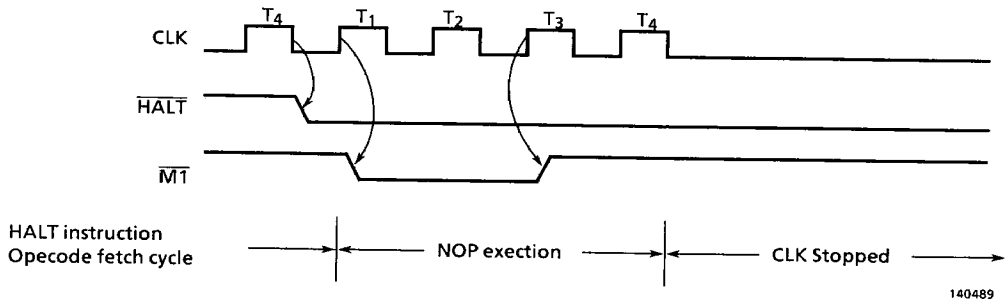


Figure 3.3.6 Basic Timing in Idle and Stop Modes

(2) Clock output restart from each mode

The clock stopped state in the Idle or Stop mode is cleared by setting any of the following signals to "0" (for the system restart operation, see Subsection 3.3.4) :

- $\overline{INT}$ (level trigger input)
- $\overline{NMI}$ (edge trigger input)
- $\overline{RESET}$ (level trigger input)

(a) Clock output restart from Idle mode

Figure 3.3.7 (a) shows the basic timing for the sequence of the output restart from the clock stopped state in the Idle 1 mode. In the restart in the Idle 1 mode, the clock output is restarted in a relatively short delay time because the internal oscillator operates even in the clock stopped state.

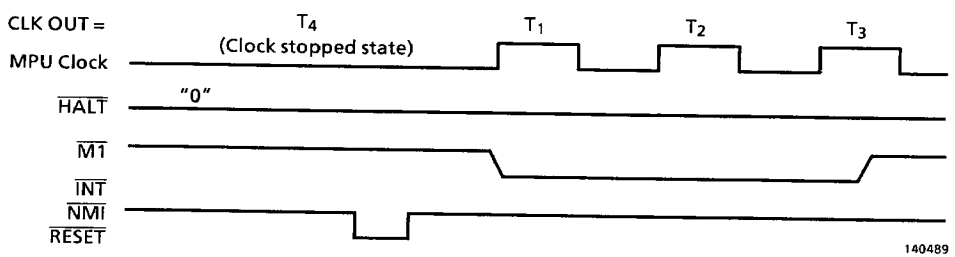
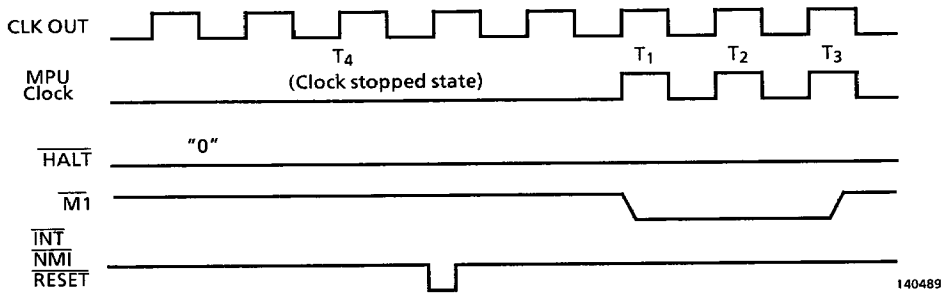


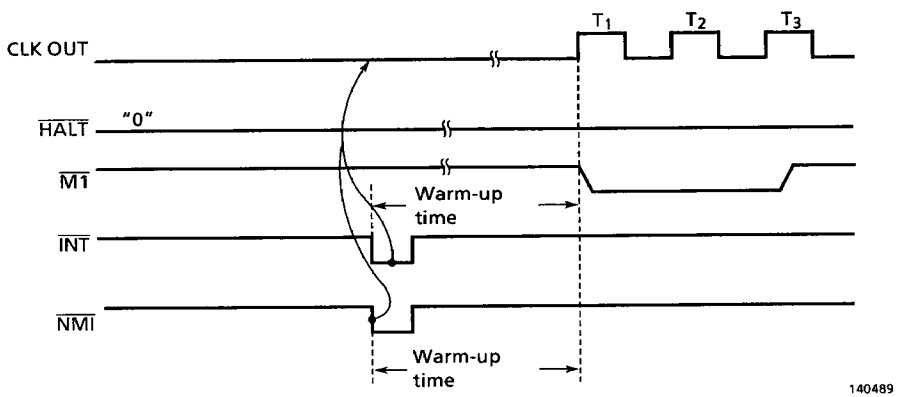
Figure 3.3.7 (a) Basic Timing for Sequence of Restart from Clock stopped State (Idle 1 MOde)

(b) Clock output restart from Stop mode

Figure 3.3.8 shows the basic timing for the sequence of the restart from the clock stopped state in the Stop mode. When restarting by setting the $\overline{INT}$ or $\overline{NMI}$ signal to "0", the warm-up time is automatically created by the internal counter. In the restart by the $\overline{RESET}$ signal, oscillation restarts without warm-up.



Figre 3.3.7 (b) Basic Timing for Sequence of Restart from Clock stopped State (Idle 2 Mode)



Figre 3.3.8 Basic Timing for Sequence of Restart from Clock stopped State (Stop Mode)

3.3.4 Relationship with MPU

The following describes the relationship between the CGC and the MPU mainly in terms of the halt clear operation.

[1] RESET Signal

Figure 3.3.9 shows an example of the timing for the restart from the Stop mode on the TMPZ84C013A using $\overline{\text{RESET}}$ signal for both the MPU and CGC . To reset the MPU, the $\overline{\text{RESET}}$ signal must be set to "0" for at least 3 stable clocks. When the $\overline{\text{RESET}}$ signal goes "1", the MPU releases the halt state after a dummy cycle of 2T clock states to start executing instructions from address 0000H.

To restart the clock output by the $\overline{\text{RESET}}$ signal in the Stop mode, the internal counter to determine the warm-up time does not operate.
Therefore, if the MPU does not restart correctly due to the unstable clock output immediately after the restart of the internal oscillator, or the unstability of the crystal at power-on, the $\overline{\text{RESET}}$ signal must be held at "0" for a time long enough for the MPU to be reset securely.

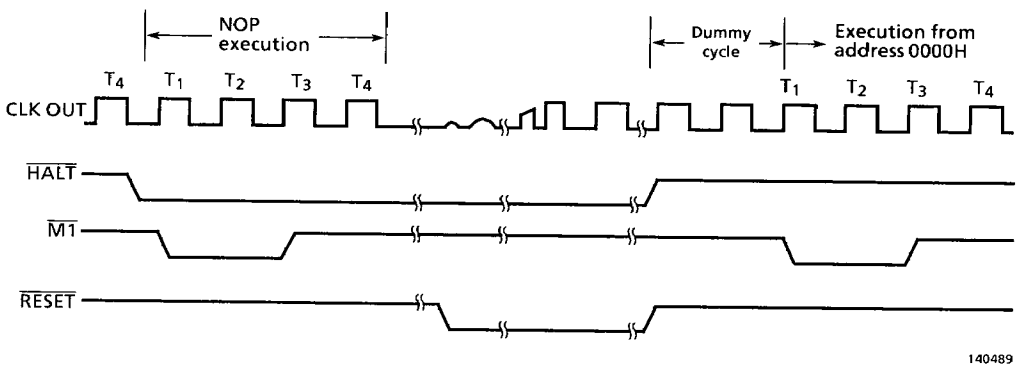


Figure 3.3.9 Example of clock Restart Timing by $\overline{\text{RESET}}$ Signal

[2] Releasing Halt State by Interrupt Signal

The CGC restarts the clock output from the Idle or Stop mode by the input of $\overline{\text{INT}}$ or $\overline{\text{NMI}}$ signal. By this clock, the MPU starts operating. However, when the CGC restarts the clock output, the MPU is still in the halt state executing NOPs. To clear the halt state, the interrupt signal must be entered into the MPU (in the case of the $\overline{\text{INT}}$ signal) for at least one instruction. The MPU interrupt is detected on the rising edge of the last clock of each instruction (NOP for the halt state).

(1) When using non-maskable interrupt (NMI)

MPU's non-maskable interrupt is edge trigger input. The MPU contains the flip-flop to detect an interrupt. The state of this internal NMI flip-flop is sampled on the rising edge of the last clock of each instruction. Therefore, when a short active low ("0") pulse has been inserted before the interrupt detection timing, the interrupt is acknowledged. The $\overline{NMI}$ input of the TMPZ84C013A is connected to the $\overline{NMI}$ input of the MPU via the CGC, performing the same operations as above. (See Figure 3.3.11)

(2) When using maskable interrupt (INT)

With a maskable interrupt, the maskable interrupt enable flip-flop (IFF) must be set to "1" by program before the $\overline{INT}$ input signal is detected "0". Even if the CGC accepts the $\overline{INT}$ signal to restart supply of the clock, no interrupt is acknowledged unless the $\overline{INT}$ signal is kept inserted until one instruction (NOP) has been executed. Figure 3.3.10 shows the timing for clearing the halt state by the interrupt signal.

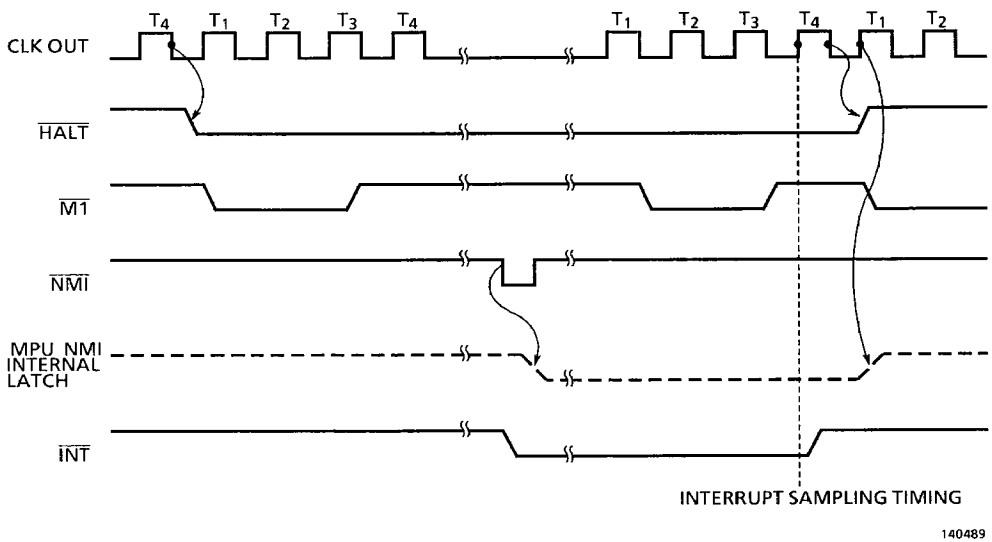


Figure 3.3.10 Timing for Clearing Halt State by Interrupt Signal

[3] Connecting CGC to MPU on TMPZ84C013A

Figure 3.3.11 shows the connection between the CGC and the MPU on the TMPZ84C013A.

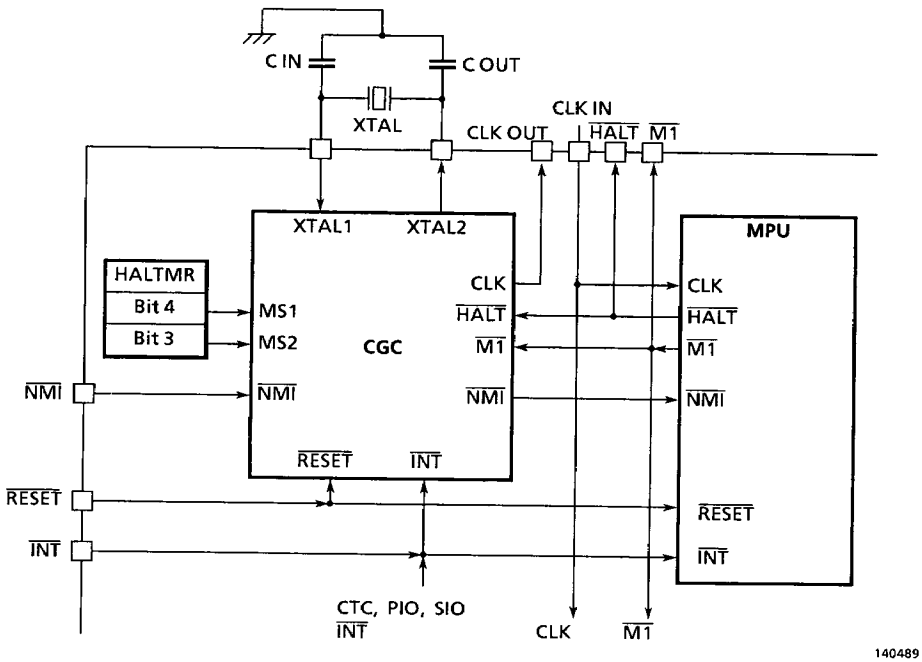


Figure 3.3.11 Connection Between CGC and MPU

3.4 CTC Operational Description

The CTC has 4 independent channels. To these channels, addresses are allocated on the TMPZ84C013A's I/O map, permitting the read/write of the channels in the MPU's I/O cycle. (See Figure 3.4.1) This subsection mainly describes the CTC operation to be performed after accessed.

3.4.1 CTC Block Diagram

Figure 3.4.1 shows the block diagram of the CTC.

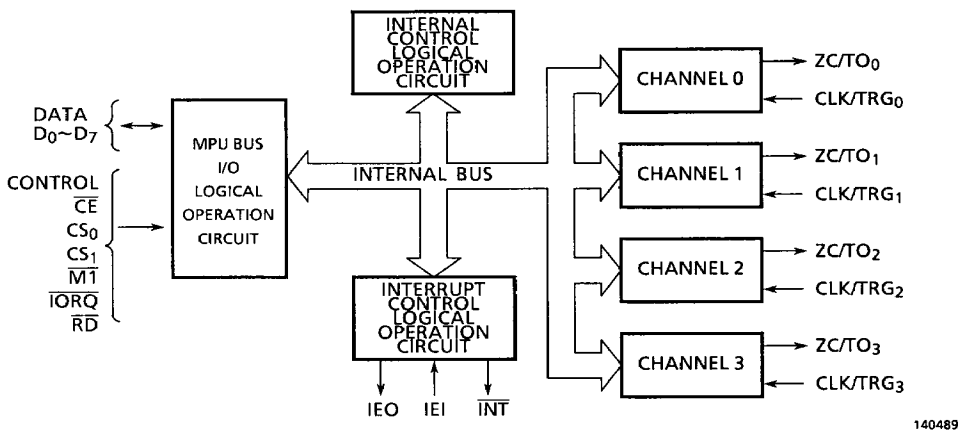


Figure 3.4.1 Block Diagram of CTC

3.4.2 CTC System Configuration

The CTC system consists of the following 4 logic circuits:

(1) MPU bus I/O logic circuit

(2) Internal control logic circuit

(3) Interrupt control logic circuit

(4) Four independent counter/timer channel logic circuits

[1] MPU Bus I/O Logic Circuit

This circuit transfers data between the MPU and the CTC.

[2] Internal Control Logic Circuit

This circuit controls the CTC operational functions such as the CTC selecting chip enable, reset, and read/write circuits.

[3] Interrupt Control Logic Circuit

This circuit performs the MPU interrupt related processing such as priority determination. The order of priority with other LSIs is determined according to the physical location in daisy chain connection.

[4] Counter/Timer Channel Logic Circuit

This circuit consists of the following 2 registers and 2 counters.
Figure 3.4.2 shows the configuration of this circuit.

- Time-constant register (8 bits)
- Channel control register (8 bits)
- Down-counter (8 bits)
- Prescaler (8 bits)

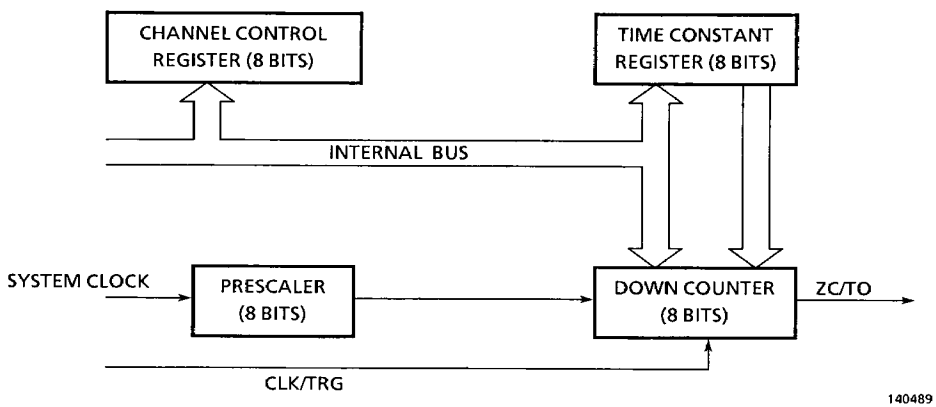


Figure 3.4.2 Configuration of Counter/Timer Channel Logic Circuit

(1) Time-constant register

This register holds the time constant to be written in the down counter. When the CTC is initialized or the down-counter has reached zero, the time constant is loaded into down-counter. The time constant is set immediately after the MPU has written the channel control word in the channel control register. For a time constant, an integer from 1 to 256 can be used.

(2) Channel control register

This register is used to choose the channel mode or condition according to the channel control word sent from the MPU.

(3) Down-counter

The contents of the time-constant register are loaded into the down counter. In the counter mode, these contents are decremented at each edge of the external clock;in the timer mode, they are decremented for each prescaler clock output. The contents of the time-constant register are loaded at initialization or when the down-counter has reached zero.

The contents of the down-counter can be read any time. Also, the system can be programmed so that an interrupt request is generated each time the down-counter has reached zero.

(4) Prescaler

The prescaler, used only in the timer mode, divides the system clock by 16 or 256. The dividing number is programmed by channel control word. The output of the prescaler becomes the clock input to the down-counter.

3.4.3 CTC Basic Operations

[1] Reset

The state of the CTC is unstable after it is powered on. To stabilize the CTC, the low level signal needs to be applied to the $\overline{\text{RESET}}$ pin. On any channel, the channel control word and time-constant data must be written to be started before it is started in the counter or timer mode. To program the system to enable interrupts, the interrupt vector word must be written in the interrupt controller. When these data have been written in the CTC, it is ready to start.

[2] Interrupt

The CTC can cause an interrupt when the MPU is operating in the mode 2. The CTC interrupt can be programmed for each channel. Each time the channel's down-counter has reached zero, the CTC outputs the interrupt request signal ($\overline{\text{INT}}$). When the MPU accepts the CTC's interrupt request, the CTC outputs the interrupt vector. Based on this interrupt vector, the MPU specifies the start address of the interrupt processing routine and calls it to start interrupt processing.

The MPU specifies the start address of the interrupt processing routine by the interrupt vector output from the CTC, so that the user can change the vector value to call any desired address.

The interrupt processing is terminated when the MPU executes an RETI instruction. The CTC has the circuit which decodes the RETI instruction. By constantly monitoring the data bus the CTC can detect the termination of the interrupt processing.

The order of interrupt priority with the Z80 peripheral LSIs is determined by the daisy chain connection. That is, the peripheral LSIs are connected one after another and the one physically near the MPU is given a higher priority. The priority of the Z80 peripheral LSIs (CTC, PIO, and SIO) contained in the TMPZ84C013A is determined by the contents of the interrupt priority register (#F4:bits 2 through 0). Inside the CTC, channel 0 is given the highest priority, followed by channels 1, 2 and 3 in this order.

The CTC and other peripheral LSIs on the TMPZ84C013A have the signal lines IEO and IEI. Connect the IEO of a higher peripheral LSI to the IEI of a lower perpheral LSI. Connect the IEI of the highest peripheral LSI to VCC. Leave the IEO of the peripheral LSI unused. In this connection, the CTC interrupt is caused under the following conditions:

- When both IEI and IEO are high, no interrupt is caused. At this time, the $\overline{INT}$ signal is high. An interrupt can be requested in this state.
- When the CTC outputs the interrupt request signal ($\overline{INT}$), the IEO of the CTC becomes low. When the MPU accepts the interrupt, the $\overline{INT}$ goes high again.
- When the IEI goes low, the IEO also goes low.
- While the IEI is low, no interrupt can be requested.
- When the IEI goes low while an interrupt is being serviced, the interrupt processing is aborted.

[3] Operation Modes

The CTC operates in either the counter mode or the timer mode. Mode is selected by writing the channel control word.

(1) Counter mode

In the counter mode, the number of edge of the pulses applied to the channel's CLK/TRG pin is counted. When pulses have been input, the contents of the down-counter are decremented synchronizing with the rising edge of the next system clock. The pulse's rising edge or falling edge to be counted can be specified by the channel control word.

When the contents of the down-counter has reached zero, the high level pulse is output from the ZC/TO pin. When the interrupt is enabled by the channel control word, the $\overline{INT}$ pin goes low and an interrupt is requested. When the contents of the down-counter has reached zero, the time constant data written in the time constant register is automatically loaded into the down-counter. To load a new time constant value into the down-counter, write the data to the time constant register, and it is loaded into the down-counter after the current count operation is terminated.

(2) Timer mode

In the timer mode, the time intervals which are integral multiples of the system clock period. A timer interval is measured according to the system clock. The system clock is supplied to the prescaler which divides it by a factor of 16 or 256. The output of the prescaler provides the clock to decrement the down-counter by 1. The time constant data is automatically loaded into the down-counter each time it has reached zero as in the counter mode. When the contents of the down-counter has reached zero, the high level pulse is output from the ZC/TO pin.

This pulse period is given by the following expression:

tc*P*TC

Where, tc = System clock period
P = Prescaler value (16 or 256)
TC = Time constant data (256 for 00H)

The user can select, by means of the channel control word, to start the timer automatically or to start the timer on the edge of the pulse at CLK/TRG pin. In case the user select the CLK/TRG pin, the user can also select the rising edge or falling edge of the pulse.

3.4.4 CTC Status Transition Diagram and Basic Timing

[1] Transition Diagram

Figure 3.4.3 shows the CTC status transition diagram.

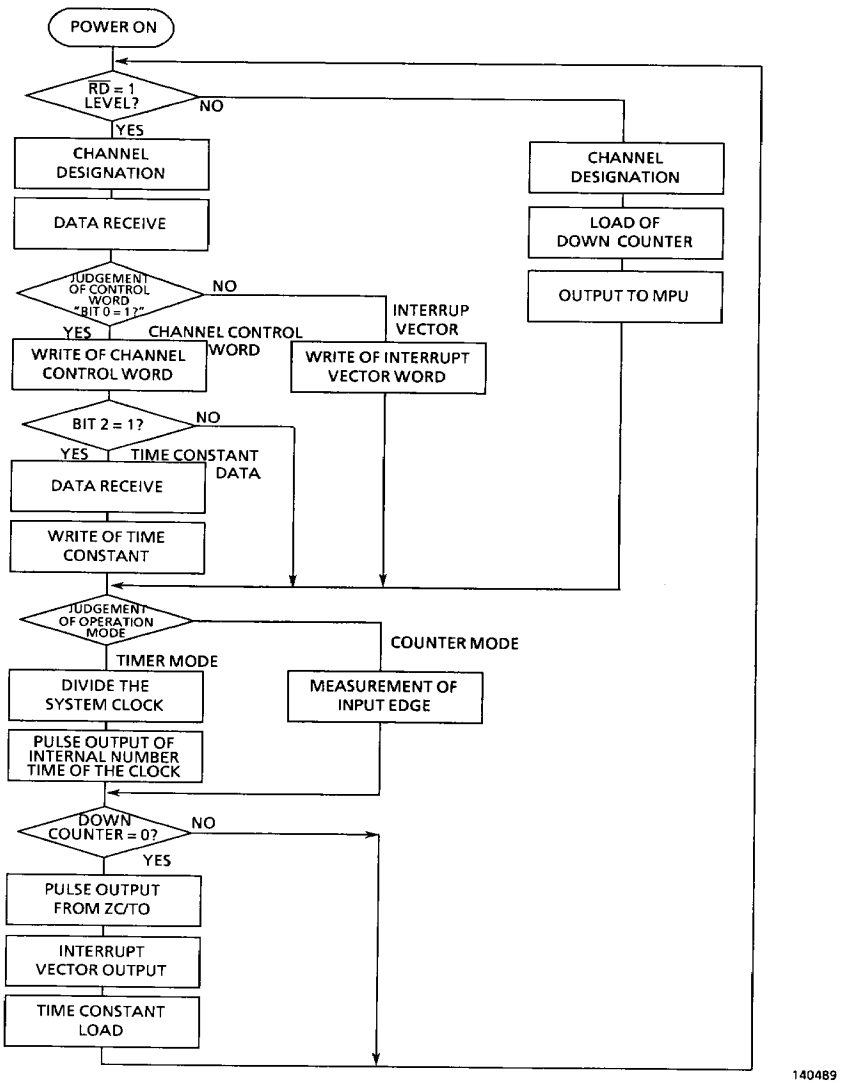


Figure 3.4.3 (a) CTC Transition Diagram (a)

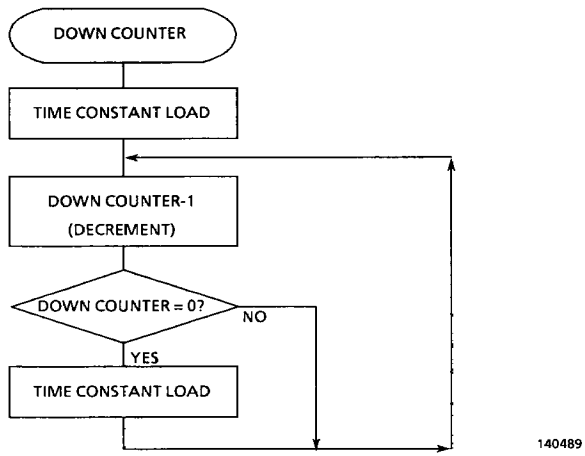


Figure 3.4.3 (b) CTC Transition Diagram (b)

[2] Basic Timing

(1) Write cycle

The write cycle is used to write a channel control word, an interrupt vector, or a time constant. The MPU drives the $\overline{IORQ}$ pin of the CTC low in the subsequent system clock cycle T2 to start the write cycle.
It is required to make the $\overline{M1}$ pin of the CTC high to indicate that the write cycle is on.

At the start of the cycle, the channel is specified by CS1 (A1) or CSO (A0) of the CTC. Thus, the CTC's internal registers are ready to accept data in system clock T3. Tw is the state to be automatically inserted by the MPU.

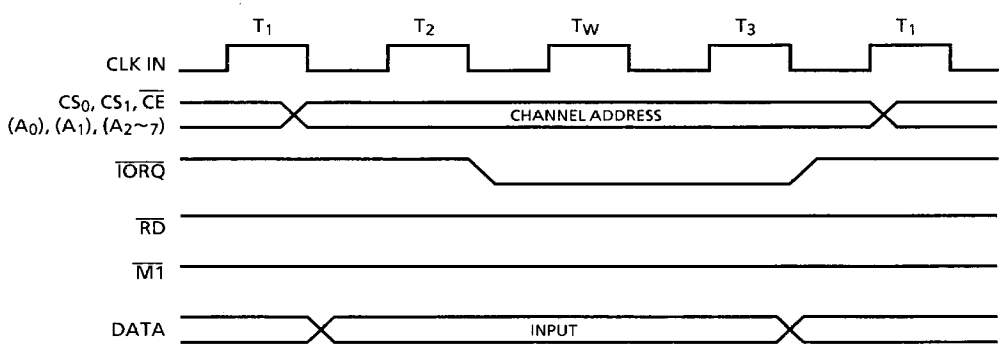


Figure 3.4.4 Write Timing

(2) Read cycle

The read cycle is used to read the contents of the down-counter. During clock cycle T2, the MPU initiates a read cycle by driving the $\overline{RD}$ and $\overline{IORQ}$ pins low. It is required to make the $\overline{M1}$ pin high to indicate that the read cycle is on. At the start of the read cycle, the channel is specified by CS1 (A1) or CS0 (A0) of the CTC.

On the rising edge of system clock TW, the contents of the down-counter at the time of the rising edge of T2 are put on the data bus. TW is the wait state to be automatially inserted by the MPU.

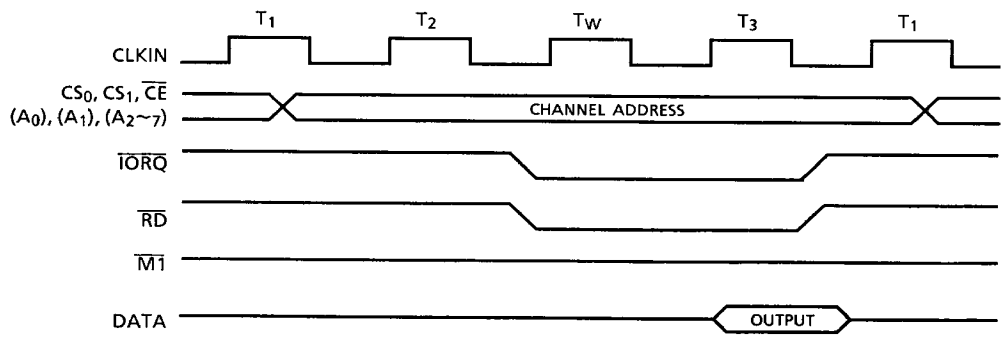


Figure 3.4.5 Read Timing

[3] Counter mode

In the counter mode, the down-counter is decremented synchronizing with the system clock, at the edge of the pulse applied from the external circuit connected to the CLK/TRG pin. The period of the pulse to be applied to the CLK/TRG pin must be greater than 2 times the system clock period. Also, it is required to insert the setup time between the active edge of the CLK/TRG pin signal and the rising edge of the succeeding system clock. When the interval between these pulses is short, the down-counter is decremented one system clock later. When the down-counter has reached zero, a high level pulse is output from the ZC/TO pin.

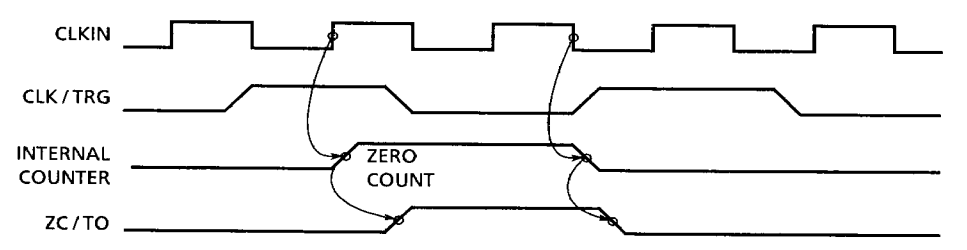


Figure 3.4.6 Counter Mode Timing

[4] Timer mode

The timer starts operating at the second rising edge of the system clock from the rising edge of the pulse applied from the external circuit connected to the CLK/TRG pin. The period of the pulse to be applied to the CLK/TRG pin must be greater than 2 times the system clock period. Also, it is required to insert the setup time between the active edge of the CLK/TRG pin signal and the rising edge of the succeeding system clock. When the interval between these pulses is short, the timer starts one system clock cycle later.

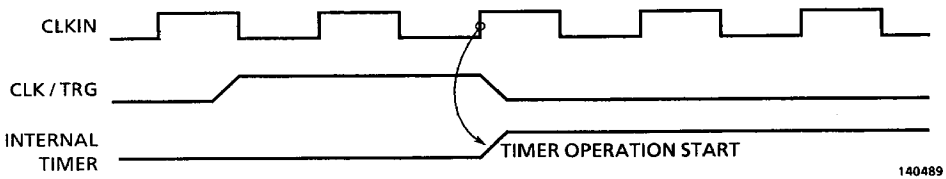


Figure 3.4.7 Timer Mode Timing

[5] Interrupt acknowledge cycle

Having received the interrupt request signal ($\overline{INT}$) from the CTC, the MPU drives the CTC's $\overline{M1}$ pin and $\overline{IORQ}$ pin low to provide the acknowledge signal. The $\overline{IORQ}$ pin goes low 2.5 system clocks later than the $\overline{M1}$ pin. To stabilize the signal lines (IEI and IEO) in daisy chain connection, the interrupt request cannot be changed on each channel while the $\overline{M1}$ pin is low. The $\overline{RD}$ pin is held high to make distinction between the instruction fetch cycle and the interrupt acknowledge cycle. While the $\overline{RD}$ pin is high, the CTC's interrupt control circuit determines the interrupt-requesting channel of highest priority. When the CTC's IEI is high and the $\overline{M1}$ pin and $\overline{IORQ}$ pin go low, the interrupt vector is output from the interrupt requesting channel of highest priority on the data bus. At this time, 2 system clock cycles are automatically inserted by the MPU as a wait state to maintain the stabilization of the daisy chain connection.

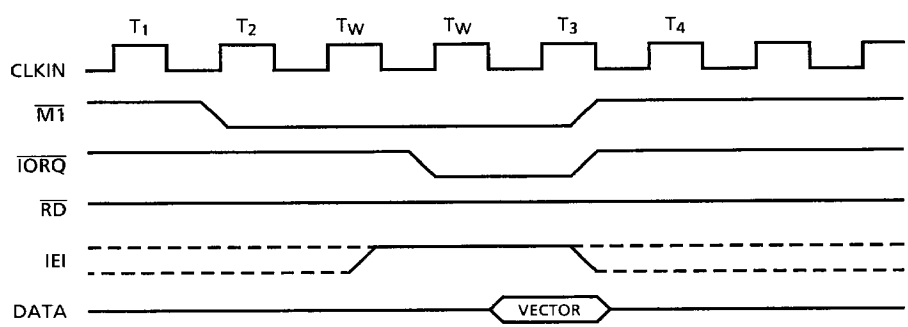


Figure 3.4.8 Interrupt Acknowledge Timing

[6] Return from interrupt processing

Return from the interrupt processing is performed when the MPU executes the RETI instruction. This RETI instruction must be used at the end of the interrupt processing routine. When this instruction is executed by the MPU, the CTC's IEI and IEO return to the state active before the interrupt has been serviced.

The RETI instruction is a 2-byte instruction. Its code is EDH 4DH. The CTC decodes this instruction to check if there is the next interrupt request channel.

In the daisy chain structure, the interrupting LSI's IEI and IEO are held high and low respectively at the time the instruction code EDH has been decoded.

The code following EDH is 4DH, only the peripheral LSI which has sent the last interrupt vector (that is, the LSI whose IEI is high and IEO is low) returns from the interrupt processing. This restarts the processing of the suspended interrupt of the peripheral LSI of the next higher priority.

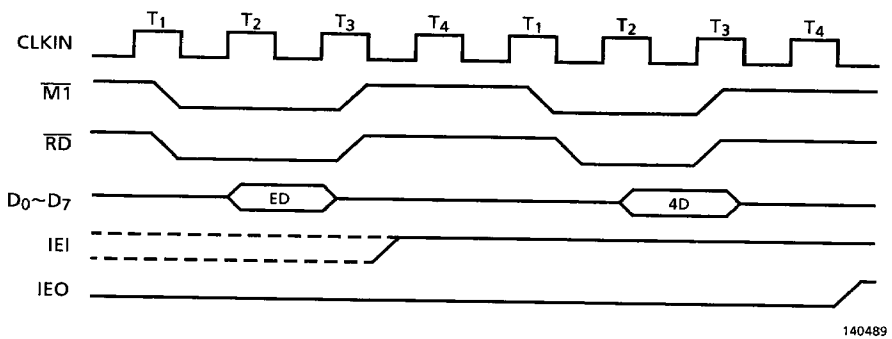


Figure 3.4.9 Interrupt Return Timing

3.4.5 CTC Operational Procedure

To operate the CTC in the counter mode or the timer mode, the channel control word and time-constant data must be written in the CTC. To enable interrupts by the channel control word, the interrupt vector must be written in the CTC.

[1] I/O Address and Channel Control Word

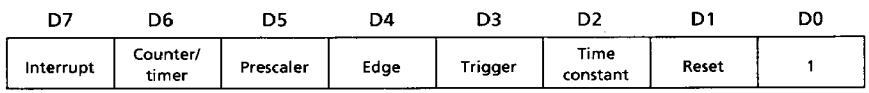
To write the channel control word in the CTC, the channel must be specified by the corresponding channel I/O address. Table 3.4.1 shows the channel I/O addresses.

Table 3.4.1 Channel I/O Addresses

Channel	I/O address
0	#10
1	#11
2	#12
3	#13

140489

The channel control word to be written in the CTC consists of 8 bits. The system data bus D0 through D7 correspond to bit 0 through 7 respectively. Figure 3.4.10 shows the meaning of each bit. Table 3.4.2 shows the function of each bit.



140489

Figure 3.4.10 Channel Control Word

For the channel control word, D0 must be always 1.

Table 3.4.2 Meanings and Function of Channel Control Words (1/3)

Bit	Meaning and function	
	0	1
Bit 7 (D7)	Disables channel interrupt	Enables channel interrupt. In either counter or timer mode, the interrupt is requested every time the down-counter has reached zero.When this bit is set to "1", the interrupt vector must be written in the CTC before the down-counter starts. When the channel control word whose D7 bit is "1" is written in an already operating channel, the interrupt occurs only when the down-counter has reached zero for the first time after the writing of the new channel control word.

140489

(2/3)

Bit	Meaning and function	
	0	1
Bit 6 (D6)	Puts the channel in the timer mode. Puts the system clock into the prescaler and outputs the divided signal to the down-counter.	Puts the channel in the counter mode. The down-counter is decremented for each edge trigger applied to the CLK/TRG pin. In the counter mode, the prescaler is not used.
Bit 5 (D5)	Used only in the timer mode. The prescaler is set to divide the system clock by 16.	Used only in the timer mode. The prescaler is set to divide the system clock by 256.
Bit 4 (D4)	In the timer mode, the timer operation starts on the falling edge of the trigger PULSE (CLK/TRG) . In the counter mode, the down-counter is decremented on the falling edge of the external clock pulse (CLK/TRG)	In the timer mode, the timer operation starts on the rising edge of the trigger pulse (CLK/TRG). In the counter mode, the down-counter is decremented on the rising edge of the trigger pulse (CLK/TRG).
Bit 3 (D3)	Used only in the timer mode. The timer oparation starts on the rising edge following 2 system clocks after the time constant has been loaded into the down-counter.	Used only in the timer mode. The timer operation is started by the external trigger pulse. That is, the timer starts operating at the trigger pulse entered after the rising edge following 2 system clocks after the time constant has been loaded into the down-counter. If the interval between the system clock and the trigger pulse satisfies the setup time, the prescaler starts operating on the second rising edge. Otherwise, the prescaler starts operating on the rising edge following 3 system clocks. If the trigger pulse is entered before the time constant is loaded the same effect as when bit 3 = "0" takes place.
Bit 2 (D2)	Indicates that the channel control word is not followed by the writing of time constant. When the channel is in the reset state, this bit cannot be set to "0" in the first channel control word . To change states with the time constant unchanged, the channel control word with this bit set to "0" must be entered.	Indicates that the channel control word is followed by the writing of time constant. If the time constant is written during down-counter operation, the new time constant is set to the time constant register with proceeding the current counting. When the down-counter has reached zero, the new time constant is loaded into the down-counter.

140489

(3/3)

Bit	Meaning and function	
	0	1
Bit 1 (D1)	Continues the current channel operation	Stops the down-counter operation. When this bit is set to "1", the channel operation stops but all the channel control register bits remain unchanged. When bit 2 = "1" and bit 1 = "1", the channel operation remains stopped until a new time constant is written. Channel restart is set up after the new time constant is programmed. The channel is restarted acoording to the state of bit 3. When bit 2 = "0" and bit 1 = "1", the channel operation does not start until a new channel control word is written.

140489

[2] Time-Constant Data

In either the time mode or the counter mode, the time-constant data must be loaded into the time constant register. When bit 2 (D2) of the channel control word is "1", the time constant is loaded into the time constant register immediately after the channel control word is written. A time-constant value must be an integer in a range of 1 to 256. When the 8 bits of a time constant are all "0"s, such a time constant is assumed to be 256. Figure 3.4.11 shows the bit configuration of time-constant data.

D7	D6	D5	D4	D3	D2	D1	D0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

140489

Figure 3.4.11 Time-Constant Data

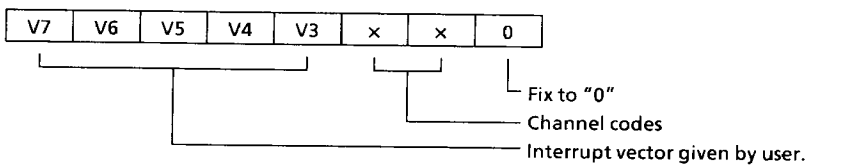
[3] Interrupt Vector

In interrupt in the MPU mode-2, the interrupting channel must give an interrupt vector to the MPU. An interrupt vector is written in the channel-0 interrupt vector register with bit 0 (D0) ="0". The vector is written in the same way as the channel control word is written on channel 0. However, bit 0 (D0) of the vector should always be "0". Bit 7 (D7) through bit 3 (D3) are user-defined values. Bit 2 (D2) and bit 1 (D1) are automatically given and contain the code of the interrupt-requesting channel having the highest priority. Table 3.4.3 shows the channel codes. Figure 3.4.12 shows the interrupt vector bit configuration.

Table 3.4.3 Channel Codes

Bit 2 (D2)	Bit 1 (D1)	Channel number	
0	0	0	(high)
0	1	1	↑
1	0	2	↓ Priority
1	1	3	(low)

140489



140489

Figure 3.4.12 Interrupt Vector

3.4.6 Using CTC

[1] Counter Mode

The following describes how to use the CTC by referring to a program using channel 0 with interrupt disabled.

(a) The counter programming procedure is shown in Figure 3.4.13.

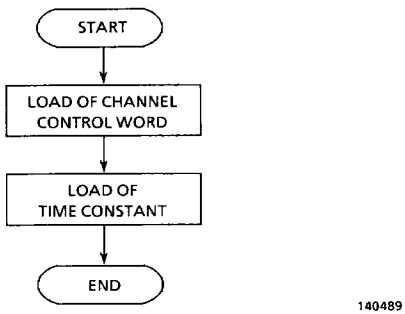


Figure 3.4.13 Counter Programming Procedure

(b) The block diagram for converting the 100 kHz system clock into the 10 kHz equivalent is shown in Figure 3.4.14.

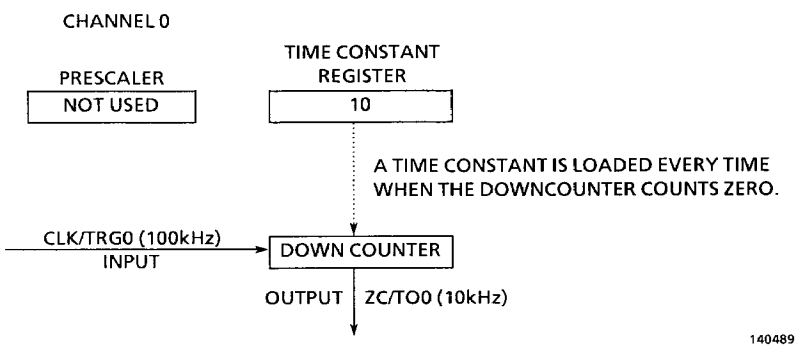


Figure 3.4.14 Down-Counter Block Diagram

(c) The channel control word configuration is shown in Figure 3.4.15.

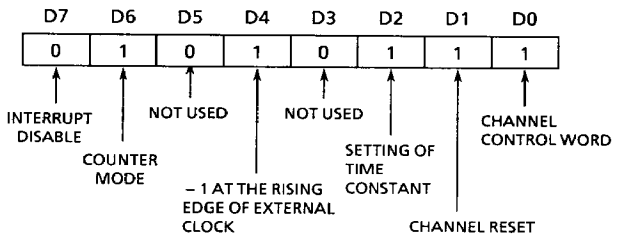


Figure 3.4.15 Channel Control Word Configuration

[2] Timer Mode

(a) The timer programming procedure with interrupt disabled is shown in Figure 3.4.16.

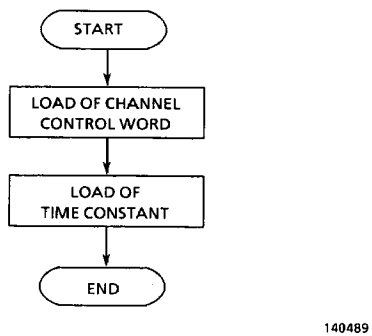


Figure 3.4.16 Timer Programming Procedure

(b) The block diagram for converting the 4 MHz system clock into the 1 kHz equivalent is shown in Figure 3.4.17.

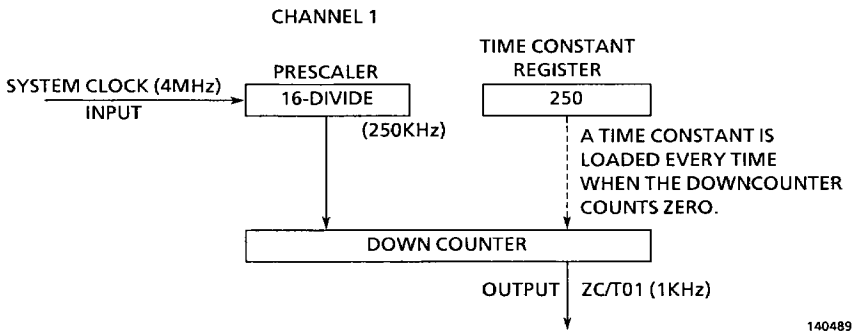


Figure 3.4.17 Timer block Diagram

(c) The channel control word is shown in Figure 3.4.18.

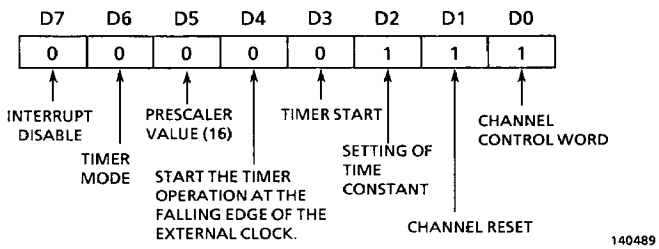


Figure 3.4.18 Channel Control Word

3.5 SIO Operational Description

The SIO has two independent, programmable full-duplex serial ports. These ports are assigned addresses on the TMPZ84C013A's I/O map. This subsection mainly describes the operations that take place after accessing the SIO.

3.5.1 SIO Block Diagram

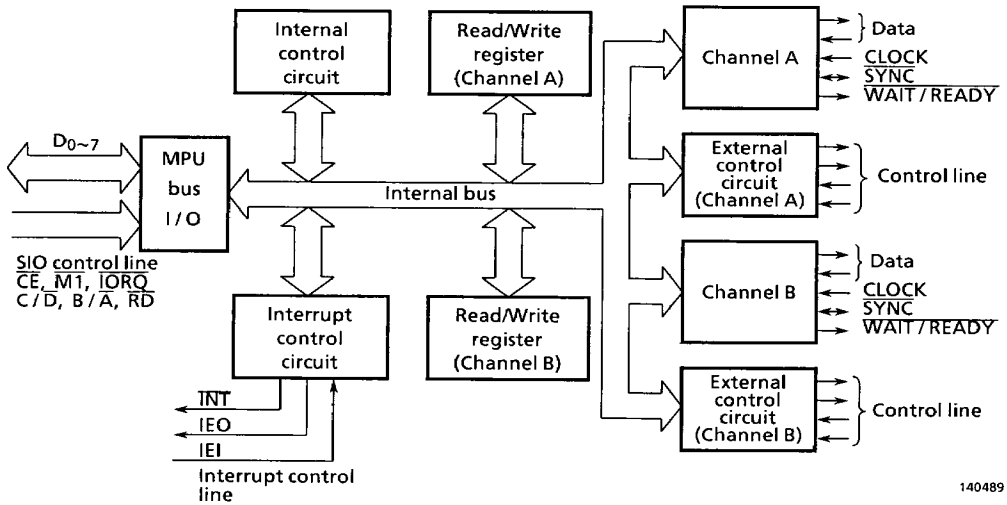
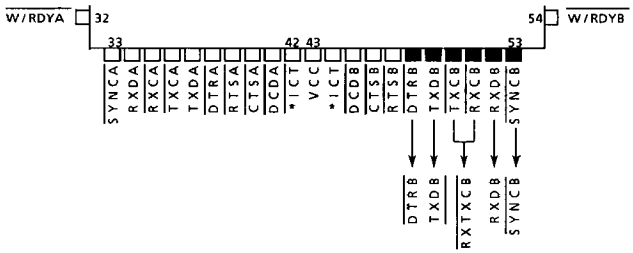


Figure 3.5.1 SIO Block Diagram

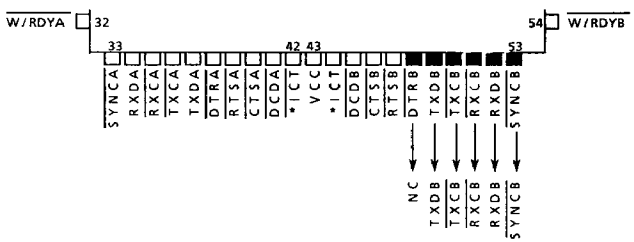
3.5.2 SIO System Configuration

As shown in Figure 3.5.1, the SIO consists of the MPU bus interface, the internal controller, the interrupt controller, and two independently operating full-duplex channels. Each channel has the read register, the write register, and the external controller which controls the connection with peripheral LSIs or external devices.

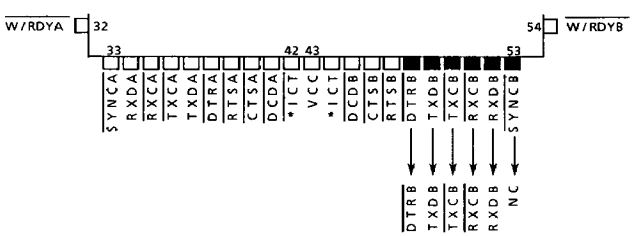
The TMPZ84C013A contains all the functions and pins of the 40-pin, DIP-type TMPZ84C40A (SIO/0), TMPZ84C41A (SIO/1), and TMPZ84C42A (SIO/2). However, when using the SIO of the TMPZ84C013A, the SIO/0, SIO/1, or SIO/2 must be used alone. The pin assignments are as shown in Figure 3.5.2.



(a) SIO/0



(b) SIO/1



(c) SIO/2

140489

Figure 3.5.2 SIO Pin Assignments

Table 3.5.1 shows the types and functions of the SIO registers. Each channel has 8 write registers and 3 read registers.

(1) Communication data path

Figure 3.5.3 shows the communication path of each channel's transfer data.

1 Receive operation

The receiver has an 8-bit receive register and a 3-stage 8-bit buffer register in FIFO configuration. This saves time in high-speed data block transfers. The receivers also have the receive error FIFO which holds the status information such as parity and framing errors. The receive data follow different paths according to the operation mode and character length as shown in Figure 3.5.3.

Table 3.5.1 (a) Write Registers

Register	Function
Write register 0 (WR0)	Resets CRC. Sets pointers of registers, and commands.
Write register 1 (WR1)	Sets the interrupt mode.
Write register 2 (WR2)	Sets the vector to be transmitted at interrupt. (Channel B only)
Write register 3 (WR3)	Provides the parameters to control the receiver.
Write register 4 (WR4)	Provides the parameters to control the receiver and transmitter.
Write register 5 (WR5)	Controls the transmitter.
Write register 6 (WR6)	Sets the sync character or the SDLC address field.
Write register 7 (WR7)	Sets the sync character or the SDLC flag.

140489

Table 3.5.1 (b) Read Registers

Register	Function
Read register 0 (RR0)	Indicates the receive/transmit buffer state and the pin state.
Read register 1 (RR1)	Indicates the error status and the end-of-frame code.
Read register 2 (RR2)	Indicates the interrupt vector contents. (Channel B only)

140489

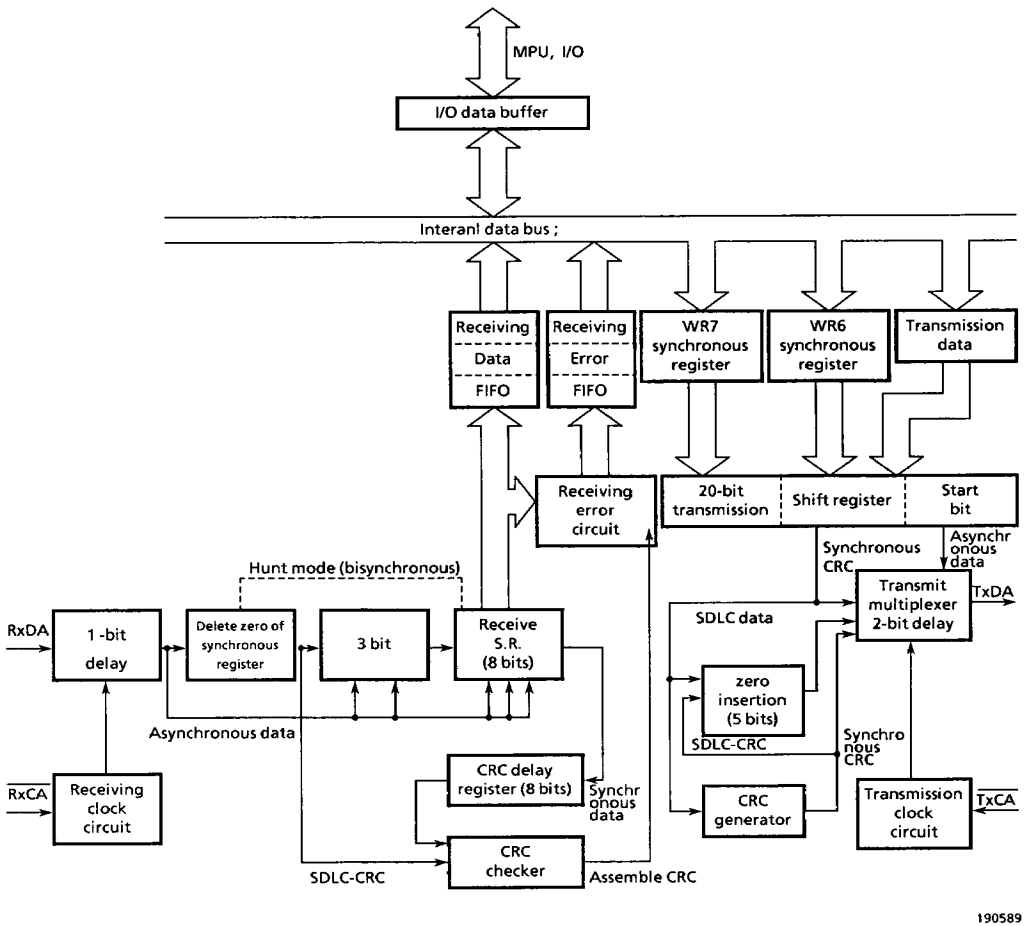


Figure 3.5.3 Transfer Data Path (Channel A)

- Asynchronous mode

  In the asynchronous mode, the receive data enters the 3-bit buffer if the character length is 7 or 8 bits or the 8-bit receive shift register if the character length is 5 or 6 bits.

- Synchronous mode

  In the synchronous mode, the data path depends on the receive processing phase at the time. The receiver operation starts from the hunt phase. In this mode, the receiver searches the receiver data for the bit pattern which matches the specified sync character. If the SIO is set in the monosync mode, the receiver searches for the bit pattern which matches the sync character set in WR7; if the SIO is set in the bisync mode, the receiver searches for the bit pattern which matches two consecutive sync characters set in WR6 and WR7. When synchronization has been established, the subsequent data enter the 3-bit buffer by bypassing the sync register.

- SDLC mode

  In the SDLC mode, the sync register constantly monitors the receive data perform-ing zero deletion as required. When the sync register detects 5 "1"s consecutively in the receive data, the following bit is deleted if it is "0". If it is "1", the bit that follows is checked. If it is "0", it is assumed as a flag, if it is "1", it is assumed an abort sequence (7 consecutive "1"s).

  The reformatted data are put in the receive shift register via the 3-bit buffer. When synchronization has been established, the subsequent data follow the same path regardless of the character length.

2 Transmission

The transmitter has an 8-bit transmit data register and a 20-bit transmit shift register. The 20-bit transmit shift register holds the data from the WR6, WR7, and transmit data register.

- Asynchronous mode

  In the asynchronous mode, the data in the 20-bit transmit shift register are added with the start and stop bits to be sent to the transmit multiplexer.

- Synchronous mode

  In the synchronous mode, the WR6 and WR7 hold the sync character. The contents of these registers are sent to the 20-bit transmit register as the sync character at the transmission of data blocks or as the idle sync character if a transmitter underrun occurs in data block transmission.

- SDLC mode

  In the SDLC mode, the WR6 holds the station address and the WR7 holds the flag. The flag (WR7) is sent to the 20-bit transmit register at the start and end of each frame. For each of the other data fields, one "0" follows five consecutive "1"s.

(2) I/O functions

To transfer data from/to the MPU, the SIO must be set in the polling, interrupt, or block transfer mode.

- Polling

  To operate the SIO in the polling mode, all interrupts mode must be disabled. In the polling mode, the MPU reads the status bits D0 and D2 in each channel's RR0 to check for reception or transmission.

- Interrupts

  There are 3 types of SIO interrupt: transmit interrupt, receive interrupt, and external/status interrupt. These interrupts can be enabled by program. The receive interrupt is further divided into the following three:

  - Interrupt on the first received character
  - Interrupt on all received characters
  - Interrupt on special receive conditions

  Higher priority is given to channel A than channel B. On the same channel, higher priority is given to reception, transmission, and external/status in this order.

  The SIO provides the daisy-chained interrupt priority control feature and the interrupt vector generating feature. Further, it provides the "status affected vector" feature. This feature outputs 4 interrupts depending on the interrupt source.

- Block transfer

  The SIO has the block transfer mode to adapt to the MPU"s block transfer and the DMA controller. For block transfer, the $\overline{W/RDY}$ line is used. For the MPU's block transfer, this line is used as the wait line; for the DMA block transfer, it is used as the ready line. The SIO's ready output indicates to the DMA controller that the data is ready to transfer. The SIO's wait output indicates to the MPU that the SIO is not ready for data transfer and therefore requesting the extension of the output cycle.

3.5.3 SIO Basic Operations

(1) Asynchronous mode

For data transfer in the asynchronous mode, the character length, clock rate, and interrupt mode must be set. These parameters are written in the write registers. Note that WR4 must be set before the other registers are set.

Data transfer does not start until the transmit enable bit is set. When the auto enable bit is set, the SIO starts transmission upon the $\overline{CTS}$ pin's going "0", allowing the programmer to send a message to the SIO without waiting for the $\overline{CTS}$ signal. Figure 3.5.4 shows the data format of the asynchronous mode.

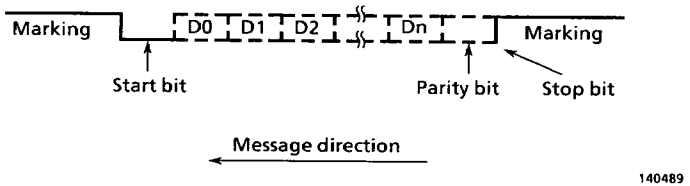


Figure 3.5.4 Data Format of Asynchoronous Mode

1 Transmission

Serial data are output from the TxD pin. Its transfer clock rate can be set to one of 1, 1/16, 1/32, and 1/64 times the clock rate to be supplied to the transmit clock input ($\overline{TxC}$). The serial data are output on the falling edge of $\overline{TxC}$.

2 Reception

The receive operation in the asynchronous mode starts when the receive enable bit (D0 of WR3) is set. When the receive data input RxD is set to "0" for the duration of at least 1/2 bit time, the SIO interprets it as the start bit, sampling the input data at the middle of the bit time. The sampling is performed on the rising edge of the $\overline{RxC}$ signal.

When the receiver receives the data whose character length is not 8 bits, it converts the data into the one composed of the necessary bits, the parity bit and the unused bit set to "1".

Example : a 6-bit character

1	P	D5	D4	D3	D2	D1	D0

When the external/status interrupt is enabled and a break state is detected in the receive data, the interrupt is generated and the break/abort status bit (D7 of RR0) is set and the SIO monitors the transmit data until the break state is cleared. The interrupt is also generated when the $\overline{DCD}$ signal is in the inactive state for more than the specified pulse width. The DCD status bit is set to "1".

In the polling mode, the MPU must refer the receive character valid bit (D0 of RR0) to read the data. This bit is automatically reset when the receive buffer is read. In the polling mode, the transmit buffer status must be checked before writing data in the transmitter to avoid overwrite.

(2) Synchronous mode

There are 3 kinds of character synchronization : monosync, bisync, and external sync. In each of these synchronous modes, the times 1 clock rate is used for both transmission and reception. The receive data is sampled on the rising edge of the receive clock input ($\overline{RxC}$). The transmit data changes on the falling edge of the transmit clock input.

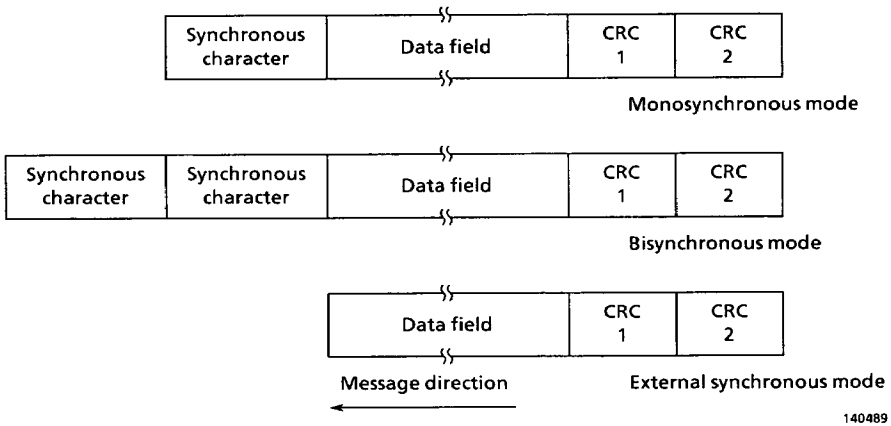


Figure 3.5.5 Data Format of Synchronous Mode

1 Monosync

In this mode, synchronization is established when a match with the sync character (8 bits) set to WR7 is found, enabling data transfer.

2 Bisync

In this mode, synchronization is established when a match with 2 consecutive sync characters set to WR6 and WR7 is found, enabling data transfer. In this mode as well as the monosync mode $\overline{SYNC}$ is active during the receive clock period in which the sync character is being detected.

3 External sync

In this mode, synchronization is performed externally. When synchronization is established, it is indicated by the $\overline{SYNC}$ pin. The $\overline{SYNC}$ input must be kept to "0" until the character synchronization is lost. Character assembly starts from the rising edge of the $\overline{RxC}$ after the falling of the $\overline{SYNC}$.

After reset, the SIO enters the hunt phase to search for the sync character. If synchronization is lost, the SIO sets the enter-hunt-phase-bit (D4 of WR3) to reenter the hunt phase.

- Transmission

(a) Data transfer using interrupt

When the transmit interrupt is enabled, the interrupt is caused upon the transmit buffer's being emptied. For the interrupt processing, other data are written in the transmitter. If these data are not ready for some reason, the transmit underrun condition occurs.

(b) Bisync mode

In the bisync mode, if the transmitter runs out of data during transmission, supply characters are inserted. This is done in two methods. In one method, sync characters are inserted. In the other, characters generated so far are transmitted followed by sync characters. Either of these methods can be selected by the reset transmit underrun/EOM command in WR0.

(c) End of transmission

Break can be performed by setting bit D4 of WR5. When break is performed, the data in the transmit buffer and the shift register are lost. When the external/status interrupt is enabled, the SIO generates the interrupt depending on the transmitter state and outputs the vector. This mode can be used for block transfer.

- Reception

(a) Interrupt on the first received character

This mode is used for ordinary block transfer. In this mode, the SIO generates the interrupt only for the first character;subsequently, it does not generate the interrupt unless special receive conditions are satisfied.

To initialize these settings, command 4 of WR0 (to be enabled by the next receive interrupt) must be set in advance.

(b) Interrupt on all received characters

In this mode, the SIO generates the interrupt for all characters coming into the receive buffer. When the status affect vector has been set, a special vector is generated on a special receive condition.

(c) Special receive condition interrupt

This interrupt occurs when any of the above interrupts is selected. The special receive conditions include parity error, receive overrun error, framing error, and end-of-frame (SDLC). These error status bits are latched, so that they must be reset after they are read. They can be reset by command 6 of WR0 (error reset).

(3) SDLC mode

The SIO supports both the SDLC and HDLC protocols. They resemble each other, so that only the SDLC mode is explained here.

Figure 3.5.6 shows the data format in the SDLC mode. In the SDLC mode, one data block is called a frame and the message in it is put between the open flag and the close flag. The address field in the frame contains the address of a secondary station. Checking this address, the SIO receives or ignores the frame.

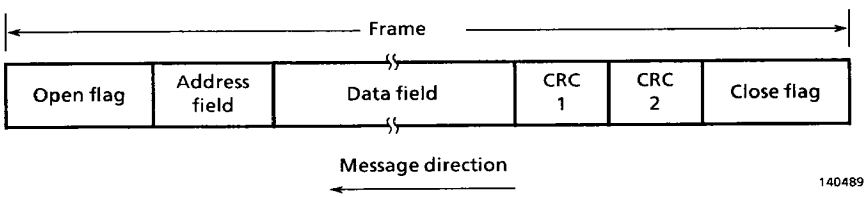


Figure 3.5.6 Data Format in SDLC Mode

- Transmission

(a) Data transfer using interrupt

When the transmit interrupt has been set, the interrupt occurs each time the transmit buffer becomes empty. In the SDLC mode, data are sent to the SIO by this interrupt.

(b) Data transfer using wait/ready

The wait function in the wait/ready capability is used to make the MPU extend the output cycle when the SIO's transmit buffer is not empty. The ready function indicates to the DMA that the SIO's transmit buffer is empty and therefore ready to receive data. If no data has been written in the transmit shift register before transmission, the SIO goes in the underrun state. This capability permits data transfer to the SIO.

(c) Transmit underrun/EOM

The SIO automatically ends the SDLC frame if there is no data to be transmitted to the transmit data buffer. To implement this, the SIO sends a 2-byte CRC when there is no data to send, then the SIO transmits one or more flags. After reset, the transmit underrun/EOM status bit is set to prevent the CRC character from being inserted when there is no data to be sent. Using this function, the SIO starts frame transmission. Here, the transmit underrun/EOM reset command must be set in advance between the transmission of the first data and the data end. Thus, the SIO goes in the reset state at the end of each message with the CRC character being sent automatically.

(d) CRC generation

For CRC calculation, the CRC generator must be reset before transmission (bits D6 and D7 of WR0). CRC calculation starts when the address field is written in the SIO (WR6). The transmit CRC enable bit (D0 of WR5) must be set before the address field is written.

(e) End of transmission

When the transmitter is disabled during transmission, the data currently transmitted is all transmitted to its end. The subsequent data is put in the marking state. When the transmitter is disabled, characters remain in the buffer. However, the abort sequence is made active when the abort command is written in the command register, deleting all data.

- Reception

As in the transmit mode, several parameters must be preset in the receive mode. The address field is written in WR7 and the flag character in WR7. Receiving the open flag, the receiver compares the contents of the following address field with the address set in WR6 or the global address ("1111 1111"). If the contents of the address field in frame matches either of these address, the SIO starts reception.

(a) Interrupt on the first received character

This mode is generally used for the block transfer using the wait/ready capability. In this mode, the SIO generates the interrupt only on the first character. The status flag of this interrupt is latched, so that command 4 (to be enabled by the next received character) of WR0 must be preset for re-initialization. When the external/status interrupt is set, an interrupt occurs every time the $\overline{DCD}$ changes. This interrupt also occurs when the special receive condition is satisfied.

(b) Interrupt on all received characters

In this mode, the SIO generates an interrupt on all received characters. When the status affect vector has been set, the SIO generates a special vector on the special receive condition interrupt.

(c) Special receive condition interrupt

Using the special receive condition, the interrupt on the first received character or the interrupt on all received characters must be selected in advance. The receive overrun status of the special receive condition interrupt is latched. The status bit can be reset by the error reset command (WR0 command).

(d) CRC check

The receive CRC check is reset when the open flag at the head of a frame is received. CRC calculation is performed on the subsequent characters up to the close flag. In the SDLC mode, the transmit CRC is inverted, so that a special check sequence is used. The check must end with "0001 1101 0000 1111." Since SIO handles the CRC character as a data, the MPU must discard it after reading it.

(e) End of transmission

When the SIO receives the close flag, the end-of-frame-bit is set to indicate that the close flag has been received. When the status affect vector has been set, the special receive condition interrupt occurs and the interrupt vector is output. Any frame can be aborted by abort transmission. When the external/status interrupt has been set, the interrupt occurs and the break/abort bit in RR0 is set.

3.5.4 SIO Status Transition Diagram and Basic Timing

[1] Status Transition Diagram

Figure 3.5.7 shows the SIO status transition diagram.

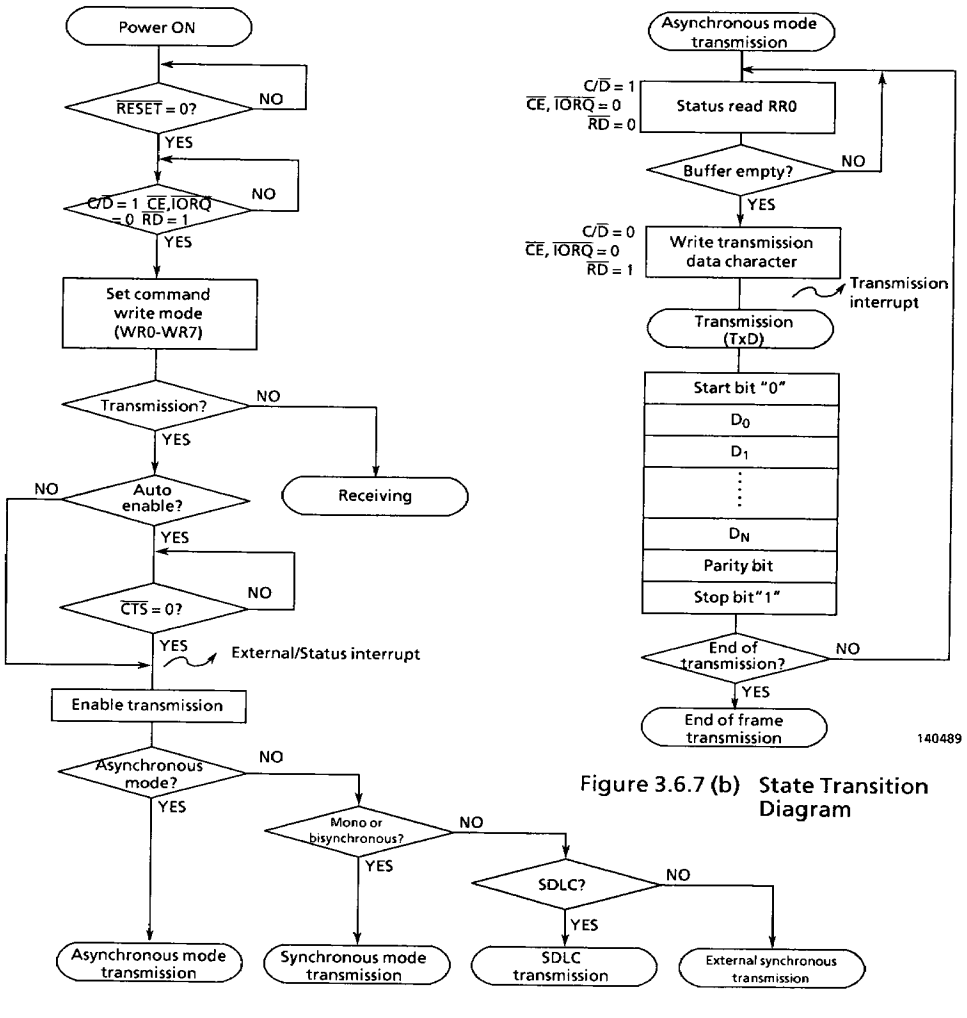


Figure 3.6.7 (b) State Transition Diagram

Figure 3.5.7 (a) SIO Status Transition Diagram

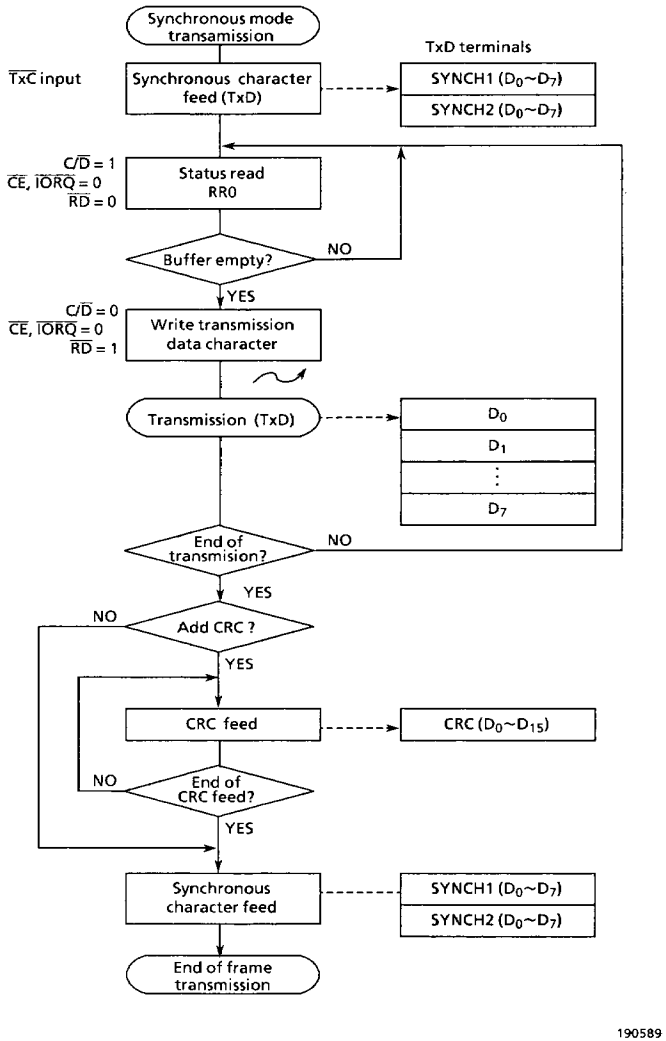


Figure 3.6.7 (c) State Transition Diagram

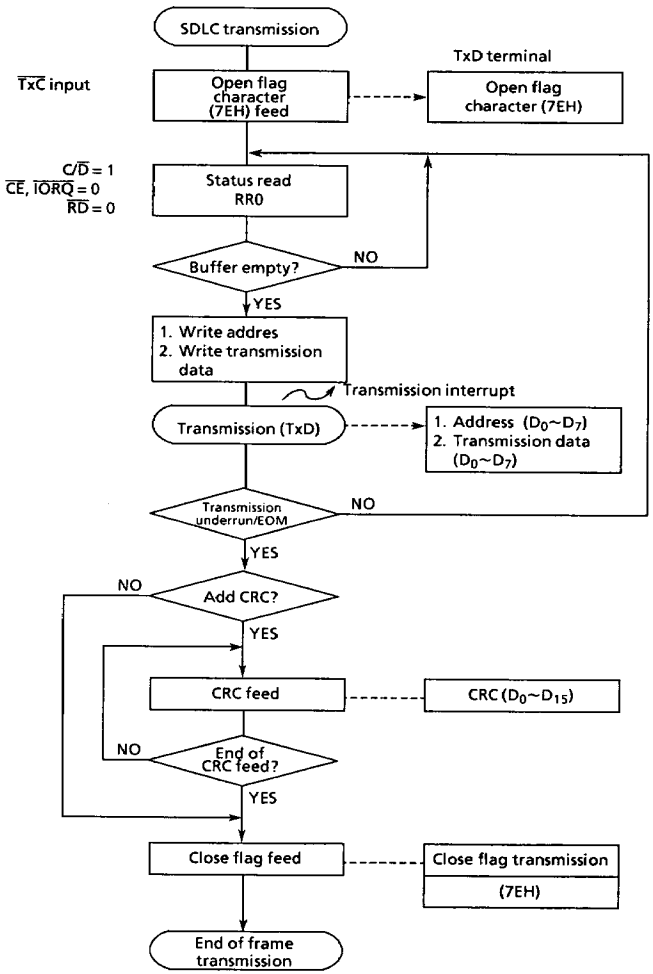


Figure 3.6.7 (d) SIO Status Transition Diagram

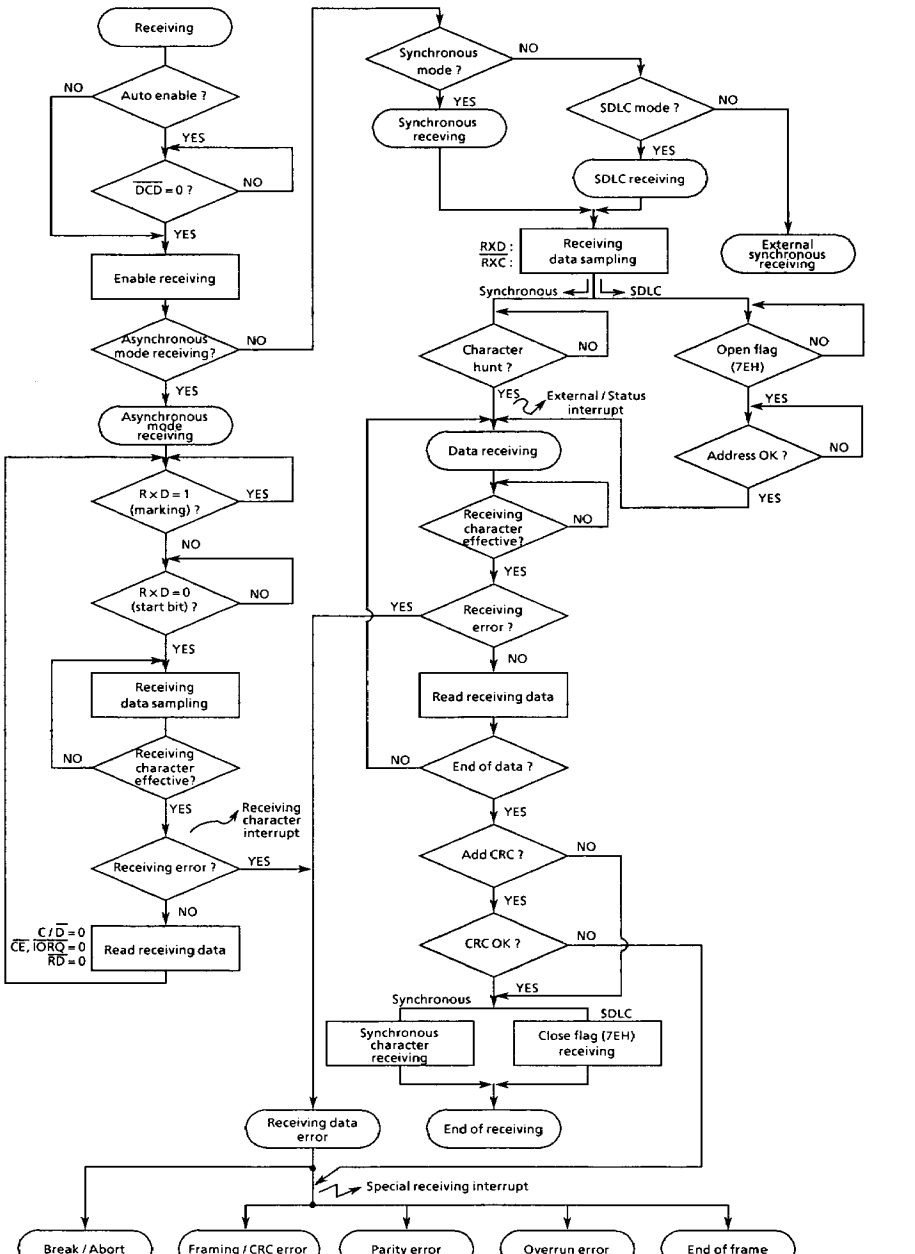


Figure 3.6.7 (e) State Transition Diagram

[2] Basic Timing

Figure 3.5.8 shows the timing in which data or a command is written from the MPU to the SIO. Figure 3.5.9 shows the timing in which data is read from the SIO to the MPU. Figure 3.5.10 shows the interrupt acknowledge timing in which the MPU gives an interrupt response to the SIO's interrupt request to set the $\overline{IORQ}$ pin to "0" several clocks after setting the $\overline{M1}$ pin to "0" as the acknowledge signal. To maintain the interrupt serviced state in daisy chain structure, the interrupt request state cannot be changed while $\overline{M1}$ is active.

Figure 3.5.11 shows the timing in which the return from interrupt is performed.

Figure 3.5.12 shows how the daisy chain structure works. First, suppose that the SIO is servicing interrupt. When the PIO issues an interrupt request immediately before the first byte "EDH" of the RETI instruction is decoded with $\overline{M1}$ being active, "IEO" of the PIO goes "0". However, when "EDH" is decoded, the PIO's interrupt request is not acknowledged. Therefore, the PIO's "IEO" returns to "1". When the second byte "4DH" is decoded, the SIO's "IEO" returns to "1". Therefore, the "IEI" and "IEO" of each peripheral LSI at this point of time all go "1", or out of the interrupt serviced state. The PIO keeps the $\overline{INT}$ pin at "0" until this state is set. Then, the interrupt is serviced starting with the peripheral device of the higher priority.

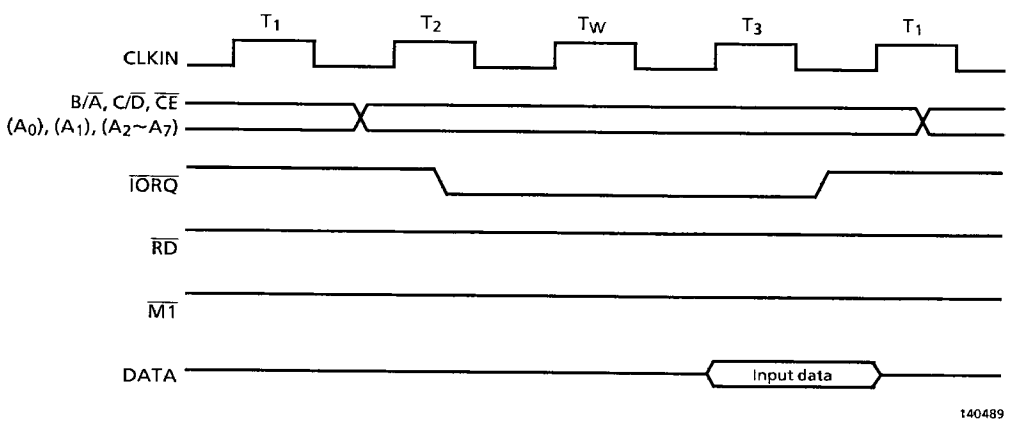


Figure 3.5.8 Write Timing

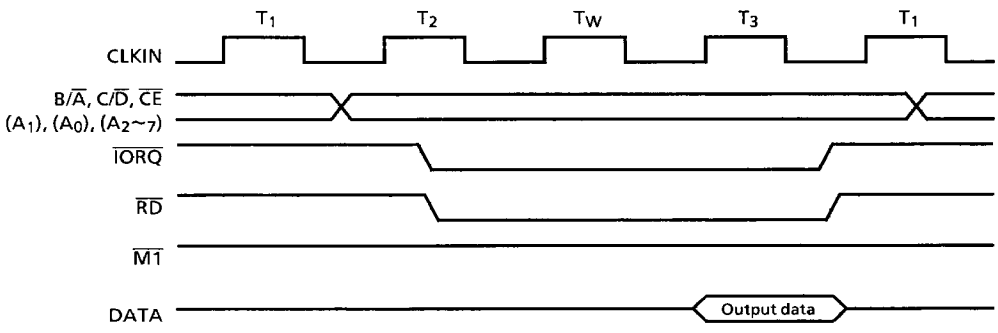


Figure 3.5.9 Read Timing

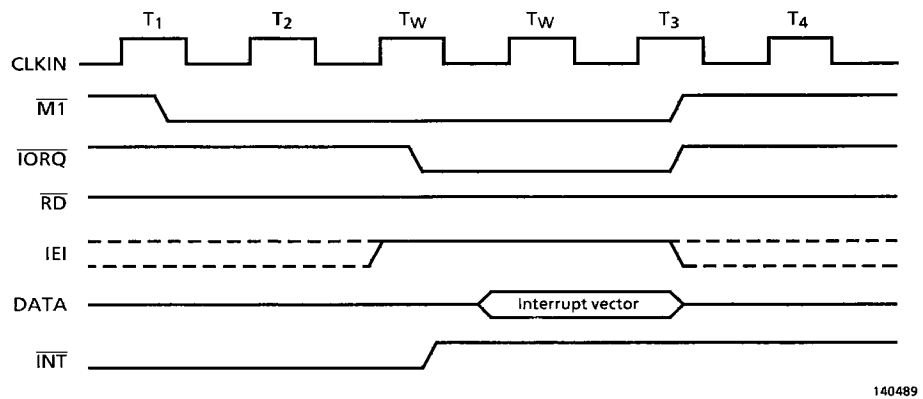


Figure 3.5.10 Interrupt Acknowledge Timing

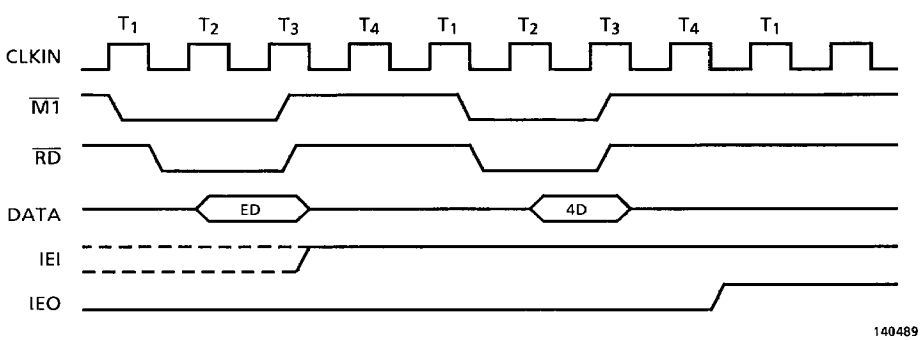
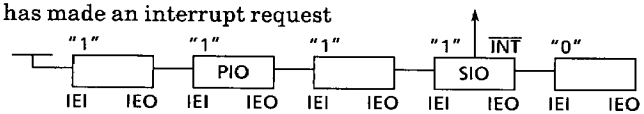
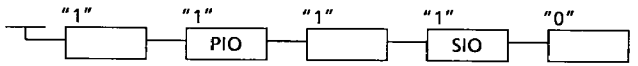


Figure 3.5.11 Return Timing from Interrupt

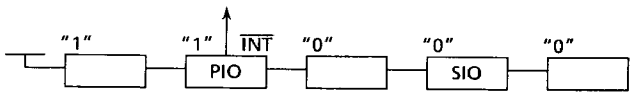
1 The SIO has made an interrupt request



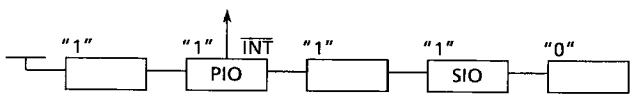
2 The SIO is servicing the interrupt.



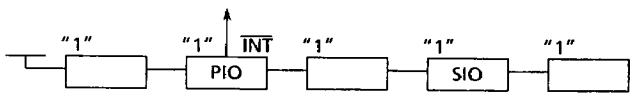
3 The PIO has made an interrupt request immediately before "EDH" is decoded by the SIO. By the PIO's interrupt request, PIO's IEO is set to "0".



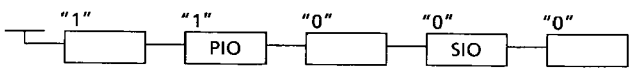
4 Because "EDH" has been decoded, the PIO's interrupt request is not acknowledged. Therefore, PIO's IEO returns to "1".



5 Because "4DH" has been decoded, the SIO's IEO is set to "1".



6 The PIO's interrupt request is acknowledged and the PIO's IEO is set to "0".



140489

Figure 3.5.12 Daisy Chain at Execution of RETI Instruction

3.5.5 SIO Operational Procedure

The following mainly describes the meaning of each bit of the write and read registers. Special attention should be directed to the fact that the parameters of the write register (WR4) should be set before the others.

Some registers can use only a signal channel. The I/O addresses listed in Table 3.5.2 must be specified to write the control word and read/write data on the SIO.

Table 3.5.2 I/O Addresses

I/O function	I/O address
Channel A data	#18
Channel A command	#19
Channel B data	#1A
Channel B command	#1B

140489

[1] Write Registers

1 WR 0; Write register 0

Table 3.5.3 Configuration of Write Register 0

D7	D6	D5	D4	D3	D2	D1	D0
CRC reset code		Primary command bit			Register pointer bit		

140489

Bits D0 through D2: Register pointer bits

These bits specify the register on which read/write is performed by the next byte. When read/write is completed, the register pointer points to WR0.

Bits D3 through D5: Basic command bits

- Command 0 (=000): No operation

  This command only sets the register pointer without making the SIO operate. It is used to invalidate the command in the command chain for the SIO or hold the location at which a command is inserted in the command chain if required.

- Command 1 (=001): Abort sequence generation

  This command is used to generate the abort sequence (7 or more consecutive "1"s). Note that command 1 is used only in the SDLC.

- Command 2 (=010): External/status interrupt reset

  Once an external interrupt or a status interrupt has occurred, the status bit of RRO is latched. This command is issued to enable the RR0's status bit in order to enable the interrupt again.

- Command 3 (=011): Channel reset

  This command performs generally the same operation as when the $\overline{\text{RESET}}$ pin is set. The difference is that reset is performed only on a single channel. The command for channel A resets the interrupt priority circuit as well.

- Command 4 (=100): Enable the interrupt at the next character reception.

  This command is used to enable an interrupt when the end of data block has been detected followed by the reception of the next block.

- Command 5 (=101): Reset transmit interrupt pending

  If the transmit buffer becomes empty in the transmit interrupt enable mode, an interrupt occurs. This command is used to disable the transmit interrupt when there is no data in the transmit buffer.

- Command 6 (=110): Error reset

  The error (parity or overrun error) caused in block transfer is latched in bits D4 and D5 of RR1. This commands is used to clear these bits.

- Command 7 (=111): Return from interrupt

  This command performs the same operation as the operation required to execute the RETI instruction on the SIO's data bus. Therefore, non-Z80 MPUs (that is, systems using no RETI instruction) can use the daisy chain in the SIO. This command is available only on channel A.

Bits D6 and D7: CRC reset code

These 2 bits allow the programmer to select between the receive CRC checker reset, the transmit CRC generator reset, and the transmit underrun/EOM reset.

Table 3.5.4 List of Reset Command Codes

Rest command	D7	D6
No operation	0	0
Reset the receive CRC checker	0	1
Reset the transmit CRC generator	1	0
Reset the transmit underrun/EOM	1	1

140489

2 WR 1; Write register 1

Table 3.5.5 Configuration of Write Register 1

D7	D6	D5	D4	D3	D2	D1	D0
Enable	Wait/ready Select function	Select receiving/ transmission	Receiving interrupt mode		Status -affect vector	Enable transmission interrupt	Enable external/ status interrupt

140489

Bit D0: External/status interrupt enable

When this bit is set, an interrupt is generated at the start of sync character transmission even if the execution is terminated upon detection of break/abort, the $\overline{DCD}$, $\overline{CTS}$ or $\overline{SYNC}$ signal has changed, or the transmit underrun/EOM latch is set.

Bit D1: Transmit interrupt enable

When this bit is set, a transmit interrupt is generated upon the transmit buffer becoming empty.

Bit D2: Status affect vector

When this bit is set, bits D1 through D3 (V1 through V3) of WR2 is changed. When this bit is not set, the same interrupt vector as the contents of WR2 issued. Note that this bit is available only on channel B.

Bits D3 and D4: Receive interrupt mode

These bits are used to select a receive interrupt mode.

Bits D5 through D7: Selection wait/ready functions

These 3 bits are used to select a $\overline{W/RDY}$ pin function. The wait or the ready function is selected by program and they are not used simultaneously. The meaning of these bits are:

- When D5 is set to "1", it indicates that the $\overline{W/RDY}$ pin responds to the receive buffer; when D5 is reset to "0", it indicates that the pin responds to the transmit buffer.
- When D6 is set to "1", the $\overline{W/RDY}$ pin functions as the $\overline{READY}$ pin; when D6 is reset to "0", the pin functions as the $\overline{WAIT}$ pin.
- When D7 is set to "1", the wait/ready function is enabled; when D7 is reset to "0", the function is disabled.

For example, when D7, D6, and D5 are "1", "1", and "0" respectively, and the transmit buffer is full, the $\overline{READY}$ pin goes "1"; when the transmit buffer is empty, the pin goes "0".

Table 3.5.6 shows the summary of the above description of bits D3 and D4 and D5 through D7.

Table 3.5.6 List of Receive Interrupt Mode Codes

Receive interrupt mode	D4	D3
Receive interrupt disable	0	0
Interrupt on first received character or special receive condition*	0	1
Interrupt on received character or special receive condition*	1	0
Interrupt on received character or special receive condition* (except for parity error)	1	1

140489

* Special receive conditions:
- End of frame (in SDLC mode only)
- Receive overrun error
- Parity error
- Framing error

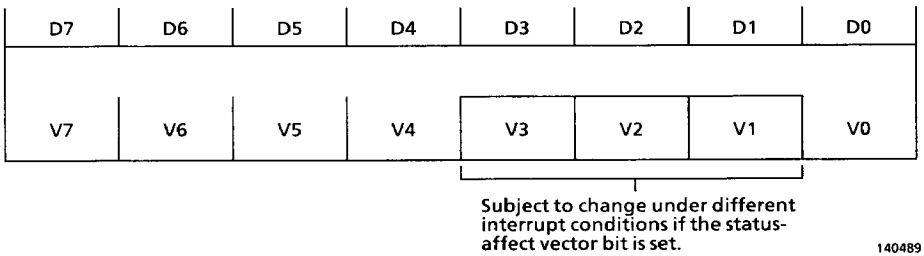
Table 3.5.7 Wait/Ready Select Function (D5 through D7)

Pin state			Buffer state	D7	D6	D5
Pin (Function)		Pin output				
DISABLE	$\overline{WAIT}$	Floating	–	0	0	–
	$\overline{READY}$	High	–		1	
ENABLE	$\overline{WAIT}$	Low	The transmit buffer is full and the SIO data port is selected.	1	0	0
		Floating	The transmit buffer is empty.			
	$\overline{READY}$	High	The transmit buffer is full.		1	
		Low	The transmit buffer is empty.			
	$\overline{WAIT}$	Floating	The receive buffer is full.		0	1
		Low	The receive buffer is empty and the SIO data port is selected.			
	$\overline{READY}$	Low	The receive buffer is full.		1	
		High	The receive buffer is empty.			

140489

3 WR 2; Write register 2

Table 3.5.8 Configuration of Write Register 2



This write register is the interrupt vector register. When bit D2 of WR1 (B channel) is not set, the interrupt vector is issued. When bit D2 of WR1 (B channel) is set, bits D1 through D3 (V1 through V3) are changed depending on the interrupt generation condition. This time, the contents of WR2 remain unchanged. Because WR2 is available only on channel B, WR2 must be programmed even if only channel A of the SIO is used.

Table 3.5.9 shows the WR2 bit states in the interrupt condition with the status affect vector being set.

Table 3.5.9 Channel Interrupt Condition Codes

Channel	Interrupt condition	V3	V2	V1
B	Transmit buffer empty	0	0	0
	Change of external/status	0	0	1
	Received character available	0	1	0
	Special receive condition*	0	1	1
A	Transmit buffer empty	1	0	0
	Change of external/status	1	0	1
	Received character available	1	1	0
	Special receive condition*	1	1	1

140489

* Special receive conditions:
- End of frame (in SDLC mode only)
- Receive overrun error
- Parity error
- Framing error

4 WR 3; Write register 3

Table 3.5.10 Configuration of Write Register 3

D7	D6	D5	D4	D3	D2	D1	D0
Receiving bit /character		Auto enable	Enter hunt phase	Enable receiving CRC	Address search mode	Prohibit synchronous character load	Enable receiving

140489

Bit D0: Receive enable

When this bit is set, the receive operation starts. Because this bit is used to start the receive operation, it must be set after the receive-associated programming has been all completed.

Bit D1: Sync character load inhibit

When this bit is set in the sync mode, the sync character is not loaded into the receive buffer. This bit is used to remove the sync character and idle sync from the received characters.

Bit D2: Address search mode

When this bit is set in the SDLC mode, any message having a programmed address or an address other than the global address (FFH) is not received by WR6. Therefore, the receive interrupt does not occur unless an address match occurs.

Bit D3: Receive CRC enable

When this bit is set, CRC calculation starts at the start of the last data transfer from the receive shift register to the receiver buffer.

Bit D4: Enter hunt Phase

When the establishment of synchronization is required, set this bit to enter the SIO into the hunt phase. The hunt phase is automatically cleared upon establishment of synchronization.

Bit D5: Auto enable

When this bit is set, the transmitter is enabled at the time the $\overline{CTS}$ pin is "0". When the $\overline{DCD}$ pin is "0", the receiver is enabled.

Bits D6 and D7: Receive character length

These bits are used to specify the number of receive bits which make up one character (character length). Table 3.5.11 shows the number of bits per character.

Table 3.5.11 Receive Character Length Codes

Bits/character	D7	D6
5	0	0
7	0	1
6	1	0
8	1	1

140489

5 WR 4; Write register 4

Table 3.5.12 Configuration of Write Register 4

D7	D6	D5	D4	D3	D2	D1	D0
Clock mode		Synchronous mode		Stop bit		Parity Even/$\overline{\text{Odd}}$	Enable

140489

Bit D0: Parity enable

When this bit is set, 1-bit transmit data is added to the number of bits specified by D6 and D7 of WR3 and the data is received in the resulting number of bits. If a character length other than 8 bits is selected, the added parity bit is set to the MSB side to be transferred to the receive data FIFO. When the 8-bit character length is selected, the parity bit is not transferred to the receive data FIFO.

Bit D1: Parity even/odd

This bit is used to determine whether to perform transfer and check in even or odd parity. (Even parity = "1", odd parity = "0")

Bit D2 and D3: Stop bit length

These bits are used to select the stop bit length in the asynchronous mode. In the synchronous mode, both D2 and D3 must be set to "0".

Table 3.5.13 Stop Bit Length Codes

top bit	D3	D2
Sync mode	0	0
1 stop bit/character	0	1
1.5 stop bits/character	1	0
2 stop bits/character	1	1

140489

Bits D4 and D5: Sync mode

These bits are used to select the sync mode.

Table 3.5.14 Sync Mode Codes

Sync mode	D5	D4
8-bit sync mode	0	0
16-bit sync mode (bisync mode)	0	1
SDLC mode (flag character; 7EH)	1	0
External sync mode	1	1

140489

Bits D6 and D7: Clock mode

These bits are used to select the factor between the transmit/receive clock and the data transfer rate. In the synchronous mode, the ×1 clock mode must be set. In the asynchronous mode, the transmit side and the receive side must have the same factor.

Table 3.5.15 Clock Mode Codes

Clock mode (data transfer rate)	D7	D6
× 1 data transfer rate	0	0
× 16 data transfer rate	0	1
× 32 data transfer rate	1	0
× 64 data transfer rate	1	1

140489

6 WR 5; Write register 5

Table 3.5.16 Configuration of Write Register 5

D7	D6	D5	D4	D3	D2	D1	D0
DTR	Transmit bit /character		Break transmission	Enable transmission	CRC-16 /SDLC	RTS	Enable CRC transmission

140489

Bit D0: Transmit CRC enable

When this bit is set at the time the transmit data is loaded from the transmit data buffer into the transmit shift register, the CRC calculation is performed on that data. If this bit is not set, the CRC calculation and transmission are not performed in the transmit underrun state in the synchronous or SDLC mode.

Bit D1: Request to send

When this bit is set, the $\overline{RTS}$ pin goes "0". When this bit is not set, the $\overline{RTS}$ pin goes "1". In the asynchronous mode, the $\overline{RTS}$ pin goes "1" when the transmit buffer becomes empty. In the synchronous or SDLC mode, this bit state is followed by the $\overline{RTS}$ pin state.

Bit D2: CRC-16 / SDLC

When this bit is set, the CRC-16 polynomial ($X^{16}+X^{15}+X^2+1$) is selected. When this bit is reset to "0", the CRC-CCITT polynomial ($X^{16}+X^{12}+X^5+1$) is selected.

Bit D3: Transmit enable

When this bit is set, the transmitter is enabled. Even if this bit is reset to "0" after the start of transmission, the sync character and the data being transmitted are transmitted to the last.

Bit D4: Transmit break

When this bit is set, transmitting any data forcibly puts the transmit data line (TxD pin) in the space state. When this bit is reset to "0", the TxD pin is put in the marking state.

Bits D5 and D6: Transmit character length

These bits indicate the character length of transmit data.

Table 3.5.17 Transmit Character Length Codes

Bits/character	D6	D5
Less than 5 bits	0	0
7 bits	0	1
6 bits	1	0
8 bits	1	1

140489

As shown in Table 3.5.17, for the transmission of less than 5 bits (4 bits or 3 bits) per character, D6 and D5 are "0" and "0", which do not indicate how many bits the transmit data consists of. To solve this problem, the data characters must be processed by the format shown in Table 3.5.18. Note that D indicates data.

Table 3.5.18 Data Transfer Format with Transmit Data Consisting of Less than 5 bits

Transmit bits/character	D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	1	1	0	0	0	D
2	1	1	1	0	0	0	D	D
3	1	1	0	0	0	D	D	D
4	1	0	0	0	D	D	D	D
5	0	0	0	D	D	D	D	D

140489

Bit D7: Data terminal ready

This bit indicates the $\overline{DTR}$ pin state. When this bit is set, the $\overline{DTR}$ pin goes "0", when it is reset, the $\overline{DTR}$ pin goes "1".

7 WR 6; Write register 6

Table 3.5.19 Configuration of Write Register 6

D7	D6	D5	D4	D3	D2	D1	D0
SYNC							
7	6	5	4	3	2	1	0

140489

This register is programmed as follows:

- In the external sync mode : Transmit sync character
- In the monosync mode : Transmit sync character
- In the bisync mode : First sync character
- In the SDLC mode : Slave station address

8 WR 7; Write register 7

Table 3.5.20 Configuration of Write Register 7

D7	D6	D5	D4	D3	D2	D1	D0
SYNC							
15 (7)	14 (6)	13 (5)	12 (4)	11 (3)	10 (2)	9 (1)	8 (0)

140489

This register is programmed as follows:

- In the monosync mode : Receive sync character
- In the bisync mode : Second sync character
- In the SDLC mode : Flag character (7EH)

This register is not used in the external sync mode.

[2] Read Registers

1 RR 0; Read register 0

Table 3.5.21 Configuration of Read Register 0

D7	D6	D5	D4	D3	D2	D1	D0
Break/ Abort	Trasmission underrun/ EOM	CTS	Synchronize /Hunt	DCD	Trasmission buffer empty	Interrupt pending	Receiving character effective

140489

Used with the external/status interrupt

Bit D0: Receive character available

This bit is set when the receive buffer holds characters of 1 byte or more. This bit is reset when the buffer becomes empty.

Bit D1: Interrupt pending

This bit is set when an interrupt occurs in the SIO regardless of the interrupt condition type. This bit is available only on channel A.

Bit D2: Transmit buffer empty

This bit is set when the transmit data buffer becomes empty or the SIO is reset. However, in the sync and SDLC modes where the CRC character is being transmitted, bit D2 is reset.

Bit D3: Data carrier detect

This bit indicates the $\overline{DCD}$ pin input state. This bit is latched when the external/status interrupt occurs.

Bit D4: Sync/hunt

The meaning of this bit depends on the operation mode:

(i) Asynchronous mode

Bit D4 indicates the SIO's $\overline{SYNC}$ pin state. When the $\overline{SYNC}$ pin state changes, the external/status interrupt occurs.

(ii) External sync mode

When synchronization has been established by the detection of external synchronization, the last bit of the sync character must be set to "0" at the second $\overline{RxC}$ falling edge from the rising edge of the received $\overline{RxC}$. That is, to set the $\overline{SYNC}$ input to "0" by the external circuit after the detection of synchronization, full 2 receive cycle clocks must be awaited.

When the $\overline{SYNC}$ input goes "0", the sync hunt bit is set. When synchronization is lost or the end of message is detected, the enter hunt phase bit is set.

(iii) Internal sync mode

In the monosync and bisync modes, bit D4 is initialized to "1" by the enter hunt phase command (D4 of WR3). This bit is reset when the SIO detects the sync character.

(iv) SDLC mode

Bit D4 is set when the receiver is disabled or the enter hunt phase command is issued. Then, when the frame open flag is detected, this bit is reset.

Bit D5: Clear to send

This bit indicates the opposite of the $\overline{CTS}$ pin input state.

Bit D6: Transmit underrun/EOM

This bit is set when the SIO is reset (including channel reset). Only the reset transmitter underrun/EOM latch command (WR0 bits D7, D6="1", "1") can reset this bit. When the transmit underrun state occurs, the external/status interrupt is generated. Bit D5 is also used to control transmission in the sync or SDLC mode.

Bit D7: Break/abort

In the asynchronous mode in reception, this bit indicates the break state detection. When the break state is detected, this bit is set, generating the external/status interrupt. This bit is reset by the external/status interrupt reset command.

After break, the external/status interrupt is generated again. In the SDLC mode, bit D7 is set when the abort sequence is detected, generating the external/status interrupt.

2 RR 1; Read register 1

Table 3.5.22 Configuration of Read Register 1

D7	D6	D5	D4	D3	D2	D1	D0
End of frame	framing error	Receiving overrun error	Parity error	Fraction			Feed all characters

140489

Bit D0: All sent

In the asynchronous mode, this bit is set when all characters are sent from the transmitter or there is no transmit data in the SIO. In the synchronous mode, this bit is always set.

Bits D1 through D3: Fraction codes

Normally, I field is an integral multiple of character length. If it is not, these bits show the number of fraction bits. These codes are effective only for the transmission for which the end of frame bit is set in the SDLC mode.

Example : Figure 3.5.13 shows examples of fractions in which the number of bits/character at the end of I field is 8 bits (1) and 4 bits (2).

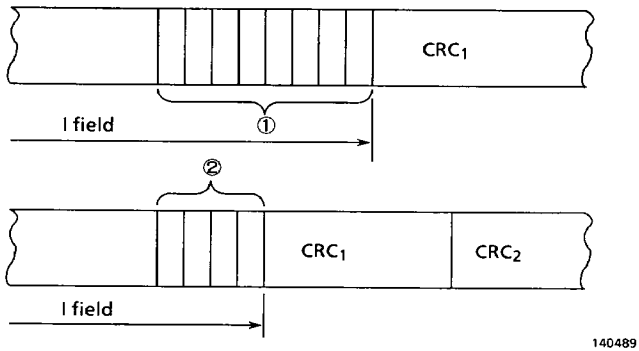


Figure 3.5.13 Examples of Fraction Bits in I Field

Table 3.5.23 (a) shows the fraction codes for the receive character whose character length is 8 bits.

Table 3.5.23 (a) Bit Patterns by Fraction Bits at End of I Field

Number of fraction bits at end of I field		D3	D2	D1
1 byte before	2 bytes before			
0	3	1	0	0
0	4	0	1	0
0	5	1	1	0
0	6	0	0	1
0	7	1	0	1
0	8	0	1	1
1	8	1	1	1
2	8	0	0	0

140489

The same table can also be provided for each character length when the receive character length of I field is other than 8 bits.

Table 3.5.23 (b) Bit Patterns by Number of Bits /Character (No Fractions)

Bits/character	D3	D2	D1
5 bits/Character	0	0	1
6 bits/Character	0	1	0
7 bits/Character	0	0	0
8 bits/Character	0	1	1

140489

Bit D4: Parity error

This bit is latched when the parity select bit (D0 of WR4) is set and a parity error is detected in the receive data. Latch can be cleared by the error reset command (WR0 bits D5, D4, D3 = "1", "1", "0").

Bit D5: Receive overrun error

The receive data FIFO holds up to 3 characters. When more characters are received without read out by the MPU, the excess character is set to the receive FIFO. When this character is read by the MPU, this receive overrun error is set. Once set, bit D5 latches that state. When the error reset command (command 6 of WR0 bits D3 through D5) is written, this bit is also reset.

Bit D6: CRC/framing error

In the asynchronous mode, this bit is set when a framing error is detected in the received character. Because this bit is not latched, it is always updated.

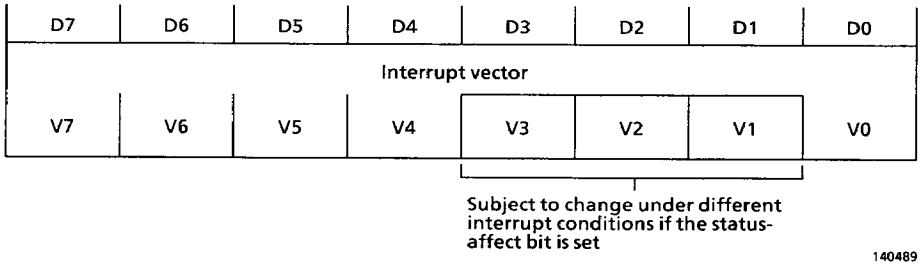
In the synchronous and SDLC modes, this bit indicates the transmitted CRC check result. This bit is reset when the error reset command (command 6 of WR0 bits D3 through D5) is written.

Bit D7: End of frame

This bit is set when the end flag is detected in the receive data and the CRC check and the fraction code are found normal. This bit is reset when the error reset command (command 6 of WR0 bits D3 through D5) is written. This bit is used only in the SDLC mode and is updated when the first character of the next frame is received.

3 RR 2; Read register 2

Table 3.5.24 Configuration of Read Register 2



When the status affect vector bit (D2 of WR1 (Channel B)) is set, bits V3 through V1 are changed depending on the interrupt condition at the time. The vector to be read is determined by the interrupt condition having the highest priority at the time of read. When the status affect vector bit is reset, the contents of this register are the same as those of WR2.

3.5.6 Using SIO

The following describes some system examples using the SIO. Figure 3.5.14 shows an inter-processor communication system. In this example, the MPU on the left side controls the data transfer with the modules on the right side. Both diagrams shown in Figure 3.5.14 (a) and (b) are communication systems. As shown, the SIO is used to interface with external devices in data communication. The greatest advantage of the SIO is the smaller number of data lines than parallel communication.

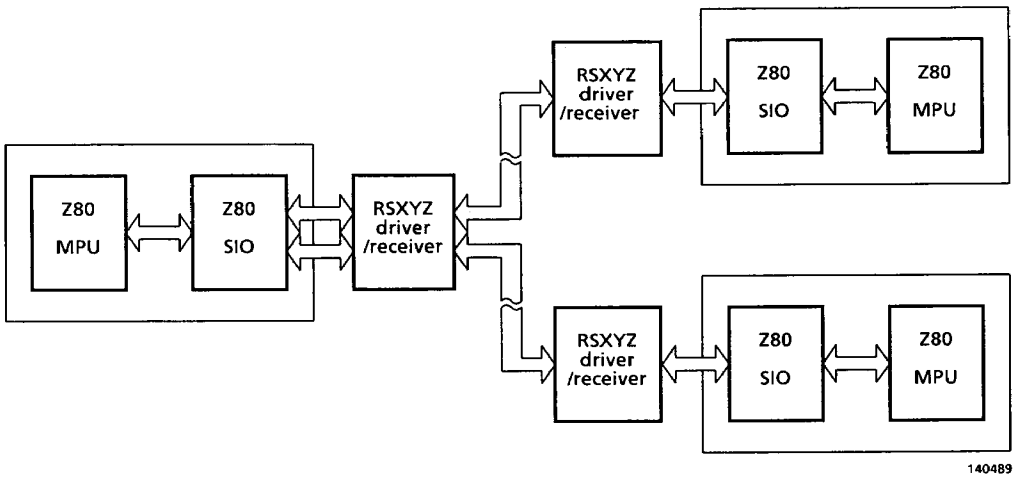


Figure 3.5.14 (a) Example of Data Communication Between Processors

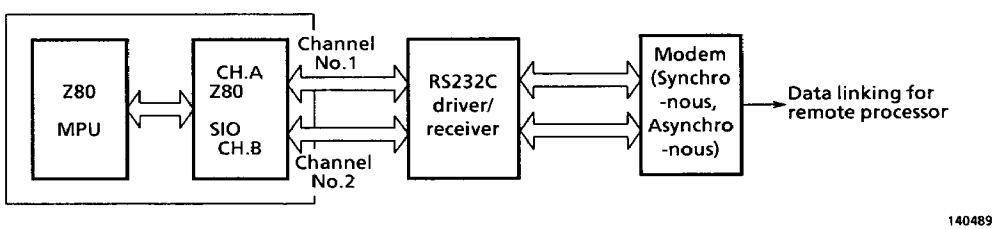


Figure 3.5.14 (b) Example of Data Communication Between Processor

3.6 Standby Capability

When a HALT instruction is executed, the TMPZ84C013A is put in one of the Run, Idle-1, Idle-2, or Stop mode depending on the contents of the halt mode setting register (#F0 : bit 4, bit 3 : HALTMR). (However, the TMPZ84C013A is put in the Run mode immediately after the reset operation by the $\overline{\text{RESET}}$ pin.) The halt mode setting register is set as follows. For the description and timing of each mode, see Subsection 3.3 "CGC Operations. "

The halt mode setting register is assigned to bits 4 and 3 of address F0 in the I/O address area. The halt mode is released by the interrupt (the nonmaskable interrupt by the $\overline{\text{NMI}}$ pin or the maskable interrupt by the $\overline{\text{INT}}$ pin) or by the reset through the $\overline{\text{RESET}}$ pin. A maskable interrupt is accepted when the MPU is in the EI state (in the state after the execution of EI instruction). A nonmaskable interrupt is accepted unconditionally. When an interrupt is accepted, the interrupt processing starts.

When the MPU is in the DI state (after the reset operation and the execution of DI instruction) with maskable interrupt, the TMPZ84C013A returns to the halt mode after executing a HALT instruction (actually a NOP instruction).

3.6.1 Setting Halt Mode

Duplicate control is provided to prevent the stop of the watchdog timer operation which may be caused by the halt mode setting error due to program runaway.

The halt mode is set by the halt mode setting register (HALTMR) and the halt mode control register (#F1 : bits 7 through 0 : HALTMCR). Figure 3.6.1 shows the contents of the halt mode control register (HALTMCR). Figure 3.6.2 shows the contents of the halt mode setting register (HALTMR).

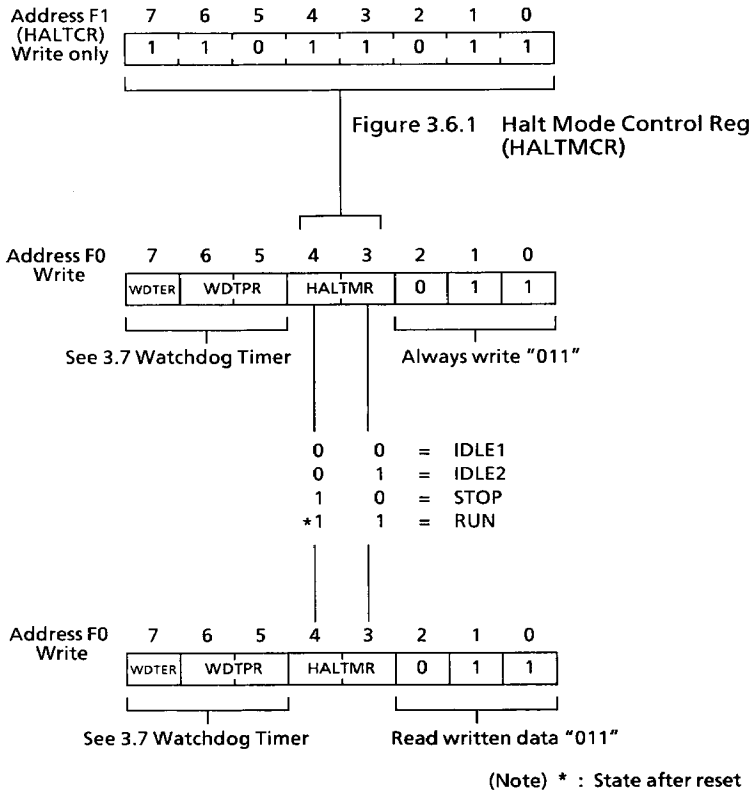


Figure 3.6.2 Halt Mode Setting Register (HALTMR)

Figure 3.6.3 shows the device states in the halt state with the CLKOUT pin connected to the CLKIN pin.

MODE	CGC	MPU	CTC	SIO	Watchdog Timer (WDT)	CLKOUT PIN
IDLE1	○	×	×	×	×	×
IDLE2	○	×	○	×	×	○
STOP	×	×	×	×	×	×
RUN	○	○	○	○	○	○

140489

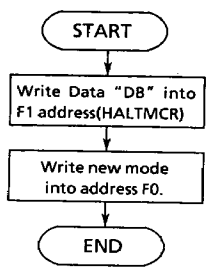
○……Operating (CLKOUT and CLKIN must be connected.)
×……Stop
Note: CLK OUT and CLK IN must be connected.

Figure 3.6.3 Device States in Halt State

For the halt mode in which the clock is supplied from the CLKIN pin (with the CGC oscillator unused), the Run mode must be used.

3.6.2 Halt Mode Setting Procedure

After reset, the halt mode is changed to the Run mode. Figure 3.6.4 shows the procedure to set a new mode.



140489

Figure 3.6.4 Setting Halt Mode

3.7 Watchdog Timer

The watchdog timer (WDT) detects an operation error caused by the program runaway to return to the normal operation.

3.7.1 Block Diagram of Watchdog Timer

Figure 3.7.1 shows the block diagram of the watchdog timer.

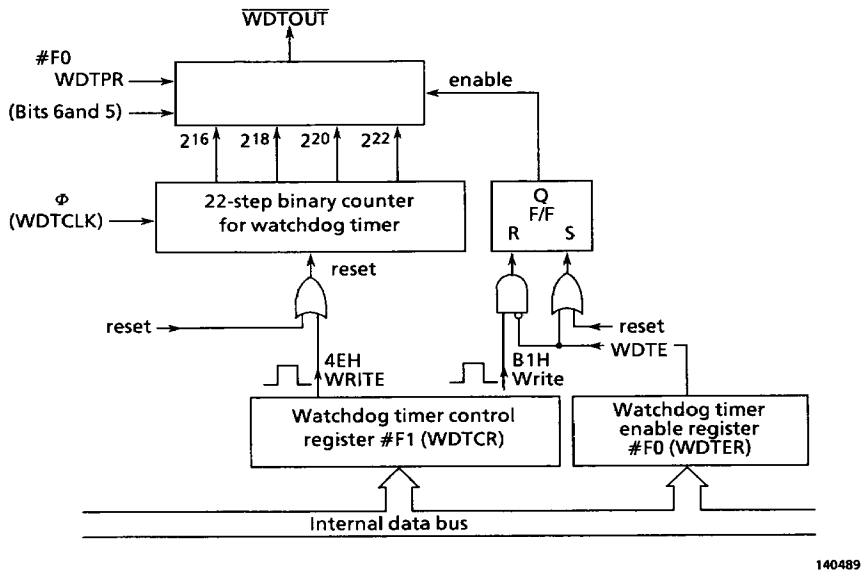


Figure 3.7.1 Block Diagram of Watchdog Timer

3.7.2 Setting Watchdog Timer

(1) Enabling the watchdog timer

The watchdog timer can be set by the watchdog timer enable register (#F0 : bit 7 : WDTER) and the watchdog timer periodic register (#F0 : bit 6, bit 5 : WDTPR).

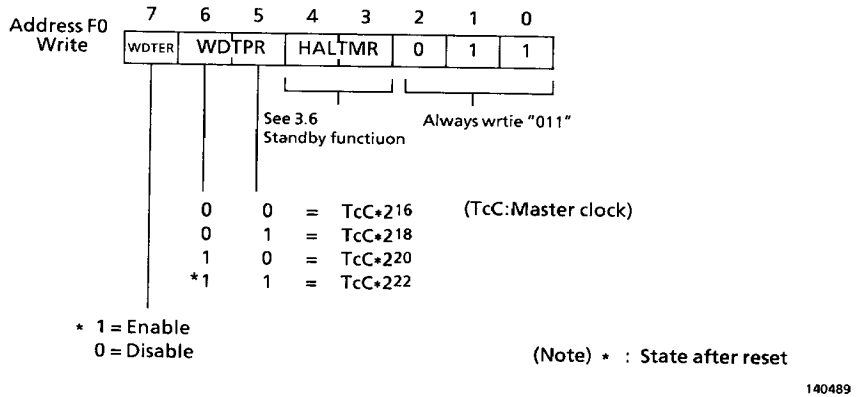


Figure 3.7.2 Enabling Watching Timer

(2) Disabling the watchdog timer

The watchdog timer can be disabled by disabling the watchdog timer enable register (WDTER) then writing data "B1" in the watchdog timer control register (#F1 : bit 7 through bit 0 : WDTCR).

This function has a duplicate structure to prevent the watchdog timer setting error, which may lead to the watchdog timer operation stop, caused by program runaway.

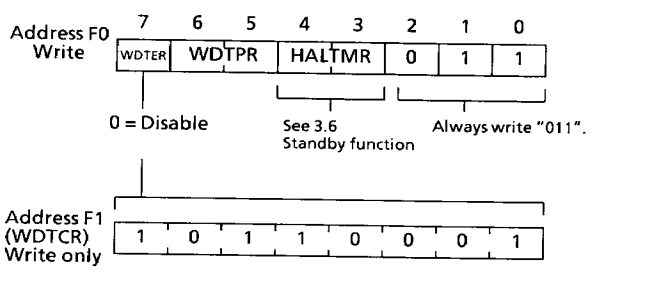


Figure 3.7.3 Disabling Watchdog Timer

(3) Clearing the watchdog timer

The watchdog timer can be cleared by writing data "4E" in the watchdog timer control register (WDTCR).

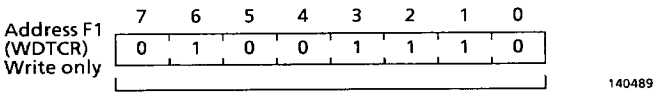


Figure 3.7.4 Cleaning The Watchdog Timer

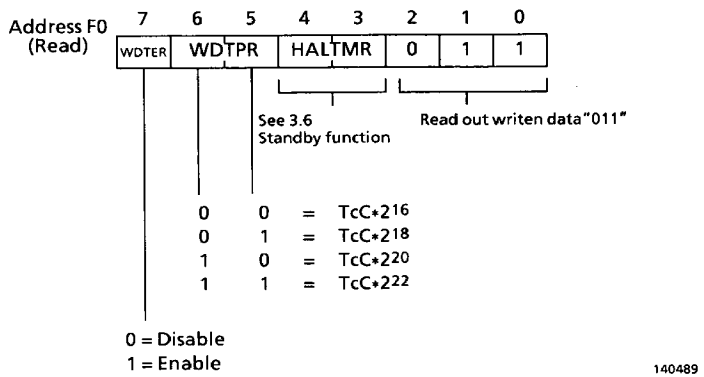


Figure 3.7.5 Reading Watchdog Timer Setting Register

3.7.3 Watchdog Timer Output

When the enabled watchdog timer is used, the "0" level signal is output to the $\overline{\text{WDTOUT}}$ pin after the duration of time specified in the watchdog timer periodic register (WDTPR). The output pulse width is one of the following two types depending on the $\overline{\text{WDTOUT}}$ pin connection :

(1) The $\overline{\text{WDTOUT}}$ connected to the $\overline{\text{RESET}}$ pin: The "0" level pulse of 5TcC (system clock) is output.

(2) The $\overline{\text{WDTOUT}}$ connected to a pin other than $\overline{\text{RESET}}$ pin : The "0" level pulse is kept output until the watchdog timer is cleared by software or reset by the $\overline{\text{RESET}}$ pin.

3.8 Interrupt Priority

The programmable interrupt priority register (#F4 : bits 2 through 0 : INTPR) is provided to determine the interrupt priority for the CTC, SIO, in the TMPZ84C013A.

3.8.1 Setting Interrupt Priority

Figure 3.8.1 shows the register to determine the daisy chain interrupt priority for the CTC, SIO.

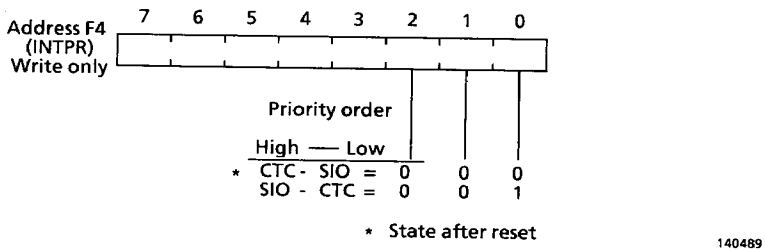


Figure 3.8.1 Interrupt Priority Register (INTPR)

[Example]

When "001" is written in address F4 (INTPR), the daisy chain interrupt priority is given as shown in Figure 3.8.2.

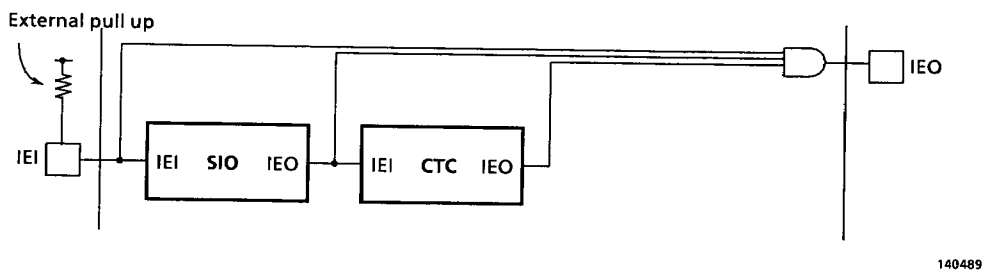


Figure 3.8.2 Daisy Chain Interrupt Priority

4. ELECTRICAL CHARACTERISTICS

4.1 Maximum Ratings

SYMBOL	ITEM		RATING
VCC	Vcc Supply Voltage with respect to Vss		– 0.5V to 7V
VIN	Input Voltage		– 0.5V to VCC + 0.5V
PD	Power Dissipation (10MHz VERSION : TA = 70°C)		250mW
TSOLDER	Soldering Temperature (Soldering Time 10sec)		260°C
TSTG	Storage Temperature		– 55°C to 125°C
TORP	Operating Temperature	10MHz VERSION	– 40°C to 70°C

170890

4.2 DC Electrical Characteristics

10MHz VERSION : TOPR = −40°C to +70°C, VCC = 5V ± 10%, VSS = 0V

(1/2)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT.
VILC	Clock Input Low Voltage (CLKIN)		– 0.3	—	0.6	V
VIHC	Clock Input High Voltage (CLKIN)		VCC-0.6	—	VCC + 0.3	V
VIL	Input Low Voltage (except XTAL1, $\overline{\text{RESET}}$)		– 0.5	—	0.8	V
VIH	Input High Voltage (except XTAL1, $\overline{\text{RESET}}$)		2.2	—	VCC	V
VILR	Input Low Voltage ($\overline{\text{RESET}}$)		– 0.5	—	0.45	V
VIHR	Input High Voltage ($\overline{\text{RESET}}$)		VCC-0.6	—	VCC	V
VOLC	Output Low Voltage (CLKOUT)	IOL = 2.0mA	—	—	0.6	V
VOHC	Output High Voltage (CLKOUT)	IOH = – 2.0mA	VCC-0.6	—	—	V
VOL	Output Low Voltage (except CLKOUT)	IOL = 2.0mA	—	—	0.4	V
VOH1	Output High Voltage 1 (except CLKOUT)	IOH = – 1.6mA	2.4	—	—	V
VOH2	Output High Voltage 2 (except CLKOUT)	IOH = – 250uA	VCC-0.8	—	—	V
ILI	Input Leakage Current	Vss ≦ VIN ≦ Vcc	—	—	± 10	μA
ILO	3-state Output Leakage Current in Float	Vss ≦ Vout ≦ VCC	—	—	± 10	μA

170890

(2/2)

SYMBOL	PARAMETER	TEST CONDITION		MIN.	TYP.	MAX.	UNIT.
ICCI	Power Supply Current	VCC = 5V, fCLK = (1) VIHC = VIHR = VIH = VCC-0.2V, VILC = VILR = VIL = 0.2V	AT-10	—	40	43	mA
ICC2	Stand-by Supply Current (See Note(2))	VCC = 5V, fCLK = Stopped, VIHC = VIH = VIHR = VCC-0.2V, VILC = VIL = VILR = 0.2V	AT-10	—	0.5	50	μA
ICC3	Power Supply Current (IDLE1 Mode)	VCC = 5V, fCLK = (1), VIHC = VIH = VIHR = VCC-0.2V, VILC = VIL = VILR = 0.2V	AT-10	—	2.5	5	mA
ICC4	Power Supply Current (IDLE2 Mode)	VCC = 5V, fCLK = (1), VIHC = VIH = VIHR = VCC-0.2V, VILC = VIL = VILR = 0.2V	AT-10	—	19	25	mA

170890

Note :

(1) fCLK = 1/TcC (MIN)

(2) ICC2 Stand-by Supply Current is guaranteed only when the supplied clock is stopped at a low level during T4 state of the following machin Cycle (M1) next to OP code fetch Cycle of HALT instruction.
Except $\overline{SYNCA} = 0$ or $\overline{SYNCB} = 0$ state

4.3 AC Electrical Characteristics (1) (in Active State)

10MHz VERSION : $TA = -40°C \sim 70°C$, $VCC = 5V \pm 10\%$, $VSS = 0V$

4.3.1 AC Characteristics of CPU (in Active State)

(1/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
1	TcC	Clock Cycle Time	100	—	DC	ns
2	TwCh	Clock Pulse Width (High)	38	—	DC	ns
3	TwCl	Clock Pulse Width (Low)	38	—	DC	ns
4	TfC	Clock Fall Time	—	—	12	ns
5	TrC	Clock Rise Time	—	—	12	ns
6	TdCr (A)	Clock ↑ to Address Valid Delay	—	—	75	ns
7	TdA (MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay	15	—	—	ns
8	TdCf (MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay	—	—	55	ns
9	TdCr (MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay	—	—	55	ns
10	TwMREQh	$\overline{MREQ}$ pulse Width (High)	32	—	—	ns
11	TwMREQ1	$\overline{MREQ}$ pulse Width (Low)	75	—	—	ns
12	TdCf (MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay	—	—	55	ns
13	TdCf (RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay	—	—	65	ns
14	TdCr (RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay	—	—	55	ns
15	TsD (Cr)	Data Setup Time to Clock ↑	25	—	—	ns
16	ThD (RDr)	Data Hold Time to $\overline{RD}$ ↑	0	—	—	ns
17	TsWAIT (Cf)	$\overline{WAIT}$ Setup Time to Clock ↓	25	—	—	ns
18	ThWAIT (Cf)	$\overline{WAIT}$ Hold Time after Clock ↓	10	—	—	ns
19	TdCr (M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay	—	—	65	ns
20	TdCr (M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay	—	—	65	ns
21	TdCr (RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Delay	—	—	80	ns
22	TdCr (RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay	—	—	80	ns
23	TdCf (RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay	—	—	55	ns
24	TdCr (RDf)	Clock ↑ to $\overline{RD}$ ↓ Delay	—	—	55	ns
25	TsD (Cf)	Data Setup to Clock ↓ during M2, M3, M4 or M5 Cycles	25	—	—	ns
26	TdA (IORQf)	Address Stable prior to $\overline{IORQ}$ ↓	70	—	—	ns
27	TdCr (IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay	—	—	50	ns

170890

(2/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
28	TdCf (IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay	—	—	55	ns
29	TdD (WRf)	Data Stable Prior to $\overline{WR}$ ↓	40	—	—	ns
30	TdCf (WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay	—	—	55	ns
31	TwWR	$\overline{WR}$ Pulse Width	75	—	—	ns
32	TdCf (WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay	—	—	55	ns
33	TdD (WRf)	Data Stable Prior to $\overline{WR}$ ↓	−8	—	—	ns
34	TdCr (WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay	—	—	50	ns
35	TdWRr (D)	Data Stable from $\overline{WR}$ ↑	12	—	—	ns
36	TdCf (HALT)	Clock ↓ to $\overline{HALT}$ ↑ or ↓	—	—	90	ns
37	TwNMI	$\overline{NMI}$ Pulse Width	65	—	—	ns
38	TsBUSREQ (Cr)	$\overline{BUSREQ}$ Setup Time to Clock ↑	30	—	—	ns
39	ThBUSREQ (Cr)	$\overline{BUSREQ}$ Hold Time after Clock ↑	10	—	—	ns
40	TdCr (BUSACKf)	Clock ↑ to $\overline{BUSACK}$ ↓ Delay	—	—	75	ns
41	TdCf (BUSACKr)	Clockt ↓ to $\overline{BUSACK}$ ↑ Delay	—	—	75	ns
42	TdCr (Dz)	Clock ↑ to Data Float Delay	—	—	65	ns
43	TdCr (CTz)	Clock ↑ to Control Outputs Float Delay ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$)	—	—	60	ns
44	TdCr (AZ)	Clock ↑ to Address Float Delay	—	—	65	ns
45	TdCr (A)	$\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time	32	—	—	ns
46	TsRESET (Cr)	$\overline{RESET}$ to Clock ↑ Setup Time	40	—	—	ns
47	ThRESET (Cr)	$\overline{RESET}$ to Clock ↑ Hold Time	10	—	—	ns
48	TsINTf (Cr)	$\overline{INT}$ to Clock ↑ Setup Time	50	—	—	ns
49	TsINTr (Cr)	$\overline{INT}$ to Clock ↑ Hold Time	10	—	—	ns
50	TdM1f (IORQf)	$\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay	222	—	—	ns
51	TdCf (IORQf)	Clock ↓ to $\overline{IORQ}$ ↓ Delay	—	—	55	ns
52	TdCr (IORQr)	Clock ↑ to $\overline{IORQ}$ ↑ Delay	—	—	55	ns
53	TdCf (D)	Clock ↓ to Data Valid Delay	—	—	110	ns

110690

4.3.2 AC Characteristics of CGC (in Active State)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
54	TcC CLK	Output clock cycle	—	100	—	ns
55	TwCh CLK	Output clock Width(High)	38	—	—	ns
56	TwCl CLK	Output clock Width (Low)	38	—	—	ns
57	TfC CLK	Output clock fall time	—	—	12	ns
58	TrC CLK	Output clock rise time	—	—	12	ns
59	TRST (INT) S	CLK out restart time by $\overline{INT}$ (STOP Mode)	—	2^{14} + 2.5TcC	—	ns
60	TRST (NMI) S	CLK out restart time by NMI (STOP Mode)	—	2^{14} + 2.5TcC	—	ns
61	TRST (INT) I	CLK out restart time by $\overline{INT}$ (IDLE 1/2 Mode)	—	2.5 *TcC	—	ns
62	TRST (NMI) I	CLK out restart time by $\overline{NMI}$ (IDLE 1/2 Mode)	—	2.5 *TcC	—	ns
63	TRST (RESET) I	CLK out restart time by $\overline{RESET}$ (IDLE 1/2 Mode)	—	TcC	—	ns

110690

4.3.3 AC Characteristics of CTC (in Active State)

NO.	SYMBOL	PARAMETER	TMPZ84C013AF-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
64	TdM1 (IEO)	Delay from $\overline{M1}$ fall to IEO fall (in case of generating only interrupt immediately before M1 cycle)	—	—	130	ns
65	TdIEI (IEOf)	Delay from IEI fall to IEO fall	—	—	50	ns
66	TdIEI (IEOr)	Delay from IEI rise to IEO rise (after ED decode)	—	—	120	ns
67	IsCLK ($\overline{INT}$)	CLK/TRG setup to TL ↑ for detection of interrupt tsCTR (c) Satisfied	TcC + 100 + T68 + T48	—	—	ns
		tsCTR (c) not Satisfied	2TcC + 100 + T68 + T48	—	—	
68	TcCTR	CLK/TRG Frequency (counter mode)	2TcC	—	—	ns
69	TrCTR	CLK/TRG rising time	—	—	30	ns
70	TfCTR	CLK/TRG falling time	—	—	30	ns
71	TwCTR1	CLK/TRG Pulse Width (Low)	90	—	—	ns
72	TwCTRh	CLK/TRG Pulse Width (High)	90	—	—	ns
73	TsCTR (Cs)	CLK/TRG ↑ to Clock ↑ Setup Time for Immediate Count (counter mode)	110	—	—	ns
74	TsCTR (CT)	CLK/TRG ↑ to Clock ↑ Setup Time for enabling of Prescaler on following clock ↑ (timer mode)	110	—	—	ns
75	TdC (ZC/TOr)	Clock ↑ to ZC/TO ↑ Delay	—	—	110	ns
76	TdC (ZC/TOf)	Clock ↓ to ZC/TO ↓ Delay	—	—	110	ns

110690

4.3.4 AC Characteristics of SIO (in Active State)

(1/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
77	TsM1 (C)	$\overline{M1}$ ↑ to clock ↑ Setup time	50	—	—	ns
78	TsIEI (IO)	IEI ↓ to $\overline{IORQ}$ ↓ Setup time (INTACK cycle)	100	—	—	ns
79	TdM1 (IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt before $\overline{M1}$)	—	—	120	ns
80	TdIEI (IEOr)	$\overline{IEI}$ ↓ to IEO ↑ Delay (after ED decode)	—	—	120	ns
81	TdIEI (IEOf)	IEI ↓ to IEO ↓ Delay	—	—	50	ns
82	TdIO (W/RWf)	$\overline{IORQ}$ ↓ or $\overline{CE}$ ↓ to $\overline{W/RDY}$ ↓ Delay (Wait mode)	—	—	130	ns
88	TdC (W/RRf)	Clock ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	—	—	80	ns
89	TdC (W/RWZ)	Clock ↓ to $\overline{W/RDY}$ float delay (Wait mode)	—	—	90	ns
90	TwPh	Pulse Width (High)	200	—	—	ns
91	TwPl	Pulse Width (Low)	200	—	—	ns
92	TcTxC	$\overline{TxC}$ cycle time	250	—	∞	ns
93	TwTxCl	$\overline{TxC}$ Width (Low)	80	—	∞	ns
94	TwTxCh	$\overline{TxC}$ Width (High)	80	—	∞	ns
95	TdTxC (TxD)	$\overline{TxC}$ ↓ toTxD Delay (x1 mode)	—	—	180	ns
96	TdTxC (W/RRf)	$\overline{TxC}$ ↓ to$\overline{W/RDY}$ ↓ Delay (Ready mode)	5	—	9	CLK Periods
97	TcRxC	$\overline{RxC}$ cycle time	250	—	∞	ns
98	TwRxCl	$\overline{RxC}$ Width (Low)	80	—	∞	ns
99	TwRxCh	$\overline{RxC}$ Width (High)	80	—	∞	ns
100	TsRxD (Rxc)	RxD to $\overline{RxC}$ ↑ Setup time (x1 mode)	10	—	—	ns
101	ThRxD (Rxc)	$\overline{RxC}$ ↑ to RxD Hold time (x1 mode)	80	—	—	ns
102	TdRxC (W/RRf)	$\overline{RxC}$ ↑ to $\overline{W/RDY}$ ↓ Delay (Ready mode)	10	—	13	CLK Periods
103	TdRxC (SYNC)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ ↓ Delay (Output mode)	4	—	7	CLK Periods
104	TsSYNC (RxC)	$\overline{SYNC}$ ↑ to $\overline{RxC}$ ↓ Setup (External SYNC modes)	− 100	—	—	ns

170890

(2/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
105	IsTxC	$\overline{TxC}$ Setup to TL ↑ for detection of interrupt	5*T1 + T48	—	9*T1 + T48	ns
106	IsRxC	$\overline{RxC}$ Setup to TL ↑ for detection of interrupt	10*T1 + T48	—	13*T1 + T48	ns

170890

4.3.5 AC Chracteristics of WDT (in Active State)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
107	TdC (WDTf)	Clock ↑ to $\overline{WDTOUT}$ ↓ Delay	—	—	120	ns
108	Tdc (WDTr)	Clock ↑ to $\overline{WDTOUT}$ ↑ Delay	—	—	125	ns
109	TcWDT	$\overline{WDTOUT}$ OUTPUT Period				
		WDT Mode0	—	$T1*2^{16}$	—	ns
		WDT Mode1	—	$T1*2^{18}$	—	
		WDT Mode2	—	$T1*2^{20}$	—	
		WDT Mode3	—	$T1*2^{22}$	—	

170890

Note1 : Timing Measurements are made at the following voltage.
Input VIH=2.4V, VIL=0.4V, VIHC=VCC−0.6V, VILC=0.6V
Output VOH=2.2V, VOL=0.8V, (Except CLK OUT)
CL=100pF

4.4 AC Timing Charts (1) (in Active State)

4.4.1 AC Timing Charts of CPU (in Active State)

Figure 4.4.1 through 4.4.8 show the basic timing charts. The circled numbers in these charts correspond to the numbers in the number column of the AC Electrical Characteristics Tables.

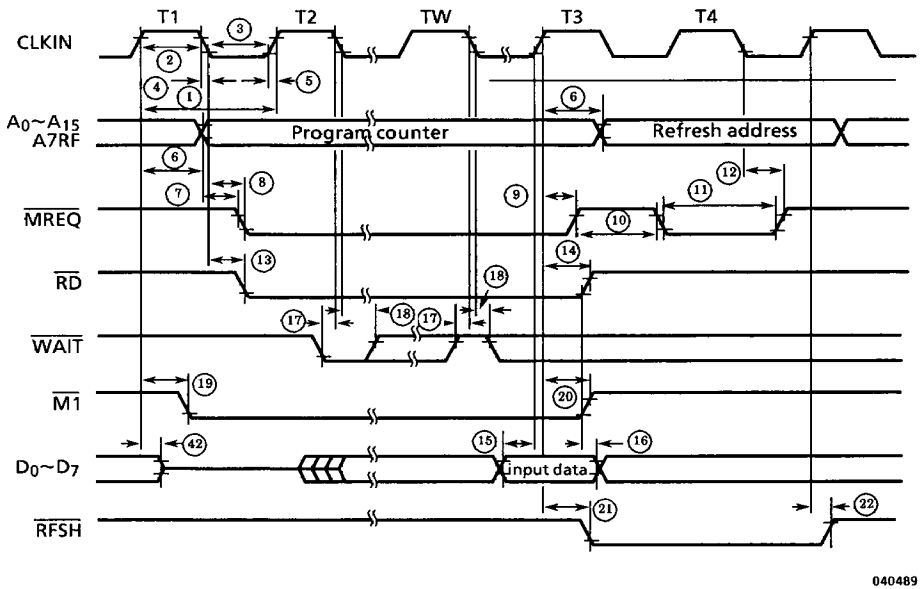


Figure 4.4.1 Opcode Fetch Cycle

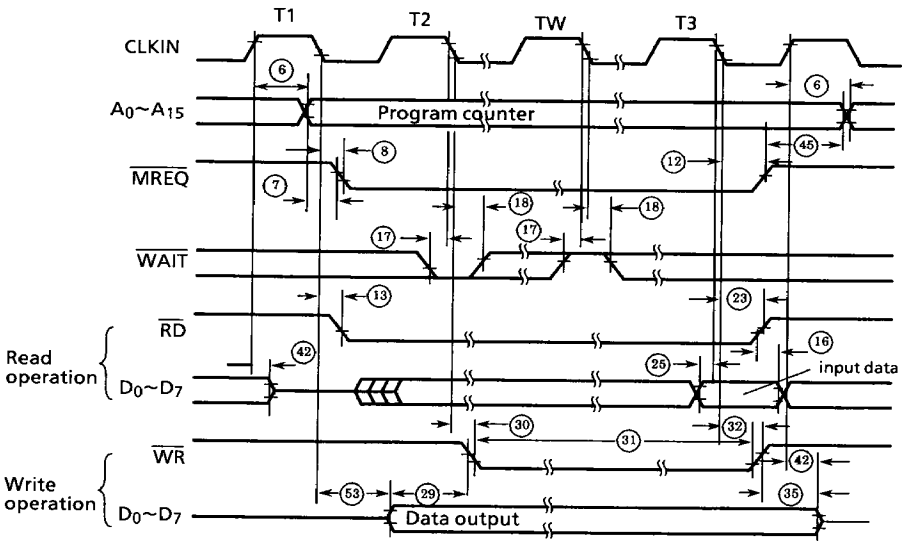
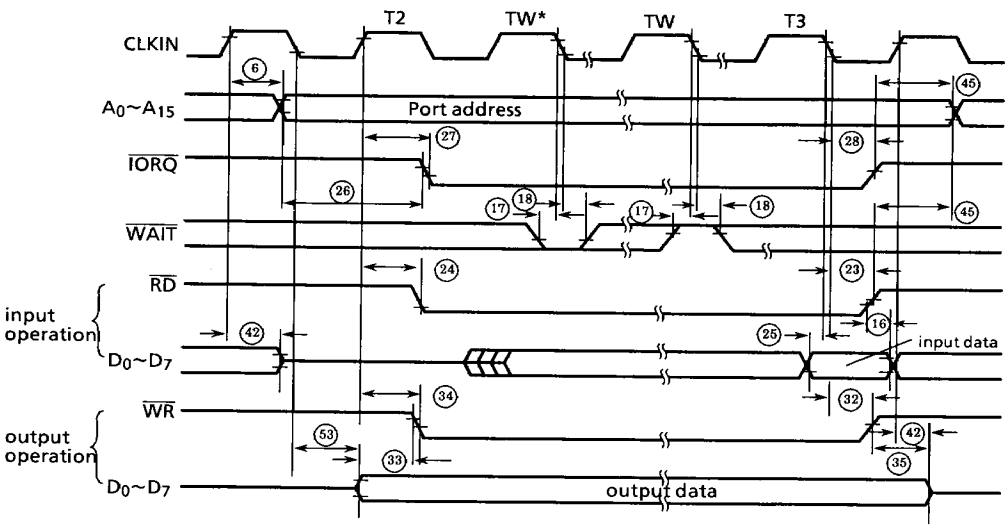
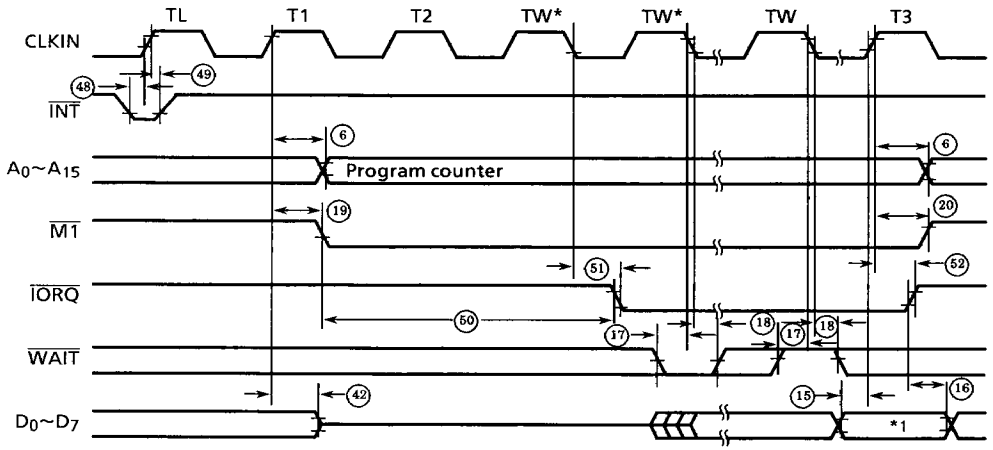


Figure 4.4.2 Memory Read/Write Cycle



Note : 1-wait state (TW*) is inserted automatically by MPU.

Figure 4.4.3 I/O Cycle

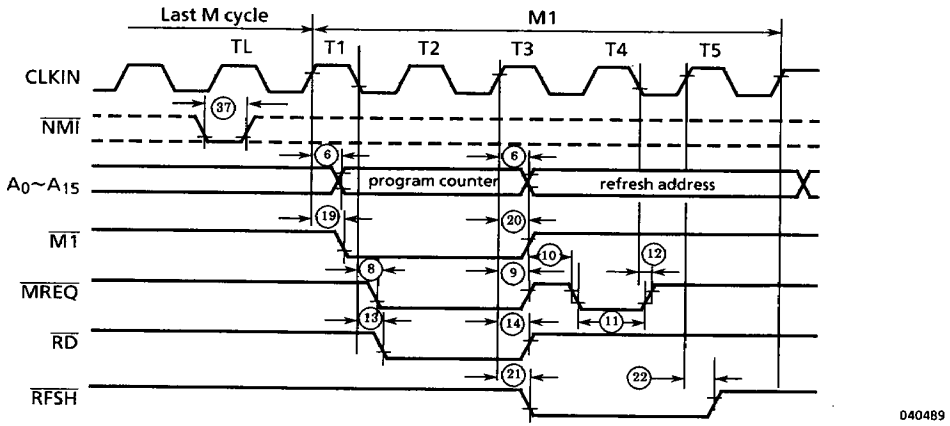


Notes:
1. TL : Last state of instruction
2. 2-wait state (TW*) is inserted automatically by MPU

*1 VALID DATA

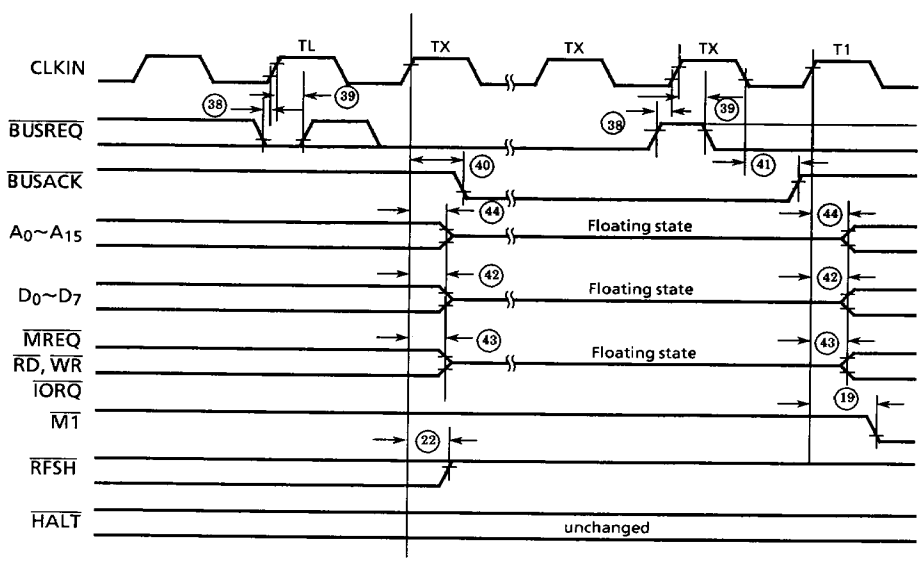
040489

Figure 4.4.4 Interrupt Request/Acknowledge Cycle



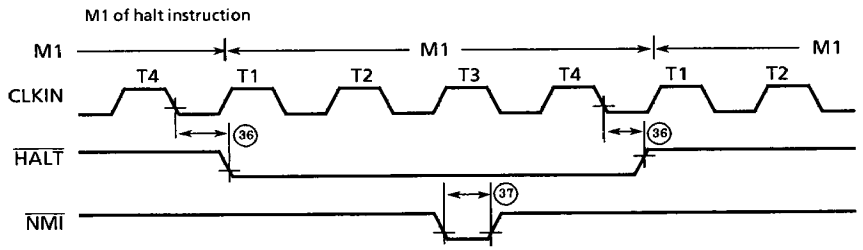
Note: $\overline{NMI}$ is asynchronous, but its falling edge signal must occur synchronously with the rising edge of previous TL state for correct response to the subsequent machine cycles.

Figure 4.4.5 Non-maskable Interrupt Request Cycle



Notes
1. TL: Last state of a given machine cycle
2. TX: Clock used by peripheral LSI that made request.

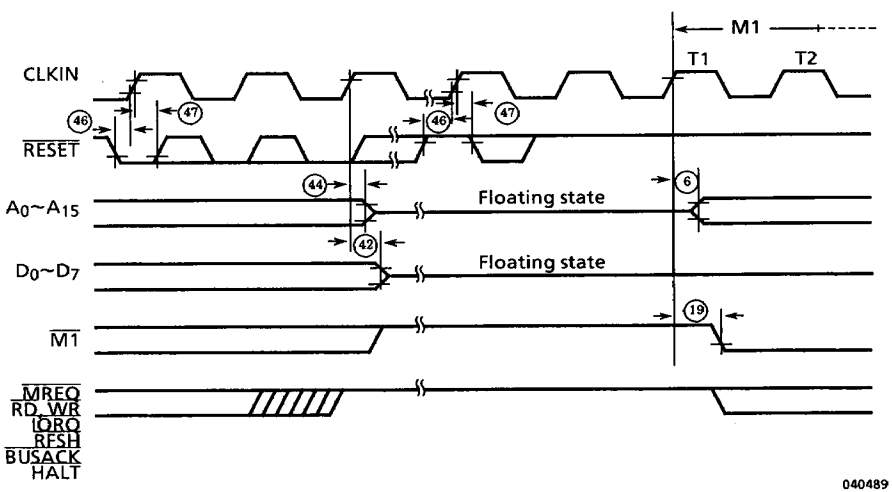
Figure 4.4.6 Bus Request/Acknowledge Cycle



Note : $\overline{INT}$ signal is also used for releasing HALT state.

040489

Figure 4.4.7 Halt Acknowledge Cycle

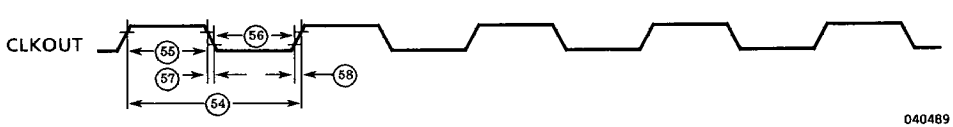


040489

Figure 4.4.8 Reset Cycle

4.4.2 AC Timing Charts of CGC (in Active State)

The following Figures show the timings in each operation mode with the CLKOUT pin connected to the CLKIN pin.



040489

Figure 4.4.9 CLKOUT Waveform

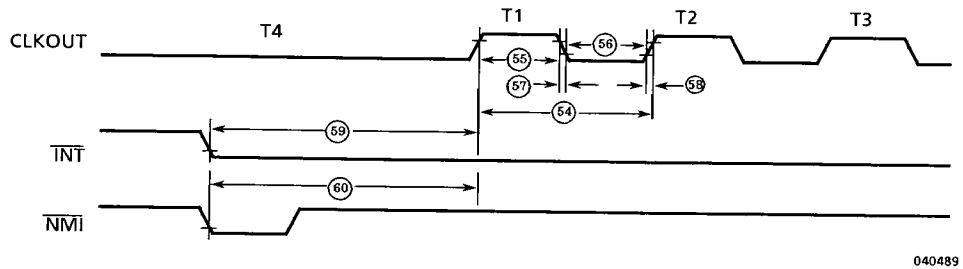


Figure 4.4.10 Clock Restart Timing (STOP mode)

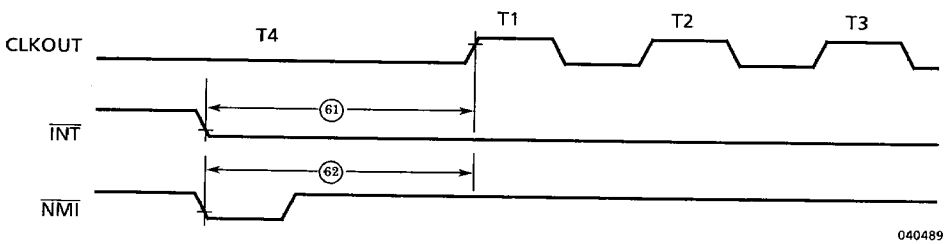


Figure 4.4.11 Clock Restart Timing (IDLE1 mode)

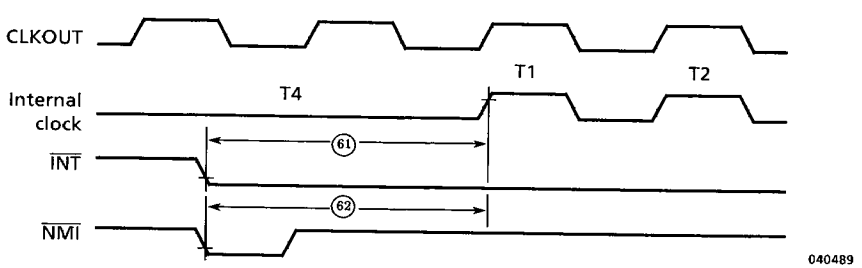


Figure 4.4.12 Clock Restart Timing (IDLE2 mode)

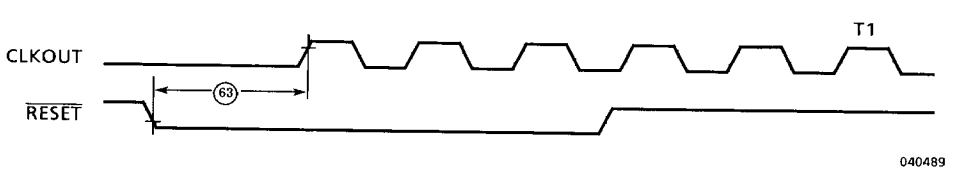


Figure 4.4.13 Timing of Clock Start by $\overline{\text{RESET}}$ (IDLE1 and IDLE2 modes)

4.4.3 AC Timing Charts of CTC (in Active State)

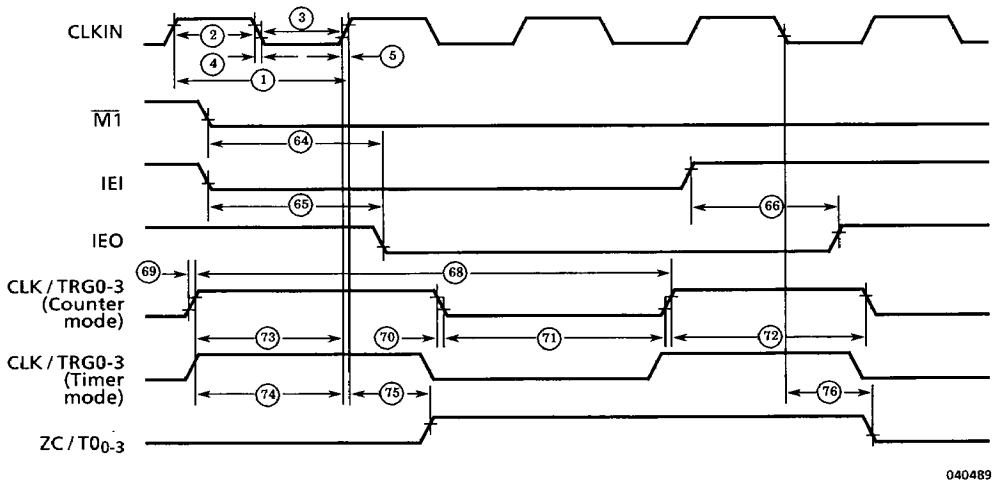


Figure 4.4.14 CTC Timing Diagram

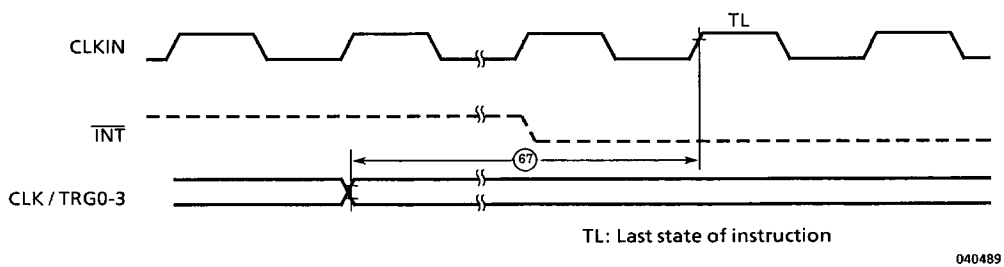


Figure 4.4.15 CTC Interrupt Occurrence Timing

4.4.4 AC Timing Charts of SIO (in Active State)

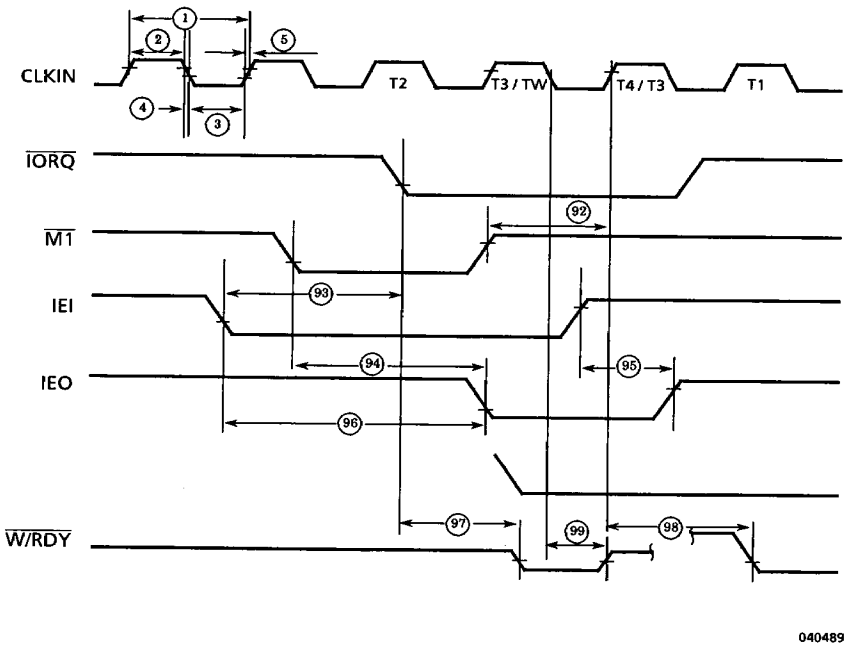


Figure 4.4.16 (a) SIO Timing Diagram

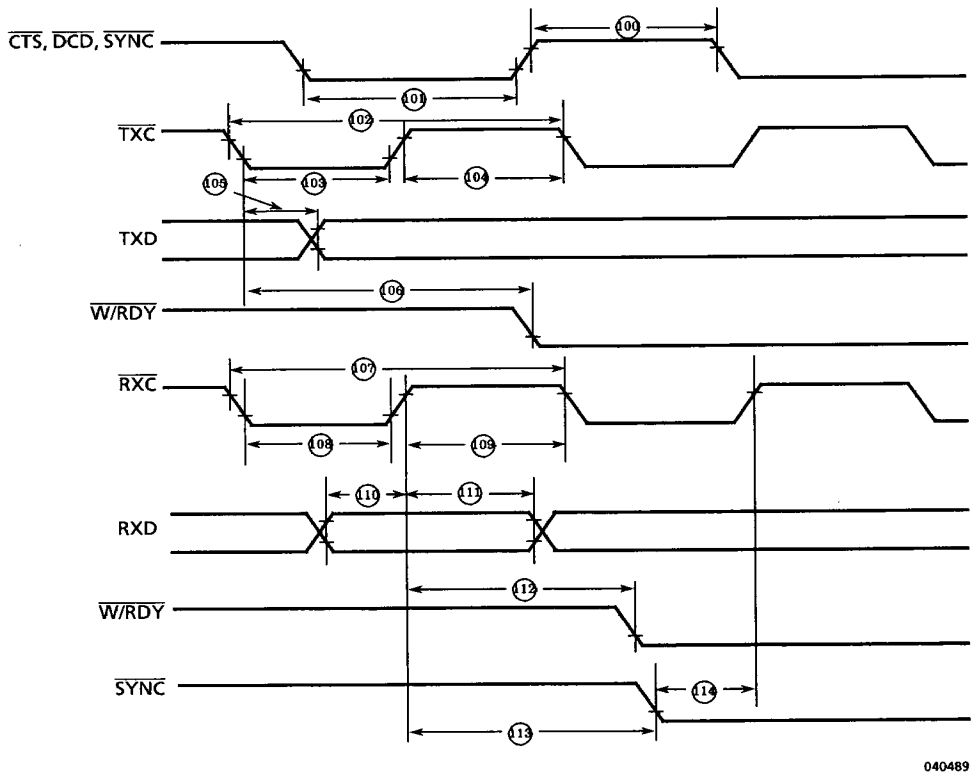


Figure 4.4.16 (b) SIO Timing Diagram

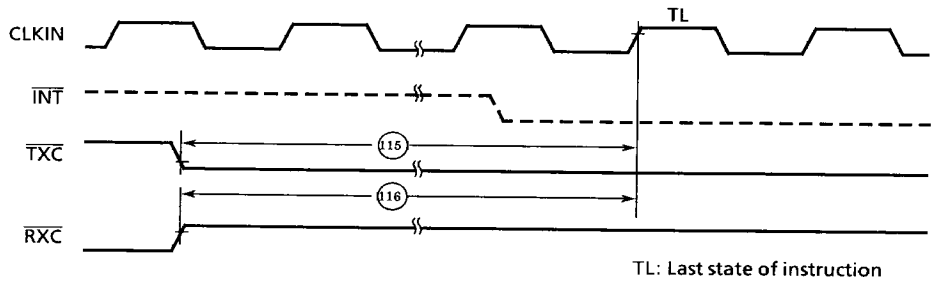


Figure 4.4.17 SIO Interrupt Occurrence Timing

4.4.5 AC Timing Charts of WDT (in Active State)

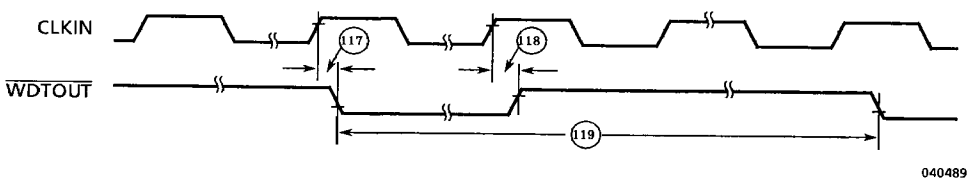


Figure 4.4.18 WDT Timing Diagram

4.5 AC Electrical Characteristics (2) (in Inactive State)

10MHz VERSION : $TA = -40°C \sim +70°C$, $VCC = 5V + 10\%$, $VSS = 0V$

4.5.1 AC Characteristics of CGC (in Inactive State)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
1	TcCCLK	Output Clock Cycle	—	100	—	ns
2	TwChCLK	Output Clock Width (High)	—	40	—	ns
3	TwCICLK	Output Clock Width (Low)	—	40	—	ns
4	TfCCLK	Output Clock fall Time	—	10	—	ns
5	TrCCLK	Output Clock rise Time	—	10	—	ns
6	TRST (INT) S	Clock (CLKOUT) restart Time by $\overline{INT}$ (STOP mode)	—	2^{14} + 2.5TcC	—	ns
7	TRST (NMI) S	Clock (CLKOUT) restart Time by $\overline{NMI}$ (STOP mode)	—	2^{14} + 2.5TcC	—	ns
8	TRST (INT) S	Clock (CLKOUT) restart Time by $\overline{INT}$ (IDLE1/2 mode)	—	2.5 *TcC	—	ns
9	TRST (NMI) I	Clock (CLKOUT) restart Time by $\overline{NMI}$ (IDLE1/2 mode)	—	2.5 *TcC	—	ns
10	TRST (RESET) I	Clock (CLKOUT) restart Time by $\overline{RESET}$ (IDLE1/2 mode)	—	1TcC	—	ns
11	TsHALT (M1r)	$\overline{HALT}$ Setup Time	10	—	—	ns

170890

4.5.2 AC Characteristics of CTC (in Inactive State)

(1/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
12	TcC	Clock Cycle Time	100	—	—	ns
13	Twch	Clock width (High)	38	—	—	ns
14	Twcl	Clock width (Low)	38	—	—	ns
15	TfC	Clock falling time	—	—	12	ns
16	TrC	Clock rising time	—	—	12	ns
17	Th	Hold Time	10	—	—	ns
18	TcCS (C)	CS (A1,A0) Setup time to clock ↑	100	—	—	ns
19	TSCE (C)	CE (A7~A2) Setup time to clock ↑	80	—	—	ns
20	TsIO (C)	$\overline{IORQ}$ ↓ Setup time to clock ↑	65	—	—	ns

170890

(2/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
21	TsRD (C)	$\overline{RD}$ ↓ Setup time to clock ↑	55	—	—	ns
22	TdC (DO)	Clock ↑ to Data Valid Delay	—	—	110	ns
23	Tdc (DOz)	$\overline{IORQ}$, $\overline{RD}$ ↑ to Data Float Delay	—	—	85	ns
24	TsDI (C)	Data Input Setup time to clock ↑	40	—	—	ns
25	TsM1(C)	$\overline{M1}$ Setup time to clock ↑	55	—	—	ns
26	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (in case of generating only interrupt immediately before M1 cycle)	—	—	110	ns
27	TdIO (DOI)	$\overline{IORQ}$ ↓ to Data out Delay (INTA Cycle)	—	—	85	ns
28	TdIEI (IEOf)	IEI ↓ to IEO ↓ Delay	—	—	60	ns
29	TdIEI (IEOr)	IEI ↑ to IEO ↑ Delay (after ED decode)	—	—	120	ns
30	TdC (INT)	Clock ↑ to $\overline{INT}$ ↓ Delay	—	—	TcC + 100	ns
31	TdA (IORQf)	CLK/TRG ↑ to $\overline{INT}$ ↓ Delay TsCTR (c) Satisfied TsCTR (c) not Satisfied	— —	TcC + 100 + 70 + T37 2TcC + 100 + 70 + T37	— —	ns
32	TcCTR	CLK/TRG Frequency	—	2TcC	—	ns
33	TrCTR	CLK/TRG rising time	—	—	30	ns
34	TfCTR	CLK/TRG falling time	—	—	30	ns
35	TwCTRl	CLK/TRG pulse width (Low)	90	—	—	ns
36	TwCTR	CLK/TRG pulse width (High)	90	—	—	ns
37	TsCTR (CS)	CLK/TRG ↑ to clock ↑ Setup Time for Immediate Count (counter mode)	110	—	—	ns
38	TsCTR (CT)	CLK/TRG ↑ to clock ↑ Setup Time for enabling of Prescaler on following clock ↑ (timer mode)	110	—	—	ns
39	TdC (ZC/TOr)	Clock ↑ to ZC/TO ↑ Delay	—	—	110	ns
40	TdC (ZC/TOf)	Clock ↓ to ZC/TO ↓ Delay	—	—	110	ns

170890

4.5.3 AC Characteristics of SIO (in Inactive State)

(1/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
41	TsCS (C)	$\overline{CE}$ (A_7toA_2), C/$\overline{D}$ (A_0), B/$\overline{A}$ (A_1) Setup Time to Clock ↑	60	—	—	ns
42	TsRD (C)	$\overline{IORQ}$, $\overline{RD}$ Setup Time to clock ↑	55	—	—	ns
43	TdC (DO)	Clock ↑ to Data output Delay	—	—	100	ns
44	TsDI (C)	Data input setup time to clock ↑ (write cycle or $\overline{M1}$ cycle)	30	—	—	ns
45	TdRD (DOz)	$\overline{RD}$ ↑ to Data Out Float Delay	—	—	70	ns
46	TdIO (DOI)	$\overline{IORQ}$ ↓ to Data output Delay (INTACK cycle)	—	—	85	ns
47	TsM1 (C)	$\overline{M1}$ Setup time to clock ↑	50	—	—	ns
48	TsIEI (IO)	IEI Setup time to $\overline{IORQ}$ ↓ (INTACK cycle)	80	—	—	ns
49	TdM1 (IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (interrupt before $\overline{M1}$)	—	—	120	ns
50	TdIEI (IEOr)	IEI ↑ to IEO ↑ Delay (After ED Decode)	—	—	120	ns
51	TdIEI (IEOf)	IEI ↓ to IEO ↓ Delay	—	—	50	ns
52	TdC (INT)	Clock ↑ to $\overline{INT}$ ↓ Delay	—	—	100	ns
53	TdIO (W/RWf)	$\overline{IORQ}$, $\overline{CE}$ (A_7 to A_2) ↓ to $\overline{W/RDY}$ ↓ Delay (Wait Mode)	—	—	130	ns
54	TdC (W/RRf)	Clock ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	—	—	80	ns
55	TdC (W/RWz)	Clock ↓ to $\overline{W/RDY}$ Float Delay (Wait Mode)	—	—	90	ns
56	Th,Th (CS)	Any unspecified hold when setup is specified	0	—	—	ns
57	TwPh	Pulse width (High)	200	—	—	ns
58	TwPl	Pulse width (Low)	200	—	—	ns
59	TcTxC	$\overline{TxC}$ Cycle time	250	—	∞	ns
60	TwTxCl	$\overline{TxC}$ width (Low)	80	—	∞	ns
61	TwTxCh	$\overline{TxC}$ width (High)	80	—	∞	ns
62	TdTxC (TxD)	$\overline{TxC}$ ↓ to TxD Delay (x1 Mode)	—	—	180	ns
63	TdTxC (W/RRf)	$\overline{TxC}$ ↓ to $\overline{W/RDY}$ ↓ Delay (Ready mode)	5	—	9	CLK Periods
64	TdTxC (INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ ↓ Delay	5	—	9	CLK Periods
65	TcRxC	$\overline{RxC}$ Cycle time	250	—	∞	ns

170890

(2/2)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
66	TwRxCl	$\overline{RxC}$ width (Low)	80	—	∞	ns
67	TwRxCh	$\overline{RxC}$ width (High)	80	—	∞	ns
68	TsRxD (RxC)	RxD to $\overline{RxC}$ ↑ Setup time (x1 mode)	0	—	—	ns
69	ThRxD (RxC)	$\overline{RxC}$ ↑ to RxD hold time (x1 mode)	80	—	—	ns
70	TdRxC(W/RRf)	$\overline{RxC}$ ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	10	—	13	CLK Periods
71	TdRxC (INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ ↓ Delay	10	—	13	CLK Periods
72	TdRxC (SYNC)	$\overline{RxC}$ ↓ to $\overline{SYNC}$ ↑ Delay (Output modes)	4	—	7	CLK Periods
73	TsSYNC (RxC)	$\overline{SYNC}$ ↓ to $\overline{RxC}$ ↑ Setup Time (External syncmodes)	−100	—	—	ns
74	TsAdd (Cr)	Address Setup time to clock ↑	150	—	—	ns
75	TsIO (Cr)	$\overline{IORQ}$ ↓ Setup time to clock ↑	70	—	—	ns
76	TdRD (Cr)	$\overline{RD}$ ↓ Setup time to clock ↑	70	—	—	ns
77	TdCr (Do)	Data out Delay to clock ↑	—	—	130	ns
78	TdIORDr (DoZ)	Data Float Delay to $\overline{IORQ}$ ↑ $\overline{RD}$ ↑	—	—	90	ns
79	TsWR (Cr)	$\overline{WR}$ ↓ Setup time to clock ↑	70	—	—	ns
80	TsDI (Cr)	Data Input Setup time to clock ↑	0	—	—	ns
81	TdIOWRf (D)	Data Hold time to $\overline{IORQ}$, $\overline{WR}$ ↑	20	—	—	ns

170890

4.5.4 AC Characteristics of WDT (in Inactive State)

NO.	SYMBOL	PARAMETER	TMPZ84C013AT-10 (10MHz)			UNIT
			MIN.	TYP.	MAX.	
82	TdC (WDTf)	Clock ↑ to $\overline{WDTOUT}$ ↓ Delay	—	—	120	ns
83	TdC (WDTr)	Clock ↑ to $\overline{WDTOUT}$ ↑ Delay	—	—	125	ns
84	TCWDTZ	$\overline{WDTOUT}$ out put period				
		WDT Mode 0	—	$T12*2^{16}$	—	ns
		WDT Mode 1	—	$T12*2^{18}$	—	ns
		WDT Mode 2	—	$T12*2^{20}$	—	ns
		WDT Mode 3	—	$T12*2^{22}$	—	ns

170890

Note1 : Timing Measurements are made at the following voltage.

Input VIH = 2.4V, VIL = 0.4V
VIHC = VCC-0.6V, VILC = 0.6V
Output VOH = 2.2V, VOL = 0.8V (Except CLKOUT)
CL = 100PF

4.6 AC Timing Charts (2) (in Inactive State)

4.6.1 AC Timing Charts of CGC (in Inactive State)

The following Figures show the timing charts in each operation mode with the CLKOUT pin connected to the CLKIN pin.

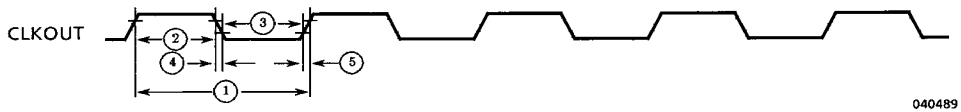


Figure 4.6.1 CLKOUT Waveform

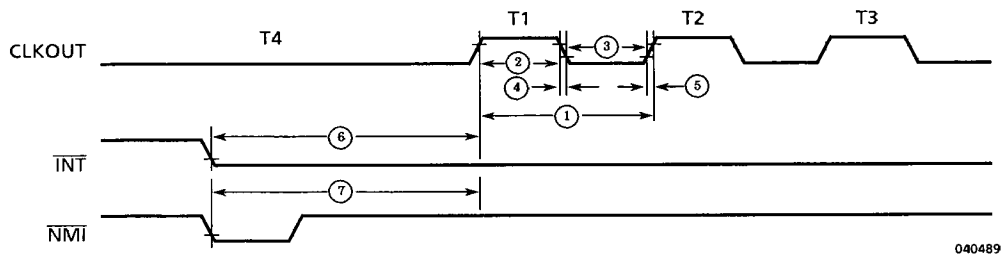


Figure 4.6.2 Clock Restart Timing (STOP mode)

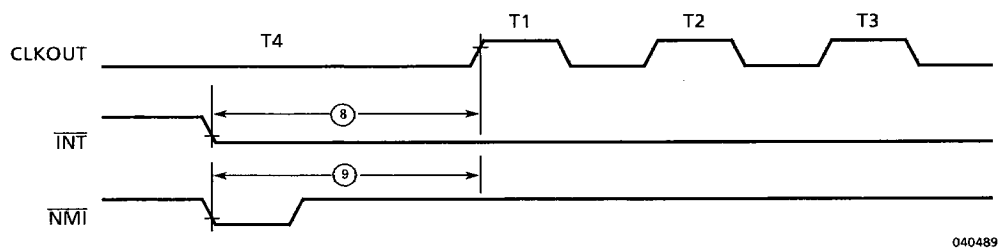


Figure 4.6.3 Clock Restart Timing (IDLE1 mode)

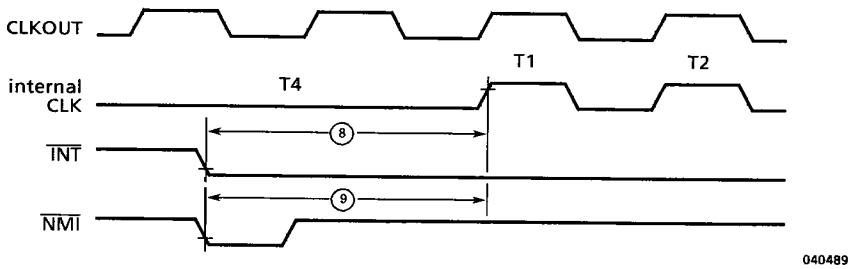


Figure 4.6.4 Clock Restart Timing (IDLE2 mode)

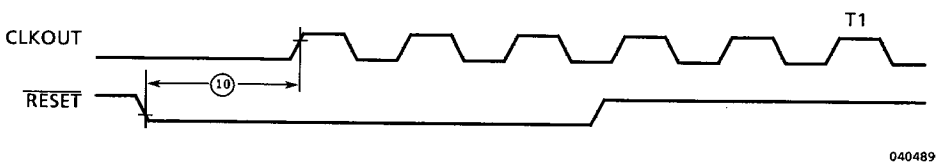


Figure 4.6.5 Timing of Clock Restart by $\overline{\text{RESET}}$ (IDLE1, IDLE2 mode)

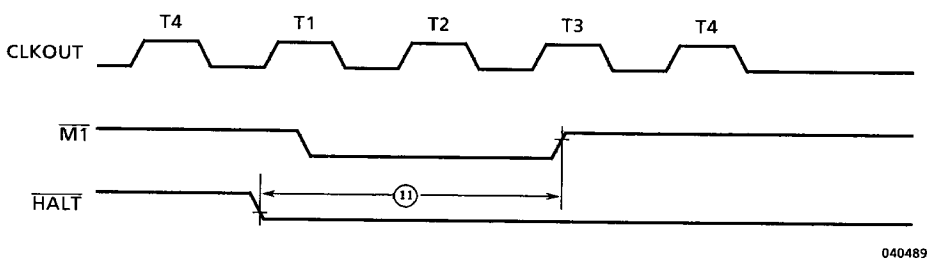


Figure 4.6.6 Clock Suspension Timing (IDLE1, IDLE2 and STOP modes)

4.6.2 AC Timing Charts of CTC (in Inactive State)

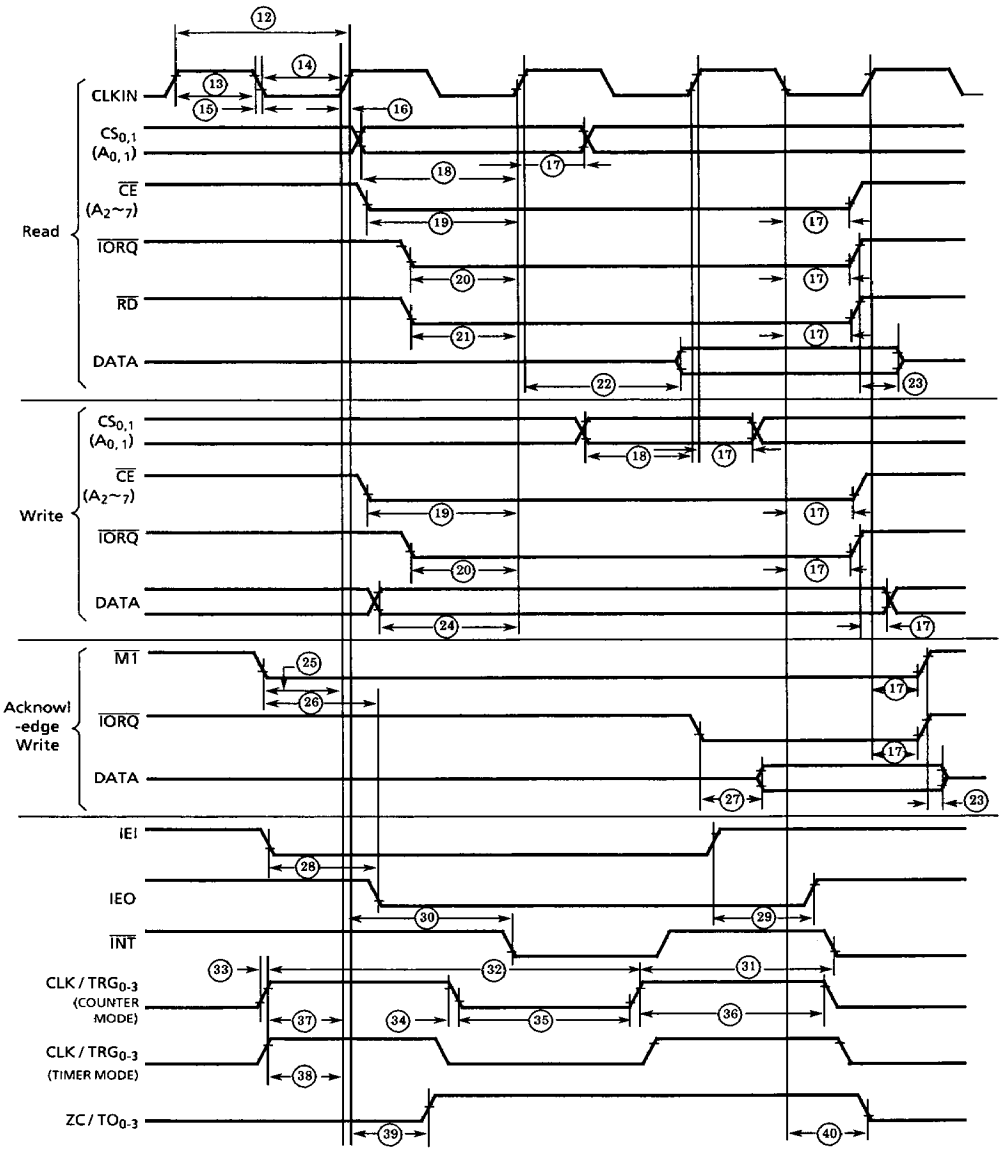
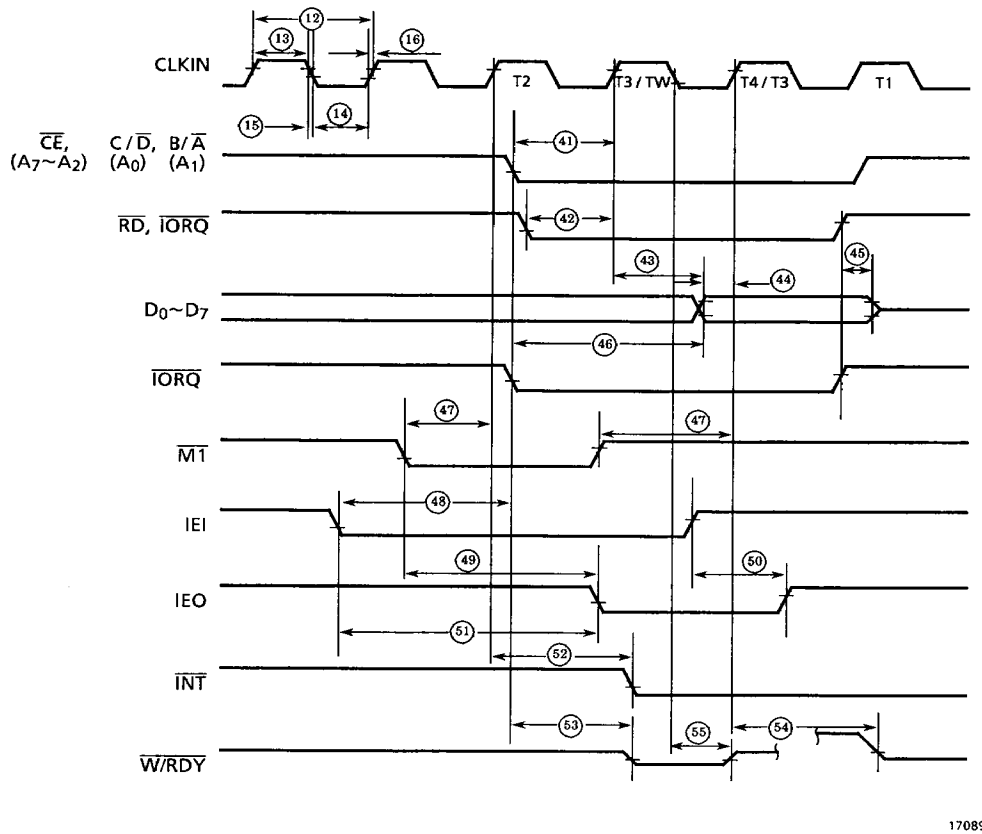


Figure 4.6.7 CTC Timing Diagram (Inactive)

4.6.3 AC Timing Charts of SIO (in Inactive State)



170890

Figure 4.6.8 SIO Timing Diagram (a) (Inactive)

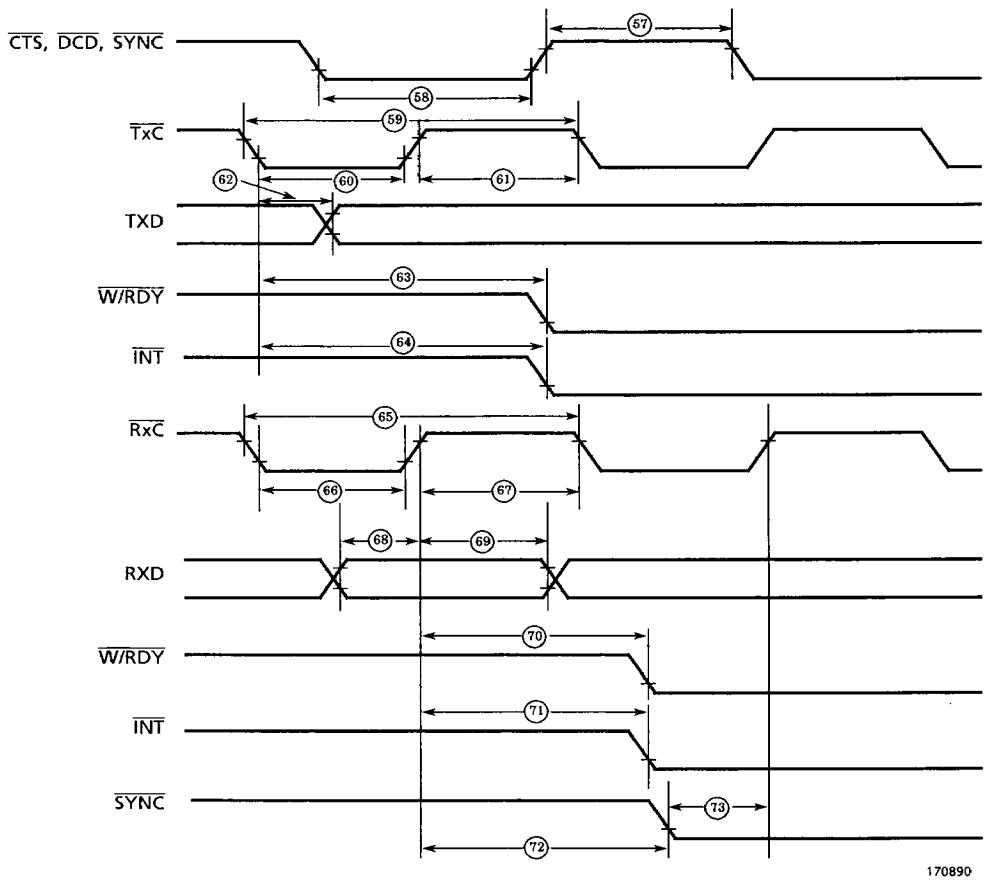


Figure 4.6.8 SIO Timing Diagram (b) (Inactive)

4.6.4 AC Timing Charts of WDT (in Inactive State)

(The mode setting and daisy chain interrupt setting registers on WDT)

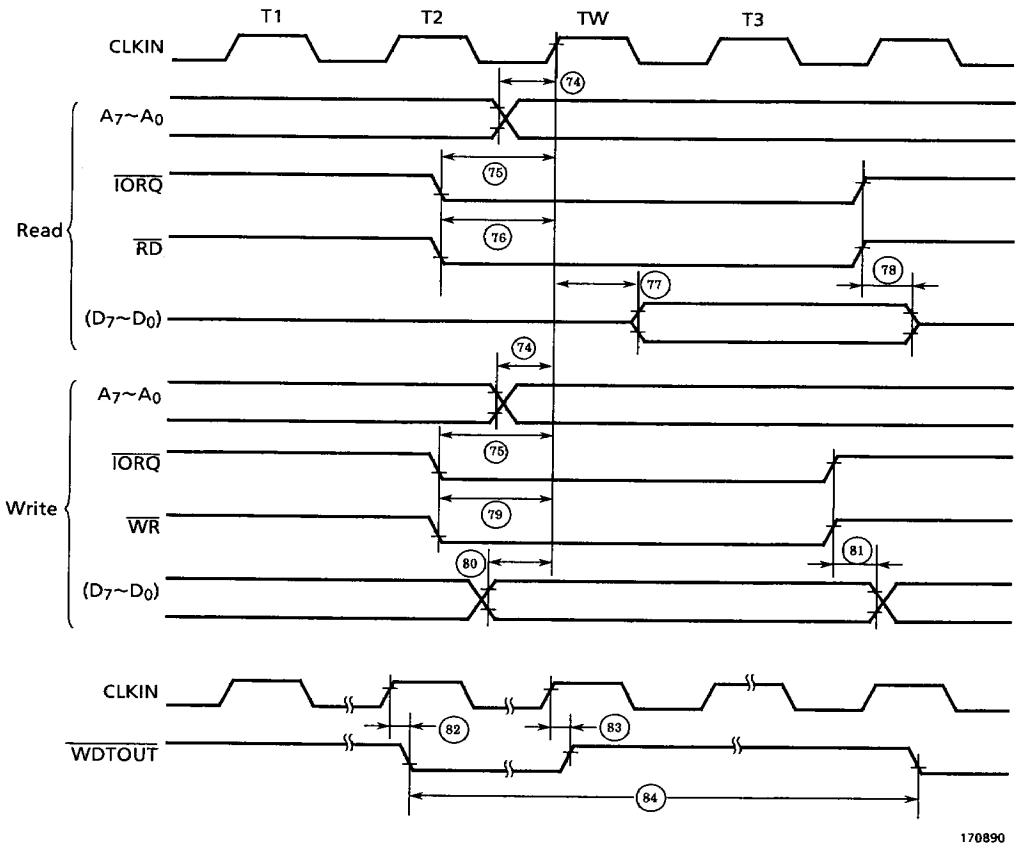


Figure 4.6.9 RD/WRITE, $\overline{\text{WDTOUT}}$ Timing Diagram

4.7 Pin Capacitance

SYMBOL	ITEM	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
CCLOCK	Clock Input Capacitance	F = 1MHz	–	–	TBD	PF
CIN	Input Capacitance	All terminals except that to	–	–		
COUT	Output Capacitance	be measured be earthed	–	–		

170890

5. EXTERNAL DIMENSIONS

QFJ84-P-S115