Department of MCA

LECTURE NOTE

ON

MICROPROCESSOR AND ASSEMBLY LANGUAGE PROGRAMMIMG

COURSE CODE: MCA-102

Prepared by :

Mrs. Sasmita Acharya

Assistant Professor

Department of MCA

VSSUT, Burla.

MCA-102 Microprocessor and Assembly Language Programming L-T-P: 3-1-0

Prerequisite: Students need to know the basic building blocks of a digital system and knowledge of computer organization.

Module-1: (12 Hours)

Microprocessors: 8085 architecture, bus organization, registers, ALU, control section, pin-diagram, basic fetch and execute cycle of a program, timing diagrams, types of instructions, instruction format, data format, addressing modes, instruction set of 8085, Programming the 8085, Interrupts and ISR

Module-2: (8 Hours)

Memory Interfacing: address space partitioning, logic devices for interfacing, R/W and ROM models, memory map addresses, memory address range of 1K memory chip, memory address lines, memory word size, memory classification, memory structure and its requirements, basic concepts in memory interfacing, address decoding and memory addresses, interfacing the 8155 memory chip, absolute vs. partial decoding.

Module-3: (10 Hours)

Data transfer techniques & support chips: Data transfer techniques, programmed data transfer, direct memory access data transfer, basic idea about 8212, 8255, 8257 and 8259, analog to digital interfacing, A/D and D/A converters, analog signal conditioning circuits, data acquisition systems.

Module -4: (10 Hours)

8086 microprocessor & Microcontroller: Features of advanced microprocessors, 8086 microprocessor architecture, register organization, addressing modes; Microcontrollers and embedded processors, overview of the 8051 microcontroller family.

Text Books :

1. B. Ram, "*Fundamentals of Microprocessors and Microcomputers*", Dhanpat Rai Publications.
2. A.K.Ray and K.M.Bhurchandi – "*Advanced Microprocessors & Peripherals"* Tata McGraw Hill.
3. M.A. Mazidi and J.G. Mazidi, "*The 8051 Microcontroller and Embedded Systems*", Pearson Education, India.

Reference Books:

1. Ramesh S. Gaonkar, "*Microprocessor Architecture, Programming and Application with the 8085*", Fourth Edition, Penram International Publishing (India).

2. D.V. Hall, "*Microprocessors and Interfacing*",2nd Edition McGraw-Hill Book Company.

MODULE – 1

MICROPROCESSOR ARCHITECTURE

The microprocessor is the central processing unit or cpu of a micro computer.it is the heart of the computer.

INTEL 8085:

It is an 8 bit Nmos microprocessor.it is an forty pin IC(integrated circuit) package fabricated on a single LSI (Large scale Integration) chip.

It uses a single +5 volt d.c.(Direct Current) supply for its operation.It clock spee is 3 mhz.It consists of 3 main sections.

1-Arithmetic Logic Unit(ALU)

2-Timing and control unit

3-Several Registers

Arithmetic Logic Unit:

It performs various arithmetic an logical operations like aition,substraction,logical an ,xor,or,not,increment etc.

Timing and control unit:

It generates timing an control signals hich are necessary for the execution of the instructions.it controls the ata flo beteen cpu an peripherals.

Several Registers:

Registers:-it is a collection of flip flops use to store a binary word.they are used by the microprocessor for the temporary storage and manipulation of data and instructions.

8085 has the following registers:

1-8 bit accumulator i.e. register A

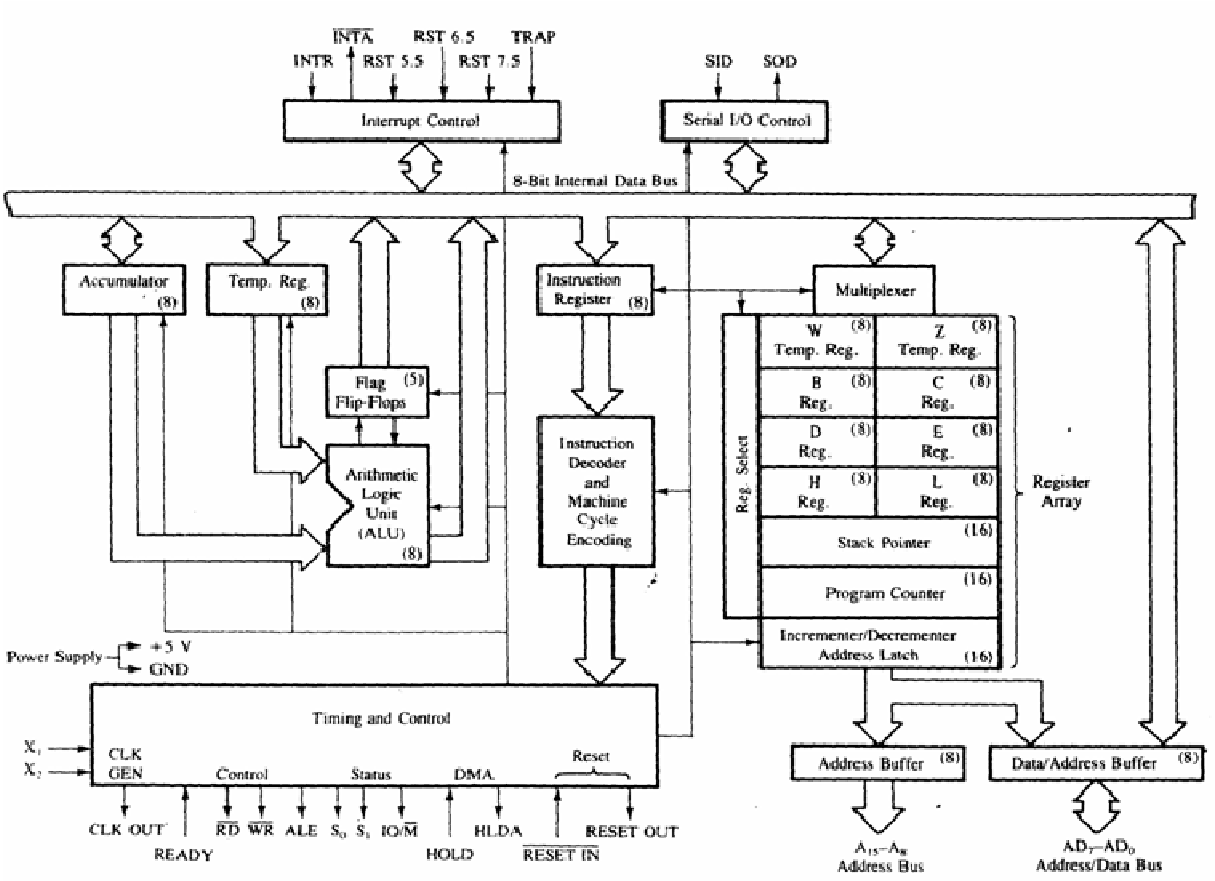
2-6 8 bits general purpose registers i.e. B,C,D,E,H,L

3-one 16 bit regiser i.e.stack pointer

4-16 bit Program counter,Status register,Temporary register,Instruction Register.

The regiser A holds the operands during program execution.

There are 6 8 bits general purpose registers B,C,,E,H,L are to handle 16 bit data.two 8 bit registers can be combined.this is called register pair.valid pair of 8085 are B-C,-E,H-L.The H-L pair is used to as address memory location.B-C,D-E are used for access another function.



BLOCK DIAGRAM OF 8085A

STACK POINTER:

Stack is a sequence of memory location defined by the programmer in LIFO function.That is last

Element to be placed on the stack is first one is to removed .The stack pointer contain the address of the stack cup.

PROGRAM COUNTER:

It is the address of the next instructions to be executed.

INSTRUCTION REGISTER:

It holds a copy of the current instruction until it is decoded.

STATUS REGISTER:

It contains the status flags of 8085 microprocessor.

TEMPORARY REGISTER:

It is used to store intermediate results and for intermediate calculations.

STATUS FLAGS:

It is a set of 5 flip-flops

i. Carry Flag(Cs)
ii. Sign Flag(S)
iii. Zero Flag(Z)
iv. Parity Flag(P)
v. Auxilarity carry flag(Ac)

Carry Flag:

It holds carry out of the resulting from the execution of an arithmetic operation.

If there is a carry from addition or a borrow from substraction or comparision,the carry flag is said to 1 ortherwise it is 0.

Sign Flag:

It is set to 1 if the MSB of the result of an arithemetic or logical operation is 1 ortherwise it is 0.

Zero Flag:

It is said to 1 if the result of an arithmetic or logical operation is zero.for non zero result,it is 0.

Parity Flag:

It is set to 1 when the the result of the operation contains even no.of 1& it is set to 0 if there are odd no.of 1.

Auxilary Carry Flag:

It holds carry from bit 3 to A resulting from the execution of an arithmetic operation.If there is a carry from bit 3 to 4,the AC flag is set to 1 ortherwise it is 0.

Program Status Word(PSW):

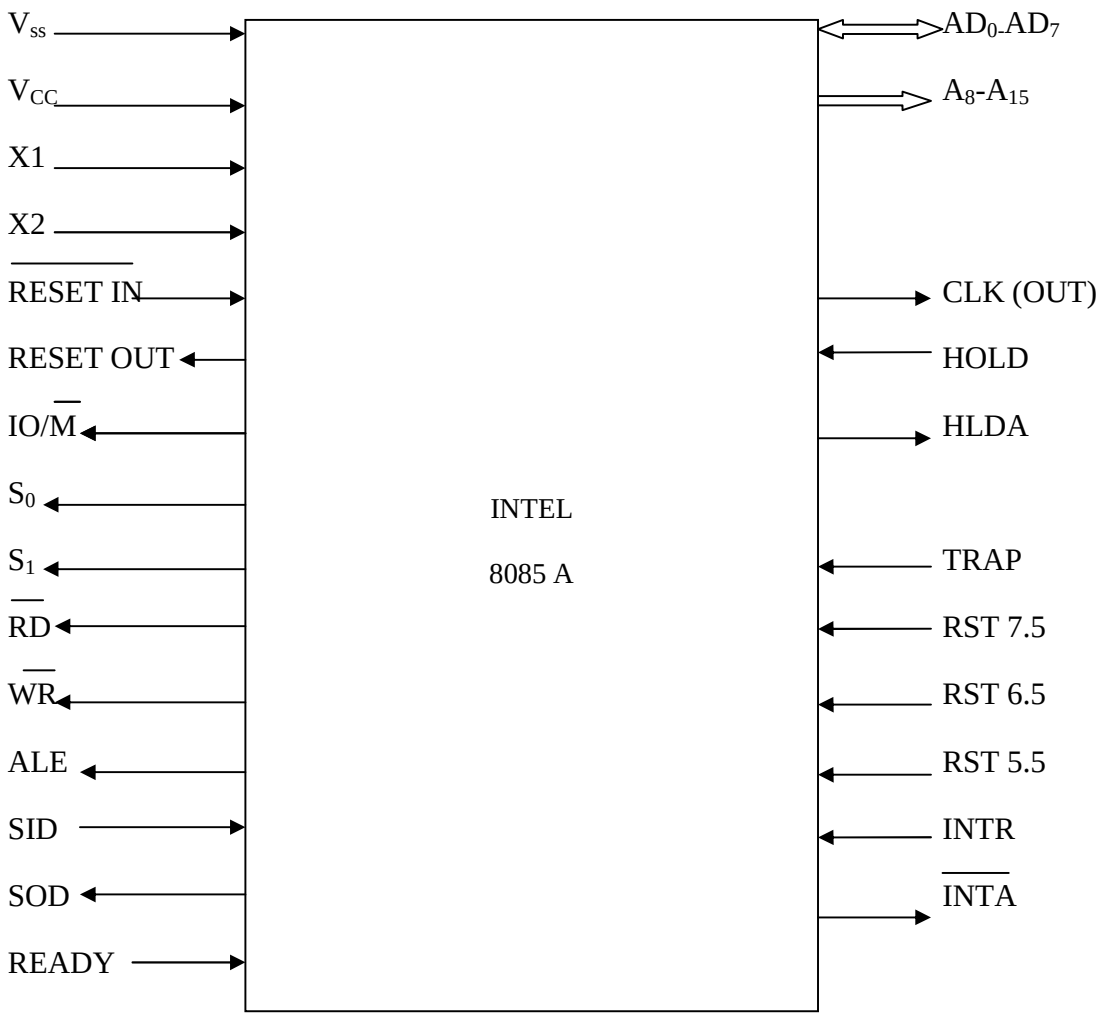
It is a combination of 8-bits where five bits indicates the 5 status flags & three bits are undefined.

Psw and the accumulator treated as a 16 bit unit for stack operation.

BUS ORGANISATION:

INTEL 8085 is a 8 bit micro processor.its data bus is 8 bit wide .8 bit of data can be transmitted in parallel form.or to the microprocessor.

Address bar is 16 bit wide as memory address are of 16 bit.8 msb is the address are transmitted by on A8-A15.8 LSB is is the address are transmitted by the data bus AD0-AD7.The address or data bus transmits data & address at different moments.it can transmits data or address at a time.



[SCHEMATIC /PIN DIAGRAM OF INTEL 8085]

PIN DESCRIPTION OF 8085

A_8-A_{15} (output)-These are address bus and are used for the most significant bits of the memory address or 8 bits of I/O address.

AD_0-AD_7 (input/output)-these are time multiplexed address /data bus that is they serve dual purpose .they are used for the least significant 8 bits of the memory address or I/O address during the first clock cycle of a machine cycle. Again they are used for data during second and third clock cycles.

ALE (output)-it is an address latch enable signal. It goes high during first clock cycle of a machine cycle and enables the lower 8 bits of the address to be latched either into the memory or external latch.

IO/$\overline{M}$(output)-it is a status signal which distinguishes whether the address is for memory or I/O. when it goes high the address on the address bus is for an I/O device. When it goes low the address on the address bus is for a memory location.

S_0, S_1 (output)-these are status signal sent by the microprocessor to distinguish the various types of operation

Status code for Intel 8085

S_1	S_0	Operations
0	0	HALT
0	1	WRITE
1	O	READ
1	1	FETCH

$\overline{RD}$ (output)-it is a signal to control READ operation .when it goes low the selected memory or I/O device is read.

$\overline{WR}$(output)-it is a signal to control WRITE operation .when it goes low the data on the data bus is written into the selected memory or I/O operation.

READY(input)-it is used by the microprocessor to sense whether a peripheral is ready to transfer data or not .a slow peripheral may be connected to the microprocessor through READY line. if READY is high the peripheral is ready .if it is low the microprocessor waits till it goes high.

HOLD (input)-it indicates that another device is requesting for the use of the address and data bus. Having received a HOLD request the microprocessor relinquishes the use of the buses as soon as the current machine cycle is completed. Internal processing may continue. the processor regains the bus after the removal of the HOLD signal. when a HOLD is acknowledged .

HLDA (output)-it is a signal for HOLD acknowledgement. It indicates that the HOLD request has been received. after the removal of a HOLD request the HLDA goes low. the CPU takes over the buses half clock cycle after the HLDA goes low.

INTR (input)-it is an interrupt request signal. Among interrupts it has the lowest priority. An interrupt is used by io devices to transfer data to the microprocessor without wasting its time.

INTA (output)-it is an interrupt acknowledgement sent by the microprocessor after INTR is received.

RST5.5, RST6.5, RST 7.5(input)-these are interrupts. Signals are the restart interrupt, they causes an internal restart to be automatically inserted each of them of a programmable mask.

TRAP-TRAP has the highest priority. It is used in emergency situation. it is an non-mask able interrupt.

Order of priority-

TRAP ⟶ RST 7.5 ⟶ RST 6.5 ⟶ RST 5.5 ⟶ INTR

When an interrupt is recognize the next instruction is executed from a fixed location in memory. A subroutine is executed which is called ISS(interrupt service subroutine).

INTERRUPTS	ISS ADDRESS
TRAP	0024
RST 5.5	002C
RST 6.5	0034
RST 7.5	003C

RESET IN (input)-it resets the program counter to zero .it also resets interrupts enable that is an HLDA flip-flops.

RESETOUT (output)-it indicates that the CPU is being reset.

X1, X2 (input)-these are terminals to be connected to an external crystal oscillator which drives an internal circuitry of the microprocessor to produce a suitable clock for the operation of microprocessor.

CLK (output)-it is a clock output for user, which can be used for other digital integrated circuits.

SID (input)-it is data line for serial input. The data on this line is loaded into the 7^{th} bit of the accumulator when rim (read interrupt mask) instruction is executed.

SOD (output)- it is data line for serial output. The 7^{th} bit of the accumulator is output on sod line when sim instruction is executed.

Vcc-it is +5 volt dc supply.

Vss-it is the ground reference.

INSTRUCTION AND DATA FORMATS

Intel 8085 is an 8-bit microprocessor. It handles 8-bit data at a time. One byte consists of 8-bits.A memory location for Intel 8085 microprocessor is designed to accumulate 8-bit data. If 16-bit data are to be stored, they are stored in consecutive memory locations. The address of memory location is 0f 16-bit i.e. 2 bytes.

The various techniques to specify data for instructions are:

(1) 8-bit or 16-bit data may be directly given in the instruction itself.

(2) The address of the memory location, I/O port or I/O device, where data resides, may be given in the instruction itself.

(3)In some instructions only one register is specified. The content of the specified register is one of the operand and other operand is the accumulator.

(4) Some instructions specify two registers. The contents of the registers are the required data.

Due to different ways of specifying data for instruction are not of same length.

So there are three types of instructions of Intel 8085:

(1)Single byte instruction

(2)2-byte instruction

(3)3-byte instruction

Single-Byte instruction.

The content information regarding operands in the opcode itself .These are of one byte.

Ex-MOV A,B ; Move the content of register B to A

78H is opcode for MOV A,B. The binary form of opcode 78H is 01111000. The first two bit i.e. 01 for MOV operation; the next 3 bits i.e. 111 for register A and last 3 bits 000 are for register B.

Two-Byte instruction.

In case of two byte instruction the 1st byte of the instruction is opcode and 2nd byte is either data or address.

Both bytes are stored in two consecutive memory locations.

Ex-MVI B,05; Move 05 to register B

06,05; MVI B,05 in the code form

Here in this case the 1st byte i.e. 06 is the opcode for MVI B and 2nd byte i.e. 05 is the data which is to be moved to register B.

Three-Byte instruction.

In case of three bytes instruction the 1st byte of instruction is opcode and 2nd and 3rd byte of instruction are either 16-bit data or 16-bit address.

They are stored in three consecutive memory locations.

Ex-LXI H, 2400H ; load H-L pair with 2400H

21,00,24; LXI H, 2400H in code form.

Here 1st byte i.e. 21 is the opcode for instruction LXI H. The 2nd byte i.e. 00 is 8 LSBs of data which is loaded in to register L. The 3rd byte i.e. 24 is 8 MSBs of data which is loaded in to register H.

ADDRESSING MODES OF 8085 :

Addressing mode: These are various technique to specify data for instruction

a) Direct addressing mode
b) Register addressing mode
c) Register addressing mode
d) Immediate addressing mode
e) Implicit addressing mode.

a) Direct addressing mode:

In this addressing mode the address of the operand is given in the instruction.

Ex: STA 2000H
IN 02H

b) Register addressing mode:

In this addressing mode the operands are in the general purpose register.

The opcode specify the address of the register and the operation to be

Perform.

Ex: MOV A,B

ADD B

c) Register indirect addressing mode:

i. In this addressing mode the address of the operand is specify by a register pair.
   Ex: LXI H,2000H
   MOV A,M

d) Immediate addressing mode:

i. In this adressing mode operand is specify with in the instruction.

ii. Ex: MVI A,05H // Move immediate data 05H to Accumulator.

e) Implicit addressing mode:

I. This instruction operates on the content of the accumulator.
II. They don't required operand address.
III. EX: CMA //Complement

DATA TRANSFER GROUP

1. MOV r1,r2 (Move data; move the content of one register to another)

[r1]←[r2]. State :none. addressing:register addressing. machine cycle:1.

The content of resister r2 is move to register is moved to register 1.For example,the instruction MOV A,B moves the contents of resister b to register A.The instruction MOV B,A moves the content of register A to register B.The time for the execution of this instruction is 4 clock period.One clock period is called is state.No lag is affected.

2. MOV r,M (move the content of memory to register)

[r]←[[H-L]]. State:7.flag:none. Addressing:register indirect. Machine cycle:2

The content of memory location,whose address is in H-L pair is moved to register r.

Example

LXI H,2000H	load H-L pair by 2000H
MOV B,M	Move the content of the memory location 2000H to register B.
HLT	Halt

In this example the instruction LXI H,2000H loads H-L pair with 2000Hwhich is the address of a memory loation.Then the instruction MOV B,M will move the content of memory location 2000H to register B.

3. MOV M,r. (Move the content of register to to memory)

[[H-L]]←[r]. States:7. Flag:none. Addressing:register indirect. Machine cycle:2

The content of register r is moved to the memory location address by H-L pair.For example,MOV M,C moves the content of register c to the memory location whose address is in H-L pair.

4. MVI r,data.(moves immediate data to register)

[r]←data. States:7. Flag:none.adressing:immediate.machine cycle :2

The 1st byte of the instruction is its opcode.the 2nd byte of the instruction is the data which is moved to register r.For example ,the MVI A,05 moves 05 to register A.In the code form it is written as 3E,05.The opcode for MVI A is 3E,05.The opcode for MVI A is 3E and 05 is the data which is to be moved to register A.

5. MVI M,data (Move immediate data to memory)

[[H-L]]←data . states:10.flags:none addressing:immediate/reg. Indirect. Machine cycle:3.

The data is moved to the memory location whose address is in H-L pair.

Example

LXI H,2400H	Load H-L pair with 2400H.
MVI M,08	Move 08 to the memory location 2400H.
HLT	Halt.

In the above example the instruction LXI H,2400H Loads H-L pair with 2400H which is the address of a memory location.Then the instruction MVI M,08 will move 08 to memory location 2400H.In the code form it is written as 36,08.The opcode for MVI M is 36 and 08 is the data which is to be moved to the memory location 2400H.

6. LXI rp, 16-bit data (load register pair immediate)

[rp]←data 16 bits, [rh]←8 MSBs, [rl]←8LSBs of data

States: 10, Flags: none, Addressing: Immediate, Machine Cycles: 3

This instruction loads 16 bit immediate data into register pair rp. This instruction is for register pair; only high order register is mentioned after the instruction. For example; H in the LXI H stands for H-L pair. Similarly, LXI B is for B-C pair. LXI H, 2500H loads 2500H into H-L pair. H with 2500H denotes that the data 2500 is in hexadecimal. In the code form it is written as 21,00,25. The 1st byte of the instruction 21 is the opcode for LXI H. The second byte 00 is of 8LSBs of the memory address and it is loaded inyo register L. The third byte 25 is 8 MSBS of the data and it is loaded into register H

7. LDA addr (Load accumulator direct)

[A]←[addr] States :13, flags :none,Addressing:Direct,Machine cycle:4

The content of memory location , whose address is specified by the 2nd and 3rd bytes of the instruction; is loaded into the accumulator.The instruction LDA 2400H will load the content of the memory location 2400H into the accumulator .Inthe code form it is written as 3A,00,24.The 1st byte 3A is the opcode of the instruction . The 2nd byte 00 is of 8LSBs of the memory address.The 3rd byte 24 is 8Msbs of the memory address.

8. STA addr (store accumulator direct)

[addr←[A].States:13.Flags:none.Addressing:direct.Machine cycle:4.

The content of the accumulator is stored in the memory location whose address is specified by tthe 2nd and 3rd byteof the instruction.STA 2000H will store the content of the accumulator in the memory location 2000H.

9. LHLD addr (load H-L pair direct).

[L]←[addr],[H]←[addr+1].States:16.Flags:none.Addressing:direct.Machine cycle:5

The content of the memory location ,whose address is specified by 2nd and 3rd bytes of the instruction,is loaded into rsister L.The content of the next memory location is loaded into resister H. For example ,LHLD 2500H will load the content of the memory location 2500H into register L. The content of the memory location 2501H is loaded into register H.

10. SHLD addr (store H-L pair direct)

[addr]←[L],[addr+1]←[H].States:16,Flags:none,Addressing:direct.Machine cycles:5

The content of the register L is stored in the memory location whose address is specified by the 2nd and 3rd bytes of the instruction.The content of register H is stored in the next memory location .For example ,SHLD 2500H will stored the content of register L in the memory location 2500H.The content of the register H is stored in the memory location2501H.

11. LDAX rp (LOAD accumulator indirect)

[A]←[[rp]].states:7,Flags:none,Addressing:register indirect,Machine cycle:2

The content of the memory location ,whose address is in the register pair rp,is loaded into the accumulator.For example ,LDAX B will load the content of the memory location,whose address is in B-C pair,into the accumulator.Th instruction is used only for B-C and D-E register pairs.

12. STAX rp (store accumulator indirect)

[[rp]]←[A].States:7.Flags:none.Addressing :register indirect.Machine cycles:2.

The content of the accumulator is stored in the memory location whose address is in the register pair rp.For example ,STAX D will stored the content of the accumulator in the memory location whose address is in D-E pair.This instruction is true only for register pair B-C and D-E.

13. XCHG (Exchange the content of the H-L with D-E pair)

[H-L]←[D-E].States:4,Flags:none,Addressing :register,Machine cycle:1.

The content of H-L pair are exchanged with contents of D-E pair

ARITHMETIC GROUP

The Instruction of this group performs arithmetic operation such as Addition, Subtraction, Increment or Decrement of the content of the register or memory.

1. Add r (Add register to accumulator)
   [A] ← [A] + [r]

The content of register r is added to the content of the accumulator,and the sum is placed in the accumulator.

2. ADD M(Add memory to accumulator)
   [A] ← [A] + [H-L]

The content of the memory location addressed by H-L pair is added to the content to the accumulator. The sum is placed in the accumulator.

3. ADC r (Add register with carry to accumulator)
   [A] ← [A] + [r] + [CS]

The content of register r and carry status are added to the content of the accumulator. The sum is placed in the accumulator.

4. ADC M (Add memory with carry to accumulator)

   [A] ← [A] + [H-L] + [CS]

The content of the memory location addressed by H-L pair add carry status are added to the content of the accumulator. The sum is placed in the accumulator.

5. ADI data (Add immediate data to accumulator)
   [A] ← [A] + data

The immediate data is added to the content to the accumulator. The 1^{st} byte of the instruction is its opcode. The 2^{nd} byte of the instruction is data and it is added to the content of the accumulator. The sum is placed in the accumulator.

FOR EXAMPLE:

The instruction ADI 08 will add 08 to the content of the accumulator and placed the result in the accumulator. In code form the instruction is written as C6 08.

accumulator.

BRANCH CONTROL GROUP

The instruction of this group change the normal sequence of the program.

There are of two types of branch instruction

- **Conditional branch instruction**
- **Unconditional branch instruction**

Conditional branch instruction:-

It transfer the program to the specified level when certain condition is satisfied.

Unconditional branch instruction:-

It transfer the program to the specified level unconditionally.

Example:-JMP addr(label).

***Conditional Jump addr(label):**

1. JZ addr(label):-Jump if the result is zero,Z=1.
2. JNZ addr(label):-jump if the result is not zero,Z=0.
3. JCaddr(label):-jump if there is a carry,CS=1.
4. JNC addr(label):-jump if there is no carry,CS=0.
5. JP addr(label):-jump if the result is plus,S=0.
6. JM addr(label):-jump if the result is minus,S=1.
7. JPE addr(label):-jump if even parity,P=1.
8. JPO addr(label):-jump if odd parity,P=0.

***CALL addr(label):-**

- Used in unconditional branch instruction.
- Used to call a sub-routine,before control its transfer to the subroutine .
- The content of program counter is saved in the stack.
- Call is 3-byte instruction.

***Conditional CALL addr(label):**

1. CC addr(label):-call subroutine if carry status CS=1.
2. CNC addr(label):-call subroutine if carry status CS=0.

3. CZ addr(label):-call subroutine if the result is zero ;the zero status Z=1.
4. CNZ addr(label):-call subroutineif the result is not zero;the zero status Z=0.
5. CP addr(label):-call subroutineif the result is plus;the sign status S=0.
6. CM addr(label):-call subroutineif the result is minus;the sign status S=1.
7. CPE addr(label):-call subroutine if even parity;the parity status P=1.
8. CPO addr(label):-call subroutine if odd parity;the parity status P=0.

***RET(Return sub routine):-**

- It is used at the end of a subroutine.
- Before the execution of a subroutine the address of the next instruction of the main program is saved in the stack.
- The content of the stack pointer is incremented by 2 to indicate the new stack top.

***Conditional Return:-**

1. RC:-Return from subroutine if carry status CS=1.
2. RNC:-Return from subroutine if carry status CS=0.
3. RZ:-Return from subroutine if the result is zero;the zero status Z=1.
4. RNZ:-Return from subroutineif the result is not zero;the zero status Z=0.
5. RP:-Return from subroutineif the result isplus;the sign statusS=0.
6. RM:-:-Return from subroutine if the result is minus ,the sign status S=1.
7. RPE:-:-Return from subroutine if even parity,the parity status P=1.
8. RPO:-_:-Return from subroutine if odd parity,the parity status P=0.

***RST n (restart) Instruction:-**

- It is a one-word call instruction the content of a program counter is saved in the stack,the program jumps to the instruction ,starting at restart location.
- The address of the restart location is 8 times n.

There are 8 RST restart instruction carrying from RST0-RST7.

- These are software interrupts used by the programmer to interrupt the microprocessor.
- The restart instruction and location are as follows:-

Instruction	**Opcode**	**Restart Location**
RST0	C7	0000

RST1	CF	0008
RST2D7		0010
RST3	DF	0018
RST4	E7	0020
RST5	EF	0028
RST6	F7	0030
RST7FF		0038

*PCHL instruction:-

- Jump to address specified by H-Lpair.
- The content of H-Lpairare transferred to the program counter.
- The content of register L will be loaded to 8 LSBs of PC and content of register H will loaded to 8 MSBs.

EXAMPLES OF ASSEMBLY LANGUAGE PROGRAMS

1. ALP FOR ADDITION OF TWO 8-BIT NUMBERS; SUM 8-BIT

Mnemonics	Operand	Comments
LXI	H, 2501 H	Get Address of 1st No. in H-L pair
MOV	A, M	1st no. in accumulator
INX	H	Increment content of H-L pair
ADD	M	Add 1st no. and 2nd no.
STA	2503 H	Store sum in 2503 H.
HLT		Stop the program.

DATA

2501- 49 H

2502- 56 H

The sum is stored in memory location 2503 H.

RESULT

2503- 9F H.

2. ALP FOR SUBTRACTION OF TWO 8-BIT NUMBERS

Mnemonics	Operand	Comments
LXI	H, 2501 H	Get address of 1st no. in H-L pair.
MOV	A, M	1st no. in accumulator.
INX	H	Content of H-L pair in 2502 H.
SUB	M	1st no. – 2nd no.
INX	H	Content of H-L pair becomes 2503 H.
MOV	M, A	Store results in 2503 H.
HLT		Stop the program.

DATA

2501- 49 H

2502- 32 H

The result is stored in memory location 2503 H.

RESULT

2503- 17 H.

3. ALP FOR ADDITION OF TWO 8-BIT NUMBERS; SUM:16-BITS

Labels	Mnemonics	Operand	Comment
	LXI	H, 2501 H	Address of 1st no. in H-L pair.
	MVI	C, 00	MSBs of sum in register C. Initial value = 00.
	MOV	A, M	1st no. in accumulator.
	INX	H	Address of 2nd no. 2502 in H-L pair.

	ADD	M	1st no. + 2nd no.
	JNC	AHEAD	Is carry? No, go to the label AHEAD.
	INR	C	Yes, increment C.
AHEAD:	STA	2503 H	LSBs of sum in 2503 H.
	MOV	A , C	MSBs of sum in accumulator.
	STA	2504 H	MSBs of sum in 2504 H.
	HLT		Stop the program.

DATA

2501- 98 H

2502- 9A H

RESULT

2503- 32 H, LSBs of sum.

2504- 01 H, MSBs of sum.

4. ALP FOR DECIMAL ADDITION OF TWO 8-BIT NUMBERS; SUM:16-BITS

Labels	Mnemonics	Operand	Comment
	LXI	H, 2501 H	Address of 1st no. in H-L pair.
	MVI	C, 00	MSDs of sum in register C. Initial value = 00.
	MOV	A, M	1st no. in accumulator.
	INX	H	Address of 2nd no. 2502 in H-L pair.
	ADD	M	1st no. + 2nd no.
	DAA		Decimal adjust.
	JNC	AHEAD	Is carry? No, go to the label AHEAD.
	INR	C	Yes , increment C.
AHEAD:	STA	2503 H	LSDs of sum in 2503 H.

MOV	A, C	MSDs of sum in accumulator.
STA	2504 H	MSDs of sum in 2504 H.
HLT		Stop the program.

DATA

2501- 84 D

2502- 75 D

RESULT

2503- 59 D, LSDs of the sum.

2504- 01 D, MSDs of the sum.

5. ALP FOR 8-BIT DECIMAL SUBTRACTION

Mnemonics	Operand	Comments
LXI	H, 2502 H	Get address of 2nd no. in H-L pair.
MVI	A, 99	Place 99 in accumulator.
SUB	M	9's compliment of 2nd no.
INR	A	10's compliment of 2nd no.
DCX	H	Get address of 1st no.
ADD	M	Add 1st no. & 10's compliment 2nd no.
DAA		Decimal Adjustment.
STA	2503 H	Store result in 2503 H.
HLT		Stop the program.

DATA

2501- 96 H.

2502- 38 H.

The result is stored in memory location 2503 H.

RESULT

2503- 58 H.

6. ALP FOR 8-BIT MULTIPLICATION: PRODUCT 16 BIT

Lable	Mnemonics	Operand	Comments
	LHLD	2501 H	Get multiplicand in H-L pair.
	XCHG		Multiplicand in D-E pair.
	LDA	2503 H	Multiplier in accumulator.
	LXI	H, 0000	Initial value of the product =00
			In H-L pair.
	MVI	C, 08	Count = 8 in register C.
LOOP:	DAD	H	shift partial product left by 1 bit.
	RAL		Rotate multiplier left 1 bit. Is multipliers bit=1?
	JNC	AHEAD	No, go to AHEAD.
	DAD	D	Product=product + multiplicand.
AHEAD:	DCR	C	Decrement count.
	JNZ	LOOP	
	SHLD	2504 H	Store result.
	HLT		Stop the program.

DATA

2501-84 H , LSBs of Multiplicand.

2502-00, MSBs of Multiplicand.

2503-56 H, Multiplier.

RESULT

2504-58 H, LSBs of product.

2505-2C H , MSBs of product.

7. ALP FOR 8-BIT DIVISION

Lable	Mnemonics	Operands	Comments
	LHLD	2501 H	Get dividend in H-L pair.
	LDA	2503 H	Get divisor from 2503 H.
	MOV	B, A	Divisor in register B.
	MVI	C, 08	Count = 08 in register C.
LOOP:	DAD	H	Shift dividend and quotient Left by one bit.
	MOV	A, H	Most significant bits of dividend In accumulator.
	SUB	B	Subtract divisor from Most significant bits of dividend.
	JC	AHEAD	Is most significant part of Dividend > divisor ? No, Go to AHEAD.
	MOV	H, A	Most significant bits of dividend In register H.
	INR	L	Yes, add 1 to quotient.
AHEAD:	DCR	C	Decrement count.
	JNZ	LOOP	Is count =0? No,Jump to LOOP.
	SHLD	2504 H	Store quotient in 2504 and Remainder in 2505 H.
	HLT		Store the program.

DATA

2501-9B H, LSBs of dividend.

2502-48 H, MSBs of dividend.

2503-1A H, divisor.

RESULT

2504-F2, Quotient.

2505-07, Remainder.

8. ALP TO SHIFT AN 8-BIT NUMBER LEFT BY ONE BIT

Example. Shift 65 left by one bit.

The binary representation of 65 is given below:

65=0110 0101

(6) (5)

Result of shifting

65 left by one bit =1100 1010=CA

(C) (A)

To shift a number left by one bit the number is added to itself. If 65 is added to 65,the result is CA as shown below.

65=0110 0101

\+ 65=0110 0101

1100 1010=CA

The number is placed in memory 2501 H.The result is to be stored in memory 2502 H.

PROGRAM

Memory Address	Machine Codes	Mnemonics	Operands	Comments
2000	3A,01,25	LDA	2501H	Get data in accumulator
2003	87	ADD	A	Shift it left by one bit
2004	32,02,25	STA	2502H	Store result in 2502H
2007	76	HLT		Halt

DATA

2501-65 H

Result

2502-CA H

The instruction LDA 2501 H transfers the number from memory location 2501 H to the accumulator. ADD A adds the contents of the accumulator to itself. The result is twice the number and thus the number is shifted left by one bit. This program does not take carry into account after ADD instruction. If numbers to be handled are likely to produce carry the program may be modified to store it.

9. ALP TO SHIFT A 16-BIT NUMBER LEFT BY ONE BIT

Shift 7596 H left by one bit. 7596=0111 0101 1001 0110

(7) (5) (9) (6)

Result of shifting left by one bit =1110 1011 0010 1100=EB2C

(E) (B) (2) (C)

The number is placed in the memory locations 2501 and 2502 H.

The result is to be stored in the memory locations 2503 and 2504 H.

PROGRAM

Memory Address	Machine Codes	Mnemonics	Operands	Comments
2000	2A,01,25	LHLD H	2501 H	Get data in H-L pair
2003	29	DAD	H	Shift left by one bit
2004	22,03,25	SHLD	2503 H	Store result in 2503 and 2504 H
2007	76	HLT		Stop

Example 1

DATA

2501-96 H, LSBs of the number.

2502-75 H, MSBs of the number.

Result

2502-2C, LSBs of the result.

2504-EB, MSBs of the result.

Example 2

DATA

2501-BF, LSBs of the number.

2502-00, MSBs of the number.

Result

2503-7E, LSBs of the result.

2504-01, MSBs of the result.

The 16-bit number has been placed in two consecutive memory location 2501 and 2502 H. The instruction LHLD 2501 H transfers the 16-bit number from 2501 and 2502 H to H-L pair.DAD H is an instruction for 16-bit addition. It adds the contents of H-L pair to itself. Thus, the 16-bit number is shifted left by one bit. The 16-bit result is stored in the memory locations 2503 and 2504 H by SHLD instruction. In some cases there may be carry after the execution of instruction DAD H. In that case carry may be stored in some register. The program may be modified accordingly.

If the shifting of an 8-bit number gives a result which is more than 8-bits, the problem can be tackled using the technique of shifting 16-bit number in Example 2.

10. ALP TO MASK OFF LEAST SIGNIFICANT 4 BITS OF AN 8-BIT NUMBER

Example .

Number=A6

=1010 0110

(A) (6)

Result=06=0000 0110

(A) (0)

We want to make off the least significant 4 bits of a given number. The LSD of the given number A6 is 6. It is to be cleared(masked off) i.e. it is to be made equal to zero. The MSD of the number A6 is A. In the binary form it is 1010. It is not to be affected. If this number is added with 1111 i.e. F, it will not be affected. Similarly, the LSD of the number is 6. In the binary form it is represented by 0110. If it is added with 0000, it becomes 0000 i.e. it is cleared. Thus, if the number A6 is added with F0, the LSD of the number is masked off.

PROGRAM

Address	Machine Codes	Mnemonics	Operands	Comments
2000	3A,01,25	LDA	2501 H	Get data in accumulator
2003	E6,F0	ANI	F0	Mask off the least significant 4 bits
2005	32,02,25	STA	2502 H	Store result in 2502 H
2008	76	HLT		Stop

DATA

2501-A6

Result

2502-A0

The instruction LDA 2501 H transfers the content of memory location 2501 H i.e. the given number to the accumulator. ANI F0 logically ANDs the content of the accumulator with F0 to clear the least significant 4 bits of the number. STA 2502 H stores the result in memory location 2502 H. HLT stops the program.

11. **ALP TO MASK OFF MOST SIGNIFICANT 4 BITS OF AN 8-BIT NUMBER**

Example.

Number=A6

=1010 0110

(A) (6)

Result=06=0000 0110

(0) (6)

To mask off 4 most significant bits of a number, 4 MSBs are added with 0000. The least significant bits are not to be affected and therefore, they are added with 1111 i.e. F. Thus, if an 8-bit number is added with 0F, the 4 most significant bits are cleared.

PROGRAM

Address	Machine Codes	Mnemonics	Operands	Comments
2000	3A,01,25	LDA	2501 H	Get data in accumulator
2003	E6,0F	ANI	0F	Mask off the most significant 4 bits
2005	32,02,25	STA	2502 H	Store result in 2502 H
2008	76	HLT		Stop

DATA

2501-A6

Result

2502-06

The instruction LDA 2501 H transfers the contents of memory location 2501 H to the accumulator. ANI 0F logically ANDs the content of the accumulator with 0F to clear the most significant 4 bits of the number. STA 2502 H stores the result in 2502 H. HLT stops the program.

12. ALP to find larger of two numbers

Example1. Find the larger of 98H and 87H.

The first number 98H is placed in the memory location 2501H.

The 2nd number 87H is placed in the memory location 2502H

The result is stored in the memory location 2503H.

The numbers are represented in the form of hexadecimal system.The first number is moved from its memory location to the accumulator.It is compared with the 2nd number.The larger of the two numbers is then placed in the accumulator.From the accumulator the larger number is moved to the desired memory location.

PROGRAM

Memory addr	machine code	labels	memonics	operands	comments
2000	21,01,25		LXI	H,2501H	Address of 1st no in HLpair
2003	7E		MOV	A,M	1st no in accumulator
2004	23		INX	H	Address of 2nd no in HLpair
2005	BE		CMP	M	Compare 2nd no with 1st no
					Is the 2nd no >1st
2006	D2,0A,20		JNC	AHEAD	No,larger number is in
					Accumulator.Goto AHEAD
2009	7E		MOV	A,M	Yes get 2nd number in the
					accumulator
200A	32,03,25	AHEAD	STA	2503H	Store larger number in
					2503H
200D	76		HLT		STOP

DATA:

2501-98H

2502-87H

Result is 98H and it is stored in memory location 2503H

RESULT:

2503-98H

13. ALP to find the largest number in a data array

The no in a series are 98,75,99 as there are three nos in the series the count=03. The count is placed in the memory location 2500H. The nos are placed in the memory location 2501 to 2503H . The result is to be stored in the memory location 2450H.The 1st no of the series is placed in the accumulator and it is compaied with the 2nd no residing in the memory.The larger of two nos is placed in the accumulator .Again this no which is in the accumulator is compared with the third no of the series and the larger no is placed in the accumulator.This process of comparision is repeated till all the nos of the series are compared and the largest nomis stored in the desired memory location.

PROGRAM:

Memory addr	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,00,25		LXI	H,2500H	Address for count In HL pair
2003	4E		MOV	C,M	Count in register C
2004	23		INX	H	Address of 1st no in HLpair
2005	7E		MOV	A,M	1ST no in accumulator
2006	0D		DCR	C	Decrement count
2007	23	LOOP	INX	H	Address of next number
2008	BE		CMP	M	Compare next number With pervious maximum. Is next no>previous maximum?
2009	D2,0D,20		JNC	AHEAD	No,larger number is in Accumulator go to the Lable AHEAD
200C	7E		MOV	A,M	Yes, get larger in Accumulator
200D	0D	AHEAD	DCR	C	Decrement count

200E	C2,07,20	JNZ	LOOP	
2011	32,50,24	STA	2450H	Store result in 2450H
2014	76	HLT		Stop

EXAMPLE:

DATA:

2500-03

2501-98

2502-75

2503-99

RESULT:

2450-99

14. ALP to find the smaller of two numbers

Find the smaller of 84H and 99H.

The first no 84H is placed in the memory location 2501H.

The 2nd no 99H is placed in the memory location 2502H

Store the result in the memory location 2503H.

The nos are represented in hexadecimal no system.The first no is moved from its memory location to the accumulator.It is compared with the 2nd no.The smaller of the two is then placed in the accumulator.From the accumulator the smaller no is moved to the desired memory location where it is to be stored.

PROGRAM

Memory addr	machine code	labels	memonics	operands	comments
2000	21,01,25		LXI	H,2501H	Address of 1st no in HLpair
2003	7E		MOV	A,M	1st no in accumulator
2004	23		INX	H	Address of 2nd no in HLpair
2005	BE		CMP	M	Compare 2nd no with 1st no

					Is the 2nd no >1st
2006	D2,0A,20		JC	AHEAD	No,larger number is in Accumulator.Goto AHEAD
2009	7E		MOV	A,M	Yes get 2nd number in the accumulator
200A	32,03,25	AHEAD	STA	2503H	Store larger number in 2503H
200D	76		HLT		STOP

Example- 1

DATA: 2501-98H

2502-87H

Result is 87H and it is stored in memory location 2503H

15. ALP to find the smallest number in a data array

The number of a series are 8658 and 75.

As there are three numbers in the series,count=03

The count is placed in the memory location 2500H

The numbers are placed in the memory location 2501 to 2503H

The result is to be stored in the memory location 2450H

The 1st no of the series is placed in the accumulator and it is compaired with the 2nd no residing in the memory.The Smaller of two nos is placed in the accumulator .Again this no which is in the accumulator is compared with the third no of the series and the Smaller no is placed in the accumulator.This process of comparision is repeated till all the nos of the series are compared and the Smallest number is stored in the desired memory location.

PROGRAM:

Memory addr	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,00,25		LXI	H,2500H	Address for count In HL pair
2003	4E		MOV	C,M	Count in register C
2004	23		INX	H	Address of 1st no in HLpair
2005	7E		MOV	A,M	1ST no in accumulator
2006	0D		DCR	C	Decrement count
2007	23	LOOP	INX	H	Address of next number
2008	BE		CMP	M	Compare next number With pervious maximum. Is next no>previous maximum?
2009	D2,0D,20		JC	AHEAD	No,larger number is in Accumulator go to the Lable AHEAD
200C	7E		MOV	A,M	Yes, get larger in Accumulator
200D	0D	AHEAD	DCR	C	Decrement count
200E	C2,07,20		JNZ	LOOP	
2011	32,50,24		STA	2450H	Store result in 2450H
2014	76		HLT		Stop

EXAMPLE:

DATA:

2500-03H

2501-86H

2502-58H

2503-75H

RESULT:

2450-58H

TIMING DIAGRAM FOR I/O READ AND MEMORY READ

I/O READ:

1. In I/O read cycle the microprocessor reads the data available at an input port/input device. The data is placed In accumulator.

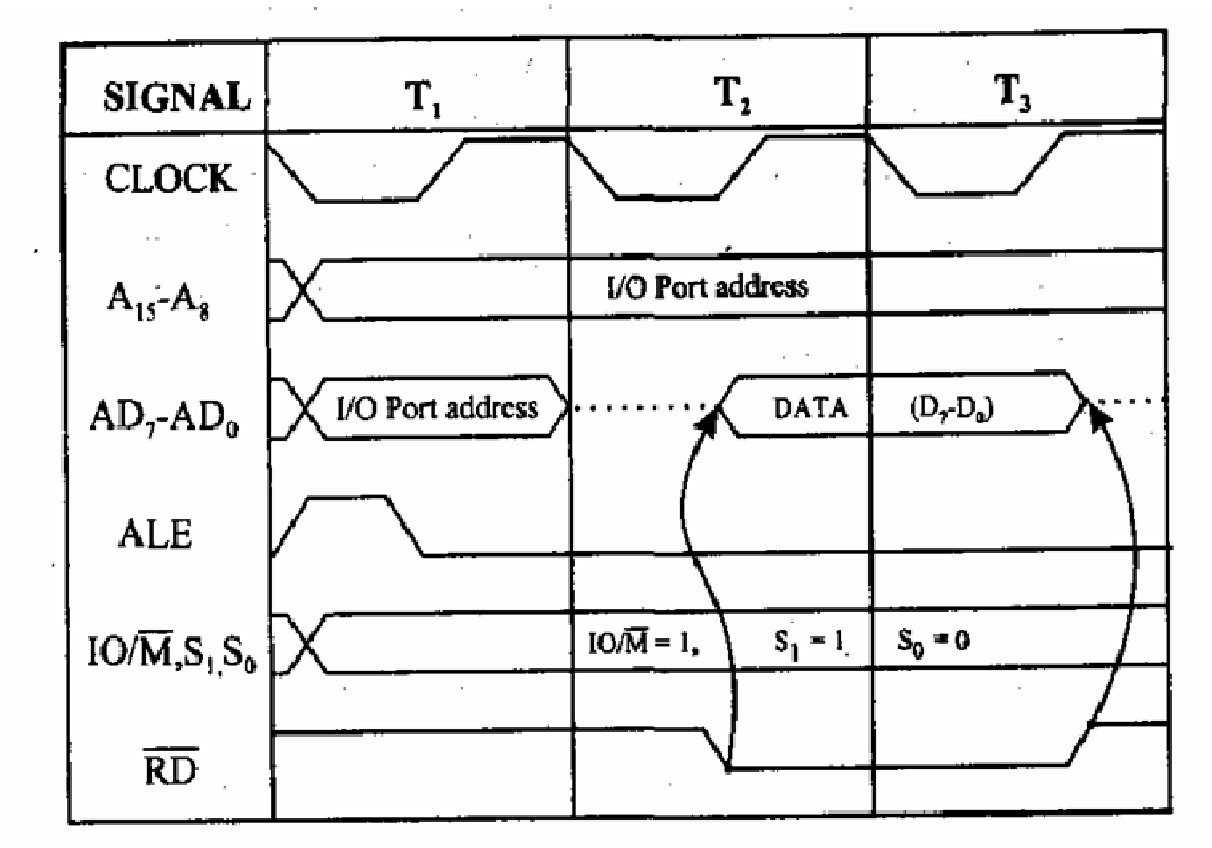
2.An I/O read cycle it is similar to memory read cycle except that signal I/O/M(bar) goes high in case of I/O read.

3.It indicates that the address on the address bus is for an input device.

4.It case of I/O device or I/O port the address is only 8 bit long. So the address of I/O device is duplicated on both address and address data bus.

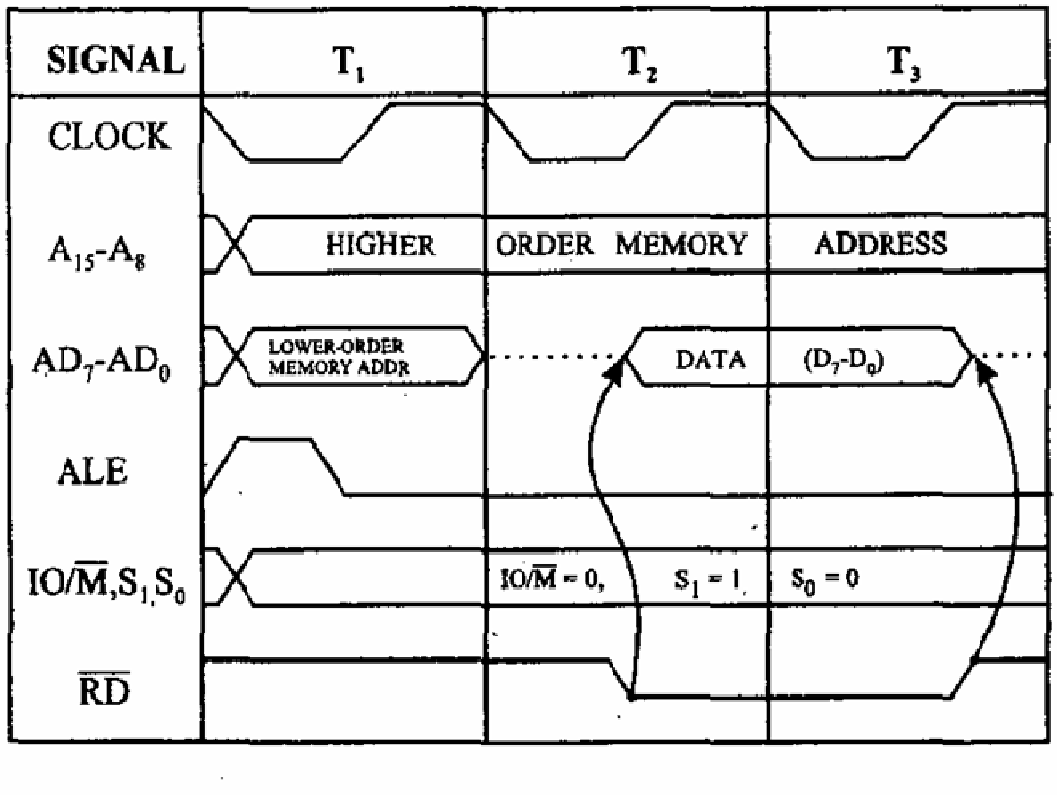
Ex-IN instruction is an example of I/O READ.

5.It is 2bytes long. So it requires 3 machine cycles such as opcode fetch cycle, memory read cycle to read the input port address, I/O read cycle to read data from input port



TIMING DIAGRAM FOR MEMORY READ CYCLE

1. In memory read cycle the microprocessor read the content of memory location
2. The content is the placed in either accumulator any other CPU register.
   Ex-MVI A, 05H
3. In this cycle io/m(bar) goes low indicate that the address is for memory.
4. S1 and S0 are set to be 1 and 0 respectively for read operation.
5. Address lines A8 to A15 carry 8MSBs of memory address of data.
6. Address lines AD0 to AD7 carry 8LSBs of memory address of data.
7. During T2 8LSBs of address is latched AD0 –AD7 are made free for data transfer.
8. RD(bar) goes low in T2 enable the memory read operation.
9. Now data is placed in data bus during T3 data enters the cpu.
   Ex-LXI H,2000H



TIMING DIAGRAM FOR MEMORY WRITE CYCLE-

1. In a memory write cycle the CPU sends data from accumulator or any other register to memory.

2. The status signal S_0 and S_1 are 1 and 0 respectively for write operation.

3. goes low in T_2 indicating that the write operation to be performed.

4. During T_2 the address/data bus is not disable but the data to be sent out to memory placed on the address/data bus.

5. As soon as goes high in T_3 the write operation is terminated.

Example: - 1. MOV M, A

2. STA 2000H

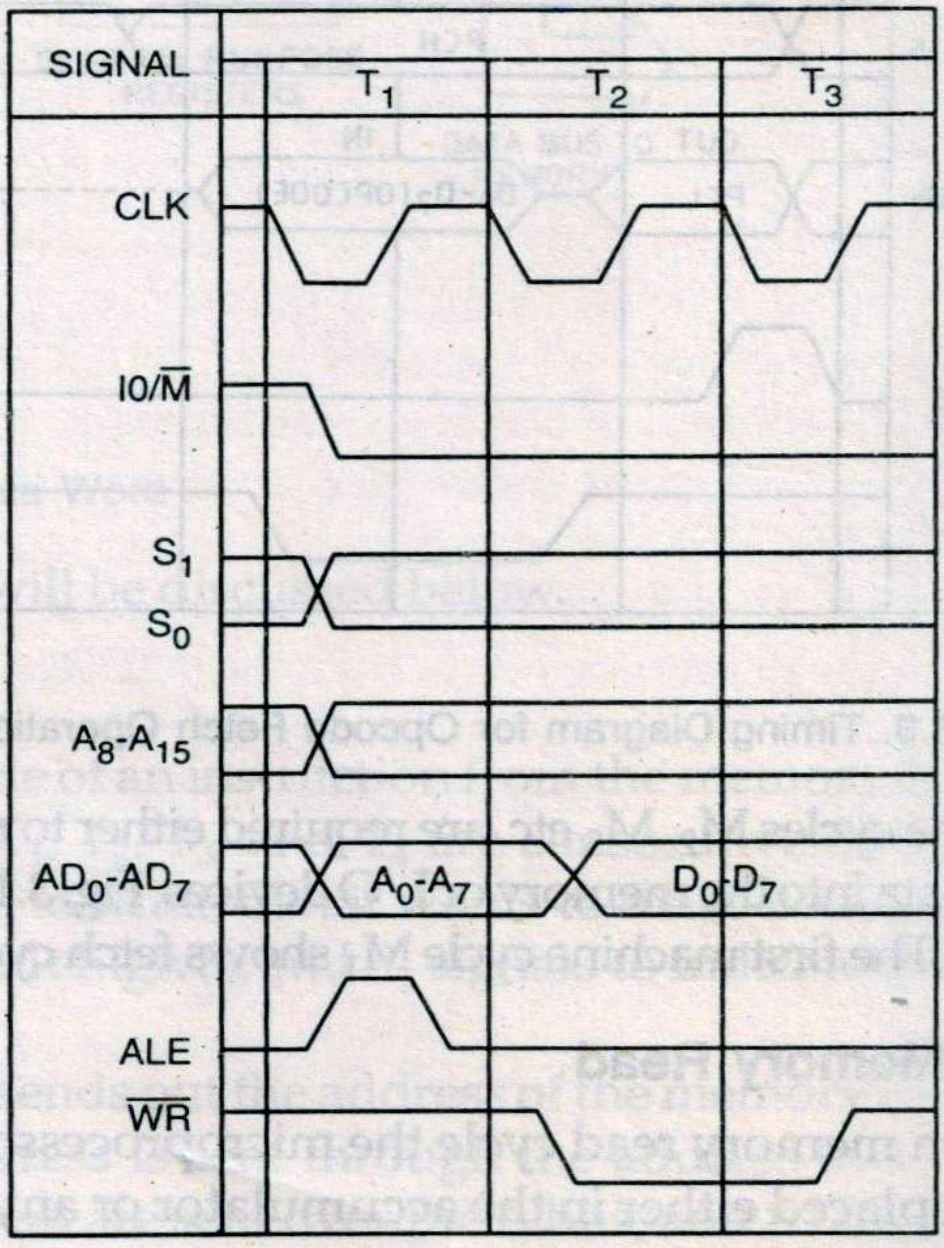


Fig. 3.12. Timing Diagram for Memory Write Operation

TIMING DIAGRAM FOR I/O WRITE CYCLE-

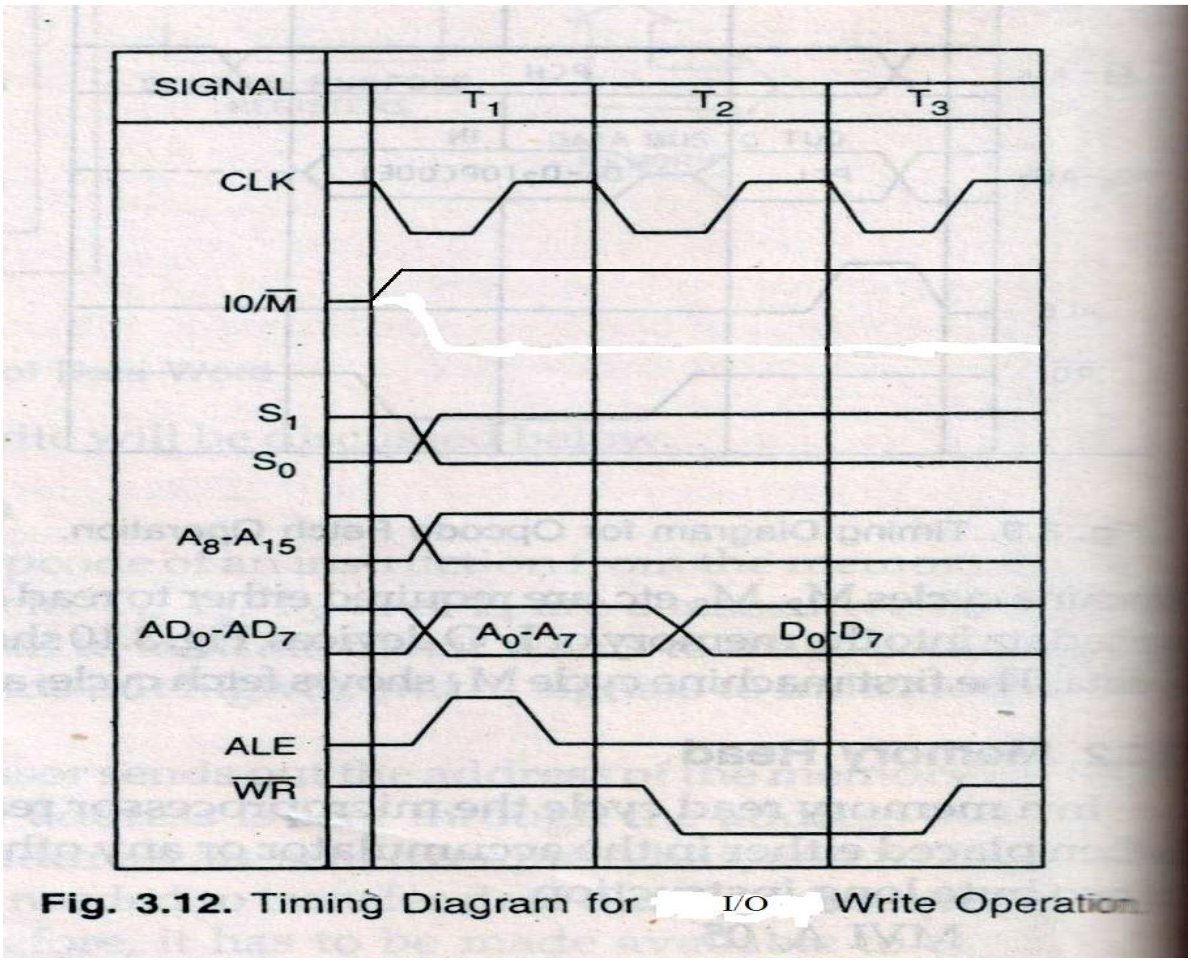


Fig. 3.12. Timing Diagram for I/O Write Operation

1. In an I/O write cycle the CPU sends data to an I/O port or an I/O device from the accumulator.

2. It is similar to a memory write cycle except that IO/$\overline{M}$ goes high indicating that the address sent by CPU is for I/O device or I/O port.

3. The address of an I/O port (8-bit) is duplicated on both address and address/data bus.

4. The OUT instruction is used for I/O write operation. It is a 2 byte instruction and required 3 machine cycle.MC1 is the opcode fetch cycle.MC2 is a memory read cycle for reading the I/O device address from memory.MC3 is an I/O write cycle for saving accumulator data to the I/O device or I/O port.

INTERRUPTS AND INTERRUPT SERVICE SUBROUTINE (ISS):-

HARDWARE INTERRUPTS AND SOFTWARE INTERRUPTS-

Interrupts caused by I/O devices to transfer data to or from the microprocessor are called Hardware Interrupts.

Example- TRAP, RST 7.5, RST 6.5, RST 5.5

When an interrupt is recognized the next instruction is executed from a fixed location in memory as given below

INTERRUPTS	ISS ADDRESS
TRAP	0024
RST 5.5	002C
RST 6.5	0034
RST 7.5	003C

The normal operation of the microprocessor can also be interrupted by abnormal internal conditions on special instruction. Such interrupts are called Software Interrupts.

Example-RST 0 to RST 7 instructions of 8085.

They are used in debugging of a program.

INSTRUCTION	OPCODE	ISS ADDRESS
RST 0	C7	0000
RST 1	CF	0008
RST 2	D7	0010
RST 3	DF	0018
RST 4	E7	0020
RST 5	EF	0028
RST 6	F7	0030
RST 7	FF	0038

The internal abnormal or unusual conditions which prevent the normal processing sequence of a microprocessor are called Exception.

Example-Divide by zero is an exception.

VECTOR INTERRUPTS AND NON VECTOR INTERRUPTS-

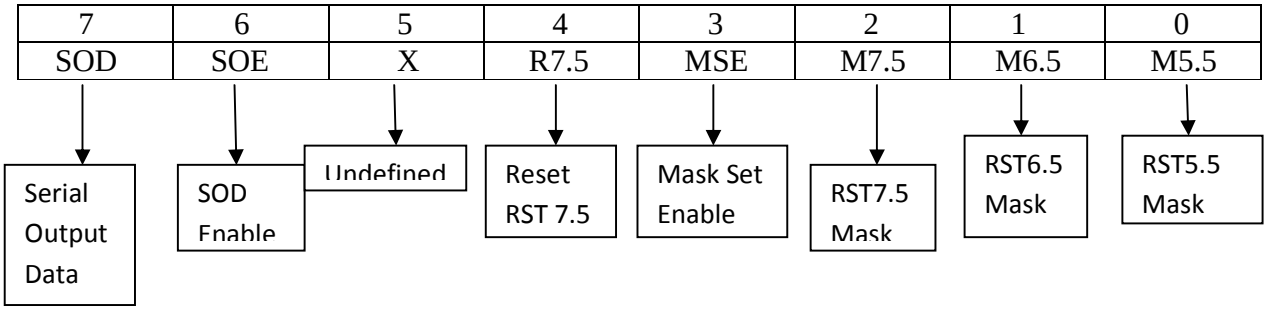
For TRAP, RST 7.5, RST 6.5, RST 5.5 interrupts, the program is automatically transferred to the ISS address without any external hardware .These interrupts for which hardware automatically transfers the program to a specific memory location is known as Vector Interrupt.

When several I/O devices are connected to INTR interrupts line an external hardware is used to interface I/O devices. This circuit generates RST-n codes to implement the multiple interrupts schemes. These are not vector interrupts.

RST 7.5, RST 6.5, RST 5.5-

RST 7.5, RST 6.5, RST 5.5 are maskable interrupts. They are enable by software using instruction EI and SIM. SIM instruction enables or disable according to the bit of accumulator.

ACCUMULATOR CONTENT BEFORE EXECUTION OF SIM-



RST 5.5 mask
RST 6.5 mask
RST 7.5 mask
} bit=1, it is masked off / bit=0, it is enable

Mask Set Enable (MSE) should be set to 1 to make bits 0 to 2 effective.

Q- Write a set of instruction to enable all the interrupts.

Ans-

Label	Mnemonics	Operands	Comments
	EI		Enable interrupts
	MVI	A, O8H	Get accumulator bit pattern to enable RST 7.5, 6.5, 5.5
	SIM		Enable RST 7.5, 6.5, 5.5

Q-Write a set of instruction to enable RST 6.5 and disable RST7.5, RST 5.5.

Ans- Label	Mnemonics	Operands	Comments
	EI		Enable interrupts
	MVI	A, 1DH	Accumulator bit pattern to enable RST 6.5 and masked Off RST 7.5 and 5.5
	SIM		Enable RST 6.5 and disable RST 7.5 and 5.5

PENDING INTERRUPTS:

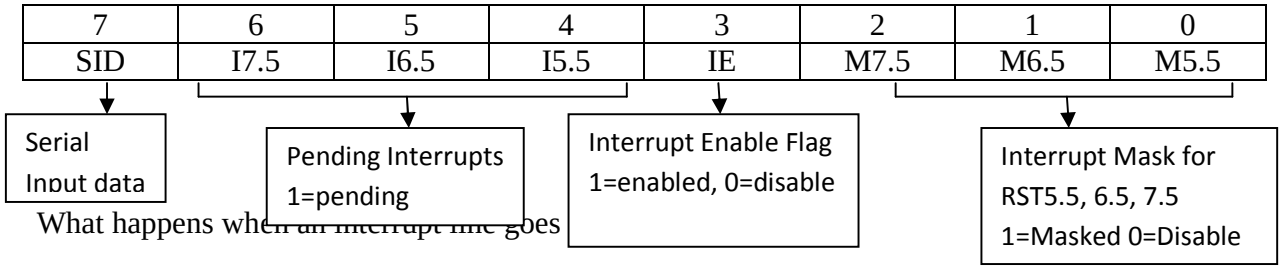
When one interrupt is being served other interrupts may occur resulting in a pending request. When more than one interrupt occur simultaneously the interrupt with higher priority is served and interrupts with lower priority remain pending.

The 8085 has an instruction RIM using which the programmer can know the current status of the pending maskable interrupts.

In case of INTR, the processor needs its status in the last state of the last machine cycle of the instruction.

The bit pattern of the accumulator after execution of RIM instruction is given by;

Bits 0-2 are for interrupt masks.
Bit 3 enable interrupt flag
Bits 4-6 indicate pending interrupts
Bit 7 is SID if any



What happens when an interrupt line goes

1. When an interrupt line goes high, the microprocessor completes its current instruction and saves program counter on the stack.
2. It also resets Interrupt Enable (IE) flip flop before taking up the ISS (Interrupt Service Subroutine) so that the occurrence of further interrupts by other devices is prevented during the execution of ISS.

All the interrupts except TRAP are disabled by disabling the IE flip flop. Before the program returns back from ISS to the main program, all the interrupts are to be enabled again. This is done by using instruction EI in ISS before using the instruction return (RET).

MODULE – 2

ADDRESS SPACE PARTITIONING

Intel 8085 uses a 16-bit wide address bus or addressing memory and I/O devices. It can access 2^{16}=64k bytes of memory and I/O devices. There are two schemes for the allocation of address to memories or I/O devices.

1. Memory mapped I/O scheme
2. I/O mapped I/O scheme

1. Memory Mapped I/O Scheme

In this scheme there is only one address space. Address space is defined as set of all possible addresses that a microprocessor can generate. Some address are assigned to memories and some address to I/O devices. Suppose memory locations are assigned the address 2000-2500. One address is assigned to each memory location. These addresses cannot be assigned to I/O devices. The addresses assigned to I/O devices are different from address assigned to memory. For example, 2500, 2501, 2502 etc. may be assigned to I/O devices. One address is assigned to each I/O device.

In this scheme all the data transfer instruction of the microprocessor can be used for both memory as well as I/O devices. For example, MOV A, M will be valid for data transfer from the memory location or I/O device whose address is in H-L pair. This scheme is suitable for small system.

2. I/O Mapped I/O Scheme

In this scheme the address are assigned to memory locations can also be assigned to I/O devices. To distinguish whether the address on an address bus is for memory location or I/O devices. The Intel 8085 issues IO/M…… signal for this purpose. When the signal is high the address of an address bus is for I/O device. When low, the address is for a memory location. Two extra instructions IN and OUT are used to address I/O device. The IN instruction is used to read data from an input device. And OUT instruction is used to an output device. This scheme is suitable for large system.

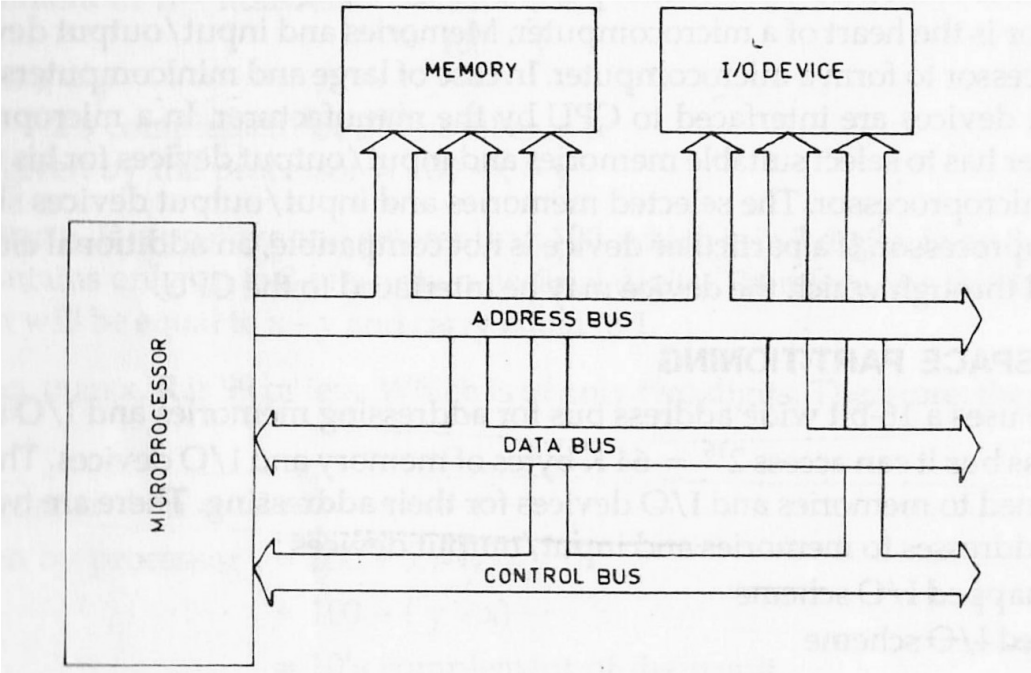


Fig. 7.1 Schematic Diagram for Memory and I/O Interfacing.

Memory Interfacing

An address decoding circuit is employed to select the required I/O device or a memory chip. When IO/M……. is high, decoder is to active and the require IO device is selected. If IO/M….. is low, the decoder1 is activated the required memory chip is selected. A few msb of address line is applied to the decoder to select the memory chip or an I/O device.

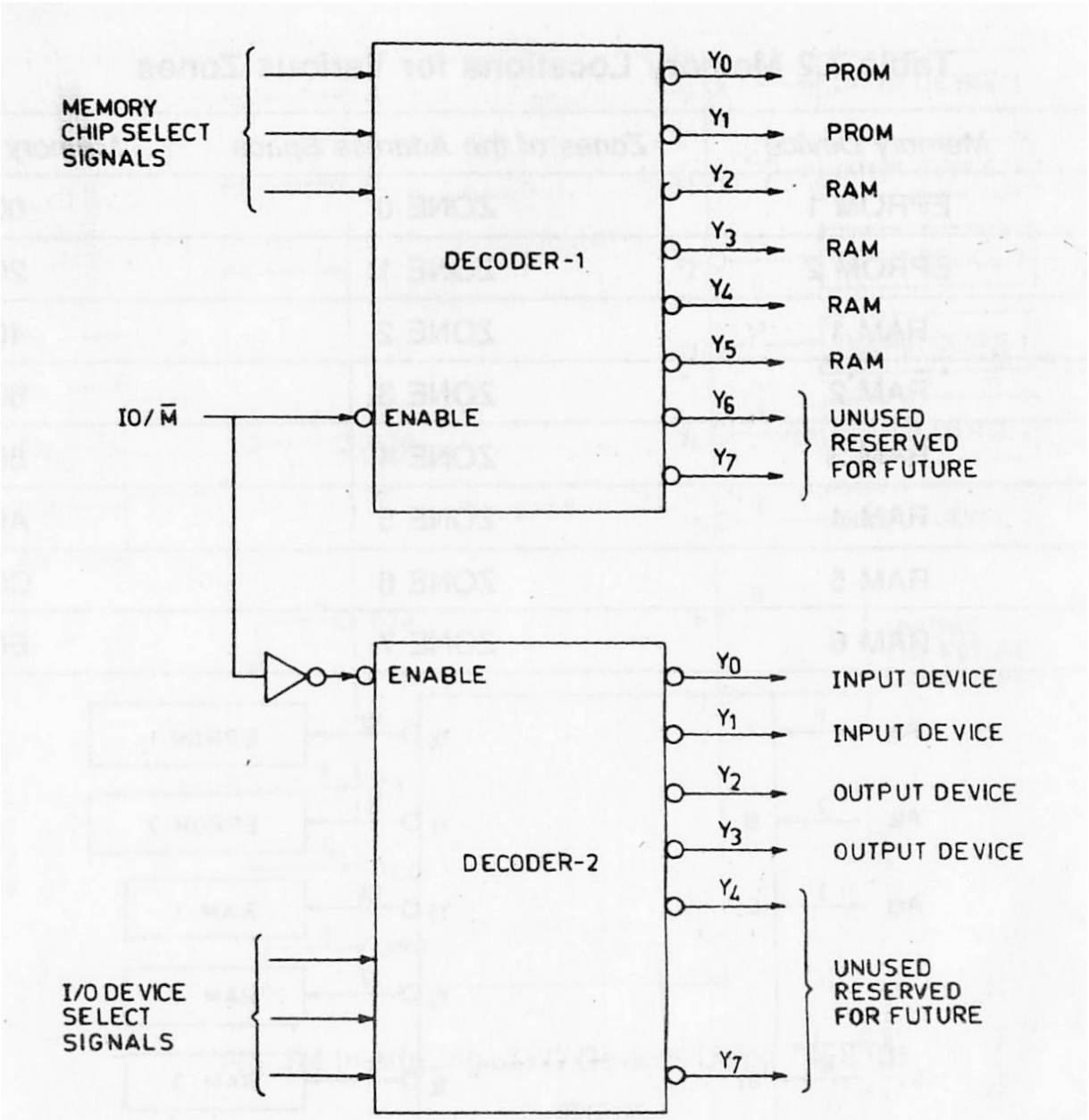


Fig. 7.2 Interfacing of Memory and I/O Devices.

A_{13}	A_{14}	A_{15}	OUTPUT
0	0	0	Y_0
0	0	1	Y_1
0	1	0	Y_2
0	1	1	Y_3
1	0	0	Y_4
1	0	1	Y_5
1	1	0	Y_6
1	1	1	Y_7

MEMORY CLASSIFICATION

The memory is used to store information used by the CPU. The information may be in the form of program, that the CPU executes or data on which the CPU operates. Memory can be classified into two groups.

1- Primary Memory
2- Secondary Memory

Primary memory is constituted by memory blocks within the address space of the processor. They are implemented by using read only memory(ROM) which are not volatile memories and read write memories or random access memory(RAM) which are volatile memories. ROMs are used to store the permanent programs and data while RAMs are used to store intermediate results and data. Secondary memories such as magnetic tapes, floppy disk, hard disk etc are used to store large amounts of data. The CPU cannot directly access secondary units.

1-Primary Memory

It is that part of the memory which can be directly access by the CPU. It can be viewed as a stack of words, each word be associated with a unique address. These words may be instruction or data. A CPU having n address line can access 2^n memory location. The total primary memory area is partition into two separate sections called the ROM area and the read/write area.

ROM Area

On power on or a reset, the CPU always starts executing program from a fixed default address which is usually the first address of the address space. These instructions are usually initialize the sequences which direct the CPU to initialize all sub system in the systems. For example CPU may initialize the display driver display outputs in a particular manner or initialize a key-board to accept the certain commands. This initialization sequence which has to perform each time computer begins to its operation is part of the monitor software which is stored in the permanent or non-volatile memory. The words permanent or non-volatile implies that the program is not lost when the power is switched off and that it is available each time the power is switched on. It is implemented by using a special kind of memory called read only memory or ROM. The data in this selection cannot be written over and can only the read. The ROM is used to store information that should not change. ROMs are available into 4 types. There are

i. Masked ROM
ii. PROM
iii. EPROM
iv. EEPROM

i-Masked ROM

The instructions in such ROMs are permanently installed by the manufacturer as for the specification provided by the system programmer and cannot be altered. This ROM contains call arrays in which 1s and 0s …….. are stored by means of a metallization interconnect most …… at the time of fabrication.

ii-PROM (Programmable Read Only Memory)

The manufacturer provides a memory device which can be programmed by the user by using a PROM program. The PROM uses a fusible links that can be burnt or melted by special PROM burning circuit. A fused link is corresponds to zero.

iii-EPROM (Erasable PROM)

It uses most charge storage technology. It is also programmable by the user. The information stored in the EPROM can be erased by exposing the memory to ultraviolet light which erase the data stored the data in total memory area. Then the memory can be reprogrammable by the user by using EPROM burning circuit.

iv-EEPROM (Electrically Erasable PROM)

This is similar to EPROM except that the erasing done by electrical signal instead of ultraviolet light and that the data in memory location can be selective erased.

Read/Write Area

It is a special kind of memory area where information can be written into or read from whenever necessary. The CPU uses this section memory as a scratch pad memory. It is also called read/write memory or RAM. The CPU can access any memory location by specifying its address. Unlike ROM the conventional is a volatile memory i.e. the contents of the RAM are used when the power is switched off. RAMs are two types.

i-Static RAM

In the case of static RAM once the data is written into a memory location, the data remain unchanged unless on same memory location is written into again. It uses flip-flops for storage elements.

ii-Dynamic RAM

In case of dynamic RAM the basic storage elements is a capacitor. This element contains a 1or a 0 depending on the presence or absence of charge. Unlike static RAM, the contents of the dynamic RAM may change with time due to leakage of charge. So it is necessary to periodically refresh the storage element in a dynamic RAM. Here it has external refresh circuitry. The advantages of dynamic RAM over static RAM are

… It consumes less power than static RAM.

… It has about 5 times more storage element per unit area.

Disadvantages are that

… DRAMS have slower access times and need special circuitry to periodically refresh memory.

i-RAMs:

RAMs are also implemented using other technologies like i-RAM (integrated RAM) which are dynamic RAM devices in which the memory refresh circuitry is implemented within the device.

Bubble Memories

It is a type of DRAM. Here the contents are not lost when the power is switched off.

2-Secondary Memory

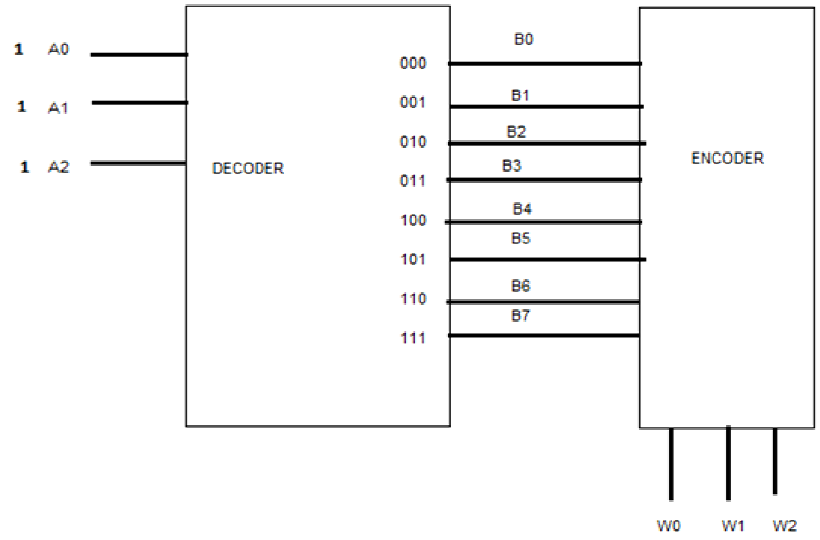
The maximum capacity of primary memory 2^n location. Each of 2^n location where n is the number of CPU address lines. Sometimes it is necessary to handle more data than allowed by the primary memory. In such cases secondary memory is used. The CPU cannot directly

access memory but can access through I/O ports. Examples are magnetic tapes, hard disk, and floppy disk.

MEMORY STRUCTURE:

ROM MODEL:

A read only memory (ROM) is an encoder to select a word in an encoder, only one of the inputs must be made active. But to save lines, the CPU directly puts out the address of the word it wants to access. So it is necessary to insert a device between the address put out by the CPU and the ROM inputs which enables a unique input to the ROM. This device must be a decoder.



(Decoded ROM of capacity 8 bytes)

This is an example with an address bus of n=3 bits and a memory of 8 bytes (2^3=8 words).

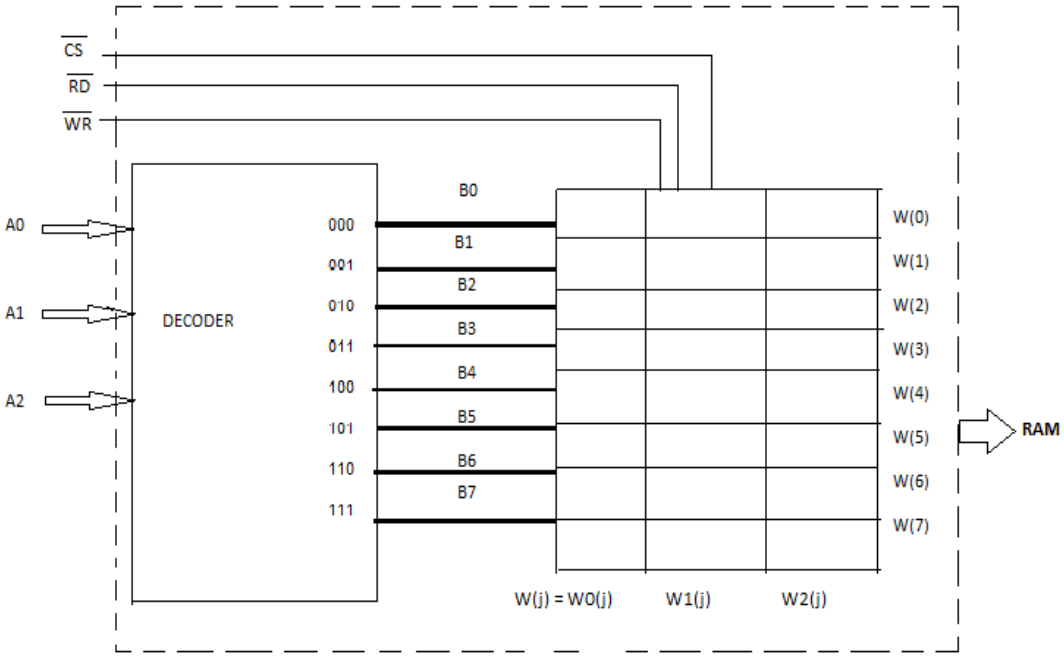
The CPU puts out the address 111 on the address lines $A_2A_1A_0$ to access the 8^{th} word B_7. B_7 is made active by the decoder and the word W (7) = W_2 (7) W_1 (7) W_0 (7) is put on the data bus.

RAM or READ/WRITE MODEL:

The internal organization of a random access memory (RAM) is similar to that of ROM. But RAM has input which is made active when data is to be written into RAM.

The figure shows a 3 bit = 8 words RAM.

One of the outputs of the decoder is made active depending on the address which is input to it. This active output is fed to the encoder to generate the output W, which is directly connected to the common data bus of the system.



(RAM or R/W MODEL)

MEMORY INTERFACING & ADDRESSING DECODING:

This is done by using 74LS138 which is a 3-8 line decoder.

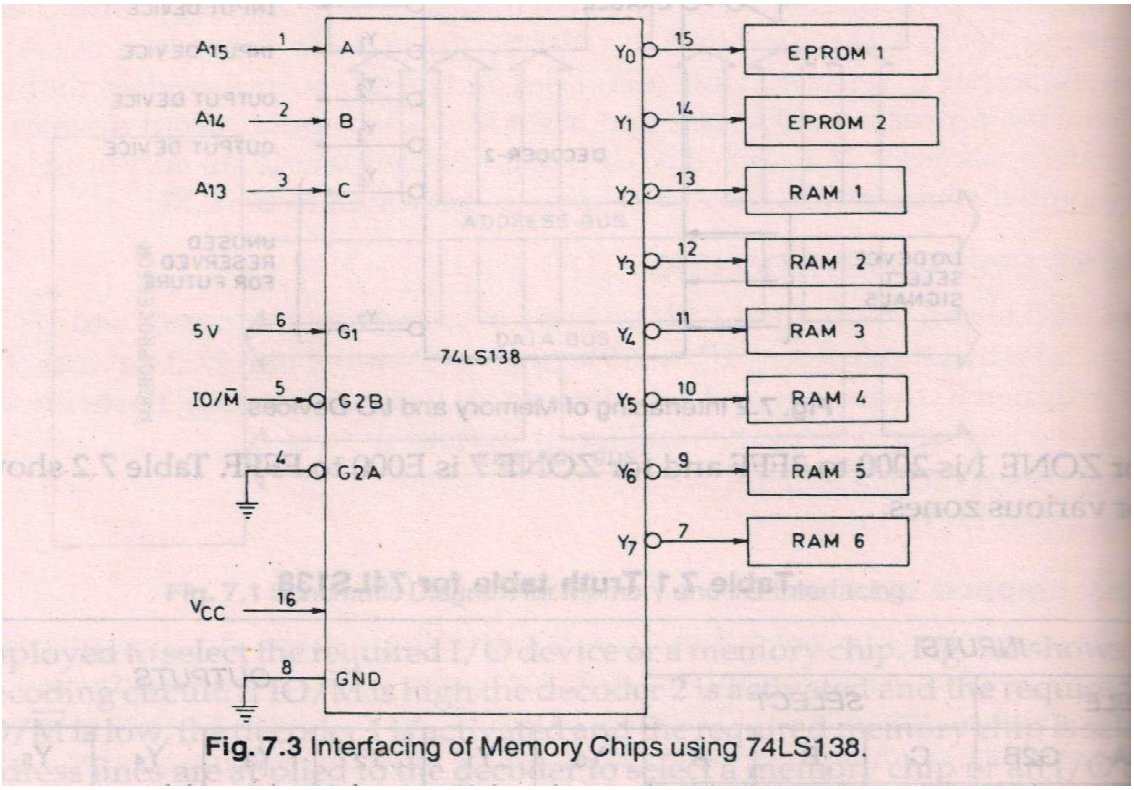


Fig. 7.3 Interfacing of Memory Chips using 74LS138.

(Memory interfacing using 74LS138)

The address of the memory location is sent out by the microprocessor. The corresponding memory chip is selected by a decoding circuit.

G1, G2A, G2B are enable signals of the decoder. To enable 74LS138 decoder, G1 should be high and G2A, G2B should be low. A, B, C are the select lines of the decoder. Y_0-Y_7 are the 8 output lines.

By applying proper logic to select lines, any one of the outputs can be selected. The selected output line goes low while other output lines remain high.

The entire memory address = 64 KB for 8085 has been divided into 8 zones. The address lines A_{13} A_{14} A_{15} have been applied to A, B, C select lines. The logic applied to these select lines A, B, C selects a particular memory device i.e. an EPROM or RAM. The address lines A_0 to A_{12} decide the address of the memory location within a selected memory chip.

IO/ is connected to G2B, G1 is connected to +5volt dc supply and G2A is grounded.

Addresses of 8 zones:

Decoder output	Memory device	Zones of the address spaces	Memory location address
Y_0	EPROM1	ZONE 0	0000 to 1FFF
Y_1	EPROM2	ZONE 1	2000 to 3FFF
Y_2	RAM1	ZONE 2	4000 to 5FFF
Y_3	RAM2	ZONE 3	6000 to 7FFF
Y_4	RAM3	ZONE 4	8000 to 9FFF
Y_5	RAM4	ZONE 5	A000 to BFFF
Y_6	RAM5	ZONE 6	C000 to DFFF
Y_7	RAM6	ZONE 7	E000 to FFFF

Total memory = 0000 to FFFF = 64 KB

Memory of each zone = 64/8 KB = 8KB

8085A Minimum System Microcomputer

An 8085 a based minimum system microcomputer needs the following components

- CPU
- MEMORY
- IO PORTS

*The CPU co-ordinates the activities job all the components on a microcomputer.

* On a reseting or after a power on the CPU executes the programs stored in the permanent memory starting from the first address in the address space.

* The data for program execution can come from memory or from IO ports.

* A scratchpad memory in the RAM area is essential to store intermediate data and also to function as a stack.

*The IO ports are essential for the CPU to collect data from an environment and sends out appropriate signals to control a processing.

*In order to avoid extra circuit tree and keep the number of components to be minimum intel corporation introduced two special chips.

- 8155/8156 chip
- 8355/8755 chip

Both of these are on chip address de-multiplexing circuits.

8155

*The 8155 is a 40 pin chip having 8 bit 256 word RAM memory

*Two programmable 8 bit IO ports i.e. port A ,port B

*one programmable 6 bit IO port i.e. port C

*one programmable 14 bit binary timer/counter

*An internal address latch

8156

8156 is functionally compatible with 8155except that it uses an active high chip enable signals while 8155 uses the active low chip enable.

8355

The 8355 has a 2 KB ROM memory and 8 bit parallel ports i.e. port A and port B.

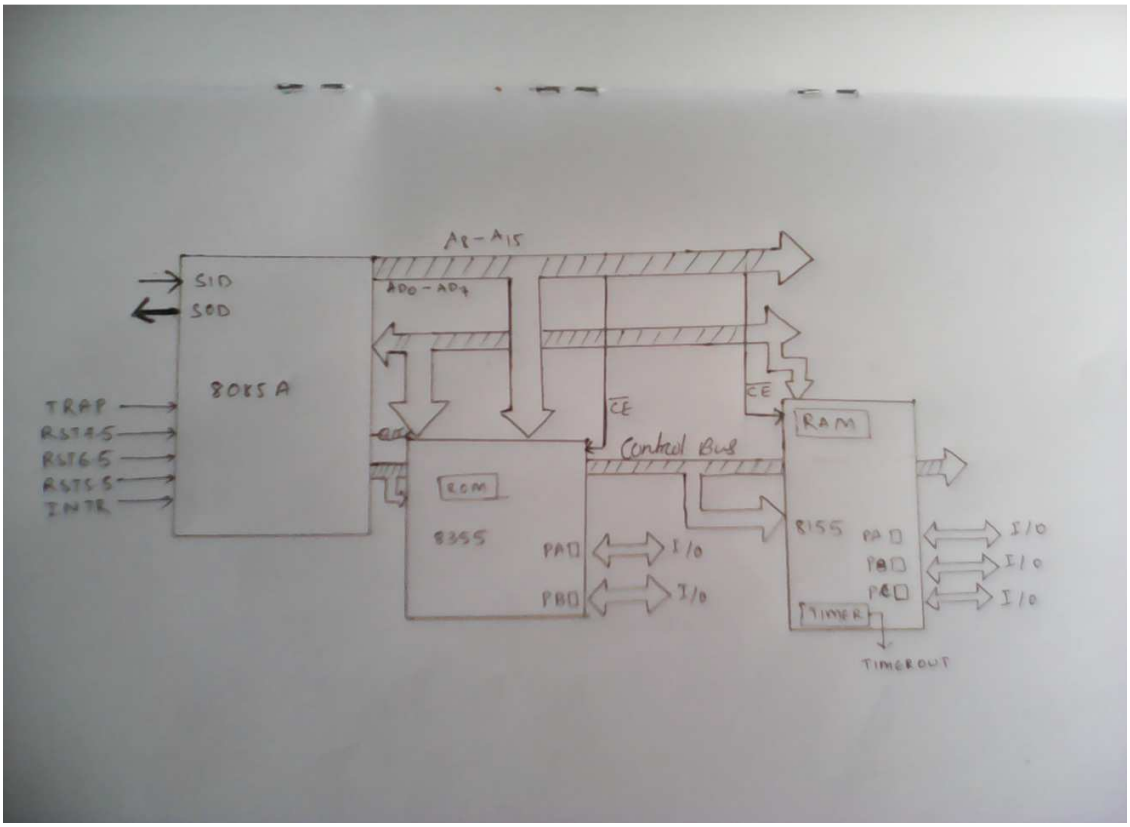
8755

The 8755 has 2 KB of EPROM memory instead of ROM memory and consist of two 8 bit parallel ports i.e. port A and port B.

* 8155/8156 is used as RAM memory.

* 8355/8755 is used as ROM memory.

*8085A is used as CPU.



SCHEMATIC DIAGRAM OF 8085A MINIMUM SYSTEM MICROCOMPUTER

*RST5.5 , RST 6.5 and RST 7.5 are maskable interrupt.

* They are enable by software using instruction EI and SIM.

* SIM instruction enable or disable interrupts according to the bit pattern of the accumulator.

ABSOLUTE VS PARTIAL DECODING

Absolute Decoding	**Partial Decoding**
*When the output port is selected by the decoding all of the 8 address lines it is called absolute decoding.	*When the output port is selected by decoding some of the address lines it is called partial decoding.
*It is good design practice but it is costly.	*It is less costly.

*The output device selected as a unique address.	*The output device selected as multiple addresses.
*It is used in large systems.	*It is used in small systems.

DATA TRANSFER GROUP

Instructions which are used to transfer data from one register to another register, from memory to register or from register to memory they come under this group.

For e.g- MOV r1, r2 – move the content of the one register to another. The content of register r2 is moved to register r1. For example , the instruction MOV A,B moves the content of register B to register A. The instruction MOV B,A moves the content of register A to register B . The time for the execution of this instruction is 4 clock period. One clock period is called state. No flag is affected.

1. MOV r,M (Move the content of the memory to register).

The content of the memory location ,whose address is in HL pair, is moved to register r.

e.g- LXI H,2000H load HL pair by 2000H

MOV B,M Move the content of the memory location 2000H to register B.

HLT Halt.

2. MOV M,r (Move the content of register to memory).

The content of register r is moved to the memory location addressed by HL pair.

3. MVI r,data (move immediate data to register .)

The 1st byte of the instruction is its opcode. The 2nd byte of the instruction is the data which is moved to register r. For example , the instruction MVI A,05 Moves 05 to register A. In the code form it is written as 3E,05. The opcode for MVI A is 3E and 05 is the data which is to be moved to register A.

4. MVI M,data (move immediate data to memory.)

The data is moved to memory location whose address is in HL pair. for example

LXI H,2400H load HL pair with 2400H

MVI M,08 Move 08 to the memory location 2400H

HLT Halt

5. LXI rp,data 16- This instruction loads 16 bit data immediate data into register pair rp. This instruction is for register pair only high order register pair is mentioned after the instruction . for example H in the instruction LXI H ,2500H into HL pair. H with 2500H denotes that the data 2500H denotes that the data 2500 is in hexadecimal. In the code form it is written as 21,00,25. The 1st byte of the instruction 21 is the opcode for LXI H. The 2nd byte 00 is 8 LSBs of the data and it is loaded into the register L. The 3rd byte 25 is 8 MSBs of the data and it is loaded into register H.

6. LDA addr (load accumulator direct).-The content of the memory location, whose address is specified by the 2nd and 3rd bytes of the instruction is loaded into the accumulator. The instruction LDA 2400H will load the content of the memory location 2400H into the accumulator . The 2nd byte 00 is of 8 LSBs of the memory address . The 3rd byte 24 is 8 MSBs of the memory address .

7. STA addr (store accumulator direct).-The content of the accumulator is stored in the memory location whose address is specified by the 2nd and 3rd byte of the instruction. STA 2000H will store the content of the accumulator in the memory location 2000H.

8. LHLD addr (Load HL Pair direct)- The content of the memory location , whose address is specified by the 2nd and 3rd bytes of the instruction ,is loaded into the register L. The content of the next memory location is loaded into the register H.

9. SHLD addr (store HLpair direct) –The content of the register L is stored in the memory location whose address is specified by the 2nd and 3rd bytes of the instruction. The content of the register H is stored in the next memory location 2500H . The content of register H is stored in the memory location 2501H.

10. LDAX rp (LOAD accumulator direct) –The content of the memory location ,whose address is in the register pair rp is loaded into the accumulator.

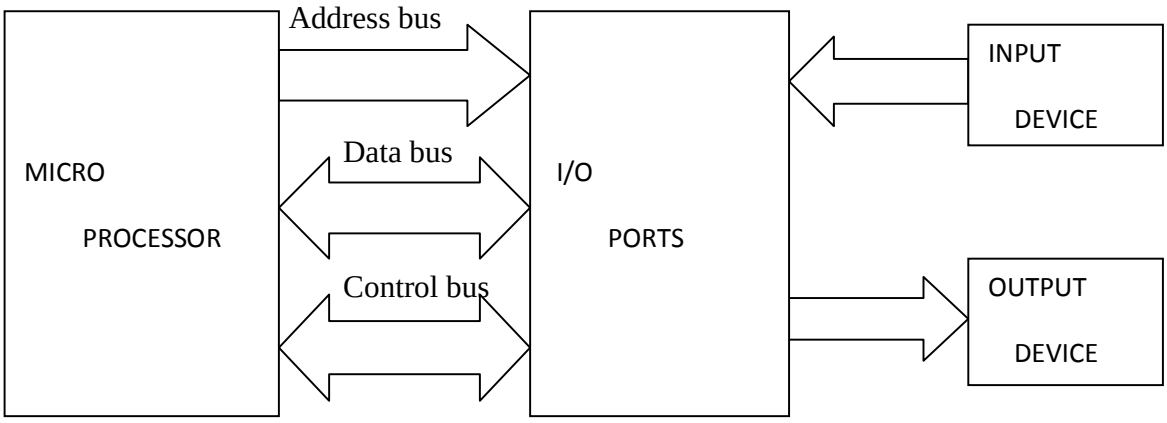
11. STAX rp (store the accumulator direct)- The content of the memory location is stored in the memory location whose address is in the register pair rp .

12. XCHG (Exchange the contents of HL with DE pair)-The contents of HL pair are exchanged with contents of DE pair.

MODULE – 3

I/O PORTS – 8212 :-

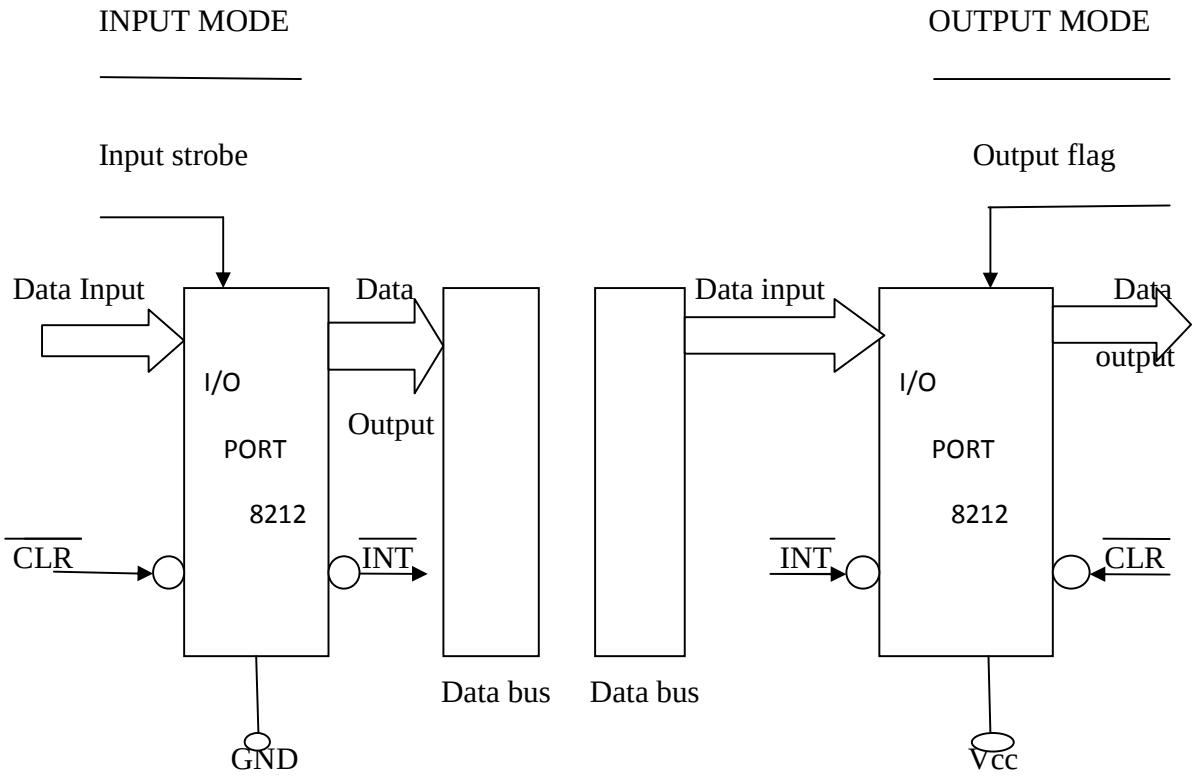
- An input device is connected to the micro processor through an input port.
- An input device is a place for unloading data. An input device unloads data into the port, & the micro processor reads data from the input port.
- Similarly, an output device is connected to the micro processor through an output port.



INTERFACING OF I/O DEVICE THROUGH I/O PORT

- An I/O port may be programmable or non-programmable.
- A non-programmable I/O port behaves as an input port, if it has been designed and connected in output mode. A port connected in output mode acts as an output port.
- But a programmable I/O port can be programmed to act either as an input port or an I/O port.

The Intel 8212 is an 8-bit non-programmable I/O port. It can be connected to the micro processor either as an input port or as an output port. If we required one input port & one output port, two units of 8212 will be required .one of them will be connected in input mode & the other in output mode.



8255A (PPI)

PPI-Programmable Peripheral Interface:

A PPI is a multiport device. The port may be programmed in a verity of ways as required by the programmer. The device is very useful for interfacing peripheral devices.

It has three, 8-bit port.

- Port A
- Port B
- Port C

Port C is further subdivided into two, 4-bit ports.

- Port C upper
- Port C lower

Each port can be programmed either as an input port or as an output port,by setting proper bits in the control word.

This control word is written into a control word register (CWR).

CONTROL GROUPS OF 8255:

The 24 lines of I/O port is divided into 2 groups.

- Group A
- Group B

Each group contains one 8-bit port & a 4-bit port.

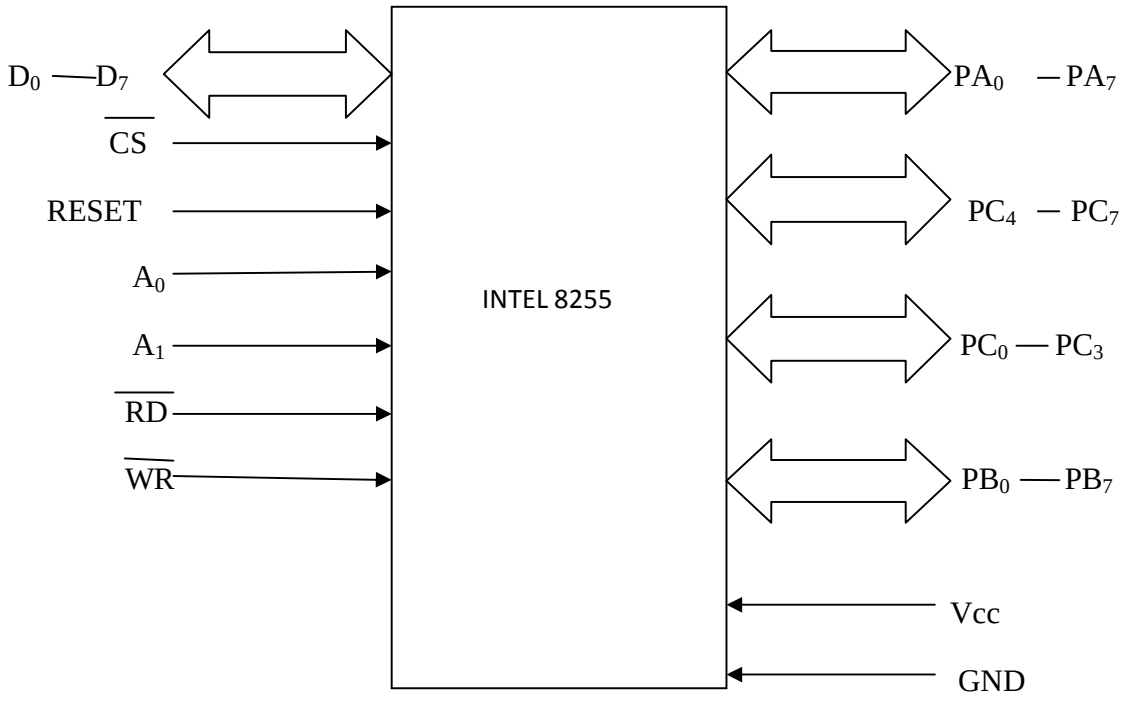
Group A	**Group B**
Port A	Port B
Port C upper	Port C lower

The control word initialises the ports.

ARCHITECTURE OF INTEL 8255A :

- It is a 40 pin IC package.
- It operates on a single +5V dc supply.

SCHEMATIC DIAGRAM OF INTEL 8255A:



PC4 – PC7 - Port C upper

PC0 – PC3 ⇨ Port C lower

The pins for various ports are:-

- $PA_0 – PA_7$ → 8 pins of Port A
- $PB_0 – PB_7$ → 8 pins of Port B
- $PC0 – PC_3$ → 4 pins of Port C lower
- $PC_4 – PC_7$ → 4 pins of Port C upper

The important control signals are:

D0 – D7 → These are 8-bit bidirectional data lines.

$\overline{CS}$ (Chip Select) → It is a chip select signal. The low status of this signal enable communication between CPU & 8255.

$\overline{RD}$ (Read) → It allows the CPU to read data from the input port of 8255.

$\overline{WR}$ (Write) → when WR goes low, the CPU writes data or control word into 8255.

.

The CPU writes data into the output port of 8255 & the control word into the control word register (CWR).

RESET → It is used to reset the 8255 chip.

A_0, A_1 → The selection of I/O port & CWR is done using A_0 & A_1 in Conjunction with RD & WR . A_0 & A_1 are normally connected to the LSBs of the address bus. They are used for addressing the 4 register, any one of ports and a i.e. 3 I/O CWR.

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A1	A2	Function
X	X	1	X	X	Data bus is tristated
1	1	0	X	X	Data bus is tristated

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A1	A0	Output (write) cycle
1	0	0	0	0	Data bus to port A
1	0	0	0	1	Data bus to port B
1	0	0	1	0	Data bus to port C
1	0	0	1	1	Data bus to CWR

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A1	A0	Input (read) cycle
0	1	0	0	0	Port A to data bus
0	1	0	0	1	Port B to data bus
0	1	0	1	0	Port C to data bus
0	1	0	1	1	CWR to data bus

X=don't care, as $\overline{CS}$ is 1

As $\overline{RD}$, $\overline{WR}$=1 so even if $\overline{CS}$ =0, A_1, A_0=X

Mode of operation of 8255:

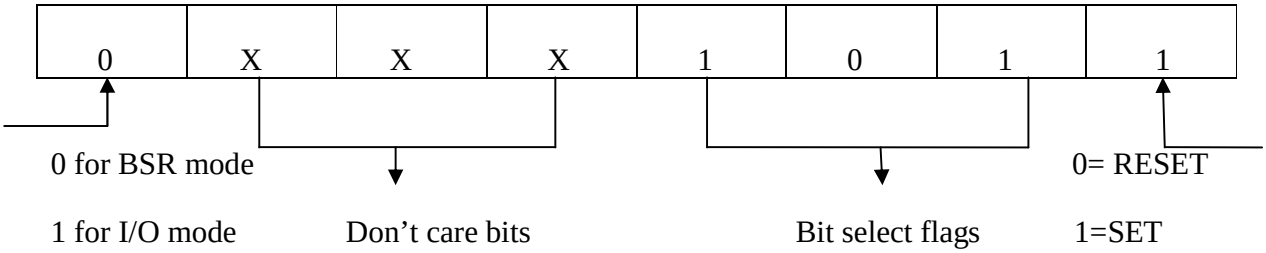
There are 2 basic modes of operation of 8255.

- **BSR Mode (Bit Set Reset mode)**
- **I/O Mode (Input / Output mode)**

BSR Mode:

In this mode, any of the 8-bits of Port C can be set or reset depending on bit B0 of the control word. The bit to set or reset is selected by bit select flag B3, B2, B1 of the CWR.

BSR mode control word register format:



B3	B2	B1	Selected bits for port C
0	0	0	B0

0	0	1	B1
0	1	0	B2
0	1	1	B3
1	0	0	B4
1	0	1	B5
1	1	0	B6
1	1	1	B7

I/O Mode:

In this mode,the 8255 works as programmable I/O ports.

There are three modes of operation of 8255 under I/O mode.

i.e. Mode 0

Mode 1

Mode 2

Mode 0 – Simple input / output

Mode 1 – Strobed input / output

Mode 2 – Bidirectional input / output

All these modes can be selected by programming a register internal to 8255 known as control word register (CWR), which has two formats – one for BSR mode and another for I/O mode.

Mode 0:

- It is also known as basic Input / Output mode.
- This mode provide simple Input / Output capability using each of the three ports.

Features of Mode 0:--

- Two, 8-bit ports i.e. port A and port B & two,4-bit ports i.e. port C upper and port C lower are avaliable. The two 4-bit ports of port C can be combinedly used as a 3rd 8-bit port.
- Any port can be used as an input or output port.
- Output ports are latched. But, input ports are not latched.
- A maximum of 4 ports are avaliable. So that 16 I/O configurations are possible.

Mode 1 :

This mode is also called strobed I/O mode. In this mode port C is used for generating hand shake. Signals to control the input or output action of port A or port B.

Features of Mode 1:

- Two groups i.e. group A and group B are avaliable for strobe data transfer. Each groups contains one 8-bit data I/O port, one 4-bit control port.
- The 8-bit data port can be either used as input or output port.
- Both the inputs and outputs are latched. Out of 8-bit port C i.e. PC0 to PC2 are used to generate control signals for port B and PC3 to PC5 are used to generate control signals for port A. The line PC6 and PC7 may be used as independent data lines.

Mode 2:

It is known as strobe bidirectional I/O mode.

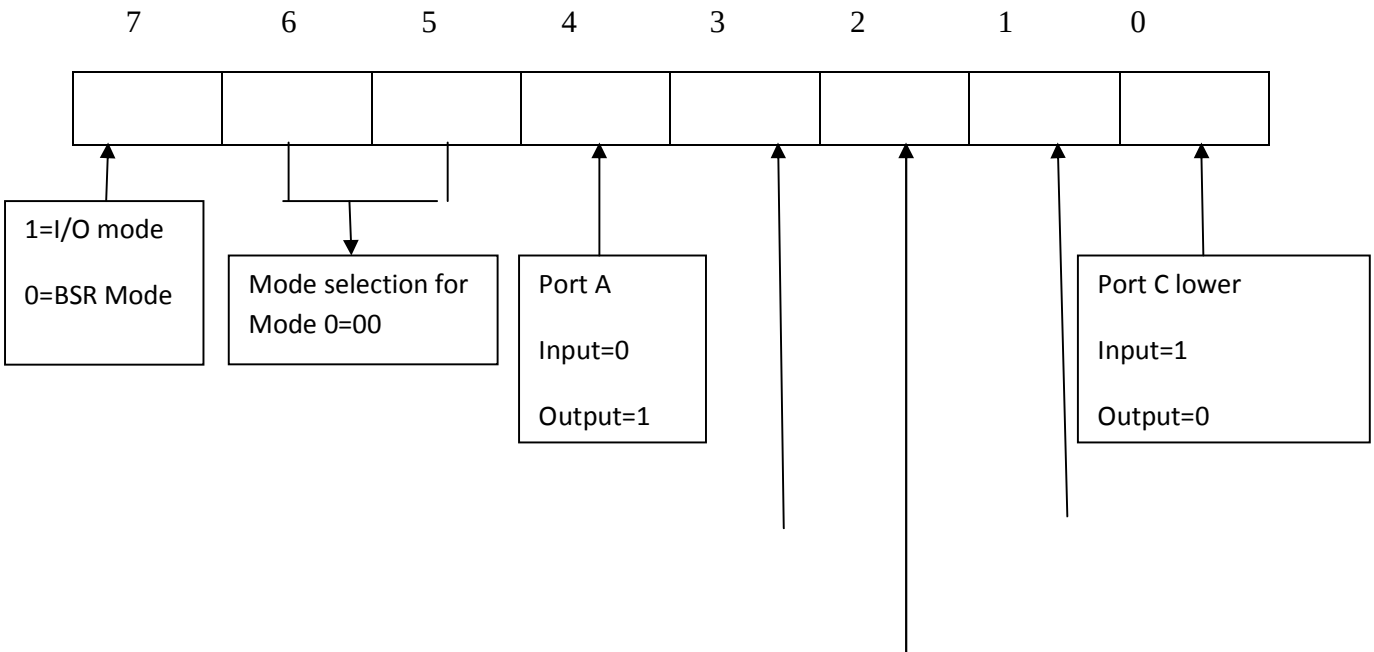
Here only port A is available. In this mode 8255 acts as a bidirectional 8-bit port with handshakes signal.

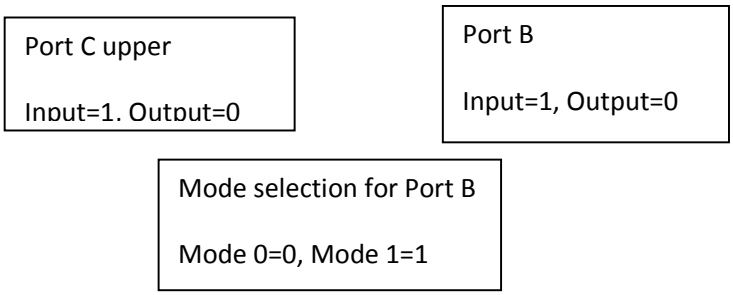
Features of mode 2:

- Here only port A is available.
- The 8-bit port A is bidirectional and additionally a 5-bit control port C is available.
- Input and outputs are both latched.
- Three I/O lines are available at port C i.e. PC0 to PC2.
- The 5-bit control port C i.e. PC3 to PC7 is used for generating handshake signals for 8-bit data transfer on port A.

Control Word format in I/O mode:

For programming the ports of 8255 a control word is formed. The CPU outputs the control word to 8255 which is written into the control word register in 8255.No read operation of control word register is allowed.




Port C upper
Input=1, Output=0
Port B
Input=1, Output=0
Mode selection for Port B
Mode 0=0, Mode 1=1

DMA CONTROLLER (Intel 8257/ 8237):

- Intel 8257 is a four channel DMA controller used to facilitted the direct memory access or DMA mode of data transfer.
- In this mode the devices transfers data directly to or from memory with any interference on the CPU.
- For facilitating DMA data transfer between several devices a DMA controller is used.

INTERNAL ARCHITEHCHTURE OF 8257:

- The chip supports four DMA channel i.e four peripheral devices can independently request for DMA data transfer through these channel at a time.
- The DMA controller has 8-bit internal data buffer ,a read/write a control unit ,a priority reserving unit along with a set of register.

REGISTER ORGANISATION OF 8257:

- Each of the four channels of 8257 has a pair of two 16-bit registers.
  i.e DMA register and terminal count register.
- There are two common register for all the channel.
  (a) Mode set register
  (b) Status register

DMA address register:

- Each DMA channel has DMA address register . Its function is to store the address of the starting memoy location which will be access by the DMA channel.

Terminal count register:

- This 16-bit register is used to make sure that the data transfer through a DMA channel stops after the required number of DMA cycles.
- The load of 14-bit terminal count register are initialize with the binary equivalent of the number of DMA cycle minus one.
- After each DMA cycle the terminal count register contained will be decremented by 1 and finally it becomes 0 after the required number of DMA cycle.
- The bits 14 and 15 of this register indicate the type of the DMA operations
- There are three type of DMA cycle.
  (a) Read DMA cycle
  (b) Write DMA cycle
  (c) Verify DMA cycle

Bits 15	Bits 14	Types of DMA operation
0	0	Verify DMA cycle
0	1	Write DMA cycle
1	0	Read DMA cycle
1	1	(Illegal)

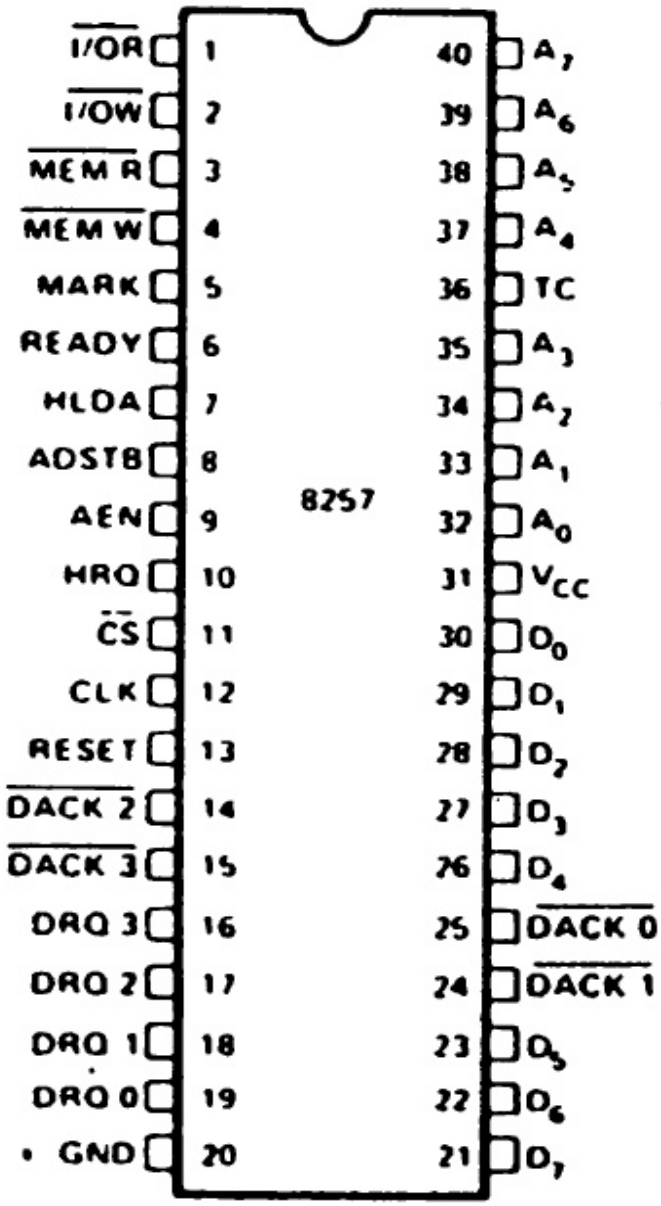
MODE SET REGISTER:

The function of this register is to enable the DMA channel indivisually and also to set the various modes of operation

- The bits D_0 –D_3 enable one of the four DMA channel of 8257.
  For example : If D_0 is 1,channel0 is enable.
  If D_4 is set rotating priority is enable
  otherwise fixed priority is enable.
- If Bit D_6 (TC) is set the selected channel is disable after terminal count register is least and it prevents any further DMA cycle on the channel.
- The auto laod bit that is D_7 if set enable channel for the repeat block chaining operations.
- After the first block is transfer using DMA the channel 2 register are reloaded with corresponding channel 3 register for the next block transfer if the update flag is set.
- Extended write bit D_5 is set to 1extends the duration of memory write ($\overline{\text{MEMW}}$) and I/O write signals $\overline{\overline{\text{IOW}}}$.

STATUS REGISTER:

- Bits D_0-D_3 contain the terminal count status for the four indivisual channels.
- If any of these is set it indicates that the specific channel is reached the terminal count register.
- Bit D_4 repreasents the update flag if it is set the contentsof channel 3 registerare reloaded with corresponding channel 2 registers without any software intervention.



(Pin diagram of 8257)

SIGNAL DESCRIPTION OF 8257

DRQ_0-DRQ_3 :

- These are the 4 individual channels DMA request inputs used by the peripheral devices for requesting DMA services.
- The DRQ_0 has the highest priority ,while DRQ_3 has the lowest,if fixed priority mode is selected.

$\overline{DACK}_0$ - $\overline{DACK}_3$:

- These are the active low DMA acknowledgement output lines which inform the requesting peripheral that the DMA request has been honoured.

D_0-D_7 :

- These are bidirectional data lines used to interface the system bus with the internal data bus of 8257.

$\overline{IOR}$:

- In Master mode ,this signal is used to read data from a peripheral during a memory write cycle.
- In slave mode, this input signal is used by the CPU to read internal registers of 8257.

$\overline{IOW}$:

- In Master mode , it is a control output that is used to write data to a peripheral during DMA memory read cycle
- In slave mode ,it is used to load the contents of the data bus to the 8 bit mode register.

CLK :

- This is a clock frequency input required to generate system timing for internal operations of 8257.

RESET :

- This input disables all the DMA channels by clearing the mode registers & translate all the control lines.

A_0-A_7 :

- These are address lines.

CS :

- It is an active low chip select line that enables R/W operations from or to 8257 in slave mode.
- In master mode it is disabled.

READY:

- This input is used to stretch memory read & write cycles of 8257 by inserting wait states.

HRQ :

- Hold Request output requests access of system bus of 8085 CPU.

HLDA :

- This input if high indicates to the DMA controller that the system bus has been generated to the requested peripheral to the CPU.

$\overline{\text{MEMR}}$:

- This active low memory read output is used to read data from the addressed memory location during DMA read cycle.

$\overline{\text{MEMW}}$:

- This active low memory write output is used to write data to the addressed memory location during DMA write operation.

ADSTB (Adress strobe) :

- This output from 8257 strobe the higher byte of the memory address generated by the DMA controller into the latches.

AEN (Adress Enble) :

- This output is used to disable the system address & data to stop the non DMA devices from responding during DMA operations.

TC (Terminal Count) :

- This output indicates to the currently selected peripheral that the present DMA cycle is the last for the previously programmed data bus.

MARK :

- The modulo 128 mark output indicates the selected peripheral that the current DMA cycle is the 128^{th} cycle since the previous marked output.

PRIORITY OF DMA REQUESTS :

The priority resolver resolves the priority of the 4 DMA channels.

There are 2 schemes :-

1. Fixed priority scheme
2. Rotating priority scheme

1.**Fixed Priority Scheme :**

In Fixed priority scheme ,each device connected to a channel is assigned a fixed priority.

DRQ_0 has the highest priorty
DRQ_1
DRQ_2
DRQ_3 has the lowest priority.

2.**Rotating Priority Scheme :**

In Rotating Priority scheme ,the priority assigned to the channels are not fixed.
A channel that gets service becomes the lowest priority channel . This avoids the dominance of any one channel. For example, after channel 0 is served, it becomes the lowest priority channel.

INTERFACING A DMA CONTROLLER WITH A SYSTEM :

The DMA controller interfacing circuit implements a switching arrangement for the address,data & control buses of the memory & peripheral sub-system from or to the CPU To or from the DMA controller.

1. The peripheral device that wants to do a DMA data transfer sends a request to DMA controller on DRQ line.
2. The DMA controller then sends a HOLD request to CPU on HRQ line to get control of system bus.
3. The CPU completes the current machine cycle,it releases the system bus to the DMA controller and sends a high HLDA signal.
4. The controller then sends a DMA acknowledgement $\overline{\text{DACK}}$ to the requesting peripheral.Now it is ready for DMA data transfer.
5. During DMA data transfer CPU remains in HOLD state.

PROGRAMABLE INTERRUPT CONTROLLER (PIC) 8259A

A PIC takes care of a number of simultaneously appearing interrupt requests along with their types and priorities in a multiple interrupt system.

ARCHITECTURE OF 8259A

- **INTERRUPT REQUEST REGISTER (IRR):**
  IRR stores all the interrupt request in it in order to solve them one by one on priority basis.

- **IN – SERVICE REGISTER (ISR):**
  It stores all the interrupt request that are being served.

- **PRIORITY RESOLVER:**
  This unit determines the priorities of the interrupt request appearing simultaneously. (In fixed priority mode IR0 has the highest priority while IR7 has the lowest)

- **INTERRUPT MASK REGISTER (IMR):**
  The register stores the bits required to mask the interrupt inputs.

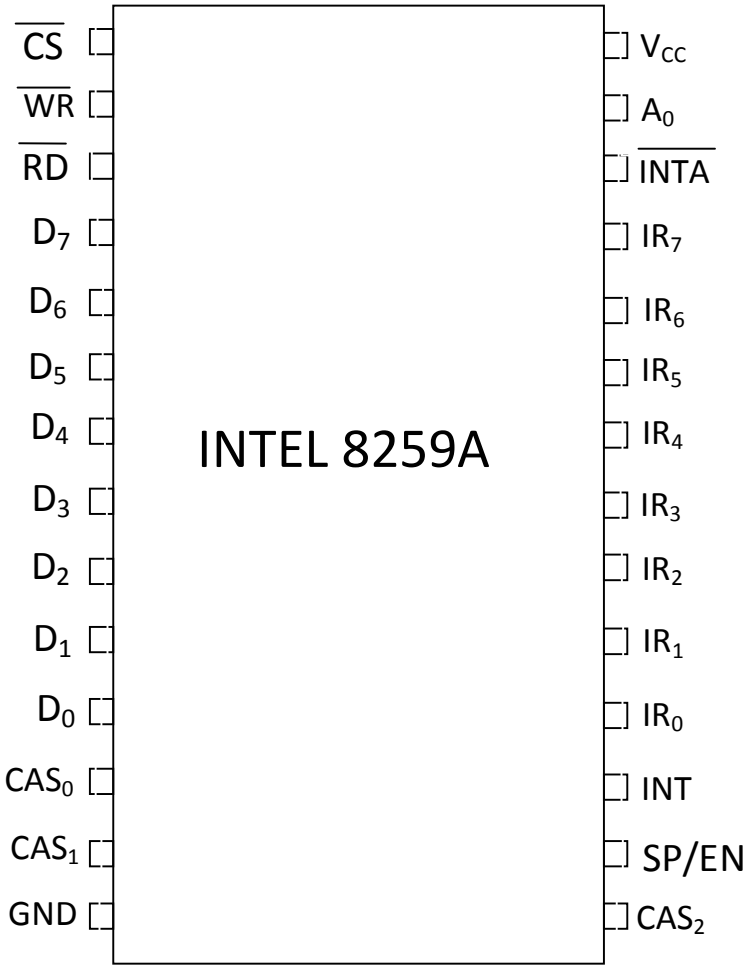
- **INTERRUPT CONTROL LOGIC:**
  This block manages the interrupt and interrupt acknowledge signal to be send to the CPU for serving one of the 8 interrupt request.

- **DATA BUS BUFFER:**
  This bidirectional buffer interface internal 8259A bus to the microprocessor system data bus

- **R/W CONTROL LOGIC:**
  This circuit accepts and decodes command from the CPU.

- **CASCADE BUFFER /COMPARATION:**
  This blocks stores and compare to IDs of all the 8259A use in the system.

8259A PIN DIAGRAM



PIN- DESCRIPTION :

It is a 28 pin IC package.

- **CS** :- It is an active low chip selected signal for enabling read and write operation of 8259A.

- **WR** :- This pin is an active low write enable input to 8259A.

- **RD** :- It is an active low read enable input to 8259A.

- **D_0-D_7** :-They form a bidirectional data bus that carry 8-bit data and interrupt vector information.

- **CAS_0-CAS_2** (CASCADE LINE) :- A single 8259A provides 8 vectored interrupts. If more interrupt are required , the 8259A is used in the cascade mode in which a master 8259A along with 8 slave 8259A can provide up to 64 vectored interrupt lines. These cascade lines attached select line for addressing the slave 8259A.

- **$\overline{SP}/\overline{EN}$** :- (SLAVE PROGRAM / ENABLE LIGIC)

  This pin is dual purpose pin. When the chip is used in buffered mode, it can be used as a buffer enable to control buffer trans-receiver.
  In buffer mode $\overline{EN} = 0$
  If not in buffer mode then the pin is used as input to designate whether the chip is used as a master. ($\overline{SP} = 1$) or a slave ($\overline{EN} = 0$)

- **INT** :- This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU.
- **$IR_0 – IR_7$** (INTERRUPT REQUEST)
  This pin act as a inputs to accept interrupt request to the CPU.

- **$\overline{INTA}$** :- (INTERRUPT ACKNOWLEDGE)
  The pin is an input used to strobe in 8259A interrupt vectored data on to the data bus.

  The device 8259A can be interfaced with any CPU using either device polling or interrupt schemes.

  DEVICE POLLING

  - → Here the CPU keeps on checking each peripheral device in sequence to make sure if it requires any service from the CPU.
  - → If any such service request is noticed, the CPU serves the request and then goes on to next device in sequence.
  - → After all the peripheral devices are scanned, the CPU again starts from the 1^{st} device.
  - → This results in reduction of processing speed because most of the CPU time is consumed in polling the peripheral devices.

COMMAND WORDS OF 8259A

There are two types of command words i.e

i. **Initialization Command Words (ICWs)**
ii. **Operation Command Words (OCWs)**

i. Initialization Command Words (ICWs)

→ Before it starts functioning, 8259A must be initialized by writing 2-4 command word in to the respective command word register. These are called as ICWs.

- If $A_0 = 0$ & $D_4 = 1$

  The control word is recognized as ICW_1.

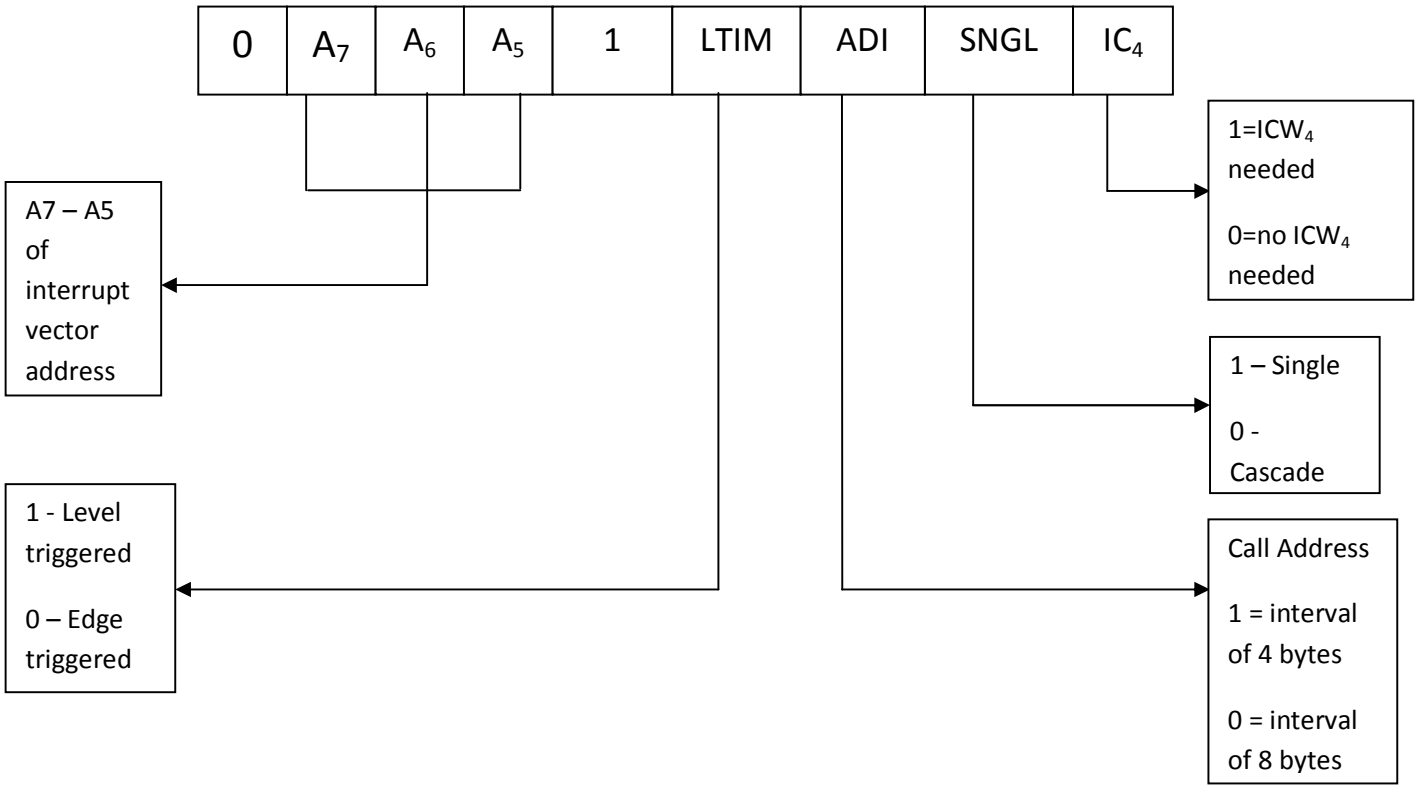
→ It contains the control bits for edge or level triggered mode, single or cascade mode, call address interval and whether ICW_4 is needed or not.

- If $A_0 = 1$

  The control word is recognized as ICW_2.

→ The ICW_2 stores details regarding interrupt vector address.

ICW_1



A_0 = 0 & D_4 = 1, ICW_1

A_0 = 1, **ICW_2**

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8

- ICW_1 & ICW_2 are compulsory command word in initialization sequence of 8259A while ICW_3 & ICW_4 are optional.
- The ICW_3 is read only when there are more than one 8259A in the system i.e cascading is used.
- The ICW_3 loads an 8-bit slave register.

MASTER MODE ICW_3

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S0

Sn = 1 – IRn (Input as a slave)
= 0 – IRn (Input doesn't have a slave)

SLAVE MODE ICW_3

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	ID_2	ID_1	ID_0

D_2 D_1 D_0 - 000 to 111 for IR_0 to IR_7 or slave 0 to slave 7.

ICW_4

The use of this Command Word depends on the IC_4 bit of ICW_1.

If IC_4 = 1, ICW_4 is used otherwise it is neglected.

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/$\overline{S}$	AEOI	MPM

SFNM :-

If SFNM = 1, the special fully nested mode is selected.

BUF:-

If BUF = 1, the buffer mode is selected.

M/$\overline{S}$:-

The bit decided the master and the slave mode of 8259A.

If M/$\overline{S}$ = 1, 8259A is a master.

If M/$\overline{S}$ = 0, 8259A is a slave.

If BUF = 0, M/$\overline{S}$ is to be neglected.

AEOI :-

If AEOI = 1, the automatic end of interrupt mode is selected.

MPM:-

If MPM bit is 0, the 8085 system operation is selected.

If MPM = 1, 8086/88 operation is selected.

Operation Command Words (OCWs):-

- The modes of operation of 8259A can be selected by writing three internal register called as operation command words register.
- The data written into them is called as operation commands word.
- There are three OCWs i.e.

i. OCW_1
ii OCW_2
iii OCW_3

  - OCW_1 is used to mask the unwanted interrupt requests.
  - OCW_2 controls the end of interrupt, the rotate mode and their combinations.
  - OCW_3 handles the special mask mode.

Operating Modes Of 8259A:-

1. **Fully nested mode:-**
   → This is the default mode of operation of 8259A.
   → IR_0 has the highest priority and IR_7 has the lowest.
   → When interrupt request are noticed, the highest priority request among them is determined and the vectored is placed on the data bus.

2. **End of Interrupt (EOI) Mode:-**
   The ISR bit can be set by EOI command issued before returning from the interrupt service in this mode.

3. **Automatic Rotation:-**
   This is used in application where all the interrupting devices are of equal priority.

4. **Automatic EOI Mode:-**
   In this mode 8259A performs an non specific EOI operation.

5. **Specific Rotation:-**
   In this mode a bottom priority level can be selected to fix other priorities.
   If IR5 is selected as a bottom priority then IR5 will have least priority and IR4 will have a next priority and IR6 will have the highest priority.

6. **Special Mask Mode:-**
   In this mode when a mask bit is set, it inhibits/stops further interrupt at that label & enables interrupt from other label which are not masked.

7. **Edge and Level Triggered Mode:-**
   This mode decides whether the interrupts should be edge triggered or label triggered.

8. **Reading 8259 Status.**
   The status of internal register of 8259A can be read using this mode.

9. **Poll Command:-**
   In this mode the INT output of 8259A is neglected.

10. **Special Fully Nested Mode:- (SFNM)**
    This mode is used in more complicated system where cascade is used.
    In this mode, the master interrupt the CPU only when the interrupting device has a higher or the same priority than the one currently being solved.

11. **Buffered Mode:-**
    When the 8259A is used in systems where bus driving buffers are used, it is usedin buffered mode.
    .
12. **Cascade Mode:-**
    In this mode, 8259SA is connected in system containing one master and maximum 8 slave to handle up to 64 priority label.

DATA ACQUISITION SYSTEM

The peripheral devices are connected to the PIC maximum 8 I/O devices can be connected to 8259 in single mode through IR0-IR7 lines. The PIC function is overall manager in an interrupts driven system. If more than one I/O devices then send interrupt request at the same time the PIC determines the priority.

It first entered into the I/O devices the higher priority it sends an interrupt request to the microprocessor through INT line. The microprocessor sends a acknowledgement through INTA line. On the receive top INTA signal all the interrupt of low priority are inhibited.

If more than 8 I/O devices to transfer data using interrupt to 8259 ICs can be connected in series such a connection is known as cascading up to 64 I/O devices connected employing to 8259 ICs.

More than one 8259 are employed in the system the 8259 which is connected to the processor is called master then 8259 which is connected to the master 8259 is called slave. For the measurement and control of physical quantity such speed, displacement etc transducers are used which give electrical voltage proportional to physical quantity.

This electrical voltage obtain as an output of an transducer is an analog quantity it must be converted into digital quantity by the A/D converter before it is applied to a microprocessor. To handle multiple signals an analog multiplexer is used.

Working Principle of Successive Approximation type A/D Converter:-

An A/D converter analog multiplexer is used to convert analog signals to digital quantity. This digital o/p is fed to the microprocessor for processing.

- The most popular method of analog to digital conversion is successive approximation method.
- Here an unknown voltage Vin is compared with a fraction of the reference voltage Vref.

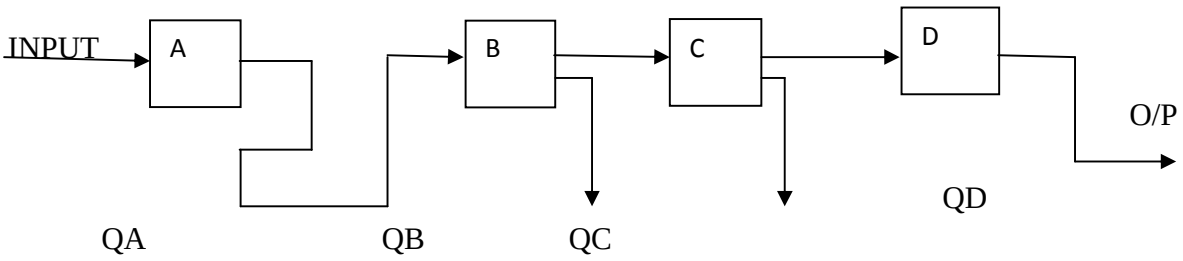
- In the first step the unknown voltage Vin is compared with ½Vr. If Vin>=1/2 Vr, the MSB of the digital output is set to 1.If Vin<1/2Vr, MSB is set to 0.
- In the next step Vin is compared with (1/2b1+1/4)Vr, where b1 is the MSB value already determine. If Vin>=1/2(b1+1/4)Vr the 2^{nd} bit is set to 1.If Vin<1/2(b1+1/4)Vr the 2^{nd} bit is set to 0.
- To obtain the 3^{rd} bit of the digital output Vin is compared with (1/2b1+1/4b2+1/8)Vr.

Clock for A/D converter:-

- The clock frequency required for A/D converter lies in the range of 50 KHz to 800 KHz.
- The clock frequency available & microprocessor kit for user is about 3MHz.

The higher frequency of the microprocessor of the microprocessor can be reduced by using 4 master slave flip-flops which reduced the frequency by $1/10^{th}$.

Fig: -



Sample & hold circuit:-

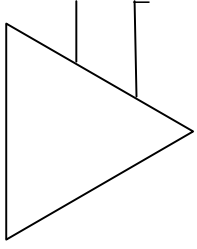
This circuit samples the instantaneous value of the AC signal and maintains it at a constant level .It make this constant voltage available to A/D converter during the conversion period.

LF398:-

LF398, LF298, LF198 are monolithic sample and hold circuit of national semiconductor. An external capacitor known as hold capacitor is used with LF398 to hold the voltage request upon it.

Droop rate:-

It is the rate at which the o/p of sample & hold circuits decreases. To make the o/p partially constant this rate should be very small. V+ V

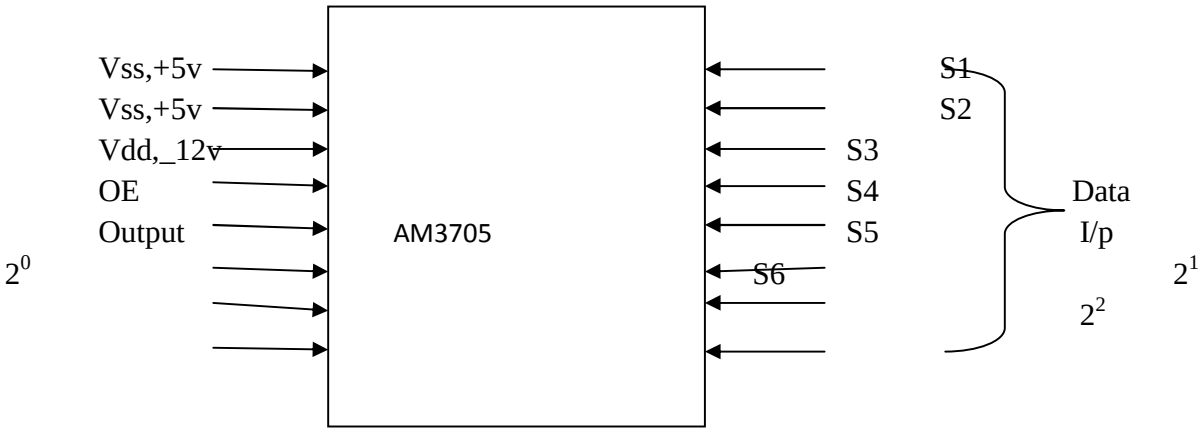


Analog Multiplexer:-

When a number of inputs are to be processed by the computer an analog multiplexer is used. The multiplexer has logic to switch ON a particular desired channel.

- The computer sends appropriate logic to the multiplexer to switch ON the desired channel.
- A multiplexer may be of 8 or 16 channel input multiplexer.
  Example:- AM3705 is an 8 channel analog multiplexer of national semiconductor.

Schematic diagram of AM3705:-



LOGIC FOR AM3705:-

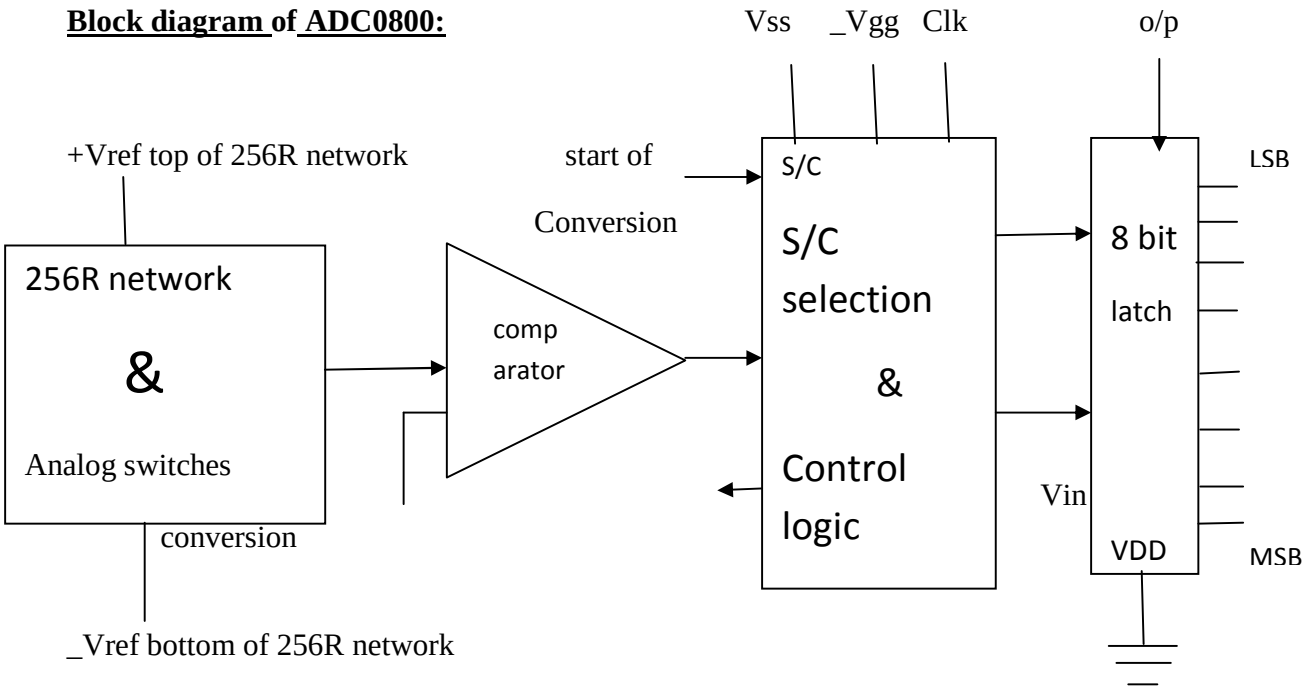
LOGIC INPUT			OUTPUT ENABLE	CHANNEL
2^2	2^1	2^0		
L	L	L	H	S_1
L	L	H	H	S_2
L	H	L	H	S_3
L	H	H	H	S_4
H	L	L	H	S_5
H	L	H	H	S_6
H	H	L	H	S_7
H	H	H	H	S_8

ADC 0800 :-

- The ADC0800 is an 8 bit monolithic A/D converter of national semiconductor .It contains a 256 series register which is called 256R network.
- Analog switches selection & control logic a high input impedance comparator and 8 bit latch have been incorporated.
- It uses accessive approximation technique for analog to digital conversion the reference voltage is applied to 256R network.
- The unknown analog input voltage is compared with a fraction of reference voltage with help of analog switches and R network.

The reference voltage applied approx the 256R network determines the analog input range in case of Vr=100volts the top of R network is connected to +5volt supply at the bottom to -5volt.

Block diagram of ADC0800:



Zero & Full scale Adjustment:-

The ADC0800 gives complementary digital output. The digital o/p corresponding to 5v input is 10000000. If the o/p is not zero when the input is +5V adjustment of Vref . This adjustment is called full scale adjustment.

When the input voltage is zero the actual output is 80 decimal. If the digital output corresponding to zero volts the output is not 80 hexadecimal then it is adjusted by using 1 kilo ohm variable register. The relationship between input voltage +5v is

Vin=Vref(1/2b7+1/4b6+1/8b5+1/16b4+1/32b3+1/64b2+1/128b1+1/256b0)_5V.

DIGITAL TO ANALOG CONVERT(D/A CONVERTER)-

Digital to analog converters are used to convert digital quantity to analog quantity.

They produce an output current or voltage proportional to digital quantity apply to its input.

They are used for the control of relay, small motor, actuator etc.

In communication system digital transmission is faster and convenient but the digital signal have to be converted back to analog signal at the receiving terminal.

D/A converter are also used as a part of circuit tree of several A/D converter.

OPERATING PRINCIPLE OF A DAC-

DAC contain a ladder network. The network has input prints for binary bits of the digital word.

When MSB of the digital word is 1 it produces an output current Iref/2.

The bit next to MSB produces Iref/4 and so on.

The output current is given by

$$Iout=Iref(1/2\ b_{n-1}+1/4\ b_{n-2}+1/8\ b_{n-3} \ldots 1/2^{n}b_0)$$

for an n-bit digital input.

For an 8-bit DAC

$$Iout = Iref\ (1/2\ b_7+1/4\ b_6+1/8\ b_5+1/16\ b_4+1/32\ b_3+1/64\ b_2+1/128\ b_1+1/256\ b_0)$$

where b0, b1, . . . b7 are the binary bits of digital word apply to DAC.

V=IR=> I=V/R

$$Iref=Vref/R$$

where R is resistance in series with Vref.

Example-

Vref=5V, R=2.5 KΩ

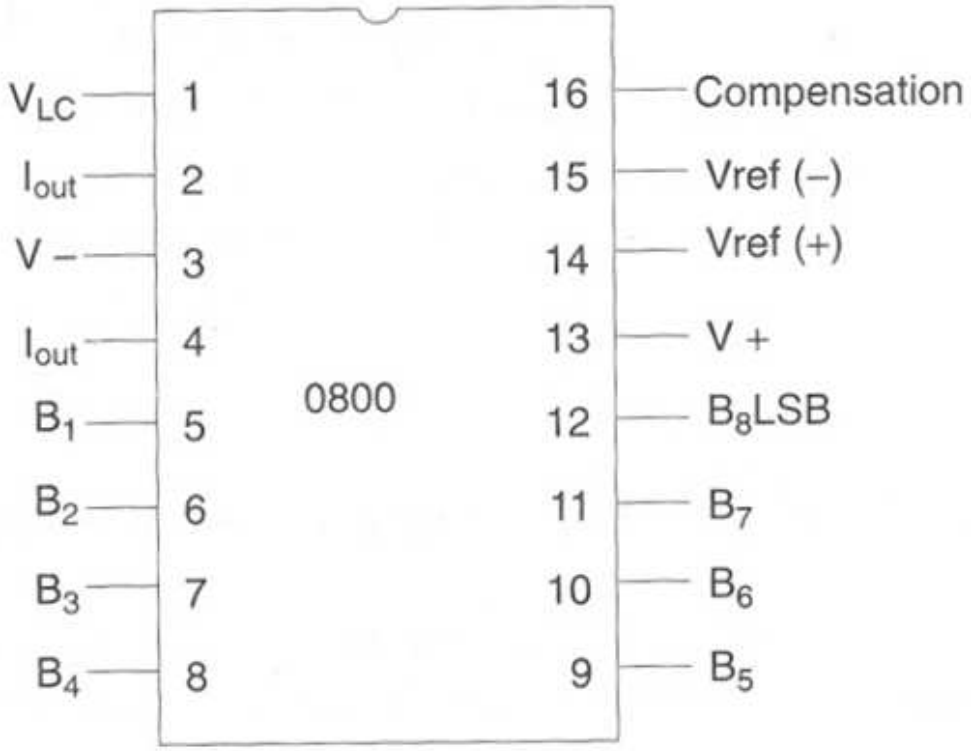
Iref= $5/2.5\times10^{3}$ =$50/25\times10^{-3}$ =2MA

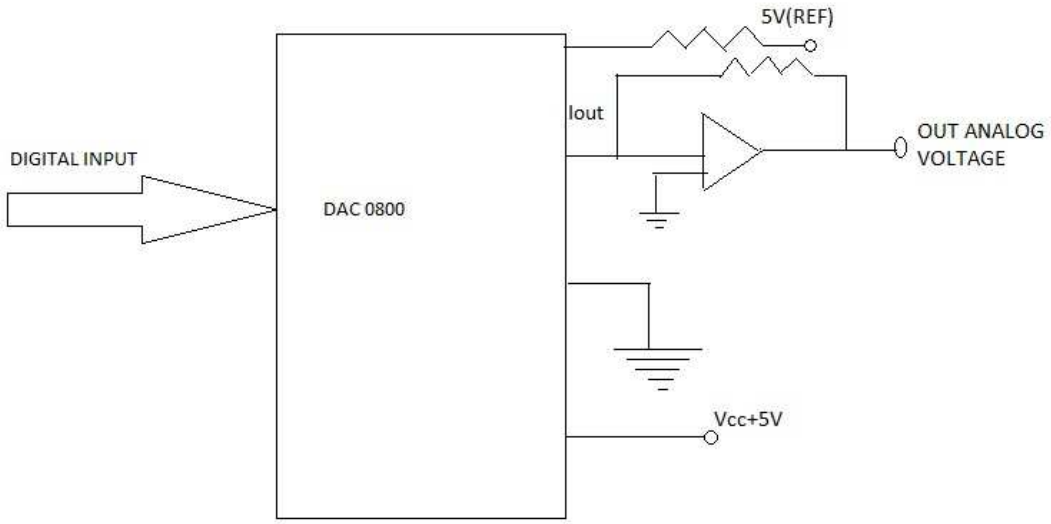
Using an external opamp a voltage output proportional to Iout is produced.

DAC 0800-

It is a simple monolithic 8-bit DA converter.

PIN DIAGRAM OF DAC 0800





ANALOG SIGNAL CONDITIONING

The signal conditioning means manipulating an analog signal in such a way that it means the requirement of the next stage for further processing.

Most common use is in A/D converter.

They are commonly used in control engineering application. Operational amplifier are use to carry out the amplification of the signal in signal conditioning stage.

INPUTS TO SIGNAL CONDITIONING

Signal inputs accepted by signal conditioner are DC voltage and current , AC voltage and current, frequency and electric charge.

Sensor inputs can be accelerometer, thermocouple, thermistor, registance thermometer etc.

Outputs for signal conditioning equipment can be voltage ,current , frequency timer or counter ,registance etc.

SIGNAL CONDITIONING PROCESSES

It includes amplification , filtering, converting , range matching ,isolation of analog signal. And any other processes required to make sensor output suitable for processing after conditioning.

1. FILTERING

It is the most common signal conditioning function as not all the signal frequency spectrum contain valid data.

60 Hz AC power lines present is most environment will produce noise if amplified.

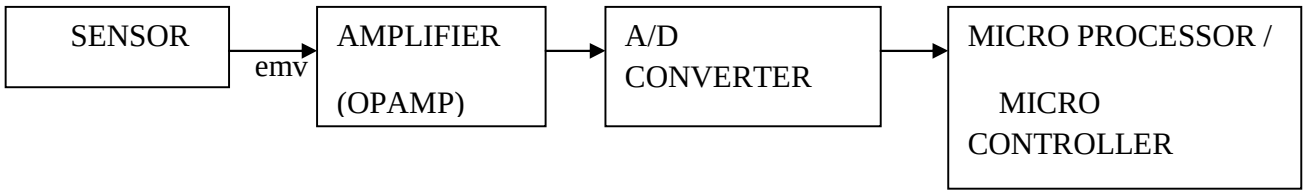
2. AMPLIFING

Signal amplification performs 2 important functions.

a. Increases the resolution of the input signal.

b. Increases its signal to noise ratio .

The output of an electronic temperature sensor which is in the range of mili volt is too low for an A/D convertor to process directly. So it is necessary to use an amplifier to bring the voltage level to that required by the ADC.



Commonly used amplifier on signal conditioning are S/H amplifier , peak detectors, log amplifier ,antilog amplifier etc.

3. ISOLATION

Signal isolation must be used in order to pass the analog signal from the source to the source to the measurement device without a physical connection.

It is important to isolate the expensive equipment used for signal processing after conditioning from the sensor.

There are 2 types of isolation

A. Magnetic

B. Optic

A. Magnetic

Magnetic isolation transforms the signals from voltage to a magnetic field allowing the signal to be transmitted without a physical connection.

B. Optic

Optic isolation takes an electronic signal and modulates its to a signal coded by light transmission (optical encoding) .Which is there used for the next stage of processing.

Types of devices that used signal conditioning are isolation amplifier ,multiplexer , A/D convertor , D/A convertor , inverter etc.

It is mostly used for data acquisition in which sensor signal must be normalised and filter to level suitable for analog frequency to voltage convertor, voltage to frequency convertor ,current to voltage convertor etc.

4. COMPARATOR

Sometimes there is no need to send the entire range of voltage from a sensor to the analog to digital convertor.

A circuit called comparator is used which takes an analog sensor voltage and compares it to a threshold voltage, Vth.

If the sensor voltage is greater than the threshold the output of the circuit is maximum.

$V_{s0} > V_{th} \longrightarrow$ maximum

$V_{S0} < V_{th} \longrightarrow$ minimum

If the sensor voltage is less than the threshold the output of the circuit is minimum.

MODULE – 4

Registers of Intel 8086

The Intel 8086 contains the following register.

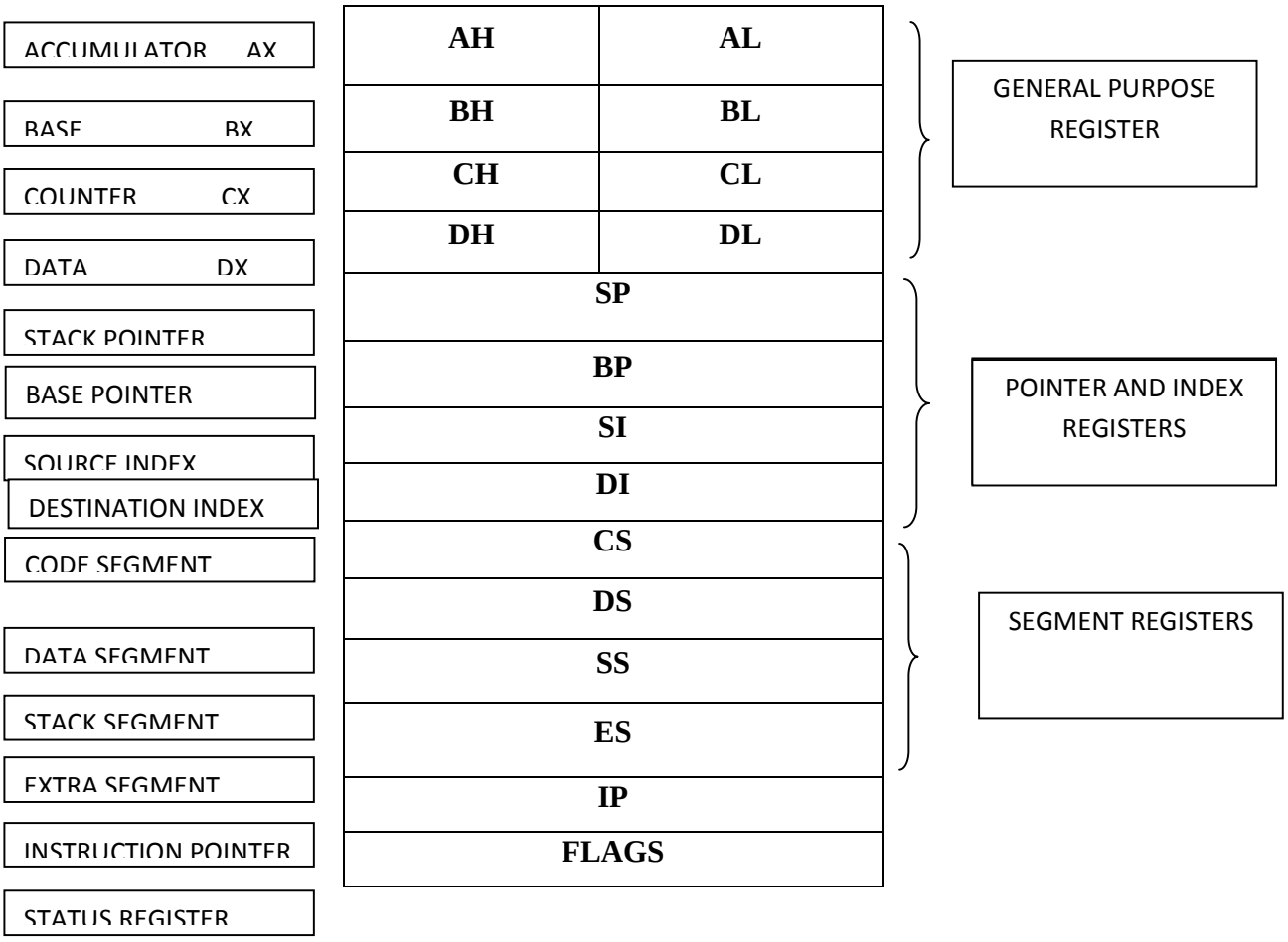
(i) General purpose Registers
(ii) Pointer and Index Registers
(iii) Segment Registers
(iv) Instruction Pointer
(v) Status Flags

General purpose Registers

There are four 16-bit general purpose register: AX, BX, CX and DX. Each of these 16-bit registers are further subdivided in to two 8-bit register as shown below.

16-bit register	8-bit high order register	8-bit low order register
AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

Register AX serves as an accumulator. Register BX,CX and DX are used as general purpose register in addition to serving as general purpose register they also serve as special purpose register . as a special purpose register BX serves as a base register for computation of memory address .in 8086 memory addresses are to be calculated using the contents of the segment register and effective memory address . These will be explained later on . Register CX is also used as a counter in case of multi-iteration instruction. When the content of CX becomes zero such instruction terminate the execution. DX register is also used for memory addressing when data are transferred between I/O port and memory using certain I/O instruction.



REGISTER ORGANIZATION OF INTEL 8086

Pointer and Index Register. The following four figures are in the group of pointer and index register.

1. Stack Pointer, SP
2. Base pointer , BP
3. Source index ,SI
4. Destination Index, DI

The function of SP is same as the function of stack pointer in Intel 8085. BP,SI and DI are used in memory address computation.

Segment Register. There four segment registers in 8086.

1.Code Segment Register , CS

2. Data Segment Register , DS

3. Stack Segment Register, SS

4. Extra Segment Register, ES

In an 8086 microprocessor-based system memory is divided in to the following four segments:

1 .Code Segment

2. Data Segment

3. Stack Segment

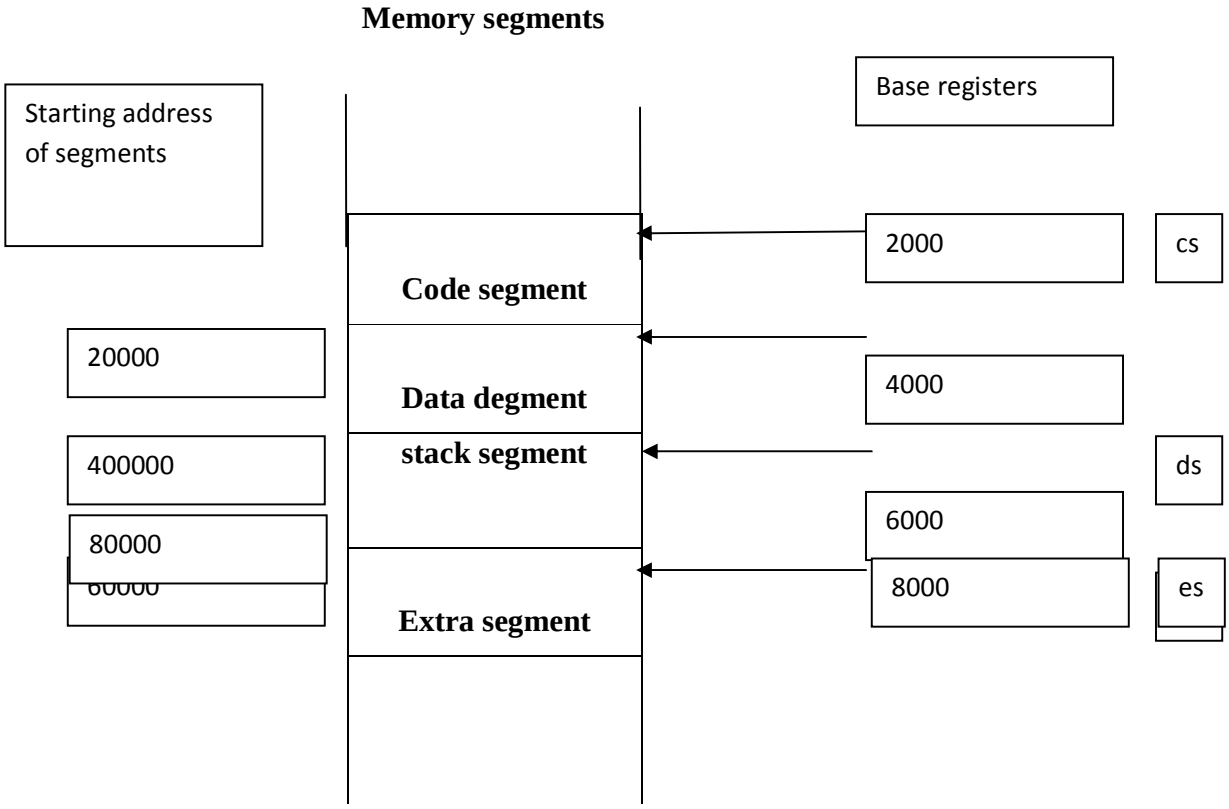
4. Extra Segment

The code segment of the memory holds instruction code of a program. The data variables and constant given in the program are held in the data segment of the memory. Stack segment holds address and data of subroutines. It also holds the contents of register and memory location given in PUSH instructions. Before attending an interrupt the microprocessor saves the contents of program counter on the stack. Also when CALL instruction is executed , before the execution of the subroutine the address of the next instruction of the program is saved on the stack. The extra segment holds the destination address of some data of certain string instruction, and so on.

A segment register point to the starting address of a memory segment currently being used. For example the code segment registers points to the starting address of the data segment, and so on. T he maximum capacity of a segment may be up to 64kbyte . The starting address of a segment is divisible by 16 . The segment shown in the figure is currently used segment. There more number of such segment to make the total memory capacity 1Mbyte.

The 8086 instruction specify 16-bit memory address. The actual addresses are of 20 bits. They are calculated using the contents of the segment register and effective memory address. The effective memory address is computed in a variety of ways. It depends on the such as PUSH,POP,CALL or RET , The content of the stack pointer(SP) and the content of the stack segment register(SS) are used to complete the stack of the location to be accessed. The index register SI and DI together with segment register DS and ES are used to perform string operations. The source addresses for string operation are computed using the content of SI and

DS . The destination addresses for string operations are computed using the contents of DI and ES.



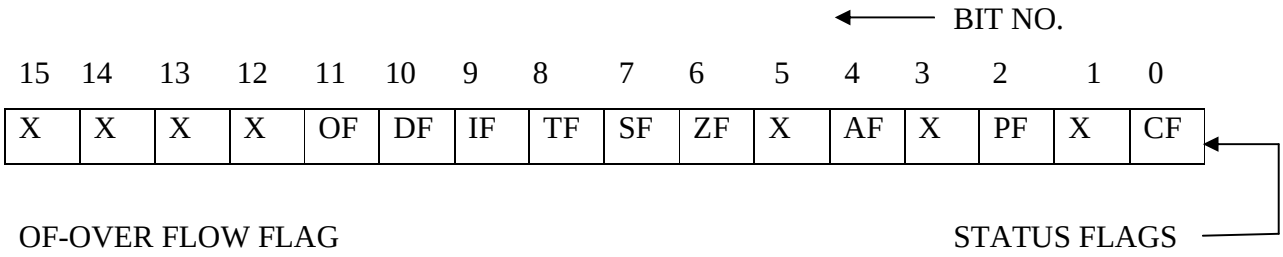
[Memory segments]

Instruction pointer (IP).:

The instruction pointer in the 8086 microprocessor act as a program counter. It point to the address of the next instruction to be executed. Its content is automatically incremented when the execution of a program proceeds further. The content of the instruction pointer (IP) and the content of the code segment register (CS) are used to compute the memory address of the instruction code to be fetched. This is done during instruction fetch operation.

Status Register :

The 8086 contains a 16-bit status register. It also called flag register or program status word (PSW) . There are 9 status flags that are: over flow flag, Direction flags, Input enable flag, Trap flag, Carry flag, parity flag, Auxiliary carry flag, zero flag, sign flag. Out of nine flag 6 are condition flag and three are control flags. These six condition flag are carry, auxiliary carry, zero, sign and parity and overflow flag. The flag are set/reset by the processor after the execution of an arithmetic and logic operation. The three control flag are trap (or trace) interrupt and directional flag. These flags are set/reset by the programmer as the required by certain instruction in the program. The overflow, trap, interrupt and directional flags are new. Other flags are same as those available in Intel 8085. The overflow flag is set to 1 if the result of a signed operation become out of range other it is reset it is made 0.



OF-OVER FLOW FLAG
DF-DIRECTION FLAG
IF-INTERRUPT ENABLE FLAG
TF-TRAP FLAG
SF-SIGN FLAG
ZF-ZERO FLAG
AF-AUXILIARY CARRY FLAG
PF-PARITY FLAG
CF-CARRY FLAG

[Status flags of Intel 8086]

When the trap flag (TF) is set to 1 a program can be run in single-step mode. The interrupt flag (IF) is set to 1 to enable INTR of 8086. If it is 0, INTR is disabled. It is set by STI instruction and declared by CLI instruction, IF is automatically cleared when an interrupt is recognised. It disabled ITNTR. IRET used at the end of interrupt service sub routine (ISS) restore IF flag in the state in the it was before interrupt occurred.

The directional flag DF is used in string operation .It can be set by STD instruction and cleared by CLD instruction. If it is set to 1, string bytes are accessed from higher memory address to lower memory address. When it is set to 0 the string bytes are accessed from lower memory address to higher memory address. For MOVES instruction, if DF is set to 1, the content of index register SI and DI are automatically decremented by the processor to access the string from the highest memory address down to the lowest memory address. If DF is made zero, SI and DI are automatically incremented to access the string starting with the lowest address.

Addressing Modes of Intel 8086

The way by which an operand is specify for instruction is called addressing mode. Intel 8086 has 8 addressing mode.

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Indexed Addressing
8. Based Indexed with Displacement Addressing

1.REGISTER ADDRESSING

- Here the operand is placed in one of the 16-bit or 8-bit general purpose registers.
- Example- MOV AX,CX (move 16-bit data in base register to accumulator)

2.IMMEDIATE ADDRESSING

- Here the operand is specified in the instruction itself.
- Example-MOV AL, 35H (move immediate data 35H to 8-bit register A)

MOV BX,0305H (move 16-bit data0305 to register base.

The remaining 6 addressing mode specify the location of an operand which is placed in memory. When an operand is stored in memory location, how far the operand's memory location is within a memory segment from the starting address of the segment, is called offset or effective address (EA).

An offset is determined by adding any combination of three address elements: displacement, base and index.

Displacement:- It is an 8-bit or 16-bit immediate value given in the instruction.

Base:- It is the content of the base register, BX or BP.

Index:- It is the content of the index register, SI or DI.

The Combination of these three address elements give six memory addressing modes as described below.

3. DIRECTADDRESSING

In this mode the operand's offset is given in the instruction as an 8-bit or 16-bit displacement element.

- Example:- 1.ADD AL,[0301]
  The contents of memory 0301 is added to the content of AL ,and the result is placed in AL .
- 2. ADD [0301],AX
  This instruction adds the content of AX to the content of memory location 0301 and 0302.

4. REGISTER INDIRECT ADDRESSING

The operand's offset is placed in any one of the registers BX,BP,SI or DI as specified in the instruction.

Example:- MOV AX,[BX]

- This instruction moves the content of memory location addressed by the register BX to the register AX.
- For examples, BX contains 0301,and the content of 0301 is 53H and the content of next memory location is 95H.The 9553 will move to AX.
-

5.BASED ADDRESSING

The operand's offset is the sum of an 8-bit or 16-bit displacement and the content of the base register BX or BP. BX is used as a base register for data segment, and BP is used as a base register for stack segment.
Offset=[BX +8-bit or 16-bit displacement]

- Examples are:-

MOV AL, [BX+05]; an example of an 8-bit displacement.

Suppose ,the register BX contain 0301.The offset will be 0301+05=0306.The content of the memory location 0306 will move to AL.

- MOV AL, [BX+1346H]; an example of 16-bit displacement.
  If [BX] =0301.The offset =0301+1346=1647H
  The content of 1647H will move to AL.

6. INDEXED ADDRESSING

The operand's offset is the sum of the content of an index register SI or DI and an 8-bit or 16 bit displacement.

Offset= [SI or DI+8-bit or 16-bit displacement]

- Examples are:
  MOV AX, [SI+05] ; an example of 8-bit displacement.
  MOV AX, [SI+1528H] ; an example of 16-bit displacement.

7. BASED INDEXED ADDRESSING

The operand's offset is the sum of the content of a base register BX or BP and an index register SI or DI.BX is used as a base register for data segment, and BP is used as a base register for stack segment.

Offset= [BX or BP] + [SI or DI]

- Examples are:
  ADD AX, [BX+SI]
  MOV CX, [BX+SI]

8. BASED INDEXED WITH DISPLACEMENT

In this mode of addressing the operand's offset given by

Offset= [BX or BP] + [SI or DI] + 8-bit or 16-bit displacement

- Examples are:
  MOV AX, [BX+SI+05]; an example of 8-bit displacement
  MOV AX,[BX+SI+1235H]; an example of 16-bit displacement.

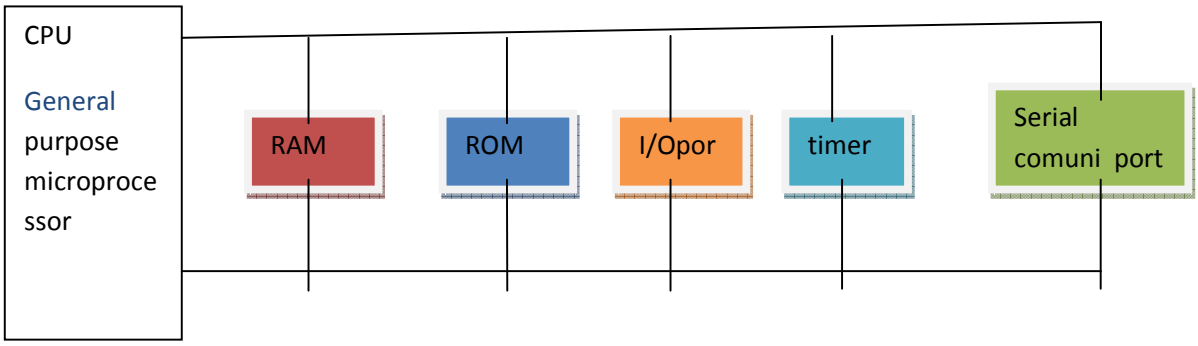
Difference between MICROPROCESSOR and MICROCONTROLLER

- Microprocessor contains no RAM,ROM and no I/O port on the chip itself.it is only a CPU.
- They are commonly refered to as general purpose microprocessor.
- The system designer using a general purpose microprocessor such as a Pentium must add RAM,ROM,I/O ports and timer externally to make them fuctional.

MICROCONTROLLER

- Microcontroller has a cpu(microprocessor) in addition to a fixed amount of RAM,ROM,I/O port and timer are all on a single chip.
- So the designer can't add any external memory, i/o port or timer to it.
- The fixed amount of on chip RAM,ROM and the no of i/o ports in microcontrollers make them ideal for any applications in which cost and space are critical.
  Ex-TV remote control
- These application open required some i/o operation to read signal and turn on/off certain bits so these processors are called IBP (Itty bitty Processor)
- In 8051 it allows to set or reset indivisual pins of i/o ports and status registers .it also provides bytes addressing so it is called as bit and byte processor.

General Purpose Microprocessor System



Microcontroller

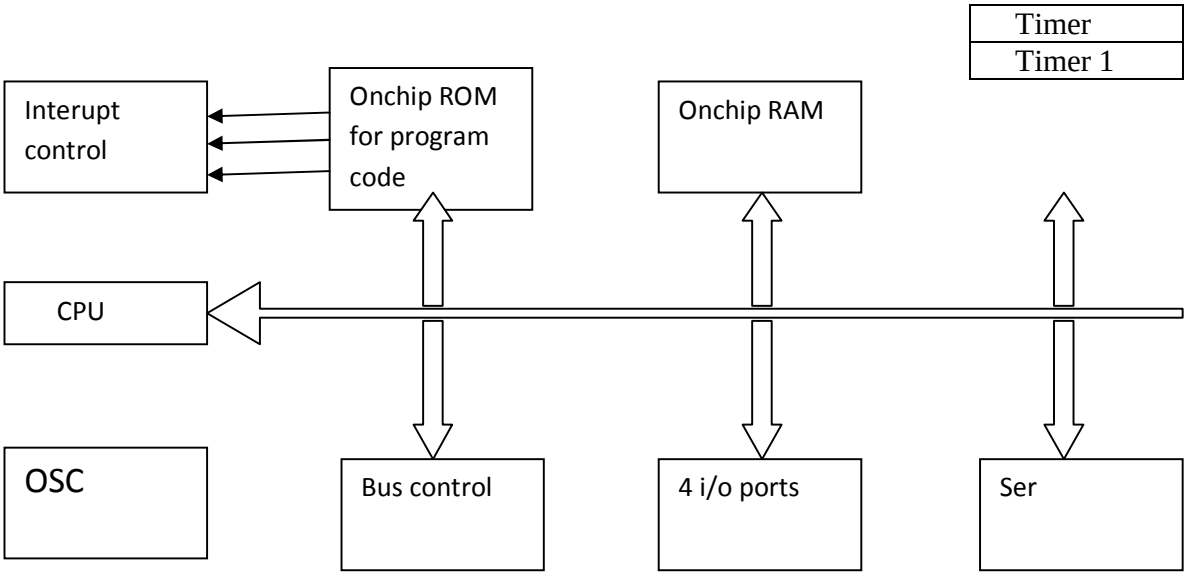
CPU	Ram	Rom	ETC
i/o port	Timer	Serial communication port	

- Disadvantages of the microcontroller is that if the system designer requires more memory or more no. of i/o ports cant add them externally.
- Microprocessors and Microcontrollers are widely used in embedded system product .
- An embedded product uses a microprocessor or microcontroller to do one task and one task only.
  Ex- a printer is an embedded system in which the processor inside it performs only one task i.e breaking the data and print it.
- In an embedded system there is only one application software that is typically burned into run.
  Ex-A pc is connected to various embedded products such as keyboard,printer,modem,disk controller,sound card,cd driver etc.
- Each of the peripheral has the microcontroller inside it that performs only one task.
  Ex-Inside every mouse there is a microcontroller that performs the task of finding the mouse position and sending it to the pc.
- Some embedded products used in homes i.e TV,telephones,remote controller,camera etc.

Overview of 8051 family

- In 1981 Intel corporation introduced an8-bit microcontroller called 8051.
- This microcontroller has 120 bytes of RAM 4-kilobytes of microchip ROM,two timers ,one serial port and four i/o ports are 8-bit wide all on a single chip.
- It is an 8- bit processor meaning that the CPU can work pon on only 8 bit of data at a time.
- Intel refers to 8051 MCS-51
- Other members of 8051 family they are 8052 and 8053.

Data larger than the 8-bit has to be broken into 8 bit process to be processed by the CPU.



- **8052 microcontroller**

It has all the standard features of 8051 along with extra 120 bytes of RAM and an extra timer or 8052 has 8 kilobytes of onchip program ROM ,256 bytes of RAM,4 i/o ports and 6 timers.

- **8031 microcontroller**

This chip refered to as a ROM less 8051.Since it has 0 kilobytes of onchip ROM.to use this chip the designer must add an external rom chip.

Comparison of 8051 family members

Feature	**8051**	**8052**	**8031**
ROM(onchip program space in bytes)	4k	8k	0k
RAM(bytes)	128	256	128
Timers	2	3	2
i/o pins	32	32	32
Serial ports	1	1	1

REFERENCES :

1. B. Ram, "*Fundamentals of Microprocessors and Microcomputers*", Dhanpat Rai Publications.

2. A.K.Ray and K.M.Bhurchandi – "*Advanced Microprocessors & Peripherals"* Tata McGraw Hill.

3. M.A. Mazidi and J.G. Mazidi, "*The 8051 Microcontroller and Embedded Systems*", Pearson Education, India.

4. Ramesh S. Gaonkar, "*Microprocessor Architecture, Programming and Application with the 8085*", Fourth Edition, Penram International Publishing (India).

5. D.V. Hall, "*Microprocessors and Interfacing*",2nd Edition McGraw-Hill Book Company.