ELECTRONICS & TELECOMMUNICATION DEPARTMENT

Govt. Polytechnic, Bhubaneswar

Faculty: Kanak Prava Swain, Lect.(ETC).

Subject: Microprocessor & Microcontroller. (Th-3)

Semester: 4th Sem. ETC/IT.

Unit-1: Microprocessor (Architecture and Programming-8085-8-bit).

1.1: Introduction to Microprocessor and Microcomputer & distinguish between them.

Microprocessor: is a computer processor which incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC) at most a few integrated circuits. The microprocessor is a multipurpose, clock driven, register based, digital-integrated circuit which accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary numeral system.

Any microprocessor based system having limited number of resources are called **microcomputers**. The main difference between microprocessor & microcomputer is that the microprocessor is a computer processor contained on an integral circuit chip and **microcomputer** is a small, relatively inexpensive computer.

1.2: Concept of Address Bus, Data Bus, Control bus & System Bus.

Bus is a group of conducting wires through which signals are passes.

A bus which is used to provide the communication between the major components of a computer is called as **System Bus**. A system bus is a single bus that connects the major components of a computer system, combining the functions of a data bus to carry information. The **Control Bus** is used to carry necessary control signals between the CPU and memory and I/O devices. A **control bus** is a computer **bus** that is used by the CPU to communicate with devices that are contained within the computer.

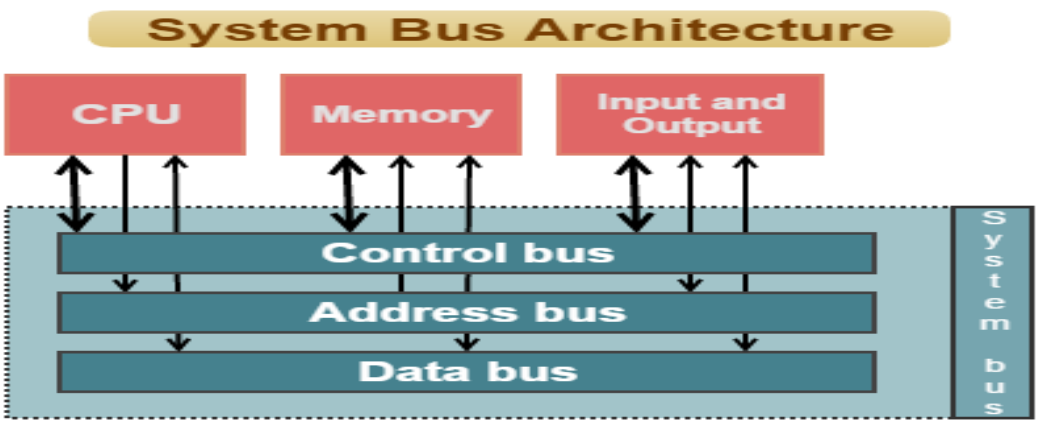
There are 3 types of Buses are used.

- **Address Bus**
- **Data Bus**
- **Control Bus**

The **Address bus** carries the address of a memory location or I/O device that the CPU wants to access. It is a unidirectional bus. The width of the address bus is decided by the designed memory addressing capability of the CPU. For example Intel 8085 has 16-bit address bus which gives 2 to the power 16 =64K byte memory **addressing capability.**

The width of **Data bus** is same as the word length of the CPU. **Data bus** has 8-bit length and it is used to transfer data between the CPU, memory and I/O devices. Both address bus & **Control Bus** are unidirectional where as data bus is bi-directional.

1.3:General Bus structure Block diagram.



1.4:Basic Architecture of 8085 (8 bit) Microprocessor.

Intel 8085 is a 8-bit, NMOS Microprocessor that can deal with the memory of 64K Byte. This microprocessor consists of 40-pins as well as works with +5V power supply. This processor can be work at a 3MHz of maximum frequency. This processor is available in three versions such as 8085 AH, 8085 AH1, and 8085 AH2 which are designed with HMOS technology. The highly developed versions use 20% of the power supply. The CLK frequencies of the versions of this processor are 8085 A- 3 MHz, 8085AH-3 MHz, 8085 AH2-5 MHz, and 8085 AH1

ALU: The arithmetic and logic unit, ALU performs the following arithmetic and logical operation.

a. Addition.
b. Subtraction.
c. Logical AND
d. Logical OR
e. Logical Exclusive OR
f. Complement
g. Increment
h. Decrement etc.

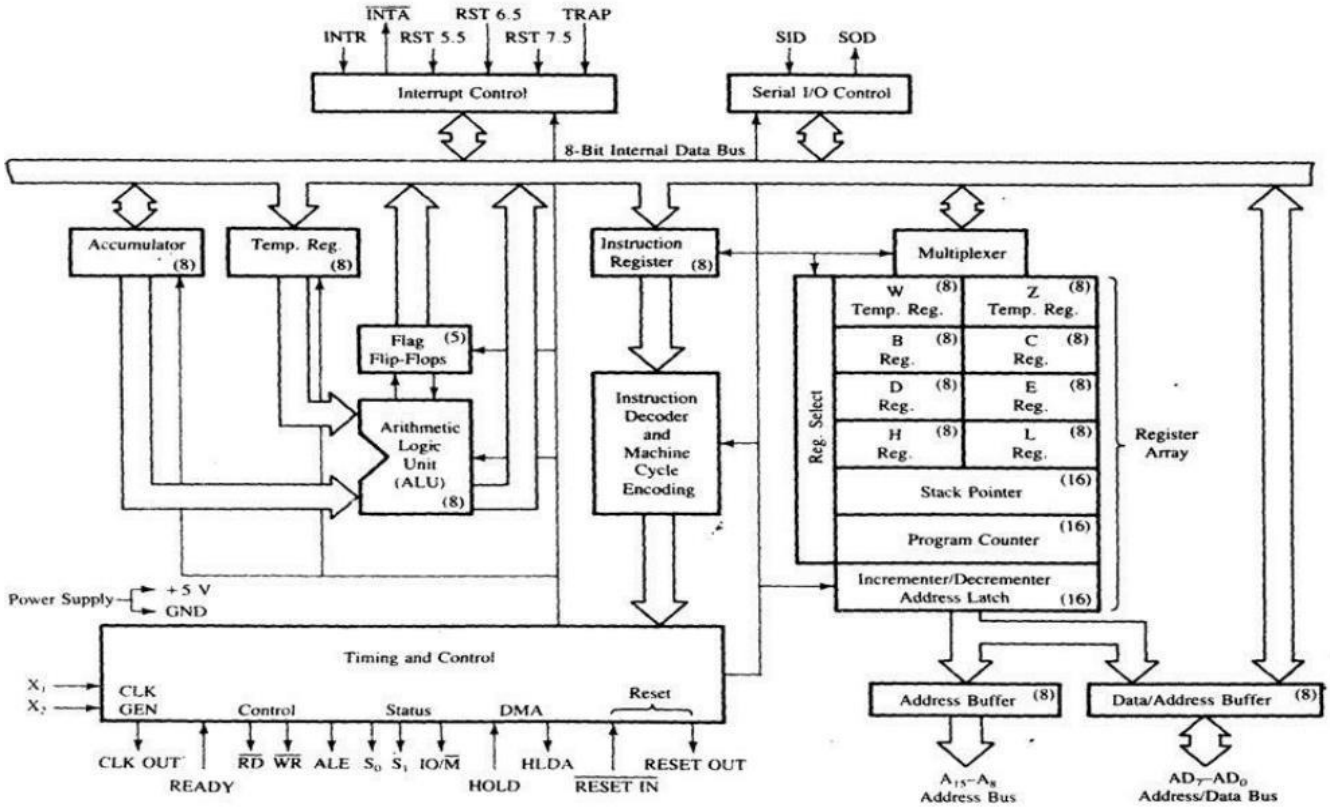
Timing and Control Unit:

The Timing & Control Unit is a section of the CPU

Registers: Registers are used by the microprocessor for temporary storage and manipulation of data and instructions. Data remain in the registers till they are sent to the memory or I/O devices. In a large computer the number of registers is more and hence the program requires less transfer of data to/from the memory. In a small computer the number of registers is small due to limited size of the chip. Intel 8085 microprocessor has the following registers.

- One 8-bit accumulator (ACC)i.e register A
- Six 8-bit General Purpose Registers. These are B,C,D,E,H and L
- One 16-bit Stack Pointer, SP
- One 16-bit Program Counter, PC
- Instruction Register
- Temporary Register

The 8085 Architecture



ACCUMULATOR (ACC): The accumulator is an 8-bit register associated with the ALU. The register **'A'** in the 8085 is an Accumulator. It is used to hold one of the operands of an arithmetic or logical operation. It serves as one input to the ALU. The other operand for an arithmetic or logical operation may be stored either in the memory or in one of the **General Purpose Registers (GPR)**. The final result of an arithmetic or logical operations placed in the Accumulator. Such instructions do not require any other register or memory location because there is no other operand .There is one typical instruction DAD rp, for 16-bit addition for which one of the 16-bit operands is kept in H-L pair and the other in the B-C or D-E pair . The result is placed in the H-L pair.

General Purpose Registers (GPR): The 8085 microprocessor contains six 8-bit general purpose registers. They are: B,C,D,E,H and H and L register. To hold 16 bit data a combination of two 8-bit registers can be employed . The combination of two 8-bit registers is known as a register pair. The valid register pairs in the 8085 are: B-C,D-E and H-L. The programmer cannot from a register pair by selecting any two registers of his choice. The H-L pair is used to act as memory pointer and for this purpose it holds the 16-bit address of a memory location. The general purpose registers and the accumulator are accessible to programmer. He can store data in these registers during writing his program.

Special Purpose Register (SPR):

- **Program Counter (PC)**: It is a 16-bit special purpose register. It is used to hold the memory address of the next instruction to be executed. It keeps the track of memory addresses of the instructions in a program while they are being the execution of an instruction so that it points to the address of the next instruction in the program at the end of the execution of an instruction.
- **Stack Pointer (SP**): It is a 16 bit special purpose register. The stack is a sequence of memory locations set aside by the programmer to store / retrieve the contents of Accumulator. Since stack works on LIFO (Last-In-First-Out) principle, its operation is faster compared to normal memory locations. The contents

of only those registers are saved, which are needed in the later part of the program. The SP holds the address of the top element of data stored in the stack.

Instruction Register: The Instruction Register holds the op-code (Operation Code) of the Instruction which is being decoded and executed.

Temporary Register: It is an 8-bit register associated with the ALU. It holds data during an arithmetic/logical operation.

Flags: The Intel 8085 microprocessor contains five flags to serve as status flags. The Flip Flops are set or reset according to the conditions which arise during arithmetic or logical operation. If a flip flop for a particular flag is set, it indicates 1, when it reset, it indicates 0.

a. Carry Flag (CS)
b. Parity Flag (P)
c. Auxiliary Carry Flag (AC)
d. Zero Flag (Z)
e. Sign Flag (S)

Carry Flag (CS): After addition of two 8 bit no, if the sum is larger than 8 bit, a carry is produced and carry flag (CS) is set to 1 **or** in case of subtraction if borrow occurs then CS is set to 1, otherwise it is 0.

Parity Flag (P): It is set to 1, If the result of an arithmetic operation/ logical operation contains even no of 1s, if odd no of 1s, it is set to 0.

Auxiliary Carry Flag (AC): If it holds the carry from 3rd bit to 4th bit it is set to 1, otherwise it is 0.

Zero Flag (Z): Z is set to 1, If the result of an arithmetic operation/ logical operation is 0. If the result is not Zero, it is set to 0.

Sign Flag (S): The Sign flag S is set to 1, if the result of an arithmetic operation/ logical operation is negative, If the result is positive, it is set to 0.

PSW: Five bits indicates status Flag and three flags indicates undefined. The combination of 8 bit is called **Program Status Word (PSW).**

Ex: 1 1 0 0 1 0 1 1 C B
1 1 1 0 1 0 0 1 E 9
1 1 0 1 1 0 1 0 0

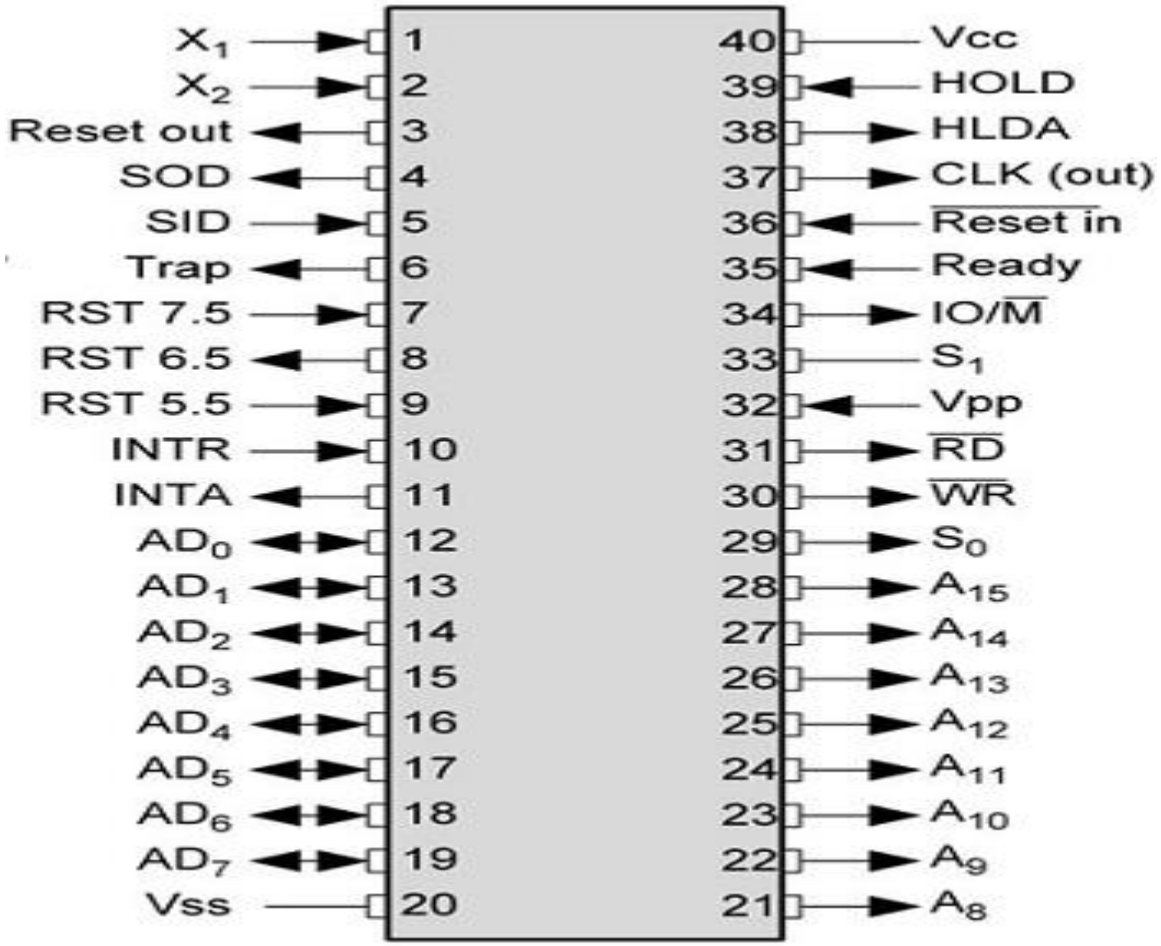
7	6	5	4	3	2	1	0	Bit no
S	Z	X	AC	X	P	X	CS	
1	0	1	1	0	1	0	0	

1. There is a carry CS is set to 1.
2. P is set to 1, even no of 1s.
3. There is a carry from 3rd to 4th bit AC is set to 1.
4. Result is Non-Zero Z is 0.
5. MSB of the Sum is 1, S is set to 1.

1.5:Signal Description (Pin diagram) of 8085 Microprocessor.

Pin Diagram of Intel 8085 Microprocessor

The 8085 microprocessor is an 8-bit general purpose processor that can deal with the memory of 64K Byte. This microprocessor consists of 40-pins as well as works with +5V power supply. This processor can be work at a 3MHz of maximum frequency. This processor is available in three versions such as 8085 AH, 8085 AH1, and 8085 AH2 which are designed with HMOS technology. The highly developed versions use 20% of the power supply. The CLK frequencies of the versions of this processor are 8085 A- 3 MHz, 8085AH-3 MHz, 8085 AH2-5 MHz, and 8085 AH1



Address Bus (A8-A15) (output):

The address bus pins are ranges from **A8 to A15** and these are mainly applicable to the most considerable memory address bit.

Address Bus and Data Bus: (Input/Output):

The address bus is a group of sixteen lines i.e A0-A15. The address bus is unidirectional. These are time multiplexed address/data i.e they serve dual purpose. They are used for 8 bit LSB of the memory address or I/O address during 1st clock cycle of a machine cycle. Again they are used for data during 2nd and 3rd clock cycle.

Address Latch Enable (ALE)(output)– It is an Address Latch Enable signal. It goes high during first T state of a machine cycle and enables the lower 8-bits of the address to be latched either into the memory or external latch.

- **IO/M' (output) –** It is a status signal which determines whether the address is for input-output or memory. When it is high(1) the address on the address bus is for input-output devices. When it is low(0) the address on the address bus is for the memory.
- **SO, S1 (output) –** These are status signals sent by the microprocessor. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.

S1	S0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

Power Supply and Clock Frequency:

- **Vcc –** +5v power supply
- **Vss –** Ground Reference
- **XI, X2 (input)–** These are the terminals to be connected to an external crystal oscillator which drives an internal circuitry of the microprocessor to produce a suitable clock for the operation of microprocessor. The

frequency is internally divided by two, therefore, to operate a system at 3MHZ the crystal should have frequency of 6MHZ.

- **CLK (OUT) –** This signal can be used as the system clock for other devices. Its frequency is same at which processor operates.

  READY (input) – It is used by the microprocessor to sense whether a peripheral is ready to transfer data or not. If READY is high the peripheral is ready. If it is low the microprocessor waits till it goes high.

  RD' (output) – It is a signal to control READ operation When it goes low the selected memory or I/O device is read.

  WR' (output) – It is a signal to control WRITE operation. When it goes low the data on the data bus is written in to selected memory or I/O location.

The 8085 has five interrupt signals that can be used to interrupt a program execution.

(i) TRAP
(ii) RST 7.5
(iii) RST 6.5
(iv) RST 5.5
(v) INTR

TRAP

These are interrupts. The microprocessor acknowledges Interrupt Request by INTA' signal. In addition to Interrupts, there are three externally initiated signals namely RESET, HOLD and READY. To respond to HOLD request, it has one signal called HLDA.

- **INTR (input)–** It is an interrupt request signal.
- **INTA' (output) –** It is an interrupt acknowledgment sent by the microprocessor after INTR is received.
- **RESET IN' (input) –** When the signal on this pin is low(0), the program-counter is set to zero, the buses are tristated and the microprocessor unit is reset.
- **RESET OUT (output) –** This signal indicates that the MPU is being reset. The signal can be used to reset other devices.

- **HOLD (input) –** It indicates that another device is requesting the use of the address and data bus. Having received HOLD request the microprocessor relinquishes the use of the buses as soon as the current machine cycle is completed. Internal processing may continue. After the removal of the HOLD signal the processor regains the bus.
- **HLDA (output) –** It is a signal which indicates that the hold request has been received after the removal of a HOLD request, the HLDA goes low.

Serial transmission in 8085 is implemented by the two signals,

- **SID (input)–** SID is a data line for serial input. The data on this line is loaded into 7^{th} bit of the accumulator when RIM instruction is executed.
- **SOD (output) –**SOD is a data line for serial output. The 7^{th} bit of the accumulator is output on SOD line when SIM instruction is executed.

1.6: Register Organizations, Distinguish between SPR& GPR, Timing & Control Module.

A **register** is a temporary storage area built into a CPU. Some **registers** are used internally and cannot be accessed outside the processor, while others are user-accessible. Most modern CPU architectures include both types of **registers.** These **registers** are used to store or copy temporary data, by using instructions, during the execution of the program.

There are two types of register in Intel 8085 Microprocessor.

a. **General Purpose Registers (GPR)**
b. **Special Purpose Register (SPR)**

Register organization is the arrangement of the **registers** in the processor. The processor designers decide the **organization** of the **registers** in a processor.

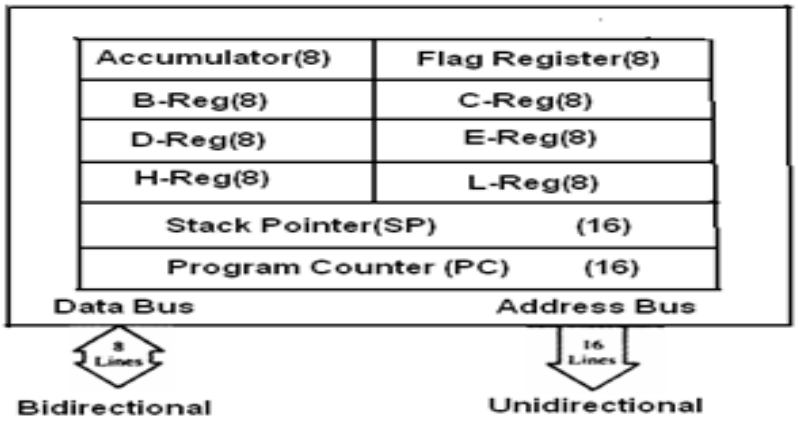


Fig. Register organization

General Purpose Registers (GPR): The 8085 microprocessor contains six 8-bit general purpose registers. They are: B,C,D,E,H and H and L register. To hold 16 bit data a combination of two 8-bit registers can be employed . The combination of two 8-bit registers is known as a register pair. The valid register pairs in the 8085 are: B-C,D-E and H-L. The programmer cannot from a register pair by selecting any two registers of his choice. The H-L pair is used to act as memory pointer and for this purpose it holds the 16-bit address of a memory location. The general purpose registers and the accumulator are accessible to programmer. He can store data in these registers during writing his program.

Special Purpose Register (SPR):

- **Program Counter (PC)**: It is a 16-bit special purpose register. It is used to hold the memory address of the next instruction to be executed. It keeps the track of memory addresses of the instructions in a program while they are being the execution of an instruction so that it points to the address of the next instruction in the program at the end of the execution of an instruction.
- **Stack Pointer (SP**): It is a 16 bit special purpose register. The stack is a sequence of memory locations set aside by the programmer to store / retrieve the contents of Accumulator. Since stack works on LIFO (Last-In-First-Out) principle, its operation is faster compared to normal memory locations. The contents of only those registers are saved, which are needed in the later part of the program. The SP holds the address of the top element of data stored in the stack.

1.7. Stack, Stack pointer & stack top:

Stack : During the execution of a programme sometimes it becomes necessary to save the contents of certain registers because the registers are required for some other operation in later stage. These contents are moved to memory locations by **PUSH** operation. After completing these operations those contents which were saved in the memory are transferred back to the registers by **POP** operation. The set of memory locations kept for this purpose is called **Stack.**

The last memory location of the stack is called **Stack Top.** The **Stack** Pointer register will hold the address of the top location of the **stack**.

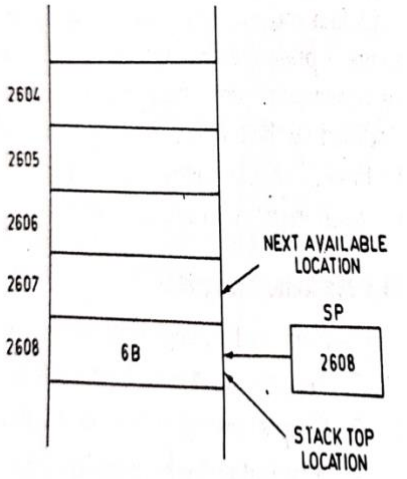


Fig. 5.1. Stack and stacktop location.

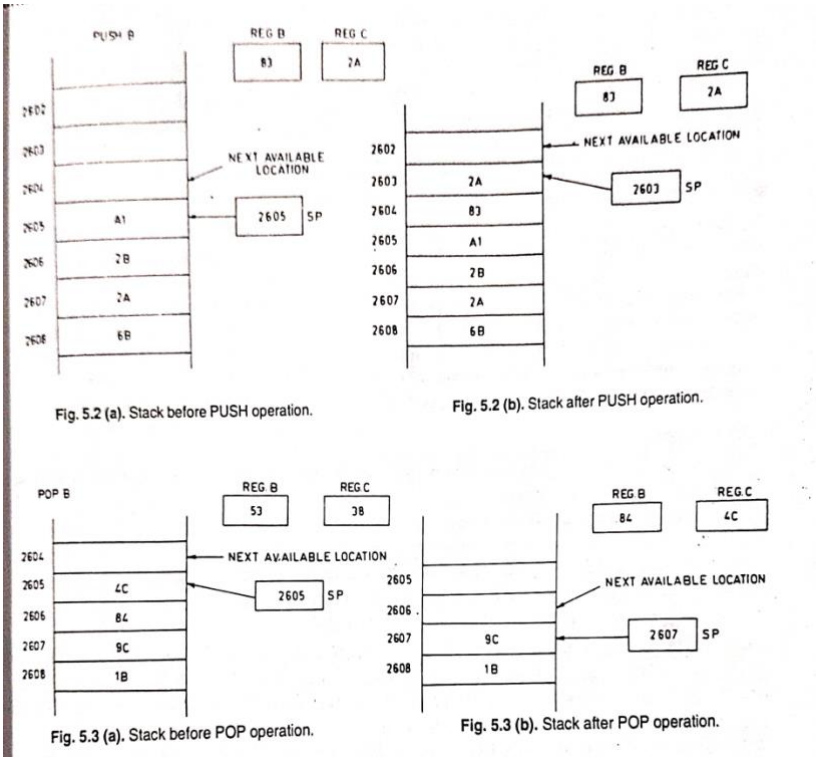


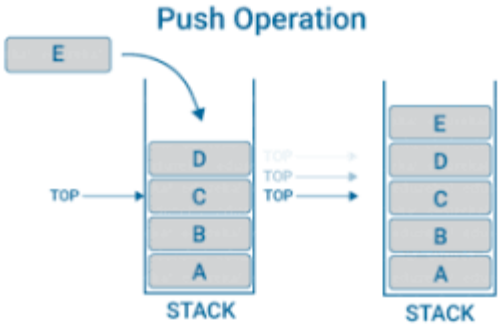
Fig. 5.2 (a). Stack before PUSH operation.

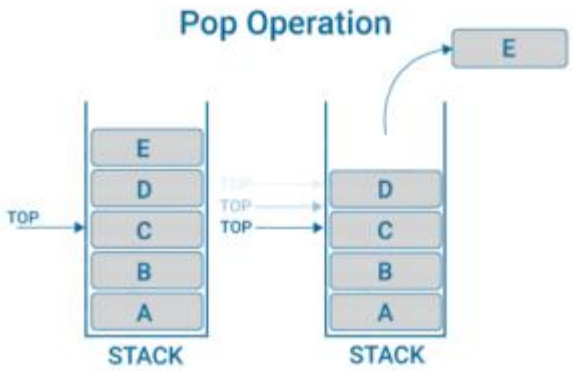
Fig. 5.2 (b). Stack after PUSH operation.

Fig. 5.3 (a). Stack before POP operation.

Fig. 5.3 (b). Stack after POP operation.

Stack Pointer (SP): It is a 16 bit special purpose register. The stack is a sequence of memory locations set aside by the programmer to store / retrieve the contents of Accumulator. Since stack works on **LIFO (Last-In-First-Out)** principle, its operation is faster compared to normal memory locations. The contents of only those registers are saved, which are needed in the later part of the program. The SP holds the address of the top element of data stored in the stack.





1.8: Interrupts:-8085 Interrupts, Masking of Interrupt(SIM,RIM).

Interrupts in 8085

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. **TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR.**

Interrupt are classified into following groups based on their parameter –

- **Vector interrupt** – In this type of interrupt, the interrupt address is known to the processor. **For example:** RST7.5, RST6.5, RST5.5, TRAP.
- **Non-Vector interrupt** – In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to perform interrupts. **For example:** INTR.
- **Maskable interrupt** – In this type of interrupt, we can disable the interrupt by writing some instructions into the program. **For example:** RST7.5, RST6.5, RST5.5.

- **Non-Maskable interrupt** − In this type of interrupt, we cannot disable the interrupt by writing some instructions into the program. **For example:** TRAP.
- **Software interrupt** − In this type of interrupt, the programmer has to add the instructions into the program to execute the interrupt. There are 8 software interrupts in 8085, i.e. RST0, RST1, RST2, RST3, RST4, RST5, RST6, and RST7.
- **Hardware interrupt** − There are 5 interrupt pins in 8085 used as hardware interrupts, i.e. TRAP, RST7.5, RST6.5, RST5.5, INTA.

Note −INTA is not an interrupt, it is used by the microprocessor for sending acknowledgement. TRAP has the highest priority, then RST7.5 and so on.

Interrupt Service Routine (ISR)
A small program or a routine that when executed, services the corresponding interrupting source is called an ISR.

The 8085 has five interrupt signals that can be used to interrupt a program execution.
(i) TRAP
(ii) RST 7.5
(iii) RST 6.5
(iv) RST 5.5
(v) INTR
TRAP
TRAP is a non-maskable interrupt, having the highest priority among all interrupts. By default, it is enabled until it gets acknowledged. In case of failure, it executes as **ISR** and sends the data to backup memory. The microprocessor acknowledges Interrupt Request by INTA' signal. In addition to Interrupts, there are three externally initiated signals namely RESET, HOLD and READY. To respond to HOLD request, it has one signal called HLDA.

- **INTR –** It is an interrupt request signal.
- **INTA' –** It is an interrupt acknowledgment sent by the microprocessor after INTR is received.

RST7.5
It is a maskable interrupt, having the second highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 003CH address.

RST 6.5
It is a maskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 0034H address.
RST 5.5
It is a maskable interrupt. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 002CH address.
INTR
It is a maskable interrupt, having the lowest priority among all interrupts. It can be disabled by resetting the microprocessor.

SIM: Set Interrupt Mask
It is a 1 byte instruction and it is a multi-purpose instruction. The main use of SIM instruction are- Masking/unmasking of RST 7.5, RST6.5 and RST 5.5. It is used for Serial output of Data (SOD).

RIM: Read Interrupt Mask
It is used
To check whether RST are masked or not.
To check whether interrupts are enable or not.
To perform Serial input data (SID).

2.1: Addressing data & Differentiate between one-byte, two-byte &three-byte instructions with examples.

A Digital Computer understands instructions written in binary codes (machine codes). The machine codes of all instructions are not of the same length. According to the word size Intel 8085 instructions are classified into the following three types.

1. 1-byte instruction.
2. 2- byte instruction.
3. 3-byte instruction.

Examples of 1-byte instruction.

MOV A,B machine code=78
ADD B machine code=80
RAL machine code=17
HLT machine code=76

Examples of 2-byte instruction.

A 2 byte Instruction is stored in two consecutive memory location.

MVI B,05 machine code=06,05

The 1st byte 06 is the machine code for MVI B and 2nd byte 05 is the data which is to be moved to register B.

IN 01 machine code= DB,01

DB is the machine code for the instruction IN and 01 is the address of a port B for interfacing.

Examples of 3-byte instruction.

LXI H,2400H machine code=21,00,24
LDA 2500H machine code= 3A,00,25

2.2:Addressing modes in instructions with suitable examples.

Addressing Modes of Intel 8085 Microprocessor

The way of specifying data to be operated by an instruction is known as **addressing modes**.Each instruction requires certain data on which it has to operate. There are various types of techniques to specify data for instructions. These techniques are called **addressing modes.**

There are 5 types of addressing modes are used

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 mode of addressing the address of the operand (data) is given in the instruction itself.
    1. **STA 2400H**
       (32,00,24) Store the content of the accumulator in the memory location 2400H.
    2. **IN 02**
       (DB,02) Read data from the port C.
2. **Register Addressing Mode:** In Register Addressing Mode the operand is one of the General Purpose Register. The opcode specifies the address of the register in addition to the operation to be performed.
    1. **MOV A,B**
       (78) Move the content of register B to register A.

2. ADD B

(80) Add the content of register B to the content of register A.

3. **Register Indirect Addressing Mode:** In this mode of addressing the address of the operand is specified by a register pair.

1. LXI H,2100H
MOV A,M
HLT

(MOV A,M is the example of register indirect Addressing Mode)

2. LXI H,2500 H
ADD M
HLT

(ADD M is the example of register indirect Addressing Mode)

4. **Immediate Addressing Mode:** In immediate Addressing Mode the operand is specified within the instruction itself.

1. MVI A,05.
2. ADI 06

(Examples of Immediate Addressing Mode)

5. **Implicit Addressing Mode:** There are some instructions which operate on the content of the accumulator. Such instructions do not require the address of the operand. **Ex-CMA, RAL, RAR.**

2.3: Instruction Set of 8085(Data Transfer, Arithmetic, Logical, Branching, Stack& I/O , Machine Control).

In **microprocessor**, the **instruction set** is the collection of the **instructions** that the **microprocessor** is designed to execute. The programmer writes a program in assembly language using these **instructions**. An **instruction** is a binary pattern designed inside a **microprocessor** to perform a specific function. The entire group of **instructions** that a **microprocessor** supports is called **Instruction Set**. **8085** has 246 **instructions**.
These instructions are of Intel Corporation. They cannot be used by other microprocessor manufacturers. These instructions are classified into the following Groups.

a. Data Transfer Group.
b. Arithmetic Group.
c. Logical Group.
d. Branch Control Group.
e. I/O and Machine Control Group.

Data Transfer Group:
Instructions which are used to transfer data from one register to another register, from memory to register or register to memory comes under this group.

MOV r1, r2 (Move data, Move the contents of one register to another)
MOV r, M (Move the contents of memory to register)
MOV M, r (Move the contents of register to memory)
MVI r, data (Move Immediate data to register)
MVI M, data (Move Immediate data to memory
LXI rp, data16 (Load register pair immediate)
LDA addr (Load Accumulator Direct)
STA addr (Store Accumulator Direct)
LHLD addr (Load H-L pair direct)
SHLD addr (Store H-L pair direct)
LDAX rp (Load accumulator indirect)
STAX rp (Store accumulator indirect)
XCHG (Exchange the content of H-L pair to D-E pair)

Arithmetic Group:

The instructions in this group perform arithmetic operations.

ADD r (Add register to accumulator)
ADD M(Add Memory to accumulator)
ADC r(Add register with carry to accumulator)
ADC M(Add Memory with carry to accumulator)
ADI data(Add immediate data to accumulator)
ACI data(Add with carry immediate data to accumulator)
DAD rp(Add register pair to H-L pair))
SUB r(Subtract register from accumulator)
SUB M(Subtract Memory from accumulator)
SBB r(Subtract register from accumulator with borrow)
SBB M(Subtract memory from accumulator with borrow)
SUI data(Subtract immediate data from accumulator)
SBI data(Subtract immediate data from accumulator with borrow)
INR r(Increment register content)
INR M(Increment memory content)
DCR r(Decrement register content)
DCR M(Decrement memory content)
INX rp(Increment register pair)
DCX rp(Decrement register pair)
DAA (Decimal Adjust Accumulator)

Logical Group:
The instructions in this group perform logical operations.

ANA r (AND register with accumulator)
ANA M (AND memory with accumulator)
ANI data (AND immediate data with accumulator)
ORA r (OR register with accumulator)
ORA M (OR memory with accumulator)
ORI data (OR immediate data with accumulator)
XRA r (Exclusive OR register with accumulator)
XRA M (Exclusive OR memory with accumulator)
XRI data (Exclusive OR immediate data with accumulator)
CMA (Complement the accumulator)
CMC(Complement the carry status)
STC(Set carry status)
CMP r(Compare register with accumulator)
CMP M(Compare memory with accumulator)
CPI data(Compare immediate data with accumulator)
RLC(Rotate accumulator left)
RRC(Rotate accumulator right)
RAL(Rotate accumulator left through carry)
RAR(Rotate accumulator right through carry)

Branch Group:
This group includes the instructions for conditional and unconditional jump, subroutine call and return and restart.
There are two types of branch instructions
1. Conditionally
2. Unconditionally

The conditional branch instructions transfer the program to the specified label when certain condition is satisfied.
The unconditional branch instructions transfer the program to the specified label unconditionally.

JMP addr (label) (Unconditional jump. to the instruction specified by the address)
JZ addr (label)(Jump if the result is zero)
JNZ addr (label)(Jump if the result is not zero)
JC addr (label)(Jump if there is a carry)
JNC addr (label)(Jump if there is no carry)
JP addr (label)(Jump if the result is plus)
JM addr (label)(Jump if the result is minus)
JPE addr (label)(Jump if even parity)
JPO addr (label)(Jump if odd parity)
CALL addr (label)(Unconditional call, call subroutine identified by the address)
RET.(Return from subroutine)
RST n(Restart)
PCHL(Jump to address specified by H-L pair)

Stack, I/O and Machine Control Group:
This group includes the instructions for input/output ports, stack and machine control.

IN port-address(Input to accumulator from I/O port)
OUT port-address(Output from accumulator to I/O port)
PUSH rp(Push the content of register pair to stack)
PUSH PSW(PUSH Processor Status Word)
POP rp(Pop the content of register pair, which was saved, from the stack)
POP PSW(POP Processor Status Word)
HLT(Halt)
XTHL(Exchange stack top with H-L)
SPHL(Movethe content of H-L pair to stack pointer)
EI (Enable Interrupt)
DI(Disable Interrupt)
SIM (Set Interrupt Mask)
RIM(Read Interrupt Mask)
NOP(No Operation)

Govt. Polytechnic, Bhubaneswar

(Summer-2021)

Unit-3: TIMING DIAGRAMS.

Faculty: Kanak Prava Swain, Lect.(ETC).

Subject: Microprocessor & Microcontroller.

Semester: 4th Sem. ETC/IT.

3.1: Define opcode, operand, T-State, Fetch cycle, Machine Cycle, Instruction cycle & discuss the concept of timing diagram.

Opcode: Each instruction contains two parts, Operation code (Opcode) and operand. The first part of the instruction which specifies the task to be performed by the computer is called opcode.

Operand: The second part of the instruction is the data to be operated on and it is called operand. The operand may be 8-bit or 16-bit data, 8-bit or 16-bit data address. In some instructions the operand is implicit. When operand is a register the data is the content of the register.

Example:

MVI A,08

Here MVI is opcode and A,08 is operand

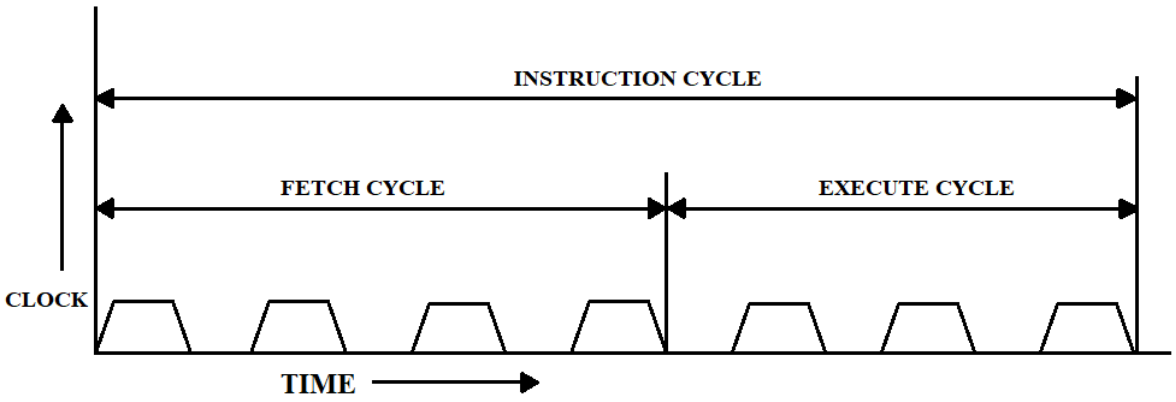
Instruction Cycle:

The necessary steps that a CPU carries out to fetch an instruction and necessary data from the memory and to execute it, constitute an Instruction cycle. An Instruction cycle consists of a fetch cycle and execute cycle.

IC=FC+EC

Fetch Operation: The 1st byte of an instruction is its opcode. In the beginning of a fetch cycle the content of the program counter, which is the address of the memory location where opcode is available, is sent to the memory. The entire operation of fetching an opcode takes three clock cycle. The clock cycle for which the CPU waits is called wait cycle.

Execute Operation: If the operand is in the general purpose register, execution is immediately performed. The time taken in decoding and execution is one clock cycle. A read cycle is similar to a fetch cycle. In write cycle the data are sent from the CPU to memory.



Machine Cycle: The necessary steps carried out to perform the operation of accessing either memory or I/O device, constitute a machine Cycle. i.e necessary steps carried out to perform a fetch, a read or a write

operation constitute a Machine Cycle. In a machine cycle one basic operation such as opcode fetch, memory read, memory write, I/O read or I/O write is performed. An instruction cycle consists of several machine cycles.

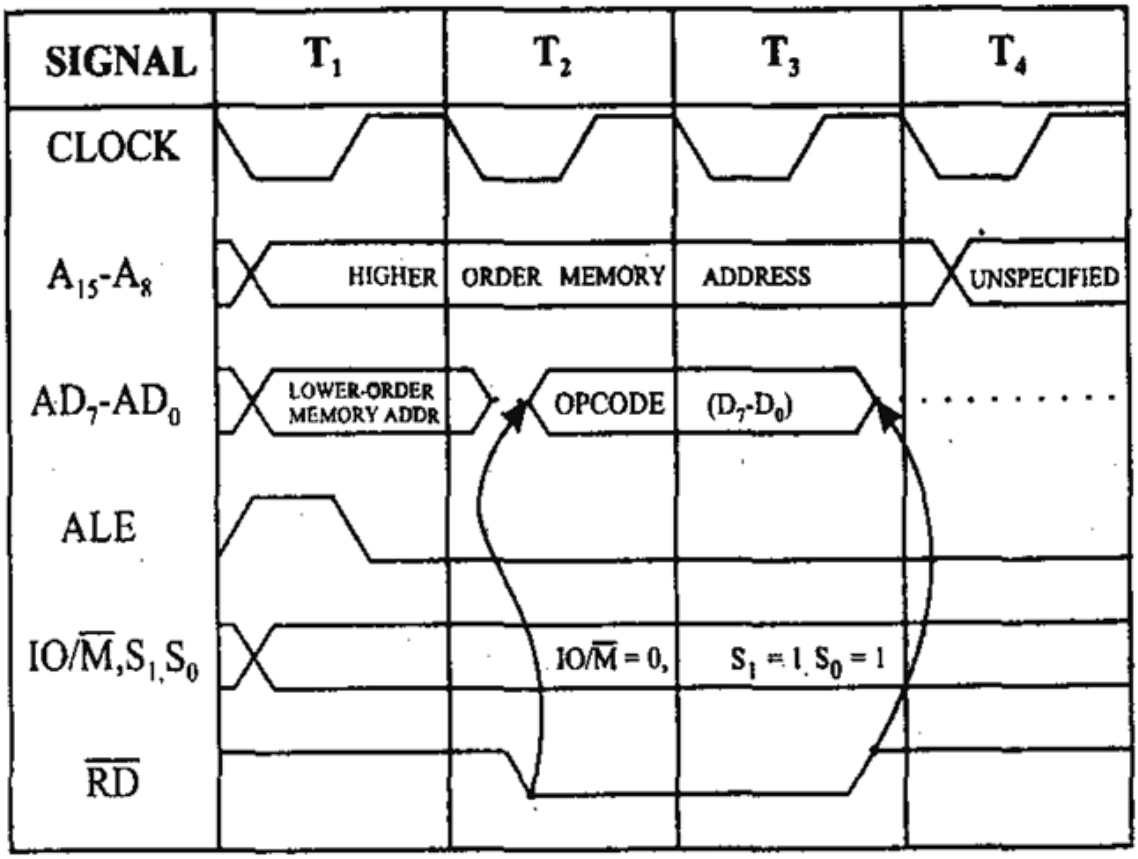
T-State: One Subdivision of an operation in one clock cycle is called a state or T-State.

Timing Diagram:

The necessary steps which are carried out in a machine cycle can be represented graphically. Such a graphically representation is called **timing diagram.** The timing diagram for opcode fetch, memory read, memory write, I/O read or I/O write.

Timing Diagram for opcode Fetch Cycle:

For opcode fetch cycle T1,T2,T3,T4 are consecutive four clock cycles. The microprocessor issues a low IO/M signal to indicate that it wants to make communicate with the memory. Again the microprocessor sends out high So & S1 signals to indicate that it is going to perform fetch operation.



During T1 the microprocessor sends out the address of the memory location where opcode is available. The 16 bit memory address is sent through Address/Data (AD) bus. 8-MSB of the memory address are sent through A8-A15 Bus and 8-LSB of the memory address are sent through AS0-AD7 bus. It is used in time-multiplexed mode. Therefore, it has to be made available to carry data during T2 & T3. The microprocessor sends an Address latch enable signal to latch the 8-LSB of the memory address. During T2 AD Bus becomes ready to carry data. In T2 microprocessor makes RD low. During T3 the opcode is placed in the Instruction register IR which is within the microprocessor. The memory is disabled when RD goes high during T3. The fetch cycle is completed by T3. The opcode is decoded in T4.

If the instruction is 1-byte long, one machine cycle is required to fetch & execute the instruction. If the instruction is 2-byte or 3-byte long, it requires more machine cycle.

3.2: Draw timing diagram for memory read, memory write, I/O read, I/O write machine cycle.

Example:

Timing diagram for MVI A,32

This instruction is 2-byte long i.e 3E,32. It requires two machine cycle M1 & M2. The first machine cycle M1 is to fetch the operation code 3E from the memory and the second machine cycle M2 is for reading the data (05) from the memory. Fetch operation consist of 4 T-state and Memory Read operation consist of 3 T-State. So for MVI A,05 7 T-State & 2 Machine cycle is required.

For Fetch Operation:

For opcode fetch cycle T1,T2,T3,T4 are consecutive four clock cycles. The microprocessor issues a low IO/M signal to indicate that it wants to make communicate with the memory. Again the microprocessor sends out high So & S1 signals to indicate that it is going to perform fetch operation.

During T1 the microprocessor sends out the address of the memory location where opcode is available. The 16 bit memory address is sent through Address/Data (AD) bus. 8-MSB of the memory address are sent through A8-A15 bus and 8-LSB of the memory address are sent through AS0-AD7 bus. It is used in time-multiplexed mode. Therefore, it has to be made available to carry data during T2 & T3. The microprocessor sends an Address latch enable signal to latch the 8-LSB of the memory address. During T2 AD Bus becomes ready to carry data. In T2 microprocessor makes RD low. During T3 the opcode is placed in the Instruction register IR which is within the microprocessor. The memory is disabled when RD goes high during T3. The fetch cycle is completed by T3. The opcode is decoded in T4.

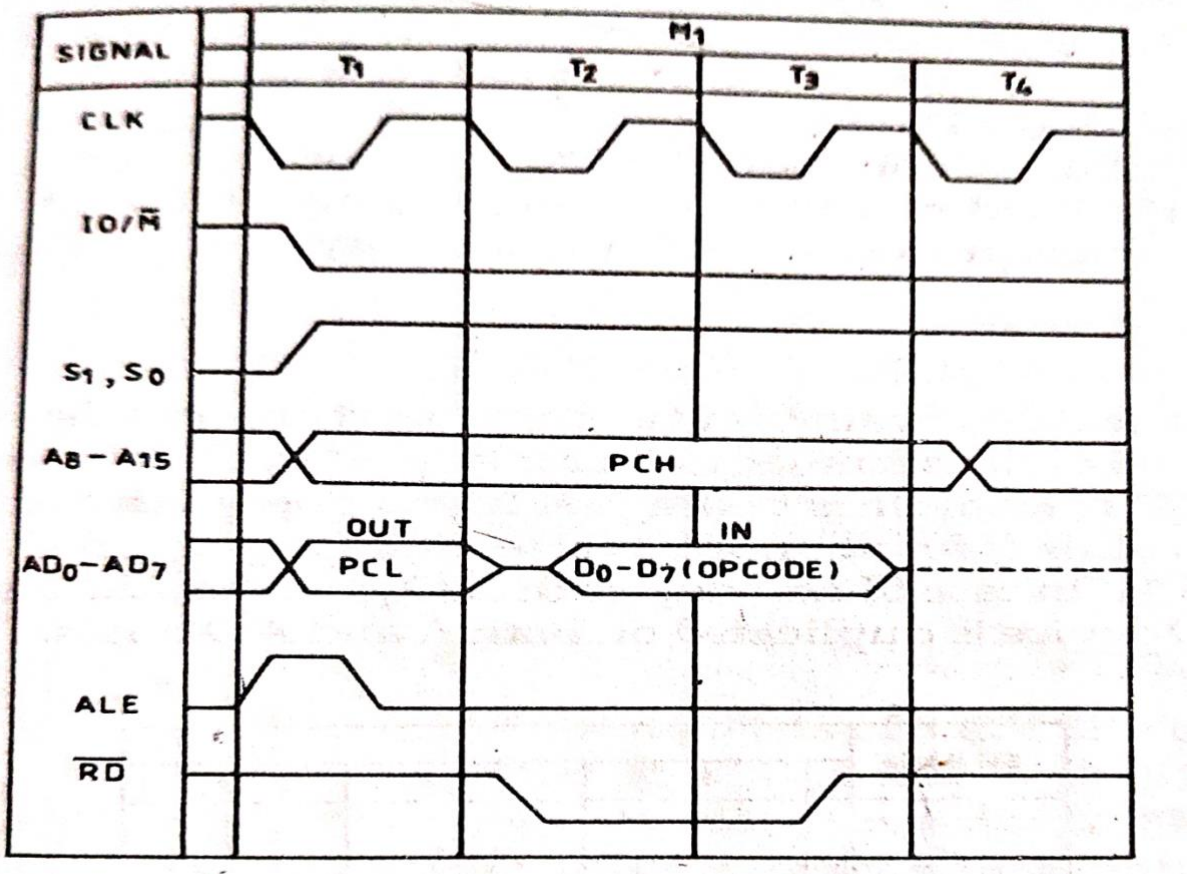


Fig. 3.9. Timing Diagram for Opcode Fetch Operation.

For Memory Read operation:

IO/M goes low indicating that the address is for memory. S1 & S0 are set to 1 & 0 respectively for read operation. On A8-A15 8-MSB of the memory address data 05 are sent. During T1 8-LSB memory address of the data are sent on AD0-AD7.RD goes low in T2. In T3 data enters into the CPU. In T3 RD goes high and disables the memory.

In **memory write operation** instead of RD, WR goes low during T2 and goes high during T3 that the write operation is terminated. The status signal S0 and S1 are 1 and 0 respectively for write operation. AD bus is not disabled during T2.

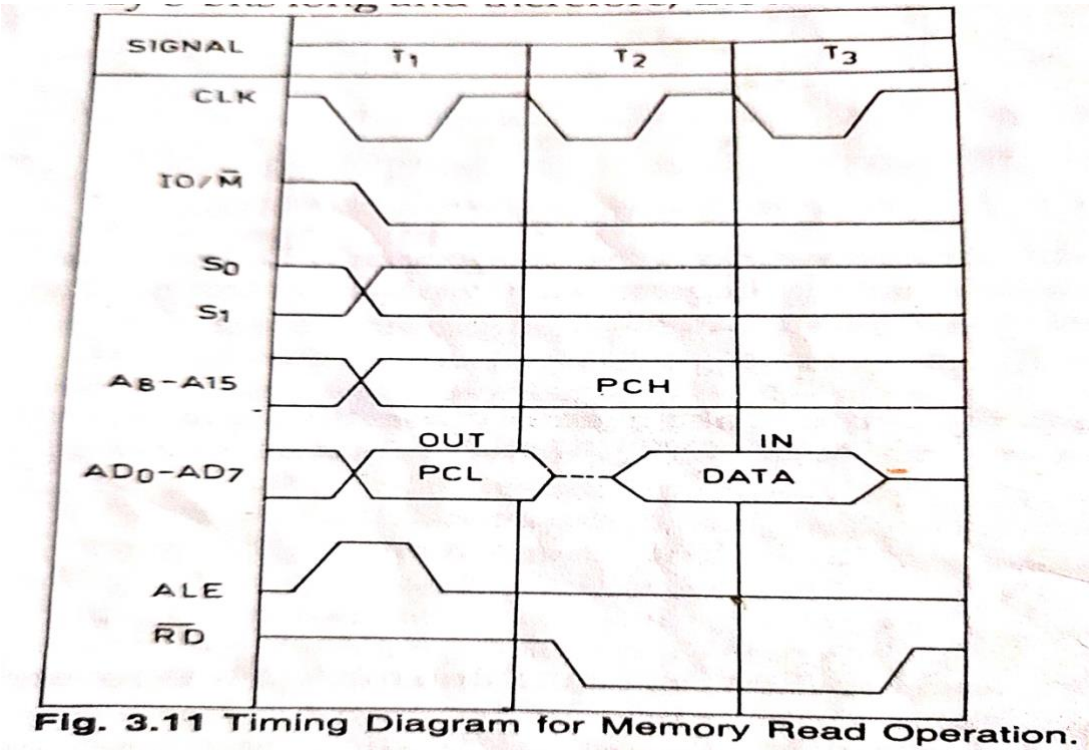


Fig. 3.11 Timing Diagram for Memory Read Operation.

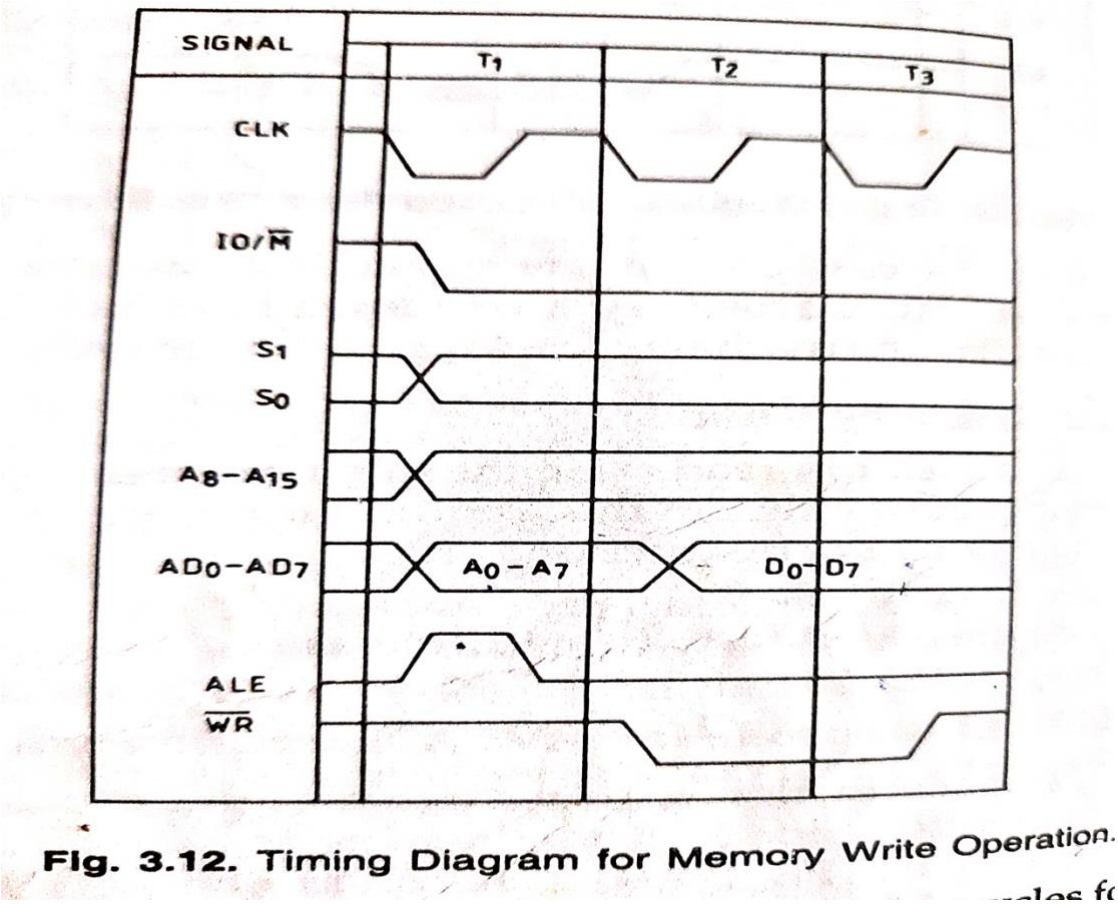


Fig. 3.12. Timing Diagram for Memory Write Operation.

I/O Read operation: In I/O Read cycle the microprocessor reads the data available at an input port or input device. An I/O Read cycle is similar to memory read cycle. The only difference between I/O Read cycle and memory read cycle is that IO/M signal goes high in I/O read cycle. In case of I/O device the address is only 8-bit long and is duplicated on both A and AD buses.

The IN instruction is used for I/O read. It requires three machine cycle, fetch cycle, memory read cycle to read input port address and I/O read cycle to read the data from the port.

I/O write operation:

In I/O Write cycle the CPU sends data to an I/O port from the accumulator. An I/O Write cycle is similar to memory Write cycle. The only difference between I/O Write cycle and memory read cycle is that IO/M signal goes high in I/O Write cycle. The address of the I/O port is duplicated on both A and AD buses.

The **OUT** instruction is used for I/O write. It requires three machine cycle, fetch cycle, memory read cycle for reading I/O device address from the memory and I/O write cycle for sending data to the I/O device.

Timing Diagram for MOV, MVI, LDA instruction

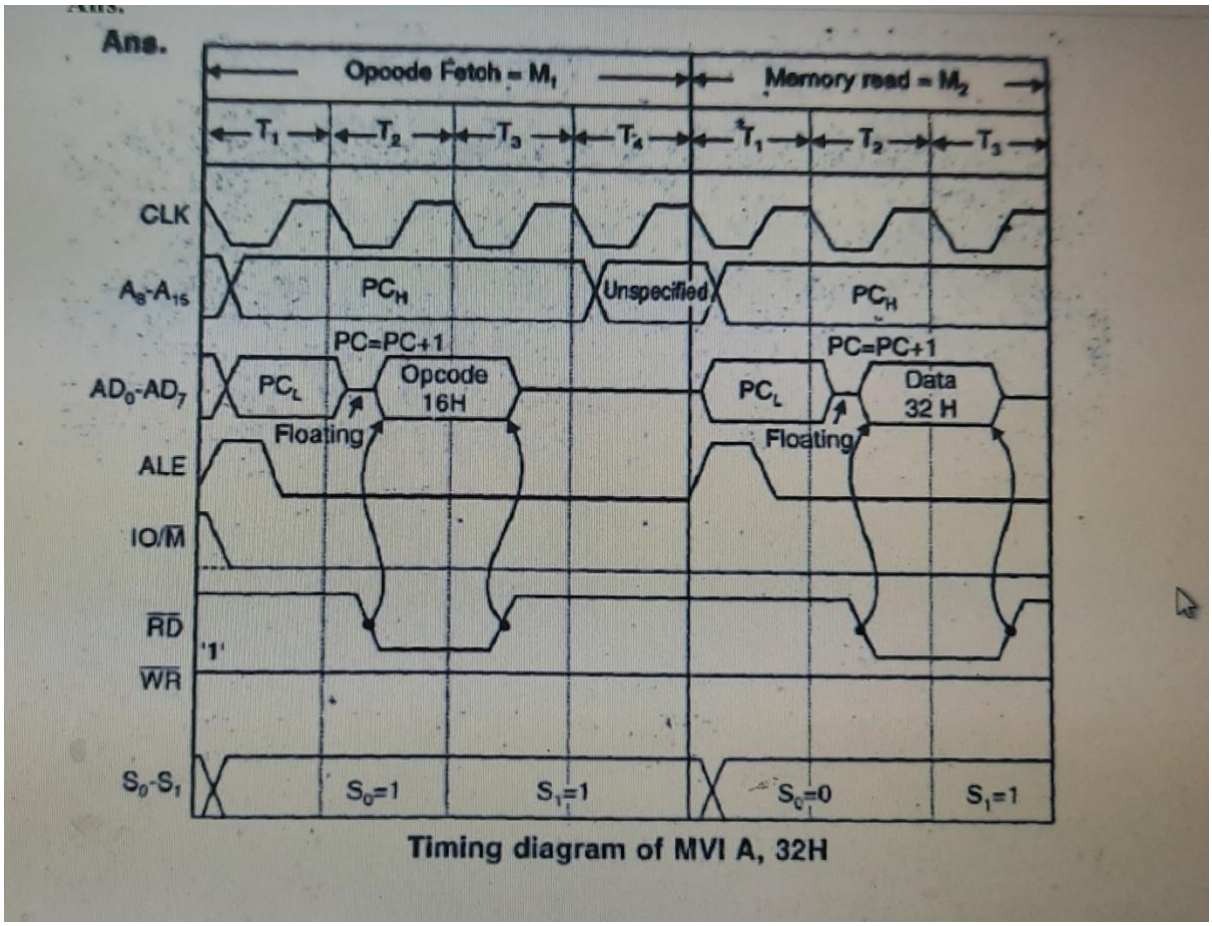
1. **MOV : (byte instruction/machine cycle/T-state)**
   - MOV R,R (1/1/4) Fetch Operation
   - MOV M,R (1/2/7) Fetch, Write Operation
   - MOV R,M (1/2/7) Fetch, Read Operation

2. **MVI: (byte instruction/machine cycle/T-state)**
   - MVI R, data (2/2/7) Fetch, Read Operation
   - MVI M, data(2/3/10) Fetch, Read, Write Operation

3. **LDA: (byte instruction/machine cycle/T-state)**
   - LDA address(3/4/13) Fetch, Read, Read, Read Operation

3.3:Draw a neat sketch for the timing diagram for 8085 instruction (MOV, MVI, LDA instruction).

Ans.



Timing diagram of MVI A, 32H

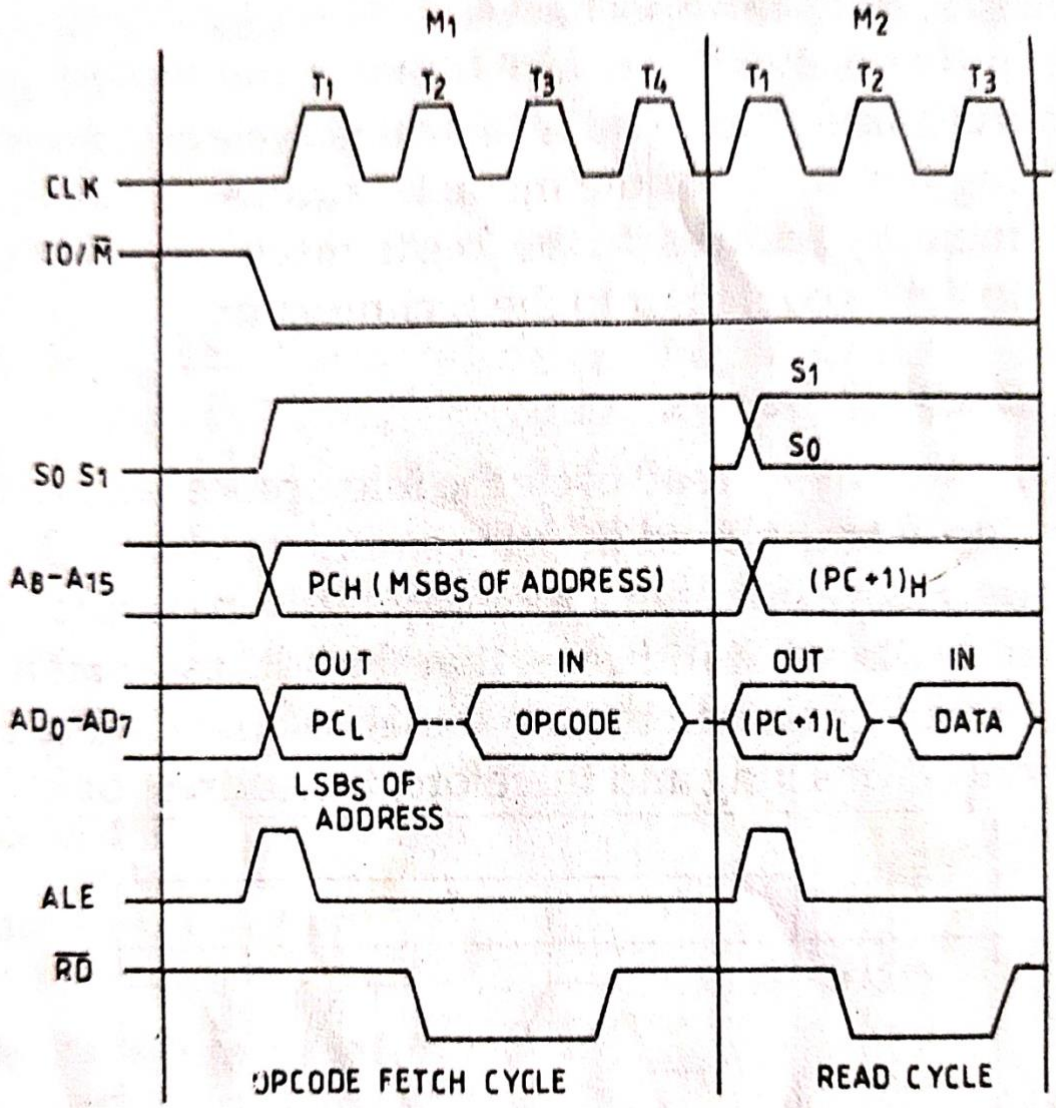


Fig. 3.10 Timing Diagram for MVI r, Data

Govt. Polytechnic, Bhubaneswar

(Summer-2021)

Unit-4: Microprocessor Based System Development Aids

Faculty: Kanak Prava Swain, Lect.(ETC).

Subject: Microprocessor & Microcontroller.

Semester: 4th Sem. ETC/IT.

4.1: Concept of interfacing.

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

Peripherals are connected to the microcomputer through electronic circuits known as **Interfacing** circuits. Each I/O device requires a separate interfacing circuit. The interfacing circuit converts the data available from the input device into compatible format for the computer. Some of the general purpose single chip **interfacing devices** are

- I/O port
- Programmable Peripheral Interface (PPI)
- DMA Controller
- Interrupt Controller
- Communication Interface

Some of the Special purpose **interfacing devices** are

- CRT Controller
- Floppy Disk Controller
- Key Board & Display interface

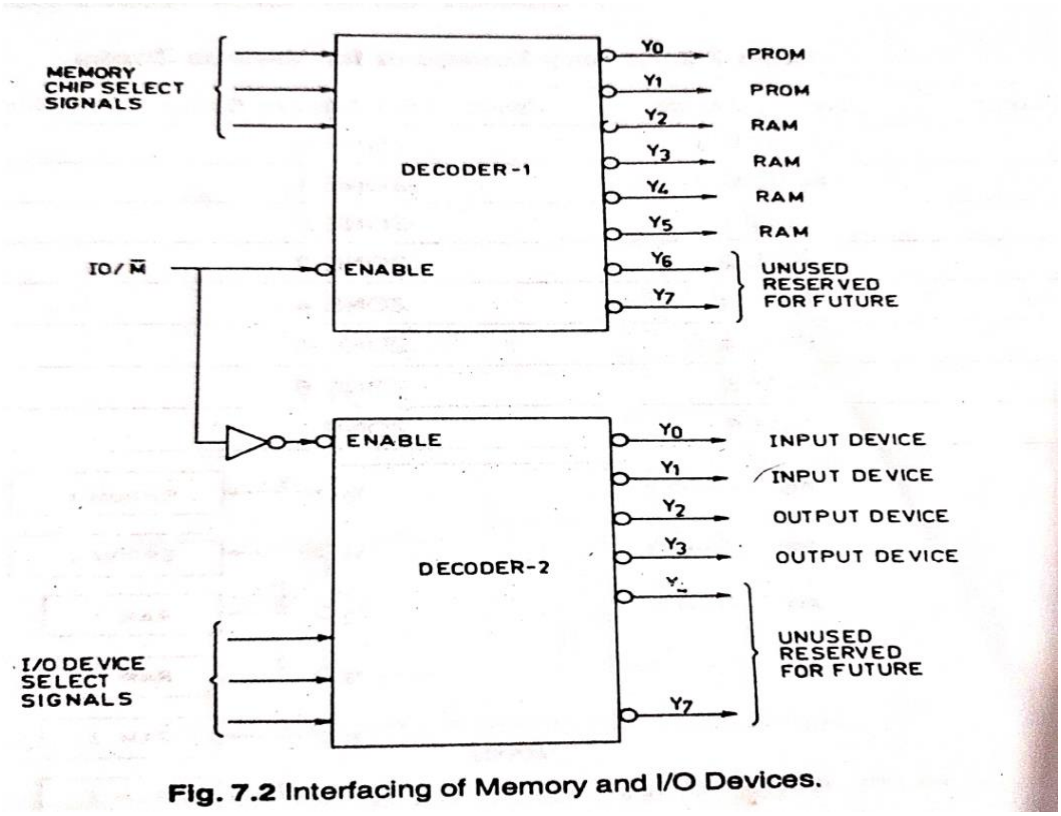


Fig. 7.2 Interfacing of Memory and I/O Devices.

In the diagram if I/O/M' is high the **decoder 2** is activated and the required I/O device is selected. If I/O/M' is low the **decoder 1** is activated and the required memory chip is selected.

4.2:Define Mapping & Data transfer mechanisms - Memory mapping & I/O Mapping.

Mapping:

Memory-mapping is a mechanism that **maps** a portion of a file, or **an** entire file, on disk to a range of addresses within **an** application's address space.

The Intel 8085 uses 16 bit address bus and it can access 2 16 =64 Kbytes. The 64 KB address are to be assigned to memory and I/O devices for their addressing. There are two schemes for the allocation of addresses to memory and input/ output devices.

- **Memory mapped I/O scheme.**
- **I/O Mapped I/O scheme.**

In Memory mapped I/O scheme there is only one address space. Some addresses are assigned to memories and some addresses to I/O devices. Suppose memory locations are assigned to the addresses 2000 to 24FF, any one of these addresses cannot be assigned to an I/O device. The addresses for I/O devices are different from the addresses which have been assigned to memories. **In I/O Mapped I/O scheme** the addresses assigned to memory location can also be assigned to I/O devices. Since the same addresses assigned to memory location or an I/O devices, the microprocessor issue a signal through IO/M to distinguish whether the address on the address bus is for a memory location or I/O device. When this **signal is low the address is for memory location** and if **it is high the address is for I/O device**. This scheme is used for large system.

Data Transfer Scheme:

Data transfer takes place between two devices such as microprocessor and memory, microprocessor and I/O devices and memory and I/O devices. A computer have several I/O devices of different speed. A slow I/O device can not transfer data when microprocessor issues instruction for the same because it takes some time to get ready. Data transfer scheme are classified as

- **Programmed data transfer scheme**
- **DMA (Direct Memory Access) Data Transfer Scheme**

Programmed data transfer scheme are controlled by the CPU. Data are transferred from I/O device to CPU or vice versa are control of programme. The Programmed data transfer scheme are classified into

- Synchronous Data transfer scheme
- Asynchronous Data transfer scheme
- Interrupt Driven Data transfer scheme

Synchronous Data transfer scheme

Synchronous means at the same time, the device which sends data and the device which receives data are Synchronised with the same clock. When the CPU & I/O devices match in speed, this technique of the data transfer is employed.

Asynchronous Data transfer scheme

Asynchronous means at irregular interval. This technique of the data transfer is used when the speed of I/O device does not match the speed of the microprocessor. In this technique the status of the I/O device i.e whether the device is ready or not is checked by the microprocessor before the data are transferred. When I/O device becomes ready, the microprocessor sends instruction to transfer data. This mode of data transfer is called handshaking mode of data transfer because some

signals are exchanged between the I/O device & microprocessor before the actual data transfer takes place.

DMA (Direct Memory Access) Data Transfer Scheme

In DMA data transfer scheme CPU does not participate. Data are directly transferred from an I/O device to memory or vice versa. The data transfer is controlled by the I/O device or a DMA controller. This scheme is employed when large amount of data are to be transferred.

An I/O device which wants to send data using DMA technique, sends the HOLD signal to the CPU. On receiving a HOLD signal from an I/O device the CPU gives up the control of buses as soon as the current machine cycle is completed.

DMA data transfer scheme is a faster scheme as compared to programmed data transfer scheme. It is used to transfer data from mass storage devices such as hard disks, floppy disks etc. It is also used for high speed printer.

4.3:Concept of Memory Interfacing:- Interfacing EPROM & RAM Memories.

The address of the memory location or an I/O device is sent out by the microprocessor. The corresponding **memory** chip or I/O device is selected by a decoding circuit. The **interfacing** process includes matching the **memory** requirements with the microprocessor signals.

When we are executing any instruction, we need the microprocessor to access the **memory** for reading instruction codes and the data stored in the **memory**. For this, both the memory and the microprocessor requires some signals to read from and write to registers. The interfacing circuit should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

8085 Interfacing Pins

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

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

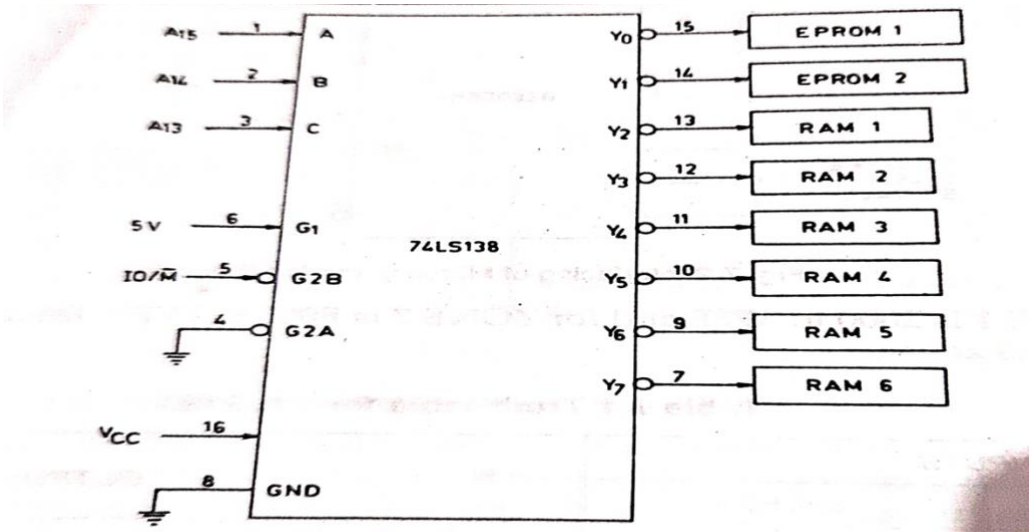


Fig. 7.3 Interfacing of Memory Chips using 74LS138.

4.4:Concept of Address decoding for I/O devices.

Address decoding refers to the way a computer system **decodes** the **addresses** on the **address** bus to select memory locations in one or more memory or peripheral devices. ... In full **address decoding**, each addressable memory location corresponds to a unique **address** value on the **address** bus.

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

The transfer of data between keyboard and microprocessor, and microprocessor and display device is called Input Output Interfacing 8085 Microprocessor or I/O data transfer. This data transfer is done with the help of I/O ports.

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

- Serial Communication Interface
- Parallel Communication interface

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

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

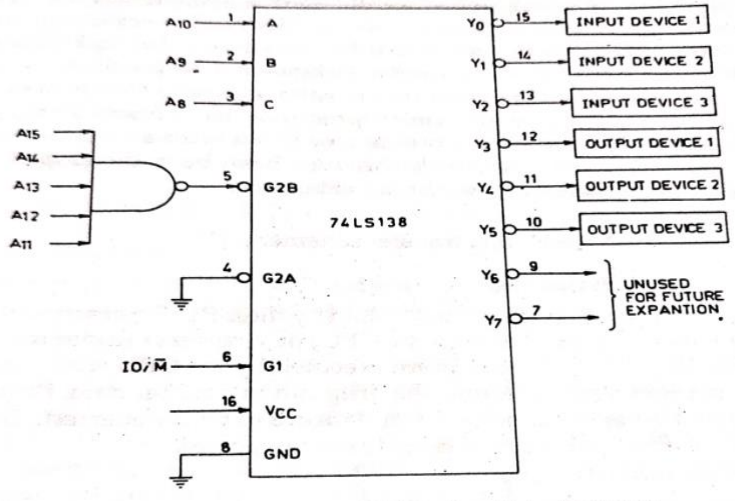


Fig. 7.4 Interfacing of I/O Devices Using 74LS138.

Table 7.2 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

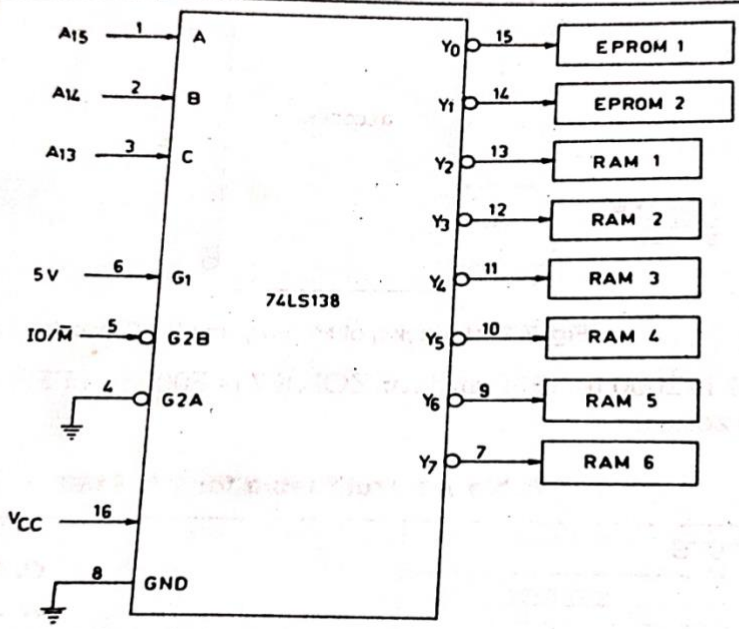


Fig. 7.3 Interfacing of Memory Chips using 74LS138.

4.5: Programmable Peripheral Interface: 8255.

Programmable Peripheral Interface (PPI) (Intel 8255):

A Programmable Peripheral Interface (PPI) is a multiport device. The main function are to interface peripheral devices to the microcomputer. It has three 8-bit port namely Port-A, Port-B, Port-C. Port-C has been divided into two 4-bit ports namely Port-Cu (upper) and Port-Cl (lower). Total two 8-bit port and two 4-bit ports are available and it can be programmed either as an input port or output port.

Architecture of Intel-8255:

7.7.2 Architecture of Intel 8255A

Fig. 7.14 shows the pin diagram of Intel 8255 A. It is a 40 pin I.C. Package. It operates on a single 5 V_{dc} supply. Its important characteristics are as follows:

Ambient temperature 0 to 70°C.

Voltage on any pin : 0.5 V to 7 V.

Power dissipation 1 Watt.

V_{IL} = Input low voltage = Minimum 0.5 V, Maximum 0.8 V.

V_{IH} = Input high voltage = Minimum 2 V, Maximum V_{CC}.

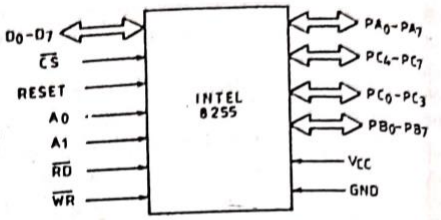


Fig. 7.14 Schematic Diagram of Intel 8255 A.

V_{OL} = Output low voltage = 0.45 V.

V_{OH} = Output high voltage = 2.4 V.

I_{DR} = Darlington drive current = Minimum 1 mA, Maximum 4 mA of any 8 pins of the port.

The pins for various ports are as follows :

$PA_0 - PA_7$ 8 pins of port A

$PB_0 - PB_7$ 8 pins of port B

$PC_0 - PC_3$ 4 pins of port C_{lower}

$PC_4 - PC_7$ 4 pins of port C_{upper}

The important control signals are as follows:

$\overline{CS}$ (Chip Select). It is a chip select signal. The LOW status of this signal enables communication between the CPU and 8255.

$\overline{RD}$ (Read). When $\overline{RD}$ goes LOW the 8255 sends out data or status information to the CPU on the data bus. In other words it allows the CPU to read data from the input port of 8255.

$\overline{WR}$ (Write). When $\overline{WR}$ goes LOW the CPU writes data or control word into 8255. The CPU writes data into the output port of 8255 and the control word into the control word register.

A_0 and A_1. The selection of input port and control word register is done using A_0 and A_1 in conjunction with $\overline{RD}$ and $\overline{WR}$. A_0 and A_1 are normally connected to the least significant bits of the address bus. If two 8255 units are used the address of ports are as follows:

For the 1st unit of 8255, i.e. 8255.1:

Port/Control word register	*Port/Control word register Address*
Port A	00
Port B	01
Port C	02
Control word register	03

For the 2nd unit of 8255, *i.e.* 8255.2:

Port/Control Word Register	*Port/Control Word Register Address*
Port A	08
Port B	09
Port C	0A
Control word register	0B

If we write the instruction IN 00, it means that it is for the Port A of 8255.1. When this instruction is executed data are transferred from the Port A to the accumulator. The instruction OUT 03 will transfer the content of the accumulator to the control word register of 8255.1. The instruction IN 09 transfers the data from Port B of 8255.2 to the accumulator. OUT 0A transfers the content of the accumulator to the Port C of 8255.2. The instruction OUT 0B transfers the content of the accumulator to the control word register of 8255.2. The IN instruction is used for the port which has been defined as input port. After IN the address of the port is specified. Similarly, OUT instruction is used for the ports which have been defined as output ports. After OUT instruction the address of the port is specified. For control word register only OUT instruction is used. Its content can not be read. How ports are defined as input or output ports has been discussed in section 7.7.4.

Intel 8255 is a 40 pin I.C package and operates in +5V regulated power supply.

- **PA0 – PA7 –** Pins of port A
- **PB0 – PB7 –** Pins of port B
- **PC0 – PC7 –** Pins of port C
- **D0 – D7 –** Data pins for the transfer of data
- **RESET –** Reset input
- **RD' –** Read input

- **WR' –** Write input
- **CS' –** Chip select
- **A1 and A0 –** Address pins

Operating Modes of 8255:

8255A has three different operating modes −

- **Mode 0 Simple Input/output** − In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
- **Mode 1 Strobed Input/output** − In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.
- **Mode 2 Bidirectional Port**− In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B

Control Word For 8255:

A control word is formed which contains the information regarding the function and modes of the ports. The CPU outputs the control word to 8255. According to the requirement a port can be programmed to act either Input port or output port. For programming the ports of 8255 a control word is formed. Control word is written into the control word register which is within 8255.

Bit no 0- It is for Port C-lower To make Input it is set to 1
To make Output it is set to 0

Bit no 1- It is Port B To make Input it is set to 1
To make Input it is set to 1

Bit no 2- It is Mode of Port B For Mode 0 it is set to 0
For Mode 1 it is set to 1

Bit no 3- It is for Port C-upper To make Input it is set to 1
To make Output it is set to 0

Bit no 4- It is for Port A To make Input it is set to 1
To make Output it is set to 0

Bit no 5,6- It is Mode of Port A

Mode of Port A	Bit no 5	Bit no 6
Mode-0	0	0
Mode-1	0	1
Mode-2	1	0 or 1

Bit no 7- It is set to 1, If Port A,B,C are defined as input/output port.
It is set to 0, If individual pin of port C are to be set or reset.

7	6	5	4	3	2	1	0

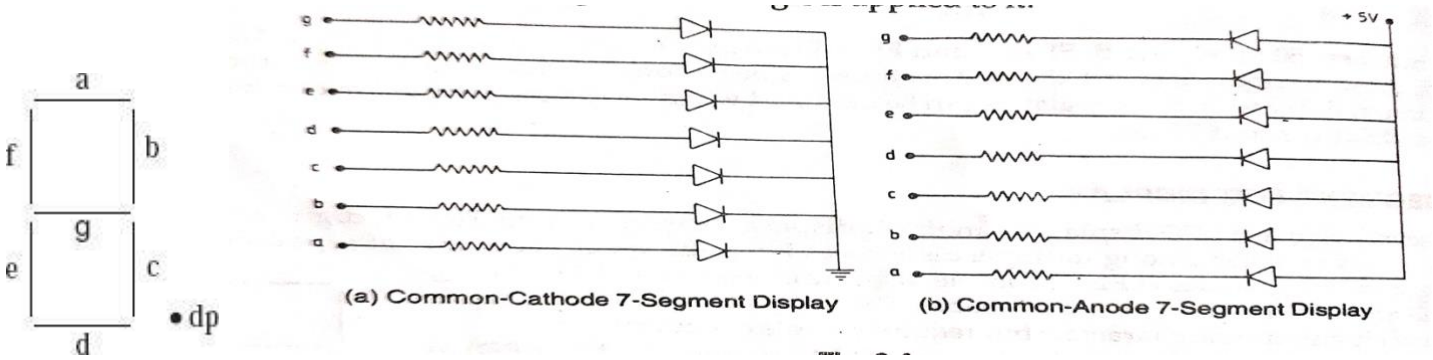
Bit no of control word.

4.6: ADC & DAC with Interfacing.

4.7: Interfacing Seven Segment Displays

The **7-segment LED display** is a multiple display. It can display 0-9 numeric, alphabetic. Each **LED** has a negative leg that is connected to one of the pins of the device. Each LED can be controlled separately. To display a digit or letter the desired segments are made ON.

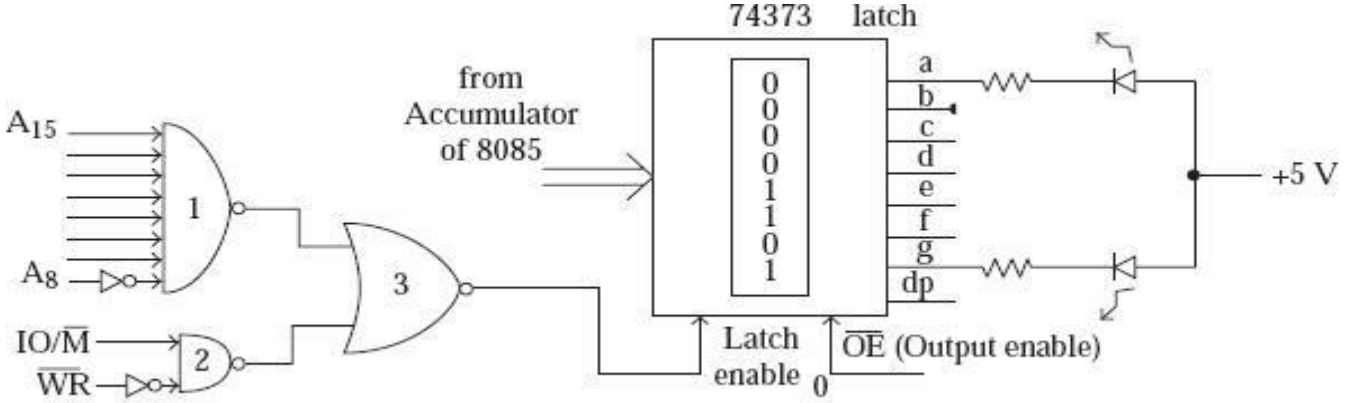
An output device which is very common is, especially in the kit of 8085 microprocessor and it is the Light Emitting Diode consisting of seven segments. We denote the segments as a, b, c, d, e, f, g. Moreover, these are LEDs or together a series of Light Emitting Diodes. a 7-segment display is as shown in the following Fig.



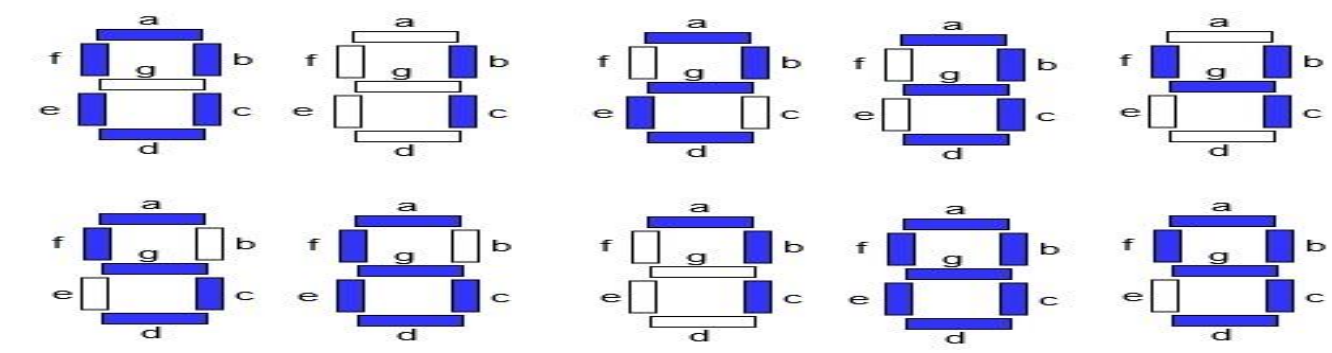
There are two types of 7-segment LED: They are the **common anode type** and the **common cathode type**. In common anode type display all the 7 anodes of LEDs are tied together to the ground. When +5v dc applied to any segment, the corresponding diode emits light. In common cathode type display all the 7 anodes of LEDs are tied together to the ground. When +5v dc applied to any segment, the corresponding diode emits light.

The 7-segment displays are not connected to I/O port directly. They are connected through drivers/decoders.

The use of 74373 latch for interfacing a 7-segment display is shown in the following Fig.



Note: To keep the figure simple, only two of the eight segment connections are shown. The other six segment connections are similar.



In the 74373 latch is used as an I/O mapped I/O port with the port address as FE H. This could be easily verified from the chip select circuit used in the figure. The following instructions are to be executed to display character '3' on the 7-segment display. The corresponding program to send 0D H to the port FE H will be -

MVI A, 0D H

OUT FE H

Using MVI instruction we are initializing Accumulator (A) with Byte 0D H i.e. 0000 1101. Then it will be sent to the port FE H by the instruction OUT.

Applications of Seven Segment Displays

Common applications of seven segment displays are in:

- Digital clocks
- Calculators
- Wrist watchers
- Speedometers
- Motor-vehicle odometers
- Radio frequency indicators
- Microwave or fancy toaster ovens

4.8: Generate square waves on all lines of 8255.

In this section you will see the assembly language code to generate Square wave using 8085 microprocessor:

At first we assume that, CWR address of 8255 is 0B and SOC pin of 0808 is connected to 0th pin of PORT B

```
MVI A,89H
OUT 0BH
MVI A,01H
OUT 09H

BACK: MVI A,FFH
OUT 08H
MVI C,BOH
LOOP1: DCR C
JNZ  LOOP1
MVI A,00H
OUT 08H
MVI C,B0H

LOOP2: DCR C
JNZ LOOP2
JMP BACK
```

4.9: Design Interface a traffic light control system using 8255.

For microprocessor based **traffic light control**, all ports of 8255 have been programmed as output ports. The control word to make all ports output ports in Mode 0 operation is 80H. The connection of pins of the ports to LED have been made through buffers (7407). Positive logic has been used to switch on LEDs. Three types of LEDs have been used **Red, Yellow and Green**. Green light glows to allow crossing, yellow to make alert and red does not allow crossing.

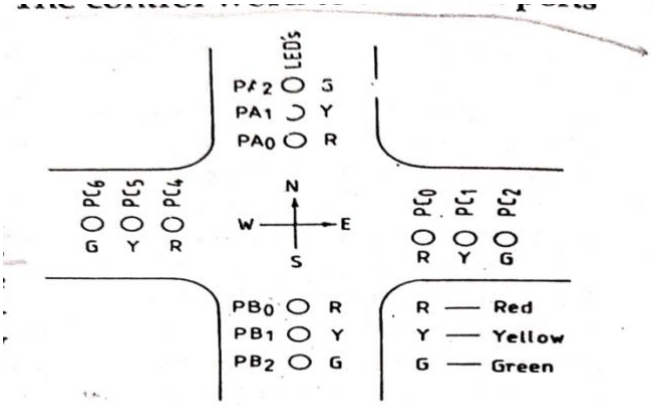


Fig. 9.54. Traffic Light Control.

Memory Address	Machine codes	Labels	Opcode	Operands	Comments
4100	3E,80		MVI	A,80H	Get Control word for 8255.
4102	D3,0B		OUT	0B	Initialize Port 8255.
4104	3E,01	LOOP	MVI	A,01	
4106	D3,09		OUT	09	**RED ON** for South.
4108	D3,08		OUT	08	**RED ON** for North.
410A	3E,44		MVI	A,44	**GREEN ON** for east and west.
410C	D3,04		OUT	0A	
410E	CD,00,42		CALL	DELAY I	
4111	3E,22		MVI	A,22	**YELLOW ON** for east and west.
4113	D3,0A		OUT	0A	
4115	3E,02		MVI	A,02	
4117	D3,09		OUT	09	**YELLOW ON** for South.
4119	D3,08		OUT	08	**YELLOW ON** for North.
411B	CD,13,42		CALL	DEPLAY II	
411E	3E,11		MVI	A,11	
4120	D3,0A		OUT	0A	**RED ON** for east and west.
4122	3E,04		MVI	A,04	
4124	D3,08		OUT	08	**GREEN ON** for north.
4126	D3,09		OUT	09	**GREEN ON** for south.
4128	CD,00,42		CALL	DELAY I	
412B	3E,22		MVI	A,22	**YELLOW ON** for east and west.
412D	D3,0A		OUT	0A	
412F	3E,02		MVI	A,02	
4131	D3,09		OUT	09	**YELLOW ON** for South.
4133	D3,08		OUT	08	**YELLOW ON** for North.
4135	CD,13,42		CALL	DEPLAY II	
4138	C3,04,41		JMP	LOOP	

DELAY I

4200	06,20		MVI	B,20H	
4202	0E,FF	G03	MVI	C,FF	
4204	16,FF	G02	MVI	D,FF	
4206	15	G01	DCR	D	
4207	C2,06,42		JNZ	G01	
420A	0D		DCR	C	
420B	C2,04,42		JNZ	G02	
420E	05		DCR	B	
420F	C2,02,42		JNZ	G03	

4212	C9	RET	
DELAY II			
4213	06,10	MVI B,10	
4215	C3,02,42	JMP 4202	G03

4.10: Design interface for stepper motor control using 8255.

A stepper motor rotates in steps in response to digital pulse inputs. The shaft of the motor rotates in equal to increments when a train of input pulses is applied. To control the direction and number of steps appropriate pulses are applied to the starter windings of the motor. There are two types of stepper motor **a.Permanent Magnet Type b. Variable reluctance Type.**

The Permanent magnet type stepper motor consists of four pole stator and a rotor with six permanent poles. The stator windings are energised by pulses. The motor has four phase excitation as there are four poles on the stator. Each pole has two coil wound in the opposite sense so that the pole can be made either a north pole or a south pole.

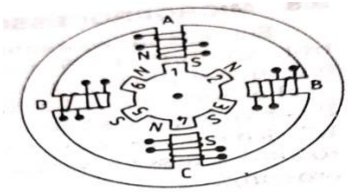


Fig. 9.51. Schematic Diagram of Stepper Motor.

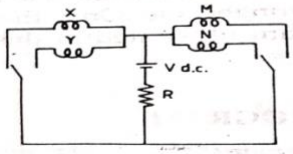


Fig. 9.52. Bifiller Pole Winding

Here X and y are two coils on the same pole. M and n are also the coils on the same pole situated at diametrically opposite position. Resistance R is used to reduce the electrical inertia of the highly inductive windings. The motor is known as bifiller wound stepper motor.

If the pole A is made north pole, pole C is made south pole. The permanent south pole no 1 of the rotor will stand just below pole A of the stator. To give a clockwise motion the supply of the pole A and C is switched off and the pole B and D are energised. The pole B is made south pole and D is north pole.

Now the permanent north pole no 2 of the rotor comes just below the pole B. In the next step pole C is made an N-pole and A is S-pole. After this D is made S-pole and B is N-pole. Again pole A is N-pole and C is S-pole and the whole sequence is repeated. In this process poles are energised to give a clockwise rotation.

To rotate the rotor anticlockwise making A pole an N-pole and C pole a S-pole, D is made S-pole and B is N-pole.

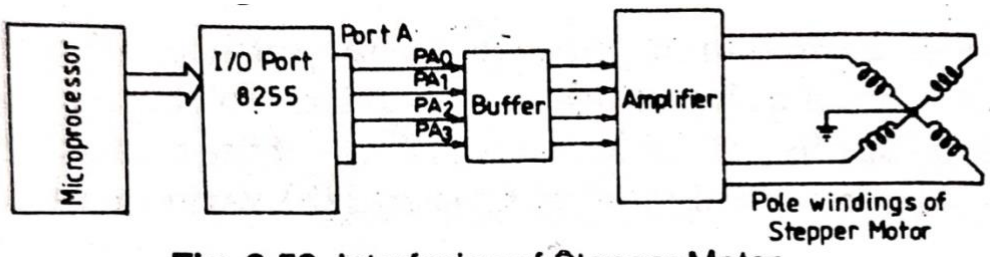


Fig. 9.53. Interfacing of Stepper Motor.

For interfacing connections of stepper motor, 12 V dc supply is used to energise the poles. Pulses sent by the microprocessor switch on rated voltage to the windings of the desired poles. After energising one set of the

pole windings some delay is provided, then power supply is switched on to the other set of pole windings. This delay time governs the speed of the motor.

4.11:Basic concept of other Interfacing DMA controller, USART.

DMA Controller:

The bulk data transfer from I/O devices to the memory or from the memory to I/O devices through the accumulator is a time consuming process. For such a situation **Direct Memory Access (DMA)** technique is used. In DMA data transfer scheme, data are directly transferred from an I/O devices to the memory and vice versa.

Intel 8257 is a Programmable DMA Controller. It is 40 pin IC package and requires 5v supply to operate. Four I/O devices can be interfaced to the microprocessor through this device.

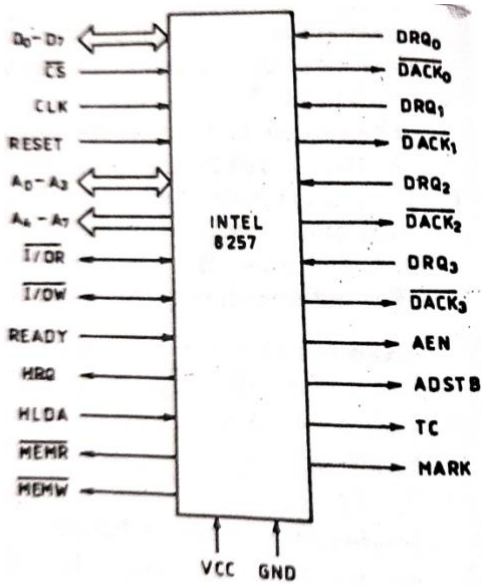


Fig. 7.24 Schematic Diagram of Intel 8257.

It is capable of performing three operations Read, Write and Verify. During the Read operation the data are directly transferred from the memory to the I/O device. During Write operation the data are directly transferred from the memory to the I/O device. On receiving a request from the I/O device, 8257 generates a sequential memory address which allow I/O device to Read or Write directly to or from the memory.

USART: (Programmable Communication Interface(PCI) Intel 8251)

The Intel 8251 is a Programmable Communication Interface. It is Universal Synchronous/ Asynchronous Receiver/ Transmitter **(USART)**. It is compatible with 8085,8086 etc. The 8251 can be used to transmit/ receive serial data. It accepts data in parallel format from the microprocessor and converts them into serial data for transmission.

Govt. Polytechnic, Bhubaneswar

(Summer-2021)

Unit-5: Microprocessor (Architecture and Programming-8086-16 bit)

Faculty: Kanak Prava Swain, Lect.(ETC).

Subject: Microprocessor & Microcontroller.

Semester: 4th Sem. ETC/IT.

5.1: Register Organisation of 8086.

Intel 8086 contains the following registers

a. General Purpose Registers.
b. Pointer and Index Registers.
c. Segment Registers.
d. Instruction Pointer.
e. Status Flags.

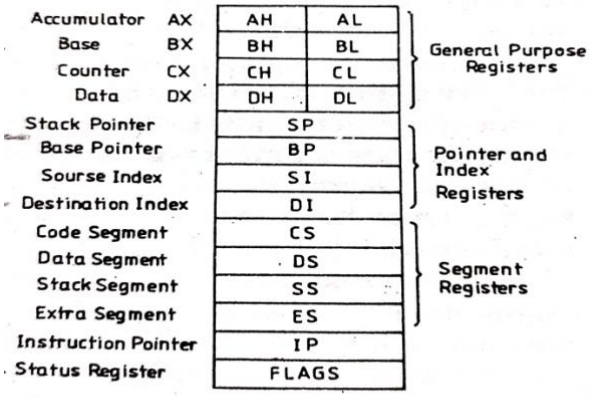


Fig. 11.3 Register organization of Intel 8086

5.2: Internal Architecture of 8086.

Functional Unit or Block Diagram of Intel 8086:

Intel 8086 is a 16 bit N channel HMOS microprocessor. HMOS is used for High Speed MOS. It is a 40 pin IC package and it is Dual In-line Package (DIP). It has 20 address lines, it can directly address up to 2 to the power 20 =**1 MB** of memory. The 16 low order address lines are multiplexed with data Bus and 4 high order address lines are time multiplexed with status signals. 8 LSB of data are transmitted on AD0-AD7 and 8 MSB of data on AD8-AD15.

8086 contains two independent functional units: **a Bus Interface Unit (BIU)** and an **Execution Unit (EU)**.

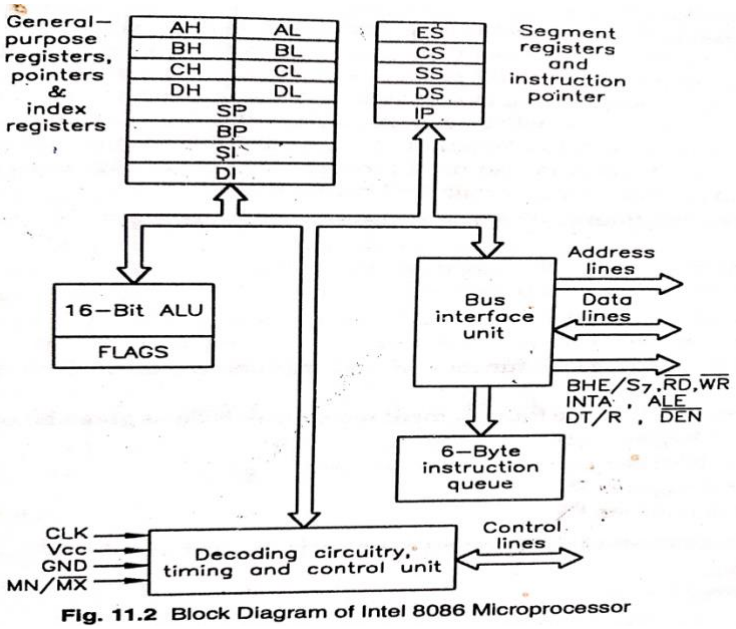


Fig. 11.2 Block Diagram of Intel 8086 Microprocessor

Bus Interface Unit (BIU)
The segment registers, instruction pointer and 6-byte instruction queue are associated with the **Bus Interface Unit (BIU)**

The Bus Interface Unit (BIU) :

- Handles transfer of data and addresses between processor, memory/IO devices,
- Fetches instruction codes, stores fetched instruction codes in first-in-first-out register set called a **queue**,
- Reads data from memory and I/O devices,
- Writes data to memory and I/O devices,
- It relocates addresses of operands since it gets un-relocated operand addresses from EU. The EU tells the BIU from where to fetch instructions or where to read data.

Execution Unit (EU):
The General purpose registers, Stack Pointer, Base Pointer, and Index Register, ALU, Flag Register, Instruction decoder, and timing & control unit constitute **Execution Unit (EU).**

- The **EU** receives opcode of an instruction from the queue, decodes it and then executes it. While Execution, unit decodes or executes an instruction, then the BIU fetches instruction codes from the memory and stores them in the queue.
- The **BIU and EU** operate in parallel independently. While **EU** executes instructions, the **BIU** fetches instructions. This type of overlapped operation of the functional unit of a microprocessor is called **Pipelining**. This makes processing faster.

General Purpose Registers: There are four 16-bit general purpose registers:

- AX
- BX
- CX
- DX.

Each of these 16-bit registers are further subdivided into 8-bit registers It has the following functional parts:

16 bit Registers	8-bit High order Registers	8-bit Low order Registers
AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

The Register AX serves as an accumulator, BX,CX,DX are used as General purpose register.

Pointer and Index Register:

- **Instruction Pointer (IP):** The instruction pointer in the 8086 microprocessor acts as a **program counter**. It indicates to the address of the next instruction to be executed.

- **Index Register:** The following four registers are in the group of pointer and index registers:
    - Stack Pointer (SP)
    - Base Pointer (BP)
    - Source Index (SI)
    - Destination Index (DI)

- **Instruction Queue:** When EU executes instructions, the BIU gets 6-bytes of the next instruction and stores them in the instruction queue and this process is known as instruction pre fetch. This process increases the speed of the processor.

- **Segment Registers:** A segment register contains the addresses of instructions and data in memory which are used by the processor to access memory locations. It points to the starting address of a memory segment currently being used.
  There are 4 segment registers in 8086 as given below:
    - **Code Segment Register (CS):** Code segment of the memory holds instruction codes of a program.

- **Data Segment Register (DS):** The data, variables and constants given in the program are held in the data segment of the memory.
- **Stack Segment Register (SS):** Stack segment holds addresses and data of subroutines. It also holds the contents of registers and memory locations given in PUSH instruction.
- **Extra Segment Register (ES):** Extra segment holds the destination addresses of some data of certain string instructions.

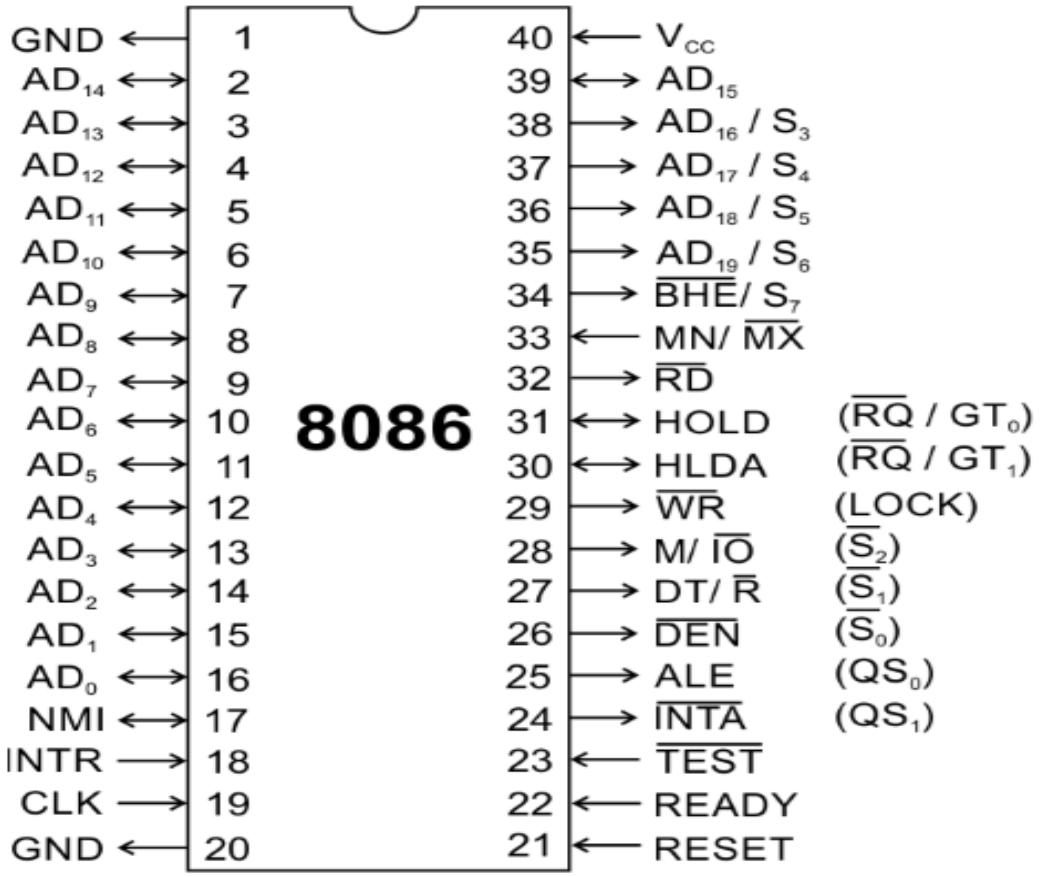
- **ALU:** It handles all arithmetic and logical operations. Such as addition, subtraction, multiplication, division, AND, OR, NOT operations.
- **Flag Register:** It is a 16-bit register which exactly behaves like a flip-flop, means it changes states according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups i.e. conditional and control flags.
- **Conditional Flags:** This flag represents the result of the last arithmetic or logical instruction executed. Conditional flags are:
    - Carry Flag
    - Auxiliary Flag
    - Parity Flag
    - Zero Flag
    - Sign Flag
    - Overflow Flag
- **Control Flags:** It controls the operations of the execution unit. Control flags are:
    - Trap Flag
    - Interrupt Flag
    - Direction Flag

5.2: Signal Description of Intel 8086.

Pin description for Intel 8086 Microprocessor:

Intel **8086** is a 16-bit HMOS **microprocessor**. It is available in 40 **pin** DIP chip. It uses a 5V DC supply for its operation. The **8086** uses 20-line address bus.

The 8086 uses 20-line address bus. It has a 16-line data bus. The 20 lines of the address bus operate in multiplexed mode. The 16-low order address bus lines have been multiplexed with data and 4 high-order address bus lines have been multiplexed with status signals.



AD0 to AD15: These lines are multiplexed Address / data lines. When AD lines are used to transmit memory address A0-A15 is used and when data are transmitted over AD lines D-D15 is used.

A16-A19 High order address lines These are multiplexed with status signals.

A16/S3, A17/S4 –A16 and A17 multiplexed with S3 and S4 respectively and S4 and S3 are used to select the segment out of the four segments.

A18/S5- A18 multiplexed with S5 and it is used as interrupt flag.

A19/S6- A18 multiplexed with S6 and S6 is used as bus master, which handles the internal bus control.

BHE' / S7: BHE stands for Bus High Enable. It is an active low signal, i.e. it is active when it is low. It is used to indicate the transfer of data over the higher order data bus (D8 to D15). BHE' decides whether the data bus will carry 16-bit data or 8-bit data. When BHE' is enabled (i.e. 0), then the bus will carry 16-bit data, else only 8-bit data through the lower order data bus lines. It is multiplexed with status pin S7.

RD': It is a read signal used for read operation. It is also an active low signal.

READY: This is an acknowledgment signal from the slower I/O devices or memory. When high, it indicates that the device is ready to transfer data, else the microprocessor is in the wait state.

RESET: By using this pin, the program control returns to $FFFF0_H$. The signal is active High.

INTR: This pin is used to receive an interrupt request signal. It is a type of maskable interrupt.

NMI: This is used for Non-Maskable Interrupt Request.

GND: There are two ground pins in the 8086, pin 1 and pin 20.

VCC: The pin 40 is for voltage input.

TEST': This is also an active low signal. This pin is used for wait instruction when the 8086 is connected with the 8087 microprocessor.

CLK: This pin tells about the clock pulse.