Intel 8085 Microprocessor Architecture

Dr. Vibhu Jately

Assistant Professor-SG

School of Computer Science

UPES

UNIVERSITY OF TOMORROW

Introduction of 8085

- Microprocessor is **CPU on a Chip**.
- Introduced in 1976 and is a 8 bit processor.
- Intel 8085 is an NMOS microprocessor.
- It is a 40 pin IC package fabricated on a single LSI chip.
- It uses +5 V for its operation.
- Its clock speed is about 3 MHz and clock cycle is 320 ns.
- It has 80 basic instructions and 246 opcodes.
- Intel has produced large number of peripheral devices for microprocessor based system.

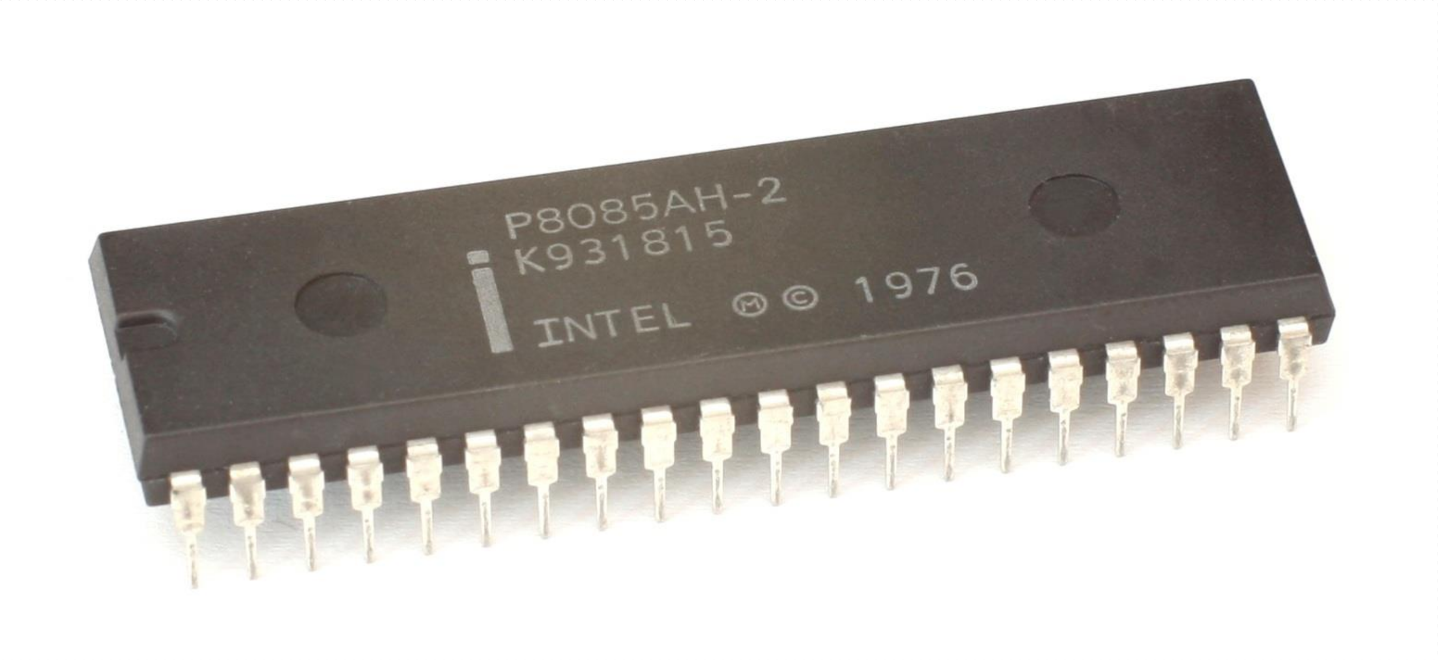


Figure 1. Intel 8085 40 pin chip

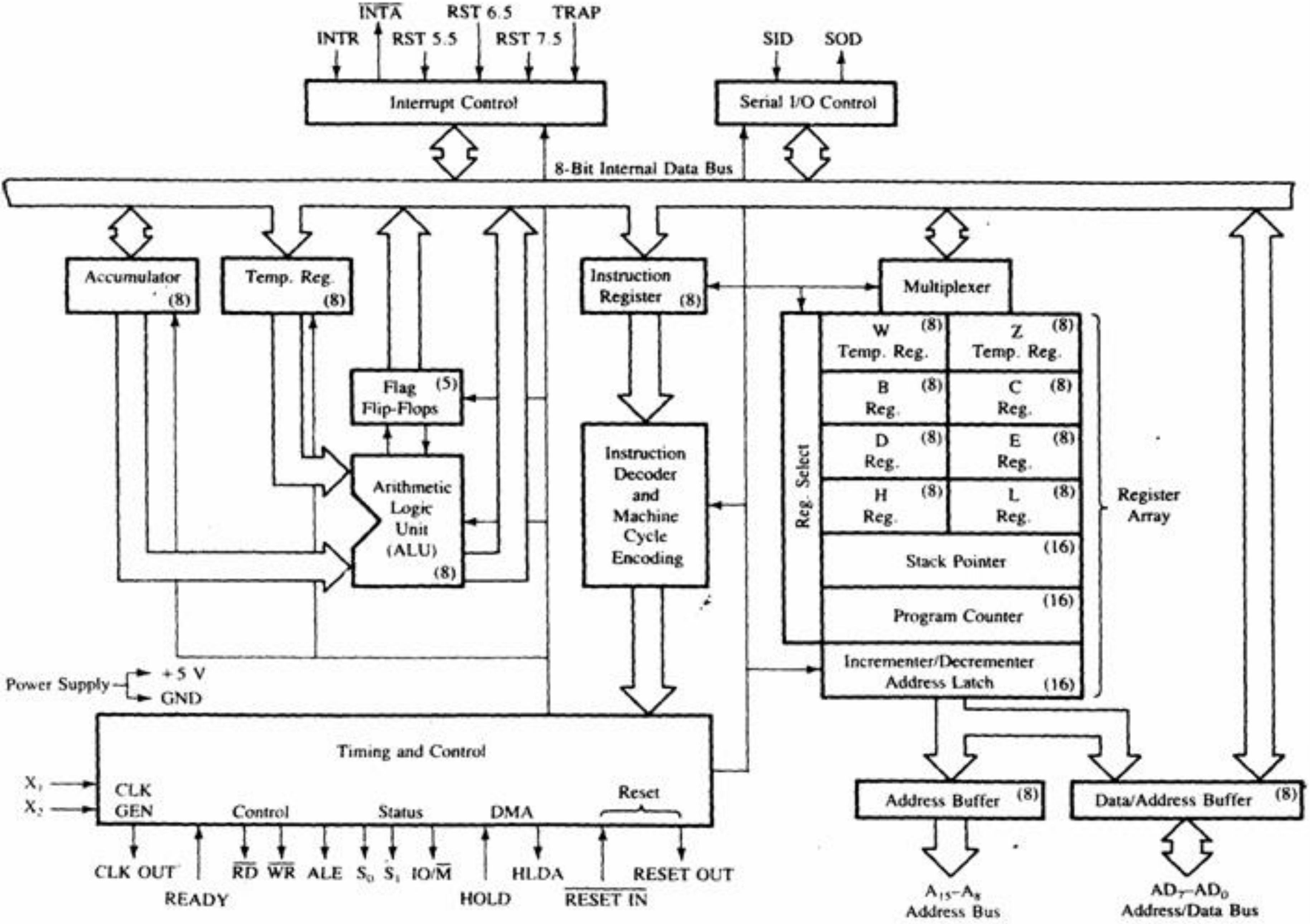
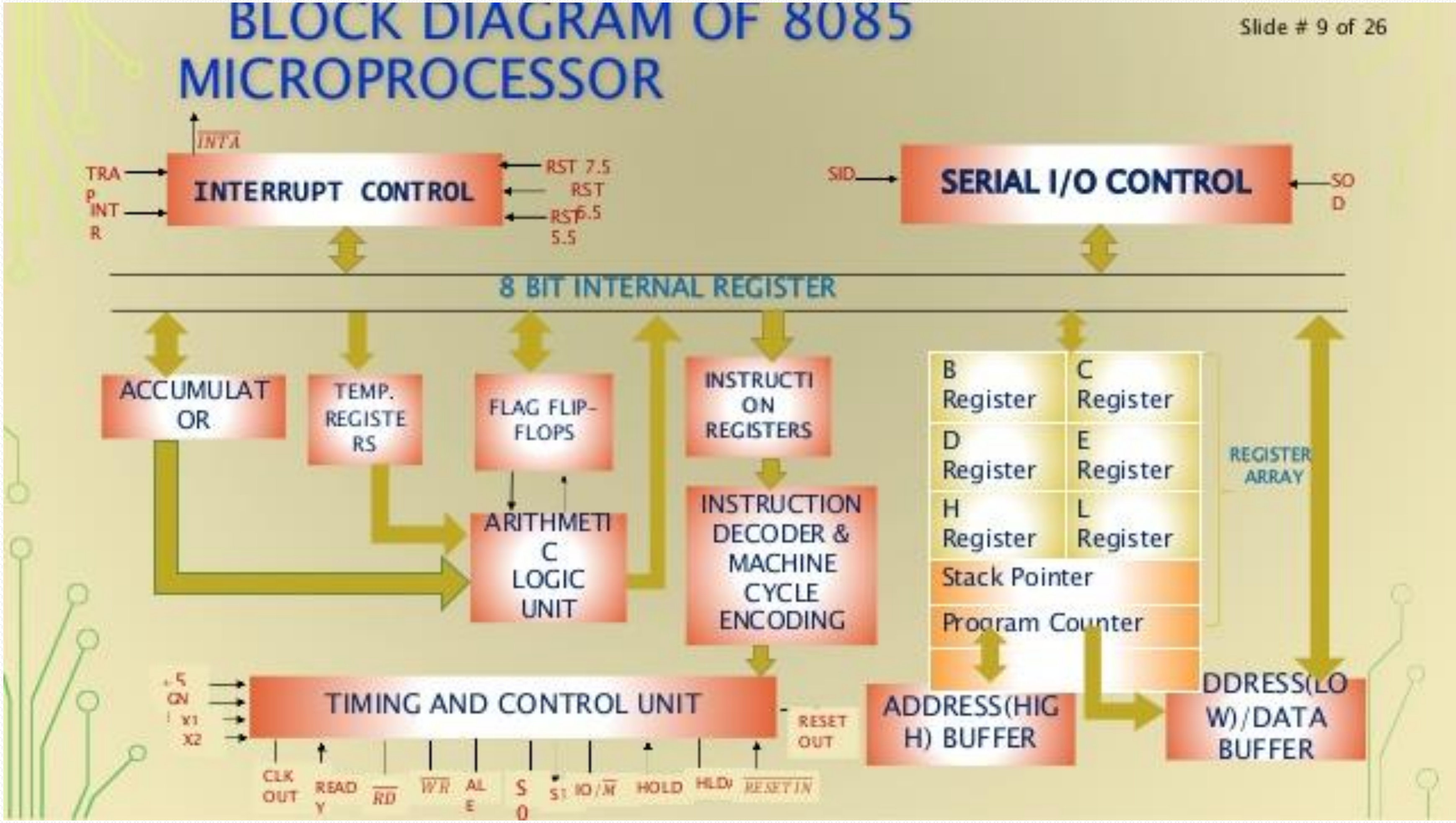


Figure 2. Block Diagram of Intel 8085



BLOCK DIAGRAM OF 8085 MICROPROCESSOR

Slide # 9 of 26

- The block diagram shows three important sections:

a) Arithmetic and Logic unit

b) Timing and control unit

c) Set of registers

Arithmetic and Logic Unit

- It performs various arithmetic and logical operations which includes:

  Addition

  Subtraction

  Increment

  Decrement

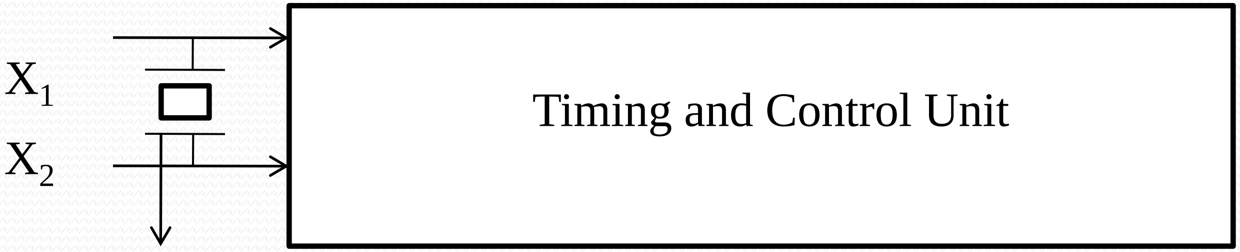
  Logical AND

  Logical OR

  Complement

  EX-OR

- **Timing and Control Unit**
- It acts as a brain of the computer system.
- It generate timing, control and status signals which are required for the operations of processor, memory and I/O devices.
- It controls data flow between processor, memory and peripheral devices.



Crystal oscillator

- **Registers**
- Registers are used by the microprocessor for temporary storage and manipulation of data and instructions.
- Data remains in the registers till they are sent to the memory or I/O devices.
- Intel 8085 has following set of registers

a) Accumulator, i.e. register A
b) General purpose register, i.e. B, C, D, E, H, L
c) Stack Pointer
d) Program Counter
e) Instruction register
f) Temporary register
g) Status Flags

❑ Accumulator

- It is an 8 bit register associated with the ALU.
- It holds one of the operands of the arithmetic and logical (AL) operation.
- The other operand is stored in the memory or in the general purpose registers.
- The final result of AL operation is placed in the accumulator.
- These are true for general cases, not for some typical or special cases.

- **Case-I**
- Some logical instructions need only one operand.

  Example-

  INR – The content of the accumulator are incremented by one.

  RAL – The content of the accumulator are rotated left by one bit.

- Some instruction has operand which is of 16 bit.

  Example-

  DAD – 16 bit addition

  One of the operand is in H-L pair and the other is in B-C or D-E pair. Result is placed in H-L pair.

- **General Purpose registers**
- The 8085 microprocessor has six 8-bit GPR, i.e. B, C, D, E, H, L.
- To hold 16 bit data a combination of two 8-bit registers can be employed.
- The combination of two 8 bit register is known as **register pair**.
- There are three register pairs, i. e. B-C, D-E, H-L.
- H-L pair is the default memory or data pointer.

❖ The general purpose registers and accumulator are accessible to the programmers.

- **Special purpose registers**

a) **Program counter –**

- It is a 16 bit register which contains or hold the address of next instruction to be executed.
- It takes care of the program flow or control.

b) **Instruction register –**

- It is a 8 bit register
- It holds the opcode of the instruction which is being decoded and executed.

- **Opcode and Operand**
- Instruction contains two parts: operation code (opcode) and operand.
- **Opcode –** It specifies the task to be performed by the computer.
- **Operand –** It specifies the data to be operated on. It can be in various forms, i.e.

i. 8 bit or 16 bit data

ii. 8 bit or 16 bit address

iii. Register

Instruction decoder and Machine cycle Encoder

- After the instruction is fetched in the IR, it is decoded into this block with the help of Micro program.
- **Micro program** is a program written by chip designer (manufacturer) to make the processor understand what a instruction is or it indicates the type of operation to be performed for an instruction.
- Number of Machine cycles are assigned according to the type of instruction.

c) **Temporary register**

- It is a 8 bit register associated with the ALU.
- It holds the data during an arithmetic and logical operation.
- It is used by the microprocessor in some instructions.
- They are not accessible by the users.
- W, Z are 8 bit temporary registers.

d) **Stack pointer**

- It is a 16 bit register which contains the address of the data present at top of the stack.
- It points to top of stack.

- **Stack**
- It is a part of R/W memory.
- Stores the content of accumulator, flags, program counter, GPR during the execution of the program.
- Stores the content of Program counter when subroutines are used.
- Any portion of the memory can be used as stack.
- It is based on LIFO (last in first out).

- **Flag register**
- The 8085 microprocessor contains five flip-flops to serve as status flags.
- The FF are set/reset according to the condition which arises during an AL operation.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	X	AC	X	P	X	CY

- The five flags of 8085 microprocessor are:

a. Carry flag – 1; if there is a carry produced after the execution of an arithmetic instruction (D_7 bit).

0; otherwise

- Parity flag – 1; if the no. of binary one is even in A.

  0; if the no. of binary one is odd in A.
- Auxiliary carry – 1; if there is a carry from D_3 bit to D_4 bit.

  0; otherwise
- Zero flag – 1; if the result of AL operation is zero.

  0; otherwise
- Sign flag- 1; if the result of AL operation is negative.

  0; otherwise

  Sign flag is judged according to the D_7 bit of accumulator.

- **Program status word (PSW)**

- PSW is a 16-bit register that stores the current status of the processor. It combines the contents of the 8-bit flag register and the contents of an 8-bit Accumulator register.
- The PSW contains information about the last operation, the state of interrupts, and the state of the processor flags.

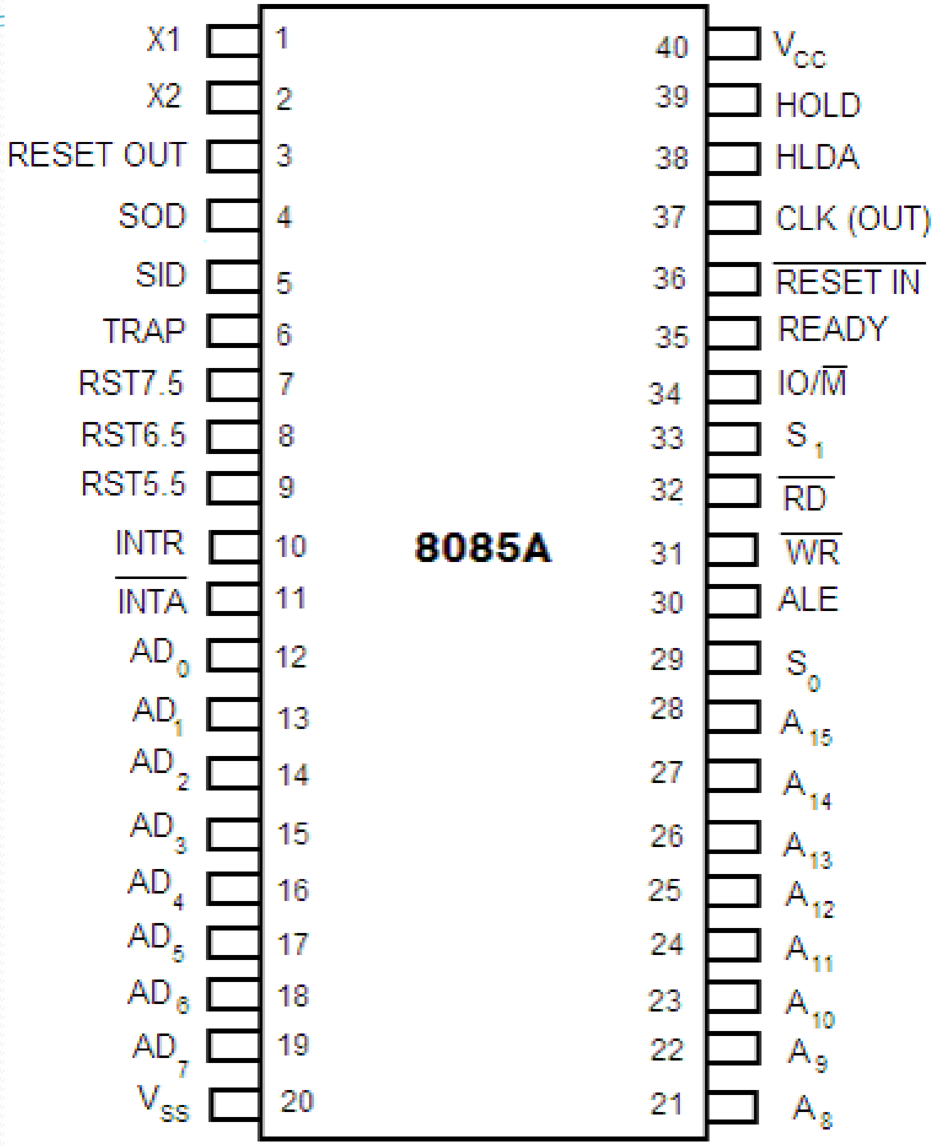


Figure 3. Intel 8085 Pin Configuration

INTERFACE SECTION

- Number of pins in the interfacing section of the microprocessor depends on the technology of that date.
- Cost of an IC depends on the number of pins and the technology of that date.
- 8085 microprocessor developed in 1976 has 40 pins.
- In today’s scenario technology support 256 pins or more for interfacing with external world, i.e. memory and I/O devices.
- Through pins different control signals are being communicated.

- **<u>Memory and I/O control lines</u>**
- $\overline{RD}$ $=$ It is a signal sent by the microprocessor to the memory/input device to control the read operation. When it goes low selected memory or input device is read.
- $\overline{WR}$= It is a signal sent by the microprocessor to the memory/ output device to control write operation. When it goes low $\xrightarrow{\hspace{2cm}}$ Data is written into selected memory or sent to output device.
- Ready = It is a signal sent by the input/output device to the microprocessor to indicate that the input/output device is ready to send or receive data.

- ALE = It is a address latch enable signal.

ALE=1	All 16 lines are used as address bus
ALE=0	A_{15}-A_8 = Address bus
	AD_7-AD_0=Data bus

- A_8-A_{15} = These are address bus and are used for the most significant bits of the memory address or I/O address.
- AD_0-AD_7 = These are time multiplexed address/data bus, i.e.

First clock cycle of a machine cycle	Least significant bits of memory or I/O address
Second and Third clock cycle	Used for data transfer

- A machine cycle is the time it takes for a microprocessor to access memory or I/O devices.

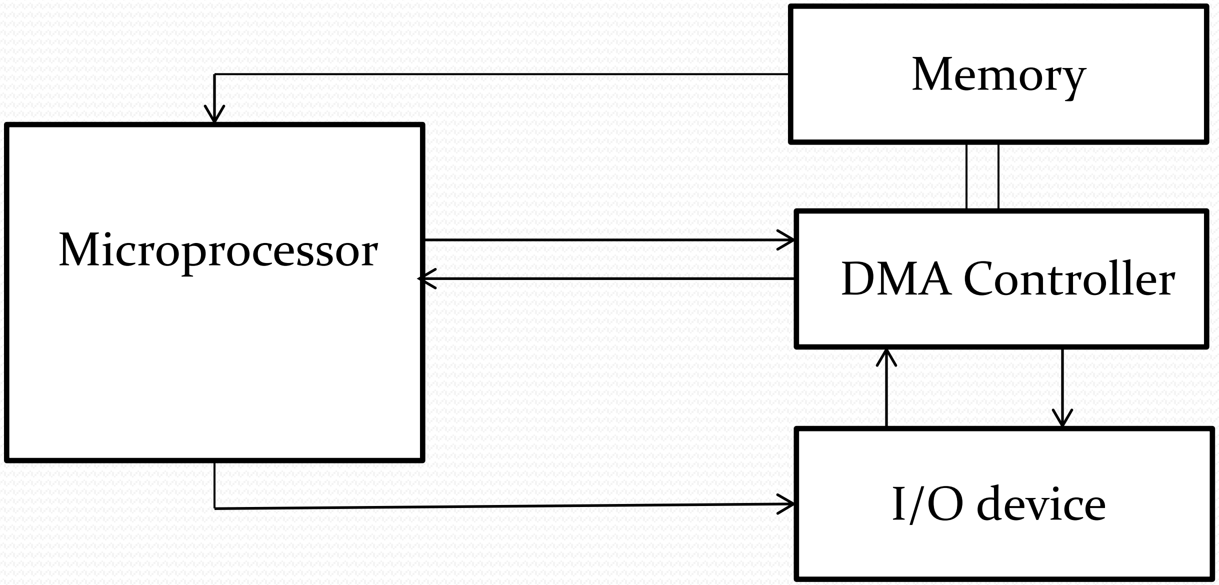
$IO/\overline{M}$=It is a status signal which distinguishes whether the address is for memory or I/O.

- Status lines (S_0 and S_1) = These are the signals sent by the microprocessor to distinguish various types of operations.

S_1	S_0	Operation
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

- **CPU and BUS Control lines**
- HOLD = When another device of the computer system, requires address/data bus for data transfer, it sends HOLD signal to the microprocessor.

- HLDA = It is a HOLD acknowledge signal sent out by the microprocessor after receiving the HOLD signal.



- DMA (Direct Memory Access) controller is an external device that controls the transfer of data between memory and input/output (I/O) devices.
- RESET IN = It reset the program counter, interrupt enable, HLDA flip-flops and instruction register.
- RESET OUT = It indicates that the CPU is being reset.

- **<u>Interrupt</u>**
- Interrupt is an internal or external signal which may disturb or alter the sequence of execution of processor.
- It is a method by which an I/O device informs the processor that it requires services of the processor.
- Interrupt can be classified as:

a. **Maskable –** Interrupt which can be avoided.

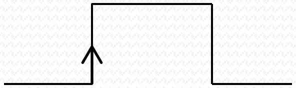
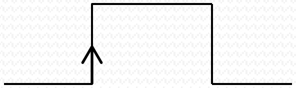
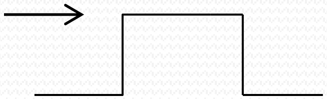
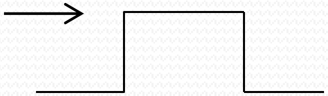
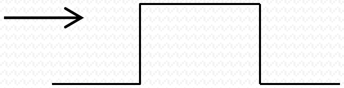
Non-Maskable - Interrupt which cannot be ignored or avoided.

b. **Vectored** – Interrupt which has specific address location in the memory.

Non-vectored – Interrupt which do not have specific address location in the memory.

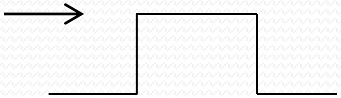
Vectored and Non-Vectored Interrupts

- In a vectored interrupt, each device is assigned a unique code, typically four to eight bits in length. When a device interrupts, it sends its unique code over the data bus to the processor. The processor uses the code to determine which interrupt service routine to execute.
- In a non-vectored interrupt, the peripheral provides the address of the interrupt service routine directly to the processor. This requires more time for an interrupt to be serviced because the address must be retrieved from the interrupting device every time the interrupt is triggered.

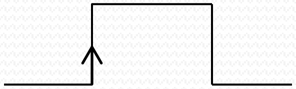
Priority	Interrupt	Type of Triggering	Vector Address
1	TRAP		0024 H
2	RST 7.5		003C H
3	RST 6.5		0034 H
4	RST 5.5		002C H
5	INTR		

MASKABLE

VECTORED



Level Triggering



Edge Triggering

- **TRAP:** The TRAP interrupt is a non-maskable interrupt that is generated by an external device, such as a power failure or a hardware malfunction. The TRAP interrupt has the highest priority and cannot be disabled.
- **RST 7.5:** The RST 7.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the second highest priority.
- **RST 6.5:** The RST 6.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the third highest priority.
- **RST 5.5:** The RST 5.5 interrupt is a maskable interrupt that is generated by a software instruction. It has the fourth highest priority.
- **INTR:** The INTR interrupt is a maskable interrupt that is generated by an external device, such as a keyboard or a mouse. It has the lowest priority and can be disabled.

- **Crystal and serial I/O lines**
- X_1 and X_2 = These are the terminals connected to external crystal oscillator which drives an internal circuitry of the microprocessor.
- SID = It is a data line for serial input. The data on this line is loaded into the 7th bit of the accumulator when Read Interrupt Mask (RIM) is executed.
  - RIM instruction reads the current value of the Interrupt Mask Register (IMR) and copies it to the accumulator.
  - An interrupt mask register (IMR) is a read and write register that specifies which interrupts should be ignored.
- SOD = It is a data line for serial output. The 7th bit of the accumulator is output on SOD line when Set Interrupt Mask (SIM) instruction is executed.
  - SIM instruction allows the microprocessor to selectively enable or disable interrupts by setting the appropriate bits in the IMR.

- **Utility lines**
- V_{cc} = +5 V supply
- V_{ss} = ground reference
- Clock = It is a output for user, which can be used for other digital ICs.

INSTRUCTION

- An instruction is a command given to the microprocessor to perform a given task on specified data.
- Instruction have two parts:
- **Opcode (Operation code)-** It specifies the task to be performed.
- **Operand** – It specifies the data to be operated on and is defined in various forms such as:

a. 8 bit or 16 bit data

b. 8 bit or 16 bit address

c. Internal register or a memory location

Instruction Word Size

- Instructions in 8085 microprocessor are classified into the following three groups according to the word (byte) size:

a. One byte instruction
b. Two byte instruction
c. Three byte instruction

- **One byte instruction**
- A one byte instruction includes the opcode and operand in the same byte.

Task	Opcode	Operand	Binary code	Hex code
Copy the contents of the accumulator in register C	MOV	C, A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator	ADD	B	10000000	80H
Invert each bit in the accumulator	CMA		0010 1111	2FH

- **Two Byte instruction**
- In a two byte instruction, the first byte specifies the operation code and the second byte specifies the operand.

Task	Opcode	Operand	Binary code	Hex Code
Load an 8 bit data in the accumulator	MVI	A,32H	0011 1110 00110010	3E 32
Load an 8 bit data in register B	MVI	B,F2H	00000110 11110010	06 F2

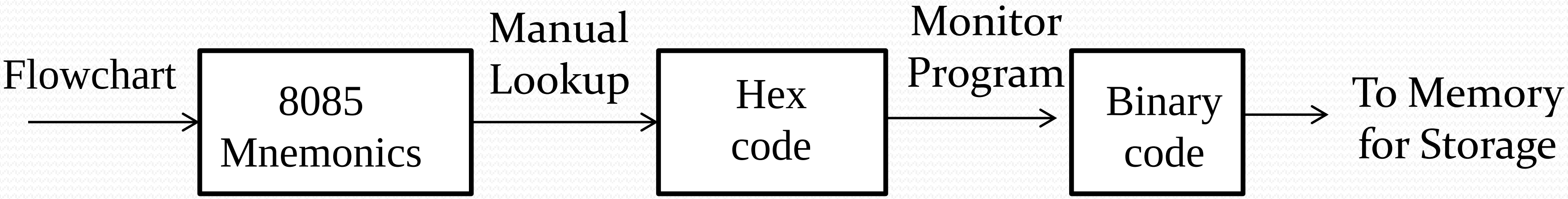
- **Three byte instruction**
- In a 3 byte instruction, the first byte represent the opcode and the second and the third byte represents the operand (16 bit address).

Task	Opcode	Operand	Binary code	Hex Code
Load the contents of the memory 2050H into A	LDA	2050H	0011 1010 0101 0000 0010 0000	3A 50 20
Transfer the program sequence to memory location 2085H	JMP	2085H	1100 0011 1000 0101 0010 0000	C3 85 20

Writing a Simple Program in 8085

- A program is a sequence of instructions written to tell a computer to perform a specific function.
- **Steps**
- Divide the problem in small steps or prepare a flowchart.
- Translate these steps into instructions, i.e. mnemonics of 8085.
- Convert the mnemonics into Hex code either manually or hand assembly.

- **Manual Assembly Process**



The Monitor program stored in the Read-Only memory (EPROM) convert the Hex code into binary code.

Addition of two numbers

- **Problem statement**

Write a program to load two hexadecimal numbers 32H and 48H in register A and B. Add the numbers and display the sum at the LED output port.

- **Problem Analysis**

a. Load the numbers in the registers
b. Add the numbers
c. Display the sum at the output port.

- Translate each step in mnemonics

MVI A,32H	Load register A with 32H
MVI B, 48H	Load register B with 48H
ADD B	Add two bytes and save the sum in A
OUT 01H	Display accumulator contents at port 01H
HLT	End

- From Assembly Language to Hex code

MVI A,32H	3E	2 byte instruction
	32	
MVI B, 48H	06	2 byte instruction
	48	
ADD B	80	1 byte instruction
OUT 01H	D3	2 byte instruction
	01	
HLT	76	1 byte instruction

Storing in memory and converting from Hex code to Binary code

a. Reset the system by pressing RESET key.

b. Enter the memory address using Hex keys where the program is to be stored.

Mnemonics	**Hex code**	**Memory contents**	**Memory Address**
MVI A, 32H	3E	0011 1110	2000
	32	0011 0010	2001
MVI B, 48H	06	0000 0110	2002
	48	0100 1000	2003
ADD B	80	1000 0000	2004
OUT 01H	D3	1101 0011	2005
	01	0000 0001	2006
HLT	76	0111 1110	2007

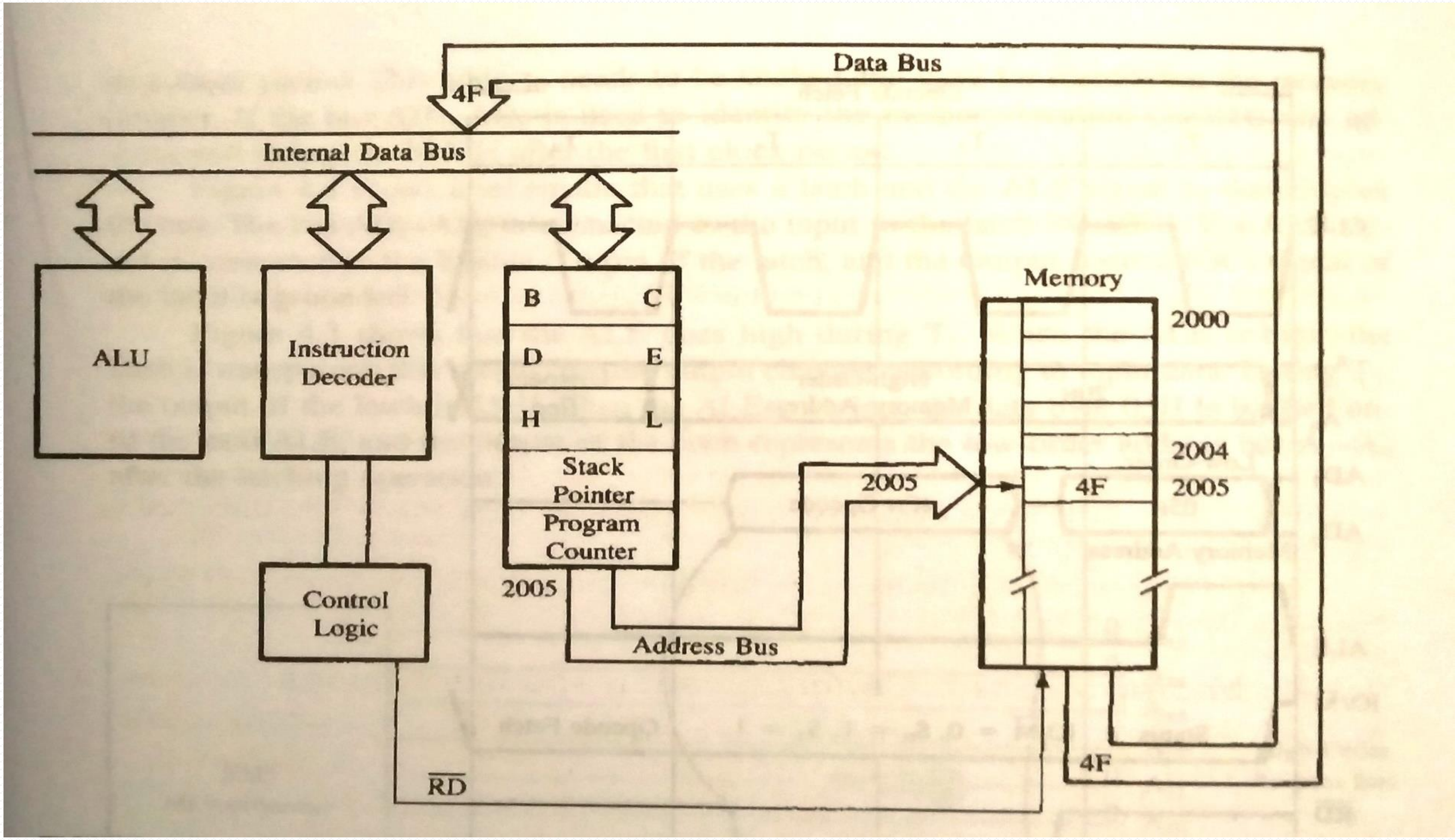
c. Execute the program.

Microprocessor Communication and Bus timings

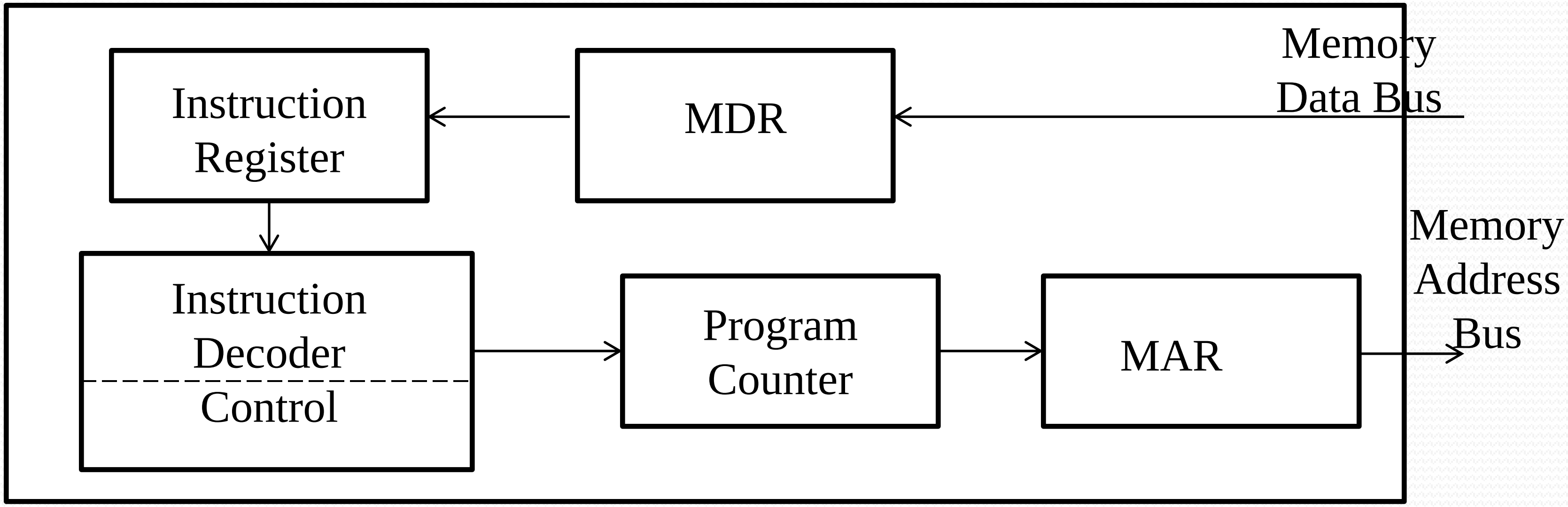
- A program constitutes a set of Instructions. The processor fetches one instruction from the memory at a time and executes it.
- The necessary steps that a processor carries out to fetch an instruction and data from memory and I/O devices, constitutes an instruction cycle.

Fetch cycle	Processor fetches opcode from memory
Execute cycle	Processor gets data from memory or I/O devices and perform specific operation
Instruction cycle	Fetch cycle+ Execution cycle

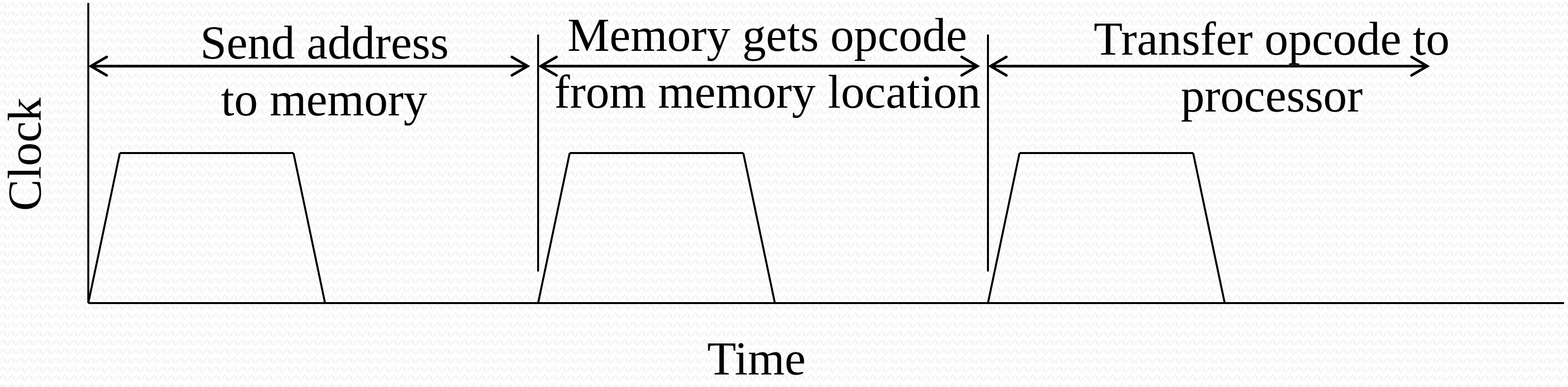
- **Fetch operation**
- The microprocessor places the 16 bit memory address from the program counter to the memory address register (MAR). The contents of MAR is transferred to the memory through the address bus.
- The control unit sends the control signal $\overline{RD}, IO/\overline{M}$ to the memory.
- The opcode from the memory location is placed on data bus.
- The opcode first come in the memory data register (MDR) and from there it goes to Instruction register.
- The instruction is decoded in the instruction decoder.



Data Flow from memory to the Microprocessor

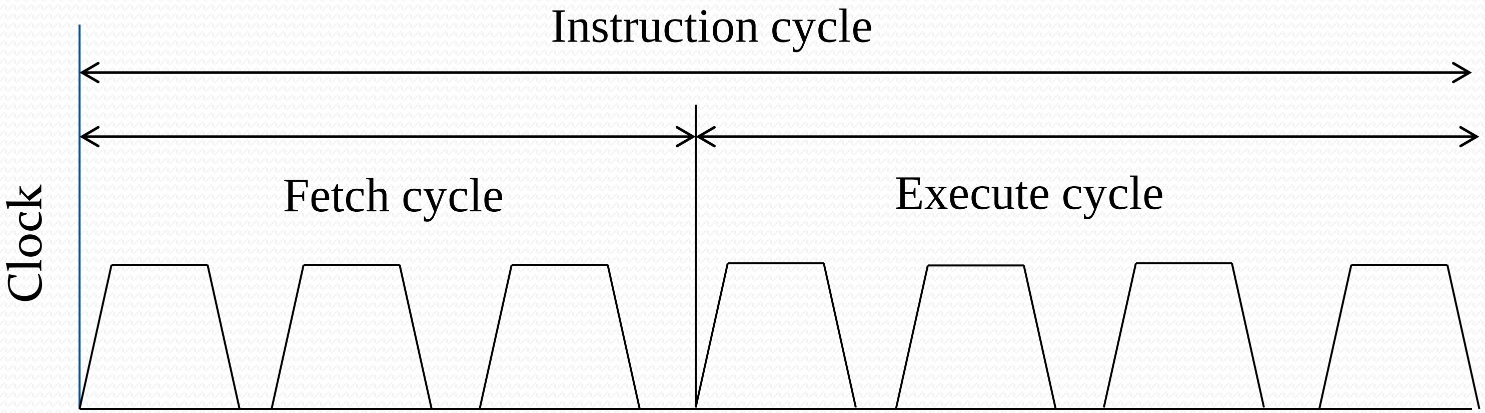


Flow of Instruction code


Send address to memory
Memory gets opcode from memory location
Transfer opcode to processor
Clock
Time

A Typical Fetch Cycle

- **Execute Operation**
- After the instruction is being decoded, execution begins.
- If the operand is in the GPR (one byte instruction), then execution is immediately performed and the time taken is one clock cycle.
- If the instruction contains data or memory address (two or three byte instruction).
- Processor performs read operation and gets data from the memory.
- In some instruction write operation is performed.
- Execution cycle contains one or more read cycles.



Instruction cycle showing FC, EC, IC

a. Instruction cycle – Time required to complete the execution of an instruction.

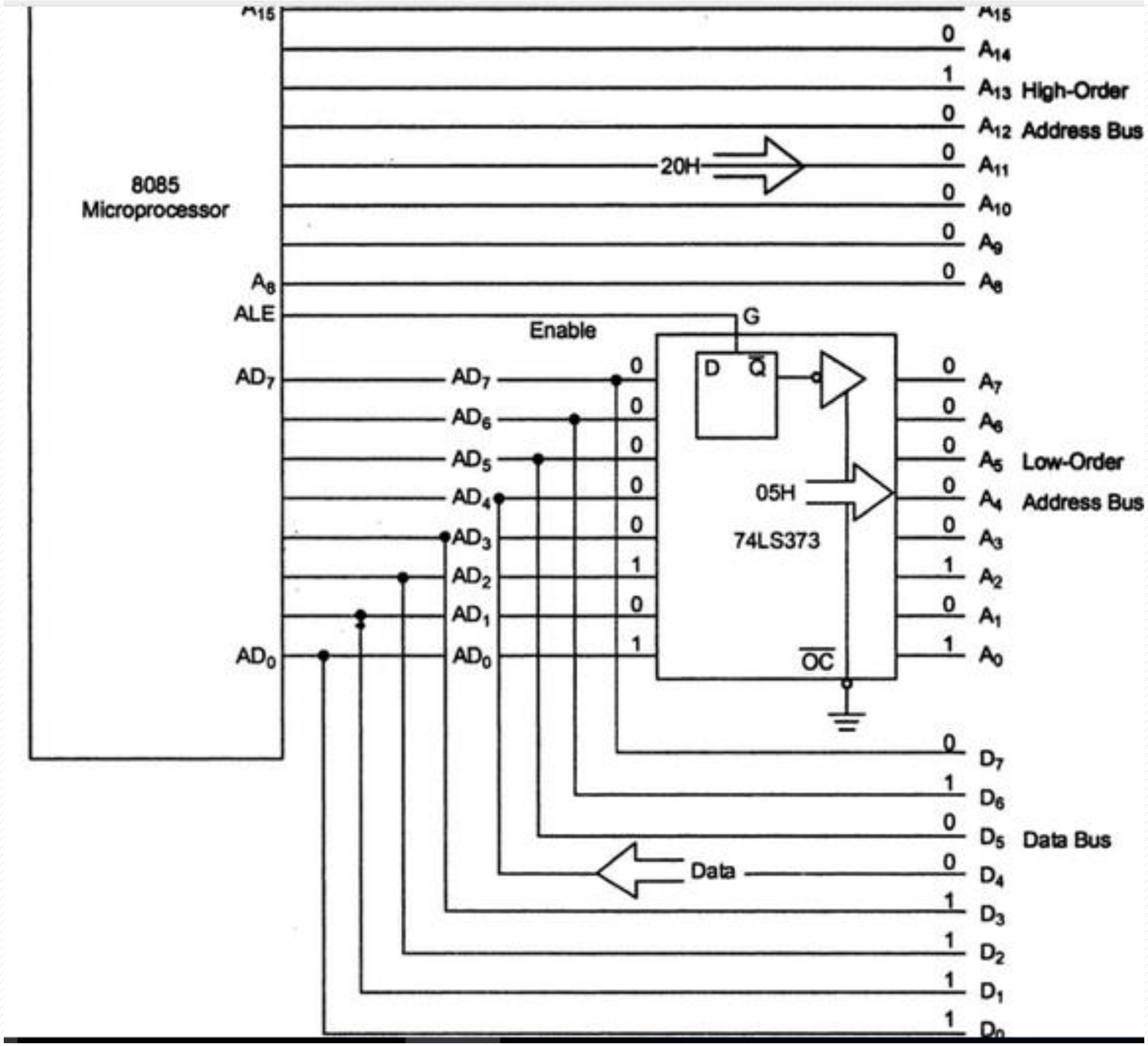
b. Machine cycle – Time required to complete one operation of either accessing memory, I/O device. Example- Fetch, Read, Write.

c. T-state- One sub division of the operation performed in one clock cycle. T-state and clock period are often used synonymously.

- Add (a) C9 H and F8 H
- (b) 98 H and 7F H
- (c) 9EH and 77H
- (d) 55H and AAH

- **Demultiplexing the Bus**
- The address/data bus transmits data and address at different moments.
- Address /data bus are time multiplexed.
- The demultiplexing is done through latch and the ALE signal.
- When ALE is high → Output changes according to the input signal.
- When ALE is low → The bus is used for carrying the data.

74LS373 is a 8-bit latch for bi-directional data transfer.



Schematic of Latching Low-Order Address Bus

Generating Control Signals

- There are different control signals which are required for the operation of microprocessor.
- The $\overline{RD}$, $\overline{WR}$, $IO/\overline{M}$ are different control signals.
- These signals are combined to generate four different signals, i.e.

a. $\overline{MEMR}$

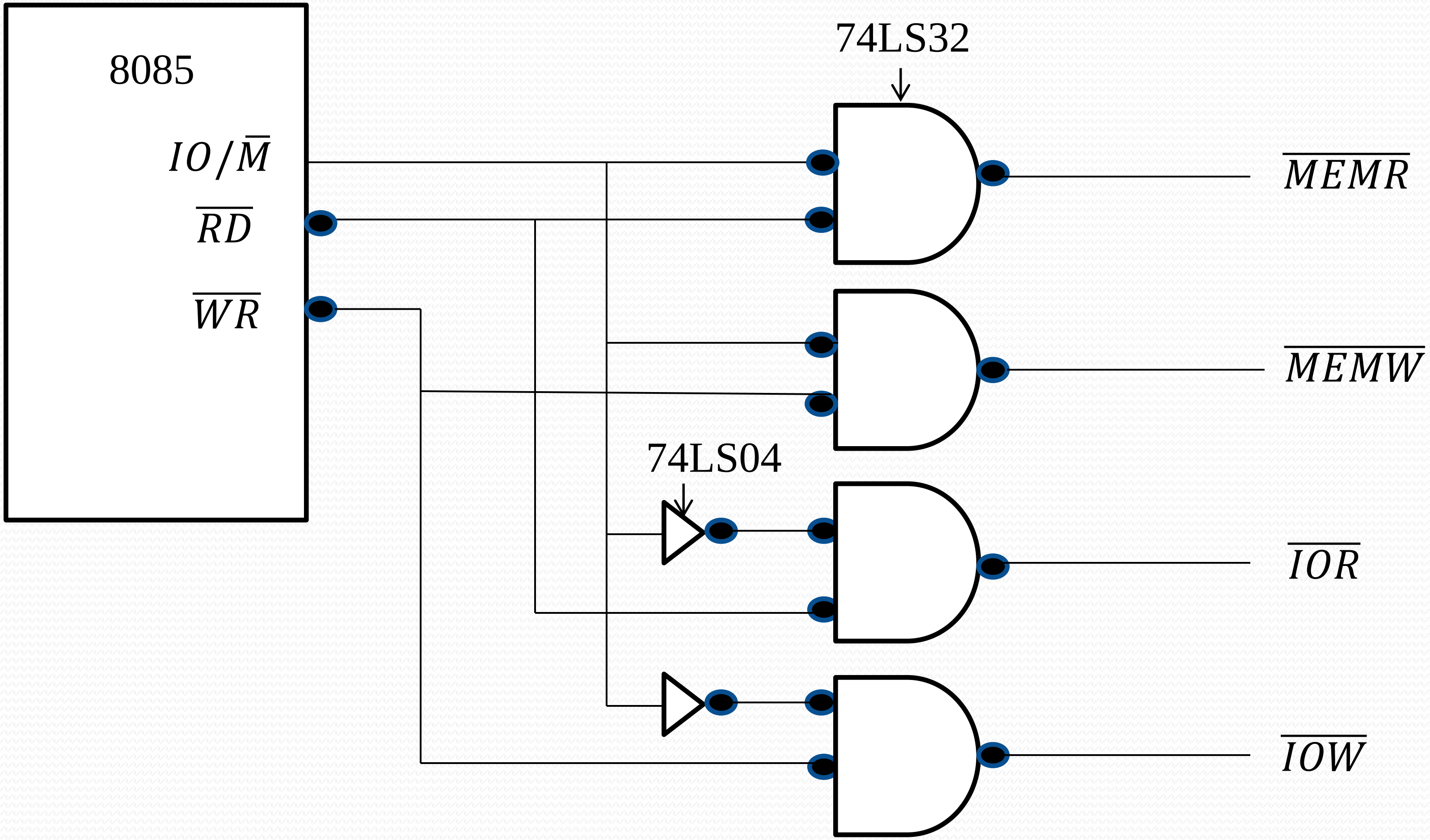
b. $\overline{MEMW}$

c. $\overline{IOR}$

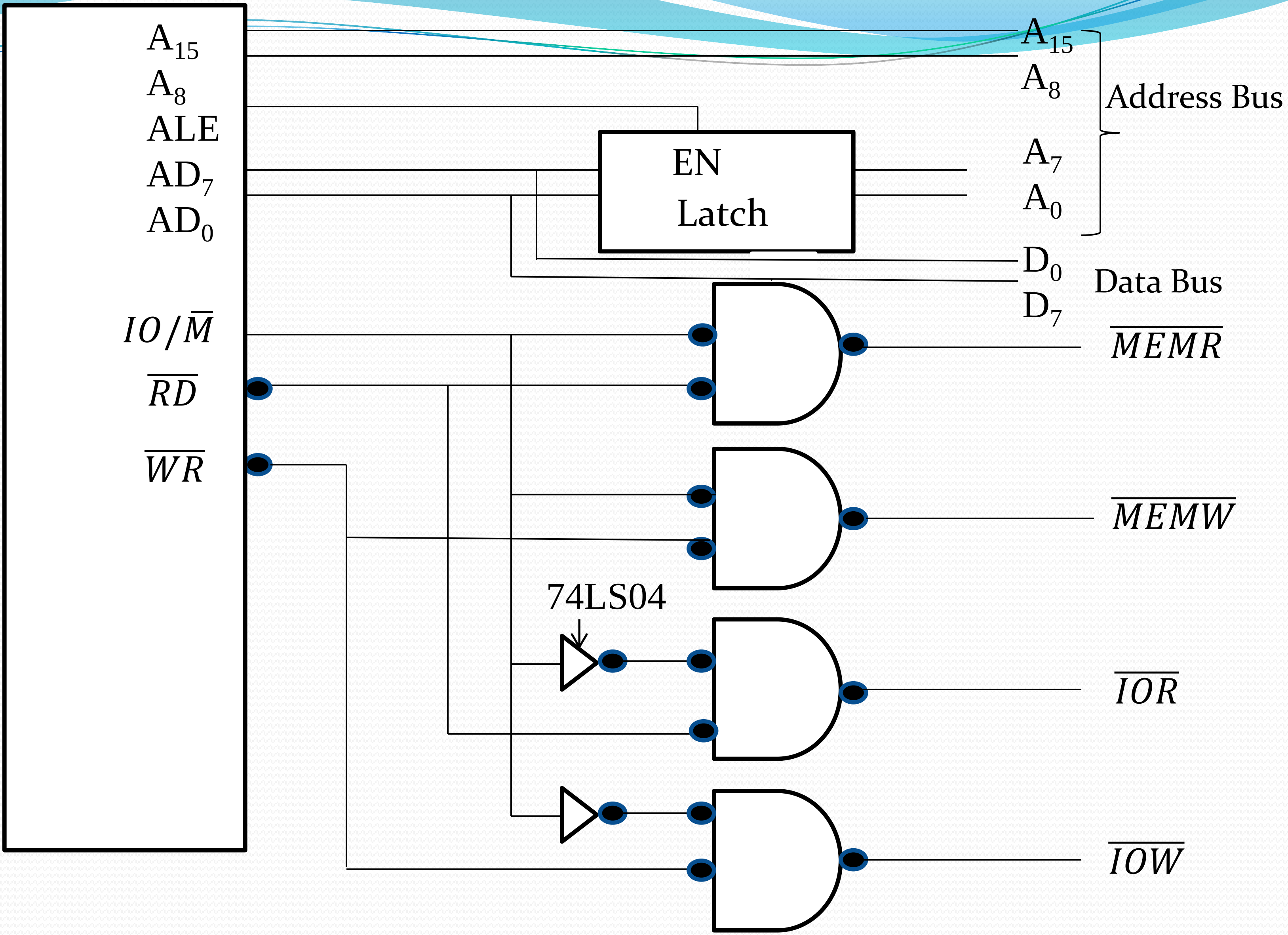
d. $\overline{IOW}$

8085 Control Signals

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	**Operation**	**Control Signals**
0	0	1	Fetch	$\overline{MEMR}$
0	0	1	M/M Read	
0	1	0	M/M Write	$\overline{MEMW}$
1	0	1	I/O Read	$\overline{IOR}$
1	1	0	I/O Write	$\overline{IOW}$



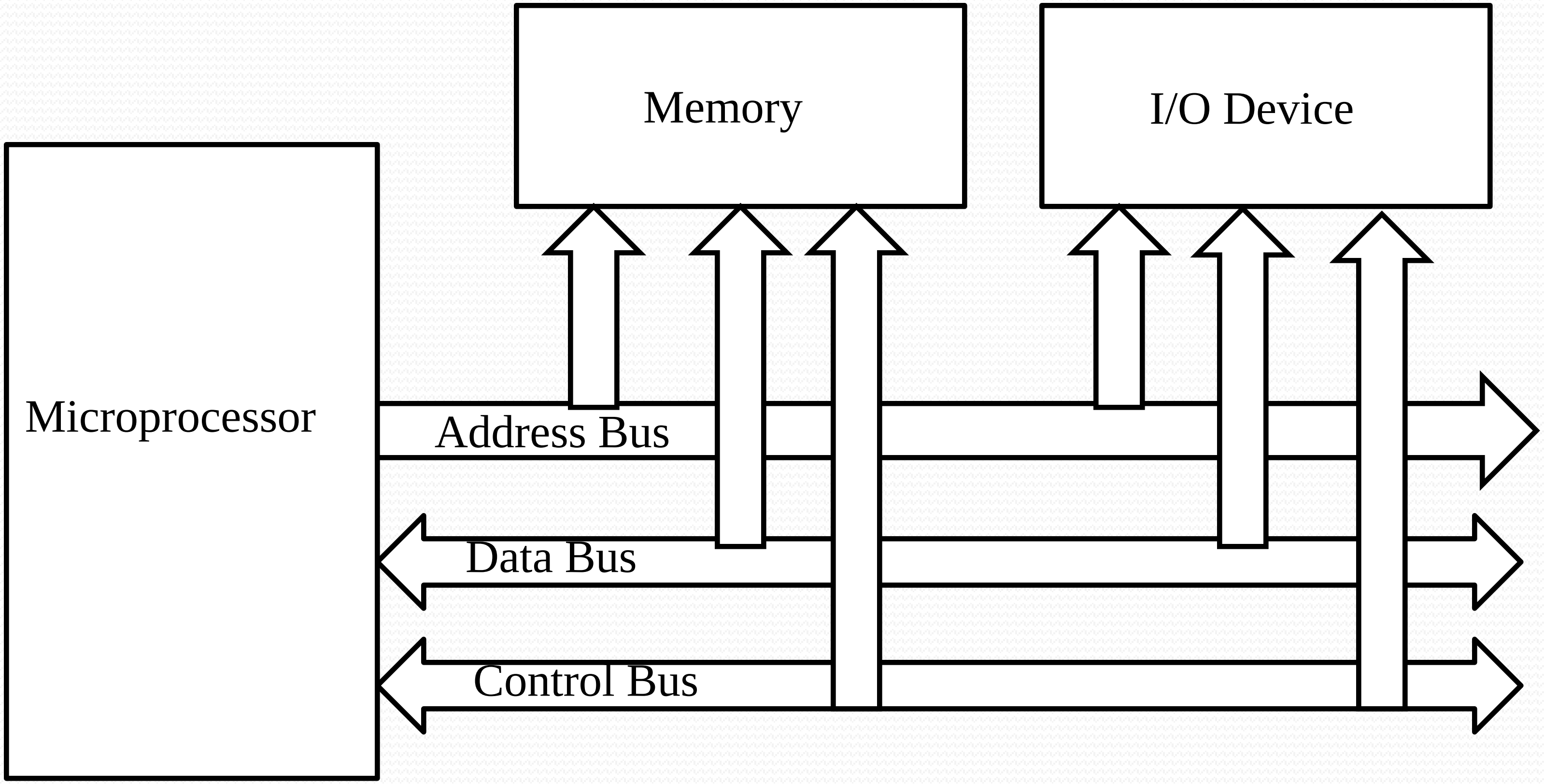
Schematic to Generate Read/Write Control Signals for Memory and IO



8085 Demultiplexed Address and Data Bus with Control Signals

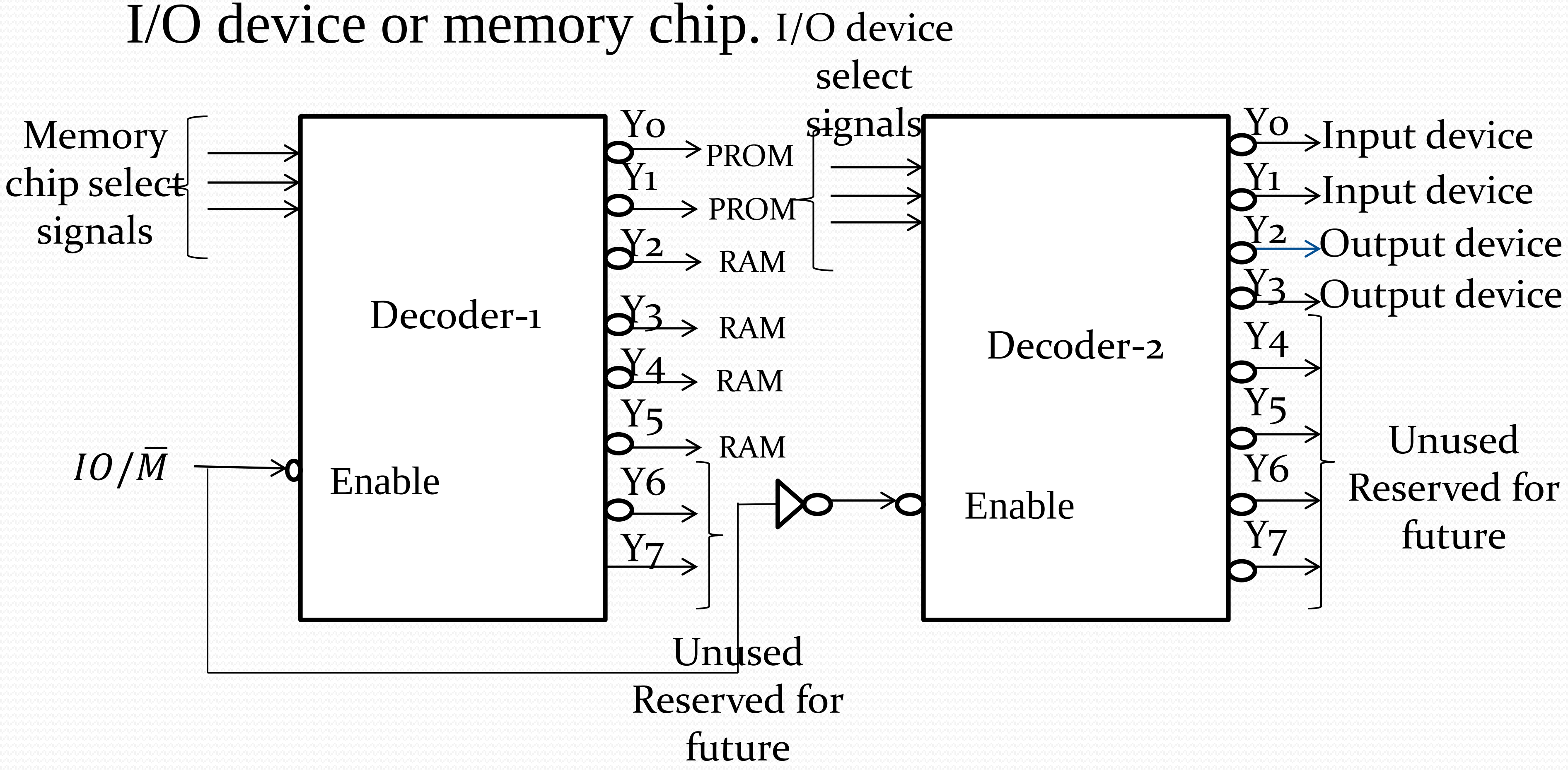
Interfacing

- Designing logic circuits (hardware) and writing instructions (software) to enable the microprocessor to communicate with peripherals is called interfacing.
- In large and minicomputers, the memories and I/O devices are interfaced to CPU by the manufacturer.
- In microprocessor based system the designer has to select compatible memories & I/O devices to interface.
- Additional electronic circuit resolve the incompatibility issue and interface memory & I/O device with processor.



Memory and I/O Interfacing

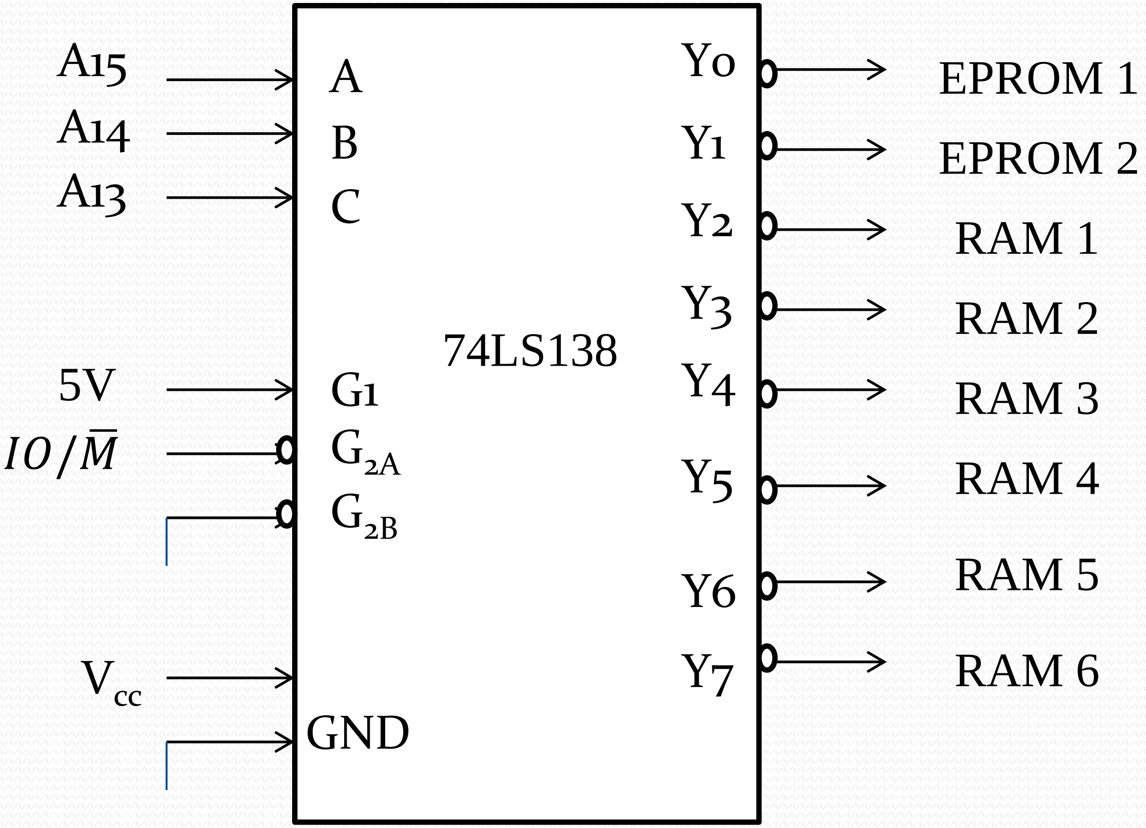
- Several memory chips and I/O devices are connected to a microprocessor.
- Address decoding circuit is required to select the required I/O device or memory chip.



Memory Interfacing

- The address of the memory location and control signals are sent by the microprocessor to initiate a read or write operation.
- Interfacing process involves designing a circuit that will match the memory requirement with microprocessor signals.
- Decoding circuit selects the corresponding memory chip or I/O device.
- Decoding can be performed by decoder, comparator, PLA.

- 74LS138 (i.e. 1 to 8 lines decoder)



Interfacing of Memory chip using 74LS138

G1, G2A, G2B are Enable signals

A, B, C are select lines

Truth Table for 74LS138

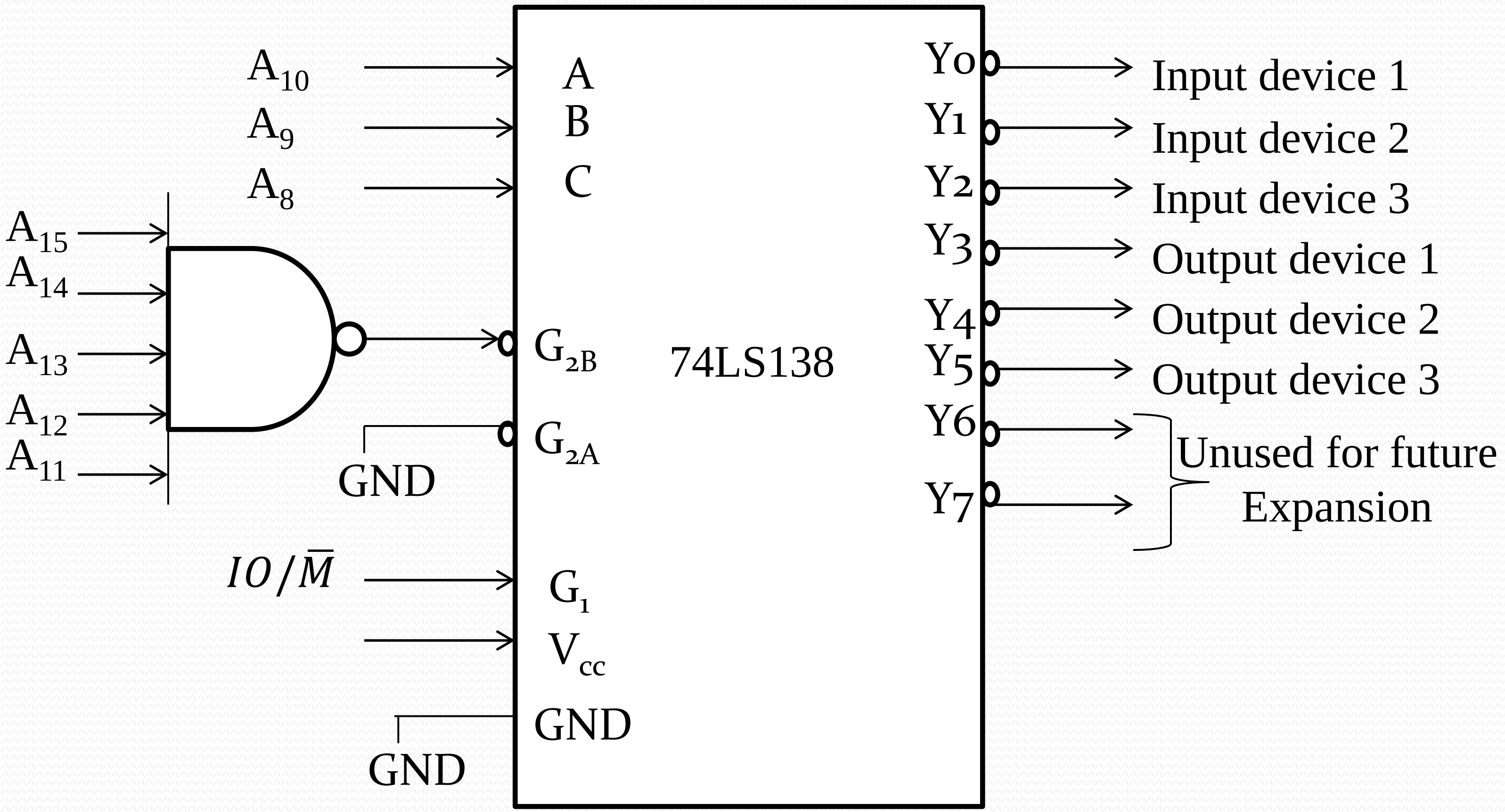
INPUTS						OUTPUTS							
ENABLE			SELECT										
G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

- Memory locations for EPROM1 will lie in the range 0000 to 1FFF. These are the memory locations for ZONE0 for the memory chip which is connected to the output line Y0 of the decoder.
- Similarly, for ZONE1 range is 2000 to 3FFF and for ZONE7 is E000 to FFFF.

Decoder Output	Memory Device	Zones of the Address Space	Memory Locations Address
Y0	EPROM 1	ZONE 0	0000 to 1FFF
Y1	EPROM 2	ZONE 1	2000 to 3FFF
Y2	RAM 1	ZONE 2	4000 to 5FFF
Y3	RAM 2	ZONE 3	6000 to 7FFF
Y4	RAM 3	ZONE 4	8000 to 9FFF
Y5	RAM 4	ZONE 5	A000 to BFFF
Y6	RAM 5	ZONE 6	C000 to DFFF
Y7	RAM 6	ZONE 7	E000 to FFFF

I/O Interfacing

- The address of an I/O device is of 8 bits, thus only A_8-A_{15} lines of address bus are used for I/O addressing.
- $A_{8,}$ $A_{9,}$ A_{10} are the select lines.
- A_{11}-A_{15} are applied to NAND gates.



Interfacing of I/O devices using 74LS138

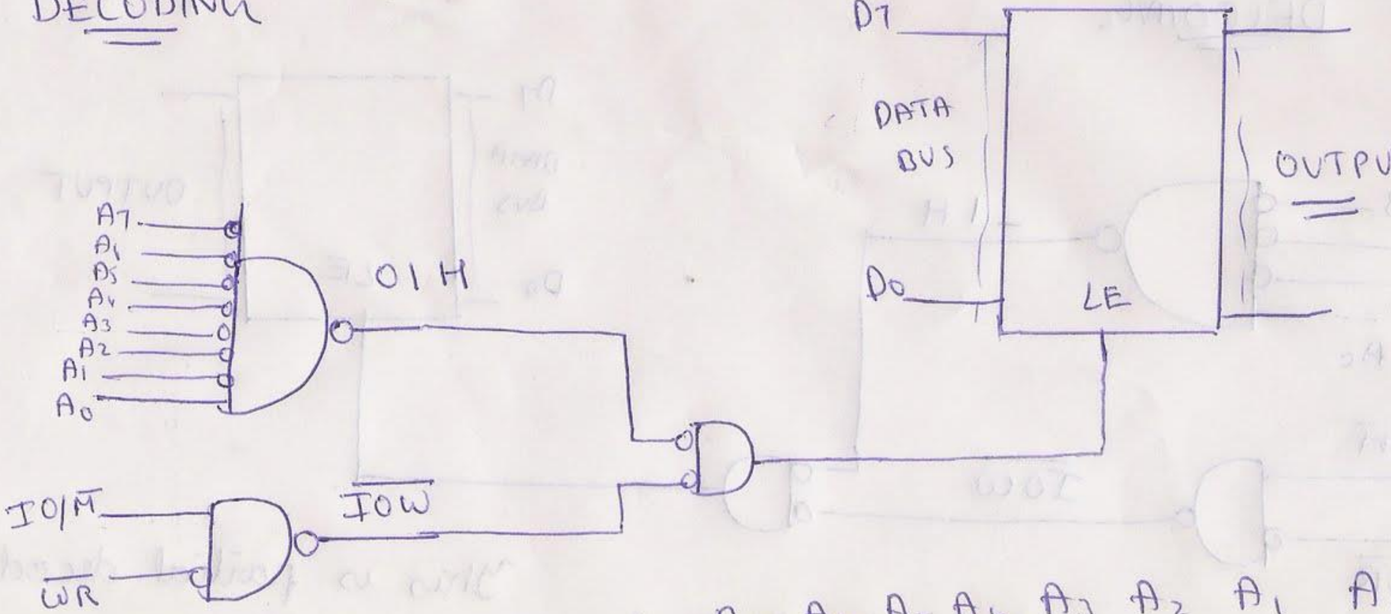
Address of I/O Devices connected to 74LS138

A15	A14	A13	A12	A11	A10	A9	A8	Selected Output Lines	Corresponding Address	I/O Device
1	1	1	1	1	0	0	0	Y0	F8	Input Device 1
1	1	1	1	1	0	0	1	Y1	F9	Input Device 2
1	1	1	1	1	0	1	0	Y2	FA	Input Device 3
1	1	1	1	1	0	1	1	Y3	FB	Output Device 1
1	1	1	1	1	1	0	0	Y4	FC	Output Device 2
1	1	1	1	1	1	0	1	Y5	FD	Output Device 3
1	1	1	1	1	1	1	0	Y6	FE	Unused
1	1	1	1	1	1	1	1	Y7	FF	Unused

Absolute vs. Partial Decoding

- **Absolute decoding-** All the high-order address lines were decoded to select the memory chip, and the memory chip is selected only for the specified logic level on these high order address lines.
- **Partial decoding-** Some of the address lines were not decoded, resulting in multiple addresses.
- **Advantage of partial decoding- Cost Saving**

EXAMPLE OF ABSOLUTE DECODING



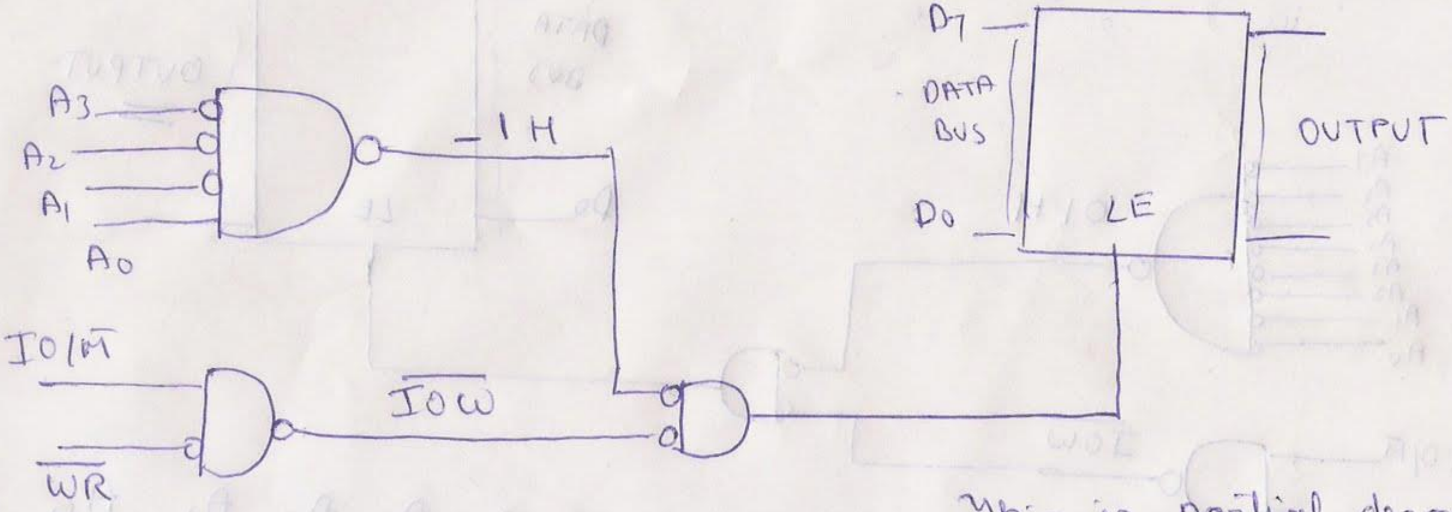
A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	1

PERIPHERAL MAPPING

(This provides a unique address of 01H)

This is absolute decoding as for I/O mapping (peripheral) 8 lines are available $A_0 - A_7$ and all these 8 lines are being used for decoding. Thats why absolute decoding.

EXAMPLE OF PARTIAL DECODING



ERIPHERAL MAPPING

A7	A6	A5	A4	A3	A2	A1	A0
x	x	x	x	0	0	0	1

Dont Care

This is partial decoding as thoug 8 lines are available only 4 lines (Lower address) is used for decoding. Means any port address X1 H can be used to access OP port like 01H 21H 31H etc.