ELECTRONICS & TELECOMMUNICATION ENGINEERING

4TH SEMESTER

SUB: MICROPROCEESER AND ITS INTERFACING

Contents Developed by:

- **Miss Jyotisree Nayak,**
  PTGF (ETC)
  UGMIT, Rayagada

&

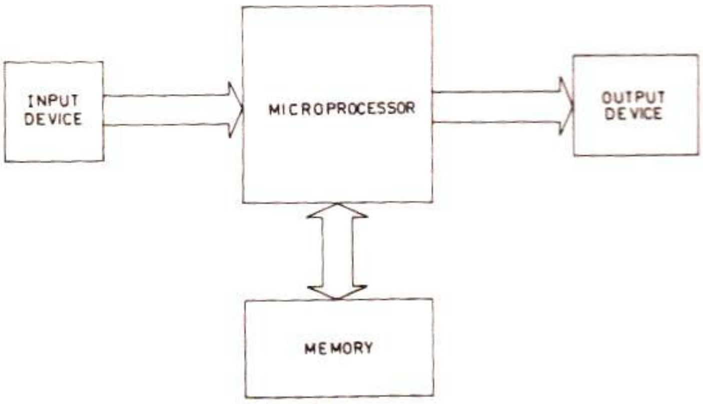
- **Sri Gadadhar Maharana**
  Senior Lecturer (ETC)
  UGMIT, Rayagada

INTRODUCTION TO MICROPROCESSOR:

Introduction:

The microprocessor is one of the most important components of a digital computer.

Digital computer makes processing of numbers and it is a programmable machine. The main components are CPU, memory, input device and output device.



Schematic diagram of a microcomputer

CPU executes instruction. The memory is a storage device. It stores programs, data and result. Output device plays, prints programs, data and result.

The CPU built on a single chip or IC is called microprocessor.

A Digital computer in which one microprocessor has been provided to act as a CPU is called micro computer.

The CPU of a large powerful digital computer contains more than one microprocessor.

A computer whose CPU contains more than one microprocessor is called multi processor computer system.

Single chip microcomputers are called microcontroller.

Application:

It is used as CPU in computers.

In automobiles.

Microcontrollers are used to measure and control in industry for process and machine control in instrumentation, for commercial and consumer appliances control etc.

This microcontroller is used in military equipment, radars, tanks etc.

Evaluation of Microprocessor:

The 1st microprocessor, Intel 4004, a 4-bit PMOS microprocessor was introduced in the year 1971 by Intel Corporation, USA. After 4004 an enhanced version of Intel 4004 was developed i.e. Intel 4040 many other companies like Rock Well International, Toshiba etc are also developed 4-bit microprocessor.

In 1972, Intel introduced the 1st 8-bit microprocessor i.e. Intel 8008 which is also uses PMOS techniques. PMOS technologies were slow and not compatible with TTL Logic. So in 1973 Intel introduced on another powerful and faster 8-bit NMOS microprocessor i. e. Intel 8080. It is faster and compatible with TTL Logic but the drawback of it was it requires 3 power supplier.

In 1975 Intel developed an improved 8-bit NMOS microprocessor, Intel 8085 which uses only one +5V supply. It is an improved version of Intel 8080.

Other companies are also manufactured the 8-bit microprocessors like Zilog's Z80 and Z800, National Semiconductor's NSC800, Motorola etc.

These 8-bit microprocessors are mostly used in general purpose computers. It is having memory addressing capacity of 8-bit microprocessors and 64 KB. The clock frequency is of 1MHz to 6 MHz. It uses LSI technology and contains 5000 to 10,000 transistors.

In the year 1978 Intel introduced a 16-bit microprocessor, Intel 8086. Other 16-bit microprocessor are Intel 80186, Intel 8088, Intel 80188, Intel 80286.

The Intel 80286 has been designed for multi user system. Beside CPU it contains integrated memory, management unit, four level memory protection and support for vertical memory and operating system. These microprocessors are used VLSI technology. These can address memory in the range 1MB to 16 MB. It is used for multiprocessor environment.

After 1985 Intel introduced the 32-bit microprocessors i.e. Intel 80386 which became very popular and it was moistly used in desktop computers. In short it is also called as 386 microprocessor. These 32-bit microprocessor are 486, Pentium, Pentium pro, Pentium MMX, Pentium-II, Pentium-IIII and Pentium-IV.

Micro processor	Year of Introduction	Word Length	Memory addressing capacity	Pins	Clock	
4004	1971	4-bit	1 KB	16	750KHz	
4040	Advanced Version of 4004					
8008	1972	8-bit				PMOS technology
8080	1973	8-bit				NMOS technology
8085	1975-76	8-bit	64 KB	40	3-6MHz	**8-bit up are based on LST technology and8085 is a NMOS up.**
8086	1978	16- bit	1 MB	40	5-10 MHz	
8088	1980	8/16-bit	1 MB	40	5-8MHz	
80286	1982	16- bit	16MB /4GB	68	6-12.5 MHz	16 bits up are based on VLSI technology.
80836	1985	32-bit	4GB	132	20-33 MHz	
80486	1989	32-bit	4GB	168		
Pentium		32-bit	4GB	237 PGA		
Pentium Pro		32-bit	64 GB-	387 PGA		
Pentium-II		32-bit	-	-		
Pentium-III		32-bit	64 GB-	370 PGA		
Pentium-IV		32-bit	64 GB-	423		

*PGA= Programmable Gate Array.

****

CHAPTER – 2

BASIC ARCHITECTURE OF 8-BIT MICROPROCESSOR (8085):

Introduction:

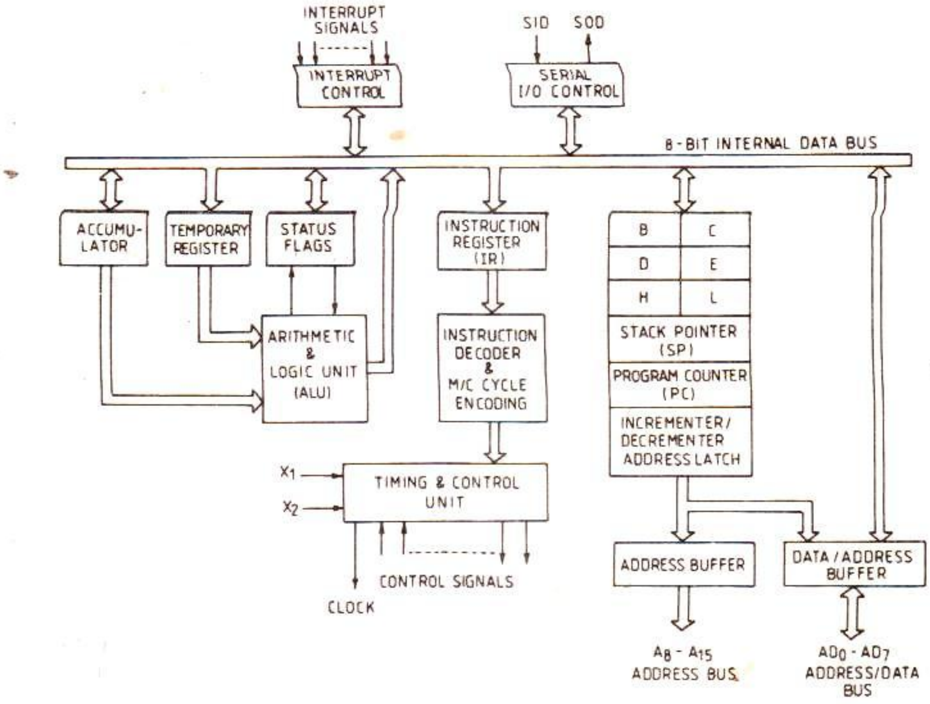
The microprocessor is a central processing unit of a computer. Intel 8085 is the 1st 8-bit microprocessor. It was invented in the year 1975.

The main features of 8-bit microprocessor 8085 is

- It is an 8-bit NMOS microprocessor.
- It having 40 pins.
- Based on LSI (Large Scale Integrated) technology
- Its clock speed is of 3MHz
- The clock cycle is of 320 ms
- It has 80 basic instructions and 246 opcodes
- It is having 3 sections i.e.

(1) ALU
(2) Timing and control unit.
(3) And a set of Registers.

Architecture of 8085 Microprocessor:



Block diagram of Intel 8085.

It is having 3 sections i.e.

(1) ALU
(2) Timing and control units.
(3) A set of Registers.

ALU :-

ALU – Arithmetic and Logic unit.

This ALU performs all the Arithmetic and Logical operations like

(1) Addition
(2) Substraction
(3) Logical AND
(4) Logical OR
(5) Logical Ex-OR

(6) Complement (NOT)
(7) Increment (add 1)
(8) Decrement (Subtract 1)
(9) Left shift, Rotate left Rotate right
(10) Clear etc.

Timing and control unit:

This timing and control unit is a section of CPU. It generates timing and control signals which are necessary for the execution of instructions.

It provides status, control and timing signals which are required for the operation of memory and I/O devices.

It controls the entire operations of the microprocessor and peripherals of connected to it. So the control unit of the CPU acts as the brain of the computer system.

Registers:

There are various types of Registers in the microprocessor 8085.

These registers are used for temporary storage and manipulation of data and instructions. The data remains in the registers till they are sent to memory or I/O devices.

In large computer the no of registers are more but in small computer the no of registers are less due to limited size of the chip.

The registers of 8085 microprocessor are:

(1) Accumulator (Acc) i.e. register (One 8-bit)
(2) 6, 8-bit general purpose registers i.e. B,C,D,E,H and L
(3) 1, 16-bit Stack pointer, SP
(4) 1, 16-bit program counter, PC
(5) Instruction register
(6) Temporary register

Accumulator:-

It is a 8-bit special purpose register which is used to store one of the operand which is required during the application of arithmetic and logical operation.

After the logical and arithmetic operation, the result is by default stored in accumulator.

This accumulation can be called as general purpose register, if it is used to store the data or operand which is required during the execution of arithmetic and logical operation.

General Purpose Register:

The 8085 microprocessor contains 6, 8-bit general purpose Registers i.e. B, C, D, E, H and L.

To hold 16-bit data 2 8-bit general purpose registers are employed as a pair i.e. called as register pair. The pairing of registers are B-C, D-E and H-L.

This general purpose registers and accumulator are accessible to programmer. He can store data in these registers during writing the program.

Program Counter:

It is a 16-bit special purpose register which is used to hold the address of the next instruction.

When microprocessor reads one instruction from memory, the PC will automatically incremented and points to the address of the next instruction.

For the above purpose we have to first initialize the P.C. by starting address of 1st instruction of the programme.

Stack Pointer (SP):

It is a special purpose 16-bit register which is used to store the address of the top of the stack.

Stack is a sequence of memory locations set aside by a programmer.

Stack operates in LIFO operation (Last in Fast Out).

The stack pointer controls the addressing of the stack.

Instruction Register:

This register is used to store the Opcode of the current instruction which is executed by the microprocessor.

Temporary Register:

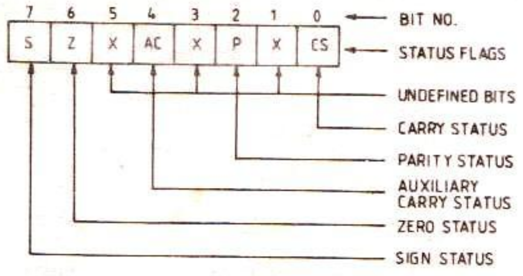
It is an 8-bit register associated with ALU. It holds data during an arithmetic or logical operation. It is used by the Microprocessor. It is not accessible to the programmer.

Flag Register:

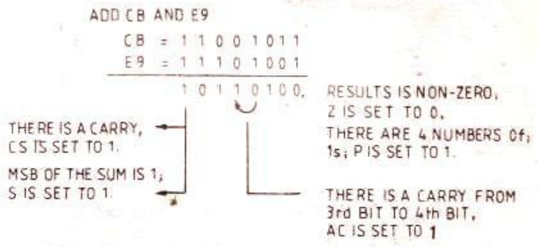
In 8085 microprocessor, it has an 8-bit flag register which is used to provide information about the result outcome from arithmetic and logical operation, out of 8-bits of this register, 3 bits are not used they are D1, D3 and D5.

The 5 status flags of Intel 8085 are:

(1) Carry flag (CS) – D0
(2) Parity Flag (P) – D2
(3) Auxiliary carry flag (AC) – D4
(4) Zero flag (Z) – D6
(5) Sign flag (S) – D7



Status Flags of Intel 8085


ADD CB AND E9
CB = 1 1 0 0 1 0 1 1
E9 = 1 1 1 0 1 0 0 1
1 0 1 1 0 1 0 0,
RESULTS IS NON-ZERO, Z IS SET TO 0,
THERE ARE 4 NUMBERS Of; 1s; P IS SET TO 1.
THERE IS A CARRY, CS IS SET TO 1.
MSB OF THE SUM IS 1; S IS SET TO 1.
THERE IS A CARRY FROM 3rd BIT TO 4th BIT, AC IS SET TO 1

Status Flags for ADD operation

Carry Flag:

Carry flag is set to 1 when a carry is generated from D7 bit of the accumulator when a borrow is required foam a non-existing position to D7 bit, then we have to set the carry bit to 1.

Parity Flag:

The parity flag is set to 1 if the result of the arithmetic and logical operation is having even no. of '1'.

Auxiliary Cary Flag (AC):

If there is a carry occurs from the 3 bit to 4th bit, at that time we have to set the Ac as 1.

Zero Flag (Z):

The zero flag is set to be ‘1’ if all the bits of the result are zero.

Sign Flag (S):

If the data are represented in signed binary form after execution of arithmetic and logical operation, if the D7 bit of the accumulation is ‘1’ then sign flag is set to ‘1’.

Example:

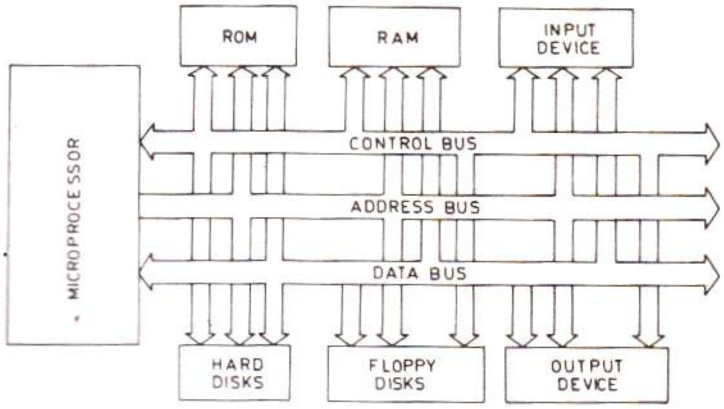
Q: - Add 1001011 with 11101001

1	1	0	0	1	0	1	1
1	1	1	0	1	0	0	1

Result –

1) **Carry there is a carry so carry flag set to ‘1’.**
2) There are 4 no’s of 1’s in the result so parity flag is set to ‘1’.
3) There is a carry occurs in D3 to D4 so AC is set to ‘1’.
4) The result is a non-zero so zero flag is set to ‘1’.
5) The D7 bit of the result is 1 so sign flag is set to ‘1’.

Buses:



Schematic connection of memory and I/O devices to microprocessor

Various I/O devices and memory devices and memory devices are connected to a CPU by groups of lines called buses. There are 3 types of Buses i.e.

(1) Address Bus
(2) Data Bus
(3) Control Bus

The address bus carries the address of a memory location or I/O device that the CPU wants to access. When an address is sent by the CPU, all devices will connected through the address bus, receives this address and responds to that device and also receives the enable signal from the CPU. The address bus is unidirectional. In short address bus is called as A-bus.

The data and control bus is also called as D-bus and C – bus respectively. The data and control bus are bidirectional.

The data bus is used to transfer data between the processor, memory and I/O devices.

The control bus is used carry necessary control signals between the CPU and memory and I/O devices. Examples of control signals are $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$ etc.

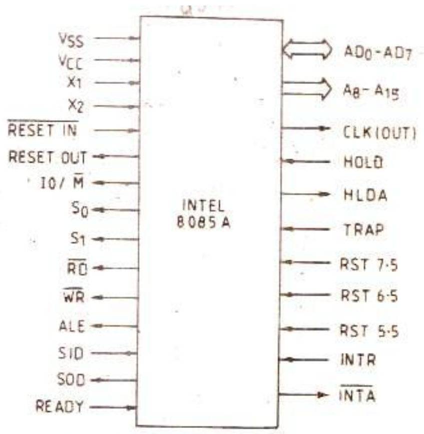
The microprocessor issues IO/$\overline{M}$ signal to indicate whether it will communicate with an I/O device or memory. If IO/$\overline{M}$ signal is high, CPU wants to communicate with an I/O device. If IO/$\overline{M}$ signal is law, the CPU wants to communicate with memory.

When the CPU sends a low $\overline{RD}$ signal, the activated devices understand that the CPU wants to read information.

The width of the data bus is same as the word length of the CPU. For example an 8-bit microprocessor has 8-bit data bus, a 16-bit microprocessor has 16-bit data bus, a 32-bit processor has 32-bit data bus and so on.

The width of the address bus is decided by the designed memory addressing capability of the CPU for example, Intel 8085 an 8-bit microprocessor and having 16-bit address bus which gives 64Kb memory addressing capability . A group of 8-bits is called as a byte.

PIN Configuration:



Schematic Diagram of Intel 8085.

A_{15} – A_8 and AD_0 – AD_7:

The address bus mean A15-A8 bit address bus means A_7-A_0, The LSB part and A_{15}-A_8 the MSB part.

As there are 2 buses available to transfer the address and data in the address bus A_{15}-A_8 (MSB) part. Each part of the address and data is passed to mp and AD_7-AD_0 is passed through data bus and mp.

A_0-A_{15} – 8 –bit

Data can't be sent without address.

So AD_0-AD_7 =A_0-A_7=8-bits

$\underline{+\ D_0\text{-}D7=8\text{-bits}}$

=AD_0-AD_7=8-bits =Bidirectional

A_0-A_7-LSB

A_8-A_{15}-MSB

A_8-A_{15} =Unidirectional

ALE:-

- Address latch enable signal.
- It is made high during the 1st clock cycle of the machine cycle.
- During this period it separates the lower bit address (A_7-A_0) and data (D_7-D_0) from AD_7-AD_0.
- When it is made high LSB part of the address is passed.
-

$\overline{RD}$:

It reads the control signal which is used for read operation when it is made low the read operation takes place.

$\overline{WR}$:

It is the control signal used for write operation when it is made low the write operation takes place.

I O/ $\overline{M}$:

It indicates that the mp doing I/P, O/P operation or memory operation.

If IO/$\overline{M}$ = 1 I/O operation
= 0 Memory operation

S1 and S0:

These are the status signal set by the mp to distinguish between read / write operation.

S_1	S_O	
0	0	Halt
1	0	Read
0	1	Write
1	1	Fetch Operation

X1 and X2:

These are the RC and LC network which is used to connect the crystal oscillator to the microprocessor.

Clock:

This signal is used for the system clock which is provided by the microprocessor.

Its frequency is same as the frequency of the mp.

INTR: This is an interrupt signal used by the I/p O/p device to transfer the data to the microprocessor after getting the interrupt signal.

$\overline{INTA}$: This is the interrupt acknowledgement signal given by the microprocessor to the i/p o/p device after getting the interrupt signal.

RST 6.5, 7.5, 5.5:

TRAP – 0024, 5.5 – 0026, 6.5 – 0034, 7.5 – 0036
These are called restart interrupt.
These are mask able signal which can be enable or disable.
This signal causes an internal restart to be automatically inserted.

TRAP:

These are non-maskable with highest priority.

HOLD (in):

It indicates that another device is requesting for the use of address and data bus.

HLDA (out):

It indicates that HOLD request has been received.

RESET (in):

This signal reset the programme counter to zero.
It also reset the HLDA counter.

RESET (out):

It indicates that microprocessor is already reset.

SID (in):

It is a data line for serial I/P. The data lines are loaded into accumulation when RIM instruction is executed.

SOD (out):

It is the data line for serial O/P. These data line are loaded into the accumulator where SIM instruction is executed.

VCC:

This pin indicates the power supply to the mp which is +5V.

VSS:

This pin indicates ground.

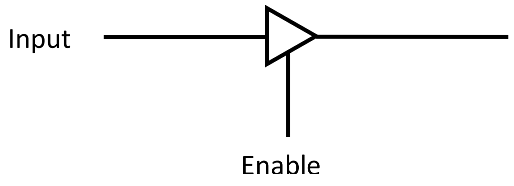
Three/Tri State Register:-

The tri state resister is a logic device having 3 states i.e. logic 0 state, logic 1state, high impedance state.

The term "tri state "is a trade mark of National Semiconductor and is used to represent 3 logic states.

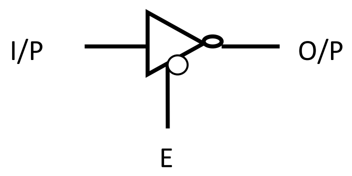
A tri state logic device having 3 lines, they are input, output and the Enable line. When enable line is activated the tri state devices functions the same way as an ordinary logic device. When the 3rd line (Enable) is disabled the logic device goes into High Impedance State. In this state the system is disconnected from the input and practically no current drawn from the system.

The symbolic representation of a tri state device is as follows

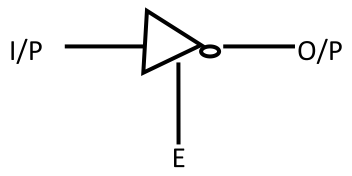


Tri-state Device

Different types of tri state devices we will found. They are tri state inverter, tri state buffer etc.



Tristate Inverter with enable LOW



Tristate Inverter with enable HIGH

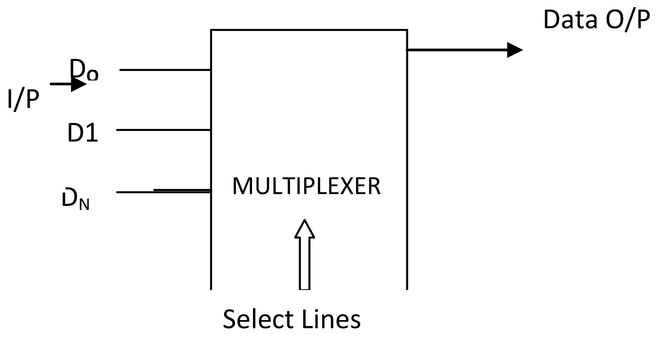
CONCEPT OF MULTIPLEXING:-

A multiplexer (MUX) is an electronics device that selects one of the several analog or digital input signals and forward the selected input into a single line.

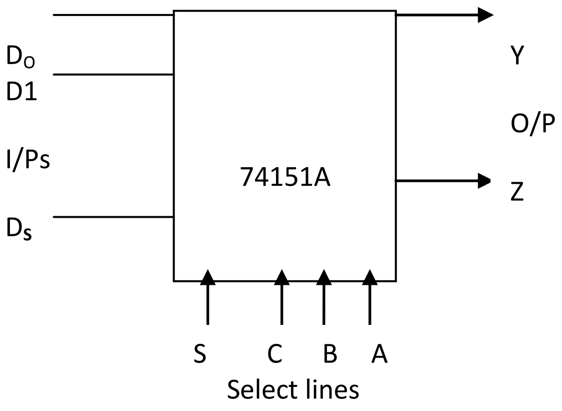
Multiplexer are mainly used to increase the amount of data that can be send over the network within a certain amount of time and bandwidth. It also called as a data selector.

So multiplexing is the process by which we can select one of several input signals and forward the selected input in to a single line.

Multiplexer having multi input and single output



Schematic diagram of a Multiplexer



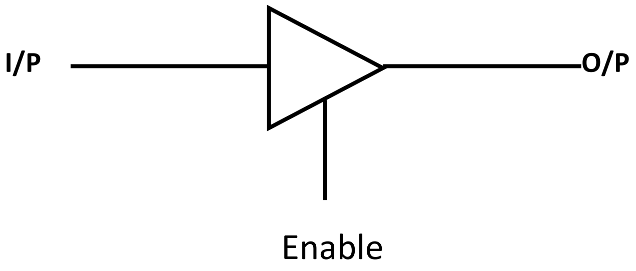
Schematic diagram of a 74151A Mux

74151A is a MUX having one out of eight lines. A,B and C are select lines.D_0-D_8 are input lines and S is the strobe line.

INPUT			OUTPUT		
C	B	A	S	Y	Z
0	0	0	0	D_0	$\overline{D_0}$
0	0	1	0	D_1	$\overline{D_1}$
0	1	0	0	D_2	$\overline{D_2}$
0	1	1	0	D_3	$\overline{D_3}$

DATA TRANSFER USING TRI STATE REGISTERS:

As we know that, the tri state logic devices having 3 states, these are logic 0,logic1 and High Impedance states.



When Enable=1 (HIGH)

On giving ‘0’ as input to input, the output becomes ‘0’ called as logic 0.

On giving ‘1’ the output becomes ‘1’ called as logic ‘1’.

When Enable=0 (LOW)

On giving ‘0/1’ to the input, there will be no effect on output .so this state is called High Impedance state. In this state practically no current flows through the system.

So the data transfer differs according to the device choose, whether it is inverter or Buffer.

APPLICATION OF TRI STATE DEVICES IN MULTIPLEXING OF BUSES:-

Generally tri state devices are essential to proper functioning of the bus oriented system in which same bus lines are shared by several components.

In microcomputer system peripherals are connected in parallel between the address bus and data bus. However because of the tri state interfacing devices, peripherals do not load the system buses.

Distinguishes between SPR and GPR:

Registers are of 2 types:-

General Purpose Registers
(B, C, D, E ,H and L)

Special Purpose Registers
= Accumulator
=Program counter
=Stack Pointer
=Instruction registers
=Flag registers.

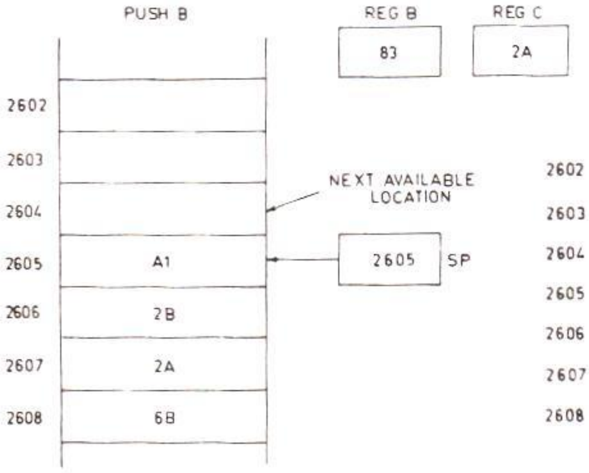
Stack:

During execution of a programme we need to save some of the contents of certain registers because those registers are required for other operation. These contents are moved to other location of memory by using PUSH operation. Then the registers are used for other operation.

After completing the operation again that contents are transferred back to its original location i.e. to the registers by POP operation. The memory location is set aside by the programmer at the beginning. A set of memory location is kept for this purpose is called stack. The last memory location occupied portion is called as the Stack top. Special 16-bit register known as Stack-Pointer holds the address of the stack top.

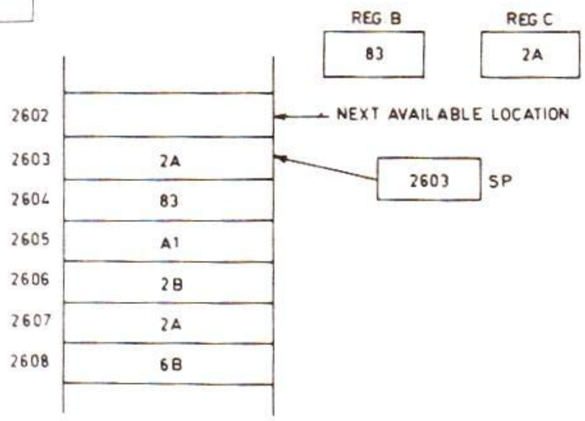
The stack pointer is initialized in the beginning of the programme by LXISP or SPHL instruction. Any area of the RAM can be used as stack. DATA are stored in the stack on LIFO (Last in fast out) Principle. Stack access are faster than memory access.

Before PUSH Operation



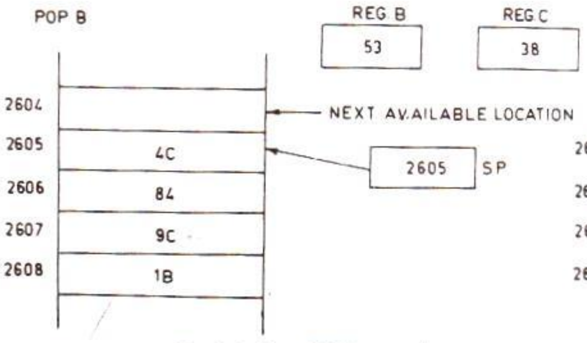
Stack before PUSH operation.

After PUSH Operation



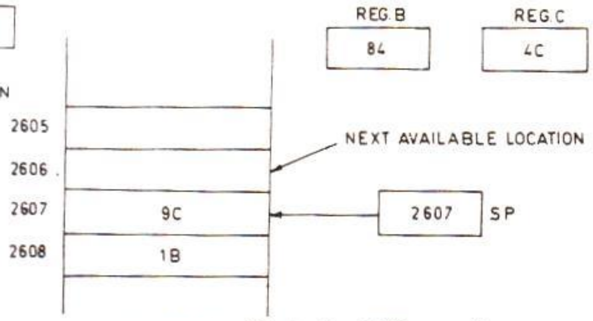
Stack after PUSH operation.

Before POP Operation



Stack before POP operation.

After POP Operation



Stack after POP operation.

****

(CHAPTER-3)

Instructions sets of Intel 8085

Introduction

An instruction is a command given to the computer to perform a specified operation on a given data. The instruction set of a microprocessor is the collection of instruction that the microprocessor is designed to execute.

DIFFERENT GROUPS OF INSTRUCTIONS:-

The programmer can write a programme in assembly language using these instructions .These instructions have been classified in to the following groups

1. Data transfer group

2. Arithmetic group

3. Logical group

4. Branch control group

5. I/O and machine control group

Data transfer group: Instructions which are used to transfer data from one register to other, register to memory, memory to register are comes under this group.exp: MOV, MVI, LXI, STA etc.

When an instruction is executed, the data are transferred from source to destination without altering the contents of the source.

Ex: MOV A, B. when this instruction executed, the contents of register B is stored to the register A, and the contents of register Remain unaltered.

Arithmetic group: This performs the arithmetic operation such as addition, Substraction, increment, decrements of the content of register or memory.

Ex: ADD, SUB, INR, DAD etc.

Logical group:

This group performs the logical operations such as AND, OR, compare, rotate etc.

Ex: ANA, XRA, ORA, CMP, RAL.

Branch control group: This group performs the conditional and unconditional jump, subroutine, call and return and

Ex: JMP, JC, JR, CALL.

I/O machine control group: This group performs the instructions for i/o ports, stack, and machine control. Ex: IN, OUT, PUPH, POP, HLT.

Instructions and Data Formats:

Intel 8085 is a 8 bit up. It handles 8 bit data at a time .One byte consists of 8 bits .A memory location for 8085 up is designed to accommodate 8 bit data. If 16 bit data are to be stored, they are stored in consecutive memory location is 16 bits. There are various techniques to specify data for instructions are: 1 .A 8 bit or 16 bit data may be directly given in the instructions itself.

2. T he address of the memory location, i/o port device, where data resides , may be given in the instruction itself.

3. In some instructions only one register is specified. The content of the register is one of the operands.

4. In some instructions specifies 2 register. The content of the register are the required data.

5. In some instructions data is implied .Most of this type instructions operate on the content of the accumulator.

Due to different ways of specifying data for instructions, the machine codes of all instructions are not of the same length.

There are 3 types of Intel 8085 instructions i.e1.single byte instruction

2. 2 byte instruction

3 . 3 byte instruction.

ADDERESSING MODES:-

The technique which is used to specify the data for instruction is called addressing mode.

The addressing mode incase of 8085 microprocessor is divided in to 5 types

1. Direct addressing mode.

2. Register addressing mode.

3. Register indirect addressing mode

4. Immediate addressing mode.

5. Implicit addressing mode.

1. .Direct addressing mode. In this addressing mode the address of the operand is directly given to the instruction.

STA2400. It means the data which is presents in the memory location 2400is stored to the accumulator .As the address of the above example is given directly in the instruction is direct addressing mode.

2. Register addressing mode: If one of the operand is in the general purpose register then the addressing mode of the instruction is called register addressing mode.

The pocode of the instruction gives the address of the instruction of the register where data is present.

Ex: MOV A,B Here data is general purpose of register B and pocode of the instruction is 78 .When we break the opcode in binary format(01111000) then last 3 zero(000) explains the address of the register B and (111) of the address of the accumulator and(01) the meaning of the operation.01-write&10-read.

3. Register indirect addressing mode: In this mode address of the operand is specified by a register pair. Ex; LXIH, 2500H

MOV A, M

HLT

Here address of the memory location is stored in the H-L pair register and the data which is in the memory is moved to the accumulator.

4. Immediate addressing mode: If the data is given to the instruction, then the addressing mode of the instruction is called immediate addressing mode.

Ex: MVI A, 05: Here 05 is the data is immediately moves to the accumulator.

5. Implicit addressing mode: There are certain instructions which operate only on the accumulator. The addressing mode which is used in this operation is called Implicit addressing mode.

Ex: CMA, RAR, RALETC.

Instructions of Intel 8085 microprocessor:

Data transfer group:

MOV r, M: Move the content of the memory to register.

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

Opcode: MOV B, M-46
MOV C, M-4E
MOV D, M-56
MOV E, M-5E
MOV H, M-66
MOV L, M-6E

MOV M,r: Move the content of register s to memory.

Opcode:

MOV M, B-70
MOV M, C-71
MOV M, D-72
MOV M, E-73
MOV M, H-74
MOV M, L-75
MOV M, A-77

MVI r, data: move immediate data to the register.

Opcode:

MVI B-06
MVI C-OE
MVI D-16
MVI E-1E
MVI H-26
MVI A-3E

MVI M, data- move immediate data to the memory.

Opcode-

MVI M-36

LXI rp, data 16: Load register pair immediate.

Load 16 bit immediate data into register pair microprocessor.

Opcode-

LXI D-11
LXI-21

LDA addr; Load accumulator direct.

Opcode-

LDA-3A

STA addr: Store accumulator direct.

Opcode-

STA-32

LHLD: Load H-L pair direct.

The content of memory location is loaded to the register.

The content of the next memory location is loaded to the H register.

Opcode:

LHLD-2A

SHLD addr: store H-L pair direct.

The contents of register L is stored in the memory location.

The content of the register H is stored in the next memory location.

Opcode:

SHLD-22

LDAX rp: Load accumulator indirect

The content of the memory location, whose address is in the register pair rp is loaded in the accumulator.

Opcode:

LDAX B-OA
LDAX D-1A

STAX rp-store accumulator indirect.

Opcode:

STAX B-02
STAX D-12

XCHG: exchange the content of H-L pair with D-E pair.

Opcode:

XCHG-EB

Arithmetic group:

ADD r: add register to accumulator.

Opcode:

ADD B-80
ADD C-81
ADD D-82
ADD E-83
ADD H-84
ADD L-85

ADD M: Add memory to accumulator

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

Opcode:

ADD M-86

ADC r: Add register with carry to accumulator.

Opcode:

ADC B-88
ADC C-89
ADC D-8A
ADC E-8B
ADC H-8C
ADC L-8D

ADC M: Add memory with carry to accumulator

Opcode:

ADC M-8E

ADI data: Add immediate data to accumulator

Opcode:

ADI –C6

DAD rp: Add register pair to H L pair

Add the content of the register pair to the H-L pair and the result is stored in H-L pair

Opcode:

DAD B-09
DAD D-19
DAD H-29

SUB r: Substract register from accumulator

Opcode:

SBB A -97
SBB B-90
SBB C-91
SBB D-92

SBB E-93
SBB H -94
SBB L -95

SBB M: Substract memory from accumulator

Opcode:

SBB M-96

SBB r: Substract register from accumulator with borrow

The content of register r carry status are substracted from the content of the accumulator

Opcode:

SBB A-97
SBB B-98
SBB C-99
SBB D-9A
SBB E-9B
SBB H-9C
SBB L-9D

SBB M: Substract memory from accumulator with borrow

Opcode:

SBB M-9E

SUI data: Substract immediate data from accumulator

It is subtracted from the accumulator

Opcode:

SUI-D6

SBI data: Subtract immediate data from accumulator with borrow

Here the data and carry status are subtracted from the accumulator

Opcode:

SBI-DE

INR r: Increment register content

Opcode:

INR A-3C
INR B-04
INR C-0C
INR D-14
INR E-1C
INR H-24
INR L-2C

INR M: Increment the memory content

Opcode:

INR M-34

DCR r : Decrement the register content

Opcode:

DCR A-3D
DCR B-05
DCR C-0D
DCR D-15
DCR E-1D
DCR H-25
DCR L-2D

DCR M: Decrement memory content

Opcode:

DCR M-35

INX rp : Increment register pair

Opcode:

INX B-03
INX D-13
INX H-23

DCX rp: Decrement registers pair

Opcode:

DCX B-0B
DCX D-1B
DCX H-2B

DAA: Decimal adjust accumulator

It is used for the instruction ADD, ADI , ACI, ADC etc

Here the result is in Hexadecimal form and to this instruction BAA is used to convert the result into decimal form

Opcode:

DAA -27

LOGICAL GROUP:

ANA r: AND register with accumulator

The content of the register R is ANDed with the content of accumulator

Opcode:

ANA B-A0
ANA C-A1
ANA D-A2
ANA E-A3
ANA H-A4
ANA L-A5
ANA A-A7

ANA M: AND memory with accumulator

Opcode :

ANA M-A6

ANI data: AND immediate data with accumulator

OPCODE:

ANI E6

ORA r: OR register with accumulator

Opcode :

ORA B-B0
ORA C-B1
ORA D-B2
ORA E-B3
ORA H-B4
ORA L-B5
ORA A-B7

ORA M OR memory with accumulator

Opcode:

ORA M -B6

ORI data: OR immediate data with accumulator

Opcode:

ORI-F6

XRA r: Execute OR register with accumulator

Opcode:

XRA B-A8
XRA C-A9
XRA D-AA
XRA E-AB
XRA H-AC
XRA L-AD
XRA A-AF

XRA M: Execute OR memory with accumulator

Opcode:

XRA M-AE

XRI data : Execute OR immediate data with accumulator

Opcode:

XRI –EE

CMA: Complement the accumulator

Opcode:

CMA-2F

CMC: Complement the carry status

Opcode:

CMC-3F

STC: Set carry status

Opcode:

STC-37

CMP r: Compare register with accumulator

Opcode:

CMP B-B8
CMP C-B9
CMP D-BA
CMP E-BB
CMP H-BC
CMP L-BD
CMP A-BF

CMP M: Compare memory with accumulator

Opcode:

CMP M-BE

CPI data : Compare immediate data with accumulator

Opcode:

CPI –FE

RLC: Rotate accumulator left

Opcode:

RLC -07

RRC: Rotate accumulator right

Opcode:

RRC –OF

RAL: Rotate accumulator left through carry

Opcode:

RAL D-17

RAR: Rotate accumulator right through carry

Opcode :

RAR -1F

Branch group:

JMP addr : Unconditional jump :jump to the instruction specified by the address

Opcode:

JMP-C3

Conditional jump adds:

JZ addr: Jump if the result is zero

Opcode:

jz-ca

JNZ adds: jump if the result is not zero

Opcode:

JNZ-C2

JC addr : jump if there is carry

Opcode:

JC -DA

JNC addr: jump if there is no carry

Opcode:

JNC-D2

JP addr :Jump if result is plus

Opcode:

JP-F2

JM addr: Jump if result is minus

Opcode:

JM-FA

JPE addr: jump if even parity

Opcode:

JPE-EA

JPO addr: jump if odd parity

Opcode :

JPO-E2

CALL addr: Unconditional call

Opcode:

call-CD

Conditional call:

CC addr: call if carry status =1

Opcode:

CC-DC

CNC addr: call if carry status =0

Opcode:

CNC-D4

CZ addr: if the result is zero

Opcode:

CNZ-CC

CNZ addr: if the result is not zero

Opcode:

CNZ-C4

CP addr: if the result is plus

Opcode:

CP-F4

CM addr: if the result is minus

Opcode:

CM –FC

CPE addr: IF there is add even number of one

Opcode:

CPE-EC

CPO addr: IF there is add odd number of one

Opcode:

CPO-E4

Conditional return: RC:IF CARRY STATUS IS 1

Opcode:

RC-D8

RNC: if carry status is zero

Opcode:

RNC-D0

RZ: if result is zero

Opcode:

RZ-C8

RNZ : if result is not zero

Opcode:

RNZ-C0

RP : If the result is plus

Opcode:

RP-F0

RM: If the result is minus

Opcode:

RM-F8

RST: Restart

Opcode:

RST 0-C7
RST 1-CF
RST 2-C7
RST 3-CF
RST 4-D7
RST 5-E7
RST 6-F7
RST 7P-FF

PCHL:-Jump to address specified by H-L Pair

Op code –

PCHL-E9

Stack, I/O and machine control group :

PUSH rp : Push the content of register pair to stack.

Op code:

PUSH B-C5
PUSH D-D5
PUSH H-E5

PUSH psw: Push process status word.

Op code-

PUSH Psw-F5

PUSH rp : Pop the content of register pair ,which was saved ,from the stack

Op code-

POP B-C1
POP-D1
POP-E1

PUSH psw: Pop process status word.

Op code –

PUSH Psw-F1

HLT: Halt

Op code- HLT-F6

XTHL: Exchange stack –top with H L pair

Opcode:

XTHL:-E3

SPHL: Move the content of H L pair to stack pointer

Opcode:

SPHL-F9

EI: Enable interrupt

Opcode:

EI-FB

DI :Disable interrupt
Opcode:
DI-F3

SIM: Set interrupt masks
Opcode:
SIM-30

RIM: Read interrupt masks
Opcode:
RIM-20

NOP: NO operation
Opcode:
NOP-00

BASIC ASSEMBLER DIRECTIVES:-

For easy understanding and error free, programmers developed a language called assembly language. It is easy to write, meaningful and easily rememberable. Ex: ADD for addition, SUB for substraction etc. Such symbols are called Mnemonics.

A program written in Mnemonics is known as assembly programs. In assembly language corresponding to one Mnemonics. There is only one machine code. But in the High level language there is a number of machine codes. The program written other languages is translated before executing it. The translation done by the help of s/w.

A program which translates an assembly language programmer into a machine language program is called an assembler.

A self assembler or resident assembler is an assembler which runs on the microcomputer for which it produces object codes. A program can be developed conveniently in short time on a large computer. When a program is developed on other computer, a cross assembler is required.

A cross assembler is an assembler that runs a computer other than for which it produces object codes.

ONE PASS ASSEMBLER AND TWO PASS ASSEMBLE:-

An assembler which goes through an assembly language program only once is known as one pass assembler. Such an assembler must have some technique to take the forward reference into account. It is faster.

It does not provide many features.

An assembler which goes through an assembly language twice is called two pass assembler. Such an assembler does not face any difficulty with forward references. During second pass it produces the machine code for each instruction. Two pass assembler are mostly used.

ADVANTAGES OF ASSEMBLY LANGUAGE:-

1. Small to moderate size of program
2. Real time control application
3. Computation time is less.
4. It is faster to produce result.
5. Easy to understand.

****

CHAPTER-4

PROGRAMING TECHNICS

INTRODUCTION:-

The sequence of instruction called program. A set of programs written for a particular computer is known as s/w for that computer. The program is stored in RAM. The CPU takes one instruction of the program at a time from the RAM and executes it. It executes all the instruction of the program one by one to produce the desired result.

A computer used the binary digits for its operation. Hence the instructions are coded and stored in the memory in the form of zeroes and ones. The program written in the form of 0's and 1's are called machine language programs.

The demerits of machine language programs are:

1. It is very difficult to understand.
2. Since each bit has to be entered individually, the entry of program is very slow.
3. Programs are long.
4. Programs writing is difficult.
5. Chances of errors.

PROGRAMS:-

1. Write a program to find 1's complement of an 8-bit number.

96H = 1001 0110

1's complement = 0110 1001 = 69H

The number is placed in the memory location 2501H

The result is stored in memory location 2502H.

DATA

2501----96H

Result

2502-----69H

2. Write a program to find 2's complement of an 8-bit number.

96 = 1001 0110

1's complement = 0110 1001

2's complement = 0110 1010 = 6A

The number is placed in memory location 2501H

The result is to be stored in memory location 2502H

Address	Machine code	Mnemonics	Operands	Comments
2000	3A,01,25	LDA	2501H	Get data in accumulator
2003	2F	CMA		Take its 1’s complement
2004	3C	INR	A	Take 2’s complement
2005	32, 02, 25	STA	2502H	Store result in 2502H
2008	76	HLT		Stop

DATA

2501----96H

Result

2502-----6AH

3. Write a program to Mask off least significant 4-bit of an 8-bit number.

Number = A6 = 1010 0110

Result = A0 = 1010 0000

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

DATA

2501----A6

Result

2502-----A0

4. Write a program to Mask off most significant 4-bit of an 8-bit number.

Number = A6 = 1010 0110

Result = 06 = 0000 0110

Address	Machine code	Mnemonics	Operands	Comments
2000	3A,01,25	LDA	2501H	Get data in accumulator
2003	E6,F0	ANI	0F	Mask off the most significant 4bits
2005	32,02,25	STA	2502H	Store result in 2502H
2008	76	HLT		stop

DATA
2501----A6
Result
2502-----06

5. Write a program to find the largest number in a data array.

Number = 98, 75 and 99

Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,00,25		LXI	H,2500H	Address for count in H-L pair
2003	4E		MOV	C,M	Count in register C
2004	23		INX	H	Address of 1st number in H-L pair
2005	7E		MOV	A,M	1st Number 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 previous maximum. Is next number > 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 number 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.

DATA
2500----03
2501-98
2502-75
2503-99
Result
2450-99

6. Write a program to find the smallest number in a data array.

The numbers of a series are: 86,58 and 75.

Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,00,25		LXI	H,2500H	Address for count in H-L pair
2003	4E		MOV	C,M	Count in register C
2004	23		INX	H	Address of 1st number in H-L pair
2005	7E		MOV	A,M	1st Number in accumulator
2006	0D		DCR	C	Decrement count
2007	23	LOOP	INX	H	Address of next number in H-L pair
2008	BE		CMP	M	compare next number with previous smallest. Is previous smallest less then next number
2009	DA,0D,20		JC	AHEAD	Yes, smallest number in accumulator. Go to the AHEAD.
200C	7E		MOV	A,M	NO, get next number in accumulator
200D	0D	AHEAD	DCR	C	Decrement count
200E	C2,07,20		JNZ	LOOP	
2011	32,50,24		STA	2450H	Store smallest number in 2450H
2014	76		HLT		Stop.

DATA
2500----03H

2501-86H

2502-58H

2503-75H

Result

2450---58H

7. Write a program to find the smaller of two number.

The numbers are 84H and 99H

Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,01,25		LXI	H,2501H	Address of 1st number in H-L pair
2003	7E		MOV	A,M	1ST number in accumulator
2004	23		INX	H	Address of 2nd number in H-L pair
2005	BE		CMP	M	Compare 2nd number with 1st.Is 1st number < 2nd number.
2006	DA,0A,20		JC	AHEAD	Yes, smaller number is in accumulator. Go to AHEAD.
2009	7E		MOV	A,M	No, get 2nd number in accumulator.
200A	32,03,25	AHEAD	STA	2503H	Store smaller number in 2503H
200D	76			HLT	stop.

DATA

2501----84H

2502—99H

Result:

2503---84H

8. Write a program to find the larger of two numbers.

The numbers are 98H and 87H.

Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,01,25		LXI	H,2501H	Address of 1st number in H-L pair
2003	7E		MOV	A,M	1ST number in accumulator
2004	23		INX	H	Address of 2nd number in H-L pair
2005	BE		CMP	M	Compare 2nd number with 1st.Is 2nd number > 1st number.
2006	D2,0A,20		JNC	AHEAD	No, larger number is in accumulator. Go to AHEAD.
2009	7E		MOV	A,M	Yes, get 2nd number in accumulator.
200A	32,03,25	AHEAD	STA	2503H	Store larger number in 2503H
200D	76		HLT		stop.

DATA
2501----98H
2502—87H
Result:
2503---98H

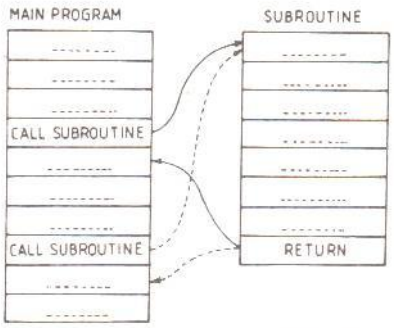
9. Write program to move a block of data from one section of memory to another section of memory.

Address	Machine Code	Mnemonics	Comments
2400	21,00,20	LXI H,2000H	Get memory address of count.
2403	4E	MOV C,M	Count in register C.
2404	23	INX H	Source address of data in H-L pair.
2405	11,01,22	LXI D 2201	Destiny address of data in D-E pair.
2408	7E LOOP	MOV A,M	Data from source address to ACC.
2409	EB	XCHG	Destiny address in H-L pair.
240A	77	MOV M,A	Data to Destiny address .
240B	EB	XCHG	Source address of data in H-L pair
240C	23	INX H	Source address of next data.
240D	13	INX D	Destiny address of next data.
240E	0D	DCR C	Decrement count.
240F	C2,08,24	JNZ LOOP	Jump to label LOOP.
2412	76	HLT	Stop.

DATA:	**RESULT:**
2000—05	2201---01
2001—01	2202---02
2002---02	2203---03
2003---03	2204---04
2004---04	2205---05
2005---05	

SUBROUTINES:-

A subroutine is a group of instructions written separately from the main program to perform a function that occurs repeatedly in main program. Instead of writing a subprogram repeatedly in main program, simply write it once in another location. Whenever needed by the main program we can call it. This technique overcomes the repetition of a subprogram in main program.



CALL RETURN structure of subroutine.

****

CHAPTER-5

TIMING DIAGRAMS

CONCEPT OF TIMING DIAGRAM:-

An instruction is a command given to the computer to perform a specific operation. The sequence of instructions is called as a program. Program and data are stored in the CPU fetches one instruction from the memory at a time and executes it.

To fetch the instruction CPU constitutes an instruction cycle consists of a fetch cycle and execute cycle.

In fetch cycle the CPU fetch opcode from the memory. The necessary steps which are carried out to fetch an opcode from the memory, constitutes a fetch cycle.

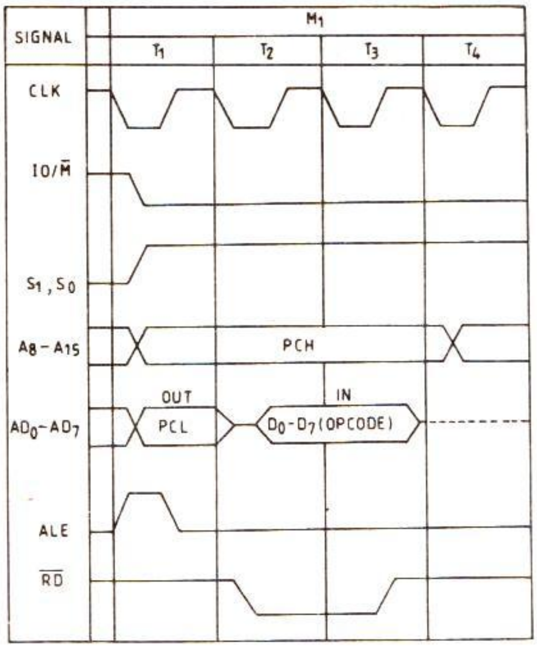
The necessary steps which are carried out to get data, if any from the memory and to perform the specific operation specified in an instruction, constitutes an execute cycle.

The necessary steps carried out tin a machine cycle can be represent graphically. The graphical representation of machine cycle is called as Timing Diagram.

The necessary steps carried out to perform the operation of accessing either memory or I/O device, constitute a machine cycle. In other words the necessary steps carried out to perform a fetch, a read or a write operation is called Machine cycle.

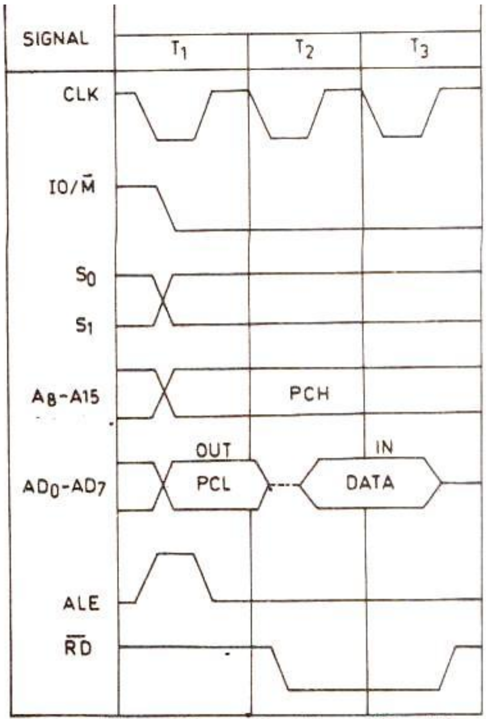
The subdivision of an operation performed in one clock cycle is called T-state. For fetch timing diagram the no. of T-state is 4 and if the data is in register pair, than the no. of T-state are 6.

1. Timing Diagram for Opcode fetches operation.



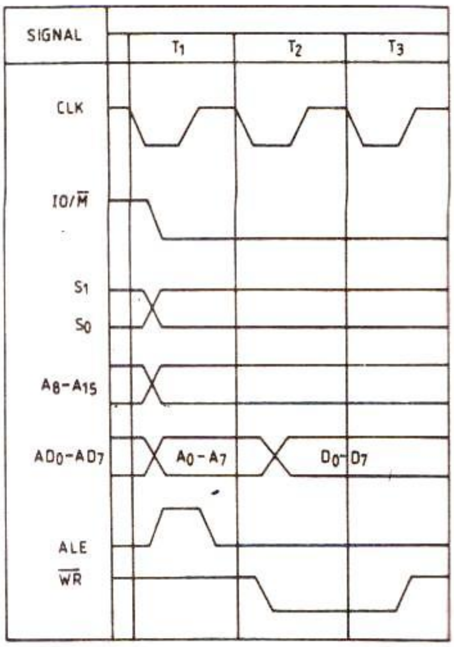
Timing Diagram for Opcode Fetch Operation.

2. Timing Diagram for Memory Read.



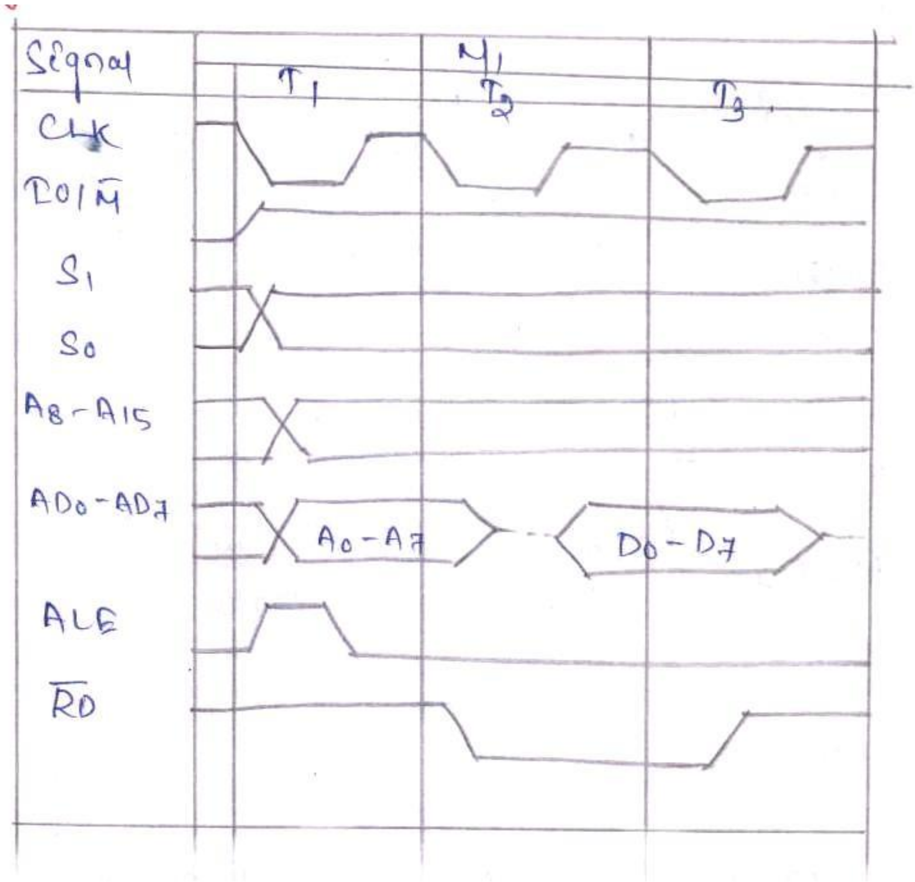
Timing Diagram for Memory Read Operation.

3. Timing Diagram for Memory Write.

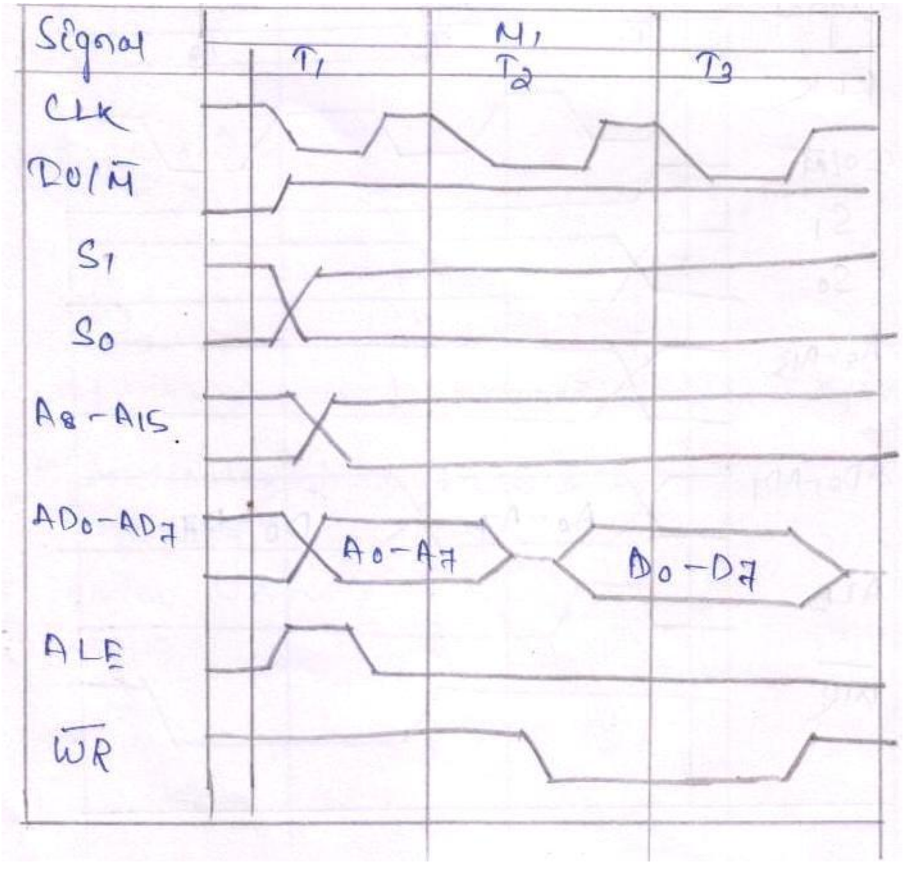


Timing Diagram for Memory Write Operation.

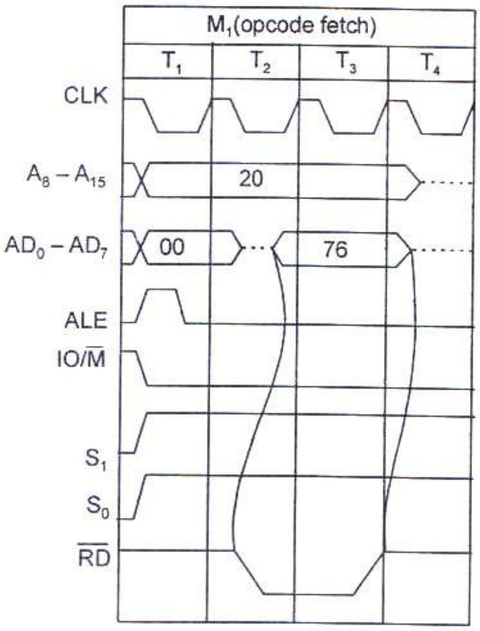
4. Timing Diagram for I/O Read.



5. Timing Diagram for I/O Write.

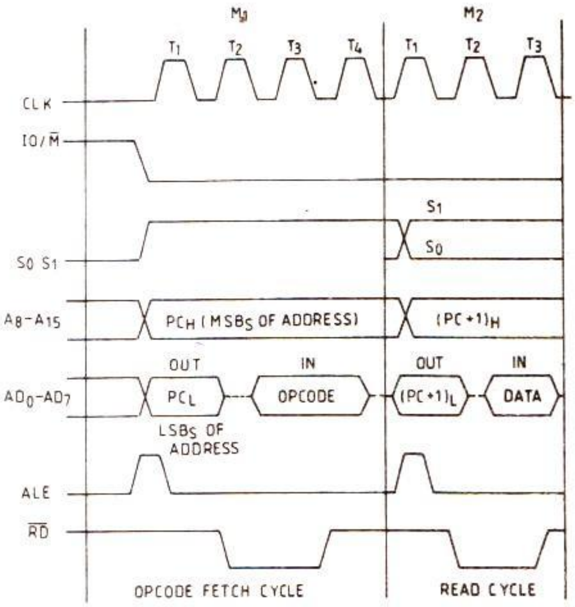


6. Timing Diagram for MOV A, B.



Timing diagram for MOV A, B

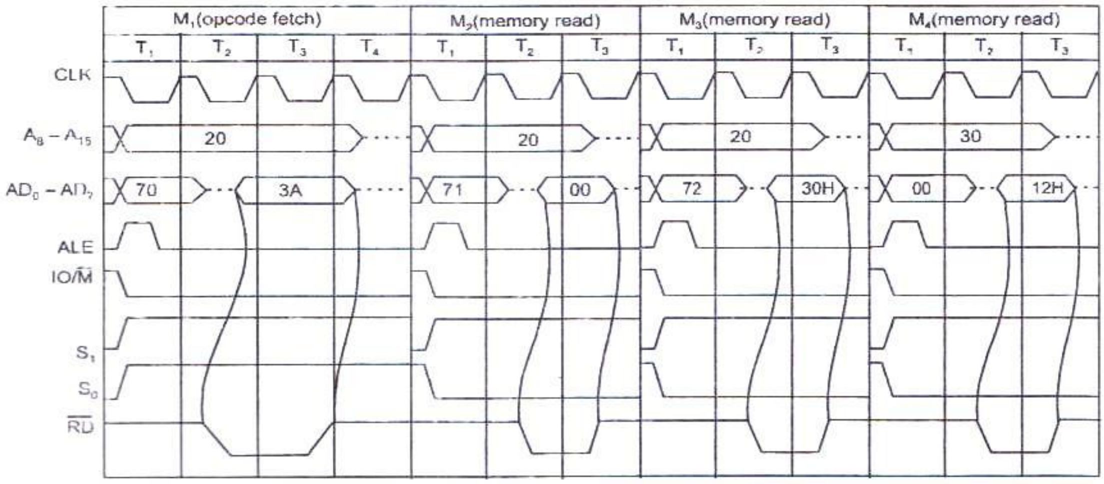
7. Timing Diagram for MVI r, Data.



Timing Diagram for MVI r, Data

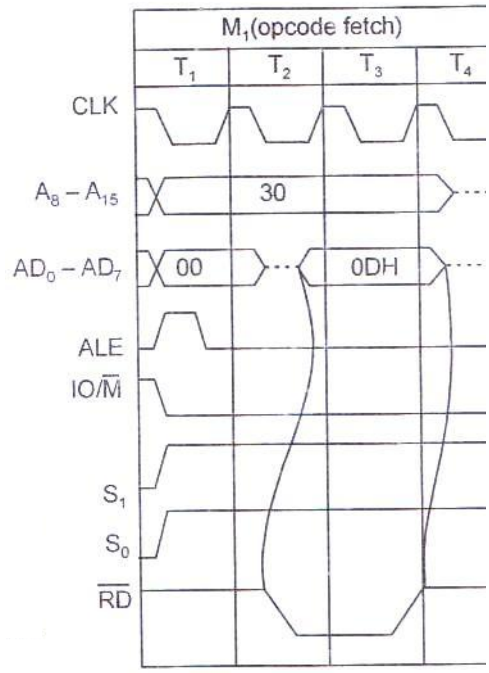
8. Timing Diagram for LDA 3000H.

Timing diagram



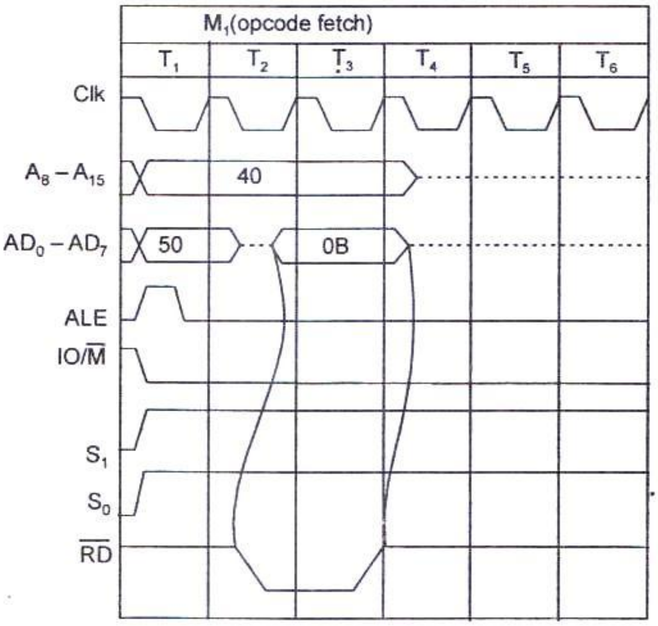
Timing diagram for LDA 3000H

9. Timing Diagram for DCR C.



Timing diagram for DC

10. Timing Diagram for DCX B.



Timing diagram for DCX B

****

INTERFACING I/O AND MEMORY PROGRAMMING

INTRODUCTION:

Some addresses are assigned to memories and some addresses to input device. Each memory location is assigned by an address.

Suppose that memory location are assigned that the address 2000 to 24FF. One address is assigned to one memory location. Any one of these address is not assigned to on input device.

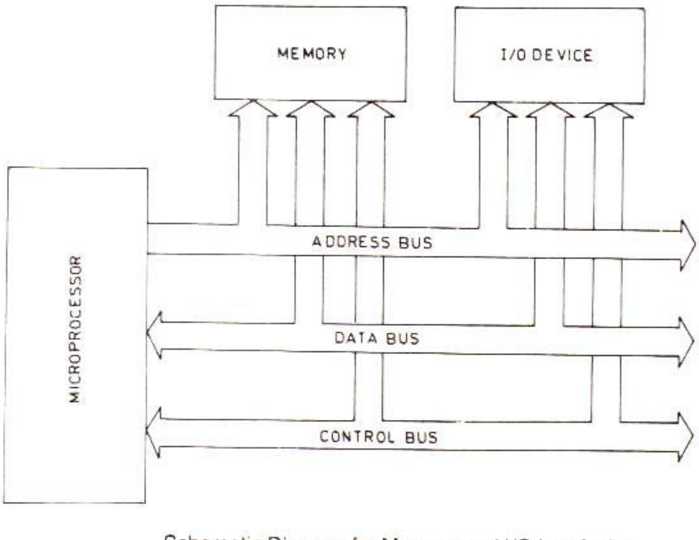
The address which is not assigned to memory is assigned to input devices. Ex: 2500, 2501, 2502, and 2503... etc. and one address is assigned to each input device. In this scheme all the data transform instruction of the microprocessor can be used for data memory as well as input devices. Ex: MOV A,M This is valid for data transfer from the memory to accumulator or form input device to accumulator. The memory mapped I/P scheme is suitable for a small system.

INPUT MAPPED I/P SCHEME:-

In this scheme the addresses assigned to memory location can also be assigned to input devices. Since the same address may be assigned to memory location or an input device, the microprocessor must issue a signal to distinguish whether the address on the Bus is for the memory location or I/O device. For this purpose Intel 8085 microprocessor issues on (IO/$\overline{M}$) signal for this purpose, when this signal is high it indicates that it is for I/O device, and when this signal is low, it indicates that it is for memory device and two extra instruction input and output are used to address I/O device. The "IN" instruction is used to read the data of an input device. The "'OUT" instruction is used to send the data to an output device. This scheme is suitable for a large system.

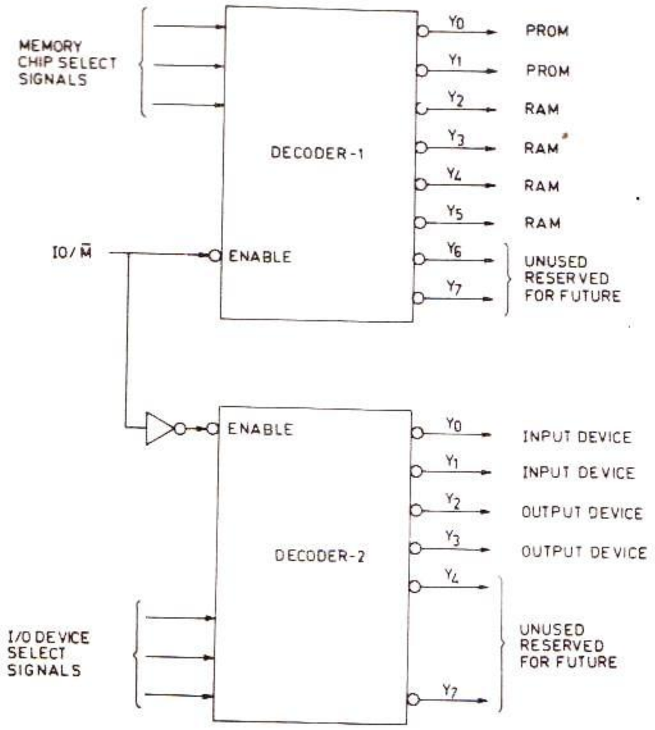
MEMORY AND I/O INTERFACING:

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



Schematic Diagram for Memory and I/O Interfacing.

MEMORY INTERFACING:-



Interfacing of Memory and I/O Devices.

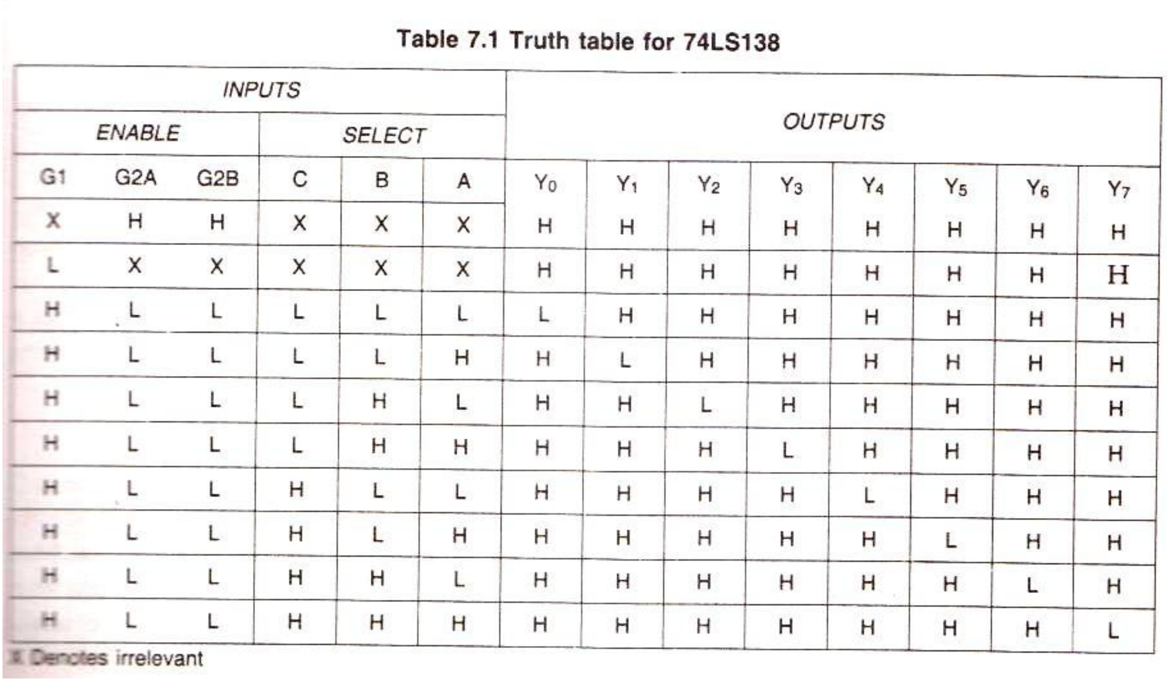
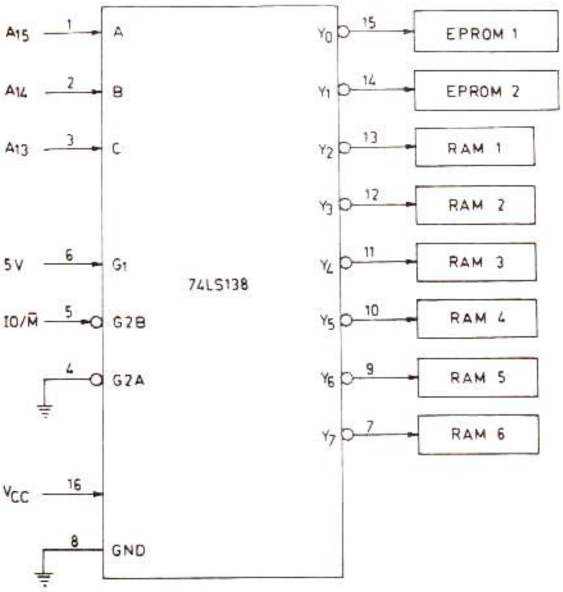


Table 7.1 Truth table for 74LS138

INPUTS						OUTPUTS							
ENABLE			SELECT										
G1	G2A	G2B	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	H	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

X Denotes irrelevant



Interfacing of Memory Chips using 74LS138.

Memory Locations for Various Zones

Decoder Output	*Memory Device*	*Zones of the Address Space*	*Memory Locations Address*
Y_0	EPROM 1	ZONE 0	0000 to 1FFF
Y_1	EPROM 2	ZONE 1	2000 to 3FFF
Y_2	RAM 1	ZONE 2	4000 to 5FFF
Y_3	RAM 2	ZONE 3	6000 to 7FFF
Y_4	RAM 3	ZONE 4	8000 to 9FFF
Y_5	RAM 4	ZONE 5	A000 to BFFF
Y_6	RAM 5	ZONE 6	C000 to DFFF
Y_7	RAM 6	ZONE 7	E000 to FFFF

Here to decoders are employed i.e. decoder 1 and decoder 2.If the (IO/$\overline{M}$) is high the decoder 2 is activated and required I/O device is selected. If (IO/$\overline{M}$) is low, decoder 1 is activated and required memory chip will be selected.

The address of a memory location or an I/O device is sent by the microprocessor, the corresponding memory chip or I/O device is selected by the decoding circuit the decoder task is performed by a bipolar PROM, PLA(Programmable logic array),Comparator etc. IN this section we are using an interface of memory chip through 74LS138.

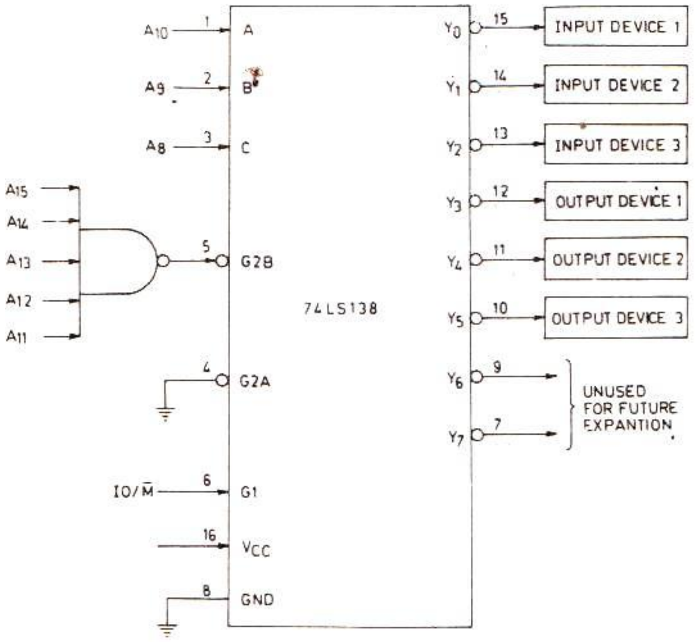
G1, G2A and G2B are the enable signals A,B,C are the selected lines by Appling the proper logic to select a line one output can be selected. i.e. Y_0, Y1, Y2....Y_7. These are the 8 output lines. When it is selected it goes low.

From the truth table when the G1 is low or G2A is high or G2B is high all output lines are high and similarly when G1 is high and G2A and G2B are low it act as a decoder.

The memory location for EPROM 1 will be lie in the range of 0000 to 1FFF, these are the memory location for ZONE 0 for the memory chip which is connected to the output line Y_0 of the decoder. Similarly for other ZONES.

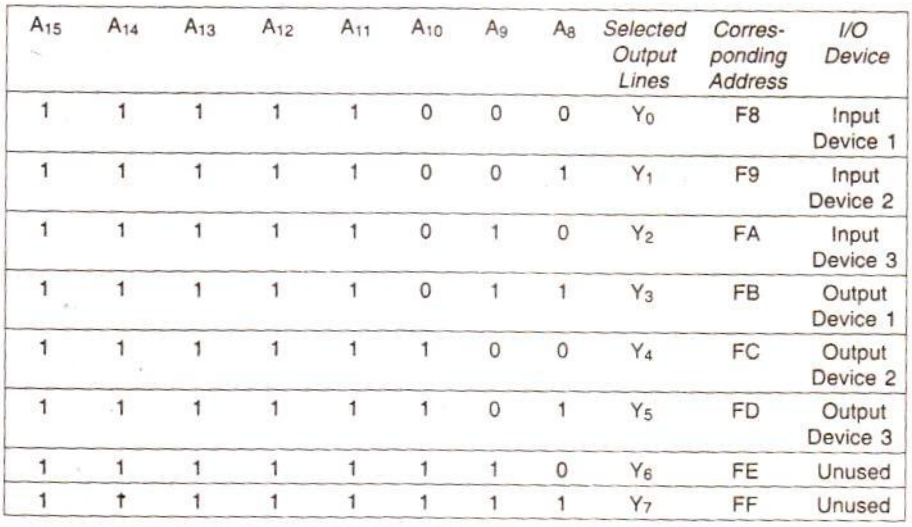
The inter memory address (64kb for 8085) has been divided into 8 ZONES address lines *A*15, A14,A13 are applied to selected the lines A,B,C of the 74LS138. The logic applied to these select a particular memory device, an EPROM or a RAM other address lines are $A_0, A_1 \ldots . A_{12}$ goes directly to the chip IO/$\overline{M}$ is connected to G2B when IO/M goes low it is to memory read or write operation. G1 is connected to +5V_{dc} is supply and G2A is grounded.

I/O INTERFACING:-



Interfacing of I/O Devices Using 74LS138.

Address of I/O Devices connected to 74LS138



A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	Selected Output Lines	Corres-ponding Address	I/O Device
1	1	1	1	1	0	0	0	Y_0	F8	Input Device 1
1	1	1	1	1	0	0	1	Y_1	F9	Input Device 2
1	1	1	1	1	0	1	0	Y_2	FA	Input Device 3
1	1	1	1	1	0	1	1	Y_3	FB	Output Device 1
1	1	1	1	1	1	0	0	Y_4	FC	Output Device 2
1	1	1	1	1	1	0	1	Y_5	FD	Output Device 3
1	1	1	1	1	1	1	0	Y_6	FE	Unused
1	†	1	1	1	1	1	1	Y_7	FF	Unused

This is the interface of I/O device through decoder 74LS138.

As the address of an I/O device is 8 bits so only A8-A15 lines of address Bus are used for I/O addressing.

The address lines A8, A9, A10 are used as select lines A, B and C of the decoder.

The address lines of A11 –A15 are applied to the G2B through NAND gate. When all the address lines (A11-A15) are high, the G2B is low.

IO/$\overline{M}$ is applied to G1 when it goes high and is indicates for I/O read or I/O write operation.

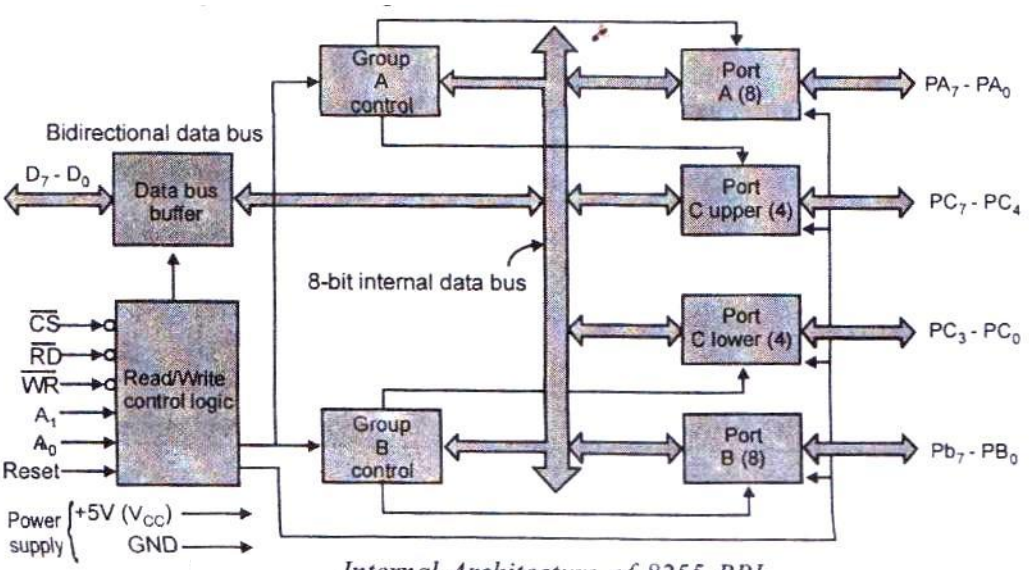
ARCHITECTURE OF 8255(PPI) :-

PPI= Programmable Peripheral Interface.

8255 PPI chip is also called as parallel input output port chip.

There are 24 I/P –O/P lines which can be individually programmed in two groups of 12 lines each these I/O lines are arranged as 3 ports i.e. port A, port B and port C. There are 2 individual groups of I/O pins called as group A and group B.

All the ports of 8255 can function independently as input or output by programming the bits of an internal register of 8255 called controlled word register (CWR).



Internal Architecture of 8255 PPI.

The functional block diagram contains

1. Data Bus buffer
2. Read/Write control logic
3. Group A and Group B control.
4. Port A, Port B, Port C.

DATA BUS BUFFER :-

It is 8 bit bidirectional data bus buffer it is used to interface the 8255 data bus with the system data bus.

Its outer pins are $D_0 – D_7$ and it is connected to system data bus. The direction of the data bus is to decided by the read/ write control signals. In read operation it transmits data to the system data bus and in write operation it receives data from the system data bus.

READ/ WRITE CONTROL LOGIC :

This block's function is to accept i/ps from system control bus and address bus.

Here the control signals like $\overline{RD}$,$\overline{WR}$, $\overline{CS}$ and address signals Ao, A1 are used.

Among these 5 signals $\overline{RD}$ and $\overline{WR}$ are connected to $\overline{IOR}$, $\overline{IOW}$ or $\overline{MEMR}$, $\overline{MEMW}$ depending upon the mapping Ao, A1 are directly connected to address lines Ao, A1 of the system address lines. The selection of 8255 is enabled or disenabled by $\overline{CS}$ signal. If $\overline{CS}$=0 the 8255 is selected and if $\overline{CS}$=1 8255 is rejected.

GROUP A AND GROUP B CONTROL:

8255 IS DIVIDED INTO 2 GROUPS I.E. Group-A and Group-B. Group-A consists of port A and port cupper and Group-B consists of Port B and port C lower. That means each group consists of 12 pins. The selection of ports are done by mode operation.

Mode Operation-	Mode 0	-	Simple I/O	-(Port A,B,C)
	Mode 1	-	Stroved I/O	-(Port A, B)
	Mode 2	-	Bidirectional Port	- (Port C)

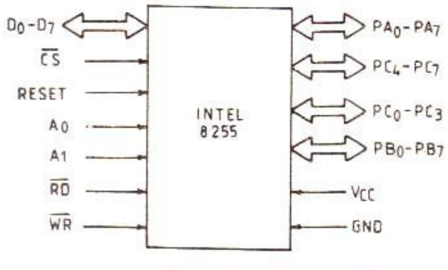
PORT A, PORT B AND PORT C :

The ports of 8255 PPI are Port A, Port B and Port C each port consists of an 8-bit data I/P buffer. The function of port A, Port B and Port C are decided by the control bit pattern of control word Register (CWR). The port C is divided into 2 groups PC upper and PC lower. Port C pins can be used as simple i/o, hand shake signals and status signals. It is used for coordination between port A and Port B.

A_1	A_o	Port/ Register Selection
0	0	Port A
0	1	Port B
1	0	Port C
1	1	CWR (Control Word Register)

PIN CONFIGURATION OF 8255 PPI CHIP :

Intel 8255 is a 40 Pin I.C. package. It operates on a single 5 vdc supply



Schematic Diagram of Intel 8255 A.

$D_0 – D_7$:

Pins = 27 to 34

Types= i/p - o/p

These are the 8-bit bidirectional data bus lines which are used to carry data or control word to / from the microprocessor.

$\overline{CS}$:

Pins = 6

Types = i/p

These are above low signals which is used to select 8255.

If $\overline{CS}$ = 0 =8255 is selected

$\overline{CS}$ = 1 =8255 is rejected

$\overline{RD}$:

Pin = 5

Types = i/p

It is a active low signal. It is used to send the data or status information to the mp on the data bus.

$\overline{WR}$:

Pin = 36

Types = I/P

This is also an active low signal. It is used to write data or control words into the 8255 PPI.

A1 and Ao :

Pin = 8 and 9

Types = I/P

It is used to select the different ports and control word Register (CWR).

The operation of $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ are :

$\overline{RD}$	$\overline{WR}$	**A1**	**Ao**	$\overline{CS}$	**I/P operation (Read)**
0	1	0	0	0	Port A to data bus
0	1	0	1	0	Port B to data bus
0	1	1	0	0	Port C to data bus
0	1	1	1	0	Not allowed

$\overline{RD}$	$\overline{WR}$	A1	Ao	$\overline{CS}$	O/P operation (Write)
1	0	0	0	0	Data bus to Port A
1	0	0	1	0	Data bus to Port B
1	0	1	0	0	Data bus to Port C
1	0	1	1	0	Data bus to CWR

RESET :

Pin = 35

Types = I/P

This is an active high signal. It is used to reset the mp. After reset the control word register is cleared and all ports are to be set to i/p mode.

PAo - PA$_7$:(Port A Pins)

Pins = 1 to 4and 37 to 40

Types = I/P or O/P

These are the 8-bit bidirectional pins used to send data to the device or to read data from device connected with the 8255 chip.

PBo – PB$_7$:- (Port B Pins):

Pins = 18 - 25

Types = i/p or o/p

These are also 8-bit bidirectional pins and it is used to send data to the device or to read data from device connected with the 8255 chip.

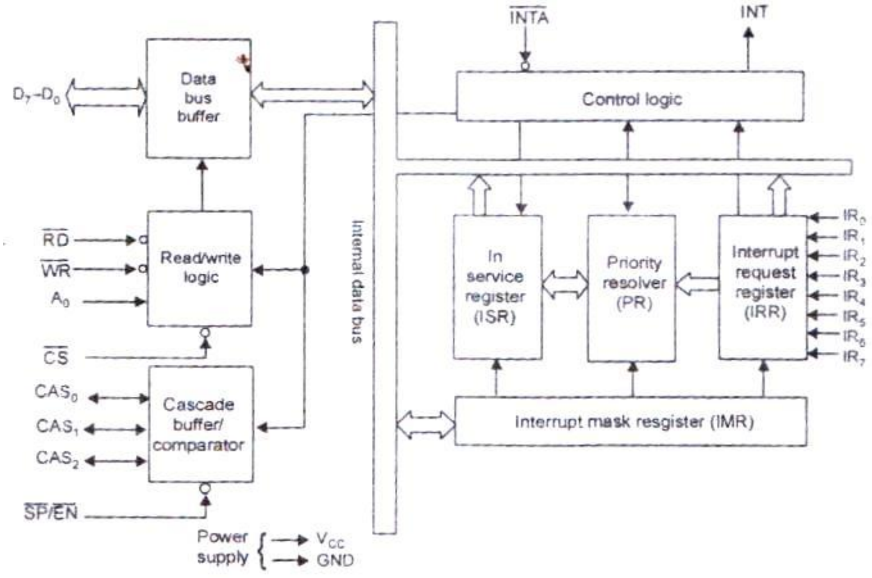
PCo – PC$_7$: (Port-C Pins) :

Pins = 10 - 17

Types = I/P or O/P

These are the 8-bit bidirectional pins. These are divided to 2 sections i.e. (PC$_4$, PC$_5$, PC$_6$, PC$_7$ PC upper and PC Lower (i.e. PCo, PC$_1$, PC$_2$, PC$_3$)

FUNCTIONAL BLOCK DIAGRAM OF 8259 PIC (PROGRAMMABLE INTERRUPT CONTROLLER):



Architecture of 8259 PIC.

(ARCHITECTURE OF 8259 PIC)

It is a 28 pin programmable interrupt controller.

The 8259 PIC contains 8 blocks

(1) Data bus buffer
(2) Read / Write logic
(3) Control logic
(4) Interrupt request register (IRR)
(5) In service register (ISR)
(6) Priority resolves (PR)
(7) Interrupt Mask register (IMR)
(8) Cascade Buffer / Comparator

DATA BUS BUFFER:

It is a 8-bit bidirectional data bus is used to interface the 8259 PIC data bus with the system data bus.

Control words and status information are transferred through the data bus buffer. It is internally connected to the data bus and its outer pi8ns Do –D_7 are connected to the system data bus directly. The direction OP the data bus is decided by the read / write control signal.

When the read signal is activated, then it transmits data to the system and when the write signal is activated then it receives data from the system data bus. The reading and writing operation is achieved by IN and OUT mp instruction.

READ / WRITE LOGIC :

The block accepts i/p from the system control bus and address bus. The control signals are $\overline{RD}$ and $\overline{WR}$, $\overline{CS}$ and address signal Ao is used. For this 5 signal $\overline{RD}$ and $\overline{WR}$

are connected to $\overline{IOR}$, $\overline{IOW}$, $\overline{MEMR}$, $\overline{MEMW}$ depending upon the mapping. $\overline{RD}$ and $\overline{WR}$ decides the operation is to performed i.e. write data to the 8259 or read data from the

8259. Ao is directly connected to the address lines Ao of the system address lines $\overline{CS}$ is connected to the Chip select decoder. It means the selection of the 8259 is enabled or

disenabled by $\overline{CS}$ signal.

If $\overline{CS}$= 0 8259 is selected

$\overline{CS}$= 1 8259 is rejected.

CONTROL LOGIC :

This block has 2 pins $\overline{INTA}$ and INT as an i/p. $\overline{INTA}$ is connected to the interrupt

Pin of the mp when a valid interrupt is occurs it goes high $\overline{INTA}$ is a interrupt acknowledgement signal from the mp.

INTERRUPT REQUEST REGISTER (IRR):

The interrupts at the Interrupt request (IR) lines are handless by the interrupt request register internally. IRR is used to store all the interrupt levels which are requesting service in it in order to serve them one by one on the priority basis.

IN SERVICE REGISTER (ISR) :

ISR is used to store all the interrupt levels which are being serviced. Each bit of this register is set by the priority resolver and reset by the end of the interrupt command word. The mp can read the contents of this register by issuing appropriate command word.

PRIORITY RESOLVER (PR):

PR determines the priorities of the bits set in the IRR. To made decision the priority resolver looks at the ISR. If the highest priority bit in the ISR is set, then it ignores the new request. If the priority resolver finds that the new interrupt has a higher priority that the interrupt currently being serviced, then it will set all the appropriate bit in the ISR and send the INT signal to the mp for the new interrupt request.

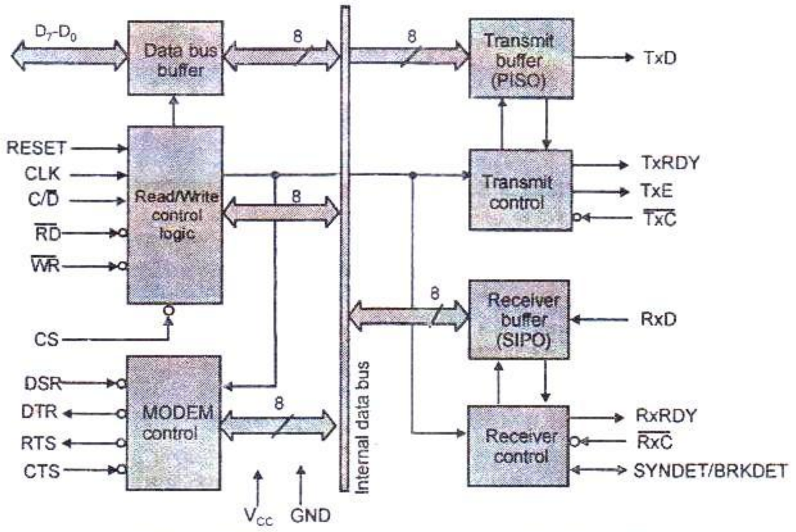
INTERRUPT MASK REGISTER (IMR):

It is a programmable register. It is used to mask unwanted interrupt request by writing appropriate control word. The IMR operates on the IRR. Masking of a higher priority i/p will not affect the interrupt request lines of the lower priority. The mp can read contents of this register without issuing any command word.

CASCADE BUFFER / COMPARATOR :

This functional block stores and compares the identification number (IDS) of all 8259s used in the system. This associate 3 Pins i.e. CAS_2 – CAS_o. These are the O/P when the 8259 is used as a master and the I/p when the 8259 is used as a Slave. As a master, the 8259 sends the ID of the interrupting Salve device on to the CAS_2 –CAS_o lines. The salve thus selected will send its programmed subroutine address on to the data bus during the next one or two consecutive $\overline{INTA}$ Pulses. In buffer mode, it generates an $\overline{EN}$ signal.

FUNCTIONAL BLOCK DIAGRAM / ARCHITECTURE OF 8251 USART:



Architecture of 8251 USART

DATA BUS BUFFER :

A 8-bit bidirectional data bus IS USED TO CONNECT 8251 usart DATA BUS TO THE SYSTEM DATA BUS. It is internally connected to the data bus and its outer pins Do-D7 are connected to the system data bus.

- For Read signal it transmits data and
- For write signal it receives data
- IN and OUT instructions are used for reading and writing operation and it is depends upon the read / write control logic.
- USART = Universal synchronous / Asynchronous receiver / transmitter.

READ / WRITE CONTROL LOGIC :

This device is used to control all the devices. After getting control signals from the control bus it generates control signals for device operation.

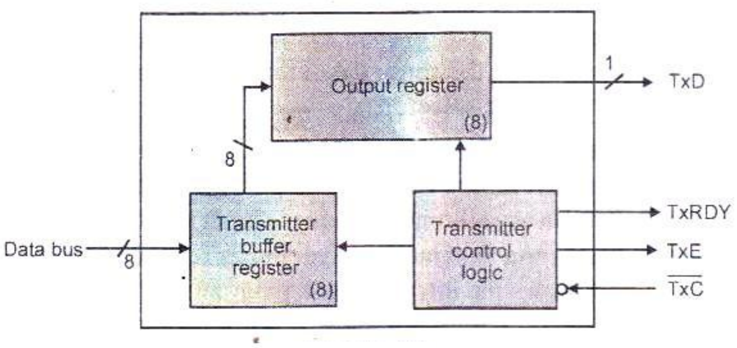
$\overline{CS}$ Pin is used to active 8251 USART. When $\overline{CS}$=0, 8251 USART is activated. The C/$\overline{D}$ (Control / Data) signal decides which part to active.
If C/$\overline{D}$=1, the control part is selected.

C/$\overline{D}$=0, the data part is selected.

The status of all signal are:

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	**Data transfer**
0	0	0	1	Receiver data register of 8251 to data bus
0	0	1	0	Data bus to 8251 transmitter data register.
0	1	0	1	Status word to data bus
0	1	1	0	Data bus to control word
1	X	X	X	Data bus tristated

TRANSMITTER SECTION:



The transmitting sections are having the signals:

TxD	- Transmit Data
TxRDy	Transmitter Ready
RxE	Transmitter Empty
TxC	Transmitter Clock

The Transmitting Section consists of transmit buffer, transmit control block and O/P register.

The transmitter Buffer register accepts data from the data bus buffer through internal data bus if Cs=0, CID =0, RD=1 and WR=0

The contents of the transmitter buffer are automatically transferred to the O/P register, if the O/P reg. is empty, the data is shifted serially on the TxD Pin, along with the appropriate bits are also added depending upon the mode selection i.e. synchronous (Synchronous Char. Are sent) or Asynchronous mode (Start and stop bits are sent).

If transmitter buffer reg. does not contain any data then TxRDY=1 and allow mp to sent next data for transmit control to indicate peripheral about in availability of data for transmission.

If TxRDY=1, it indicates transmitter buffer is empty. Here the data 1st cone to the transmit buffer reg. and then it transfers to the O/P register and finally from O/P reg it transfers one bit at a time on TDx Pin.

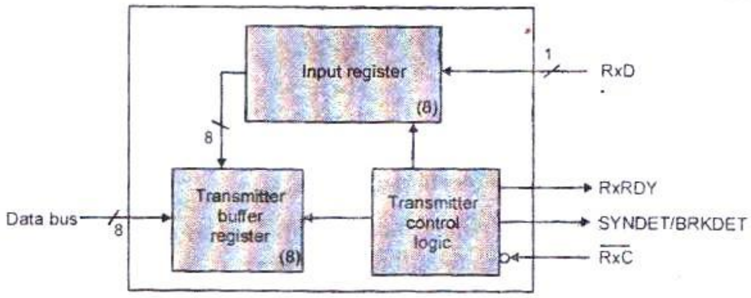
If TxRDY=0, it indicates that the transmitter buffer is not empty and contains some data when last data is transferred from mp to the transmit buffer register at the same time this last data is transferred from O/P register to the TxD Pin. It means the transmit buffer reg and O/P reg. are both empty, then it gives TxE=1

TxC is used to apply –ve edge clock pulses.

The condition for making TxE=1 are

(i) Transmit buffer reg. is empty
(ii) TxE=1
(iii) CTS=o
(iv) Upon master reset signal.

RECEIVER SECTION:-



This section contains the signals

RxD= Receive data

RXRDY= Receiver ready

SYNDET/BRKDET=Synchronous detect / break detect

$\overline{RxC}$= Receiver Clock

This section consists of Receiver buffer Register, receiver control logic and i/p reg. The function of i/p reg. is to accept the serial data through RxD. The function of i/p reg. is to convert the serial data to parallel from.

In synchronous mode it check the start bit and if the start bit is detected then the bit after start are converted to parallel form and then transferred to receiver buffer reg.

In asynchronous mode the i/p reg. will accept the data and converts it to parallel form and load into receiver buffer reg. if the SYNDET signal is 1.

In asynchronous mode after the data bits are accepted receiver checks programmed parity bit. If both are not same then a error signal is generated.

After the data bytes are transferred from i/p reg. to receiver buffer reg. the control logic generates a signal RxRDY to signal the mp about availability of data bytes to read by mp.

MODEM CONTROL :

Telephone lines are used to send the data over long distance. The telephone lines are analog in nature, so MODEMs (Modulator-Demodulator) are used to convert datas. To communicate MOIDEMs with 8251, the 8251 USART provides a block called MODEM control. The various signals under this unit are $\overline{RTS}$, $\overline{CTS}$, $\overline{DTR}$ and $\overline{DSR}$.

$\overline{DTR}$ (DATA TERMINAL READY):-

After the terminal is mode ON, different operation will be performed. At the time of data transmission / reception it generates a signal DTR to indicate its readiness for data transfer.

$\overline{DSR}$ (DATA SET READY):

This i/p is used as a general purpose one bit inverting i/p port. Its status can be checked by mp using a status read operation. This bit is used to check, if the data set is ready while communicating with a modem.

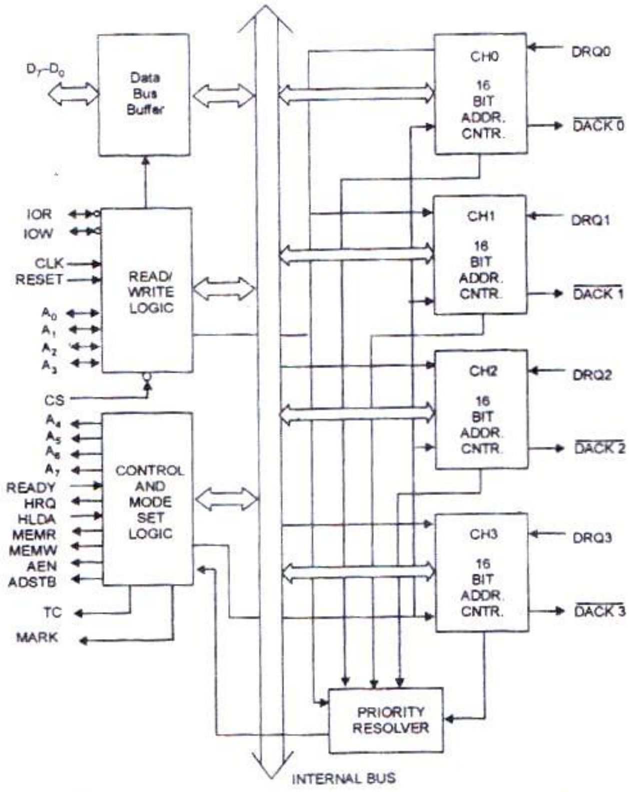
$\overline{CTS}$ (CLEAR TO SEND):

When MODEM is ready to transmit data it makes $\overline{CTS}$ signal low . The terminal after receiving $\overline{CTS}$ signal sends serial data character to the MODEM.

$\overline{RTS}$ (REQUEST TO SEND) :

Before the terminal is ready to transmit data and has a data character to be transmitted the terminal makes $\overline{RTS}$ signal low.

FUNCTIONAL BLOCK DIAGRAM / ARCHITECTURE DIAGRAM OF 8257 DMA CONTROLLER:



Internal Architecture of 8257

The 8257 supports 4 DMA channels. That means 4 peripheral devices can request for DMA data transfer through these channels at a time.

It has

(1) 8-bit internal Data buffer
(2) A read / write unit
(3) A Control unit and
(4) Priority Resolving unit along with a set of registers.

REGISTER ORGANISATION OF 8257:

The 8257 has 4 independent DMA channels . Each channel has a pair of 2 16-bit registers:

(1) DMA address register
(2) Terminal count register

There are 2 common registers for all the channels i.e. mode set registers and status register. The address lines Ao-A3 are used to select one of the registers.

DMA ADDRESS REGISTERS:

Every DMA channel has one DMA address register. The Primary function this register is to store the address of the starting memory location which is access by the DMA channel.

TERMINAL COUNT REGISTERS:

Each channel of 8257 DMA has one terminal count register (TC) . It is a 16 bit register. It ascertains that the data transfer through a DMA channel stops after the required number of DMA cycles.

It means this register should be appropriately written before the actual DMA operation starts.

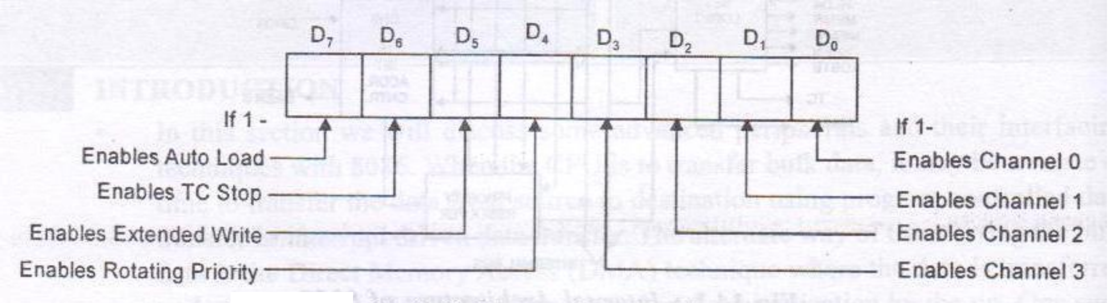
This terminal count register are initialized with the binary equivalent of the no. of required DMA cycles minus 1

After each DMA operation the terminal count register content will be decremented by one and finally it becomes zero after required no of DMA cycles are evered.

MODE SET REGISTERS:

As per the requirements of system the mode set register (MSR) is used for programming the 8257.

The Format of MSR is



Bit Definitions of the Mode Set Register.

Do - D3 Used to enable the 4 DMA channels of 8257

D6 If IC bit stops then the selected channel is disenabled. The IC stop bit is zero.

D4 It is set when the rotating priority is enables. Otherwise fixed priority is enabled.

D7 If it set, enables channel 2 for repeat block chaining operation.

D5 If it set, then the duration of $\overline{MEMW}$ and $\overline{IOW}$ signal is activated.

ADSTB (ADDRESS STOBE):

It is the higher byte of the memory address generated by the DMA controller in to the Latches.

AEN (ADDRESS LATCH ENABLE):-This o/p is used to disable the system data bus and control bus given by the CPU.

TC (TERMINAL COUNT): It indicates to the currently selected peripheral.

MARK (The Modulo-128): It indicates to the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK O/P . It will be activated after 128th cycles.

PROGRAMMABLE DMA CONTROLLER:

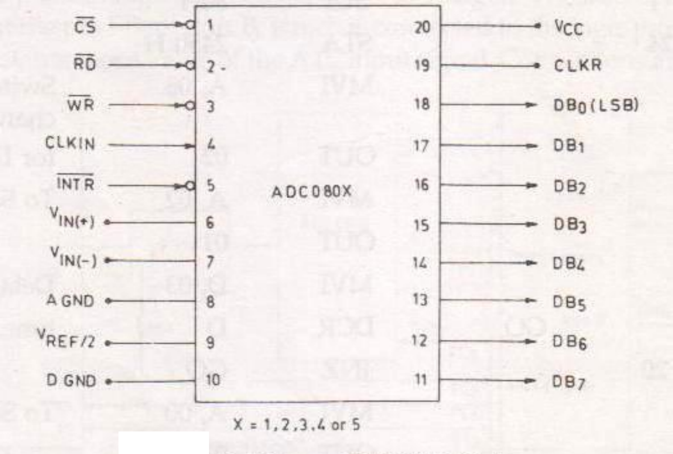
The bulk data transfer from I/O devices to memory or from memory to I/O devices through the accumulator is a time consuming process. For this process Direct Memory Access (DMA) technique is preferred.

In DMA data transfer scheme, data are directly transferred from an I/O device to RAM or from RAM to I/o device.

OPERATING PRINCIPLE OF ADC 0801 SERIES:-

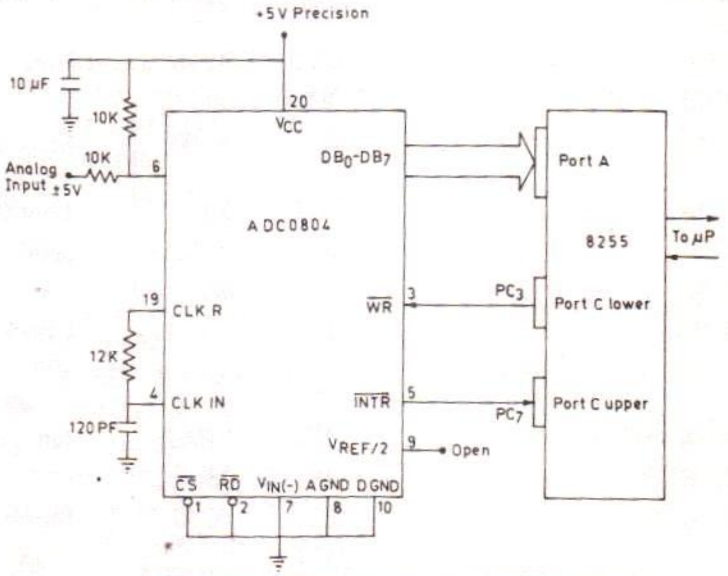
It's a 8-bit successive approximation A/D converters. Its important features are: on-chip clock generator, differential analog voltage input and no requirement of zero-adjustment. To start interfacing the $\overline{CS}$ should be low. The range of clock frequency is 100KHZ to 800KHZ.The clock frequency is: F=1/(1.1 RC).

A typically range of R=10KΩ to 50KΩ. Corresponding to R=12kΩ and C=120pf, the clock frequency=632KHZ.



Pin Diagram of ADC 0801 Series

INTERFACING OF ADC:



Interfacing of ADC 0804 for ± 5 V Analog Input Voltage.

It is a interfacing circuit of ADC 0808/0809 to intel 8085 microprocessor. An analog input is connected to IN3. 5Vdc is applied to pin no 12 i.e.(ref +ve).It should not be given from a stabilized power unit. The pin no 9 and 11 are connected to the 5Vdc stabilized power unit.

PROGRAM:

Mnemonics	Operands	Comments
MVI	A,98H	Initialize I/O ports of 8255
OUT	0B	
MVI	A,03	Switch ON multiplexer channel IN3
OUT	0A	
MVI	A,0B	Start of conversion pulse without affecting multiplexer's channel.
OUT	0A	
MVI	A,03	
OUT	0A	
IN	0A	Read E/C signal.
RAL		Rotate accumulator left.
JNC	READ	Is conversion over? No, jump to READ. Read digital o/p of A/D converter.
IN	08	
STA	FC50H	Store the result.
HLT		Stop.

OPERATING PRINCIPLE OF DAC:

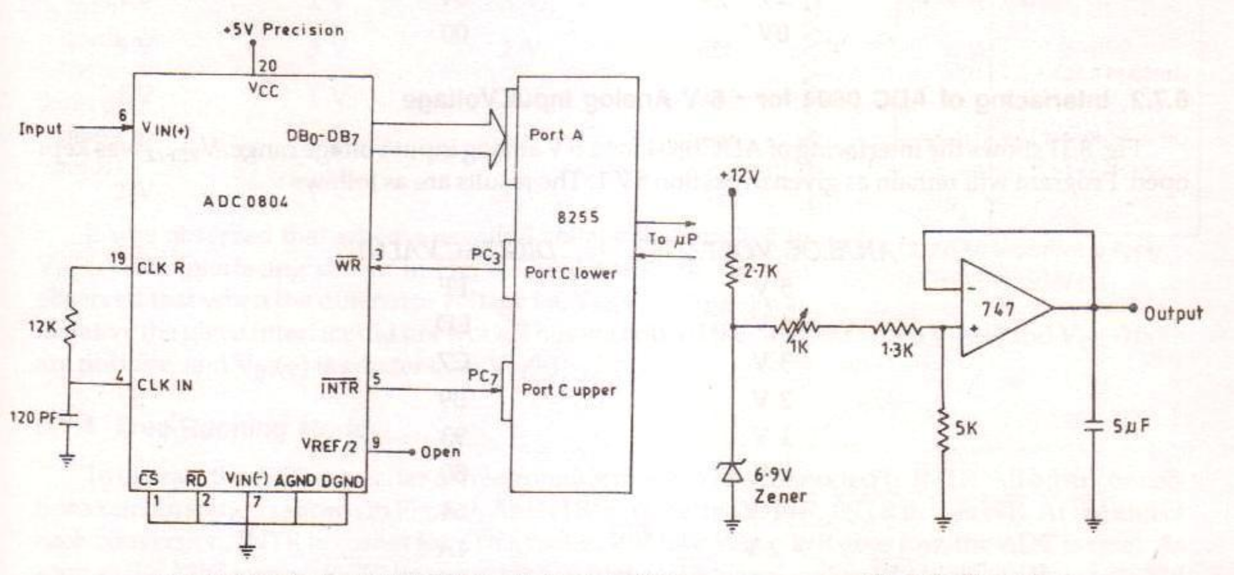
A DAC contains a ladder n/w. The n/w has i/p for binary bits of the digital word. When the MSB=1, it produces an o/p current, $(I_{REF})/4$. The bit next to the MSB produces $(I_{REF})/4$ and so on. It produces an o/p current or voltage proportional to the magnitude of the digital word applied to it. The o/p current is given by

$I_{out}= I_{REF}\ (1/2\ B_{n-1}+1/4\ B_{n-2}+...+1/2^n B_0)$

Where B_0, B_{n-1}, B_{n-2}.... Are the binary bits of digital word applied to DAC.

$I_{REF}=(V_{ref})/R$ Where R=2.5K

INTERFACING OF DAC:



Interfacing of ADC 0804 for Positive Input Voltage | Circuit for VREF

The DAC 0800 is a simple monolithic 8-bit D/A converter. It has fast settling time,100ns. It can directly interface to TTL,CMOS,PMOS and others. It operates at4.5V to +18V supply. The supply V^+ may be either 5V or +12V. V^- is kept -12V being easily available on standard power supply unit.

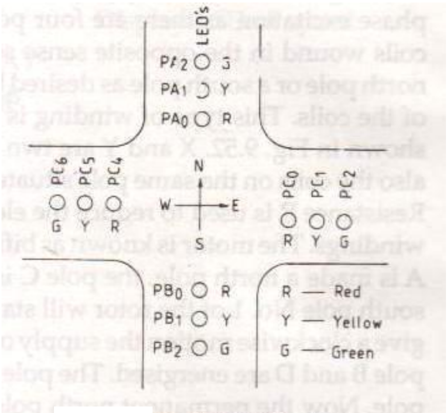
PROGRAM:

Mnemonics	Operands	Comments
MVI	A,98H	Get the control word for 8255
OUT	0B	Initialize ports.
MVI	A,80	Get 80 for digital i/p to DAC.
OUT	09	I/P 80to DAC through port B
HLT		Stop.

For bipolar operation the following modification in the circuit has to be done.

Connect pin 2 of DAC to non-inverting terminal of Op-amp. This common point is earthed through a 5KΩ resistor. The connection of pin4 of DAC and inverting terminal Op-amp will remain as before.

INTERFACING OF TRAFFIC LIGHT CONTROL SYSTEM USING 8255:



Traffic Light Control.

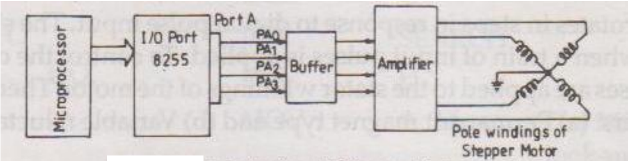
All the ports of 8255 have been programmed as O/P ports. The control word to make all the ports o/p ports in mode 0 operation is 80H. The connection of pins of the ports to LED have been made through buffer(7407).+ve logic have been used to switch on LEDs. Three types of LEDs have been used to switch on LEDs.

3 types of LEDs are Yellow, Red, Green.
Yellow to make alert
Red does not allow crossing.
Green allow crossing.

PROGRAM:

Mnemonics	Operands	Comments
MVI	A,80H	Get control word for 8255
OUT	0B	Initialize ports of 8255
MVI	A,01	
OUT	09	Red ON for South
OUT	08	Red ON for North
MVI	A,44	Green ON for East and West.
OUT	0A	
CALL	DELAY 1	
MVI	A,22	Yellow ON for East and West.
OUT	0A	
MVI	A,02	
OUT	09	Yellow ON for South
OUT	08	Yellow ON for North
CALL	DALAY 2	
MVI	A,11	Red ON for East and West
OUT	0A	
MVI	A,04	
OUT	08	Green ON for North
OUT	09	Green ON for South
CALL	DELAY 1	
MVI	A,22	Yellow ON for East and West
OUT	0A	
MVI	A,02	
OUT	09	Yellow ON for South
OUT	08	Yellow ON for North.
CALL	DELAY 2	

INTERFACING OF STEPPER MOTOR CONTROL:



Interfacing of Stepper Motor.

The necessary steps to interface the 8051 with the stepper motor.

1. The ohm meter is used to measure the resistance of the leads. It is used to identify which COM lads are connected to which winding leads.

2. The common wires are connected to the +5v side of the motor power supply.

3. The 4 bit of 8051 i.e. P1.0, P1.1, P1.2, P1.3 are used to control the 4 leads of the stator winding used a driver to energized the stator. The driver has an internal diode to take care of back emf.

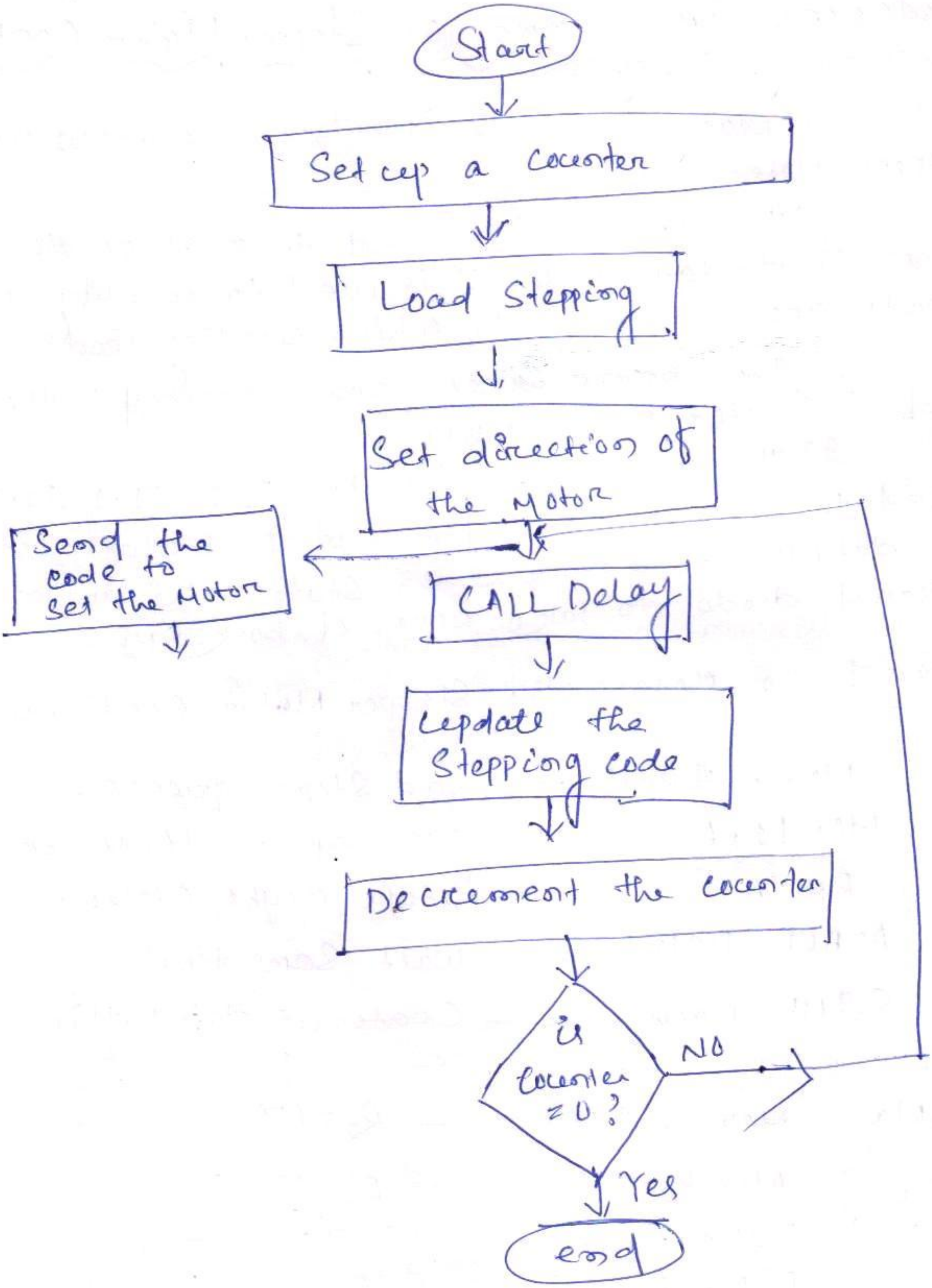
PROGRAM: To rotate the stepper motor continuously.

MOV A,# 66H	Load step sequence
Back: MOV P1,A	Issue sequence to motor
RR A	Rotate right clockwise
A CALL Delay	Wait some time
S JMP Back	Continue doing this

Delay MOV R_2, # 100	R_2=100
L1: MOV R_3, # 255	R_3=255
L2: DJNZ R_3, L_2	
DNZ R_2,L_1	
RET	

Relationship between steps per second and rpm

Steps per second= (Rpm * Steps per revolution)/60.


Start
Set up a counter
Load Stepping
Set direction of the Motor
Send the code to set the Motor
CALL Delay
Update the Stepping code
Decrement the counter
is counter = 0 ?
NO
Yes
end

Chapter-7

16-Bit Microprocessor

Introduction:

8086 is a 16-bit of microprocessors. Its internal data lines are of 16. The 8088 and 80188 have their internal architecture of 16 bit but their data lines are only 8. All these microprocessors come under Intel 8086's family.

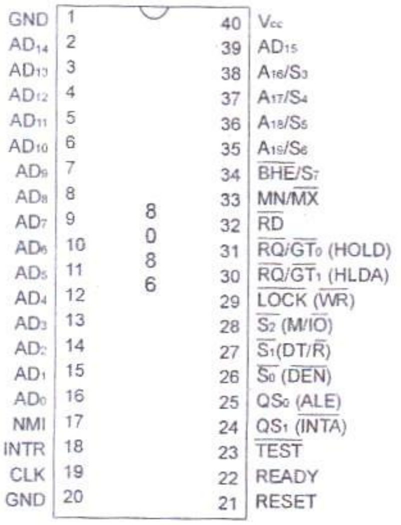
8086, 80186,80286,8088 and 80188 microprocessor have the same basic set of registers, instruction and addressing modes.

The 8086 is a 16 bit, N-channel, HMOS microprocessor the term HMOS is used for "high-speed MOS". It is a set of 40 pins IC package. The types of packaging are DIP (dual inline package).

AD_0-AD_{15} are the 16 low order address line 8 LSB of data are transmitted on AD_0-AD_7 & 8 MSBs of data on AD_8-AD_{15}.

PIN CONFIGURATION OF 8086

There are 40 numbers of pins in 8086.



8086 pin configuration

AD_0-AD_{15}:

Address / data lines. There are low-order address buses. They are multiplexed with data.

When AD lines are used to transmit memory address the symbols A is used instead of AD, EXP: A0-A15 & when the data are transmitted D is used instead of AD. EXP: D_0-D7, D_8-D_{15} or D_0-D_{15}.

A_{16}-A_{19} (output):

There are the high order address lines those are multiplexed with status signals.

A_{16}/S_3, A_{17}/s_4=A_{16} & A_{17} are multiplexed with segment identified signal S_3 & S_4.

A_{18}/s_5=A_{18} is multiplexed with the status signal S5.

A_{19}/S_6=A_{19} is multiplexed with status signal.

$\overline{BHE}$/S_7 (o/p):-

Bus high enable /status signal .during T1 is low ,it is to enable data into the most significant half of data bus D_8-D_{15} 8-bit device connected to upper half of data bus use $\overline{BHE}$ signal is available during T3 and T4.

$\overline{RD}$ (read):-

It is used for read operation. It is active when it is low

READY (input):-

The addresses I /O or memory sends acknowledgement through this pin. When it is high it indicates that the peripheral is ready to transfer the data.

RESET(i/p):-

Used for system reset. It is active when it is high

CLK (i/p):-

Clock .10 MHz

INTR:-

Interrupt request

NMI (i/p):-

Non-maskable interrupt request

$\overline{TEST}$ (i/p):-

Wait for test control when it is low the microprocessor continuous execution otherwise waits.

VCC:-

Power supply +5v dc

GND:-

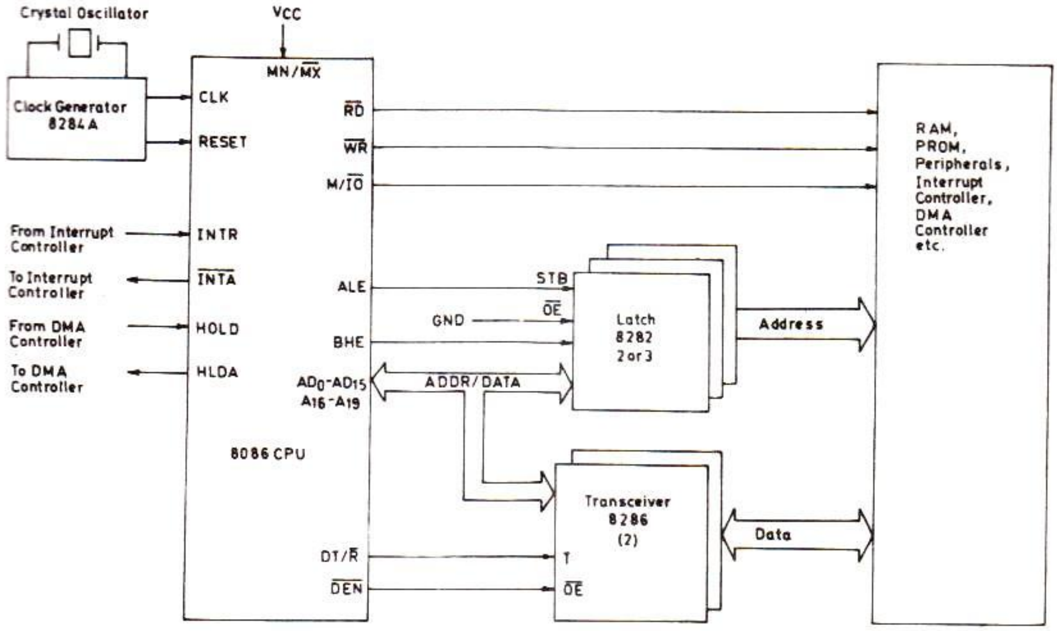
Ground

MINIMUM MODE-MAXIMUM MODE:-

There are 2 available modes of operation for the 8086/8088 microprocessor. i.e. maximum and minimum mode maximum mode operation is obtained by connecting the mode selection pin . MN/$\overline{MX}$ to +5v and maximum mode to ground that pin.

MINIMUM MODE OPERATION:-

Minimum mode operation is the least-expensive way to operate the 8086/8088 microprocessor. all the control signal for the memory and D10 are generated by the microprocessor the minimum mode allows the 8085A ,8bit peripheral to be used with the 8086/8088 with the out any special consideration.



Typical 8086-based Computer System in Minimum Mode Configuration

For the minimum mode operation the pin MN/$\overline{MX}$ is connected to 5v dc supply IC. MN/$\overline{MX}$=VCC

$\overline{INTA}$ (o/p):-

Interrupt acknowledge on receiving interrupt signal the process issues an interrupt acknowledge signal. It is an active low signal.

ALE (o/p):-

Address latch enable .it goes high during T1 the microprocessor sends this signal to latch the addresses.

$\overline{DEN}$(o/p):-

Data enable .it is used for as output enable signal .it is active low.

DT/$\overline{R}$ (o/p):

Data transmit/receive. It is used to control the direction of data flow through the transceiver when it is high bit sent the data and when it is low it receives the data

M/$\overline{IO}$ (o/p):-

When it is high the microprocessor wants to access the memory and when it is low the microprocessor access the I/p device

$\overline{WR}$ (o/p):-

When this signal is low it perform the write operation for memory and i/o device

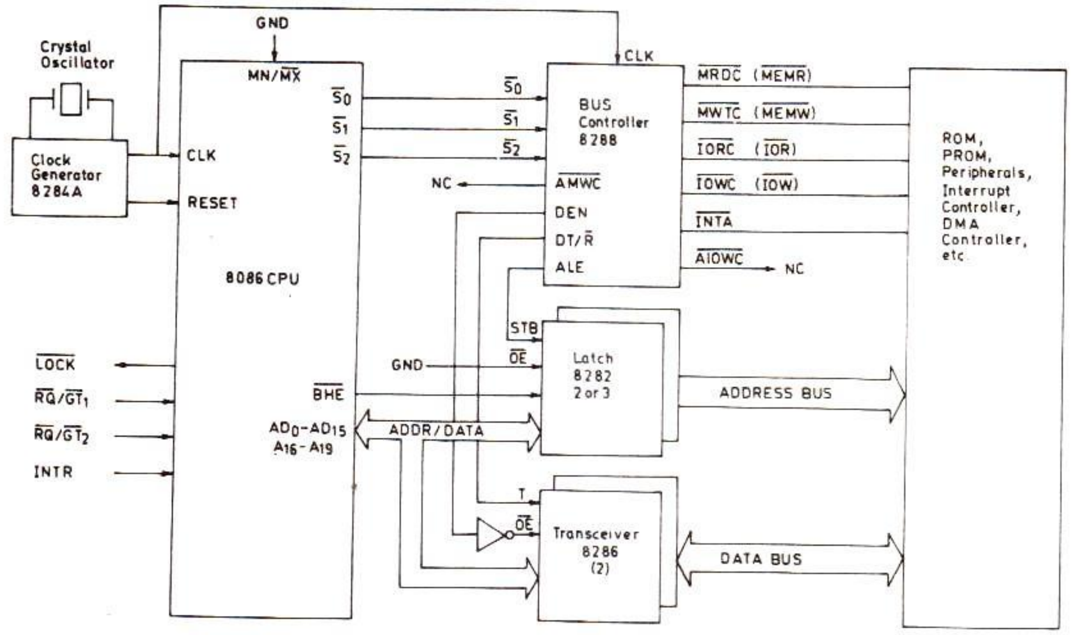
HLDA (o/p):-

Hold acknowledge .it is issued by the microprocessor when it receives hold signal .it is an active high signal. When the hold request is removed HLDA goes low

HOLD (I/O):-

When another device in microprocessor system wants to use the address and data bus , it sends a HOLD request to CPU through this pin , it is also a active high signal

MAXIMUM MODE OPERATION:



Typical Intel 8086-based Computer System in Maximum Mode Configuration

Maximum modes operation differs from minimum mode in that same ctrl signal must be externally generated.

This requires the addition external bus controller there are not enough pins on 8086/8088 for bus ctrl, so for maximum mode operation same new pins and few features have replaced some of them. This maximum mode is used when the system contains external co-processors such as the 8087 arithmetic co-processor.

For the maximum mode operation the pin MN/$\overline{MX}$ is mode low it is grounded.

QS_1, QS_0 (o/p):-

Instruction queue status.

QS_1	QS_0	
0	0	no operation
0	0	1st byte of opcode from queue
1	0	empty the queue
1	1	subsequent byte from queue

$\overline{S0}$, $\overline{S1}$, $\overline{S2}$ (o/p):-

These are the signal connected to the controller 8288

S_2	S_1	S_0	
0	0	0	interrupt acknowledge
0	0	1	read data from i/o port
0	1	0	write data from i/o port
0	1	1	halt

1	0	0	opcode fetch
1	0	1	memory read
1	1	0	memory write
1	1	1	passive status

$\overline{LOCK}$ (o/p):-

It is an active low signal when it is low all interrupts are masked and no hold request are generated

$\overline{RQ}/\overline{GT1}$, $\overline{RQ}/\overline{GT2}$;-

Local bus priority control other processers ask the CPU though these lines to release the local bus $\overline{RQ}/\overline{GT0}$ has higher priority than $\overline{RQ}/\overline{GT1}$,

In maximum mode operation $\overline{WR}$, ALE,$\overline{DEN}$, DT/$\overline{R}$ etc, are not available directly from the processer . These signals are available from the controller 8288.

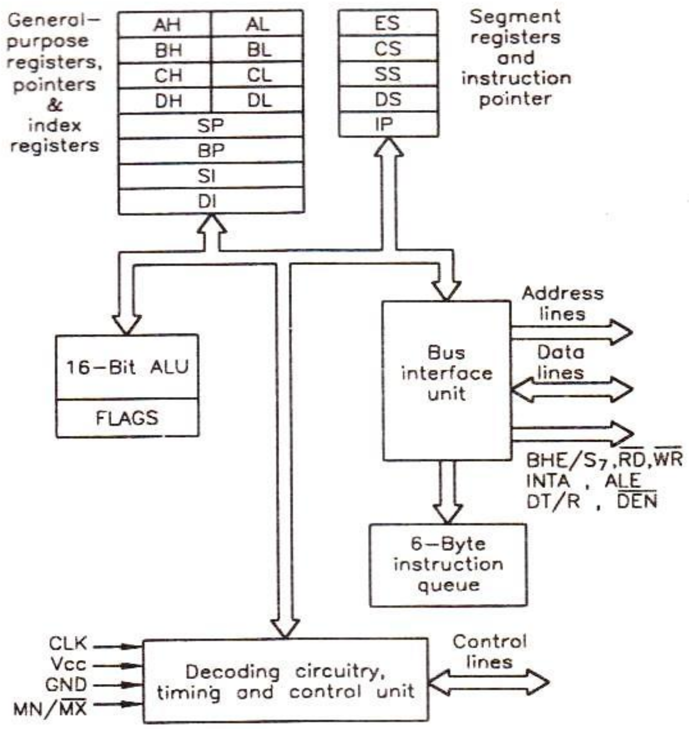
CLK:-

Clock I/p

ALE:-
Address latch enable.
$\overline{DEN}$:-
Data bus enable.
DT/$\overline{R}$:-
Data transmit/receive.
$\overline{AEN}$:-
Address enable.
CEN:-
Control enable.
$\overline{ALOWC}$:-
Advanced I/O write command.
$\overline{IOWC}$:-
I/p write command.
$\overline{IORC}$:-
I/O read command.
$\overline{AMWC}$:-
Advance memory write control.
$\overline{MWTC}$:-
Memory write control.
$\overline{MRDC}$:-
Memory read control.
$\overline{INTA}$:-
Interrupt acknowledge.

Architecture of 8086 microprocessor:-

INTEL 8086 AND INTEL'S OTHER 16-BIT MICROPROCESSORS



Block Diagram of Intel 8086 Microprocessor

The 8086 microprocessor contains 2 independent units i.e. bus interface unit bus interface unit (BIU) and an execution unit (EU).

The general purpose registers, stack pointer, base pointer and index registers, ALU, Flag register, instruction decoder and timing and control unit constitute execution unit (EU).The segment register, instruction pointer,6-byte instruction queue are associated with the bus interface unit (BIU).

The BIU handles transfer of data and address between the processor and memory i/o device.

While EU executes instruction the BIU fetches instruction. This type of overlapped operation of the functional units of a microprocessor is called pipeline.

REGISTERS OF INTEL 8086:-

The 8086 contains the following registers

(1) General purpose registers
(2) Pointer and index registers
(3) Segment register
(4) Instruction pointer
(5) Status flags.

(1) GENERAL PURPOSE REGISTERS:-

There are four 16-bit general purpose register: AX, BX, CX and DX. Each of these 16-bit register are further subdivided into 2 8-bit registers i.e.

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 is serves as an accumulator .register BX CX, and DX are used as general purpose registers sometimes it serves as special purpose register. BX serves as a base register for the computation of memory address. XC is used counter in case of multi iteration instruction. When the content of CX becomes zero such instruction terminate the execution. DX is also used for memory addressing when the data are transferred between i/o port and memory using certain i/o instruction.

POINTER AND INDEX REGISTER:-

(1) stack pointer ,SP
(2) Base pointer , BP
(3) Source index ,SI
(4) Destination index ,DI

STACK POINTER:-
A 16 bit register which is used to store top of the stack , it is pointing to a memory location inside the stack segment in main memory .
The contain of the stack pointer is added with the stack segment to get the physical address of the top of stack pointer .
BASE POINTER:-
16 bit register which is also holds the offset address of stack segment.
It may locate any memory location inside the stack segment.

SOURCE POINTER:-

These are used to store the offset address of data and extra segment.

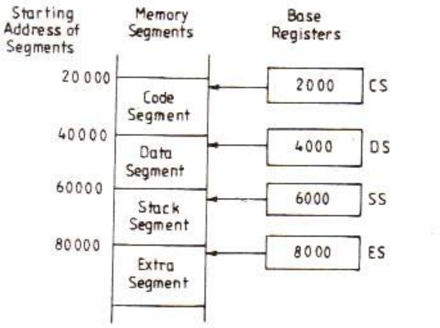
Source index store the offset Address of the instruction and the destination index stores the extra segment of instruction.

SEGMENT REGISTER:-

There are 4 segment register in 8086 i.e.

(1) Code segment
(2) Data segment
(3) Stack segment
(4) Extra segment

In 8086 the memory of 8086 are divided in to 4 segment i.e. code, data, stack and extra .



Memory Segments

CODE SEGMENT:-

The code segment of the memory holds instruction codes of a program.

Code segment points to the starting address of the codes segment.

DATA SEGMENT:-

The data variable and constants given in the program are held in the segment of memory .

Data segment points to the starting address of the data segment.

STACK SEGMENT:-
Stack segment holds address and data of subroutines. it also holds the contents of registers or memory location given in PUSH operation .
EXTRA SEGMENT:-
The extra segment holds the destination address of some of certain string instruction and so on.
A segment register pointer to the starting address of memory segment currently being used.
The content of the stack pointer and the content of stack segment register are used to compute the address the stack lo0cation to be accessed.
The index register i.e. SI and DI together with segment register DS and ES used to perform string operation .

INSTRUCTION POINTER (ip):-

It also refers as a program counter.

It holds the address of the next instruction that means it point to the memory location in the code register to get the instruction address.

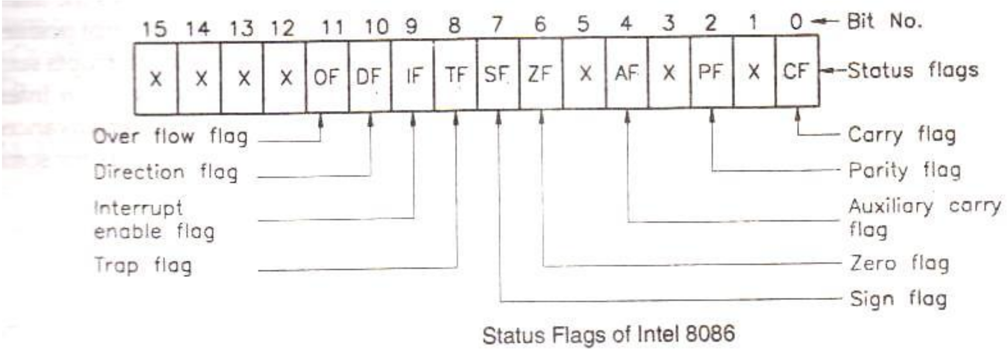
After fetching each instruction the instruction pointer is incremented according to the size of the instruction.

STATUS REGISTERS:-

8086 microprocessor contains a 16 bit status register, it is also called as flag register or program status word (PSW).

There are 9 status flags and other 7bit are not used. The status flag are:

(1) Carry flag
(2) Parity flag
(3) Auxiliary carry flag
(4) Zero flag
(5) Sign flag
(6) Trap flag
(7) Interrupt enable flag
(8) Direction flag
(9) Over flow flag.



Status Flags of Intel 8086

Out of these 9, 6 are the conditional flag and other 3 are the control flag.

The 6 conditional flag are CF, AF, ZF, SF, PF and OF. These flag are set or reset by the processor after the execution of arithmetic or logical instruction.

The 3 control flag are TF, IF and DF . These flag are set or reset by the programmer as required by certain instruction in the program.

All the flags except TF, OF, IF and DF are same as in 8085.

The OF set to 1, when result of signed operation is out of range.

The TF is set to 1 that means a program can be run in single-step mode.

The IF is set to 1, when the INTR of 8086 is enabling.

The DF is used for string operation .it is set to 1, when the string byte are accessed from higher memory address to lower memory address.

INTERRUPTS:-

An interrupt may occurs at normal program execution of a microprocessor .an interrupt caused by a external device is called as hardware interrupt. A microprocessor can also be interrupted by the internal abnormal condition like overflow, division by zero etc.

A programmer can also interrupt microprocessor by inserting INT instruction at desired point in the program while debugging a program. Such interrupt is called software interrupt.

8086 microprocessor can handle up to 256, hardware and software interrupts to store the starting address ISS for 256 interrupts 1kb memory from 0000 to 003FF is set . The starting address of an ISS stored in the 1kb memory space is called interrupt vector or interrupt pointer . From 4bytes memory 2 bytes for store the CS and 2bytes for IP values.

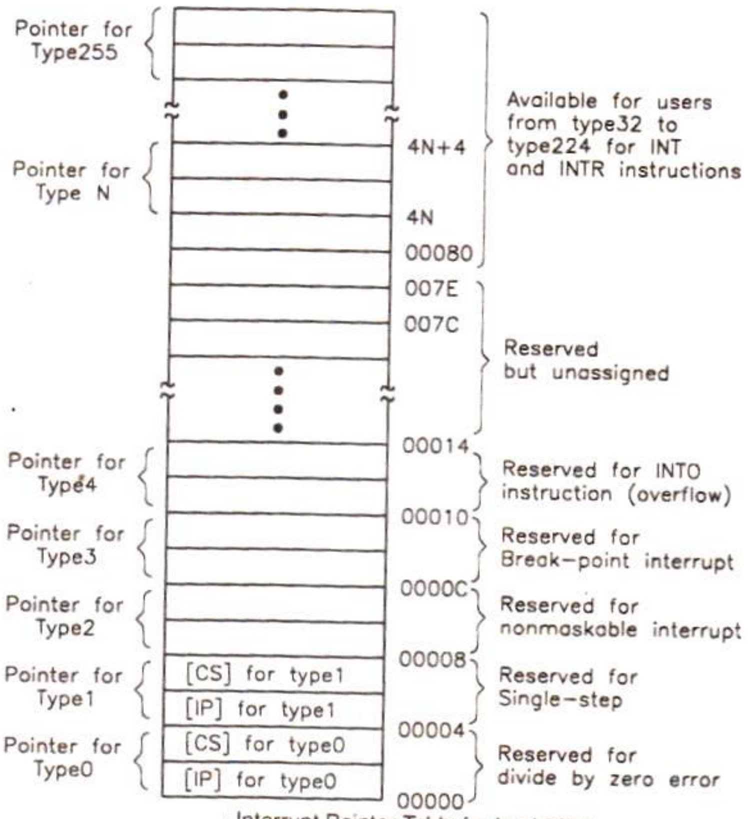
1kb memory space act as a table to contain interrupt vector and hence it is called interrupt vector table or interrupt pointer table .the256 IP is number from 0 to 255 .

1st 5 interrupt are specific interrupt such as divide-by-zero, single-step control, non-maskable interrupt NMI, break point and overflow interrupt. The next 5 to 31 are reserved for other advanced microprocessor of Intel Corporation. The upper 224 interrupt from type 32 to type 255 are available to user for hardware and software interrupt.

The 8086 has 2 hardware interrupt. It can't be disabled by user by using software. It is used by the processor to handle emergency condition.

The ISS for type 2 interrupt saves program after power failure in some RAM provided from the point at which it was interrupt.

INTR is a maskable interrupt it can be enable /disenabled using interrupt flag (IF) .after receiving INTR interrupt from an external device, 8086 acknowledges through INTA signal.



Interrupt Pointer Table for Intel 8086

ADDRESSING MODE OF INTEL 8086:-

An instruction performs specific operation on the specified data .hence, the programmer must specified the required data for an instruction. The way by which an operand is specified for an instruction is called addressing mode.

The 8086 has 8 addressing modes i.e.

(1) register addressing mode
(2) immediate addressing mode
(3) direct addressing mode
(4) register indirect addressing mode
(5) based addressing mode
(6) indexed addressing mode
(7) based index addressing mode
(8) based indexed with displacement

(1) **REGISTER ADDRESING MODE:**

In this addressing mode the operand is placed in 1 of the 16bit or 8bit general purpose registers.

Ex: MOV AX, CX

ADD AL, BL

ADD CX, DX

(2)IMMEDIATE ADDRESSING MODE:

In immediate addressing the operand is specified in the instruction itself ,

Ex: MOV AL, 35H

```
MOV BX, 0301H

MOV [0401], 3598H

ADD AX, 4836H
```

Displacement: it is an 8bit or 16bit 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

(3)DIRECT ADDRESSING MODE:

In direct addressing mode the operands offset is given in the instruction as an 8bit or 16bit displacement element.

```
Ex:  ADD AL, [0301]

     ADD [0301], AX
```

(4)<u>REGISTER INDIRECT ADDRESSING MODE:</u>

The operand offset is placed in any 1 of the register BX, BP, SI or DI as specified in the instruction

```
Ex:  MOV AX, [BX]

ADD AL, [SI]
```

(5) <u>BASED ADDRESSING MODE :</u>

The operand offset is the sum of and 8bit or 16bit displacement and the contents of the based register BX or BP .BX is used as base register for data segment and BP is used as a base register for stack segment.

```
Ex: MOV AL, [BX+05]

    MOV AL, [BX+1346H]
```

(6) <u>INDEX ADDRESSING MODE:</u>

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

(7) <u>BASED INDEXED ADDRESSING MODE:</u>

The operand offset is the sum of the content of a base register BX or BP and an index register SI or DI

```
Ex: ADD AX, [BX+SI]
    MOV CX, [BX+SI]
```

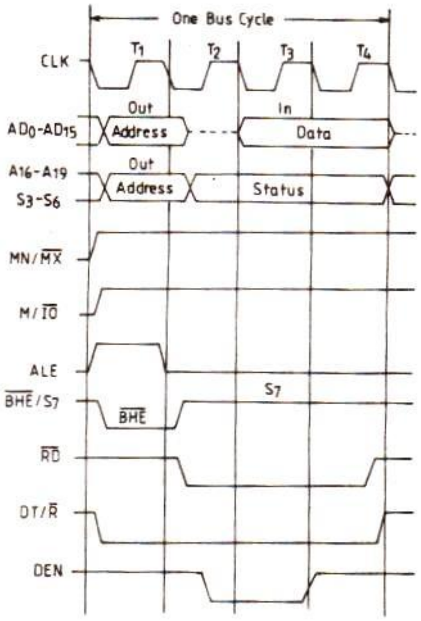
(8)<u>BASED INDEXED WITH DISPLACEMENT:</u>

In this addressing mode the operand offset is given by offset = [BX or BP]+[SI or DI]+8bit or 16bit displacement

```
Ex: MOV AX, [BX+SI+05]

    MOV AX [BX+SI+1235H]
```

BASIC TIMING DIAGRAM OF INTEL 8086 MICROPROCESSOR:



' Timing Diagram of Intel 8086 for Memory Read in Minimum Mode

CLASSIFICATION OF 8086 INSTRUCTIONS:

Instruction are classified a the basic of function of they perform. There are categorized into:-

1. Data transfer instruction
2. Arithmetic instruction
3. Logical instruction
4. Program Execution transfer/Branch instruction

1.DATA TRANSFER INSTRUCTIONS :

This instruction performs all the data movement operation like MOV, Load, Store, Exchange, I/P,O/P,PUSH,POP instruction.

The source of the data may be a register memory location, port etc . The destination may be also the same.

Ex: MOV, XCHG, PUSH, POP, LDS, LEA , IN,OUT ETC .

2. ARITHMATIC INSTRUCTIONS:

The instruction performs all the arithmetic operation like addition, substraction, multiplication, division, increment, decrement, comparison, etc.

Ex: ADD, SUB, INC, MUL, DIV, CMP, etc

3. LOGICAL INSRTUCTION :

This perform all the logical operation like AND, OR, X-OR, NOT, TEST ETC

4. PROGRAM EXECUTION TRANSFER OR BRANCH INSTRUCTION:

The instruction of this group transfers program execution from the normal sequence of instruction .to the specified destination or target.

After the execution of such instruction, the processor starts instruction

Ex: JMP, JC, JZ, CALL, RET etc

ITERATION CONTROL INSTRUCTIONS:

Instruction like loop loopz, loopne etc are come under this group

INTERRUPT INSTRUCTION:

Instruction such as INT, INTO and IRET are comes under this group.

PROCESS CONTROL INSTRUCTION :

Instruction like flag manipulation and machine ctrl. Are comes under this group.

Ex : CLC< CLD< STC <LOK etc

STRING INSTRUCTION:

It handles the string operation such as string movement, comparison, scan, load and store.

Ex.; MOV S/MOV SB/ MOV SW, CMPS/CMPSB/CMPSW, SCAS/SCAB/SCASW, LODS/LODSB/LOSDW etc.

****