

## 1. General Descriptions

KL5C8400 is a fast 8 bit CPU, which is compatible with the Z80 microprocessor at binary level. With an internal 16 bit RISC-like architecture, the performance of the CPU is equivalent to the Z80 microprocessor at 44 MHz. With this high performance, the CPU is faster than typical 16 bit microcontrollers and CPUs. The advanced CMOS technology provides high performance at low power consumption.

The CPU has two operation modes controlled by CNFG input (mode pin). The CPU provides 1.2 times higher performance at the same clock rate in Z80 mode. In KC80 mode, the CPU provides 1.3 times higher performance at the same clock rate. There is also an advantage in memory access time in KC80 mode, because the CPU doesn't have M1 cycle.

In Z80 mode, the CPU operates in Z80 compatible bus cycles. Then, Z80 peripherals can be used with the CPU in mode 2 interrupt.

In KC80 mode, the CPU fetches its opcode using the

same bus cycle as memory read. Then, slower memory can be used with the CPU in KC80 mode. There is also a performance advantage because it fetches its opcode in shorter opcode fetch cycle. In KC80 mode, only mode 1 interrupt can be used.

### Features

- Instruction set: Fully compatible with Z80 MPU at binary level
- High speed operation: 33 MHz (equivalent performance of Z80 at 44 MHz)
- Minimum execution time: 90 ns (3 clocks)
- Low power consumption:
- Address space: 64 Kbyte
- Operational clock frequency: 0 ~ 33 MHz
- Interrupt: Maskable interrupt 1, Non-maskable interrupt 1
- Package: 44 pin QFP package
- Two operation modes: Z80 mode (CNFG = "H"), KC80 mode (CNFG = "L")

Instruction execution time comparison

instructions	KL5C8400 (Z80 mode)	KL5C8400 (KC80 mode)	Z80
LD B, C	4 clocks	3 clocks	4 clocks
ADD HL, BC	4 clocks	3 clocks	11 clocks
DEC DE	4 clocks	3 clocks	6 clocks
POP AF	10 clocks	9 clocks	10 clocks
JR Z, +20h	10 clocks	9 clocks	12 clocks

## 2. Block Diagram

Figure 1a shows the block diagram of KL5C8400.

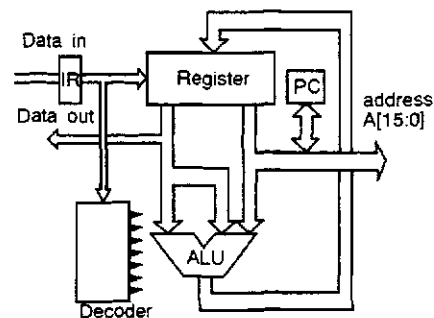
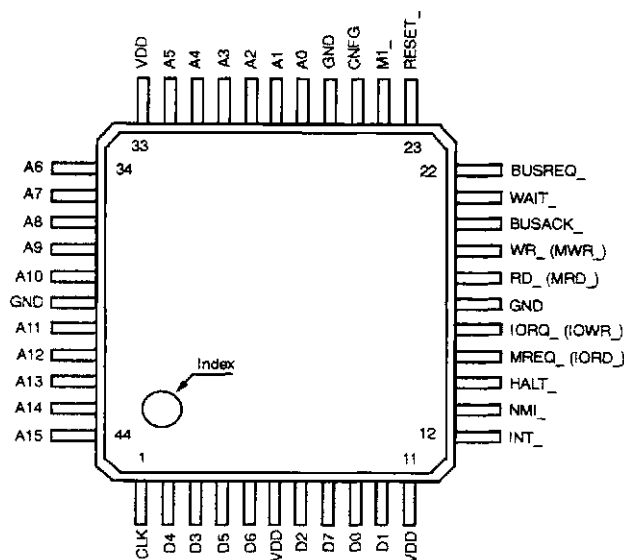


Fig.1a Block Diagram

### 3. Pin Description



( ) indicates signal names in KC80 Mode.

**Fig.1b Pin assignment**

pin name	I/O	description
GND	Power supply	Connect 0V.
VDD	Power supply	Connect 5V.
CNFG	I	Input for setting mode. When CNFG = "H", the setting is Z80 mode. When CNFG = "L", the setting is KC80 mode. Fix this pin to "H" or "L" on the target board.
RD_	O	Read (active "L", 3-state output, Z80 mode signal) This signal goes active while reading data from memory or an I/O device.
WR_	O	Write (active "L", 3-state output, Z80 mode signal) This signal goes active while writing data to memory or an I/O device.

pin name	I/O	description
IORQ_	O	I/O request (active "L", 3-state output, Z80 mode signal) This signal goes active while accessing I/O.
MREQ_	O	Memory request (active "L", 3-state output, Z80 mode signal) This signal goes active while accessing memory.
CLK	I	System clock
WAIT_	I	Wait (active "L") Wait states are inserted when this signal is "L".
A[15:0]	O	Address bus (3-state output)
D[7:0]	I/O	Data bus
RESET_	I	Reset (active "L") The CPU is initialized when this signal is "L".
NMI_	I	Non-Maskable interrupt (falling edge trigger) This input accepts the non-maskable interrupt. The priority of this input is higher than INT_ but lower than BREQ_. As soon as KL5C8400 completes a current instruction, it jumps to 0066H regardless of the status of interrupt enable flip-flop (IFF).
HALT_	O	Halt (active "L") This signal indicates that the CPU executes HALT instruction and is in the HALT status. The CPU resumes from the HALT status by receiving NMI, INT or reset. During HALT status, the CPU continuously executes NOP instructions.
INT_	I	Maskable interrupt (active "L") This input accepts the interrupt from I/O devices. If the interrupt enable flip-flop (IFF) of the CPU is set, and BUSREQ_ is inactive, the CPU completes the execution of the current instruction, and starts interrupt service.

pin name	I/O	description
M1_	O	Machine Cycle One (active "L") This signal indicates that the current machine cycle is the opcode fetch cycle of a executing instruction.
BUSREQ_	I	Bus request (active "L") The adress bus and the data bus go to high-impedance state after completion of the current instruction execution when this signal goes active. This signal has higher priority than NMI_ and INT_.
BUSACK_	O	Bus acknowledge (active "L") This signal indicates that the CPU released the bus for an external bus master.
MRD_	O	Memory read (active "L", 3-state output, KC80 mode) This signal goes active while reading data from a memory.
MWR_	O	Memory write (active "L", 3-state output, KC80 mode) This signal goes active while writing data to a memory.
IORD_	O	I/O read (active "L", 3-state output, KC80 mode) This signal goes active while reading data from an I/O device.
IOWR_	O	I/O write (active "L", 3-state output, KC80 mode) This signal goes active while writing data to an I/O device.

INT\_, NMI\_, WAIT\_, BUSREQ\_, RESET\_ are provided with internal pull-up resistors.  
About those characteristics (the pull-up current value), see chapter 8.

## 4. Register architecture

### 4.1 Special purpose registers

#### Program Counter (PC)

Program Counter holds the address of the next instruction. The next instruction is fetched from memory address Program Counter indicates.

#### Stack Pointer (SP)

Stack Pointer holds the current top address of stack area in RAM.

#### Index registers (IX, IY)

Index registers hold base address for indexed addressing. There are two index registers. The one is IX, and the other IY.

#### Interrupt page address register (I)

Interrupt page address register holds indirect upper 8 bit address for indirect jump in Mode 2.

#### Memory refresh register (R)

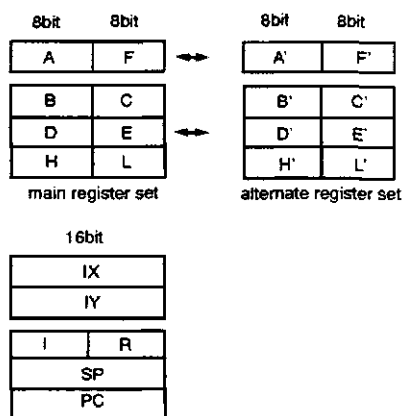
This register automatically increments by one on instruction fetch. Bit D7 of this register will not change by increments on D6-D0.

#### Accumulator (A, A'), and Flag register (F, F')

There are two 8 bit accumulators (A, A') and two flag registers. Accumulator holds the result of arithmetic and logical operation. Flag register holds status of 8 bit or 16 bit operation. The instruction, EX AF, AF' exchanges A, F and A', F'.

### 4.2 General purpose registers

There are two sets of general purpose registers. Each of them can be used as an 8 bit register independently (B, C, D, E, H, L or B', C', D', E', H', L'). Paired registers (BC, DE, HL, BC', DE' and HL') can be used as a 16 bit register. The instruction, EXX exchanges B,C, D, E, H, L and B',C', D', E', H', L'.

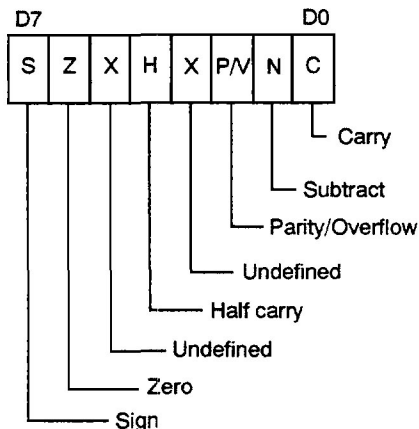


Special purpose registers

Fig.2 KL5C8400 register architecture

## 5. Flags

Bit assignment of the Flag register.



**Fig.3 Flag register**

The function of each bit in the Flag register is defined as below.

### Carry flag (C)

Carry flag is set by carry produced by Accumulator from the MSB at execution of add instructions, subtract instructions, shift and rotate instructions, etc.

### Zero flag (Z)

Zero flag is set if the operation result is zero, when add instructions, subtract instructions, logical instructions including INC, DEC, DAA instructions. This also occurs on block I/O instructions, rotate and shift instructions, string search instructions, bit test instructions and data detection of block search instructions.

### Sign flag (S)

Sign flag is set when the result of sign number operations is negative.

### Parity/Overflow flag (P/V)

Parity/Overflow flag has two functions. It indicates the parity on logical operations, and also the overflow on arithmetic operations. The overflow means that the

content of Accumulator is less than -128 in 2's complement, or larger than 127. It is also set by block search instructions, block I/O instructions, LD A, R instruction and LD A, I instruction.

### Half carry flag (H)

Half carry flag is set when the carry or borrow from lower 4 bits is produced.

### Subtract flag (N)

Subtract flag is set on subtract instructions. KC82 checks if the previous instruction is addition or subtraction with this flag on DAA instruction.

## 6. Functional description and timing

In this section, functional description and timing of Z80 mode are described. KC80 mode is described only the difference between Z80 mode in section 7.

### 6.1 Basic operation (Instruction cycle)

Basic operation of KL5C8400 can be divided into the following 5 cycles.

- 1) Opcode fetch cycle - Fetches opcode of an instruction from memory.
- 2) Memory read cycle - reads data from memory
- 3) Memory write cycle - writes data to memory
- 4) I/O read cycle - reads data from I/O device
- 5) I/O write cycle - writes data to I/O device

Follows are the description of the each machine cycles.

#### Opcode fetch cycle

As shown Figure 4, the CPU fetches one byte of an instruction in 4 clock cycles unless there is a wait. During this cycle, MREQ\_, RD\_ and M1\_ go active. This cycle appears at the end of the instruction execution sequence. the CPU outputs the refresh address to the address bus in the latter half of the opcode fetch cycle. Note that the CPU doesn't have the refresh signal. When WAIT\_ goes active ("L") at the rising edge of  $T_2$ , the wait cycle  $T_w$  is entered.

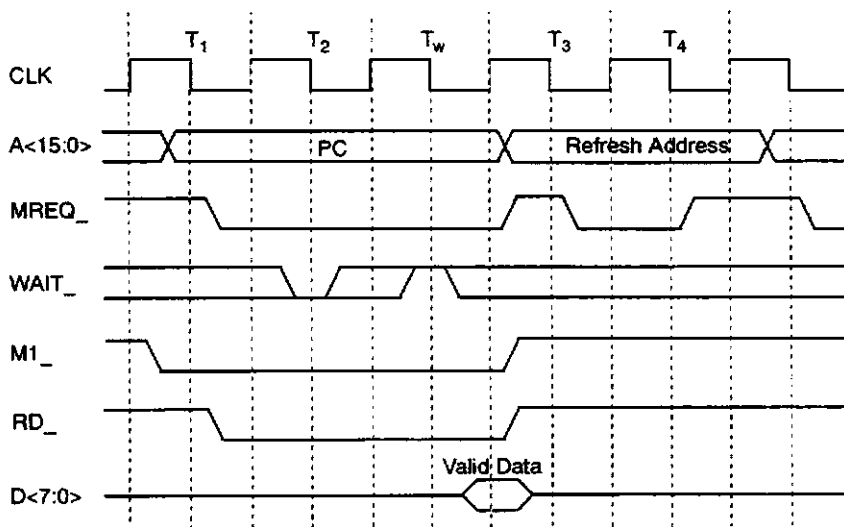
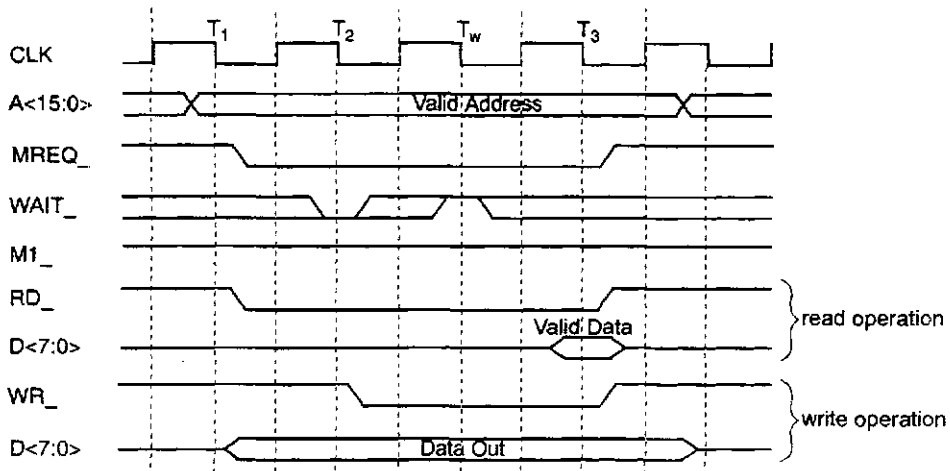


Fig.4 Opcode fetch cycle (Z80 mode)

**Memory read/write cycle**

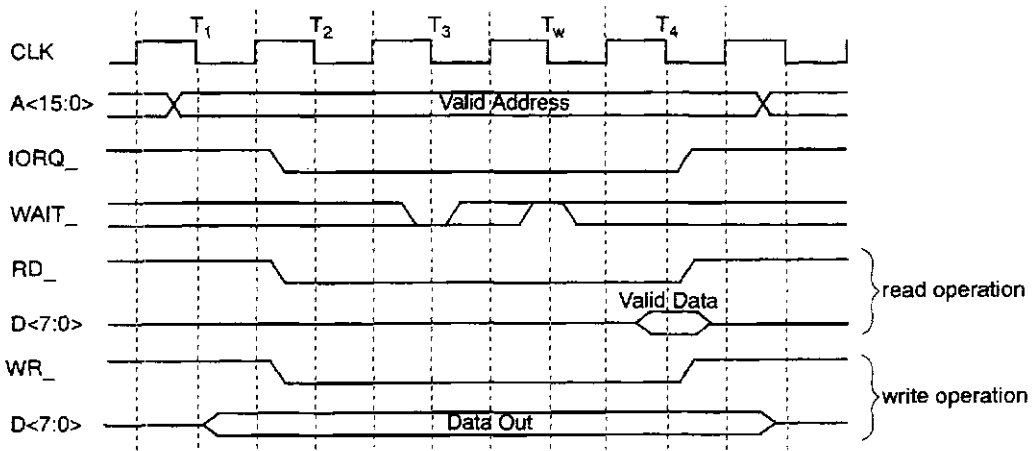
The difference between the opcode fetch cycle is whether M1\_ goes active or not, and is one bus cycle in three -clock cycles. When WAIT\_ goes active ("L") at the rising edge of T<sub>2</sub>, the wait cycle T<sub>w</sub> is inserted.



**Fig.5 Memory read/write cycle (Z80 mode)**

**I/O read/write cycle**

This cycle is minimum of four-clock cycles. When WAIT\_ goes active ("L") at the rising edge of T<sub>3</sub>, the wait cycle T<sub>w</sub> is inserted.



**Fig.6 I/O read/write cycle (Z80 mode)**



### Instruction Prefetch cycle

The CPU has the prefetch cycle at the end of instruction execution cycle all the time. Figure 8 shows that the CPU executes the instruction sequence indicated in Figure 7 as an example.

↔ (referred as "arrow" hereinafter) in the Figure 8 denotes the prefetch cycle. The arrow ① part prefetches the instruction at n+3 (ADD A, D), the arrow ② part is for prefetching instruction at n+4 (opcode of JP instruction, C3H), the arrow ③ part is for prefetching 77H at 1000H (LD [HL], A instruction). As shown, these cycles appear at the end of execution of current instruction.

address	mnemonic	code
n	LD A, [1234H]	3A 34 12
n+3	ADD A, D	82
n+4	JP 1000H	C3 00 10
.	.	.
1000H	LD [HL], A	77
1234H	.	5A

Fig.7 Source code for Fig. 8

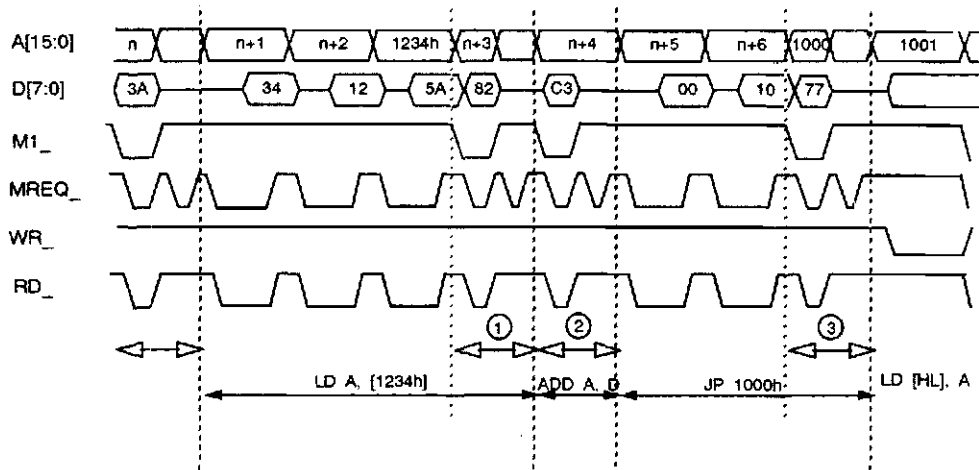


Fig.8 Prefetch cycle

(↔ denotes the prefetch cycle. Assumes memory location 1000H has 77H, memory location 1234H has 5AH, and D register has 81H.)

### Special case for prefetch.

#### (Discard of the prefetched instruction)

For example, under normal condition, the prefetched instruction, at address 1000H, data 77H at ③ in Figure 8 on execution of the instruction, JP 1000H is held and execute correctly. However, on interrupt the prefetched instruction will be discarded and refetch on return from an interrupt service routine. On interrupt recognition, PC is decremented by one before saved onto stack. On RETI or RETN instruction the PC will be loaded with the address of the discarded instruction. Figure 13 has the timing diagram for the case of receiving interrupt in mode 1 when executing LDIR instruction. In the figure, the data "EDH" prefetched at ③, is discarded and then go to interrupt service. On return, the CPU resume from the prefetch of this "EDH". For the bus request cycle, the data "EDH" prefetched at ③ prior to the bus request will be kept, and on bus release the CPU starts to prefetch from the next address to the "EDH".

### 6.2 Bus release (Bus request/acknowledge cycle)

Under normal operation, the CPU holds the control of address bus, control bus (RD\_, WR\_, MREQ\_ and IORQ\_). However, if there is an external bus request (BUSREQ\_ = "L"), address bus goes to high-impedance state, all the interrupts are disabled, BUSACK\_ turns to "L", and the CPU releases the bus control to the external device. By using this feature, data transfer without CPU intervention (DMA transfer) can be done. Figure 9 shows basic timing for the bus request cycle. This cycle continues while BUSREQ\_ stays "L". Note that bus request is accepted at the end of the bus cycle.

### 6.3 Interrupt and timing

The CPU can handle the following two kinds of interrupts.

- 1) maskable interrupt on INT\_
  - 2) Non-maskable interrupt on NMI\_
- 2) has the higher priority over 1). If both request made simultaneously, 2) will be accepted over 1).

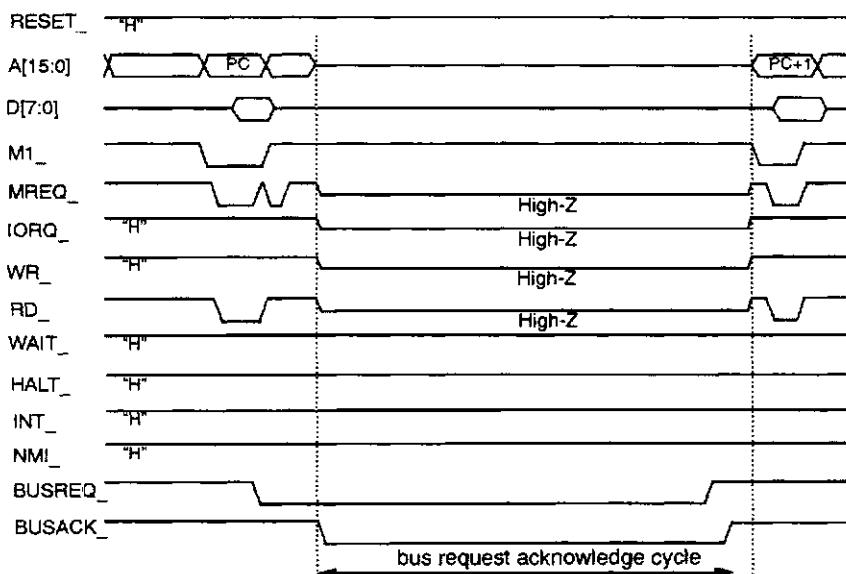


Fig.9 Bus request timing

### Maskable interrupt

The EI instruction enables interrupts, and the DI instruction disables interrupts. The control of interrupt is implemented using two flip-flops (IFF1 and IFF2). Figure 10 has the state table for these flip-flops.

The maskable interrupt will be accepted if all of the following conditions are met:

1) Both IFF1 and IFF2 are set. (The EI instruction and RETN instructions change the status of these flags after the execution of the following instruction. So if there are EI instruction followed by DI instruction, the interrupt request will not be accepted.)

2) BREQ\_ is inactive ("H").

3) No NMI\_ falling edge has been detected.

As shows in the Figure 11, the CPU reads the interrupt vector when M1\_ and IORQ\_ go active ("L"). When WAIT\_ goes active ("L") at the rising edge of T<sub>4</sub>, the wait cycle T<sub>w</sub> is entered.

### Maskable Interrupt modes

The CPU has the following three different interrupt handling modes, and interrupt sequence is different by each modes. The following descriptions are the operations in each mode.

events	IFF1	IFF2	
reset	0	0	
DI instruction	0	0	
EI instruction	1	1	
INT accepted	0	0	
NMI accepted	0	-	
RETN instruction accepted	IFF2	-	IFF2 is copied into IFF1
LD A, I instruction	-	-	IFF2 is copied into P/V
LD A, R instruction	-	-	IFF2 is copied into P/V

Fig.10 State diagram for IFF1 and IFF2

("-" denotes remain unchanged.)

#### 1) Mode 0

This is the default mode and set to this on reset automatically. Also, executing "IM 0" instruction sets the CPU to this mode. In this mode, the CPU executes the instruction read during interrupt acknowledge cycle. Usually the instruction to be used in this mode is RST instructions (RST 00H ~ RST 38H). Figure 12 show each timing for RST instructions. For the all interrupt modes, if INT\_ goes active during execution of a mode change instruction (IM 0 ~ 2), the new mode become effective right after these instructions.

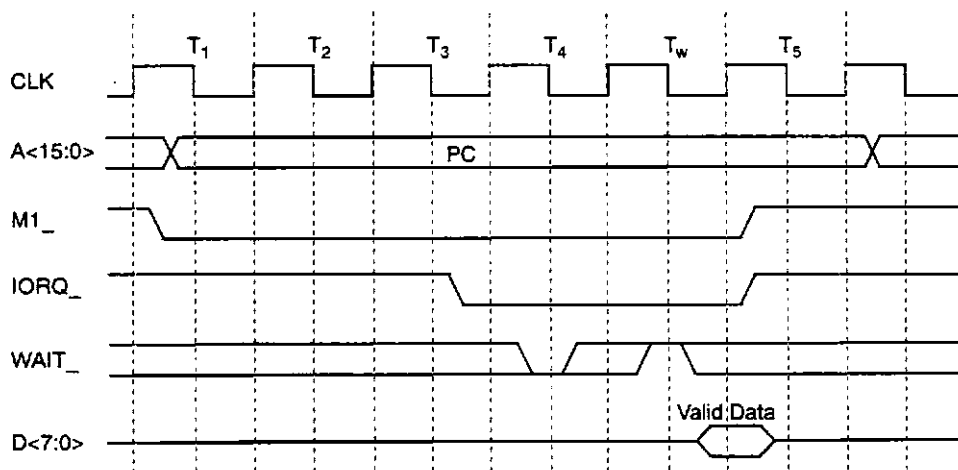


Fig.11 Z80 special M1 cycle (Z80 mode)

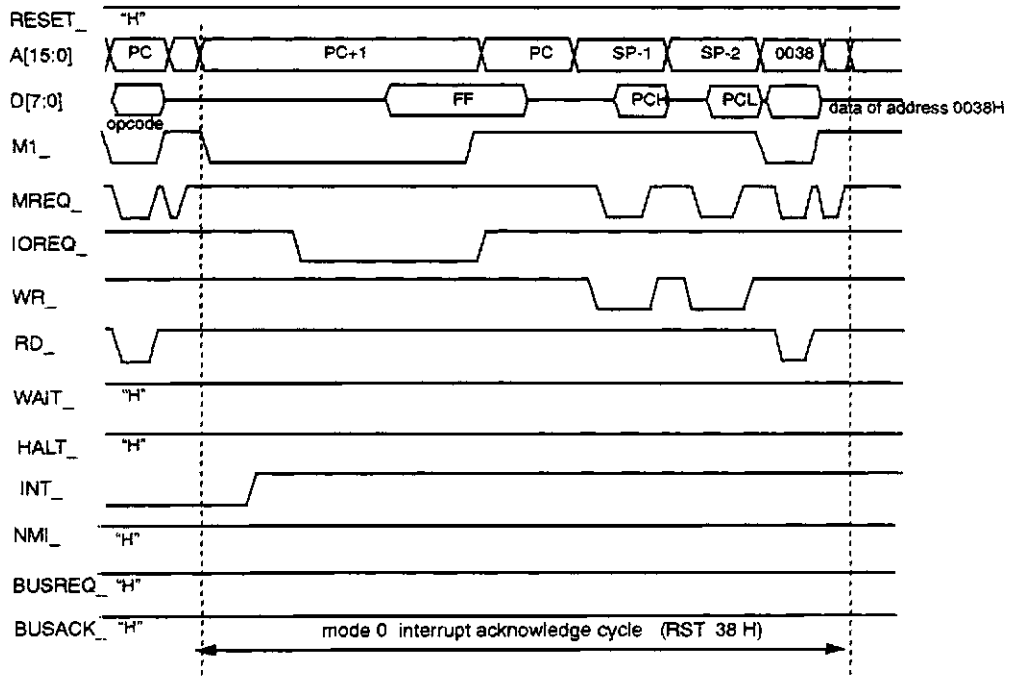


Fig.12 Mode 0 timing

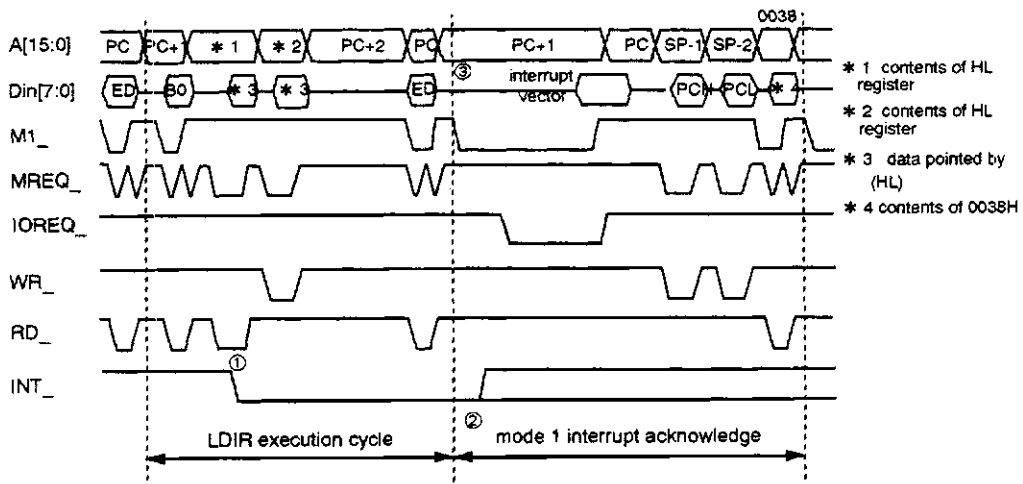


Fig.13 Interrupt acknowledge cycle (Mode 1 timing)

(The figure shows the case when the interrupt occurred at ①, accepting it at the instruction boundary ②. This case restarts from the fetch of ED at ③ on the return from an interrupt.)

## 2) Mode 1

The "IM 1 instruction" sets the CPU in this mode. In this mode, the CPU saves the contents the PC (Program Counter) onto the stack, ignores the data read during the interrupt acknowledge cycle, and executes "RST 38H" instruction internally. Figure 13 has the timing for Mode 1 interrupt.

## 3) Mode 2

The "IM 2 instruction sets" the CPU to this mode. In this mode, as shown in the Figure 14, the 8-bit wide

vector with 0 in LSB read during the interrupt acknowledge cycle as a lower byte, and the content of the I register as a upper byte form up a pointer which points an entry in a table of address for the interrupt service routines. After forming up the 16-bit address, the CPU fetches the table location and gets the start address, saves the return address onto the stack, then jumps to the service routine. Figure 15 has the timing diagram for Mode 2 interrupt acknowledge cycle.

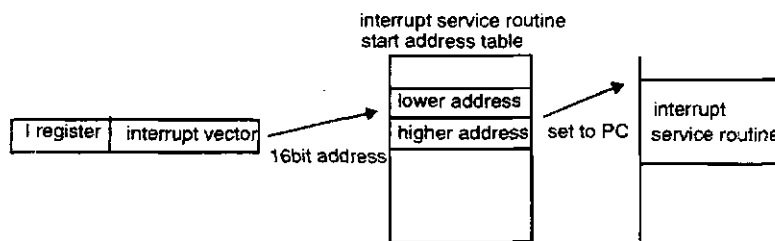


Fig.14 Mode 2 interrupt

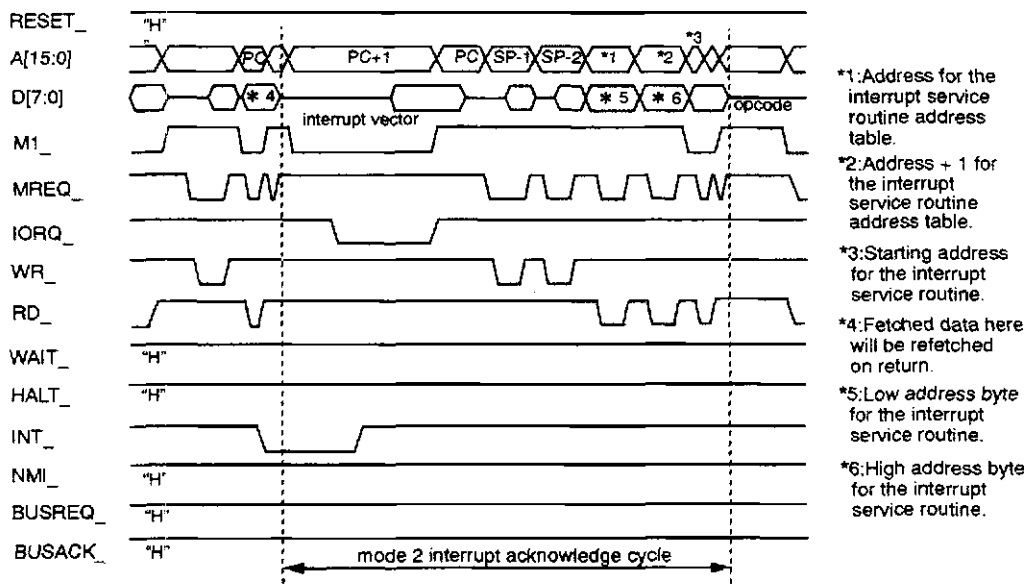


Fig.15 Mode 2 timing

### Timing on interrupt acceptance

The interrupt is accepted at the end of instruction. In case of block move instruction, the interrupt accepted at the end of repeat movement. The Figure 13 show the interrupt acknowledge timing. In this figure, an interrupt is accepted during block move instruction. ④ shows that when an interrupt request occurs, the interrupt will not be accepted until the prefetch cycle of the last instruction, ⑤. In this case, the data "EDH" fetched at ⑥ will not be kept by the CPU, and it will be refetched on the return from an interrupt.

### Non-maskable interrupt

Non-maskable interrupt is an interrupt which can not be masked by software. The falling edge of the signal NMI\_ is latched internally, and checked on the clock falling edge of the last cycle of every instruction. Non-maskable interrupt takes place if BREQ\_ is inactive. When a non-maskable interrupt is accepted, the CPU saves the contents of the PC onto the stack, and jumps to 0066H. Return from non-maskable service is done by "RETN" instruction. Figure 16 shows the timing for NMI\_ acknowledge cycle.

### Enable/disable of interrupts

Enable/disable of an interrupt is controlled by two flip-flops - IFF1 and IFF2. Refer to Figure 10 on the state diagram for these flip-flops. As shown, maskable interrupt will be accepted when both IFF1 and IFF2 set to 1. After reset and execution of DI instruction maskable interrupt is disabled, because both of these flip-flops are reset. An interrupt can be accepted while both IFF1 and IFF2 are set to 1 by execution of an EI instruction. The reason there are two flip-flops is to memorize the status of EI/DI condition with IFF2 on acceptance a Non-maskable interrupt. For example, if both IFF1 and IFF2 are '1' (maskable interrupt is enabled) on acceptance a non-maskable interrupt, in a non-maskable interrupt acknowledge cycle IFF1 is cleared to 0 and maskable interrupts are disabled. At this stage, IFF2 still holds "1" and the content of IFF2 is copied back into IFF1 on RETN instruction so that maskable interrupts could be enabled automatically again.

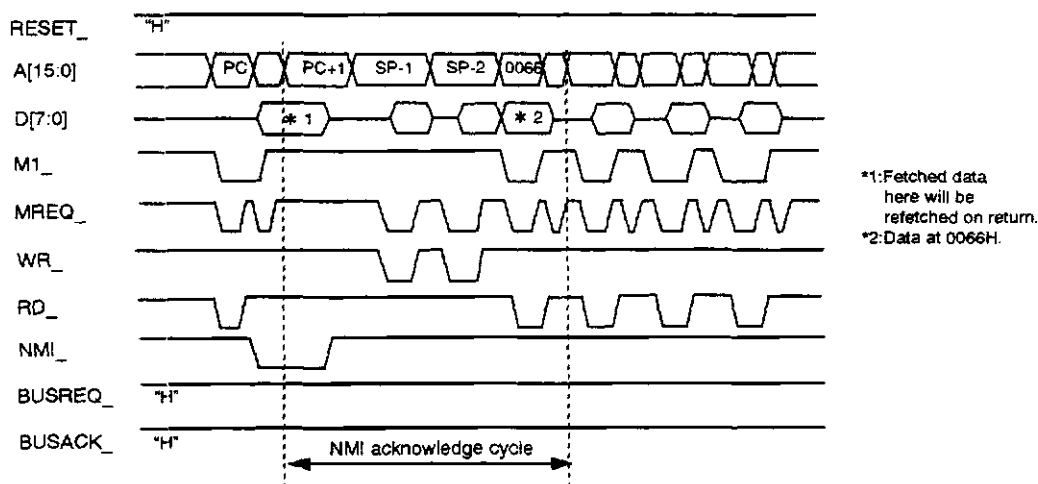


Fig.16 Non-maskable interrupt timing

#### 6.4 Halt enter and exit

By executing the HALT instruction, the CPU enters into the HALT cycle. In this cycle, the CPU continues to execute NOP instruction internally. The CPU exits from this cycle by either reset, or interrupts (Non-maskable interrupt or interrupt with IFF set). Figure 17 shows timing for the case of acceptance an interrupt in mode 2 during the HALT cycle. Then the CPU exits from the HALT cycle. The HALT cycle made up with two separate bus cycles; the first cycle is idle cycle and the second cycle is opcode fetch cycle. The address lines holds the next address to the HALT instruction during this cycle. The instruction fetched during the second opcode fetch cycle will not be read by the CPU. The INT\_ input is sampled on the falling CLK edge of the second cycle, and forces HALT\_ back to "H".

#### 6.5 Reset timing

By keeping RESET\_ input "L" for the minimum three-clock cycles, the CPU is reset. During reset cycle, the address bus goes to high-impedance state. When RESET\_ goes inactive ("H"), opcode fetch cycle will be initiated, and starts from 0000H. The interrupt mode is set to mode 0. IFF1, IFF2 flags and I, R registers are reset.

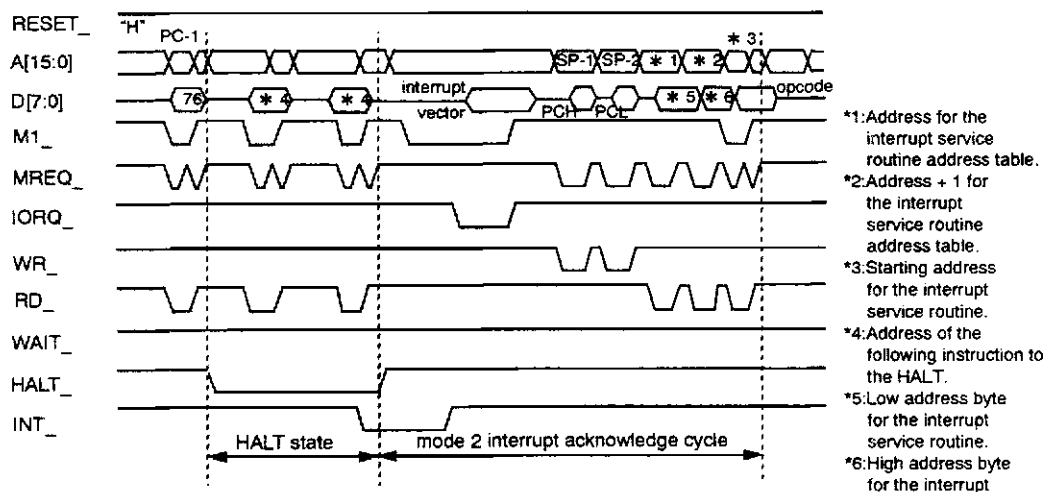


Fig.17 HALT Exit (Mode 2)

## 7. KC80 Mode

KL5C8400 operates in KC80 mode when the CNFG\_ input is "L". In KC80 mode, the CPU fetches its opcode using the same bus cycle as the memory read. Then, slower memories can be used with the CPU in KC80 mode. There is also a performance advantage because it fetches its opcodes in shorter opcode fetch cycles. In KC80 mode, only mode 1 interrupt can be used.

In KC80 mode, the CPU outputs MRD\_, MWR\_, IORD\_ and IOWR\_ instead of MREQ\_, IORQ\_, RD\_ and WR\_ in Z80 mode. Therefore each cycle of the memory read/write cycle and the I/O read/write cycle changes as shown in Figure 18 and 19.

Figure 18 shows the memory read/write cycle. The memory read/write cycle and the opcode fetch cycle are three-clock bus cycles. When WAIT\_ goes active ("L") at the rising edge between T<sub>2</sub> and T<sub>3</sub>, the wait cycle T<sub>w</sub> is inserted.

In KC80 mode, the opcode fetch cycle is the same as the memory read cycle except M1\_ signal. Note that M1\_ is inactive ("H") in the opcode fetch cycle but goes active ("L") in the memory read/write cycle.

Figure 19 shows the I/O read/write cycle. The I/O read/write cycle is four-clock bus cycles. When WAIT\_ goes active ("L") at the rising edge between T<sub>3</sub> and T<sub>4</sub>, the wait cycle T<sub>w</sub> is inserted. Note that M1\_ goes active ("L") also in the I/O read/write cycle.

### Interrupt acknowledge cycle in KC80 mode

In KC80 mode, only mode 1 interrupt can be used as maskable interrupt. Figure 19 shows maskable interrupt (mode 1) timing. When WAIT\_ goes active ("L") at the rising edge between T<sub>4</sub> and T<sub>5</sub>, the wait cycle T<sub>w</sub> is inserted as well as in Z80 mode.

Non-maskable interrupt (NMI) can be naturally used in KC80 mode.

### Other timings

The operations in KC80 mode is the same as ones in Z80 mode except the basic bus cycles (memory read/write cycle, I/O read/write cycle, interrupt acknowledge cycle) shown Figure 18, 19 and 20.

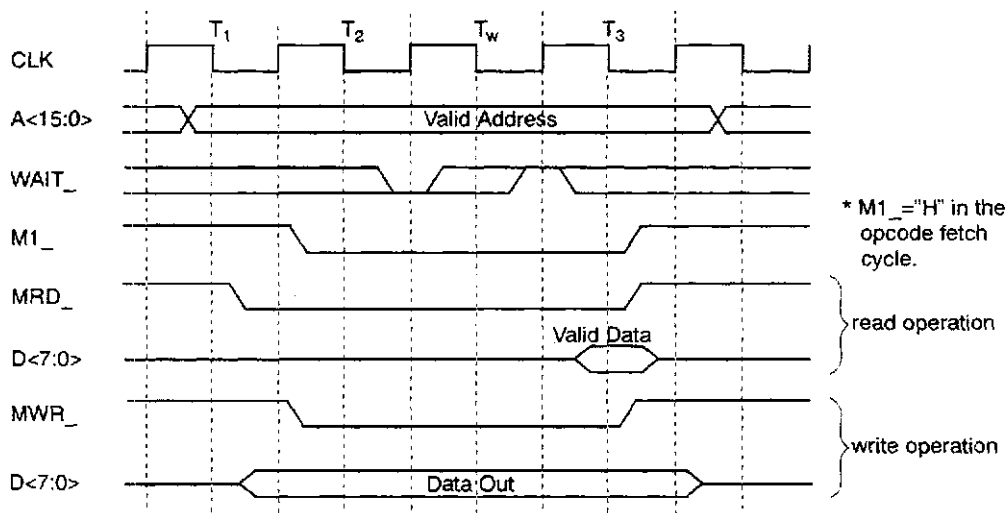


Fig.18 Memory read/write cycle (KC80 mode)



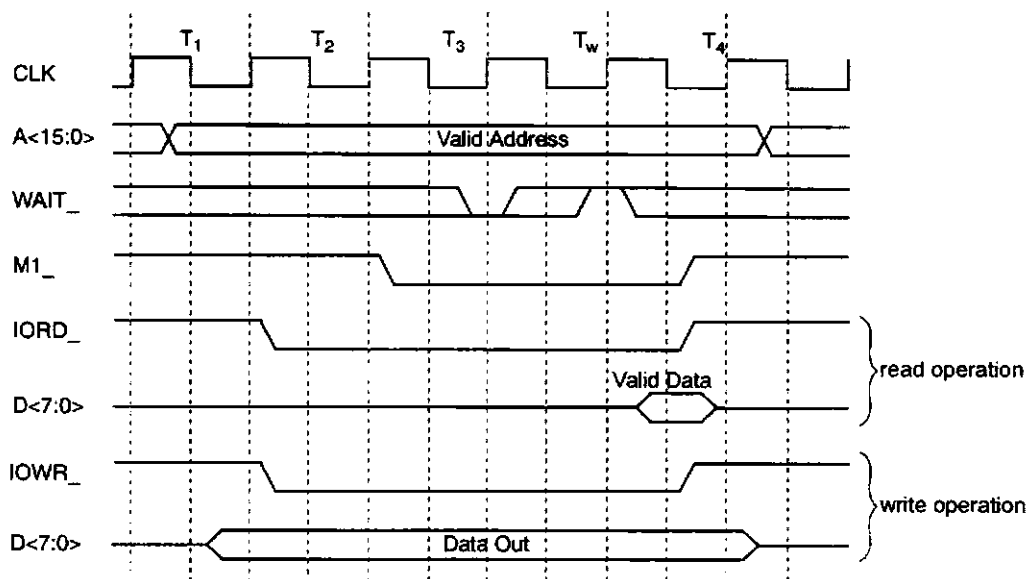


Fig.19 I/O read/write cycle (KC80 mode)

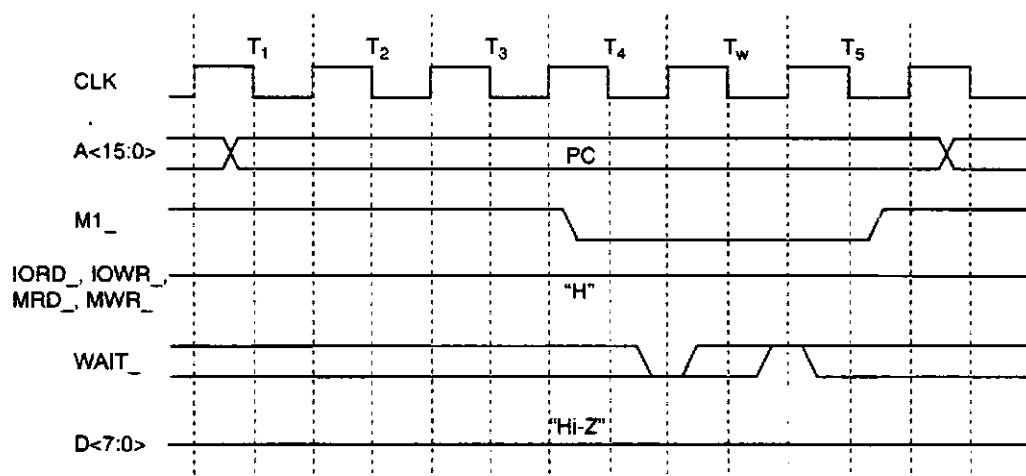


Fig. 20 Mode 1 interrupt accepting timing (KC80 mode)

## 8. Electrical Characteristics

### 8.1 Absolute Maximum Ratings

Table 8-1 Absolute Maximum Ratings (with respect to GND)

Item	Symbol	Rating	Unit
Power supply voltage	VDD	-0.6 ~ +7.0	V
Input voltage	VIN	-0.6 ~ VDD+0.6	V
Storage temperature	TSTG	-40 ~ +125	°C

### 8.2 D.C. Characteristics (5V ± 10%)

Table 8-2 Recommended Operation Conditions

Item	Symbol	Rating	Unit
Power supply voltage	VDD	4.5 ~ 5.5	V
Ambient temperature	TA	0 ~ +70	°C

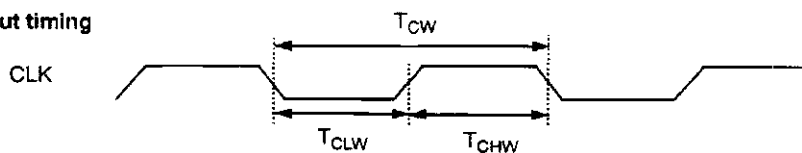
Table 8-3 Electrical Characteristics (under recommended operation conditions)

Item	Symbol	Rating			Unit	Test condition
		Min.	Typ.	Max.		
Input voltage (all input pins except RESET_)	V <sub>IH</sub>	3.6	—	VDD	V	
	V <sub>IL</sub>	GND	—	1.4	V	
RESET_ input pin (Schmitt trigger input)	V <sub>+</sub>	2.4	—	4.0	V	
	V <sub>-</sub>	0.9	—	2.3	V	
	V <sub>h</sub>	0.9	—	—	V	
Output voltage	V <sub>OH</sub>	3.5	—	—	V	I <sub>OH</sub> =-4mA
	V <sub>OL</sub>	—	—	0.4	V	I <sub>OL</sub> =4mA
Output current	I <sub>OUT</sub>			± 4	mA	
Input leakage current	I <sub>IL</sub>	-10	—	—	μA	V <sub>IN</sub> =GND
	I <sub>IH</sub>	—	—	10	μA	V <sub>IN</sub> =VDD
Output leakage current	I <sub>OZ</sub>	-10	—	10	μA	at high impedance output
Pull-up current	I <sub>PU</sub>	20	95	250	μA	V <sub>IN</sub> =GND
Standby supply current	I <sub>DDS</sub>		1.0*	100	μA	CLK stop, all pins="H"
Power supply current	I <sub>DDOP</sub>		27		mA	f(CLK)=20MHz

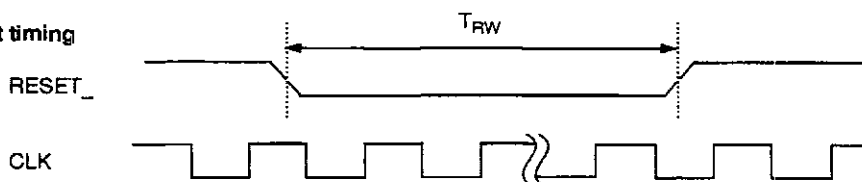
\* at TA=25 °C

## 8.3 A.C. Characteristics

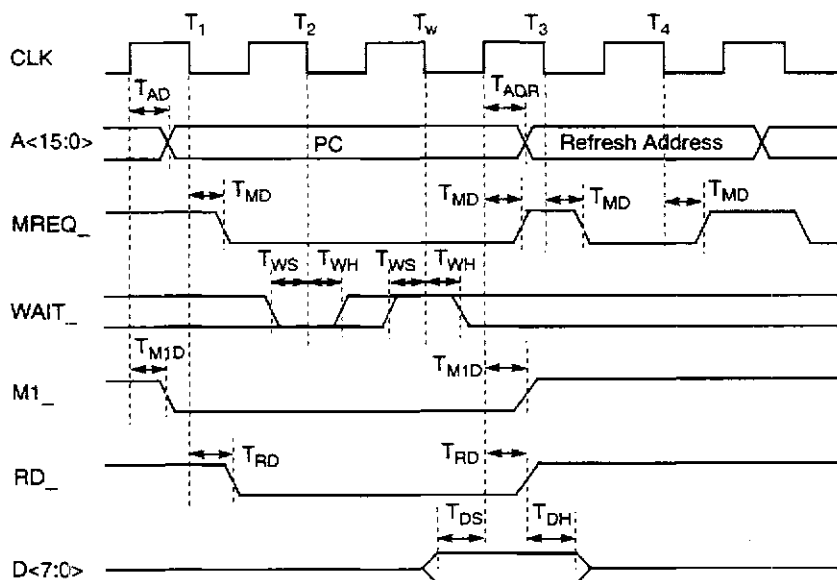
## Clock output timing



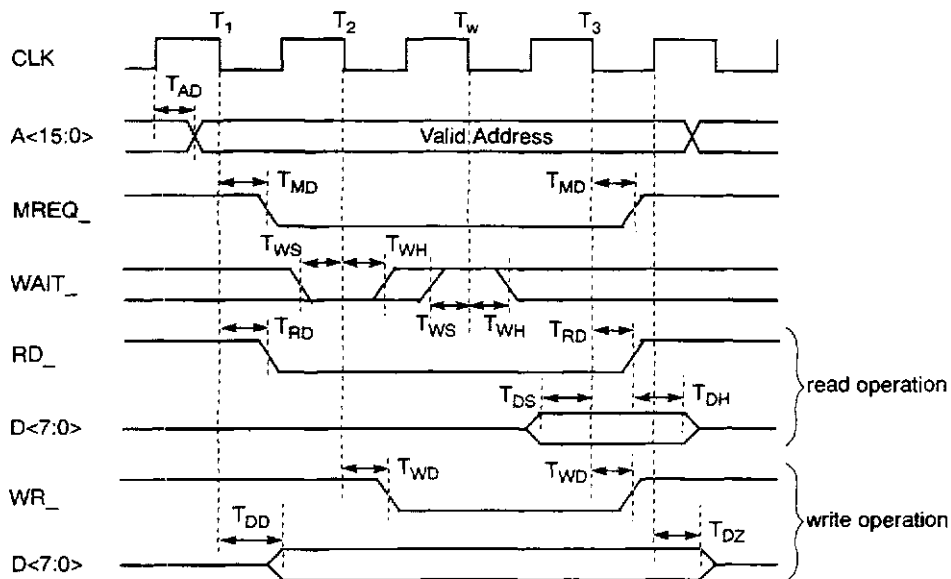
## Reset input timing



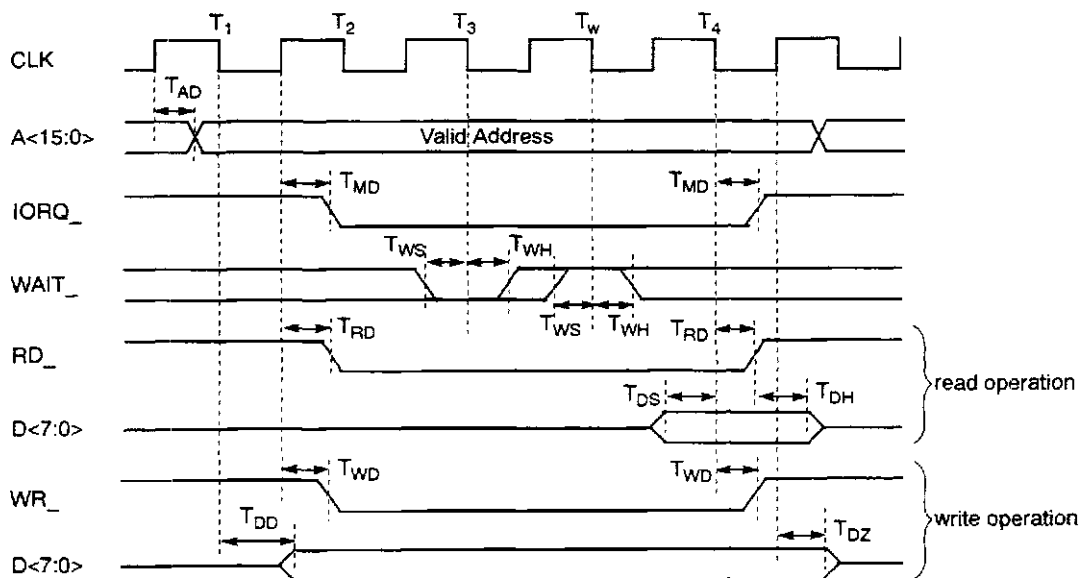
## Opcode fetch cycle timing (Z80 mode)



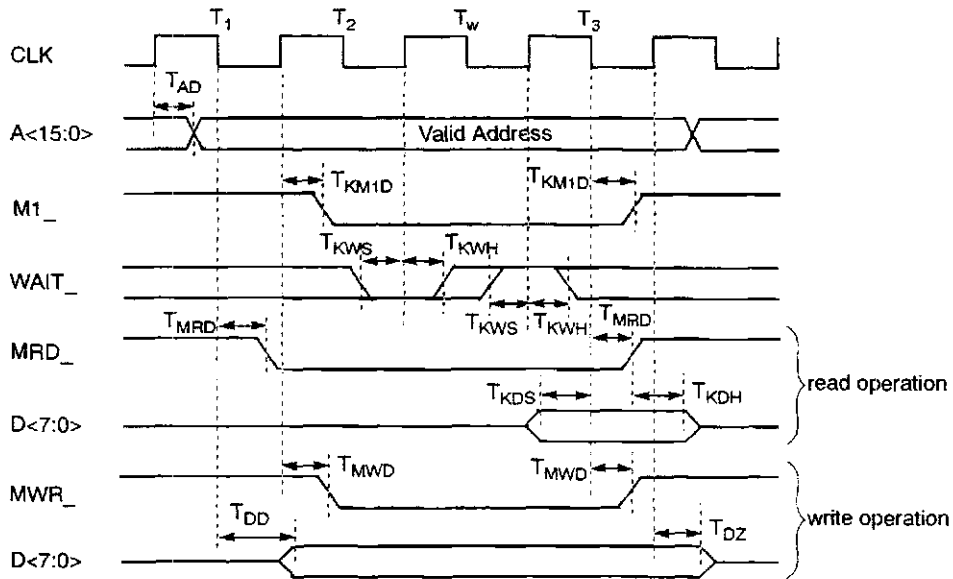
## Memory read/write cycle timing (Z80 mode)



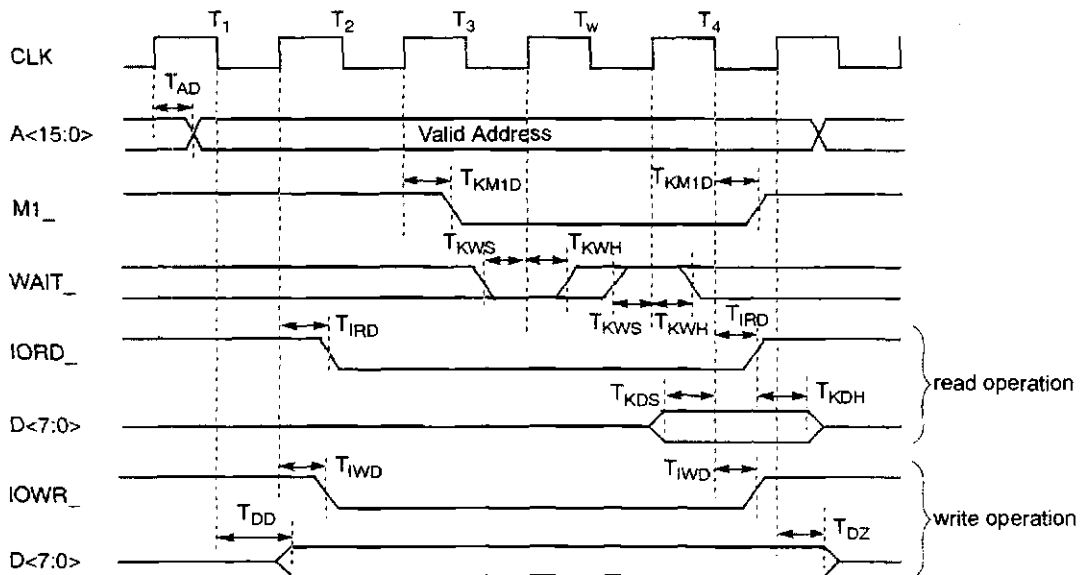
## I/O read/write cycle timing (Z80 mode)

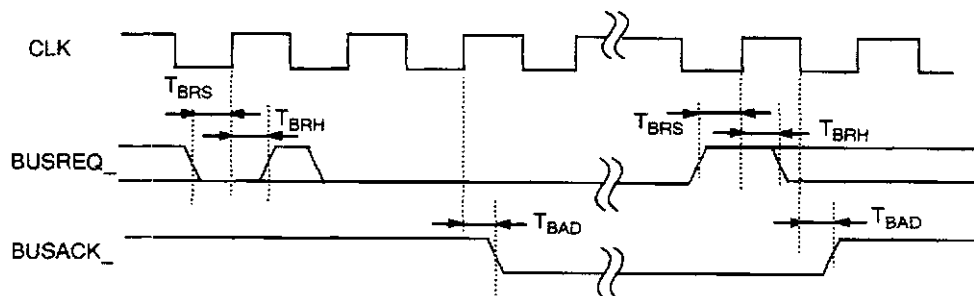
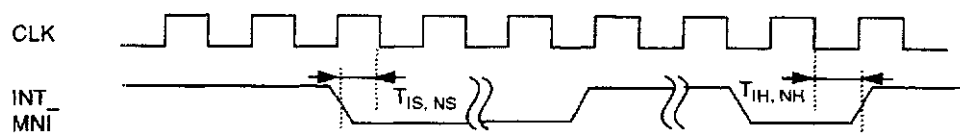
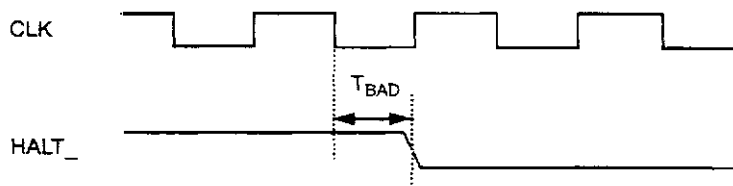


## Memory read/write cycle timing (KC80 mode)



## I/O read/write cycle timing (KC80 mode)



**Bus control timing****Interrupt input timing****Halt output timing**

KL5C8400 AC Characteristics (5V  $\pm$  10%)

[Z80 mode]

No.	Item	Min	Typ	Max	Unit
T <sub>CW</sub>	CLK cycle time	30.0			ns
T <sub>CLW</sub>	CLK "L" pulse width	12.0			ns
T <sub>CHW</sub>	CLK "H" pulse width	12.0			ns
T <sub>RW</sub>	RESET_ pulse width	3			clk
T <sub>AD</sub>	address delay (TCLW $\geq$ 25.0[ns])			26.0	ns
	address delay (TCLW < 25.0[ns])			51.0 - T <sub>CLW</sub>	ns
T <sub>ADR</sub>	refresh address delay	9.0		35.0	ns
T <sub>DD</sub>	CLK $\rightarrow$ data output delay			28.0	ns
T <sub>DZ</sub>	CLK $\rightarrow$ data output off delay	4.0			ns
T <sub>DS</sub>	data input set up time (vs. CLK)	0.0			ns
T <sub>DH</sub>	data input hold time (vs. RD_)	0.0			ns
T <sub>RD</sub>	CLK $\rightarrow$ RD_ delay	5.0		27.0	ns
T <sub>WD</sub>	CLK $\rightarrow$ WR_ delay			26.0	ns
T <sub>WS</sub>	WAIT_ input set up time	0.0			ns
T <sub>WH</sub>	WAIT_ input hold time	6.0			ns
T <sub>MD</sub>	CLK $\rightarrow$ MREQ_ delay			25.0	ns
T <sub>ID</sub>	CLK $\rightarrow$ IORQ_ delay	6.0		24.0	ns
T <sub>M1D</sub>	CLK $\rightarrow$ M1_ delay (TCLW $\geq$ 21.0[ns])			31.0	ns
	CLK $\rightarrow$ M1_ delay (TCLW < 21.0[ns])			52.0 - T <sub>CLW</sub>	ns
T <sub>HD</sub>	CLK $\rightarrow$ HALT_ delay			41.0	ns
T <sub>BRS</sub>	BUSREQ_ input set up time	3.0			ns
T <sub>BRH</sub>	BUSREQ_ input hold time	5.0			ns
T <sub>BAD</sub>	BUSACK_ output delay			23.0	ns
T <sub>IS</sub>	INT_ input set up time	3.0			ns
T <sub>ID</sub>	INT_ input hold time	10.0			ns
T <sub>NS</sub>	NMI_ input set up time	3.0			ns
T <sub>NH</sub>	NMI_ input hold time	5.0			ns

Note 1) Output load C<sub>L</sub> is 70pF.

2) "clk" in unit is numbers of the system clock.

KL5C8400 AC Characteristics (5V  $\pm$  10%)

[KC80 mode]

No.	Item	Min	Typ	Max	Unit
T <sub>KDS</sub>	data input set up time (vs. CLK)	0.0			ns
T <sub>KDH</sub>	data input hold time (vs. RD <sub>⌋</sub> )	0.0			ns
T <sub>MRD</sub>	CLK→MRD <sub>⌋</sub> delay	6.0		24.0	ns
T <sub>MWD</sub>	CLK→MWR <sub>⌋</sub> delay			24.0	ns
T <sub>KWS</sub>	WAIT <sub>⌋</sub> input set up time	0.0			ns
T <sub>KWH</sub>	WAIT <sub>⌋</sub> input hold time	5.0			ns
T <sub>IRD</sub>	CLK→IORD <sub>⌋</sub> delay			23.0	ns
T <sub>IWD</sub>	CLK→IOWR <sub>⌋</sub> delay	6.0		24.0	ns
T <sub>KM1D</sub>	CLK→M1 <sub>⌋</sub> delay			25.0	ns

Note 1) Other AC characteristics are the same as ones in Z80 mode.

2) Output load C<sub>L</sub> is 70pF.

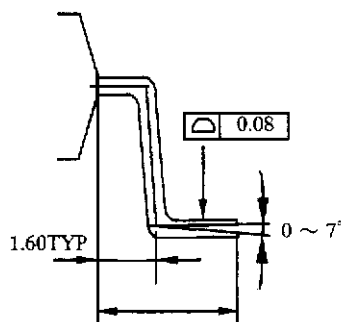
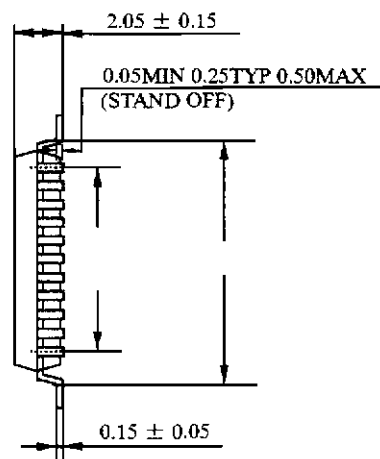
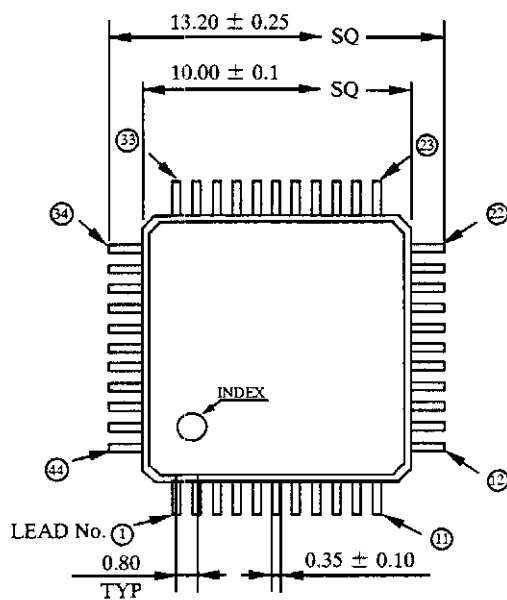


## 9. Physical Dimension

The package of KL5C8400 is a plastic QFP44.

The following is the physical dimensions of QFP44.

"KL5C8400C" is printed on its package.



Note: All dimensions are in "mm".