Unit 4

Interfacing
IO and memory devices

==========

Timing Diagrams of 8085

It is one of the best way to understand to process of micro-processor/controller. With the help of timing diagram we can understand the working of any system, step by step working of each instruction and its execution, etc.

It is the graphical representation of process in steps with respect to time. The timing diagram represents the clock cycle and duration, delay, content of address bus and data bus, type of operation ie. Read/write/status signals.

Important terms related to timing diagrams:

1. **Instruction cycle:** this term is defined as the number of steps required by the cpu to complete the entire process ie. Fetching and execution of one instruction. The fetch and execute cycles are carried out in synchronization with the clock.
2. **Machine cycle:** It is the time required by the microprocessor to complete the operation of accessing the memory devices or I/O devices. In machine cycle various operations like opcode fetch, memory read, memory write, I/O read, I/O write are performed.
3. **T-state:** Each clock cycle is called as T-states.

Rules to identify number of machine cycles in an instruction:

1. If an addressing mode is direct, immediate or implicit

   Then

   No. of machine cycles = No. of bytes.

2. If the addressing mode is indirect

   Then,

   No. of machine cycles = No. of bytes + 1

   Add +1 to the No. of machine cycles if it is memory read/write operation.

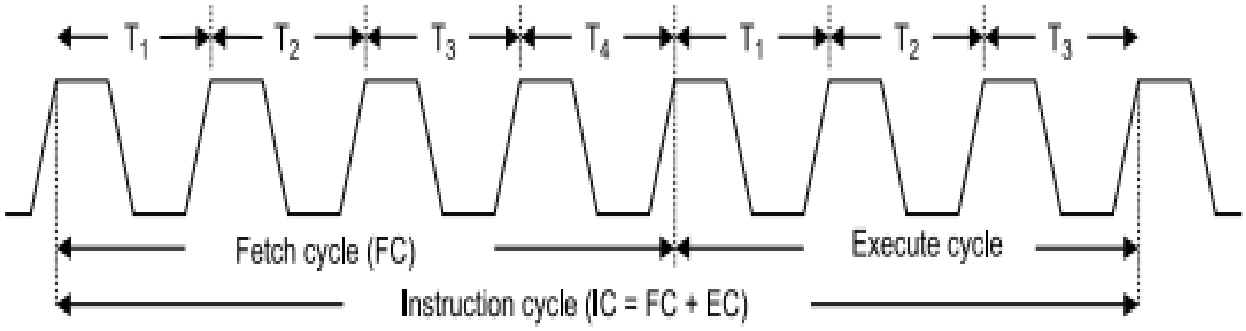
3. If the operand is 8-bit or 16-bit address

then,

No. of machine cycles = No. of bytes +1

4. These rules are applicable to 80% of the instructions of 8085.

Timing Diagram:



Where,

Instruction cycle = Fetch Cycle(FC) + Execute cycle(EC)

Following table show the status of different control signal for different operation. We should remember that to complete our timing diagram of 8085 microprocessors.

Machine cycle	*Status*			*Controls*		
	IO / $\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read (I/OR)	1	1	0	0	1	1
I/O Write (I/OW)	1	0	1	1	0	1

Opcode fetch:

The microprocessor requires instructions to perform any particular action. In order to perform these actions microprocessor utilizes Opcode which is a part of an instruction which provides detail (i.e. Which operation µp needs to perform) to microprocessor.

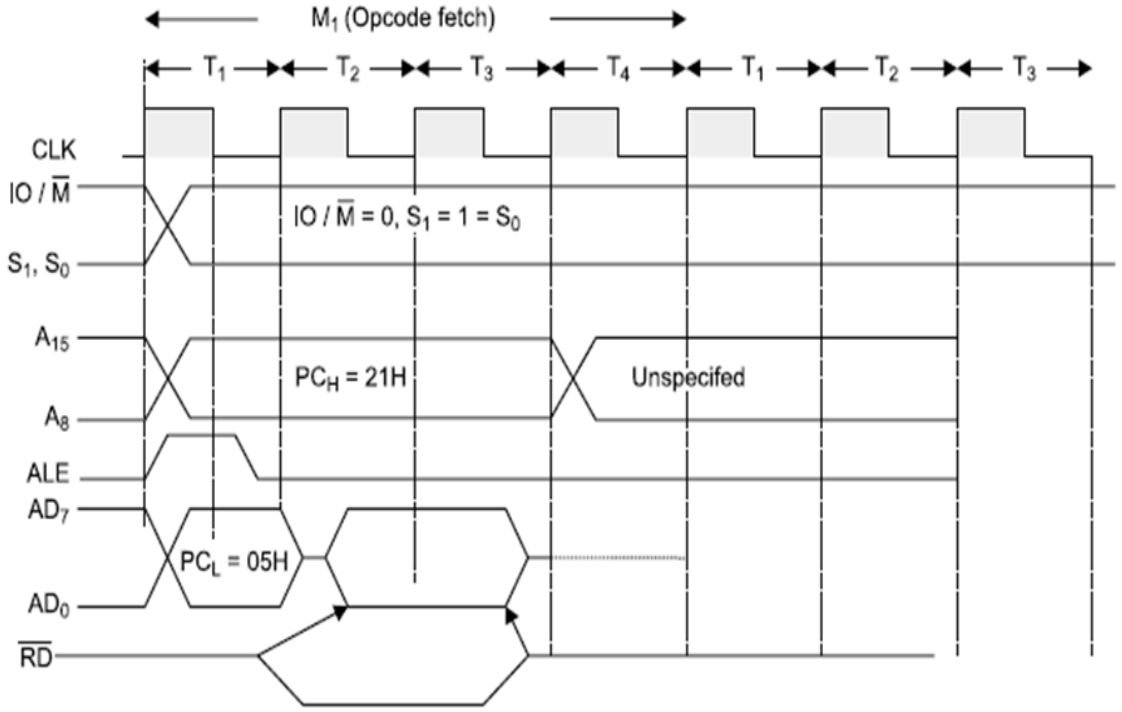


Fig: Opcode fetch timing diagram

Operation:

- During T1 state, microprocessor uses IO/M(bar), S0, S1 signals are used to instruct microprocessor to fetch opcode.

- Thus when IO/M(bar)=0, S0=S1= 1, it indicates opcode fetch operation.

- During this operation 8085 transmits 16-bit address and also uses ALE signal for address latching.

- At T2 state microprocessor uses read signal and make data ready from that memory location to read opcode from memory and at the same time program counter increments by 1 and points next instruction to be fetched.

- In this state microprocessor also checks READY input signal, if this pin is at low logic level ie. '0' then microprocessor adds wait state immediately between T2 and T3.

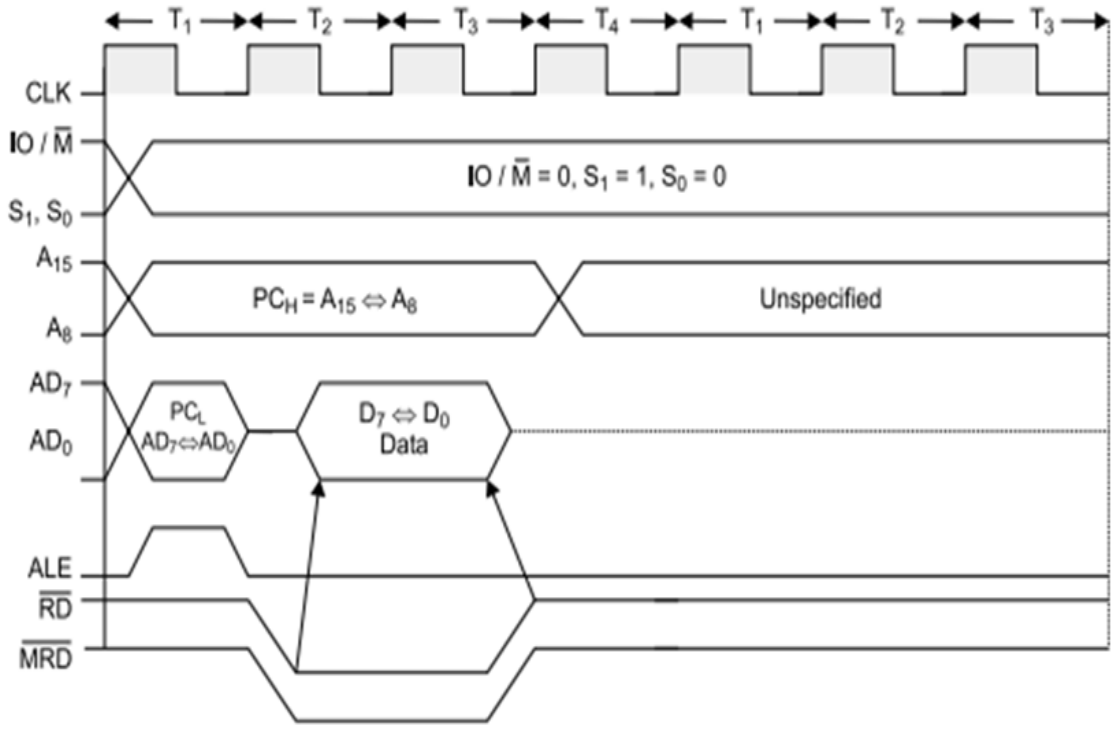
- At T3, microprocessor reads opcode and store it into instruction register to decode it further.

- During T4 microprocessor performs internal operation like decoding opcode and providing necessary actions.

- The opcode is decoded to know whether T5 or T6 states are required, if they are not required then µp performs next operation.

Read and write timing diagram for memory and I/O Operation

Memory Read:



<>Figure: Memory read timing diagram

Operation:

- It is used to fetch one byte from the memory.

- It requires 3 T-States.
- It can be used to fetch operand or data from the memory.
- During T1, A8-A15 contains higher byte of address. At the same time ALE is high. Therefore Lower byte of address A0-A7 is selected from AD0-AD7.
- Since it is memory ready operation, IO/M(bar) goes low.
- During T2 ALE goes low, RD(bar) goes low. Address is removed from AD0-AD7 and data D0-D7 appears on AD0-AD7.
- During T3, Data remains on AD0-AD7 till RD(bar) is at low signal.

Memory Write:

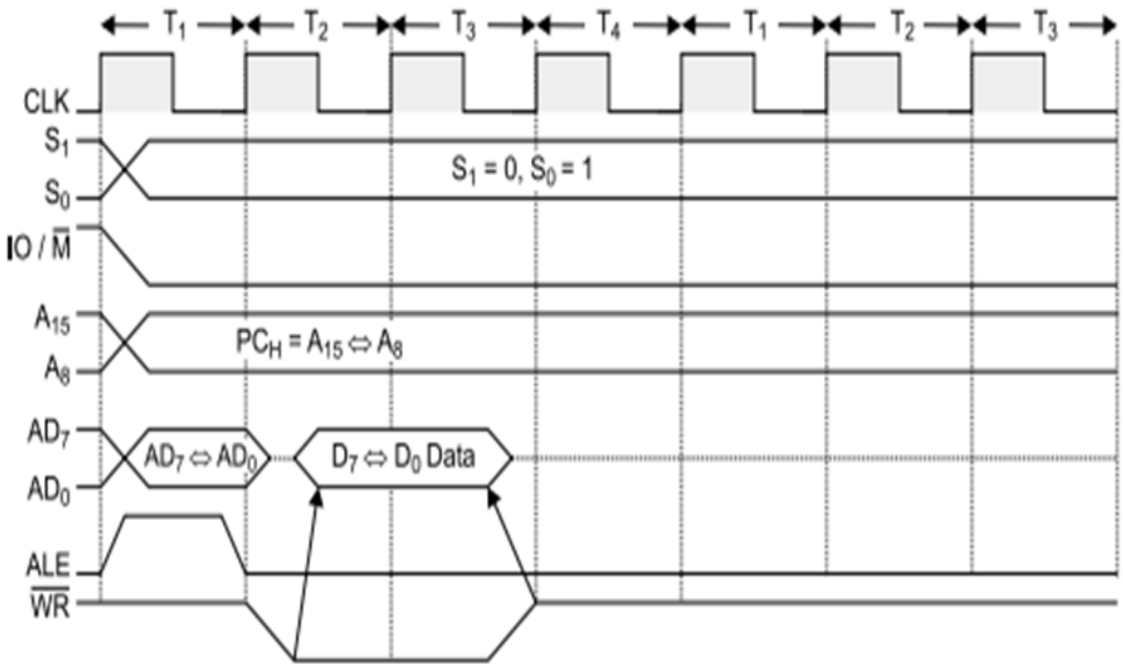


Figure: Memory write timing diagram

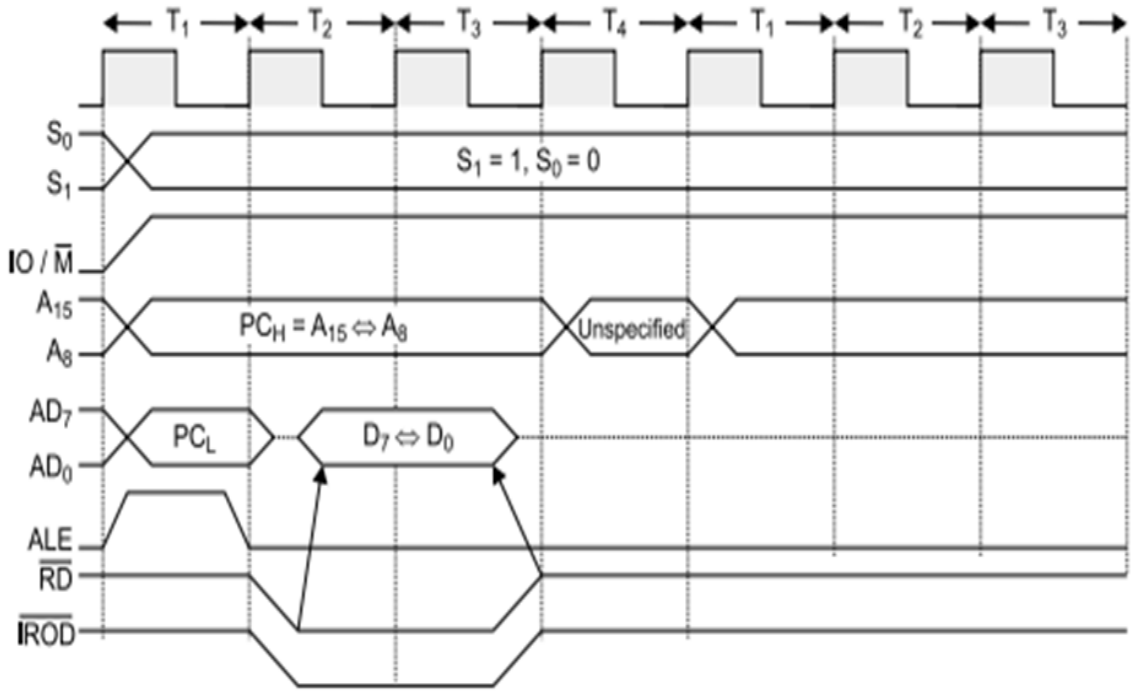
Operation:

- It is used to send one byte into memory.
- It requires 3 T-States.
- During T1, ALE is high and contains lower address A0-A7 from AD0-AD7.
- A8-A15 contains higher byte of address.

- As it is memory operation, IO/M(bar) goes low.

- During T2, ALE goes low, WR(bar) goes low and Address is removed from AD0-AD7 and then data appears on AD0-AD7.

- Data remains on AD0-AD7 till WR(bar) is low.



IO Read:

Figure: I/O read timing diagram

Operation:

It is used to fetch one byte from an IO port.

It requires 3 T-States.

During T1, The Lower Byte of IO address is duplicated into higher order address bus A8-A15.

ALE is high and AD0-AD7 contains address of IO device.

IO/M (bar) goes high as it is an IO operation.

During T2, ALE goes low, RD (bar) goes low and data appears on AD0-AD7 as input from IO device.

During T3 Data remains on AD0-AD7 till RD(bar) is low.

IO Write:

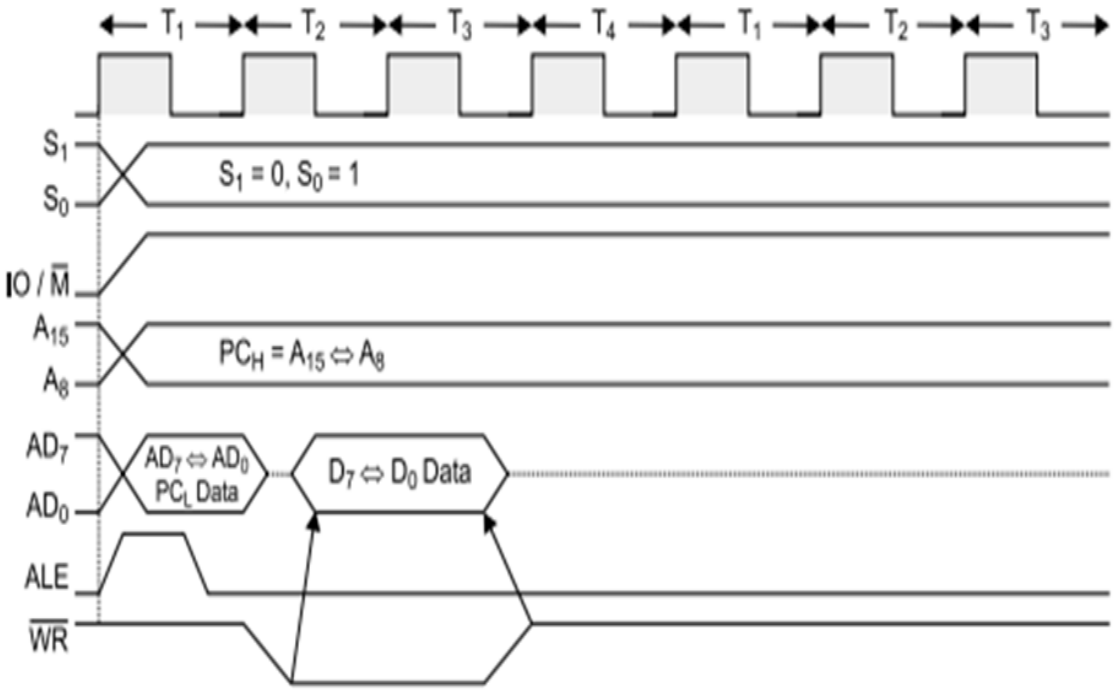


Figure:I/O write timing diagram

Operation:

- It is used to writ one byte into IO device.
- It requires 3 T-States.
- During T1, the lower byte of address is duplicated into higher order address bus A8-A15.
- ALE is high and A0-A7 address is selected from AD0-AD7.
- As it is an IO operation IO/M (bar) goes low.
- During T2, ALE goes low, WR (bar) goes low and data appears on AD0-AD7 to write data into IO device.
- During T3, Data remains on AD0-AD7 till WR(bar) is low.

Timing Diagram of 8085 MP

Instruction		Opcode	Operand
1 byte	MOV A,B	78H	-
2 byte	MVI A,05H	3E	05H
3 byte	LDA 3000H	3A	00h and 30H

T-state:

This is defined as one sub division of the operation performed in 1 clock period.

Instruction cycle:

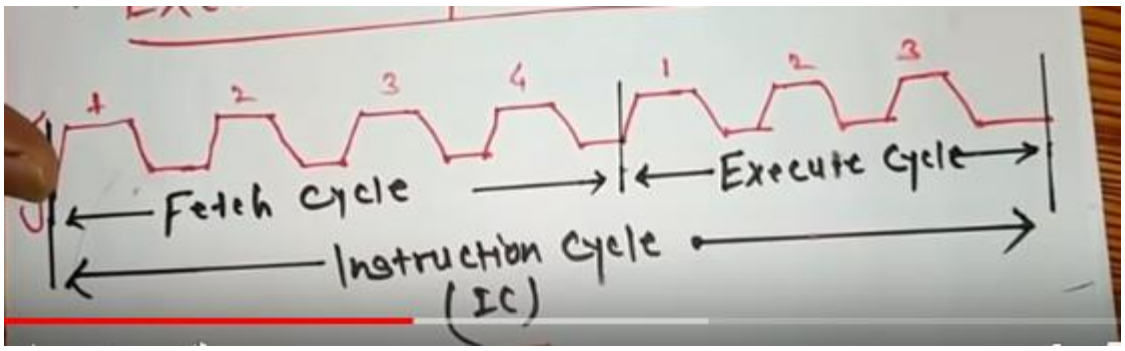
This is defined as the total time required in executing an instruction completely.

IC= FC + EC

FC-4T

MR-3T, MW-3T

Execution Operation:



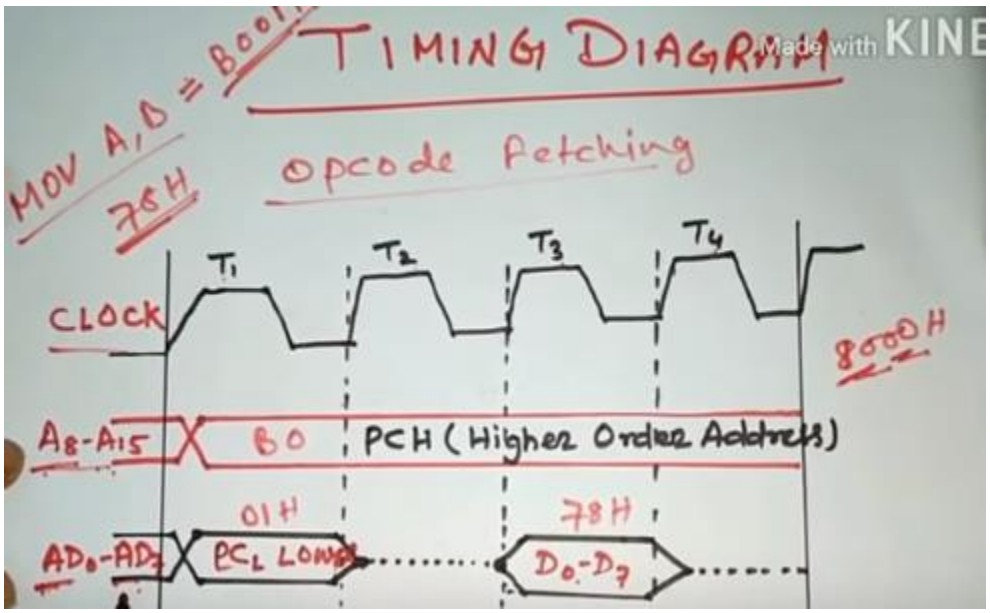
Timing diagram for Opcode fetching (4T)

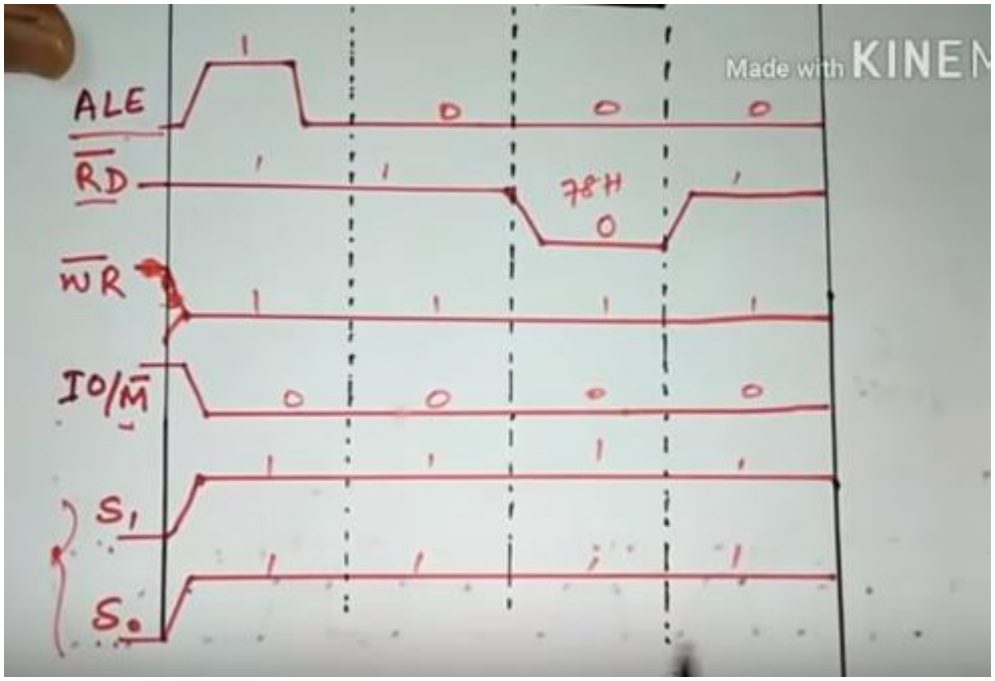
MOV A, B = B001H (Address Location) and 78H (Opcode)

(T1= RLC network delay, component voltage initiate, voltage drop)

AD0-AD7: …….. means high impendence , that data cannot get losted.

ALE: 1= addr, 0 =data





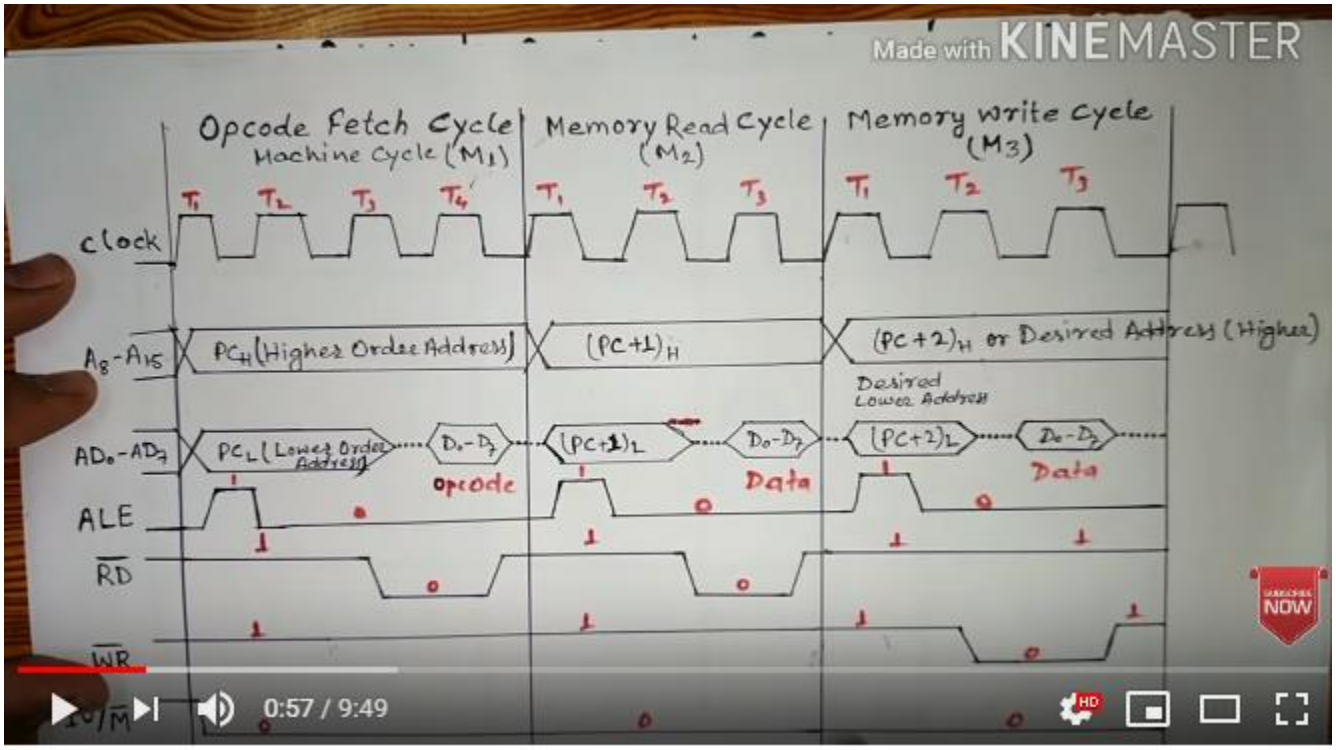
S0	S1	
1	1	Opcode Fetch
0	1	For Read MEM
1	0	For Write MEM

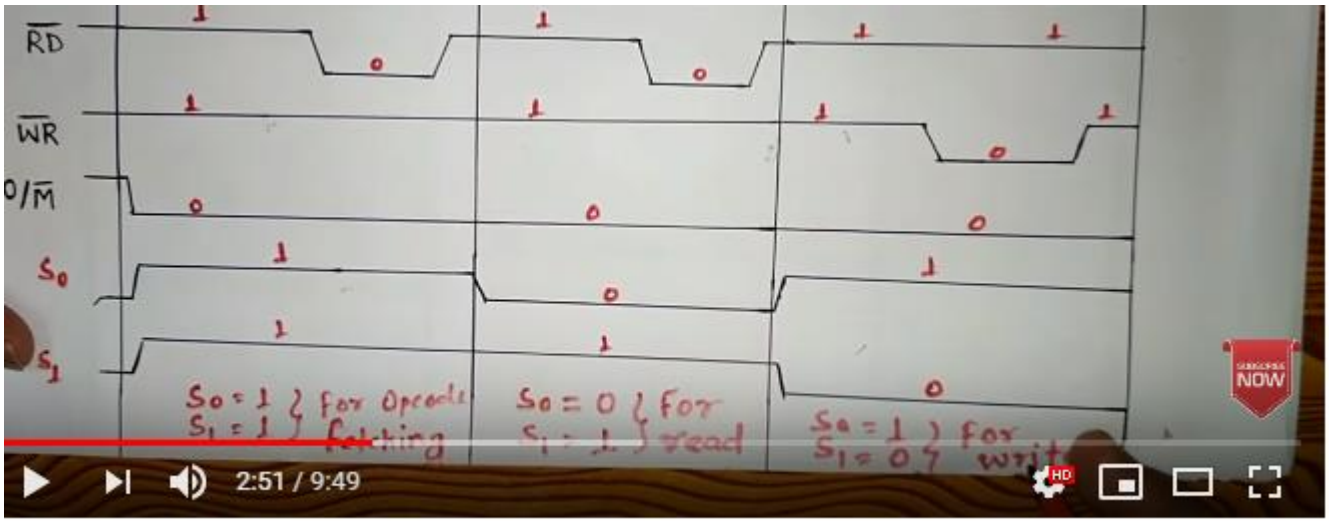
Timing diagram for Read and write operation

Read operation

Write operation

Read and write operation





Example

Timing diagram of

STA 8000H

Opcode – 32H 1 Byte

Operand --00H 1Byte

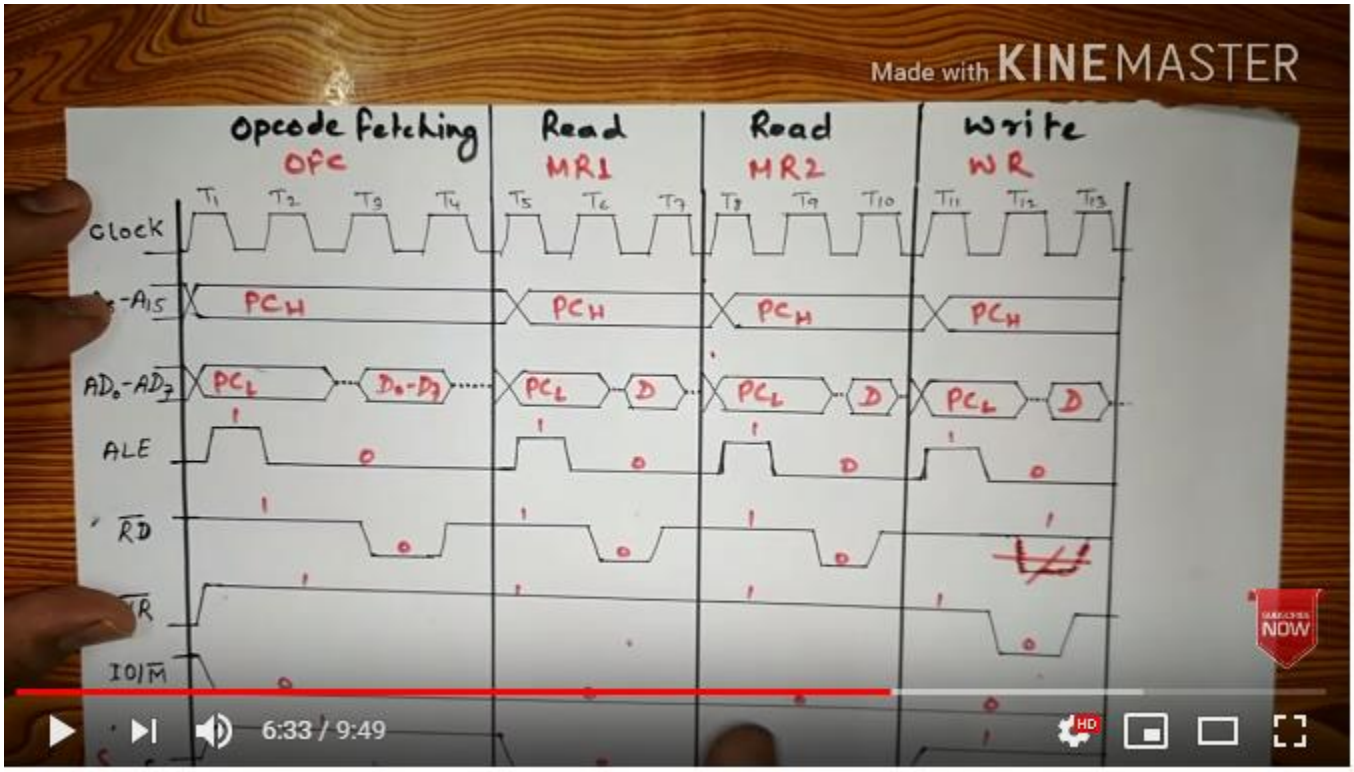
80H 1 Byte

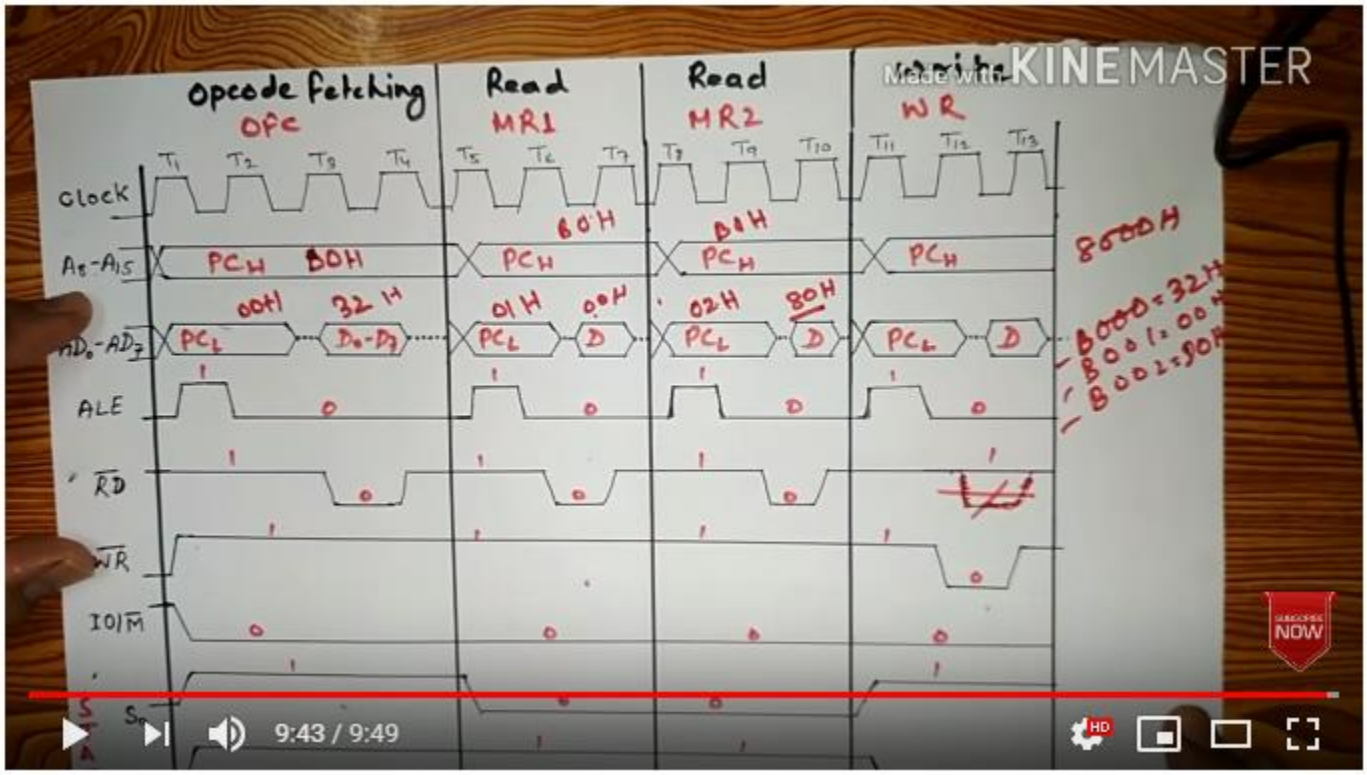
Opcode – 4 T state

Operand - Read = 3+ 3 = 6 T state (read location 32 H and 00h)

Write= 3 T state

Total T-state= 4+6+ 3= 13 T sates




Opcode fetching
OFC
Read
MR1
Read
MR2
WR
Clock
ALE
RD
WR
IO/M
9:43 / 9:49

Address Decoding

Introduction:

- Consider that we have a microprocessor interfaced to both I/O device and also a memory chip. Now how to select between the two devices according to the requirement?
- For this purpose, an address decoding circuit is used. An address decoding circuit aids in selecting the required I/O device or a memory chip.
- Let us discuss the concept of Memory interfacing and I/O interfacing here.
- The processor communicates with all the parts interconnected in the system through a common address and data bus. As a result of this, only one device can transmit data at a time and others can only receive that data. If more than one device attempt to send data through the bus at the same time, the proper communication among the devices does not become possible because the data sent by them gets garbled. To avoid this situation, ensuring that the proper device gets addressed at proper time, the technique called "address decoding" is used. Generally, there are two common methods for mapping address of these devices.

They are:

- I/O Mapped I/O :

  It can address 2^8=256 bytes if mapped in I/O mode. They are used to read 8-bit data from or write 8-bit data to selected device. In this, the I/O device is addressed with 8-bit address.

- Memory Mapped I/O :

  In this mode, the I/O devices are addressed with 16-bit address. The total addressing capability of the processor 8085 in this mode is 2^16 = 65536 bytes = 64 KB.

  In both modes described above, depending on the addresses that are allocated to the device, the address decoding are categorized in the following two groups:

- Unique Address Decoding:

  If all of the address lines available on that mapping mode are used for address decoding, then that decoding is called unique address decoding.

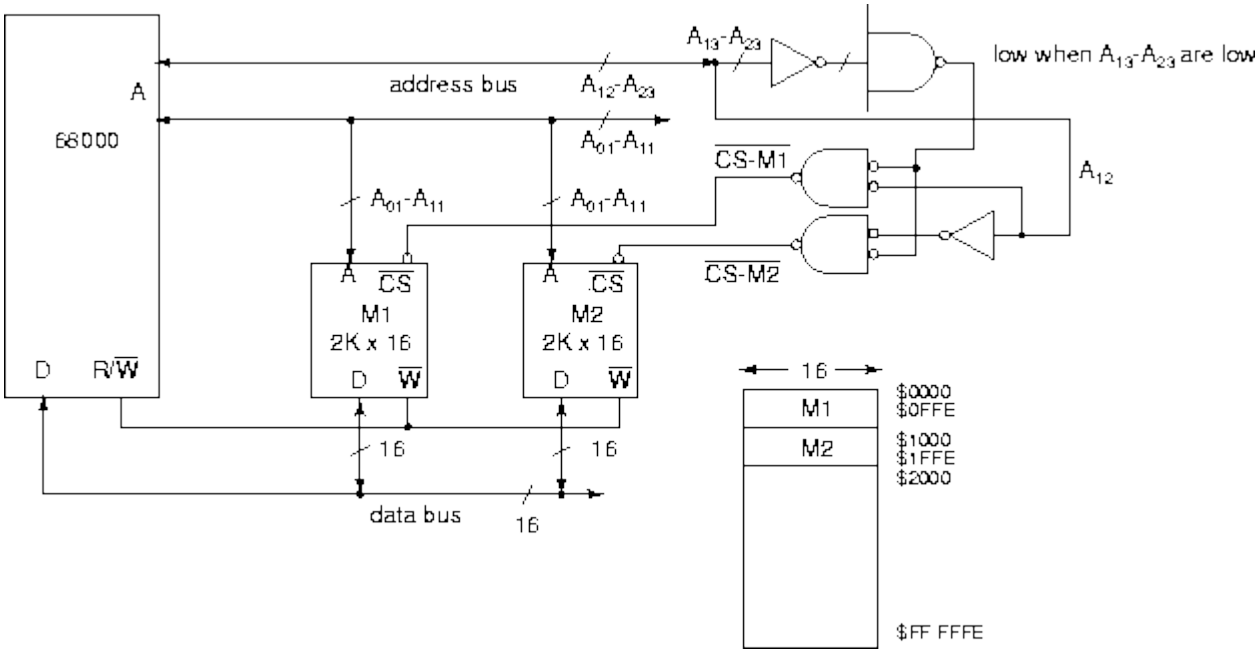
- Non-unique Address Decoding:

If all of the address lines available on that mapping mode are not used for address decoding, then that decoding is called unique address decoding.

Address decoding refers to the way a computer system decodes the addresses on the address bus to select memory locations in one or more memory or peripheral devices. The 68000's 23-bit address bus permits 2^{23} 16-bit words to be uniquely addressed.

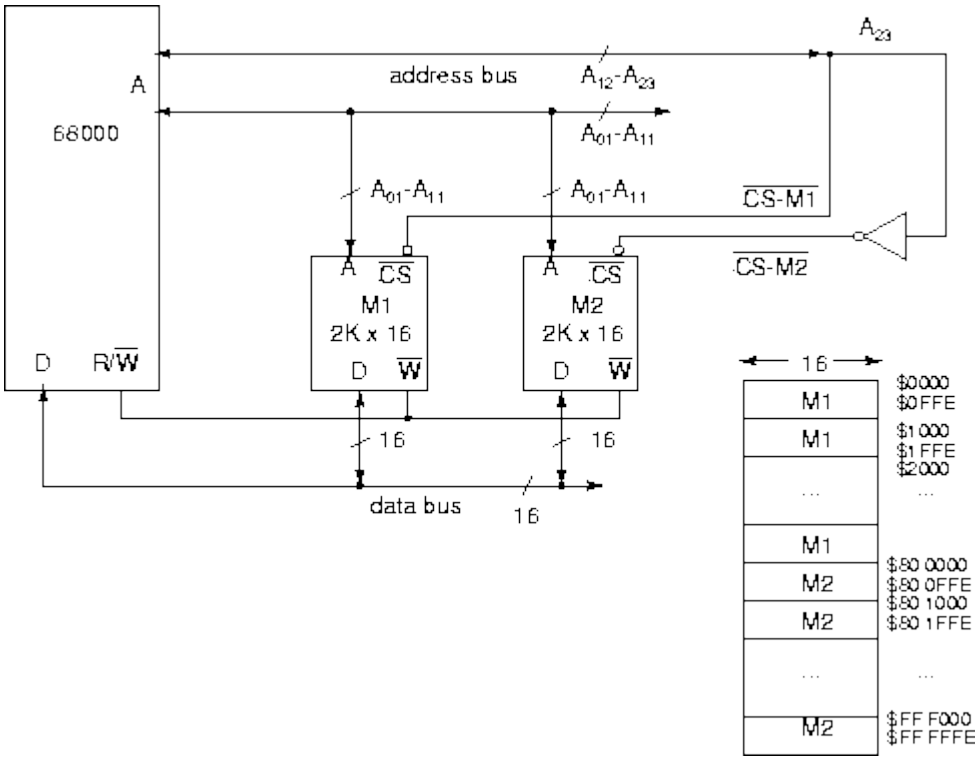
In ***full address decoding***, each addressable memory location corresponds to a unique address value on the address bus. Figure 126 shows an example of two memory devices configured using full address decoding. Memory M1 is selected whenever A_{12}-A_{23}=000000000000, while M2 is selected whenever A_{12}- A_{23}=100000000000.

Figure: Full address decoding of two memory devices.



In ***partial address decoding***, not all address lines in the address bus are used in the decoding process. Figure 127 shows two memory devices configured using partial decoding, where A_{23} is used to distinguish between the two. In this example, M1 and M2 are repeated 2,048 times through the memory space. When A_{23}=0, M1 is selcted; when A_{23}=1, M2 is selected.

Figure Partial address decoding of two memory devices.



Microprocessor - I/O Interfacing Overview

In this chapter, we will discuss Memory Interfacing and IO Interfacing with 8085.

Interface is the path for communication between two components. Interfacing is of two types, memory interfacing and I/O interfacing.

Memory Interfacing

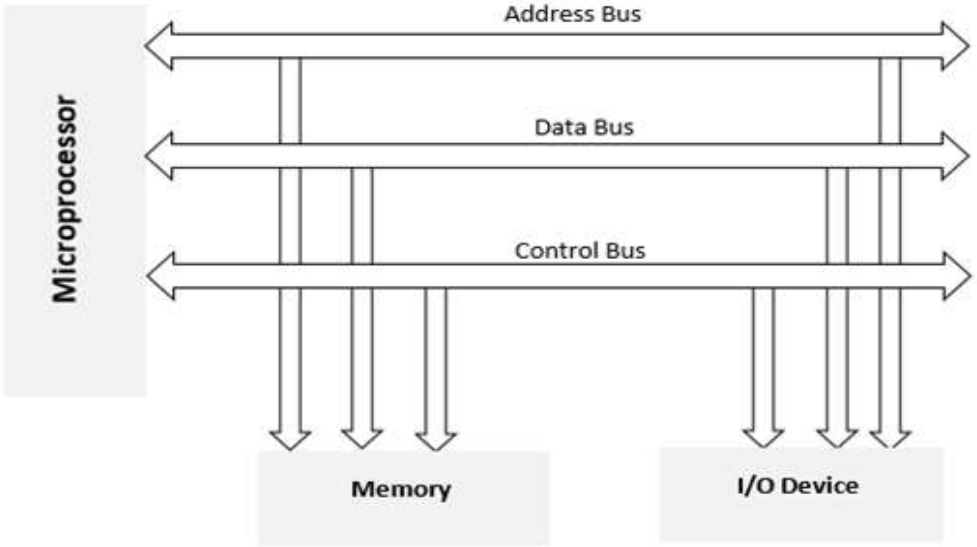
When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory. For this, both the memory and the microprocessor requires some signals to read from and write to registers.

The interfacing process includes some key factors to match with the memory requirements and microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

IO Interfacing

There are various communication devices like the keyboard, mouse, printer, etc. So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers. This type of interfacing is known as I/O interfacing.

Block Diagram of Memory and I/O Interfacing



8085 Interfacing Pins

Following is the list of 8085 pins used for interfacing with other devices

- A_{15} - A_8 (Higher Address Bus)
- AD_7 - AD_0(Lower Address/Data Bus)
- ALE
- RD
- WR
- READY

Ways of Communication – Microprocessor with the Outside World?

There are two ways of communication in which the microprocessor can connect with the outside world.

- Serial Communication Interface
- Parallel Communication interface

Serial Communication Interface – In this type of communication, the interface gets a single byte of data from the microprocessor and sends it bit by bit to the other system serially and vice-a-versa.

Parallel Communication Interface – In this type of communication, the interface gets a byte of data from the microprocessor and sends it bit by bit to the other systems in simultaneous (or) parallel fashion and vice-a-versa.

8279 – Programmable Keyboard

8279 programmable keyboard/display controller is designed by Intel that interfaces a keyboard with the CPU. The keyboard first scans the keyboard and identifies if any key has been pressed. It then sends their relative response of the pressed key to the CPU and vice-a-versa.

How Many Ways the Keyboard is Interfaced with the CPU?

The Keyboard can be interfaced either in the interrupt or the polled mode. In the **Interrupt mode**, the processor is requested service only if any key is pressed, otherwise the CPU will continue with its main task.

In the **Polled mode**, the CPU periodically reads an internal flag of 8279 to check whether any key is pressed or not with key pressure.

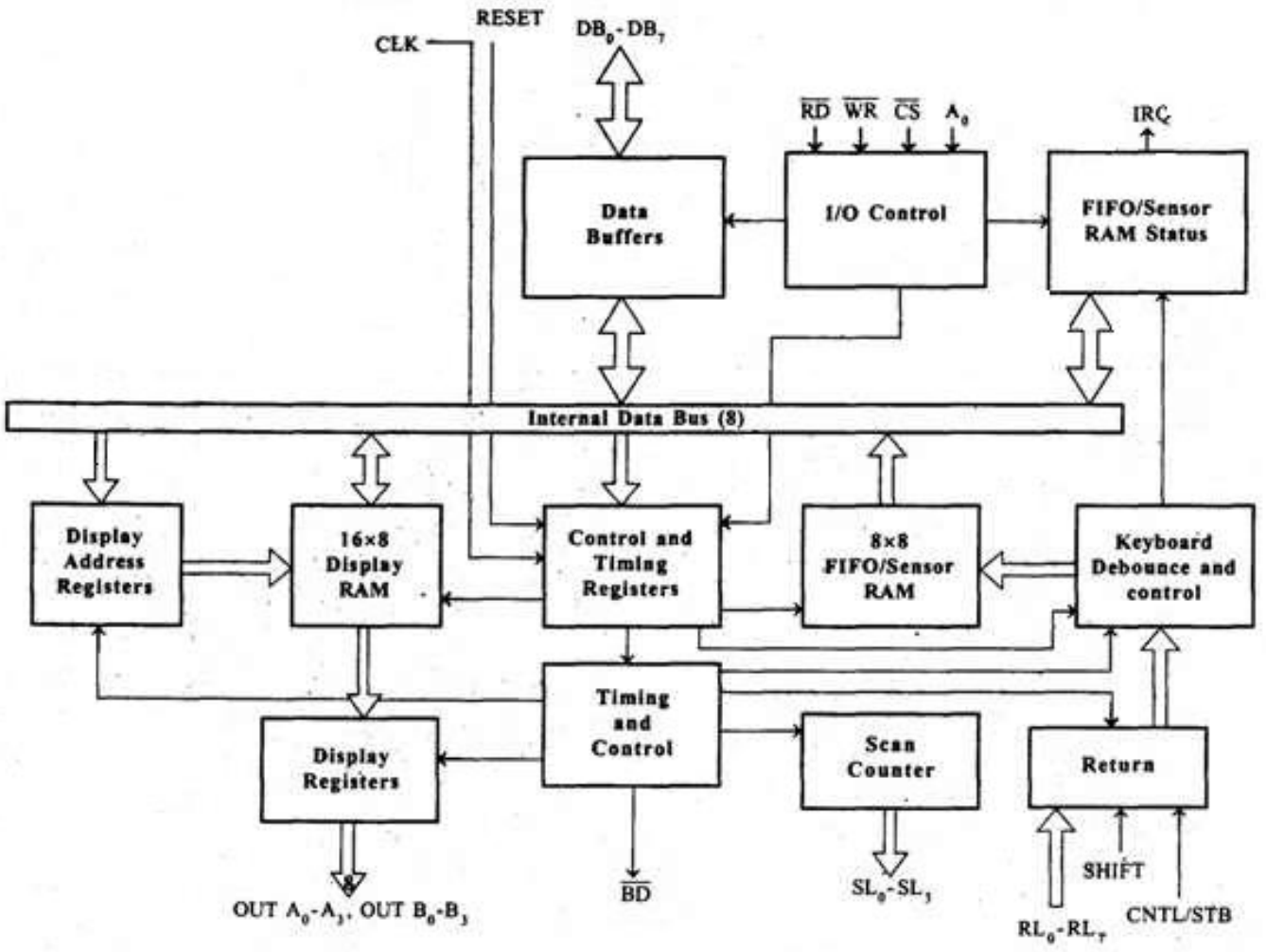
How Does 8279 Keyboard Work?

The keyboard consists of maximum 64 keys, which are interfaced with the CPU by using the key-codes. These key-codes are de-bounced and stored in an 8-byte FIFORAM, which can be accessed by the CPU. If more than 8 characters are entered in the FIFO, then it means more than eight keys are pressed at a time. This is when the overrun status is set.

If a FIFO contains a valid key entry, then the CPU is interrupted in an interrupt mode else the CPU checks the status in polling to read the entry. Once the CPU reads a key entry, then FIFO is updated, and

the key entry is pushed out of the FIFO to generate space for new entries.

Architecture and Description



I/O Control and Data Buffer

This unit controls the flow of data through the microprocessor. It is enabled only when D is low. Its data buffer interfaces the external bus of the system with the internal bus of the microprocessor. The pins A0, RD, and WR are used for command, status or data read/write operations.

Control and Timing Register and Timing Control

This unit contains registers to store the keyboard, display modes, and other operations as programmed by the CPU. The timing and control unit handles the timings for the operation of the circuit.

Scan Counter

It has two modes i.e. **Encoded mode** and Decoded mode. In the encoded mode, the counter provides the binary count that is to be externally decoded to provide the scan lines for the keyboard and display.

In the **decoded scan mode,** the counter internally decodes the least significant 2 bits and provides a decoded 1 out of 4 scan on SL_0-SL_3.

Return Buffers, Keyboard Debounce, and Control

This unit first scans the key closure row-wise, if found then the keyboard debounce unit debounces the key entry. In case, the same key is detected, then the code of that key is directly transferred to the sensor RAM along with SHIFT & CONTROL key status.

FIFO/Sensor RAM and Status Logic

This unit acts as 8-byte first-in-first-out (FIFO) RAM where the key code of every pressed key is entered into the RAM as per their sequence. The status logic generates an interrupt request after each FIFO read operation till the FIFO gets empty.

In the scanned sensor matrix mode, this unit acts as sensor RAM where its each row is loaded with the status of their corresponding row of sensors into the matrix. When the sensor changes its state, the IRQ line changes to high and interrupts the CPU.

Display Address Registers and Display RAM

This unit consists of display address registers which holds the addresses of the word currently read/written by the CPU to/from the display RAM.

8279 – Pin Description

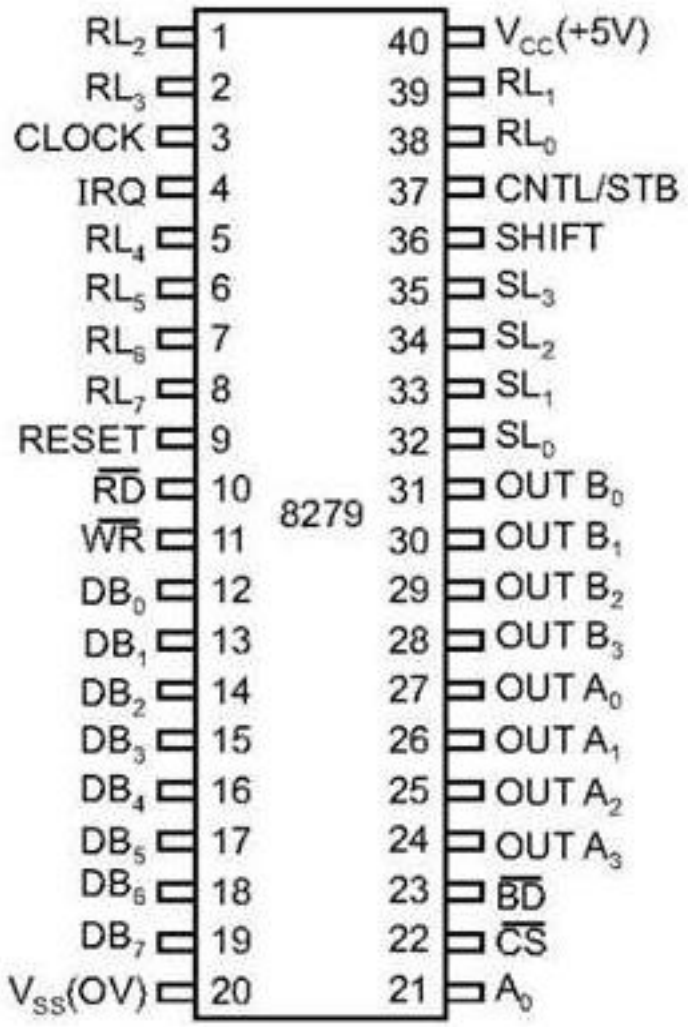
The following figure shows the pin diagram of 8279 –

Data Bus Lines, DB_0 - DB_7

These are 8 bidirectional data bus lines used to transfer the data to/from the CPU.

CLK

The clock input is used to generate internal timings required by the microprocessor.



RESET

As the name suggests this pin is used to reset the microprocessor.

CS Chip Select

When this pin is set to low, it allows read/write operations, else this pin should be set to high.

A_0

This pin indicates the transfer of command/status information. When it is low, it indicates the transfer of data.

RD, WR

This Read/Write pin enables the data buffer to send/receive data over the data bus.

IRQ

This interrupt output line goes high when there is data in the FIFO sensor RAM. The interrupt line goes low with each FIFO RAM read operation. However, if the FIFO RAM further contains any key-code entry to be read by the CPU, this pin again goes high to generate an interrupt to the CPU.

V_{ss}, V_{cc}

These are the ground and power supply lines of the microprocessor.

SL_0 – SL_3

These are the scan lines used to scan the keyboard matrix and display the digits. These lines can be programmed as encoded or decoded, using the mode control register.

RL_0 – RL_7

These are the Return Lines which are connected to one terminal of keys, while the other terminal of the keys is connected to the decoded scan lines. These lines are set to 0 when any key is pressed.

SHIFT

The Shift input line status is stored along with every key code in FIFO in the scanned keyboard mode. Till it is pulled low with a key closure, it is pulled up internally to keep it high

CNTL/STB - CONTROL/STROBED I/P Mode

In the keyboard mode, this line is used as a control input and stored in FIFO on a key closure. The line is a strobe line that enters the data into FIFO RAM, in the strobed input mode. It has an internal pull up. The line is pulled down with a key closure.

BD

It stands for blank display. It is used to blank the display during digit switching.

$OUTA_0$ – $OUTA_3$ and $OUTB_0$ – $OUTB_3$

These are the output ports for two 16x4 or one 16x8 internal display refresh registers. The data from these lines is synchronized with the scan lines to scan the display and the keyboard.

Operational Modes of 8279

There are two modes of operation on 8279 – **Input Mode** and **Output Mode.**

Input Mode

This mode deals with the input given by the keyboard and this mode is further classified into 3 modes.

- **Scanned Keyboard Mode** – In this mode, the key matrix can be interfaced using either encoded or decoded scans. In the encoded scan, an 8×8 keyboard or in the decoded scan, a 4×8 keyboard can be interfaced. The code of key pressed with SHIFT and CONTROL status is stored into the FIFO RAM.
- **Scanned Sensor Matrix** – In this mode, a sensor array can be interfaced with the processor using either encoder or decoder scans. In the encoder scan, 8×8 sensor matrix or with decoder scan 4×8 sensor matrix can be interfaced.
- **Strobed Input** – In this mode, when the control line is set to 0, the data on the return lines is stored in the FIFO byte by byte.

Output Mode

This mode deals with display-related operations. This mode is further classified into two output modes.

- **Display Scan** – This mode allows 8/16 character multiplexed displays to be organized as dual 4-bit/single 8-bit display units.
- **Display Entry** – This mode allows the data to be entered for display either from the right side/left side.

Microprocessor - 8257 DMA Controller

DMA stands for Direct Memory Access. It is designed by Intel to transfer data at the fastest rate. It allows the device to transfer the data directly to/from memory without any interference of the CPU.

Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.

How DMA Operations Are Performed?

Following is the sequence of operations performed by a DMA –

- Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.
- The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.
- Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.
- Now the CPU is in HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

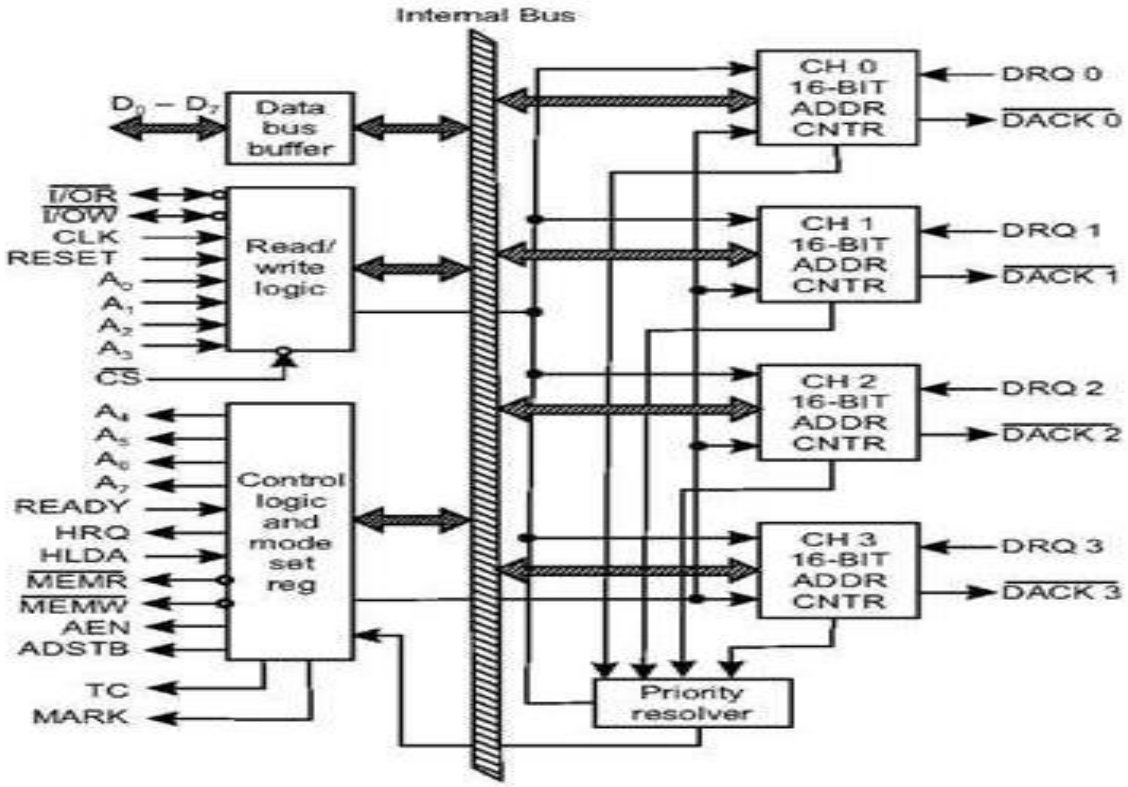
Features of 8257

Here is a list of some of the prominent features of 8257 –

- It has four channels which can be used over four I/O devices.
- Each channel has 16-bit address and 14-bit counter.
- Each channel can transfer data up to 64kb.
- Each channel can be programmed independently.
- Each channel can perform read transfer, write transfer and verify transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- It requires a single phase clock.
- Its frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., **Master mode** and **Slave mode.**

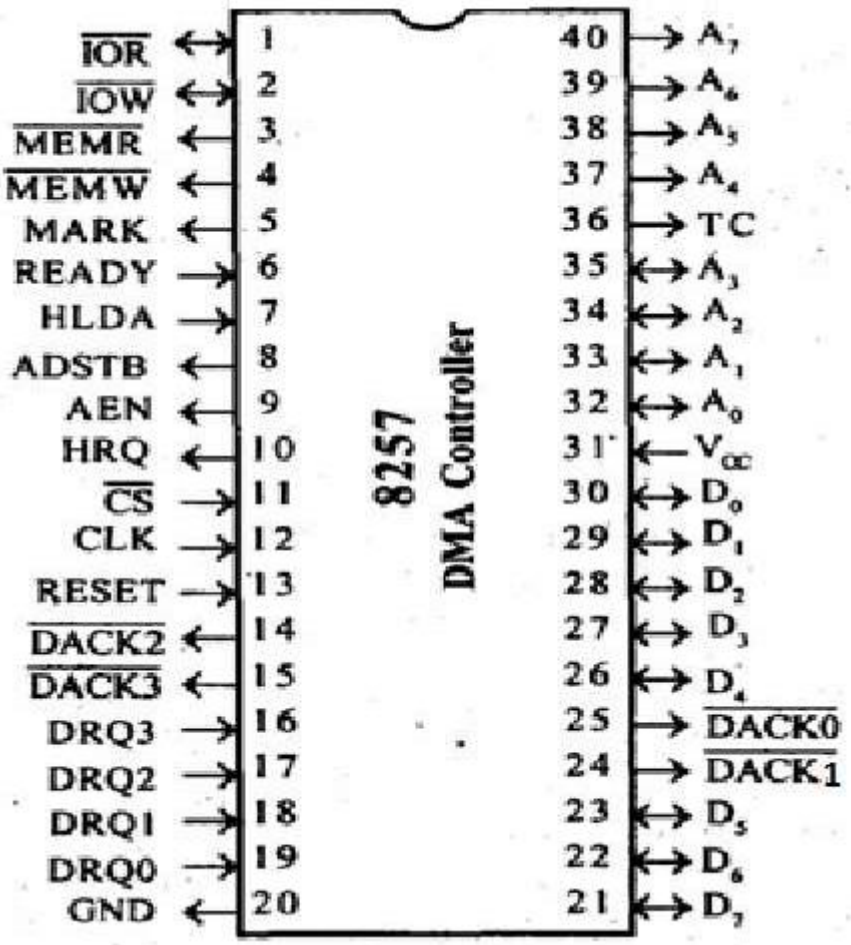
8257 Architecture

The following image shows the architecture of 8257 –



8257 Pin Description

The following image shows the pin diagram of a 8257 DMA controller –



DRQ_0–DRQ_3

These are the four individual channel DMA request inputs, which are used by the peripheral devices for using DMA services. When the fixed priority mode is selected, then DRQ_0 has the highest priority and DRQ_3 has the lowest priority among them.

$DACK_0$ – $DACK_3$

These are the active-low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.

D_0 – D_7

These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

IOR

It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

IOW

It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

CLK

It is a clock frequency signal which is required for the internal operation of 8257.

RESET

This signal is used to RESET the DMA controller by disabling all the DMA channels.

A_o - A_3

These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.

CS

It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

A_4 - A_7

These are the higher nibble of the lower byte address generated by DMA in the master mode.

READY

It is an active-high asynchronous input signal, which makes DMA ready by inserting wait states.

HRQ

This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

HLDA

It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

MEMR

It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

MEMW

It is the active-low three state signal which is used to write the data to the addressed memory location during DMA write operation.

ADST

This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

AEN

This signal is used to disable the address bus/data bus.

TC

It stands for 'Terminal Count', which indicates the present DMA cycle to the present peripheral devices.

MARK

The mark will be activated after each 128 cycles or integral multiples of it from the beginning. It indicates the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

V_{cc}

It is the power signal which is required for the operation of the circuit.