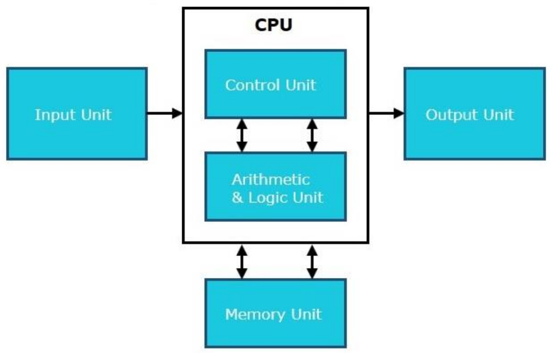
Department of Computer Engineering

Semester: 5th

Computer Organization and Architecture(4350701)

IMP Question with Answer

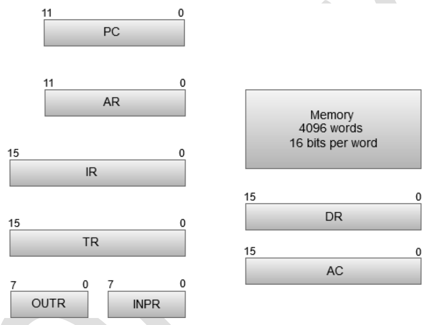
1) Explain Basic CPU Structure.



- A computer consists of five main components: Input unit, Central Processing Unit, Memory unit Arithmetic & logical unit, Control unit and an Output unit.
- ALU Control unit Input Unit Output Unit Memory Unit CPU (eg:-Mouse, Keyboard) (eg:-Monitor, Printer)
- Input unit
  - Input units are used by the computer to read the data.
  - The most commonly used input devices are keyboards, mouse, joysticks, trackballs, microphones, etc.
- Central processing unit
  - Central processing unit commonly known as CPU can be referred to as an electronic circuitry within a computer that carries out the instructions given by a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.
- Memory unit
  - The Memory unit can be referred to as the storage area in which programs are kept which are running, and that contains data needed by the running programs.
  - The Memory unit can be categorized in two ways namely, primary memory and secondary memory.
  - It enables a processor to access running execution applications and services that are temporarily stored in a specific memory location.
- Arithmetic & logical unit
  - Most of all the arithmetic and logical operations of a computer are executed in the ALU (Arithmetic and Logical Unit) of the processor.
  - It performs arithmetic operations like addition, subtraction, multiplication, division and also the logical operations like AND, OR, NOT operations.

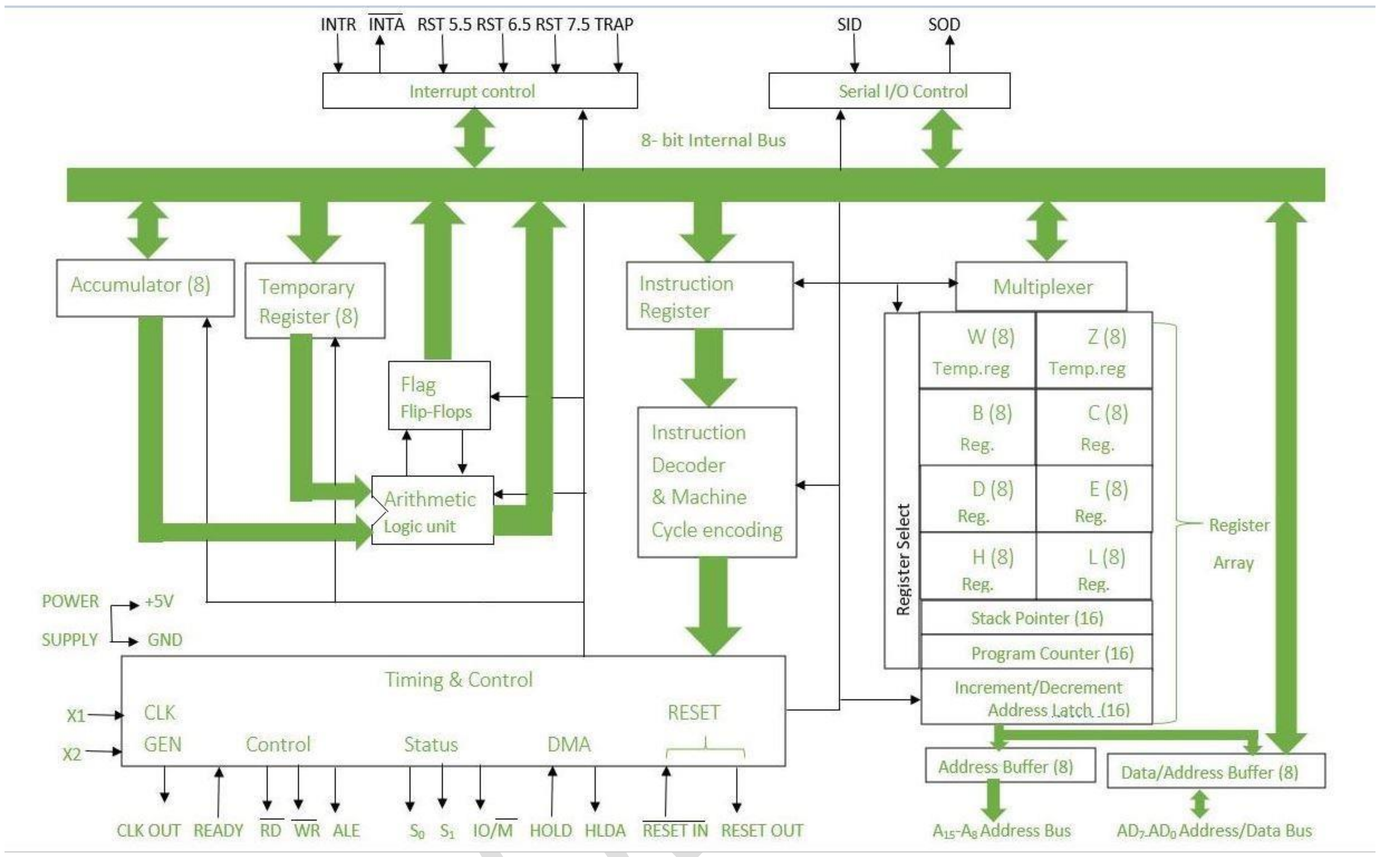
- Control unit
  - ○ The control unit is a component of a computer's central processing unit that coordinates the operation of the processor. It tells the computer's memory, arithmetic/logic unit and input and output devices how to respond to a program's instructions.
  - ○ The control unit is also known as the nerve center of a computer system.
- Output Unit
  - ○ The primary function of the output unit is to send the processed results to the user. Output devices display information in a way that the user can understand.
  - ○ Output devices are pieces of equipment that are used to generate information or any other response processed by the computer. These devices display information that has been held or generated within a computer.
  - ○ The most common example of an output device is a monitor.

2) Explain Various Registers used in CPU with its applications.



Register	Symbol	Number of Bits	Function
Accumulator	AC	16	It's a processor register.
Program counter	PC	12	It stores the address of the instruction.
Address Register	AR	12	It is used for storing memory addresses.
Data Register	DR	16	It is a general-purpose register used for storing data during calculations.
Instruction Register	IR	16	It stores the current instruction being executed.
Temporary Register	TR	16	It holds the temporary data.
Input Register	INPR	8	It carries the input character.
Output Register	OUTR	8	It carries the output ch

3) Draw and explain the internal architecture of 8085.



- The architecture of 8085 Microprocessor is shown in figure given below.
- The internal architecture of 8085 includes following section
  - ALU-Arithmetic and Logic unit
  - Timing and control unit
  - Instruction register and decoder,Register array,
  - Interrupt control and serial I/O control.

ALU:-

- The ALU performs the arithmetic and logical operations.
- The operations performed by ALU of 8085 are addition, subtraction, increment, decrement, logical AND, OR, EXCLUSIVE -OR, compare, complementand left / right shift.
- The accumulator and temporary register are used to hold the data during anarithmetic / logical operation. After an operation the result is stored in the accumulator and the flags are set or reset according to the result of the operation.

TIMING & CONTROL UNIT:

- The timing and control unit synchronizes all the microprocessor operations with the clock and generatesthe control signals necessary for communication between the microprocessor and peripherals.

INSTRUCTION REGISTER & DECODER:

- When an instruction is fetched from memory it is placed in instruction register. Then it is decoded and encoded into various machine cycles.

REGISTER ARRAY:

- Apart from Accumulator (A-register), there are six general-purpose programmable registers B, C, D, E, Hand L.
- They can be used as 8-bit registers or paired to store l6-bit data.
- The allowed pairs are B-C, D-Eand H-L.
- The temporary registers W and Z are intended for internal use of the processor and it cannot be used by the programmer.

STACK POINTER (SP):

- The stack pointer SP, holds the address of the stack top.
- The stack is a sequence of RAM memory locations defined by the programmer.
- The stack is used to save the content of registers during the execution of a program.

PROGRAMCOUNTER (PC):

- The program counter (PC) keeps track of program execution.
- To execute a program the starting address of the program is loaded in program counter. The PC sends out an address to fetch a byte of instruction from memory and increment its content automatically.
- Hence, when a byte of instruction is fetched, the PC holds the address of the next byte of the instruction or next instruction.

INTERRUPT CONTROL:-

- interrupt control provide control for following interrupt INT, INTA, RST5.5,RST6.5, RST7.5, TRAP

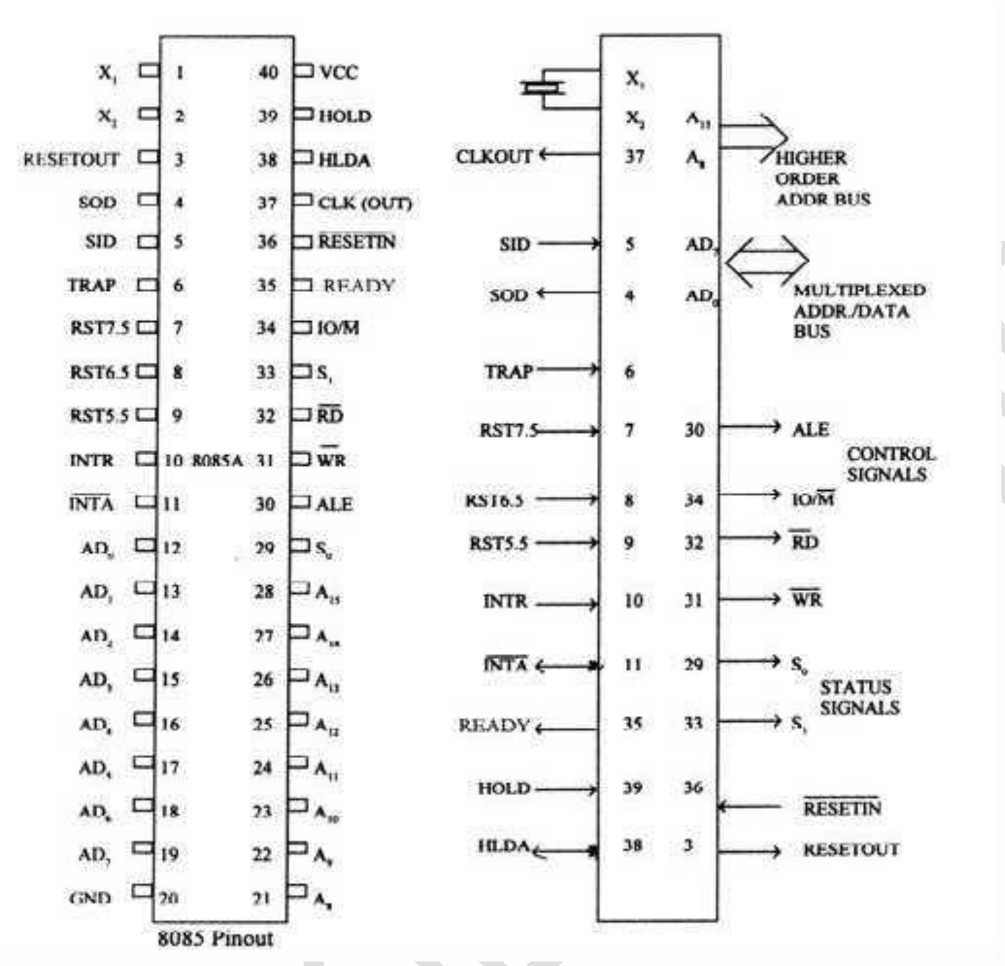
SERIAL I/O CONTROL:

- IT provide two pin for serial communication SOD-Serial output data and SID-Serial input data

4) Draw the PIN DIAGRAM of 8085 & explain the detail of each pin OR Draw & Explain 8085 Pin out signals.

X1 & X2

- These are also called Crystal Input Pins.
- 8085 can generate clock signals internally.
- To generate clock signals internally, 8085 requires external inputs from X1 and X2.



RESET IN:

- ❖ It is used to reset the microprocessor. It is active low signal. When the signal on this pin is low (for atleast 3 clocking cycles), it forces the microprocessor to reset itself.
- ❖ Resetting the microprocessor means:Clearing the PC and IR.
    - ➢ Disabling all interrupts (except TRAP).
    - ➢ Disabling the SOD pin.
    - ➢ All the buses (data, address, control) are tristated.
    - ➢ Gives HIGH output to RESET OUT pin.

RESET OUT:

- ❖ It is used to reset the peripheral devices and other ICs on the circuit.
- ❖ It is an output signal.
- ❖ It is an active high signal.
- ❖ The output on this pin goes high whenever RESET IN is given low signal.
- ❖ The output remains high as long as RESET IN is kept low

SID (Serial Input Data):

- ❖ It takes 1 bit input from serial port of 8085 and Stores the bit at the 8th position (MSB) of the Accumulator.
- ❖ RIM (Read Interrupt Mask) instruction is used to transfer the bit.

SOD (Serial Output Data):

- It takes 1 bit from the 8th position (MSB) of the Accumulator to serial port of 8085.
- SIM (Set Interrupt Mask) instruction is used to transfer the bit.

<u>Interrupt Pin:</u>

TRAP:-

- It is a non-maskable interrupt.
- It has the highest priority.
- It cannot be disabled.
- It is both edge and level triggered.
- It means TRAP signal must go from low to high and must remain high for a
- certain period of time.
- TRAP is usually used for power failure and emergency shutoff.

RST 7.5:-

- It is a maskable interrupt.
- It has the second highest priority.
- It is positive edge triggered only.
- The internalflip-flop is triggered by the rising edge.
- The flip-flop remains high until it is cleared by RESET IN.

RST 6.5:-

- It is a maskable interrupt.
- It has the third highest priority.
- It is level triggered only.
- The pin has to be held high for a specific period of time.RST 6.5 can be enabled by EI instruction.
- It can be disabled by DI instruction.

RST 5.5:-

- It is a maskable interrupt.
- It has the fourth highest priority.
- It is also level triggered.
- The pin has to be heldhigh for a specific period of time. This interrupt is very similar to RST 6.5.

INTR:-

- It is a maskable interrupt.
- It has the lowest priority.
- It is also level triggered.
- It is a general purpose interrupt.
- By general purpose we mean that it can be used to vector microprocessor to any specific subroutine having any address.

INTA:

- It stands for interrupt acknowledge.
- It is an outgoing signal.
- It is an active low signal.
- Low output on this pin indicates that microprocessor has acknowledged the INTR request.

<u>Address and Data Pins</u>

AD0 – AD7:- (Bidirectional)

- These pins have dual purpose of transmitting lower order address and data byte.
- During 1st clock cycle, these pins act as lower half of address.
- In remaining clock cycles, these pins act as data bus.

- The separation of lower order address and data is done by address latch.

A8 – A15:-(Unidirectional)

- These pins carry the higher order of address bus. The address is sent from microprocessor to memory.

ALE:- (Output)

- It is used to enable Address Latch. It indicates whether bus functions as address bus or data bus.
- If ALE = 1 then Bus functions as address bus.
- If ALE = 0 then Bus functions as data bus.

S0 and S1 :-(output)

S0 and S1 are called Status Pins. They tell the current operation which is in progress in 8085.

S0	S1	Operation
0	0	Halt
0	1	Write
1	0	Read
1	1	Opecode

IO/M:-

- This pin tells whether I/O or memory operation is being performed.
- If IO/M = 1 then I/O operation is being performed.
- If IO/M = 0 then Memory operation is being performed.

RD:-

- RD stands for Read. It is an active low signal.
- It is a control signal used for Read operation either from memory or from Input device.
- A low signal indicates that data on the data bus must be placed either from selected memory location or from input device.

WR:-

- WR stands for Write. It is also active low signal.
- It is a control signal used for Write operation either into memory or into output device.
- A low signal indicates that data on the data bus must be written into selected memory location or into output device.

READY:-

- This pin is used to synchronize slower peripheral devices with fast microprocessor

HOLD:-

- HOLD pin is used to request the microprocessor for DMA transfer.
- A high signal on this pin is a request to microprocessor to come off from the hold on buses.
- This request is sent by DMA controller.

HLDA:-

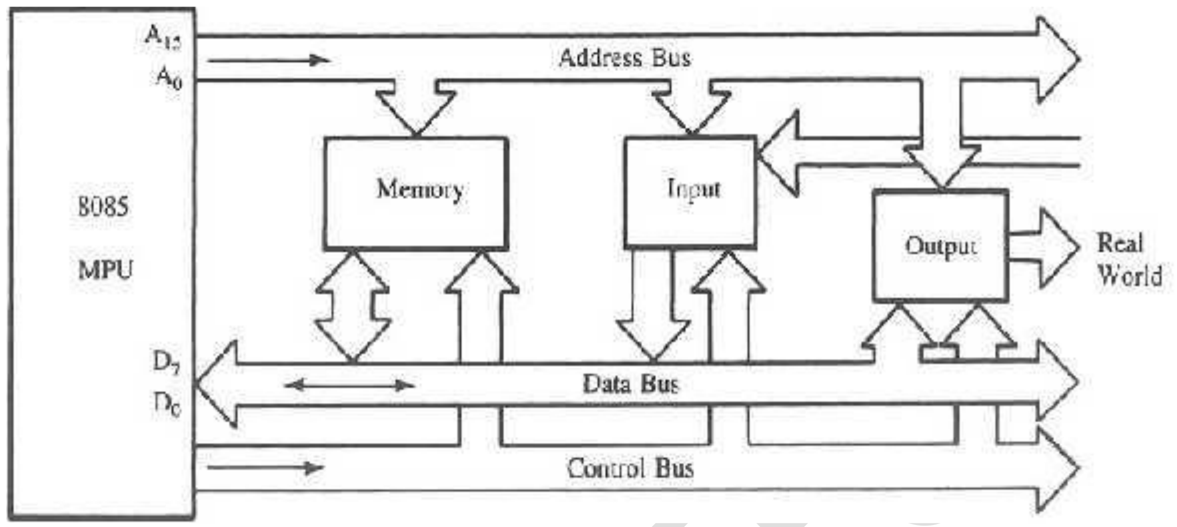
- HLDA stands for Hold Acknowledge.
- The microprocessor uses this pin to acknowledge the receipt of HOLD signal. When HLDA signal goes high, address bus, data bus, RD’,WR’,
- IO/M’ pins are tri-stated. This means they are cut-off from external environment.
- The control of these buses goes to DMAController.

VSS and VCC:-

- +5V power supply is connected to VCC. Ground signal is connected to VSS.

5) Discuss the bus organization of 8085.

The 8-bit 8085 CPU (or MPU – Micro Processing Unit) communicates with the other units using a 16-bit address bus, an 8-bit data bus and a control bus.



Address Bus

- It consists of 16 address lines: A0 – A_{15}
- It operates in **unidirectional** mode: The address bits are always sent from the MPU to peripheral devices in one direction, not reverse.
- MPU uses the address bus to perform first function : identifying a peripheral or a memory location.16 address lines are capable of addressing a Total of 2^{16}=65,536 (64k) memory locations.
- Address locations: 0000 (hex) to FFFF (hex)
- When the 8085 wants to access a peripheral or a memory location, it places the 16-bit address on the address bus and then sends the appropriate control signals.

Data Bus

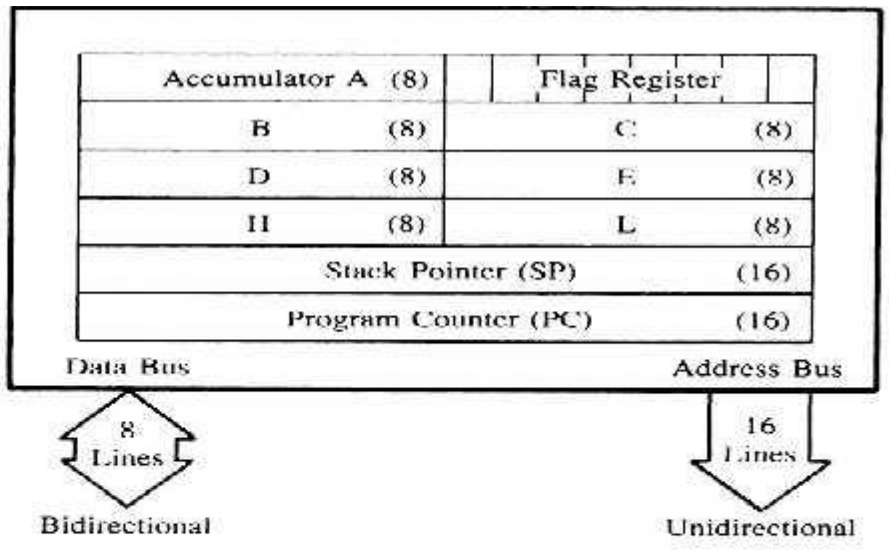
- It consists of 8 data lines: D0 – D7
- It operates in bidirectional mode: The data bits are sent from the MPU to peripheral devices, as
- well as from the peripheral devices to the MPU.
- The MPU uses the data bus to perform second function : Transfer binary information (data and
- instructions)
- Data range: 00 (hex) – FF (hex)

Control Bus

- It comprised of various single lines that carry synchronization signals.
- The MPU uses such lines to perform third function: Provide timing or synchronization signals (control signals)

6) Explain registers organization of 8085.

The 8085 can perform a number of internal operations. Such as: storing data, Arithmetic & Logic operations, testing for condition, etc. To perform these operations, the microprocessor needs an internal architecture similar to the following:



Registers
- ❖ Six general purpose 8-bit registers: B, C, D, E, H, L
- ❖ They can also be combined as register pairs to perform 16-bit operations: BC, DE, and HL
- ❖ Registers are programmable (data load, move, etc.)

Accumulator
- ❖ Single 8-bit register that is part of the ALU
- ❖ Used for arithmetic / logic operations – the result is always stored in the accumulator.

Flag Bits
- ❖ Indicate the result of condition tests.
- ❖ Carry, Zero, Sign, Parity, etc.
- ❖ Conditional operations (IF / THEN) are executed based on the condition of these flag bits.

The Program Counter (PC)
- ❖ This is a register that is used to control the sequencing of the execution of instructions.
- ❖ This register always holds the address of the next instruction.
- ❖ Since it holds an address, it must be 16 bits wide.

The Stack pointer
- ❖ The stack pointer is also a 16-bit register that is used to point into memory.
- ❖ The memory this register points to be a special area called the stack.
- ❖ The stack is an area of memory used to hold data that will be retrieved soon.
- ❖ The stack is usually accessed in a Last in First out (LIFO) fashion.

7) Define following terms: Instruction, Machine Cycle, Opcode, Oprand &Instruction Cycle.

Instruction:
- ❖ Instruction is the command given by the programmer to the Microprocessor to Perform the
- ❖ Specific task.
- ❖ For example, transfer a data, to do addition etc.

Machine Cycle:
- ❖ Machine cycle is the time required to transfer data to or from memory or I/O devices.
- ❖ Each read or writes operation constitutes a machine cycle.

- The instructions of 8085 require 1–5 machine cycles containing 3–6 clocks.
- The 1st machine cycle of any instruction is always an Op code fetching cycle in which the processor decides the nature of instruction.
- It is of at least 4-clocks. It may go up to 6-Clocks.

Instruction Cycle:

- An instruction cycle is defined as the time required for fetching and executing an instruction.
- For executing any program, basically 3-steps are followed sequentially that is Fetch, Decode and Execute.

Op code:

- Operation Perform by the microprocessor is called Op code.

Operand:

- The Data on which Microprocessor perform operation is called Operand.

8) Explain the classification of instructions of 8085 on the basis of their operation.

The 8085 instruction set can be classified into the following seven functional group.

1) Data Transfer Instructions
2) Arithmetic Instructions
3) Logical Instructions
4) Branching Instructions
5) Stack related instructions
6) Input/output instructions
7) Machine Control Instructions

Group I - DATA TRANSFER INSTRUCTIONS:

- These instructions move data between registers, or between memory and registers.
- These instructions copy data from source to destination.

Move Instruction	MOV Rd,Sd
	MOV R,M
	MOV M,R
	MVI R,data-8
	MVI M, data-8
	LXI Rp, data-16
Load and Store Instruction	LDA addr-16
	STA addr-16
	LDAX Rp
	STAX Rp
	LHLD addr-16
	SHLD addr-16
Exchange Instruction	XCHG

Group II - ARITHMETIC INSTRUCTIONS:

- These instructions perform the operations like: Addition, Subtract, Increment, and Decrement.

Addition Instruction	ADD R/M
	ADI DATA-8
	ADC R/M
	ACI data-8
	DAD Rp
	DAA
Subtraction Instruction	SUB R/M
	SUI data-8
	SBB R/M
	SBI data-8
Increment and Decrement	INR R/M
	INX Rp
	DCR R/M
	DCX Rp

Group III - LOGICAL INSTRUCTIONS:-

These instructions perform logical operations on data stored in registers, memory and status flags.

Logical AND Instruction	ANA R/M
	ANI data-8
Logical OR Instruction	ORA R/M
	ORI data-8
Logical Ex-OR Instruction	XRA R/M
	XRA data-8
Compare Instruction	CMP R/M
	CPI data-8
Rotate Instruction	RLC
	RAL
	RRC
	RAR
Complement Instruction	CMA
	CMC
	STC

Group IV - BRANCHING INSTRUCTIONS:

The branching instruction changes the normal sequential flow of the Program. These instructions alter either unconditionally or conditionally.

Jump Instruction	JMP addr-16
	JC addr-16/ JNC addr-16
	JZ addr-16/ JNZ addr-16
	JPE addr-16/ JPO addr-16
	JP addr-16/JM addr-16
Call and Return Instruction	CALL addr-16
	RET

Group V - STACK RELATED INSTRUCTIONS:
Stack Related instructions are used for accessing the stack.
e.g PUSH,
POP,
XTHL,
SPHL

Group VI - I/O INSTRUCTION
I/O instructions are used for reading or writing the input output port.
e.g IN addr-8
OUT addr-8

Group VI -MACHINE CONTROL INSTRUCTIONS:
The control instructions control the operation of microprocessor.
e.g. HLT,
NOP,
EI,
DI

9) **Explain the Data transfer instructions of 8085 with example.**

Copy from source to destination		
MOV	Rd, Rs	This instruction copies the contents of the source register into the destination register, the contents of Rd, M the source register are not altered. If one of the operands is a memory location, its location is specified by the contents of the HL registers. **Example:** MOV B, C or MOV B, M
	M, Rs	
	Rd, M	

Move immediate 8-bit		
MVI	Rd, data	The 8-bit data is stored in the destination register or memory. If the operand is a memory location, its location is specified by the contents of the HL registers. **Example:** MVI B, 57H or MVI M, 57H
	M, data	

Load accumulator		
LDA	16-bit address	The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator. The contents of the source are not altered. **Example:** LDA 2034H

Load accumulator indirect		
LDAX	B/D Reg. pair	The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the Accumulator. The contents of either the register pair or the memory location are not altered. **Example:** LDAX B

Load register pair immediate		
LXI	Reg. pair, 16-bit data	The instruction loads 16-bit data in the register pair designated in the operand. **Example:** LXI H, 2034H or LXI H, XYZ

Load H and L registers direct		
LHLD	16-bit address	The instruction copies the contents of the memory location pointed out by the 16-bit address into register L and copies the contents of the next memory location into register H. The contents of source memory locations are not altered. **Example**: LHLD 2040H

Store accumulator direct		
STA	16-bit address	The contents of the accumulator are copied into the memory location specified by the operand. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. **Example:** STA 4350H

Store accumulator Indirect		
STAX	Reg. pair	The contents of the accumulator are copied into the memory location specified by the contents of the operand (register pair). The contents of the accumulator are not altered. **Example:** STAX B

Store H and L registers direct		
SHLD	16-bit address	The contents of register L are stored into the memory location specified by the 16-bit address in the operand and the contents of H register are stored into the next memory location by incrementing the operand. The contents of registers HL are not altered. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. **Example:** SHLD 2470H

Exchange H and L with D and E		
XCHG	none	The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E. **Example:** XCHG

10) Explain the Arithmetic instructions of 8085 with example.

ADD R/M –Add register or memory to accumulator:

- ❖ The contents of the operand (register or memory) are added to the contents of the accumulator andthe result is stored in the accumulator.
- ❖ If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the addition.
- ❖ Example: ADD B or ADD M

ADC R/ M- Add register to accumulator with carry:

- ❖ The contents of the operand (register or memory) and the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator.
- ❖ If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the addition.
- ❖ Example: ADC B or ADC M

ADI 8-bit data - Add immediate to accumulator:

- ❖ The 8-bit data (operand) is added to the contents of the accumulator and the result is

stored in the accumulator.
- All flags are modified to reflect the result of the addition.
- Example: ADI 45H

ACI 8-bit data- Add immediate to accumulator with carry:
- The 8-bit data (operand) and the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator.
- All flags are modified to reflect the result of the addition.
- Example: ACI 45H

DAD Reg. pair - Add register pair to H and L registers:
- The 16-bit contents of the specified register pair are added to the contents of the HL register and the sum is stored in the HL register.
- The contents of the source register pair are not change. If the result is larger than 16 bits, the CY flag is set. No other flags are affected.
- Example: DAD H

SUB R / M- Subtract register or memory from accumulator:
- The contents of the operand (register or memory) are subtracted from the contents of the accumulator, and the result is stored in the accumulator.
- If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the subtraction.
- Example: SUB B or SUB M

SBB R / M- Subtract source and borrow from accumulator:
- The contents of the operand (register or memory) and the Borrow flag are subtracted from the contents of the accumulator and the result is placed in the accumulator.
- If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the subtraction.
- Example: SBB B or SBB M

SUI 8-bit data- Subtract immediate from accumulator:
- The 8-bit data (operand) is subtracted from the contents of the accumulator and the result is stored in the accumulator.
- All flags are modified to reflect the result of the subtraction.
- Example: SUI 45H

SBI 8-bit data - Subtract immediate from accumulator with borrow:
- The 8-bit data (operand) and the Borrow flag are subtracted from the contents of the accumulator and the result is stored in the accumulator.
- All flags are modified to reflect the result of the subtraction.
- Example: SBI 45H

INR R/ M - Increment register or memory by 1:
- The contents of the designated register or memory) are incremented by 1 and the result is stored in the same place.
- If the operand is a memory location, its location is specified by the contents of the HL registers.

- ❖ Example: INR B or INR M

INX R - Increment register pair by 1:

- ❖ The contents of the designated register pair are incremented by 1 and the result is stored in the same place.
- ❖ Example: INX H

DCR R/ M- Decrement register or memory by 1:

- ❖ The contents of the designated register or memory are decremented by 1 and the result is stored in the same place.
- ❖ If the operand is a memory location, its location is specified by the contents of the HL registers.
- ❖ Example: DCR B or DCR M

DCX R - Decrement register pair by 1:

- ❖ The contents of the designated register pair are decremented by 1 and the result is stored in the same place.
- ❖ Example: DCX H

DAA none - Decimal adjust accumulator:

- ❖ The contents of the accumulator are changed from a binary value to two 4-bit binary coded decimal (BCD) digits. This is the only instruction that uses the auxiliary flag to perform the binary to BCD conversion, and the conversion procedure is described below. S, Z, AC, P, CY flags are altered to reflect the results of the operation.
- ❖ If the value of the low-order 4-bits in the accumulator is greater than 9 or if AC flag is set, the instruction adds 6 to the low-order four bits. If the value of the high-order 4-bits in the accumulator is greater than 9 or if the Carry flag is set, the instruction adds 6 to the high-order four bits.
- ❖ Example: DAA

11) What is stack? Explain stack related instruction with example.

- ❖ The stack is a collection of memory locations containing a register that stores the top-of-element address in digital computers.
- ❖ Stack's operations are:

Push: Adds an item to the top of the stack.
Pop: Removes one item from the stack's top.

Push register pair onto stack		
PUSH	Reg. pair	The contents of the register pair designated in the operand are copied onto the stack in the following sequence. The stack pointer register is decremented and the contents of the high- order register (B, D, H, A) are copied into that location. The stack pointer register is decremented again and the contents of the low-order register (C, E, L, flags) are copied to that location. **Example:** PUSH B or PUSH A

Pop off stack to register pair		
POP	Reg. pair	The contents of the memory location pointed out by the stack pointer register are copied to the low-order register (C, E, L, status flags) of the operand. The stack pointer is incremented by 1 and the contents of that memory location are copied to the high-order register (B, D, H, A) of the operand. The stack pointer register is again incremented by 1. **Example:** POP H or POP A

12) **Explain Subroutine with CALL and RET Instruction**.

<u>Unconditional subroutine call</u>

CALL 16-bit address:

- ❖ The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- ❖ Before the transfer, the address of the next instruction after CALL(the contents of the program counter) is pushed onto the stack.
- ❖ Example: CALL 2034H or CALL XYZ

<u>Call conditionally:</u>

- ❖ The program sequence is transferred to the memory location specified by the 16-bit address given in the operand based on the specified flag of the PSW as described below.
- ❖ Before the transfer, the address of the next instruction after the call (the contents of the program counter) is pushed onto the stack.
- ❖ Example: CZ 2034H or CZ XYZ

Opcode	Description	Flag Status
CC	Call on Carry	CY=1
CNC	Call on no Carry	CY=0
CZ	Call on zero	Z=1
CNZ	Call on no zero	Z=0
CP	Call on positive	S=0
CM	Call on minus	S=1
CPE	Call on parity even	P=1
CPO	Call on parity odd	P=0

<u>Return from subroutine unconditionally</u>

RET none:

- ❖ The program sequence is transferred from the subroutine to the calling program.
- ❖ The two bytes from the top of the stack are copied into the program counter, and program execution begins at the new address.
- ❖ Example: RET

<u>Return from subroutine conditionally</u>

- ❖ The program sequence is transferred from the subroutine to the calling program based on the specified flag of the PSW as described below.
- ❖ The two bytes from the top of the stack are copied into the program counter, and program execution begins at the new address.
- ❖ Example: RZ

Opcode	Description	Flag Status
RC	Return on Carry	CY=1
RNC	Return on no Carry	CY=0
RZ	Return on zero	Z=1
RNZ	Return on no zero	Z=0
RP	Return on positive	S=0
RM	Return on minus	S=1
RPE	Return on parity even	P=1
RPO	Return on parity odd	P=0

13) Explain the addressing mode of 8085.

OR

What do you mean by addressing mode? Explain diff. addressing mode for 8085 with examples.

- ❖ Every instruction of a program has to operate on a data.
- ❖ Data may be direct in instruction, in Register or in Memory. The method of specifying the data into the instruction is called Addressing mode.
- ❖ The 8085 has the following 5 different types of addressing.
  1. Immediate Addressing mode

2. Register addressing mode
3. Direct Addressing mode
4. Indirect Addressing mode
5. Implied Addressing mode

Immediate Addressing

- In immediate addressing mode, the data is specified in the instruction itself. The data will be a part
- of the program instruction. All instructions that have ‘ I ’ in their mnemonics are of immediate
- addressing type. For Example, MVI B, 3EH - Move the data 3EH given in the instruction to B
- register.

Register Addressing

- In register addressing mode, the instruction specifies the name of the register in which the data is
- available. This type of addressing can be identified by register names (such as ‘A’, ‘B’ etc.) in the
- instruction. For Example, MOV A, B -Move the content of B register to A register.

Direct Addressing

- In direct addressing mode, the data will be in memory. The address of the data is specified in the
- instruction directly.
- For Example, LDA 1050H - Load the data available in memory location 1050H in accumulator.

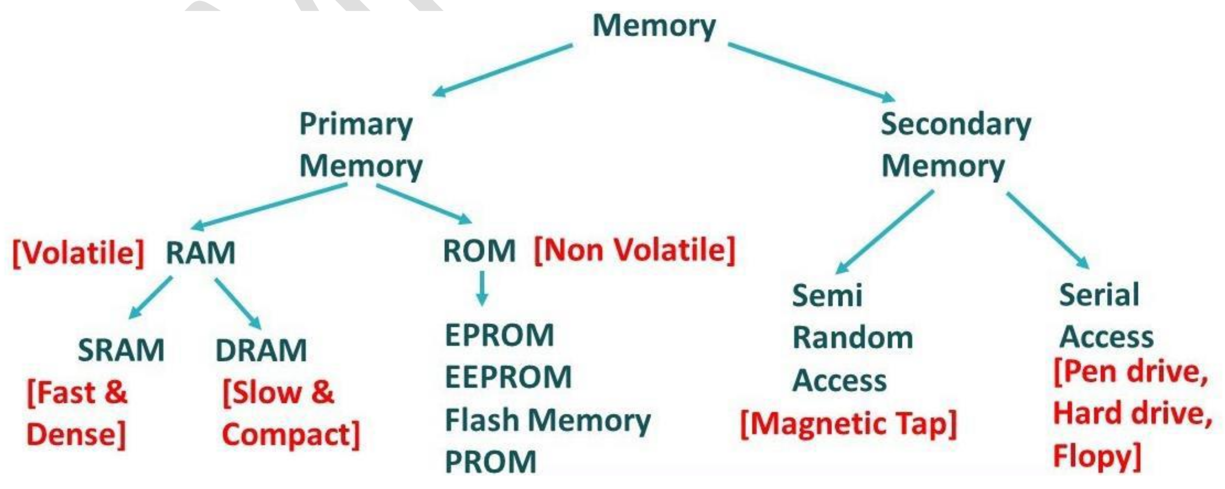
Indirect Addressing

- In indirect addressing mode, the data will be in memory. The address of the data is specified in the
- instruction indirectly i.e. address is store in Registers. This type of addressing can be identified by
- letter ‘M’ present in the instruction.
- For Example: MOV A, M - The memory data addressed by HL pair is moved to A register.

Implied Addressing

- In implied addressing mode, there is no operand. i.e. This type of instruction does not have any
- address, register name, immediate data specified along with it.
- For Example, CMA - Complement the content of accumulator.

14) Explain Memory classification



Various types of Main memories

RAM

- Random Access Memory
- RAM losses its contents when the power is turned OFF. So, RAM is called volatile memory.
- User save data in memory but they may not be store permanently.
- It is called due to Random selection of Memory Location.
- The microprocessor can write into or read from this memory.
- Types of RAM are:
  - Static RAM (SRAM)
  - Dynamic RAM(DRAM)
- Static RAM
  - It consists of flip flops that stores the binary information.
  - It is used in implementing the cache.
  - It is expensive, fast, low density, high power in operation.
  - Easy to use and shorter Read and Write cycles.
- Dynamic RAM
  - It is used in implementing the main memory.
  - It stores the binary information in the form of electric charges.
  - It offers reduced power consumption and large storage capacity.

ROM

- ROM does not loss its contents when the power is turned OFF. So, ROM is called Non volatile memory.
- The microprocessor can only read from this memory.
- It is programmable once.
- Types of ROM are:
  - PROM
  - EPROM
  - EEPROM
  - Masked ROM
- The two chip select inputs must be CS1=1 and CS2=0 for the unit to operate.
- Otherwise the data bus is in High Impedance State.
- When the chip is enabled by the two select inputs, the byte selected by the address lines appears on the data bus.

- **PROM :**
  - Programmable Read only memory
  - PROM Programmer program in a blank chip.
  - Once you program it, You can never change.
  - It maintains large storage media but can not offer the erase feature.

- Write Data once and Read many times.
- It has poly silicon wires arranged in a matrix.
- These wires can be functionally viewed as diodes or fuses.

❖ **EPROM(Erasable Programmable Read only memory)**

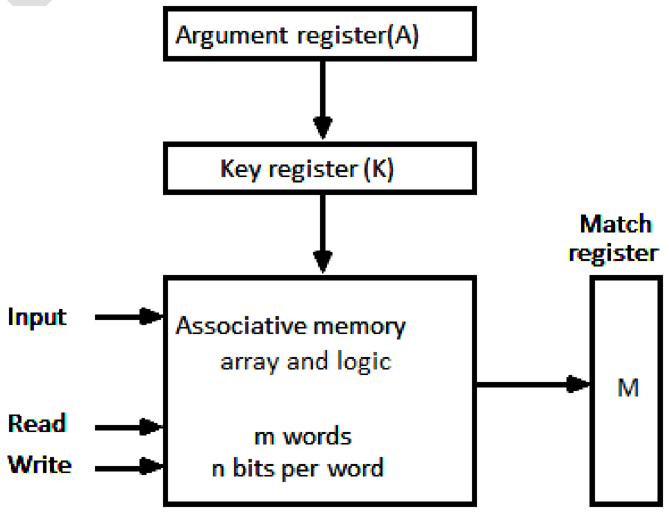
- This memory stores a bit by charging the gate of field effect transistor.
- Information is stored by EPROM Programmer, which apply high voltage to charge the gate.
- User can delete the data of EPROM through pass on Ultraviolet Light and the chip can be reprogrammed.
- Chip can be used many times.
- Erasing Process takes 15 to 20 minutes.

❖ **EEPROM (Electrically Erasable Programmable Read only memory)**

- Functionally similar to EPROM.
- Information can be changed by using electrical signals at the register level instead of all the information.
- Entire chip can be erased in 10ms.
- This memory is expensive compared to EPROM.

15) Explain Associative Memory.

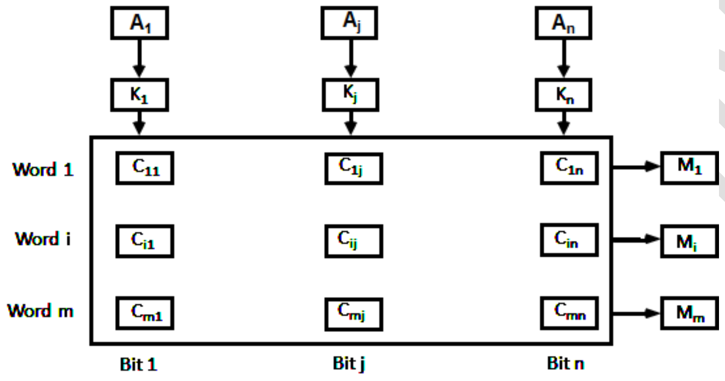
- Search Process:
- Choose a sequence of addresses
- Read the content of memory at each address
- Compare the information with searched items.
- A memory unit accessed by content is called an Associative Memory or CAM(Content Addressable Memory)



- When a word is written in an associative memory, no address is given.
- The memory is capable of finding an empty unused location to store the word.

- When the word is to be read from memory, the content of the word is specified.
- Associative memory is more Expensive than a RAM.
- The argument register A and key register have n bits.
- Match register M has m bits.

Example: A 101 111100
K 111 000000
Word1 100 111100 No match
Word2 101 000011 Match Occur



16) Explain Cache Memory.

❖ Cache memory is a small, temporary fast memory.
- It is placed between CPU and Memory.
- It runs at speeds similar to CPU Registers.
- Access Time is less than the access time of main memory.

❖ Operation of Cache Memory:
- When the CPU needs to access memory, the cache is examined.
- If the word is found in the cache, It is read from that.
- If the word is not found, the main memory is accessed to read the word and block of words are transferred from main memory to cache memory for future references.

❖ Performance Parameter of Cache Memory
- Hit: When the CPU refers to memory and find the word in cache , it is said to produce a HIT.
- Miss: The requested data is not found in the cache memory, then it counts as a MISS.
- Hit Rate: Performance of cache memory is measured in terms of quality, is called Hit Rate(Hit Ratio)
- Hit Rate = No. of Hits / Total CPU references to Cache memory
- Hit Rate= No. of Hits / (Hits + Misses)
- Miss Rate: The percentage of memory accesses not found in a given level of memory.
- Miss Rate=1- Hit Rate
- Hit Time: The time required to access the requested information in a given level of memory.
- Miss Penalty: The time required to process a miss, which include replacing a block in memory plus the time required to deliver requested data to the processor.

❖ Writing into Cache

Read Operation: When CPU finds a word in a Cache Memory, Main Memory is not involved.

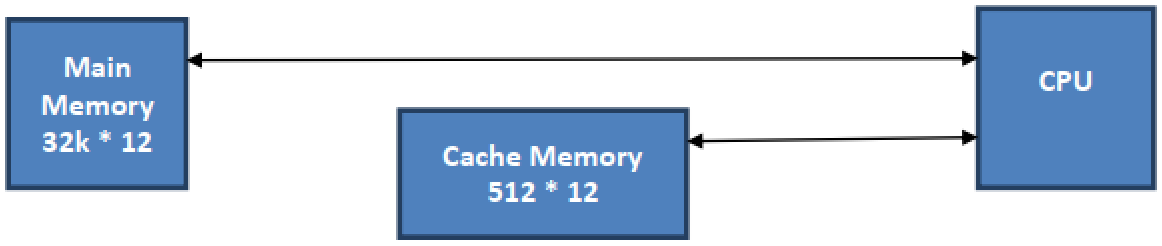
Write Operation: When CPU finds a word in a Cache Memory, Main Memory is involved.

Write Through Method:
- When Cache memory is update, the main memory is also updated in Parallel.
- Main Memory always content same data as cache.

Write Back Method:
- Only the cache location is updated during a write.
- The location is then marked by flag.
- So, when the word is removed from cache, it is copied into memory.
- When the word is resides in the cache, it may be updated several times.

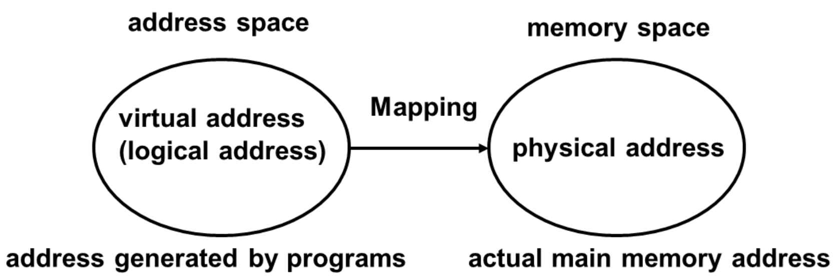
❖ The transformation of data from main memory to cache memory is referred to as a mapping process. Types of Mapping:

1) Associative Mapping
2) Direct Mapping
3) Set-associative Mapping



17) Explain Virtual Memory

❖ It is a concept used in a large computer system.
❖ Virtual memory in computer organization architecture is a technique and not actually a memory in physical form present in computer system.
❖ Virtual memory in COA is simply a technique used to provide an illusion of presence of large main memory to the programmer, when in actual it's not present physically.


address space
memory space
virtual address (logical address)
Mapping
physical address
address generated by programs
actual main memory address

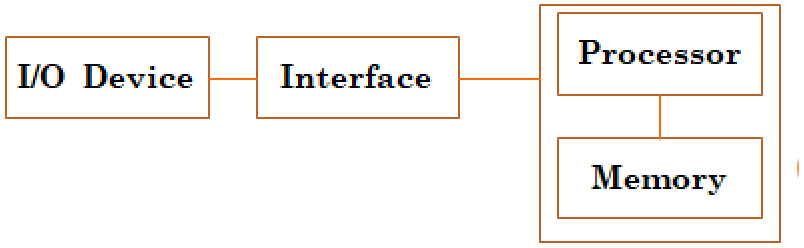
- The size of virtual memory is equivalent to the size of secondary memory. Each virtual address or logical address referenced by the CPU is mapped to a physical address in main memory.
- A hardware device called **Memory Management Unit** (MMU) performs this mapping during run time. To perform this activity MMU actually takes help of a memory map table, which is maintained by the operating system.

18) Differentiate various Modes of Data Transfer with I/O

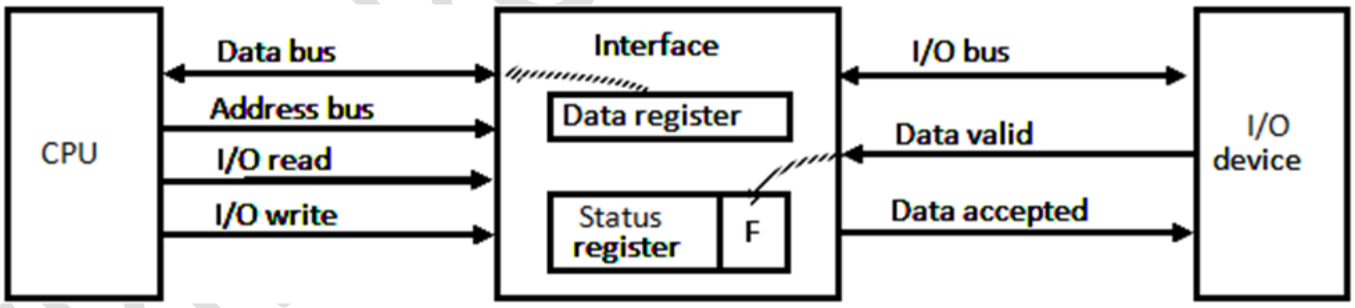
- Data transfer to and from peripherals may be handled in one of three possible modes.
  1) Programmed I/O
  2) Interrupt initiated I/O
  3) Direct Memory Access(DMA)

1. Programmed I/O

- The I/O device does not have direct access to memory.
- A transfer from I/O device to memory require the execution of several instruction by CPU(device to CPU and CPU to memory, no of words transferred)



.



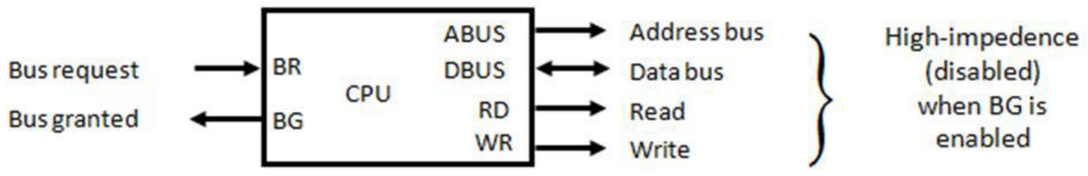
- When a byte of data is available, the device places it in the I/O bus and enables its data valid line.
- The interface accepts the byte into its data register and enables the data accepted line.
- The interface sets a bit in the status register that we will refer to as an F or "flag" bit.
- The device can now disables the data valid line.
- Then reading the status register into a CPU register and check the value of the flag bit .
- If the flag is equal to 1, the CPU reads the data from the data register.
- The flag bit is then cleared to 0 by CPU or Interface(depends o IC)
- Then Interface disable the data accepted line and the device can transfer the next data byte

2. **Interrupt Initiated I/O.**

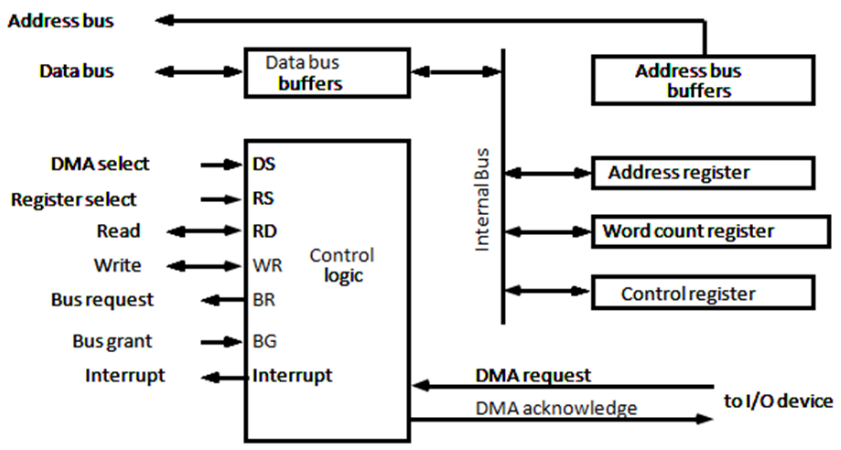
- In programmed initiated, CPU stays in a program loop until the I/O unit indicates that it is ready for data transfer.
- It keeps the processor busy without need.
- It can be avoided by using an interrupt facility .
- When the data are available from devices, interface issues an interrupt request signal.
- In the meantime CPU can proceed to execute another program.
- The interface meanwhile keeps monitoring the device.
- When the interface determines that the device is ready for data transfer, it generates an interrupt request to the computer.
- Upon detecting the external interrupt signal, CPU stops the task, branches to the service program to process I/O and then return to the task it was originally performing.

3. **DMA(Direct Memory Access) (OR Explain DMA.)**

- The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU.
- Removing the CPU from the path and peripheral device manage the buses directly will improve the speed of transfer.
- This transfer technique is called DMA.
- A DMA controller takes over the buses to manage the transfer directly between the I/O devices and memory.
- BR is used by DMA Controller to request CPU to relieve control of buses.
- BR is active, then CPU place address bus, data bus, read and write line into high impedance state.

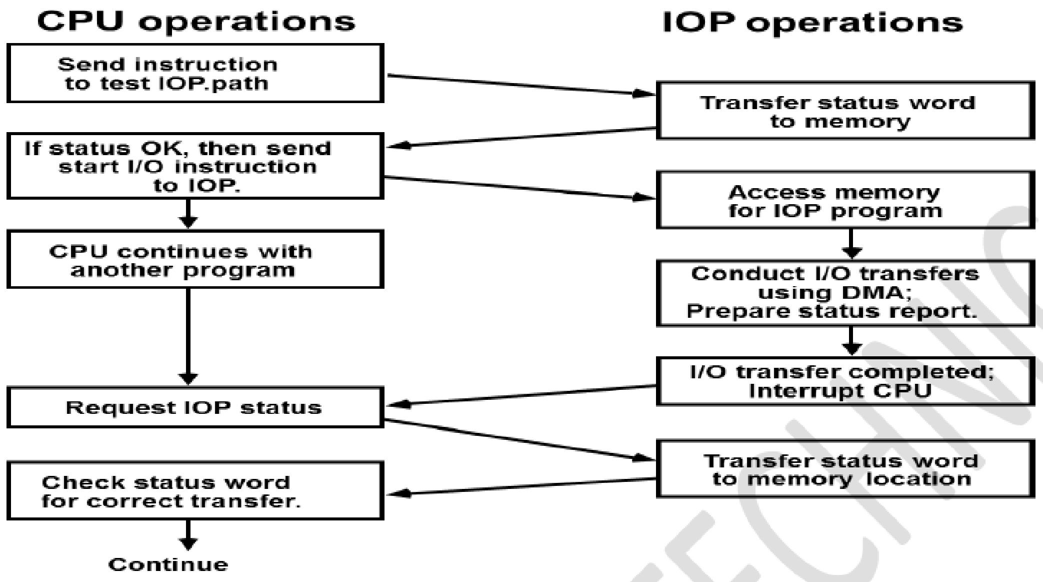


- The CPU activate the BG to inform DMA that buses are in high impedance state.
- DMA takes the control of buses and conduct memory transfer.
- When the DMA terminates the transfer, it disable the bus request line(BR).
- CPU disable the BG.



- ❖ The register in the DMA are selected by the DS (DMA select) and RS (register select) inputs.
- ❖ The RD (read) and WR (write) inputs are bidirectional.
- ❖ When the BG is 0, the CPU can communicate with the DMA registers through the data bus to read from or write to the DMA registers.
- ❖ When BG= 1, the CPU relieve the buses and the DMA can communicate directly with the memory
- ❖ The word count register holds the number of words to be transferred.

19) Describe CPU-IOP communication


CPU operations
IOP operations
Send instruction to test IOP.path
Transfer status word to memory
If status OK, then send start I/O instruction to IOP.
Access memory for IOP program
CPU continues with another program
Conduct I/O transfers using DMA; Prepare status report.
I/O transfer completed; Interrupt CPU
Request IOP status
Transfer status word to memory location
Check status word for correct transfer.
Continue

- ❖ CPU send an instruction to test the IOP path.
- ❖ The IOP responds by inserting a status word in memory for CPU.
- ❖ Status word indicate the condition of the IOP and I/O device(IOP overload condition, device busy, device ready)
- ❖ The CPU check the status word in memory to decide what to do.
- ❖ If status is OK, send start I/O instruction to IOP.
- ❖ Now CPU is continue with another program while IOP is busy with I/O program.
- ❖ When IOP terminates the execution, it send interrupt request to CPU.
- ❖ CPU send request for IOP Status.IOP send status report. It indicates the transfer has been completed or if any error occurs during the transfer.CPU check status word for correct transfer.