

Most IMP Questions of COA

UNIT : 1

- 1) Define Flip Flop. Explain any two application of it.
- 2) Draw and explain a common bus system for four registers.
- 3) Explain memory transfer.
- 4) Explain bus transfer.
- 5) Explain logic micro operation and list application of it.
- 6) Explain three state bus buffer.

UNIT : 2

- 1) List Memory Reference Instruction and explain any one of it.
- 2) Give list of basic computer register with their size and draw register organization with memory bank.
- 3) Draw and explain control unit of basic computer system.
- 4) Explain Instruction cycle.
- 5) Explain Instruction format.
- 6) Explain AND and BSA instructions.
- 7) Explain scheme used to differentiate 3 types of instructions in basic computer.

UNIT : 3

- 1) Explain stack organization of CPU.
- 2) Explain general register organization.
- 3) Explain any four addressing mode.
- 4) Explain characteristics of RISC and CISC.
- 5) $(3*4) + (5*6)$ convert into RPN and show stack operations.

UNIT : 4

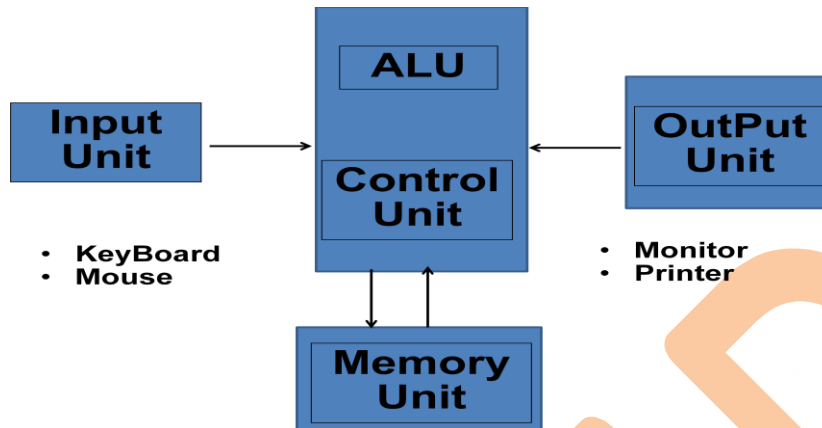
- 1) Explain RAM, ROM, EPROM and EEPROM.
- 2) Explain Main Memory.
- 3) Explain Virtual Memory.
- 4) Explain cache memory with any one mapping technique.
- 5) Explain Associative Memory.
- 6) Explain Auxiliary Memory.

UNIT : 5

- 1) Explain Method of Asynchronous Data Transfer.
- 2) Explain modes of Data Transfer.
- 3) Explain DMA.
- 4) Explain CPU-IOP communications.

Q-1 Draw and explain Block Diagram of Digital Computers.

Ans



- It is an interconnection of digital modules.
- It is a system that manipulates discrete elements of information that is represented internally in a binary form.
- Advantages:
 - Easy to Design
 - Low Cost
 - Very Fast Speed
 - More Popular with upgrading Technology
 - Easy to Function
 - Easy to Program

Q-2 Explain the Register Transfer Language.

Ans

- **Definition:** The symbolic notation used to describe the micro operation transfers among registers is called a register transfer language.
- It is a convenient tool for describing the internal organization of digital computers in concise and precise manner.
- The statement below denotes a transfer of the content of register R1 into register R2.

$$R2 \leftarrow R1$$

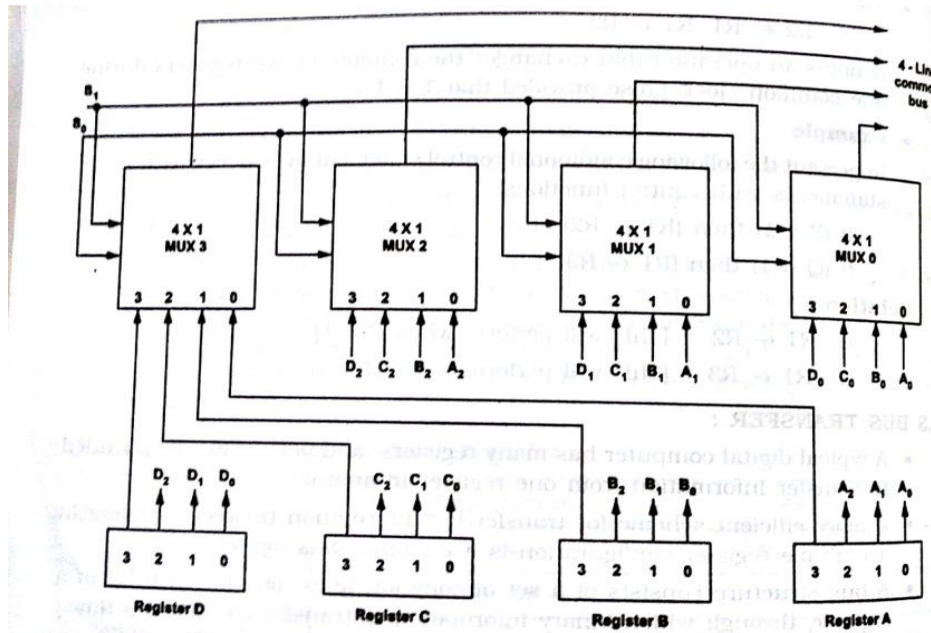
Q- 3 Define Flip Flop. Explain any two application of it.

- It can store 1-bit information.
- It can be designed using NAND or NOR gates.
- It has two stable states: Logic 1 and Logic 0
- **Types of Flip Flop**
 1. RS Flip Flop
 2. D Flip Flop
 3. T Flip Flop
 4. JK Flip Flop
 5. Master Slave JK Flip Flop
- **Application Of Flip Flop**
 1. Used as a memory element

2. Used as a delay element
3. Used as a basic block in counters and registers

Q-4 Draw and explain a common bus system for four registers.

Ans



- **Bus is a path (of a group of wires) over which information is transferred, from any of several sources to any of several destinations.**
- A bus structure consists of a set of common lines, one for each bit of a register, through which binary information is transferred one at a time.
- Control signal determine which register is selected by the bus.
- The two selection lines S_1 and S_0 are connected to the selection inputs of all four multiplexers.
- The selection lines choose the four bits of one register and transfer them into the four-line common bus.

S_1	S_0	Register Selected
0	0	Register A
0	1	Register B
1	0	Register C
1	1	Register D

Function Table for Bus

Q- 5 Explain Memory Transfer.

Ans

- **Read Operation:** The transfer of information from a memory word to the outside environment is called a read operation.

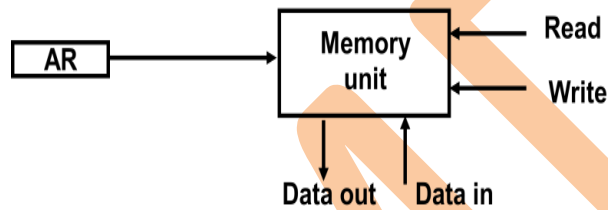
- **Write Operation:** The transfer of new information to be stored into the memory is called a write operation.
- A memory word will be symbolized by the letter M.

Read: $DR \leftarrow M[AR]$

This causes a transfer of information into DR from the memory word M selected by the address in AR.

Write: $M[AR] \leftarrow R1$

This causes a transfer of information from R1 into memory word M selected by address AR.



Q-6 Draw and explain Logic Micro-operations in detail. (Hardware implementation of Logic micro operation.)

Ans

Logic micro operations specify binary operations for strings of bits stored in registers.

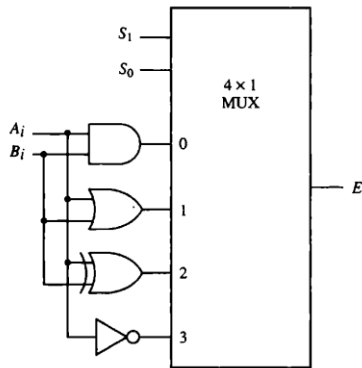


Fig. Hardware implementation of Logic micro operation

S ₁	S ₀	Output	Operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \hat{A}$	Compliment

Function table

Q-7 Explain application of logic micro operations.

OR

Explain selective set, selective complement and selective clear.

Logic micro operations specify binary operations for strings of bits stored in registers.

1) Selective-Set operation:

- a. The selective-set operation sets to 1 the bits in register A where there are corresponding 1's in register B. It does not affect bit positions that have 0's in B. The following numerical example clarifies this operation:

1010 A before

1100 B(logical operand)

1110 A after

- b. The two leftmost bits of B are 1's, so the corresponding bits of A are set to 1.
c. One of these two bits was already set and the other has been changed from 0 to 1.
d. The two bits of A with corresponding 0's in B remain unchanged.
e. The OR microoperation can be used to selectively set bits of a register.

2) Selective-Complement operation:

- The selective-complement operation complements bits in A where there are corresponding 1's in B. It does not affect bit positions that have 0's in B. For example:

1010 A before

1100 B(logical operand)

0110 A after

- Again the two leftmost bits of B are 1's, so the corresponding bits of A are complemented.
- The exclusive-OR microoperation can be used to selectively complement bits of a register.

Q- 8 Explain three-state bus buffer. OR

Explain the operation of three state bus buffers and show its use in design of common bus.

- A bus system can be constructed with three-state gates instead of multiplexers.
- A three-state gate is a digital circuit that exhibits three states.

State 1: Signal equivalent to Logic 1

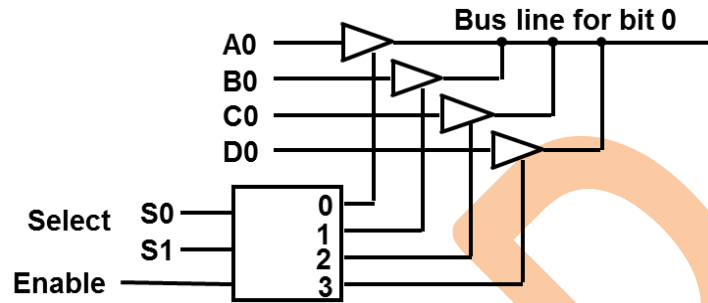
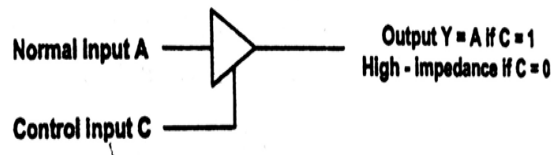
State 2: Signal equivalent to Logic 0

State 3: High Impedance State (behaves as open circuit)

- The **high-impedance** state behaves like an open circuit.
- The graphic symbol of a three-state buffer gate is shown in figure below:

Output $Y=A$ if $C=1$

High Impedance if $C=0$



Bus line with three state-buffers

Q-9 Draw registers organization with memory bank. Give list of Basic computer register with their Size and Range. OR Explain Registers of basic computer.

Ans:

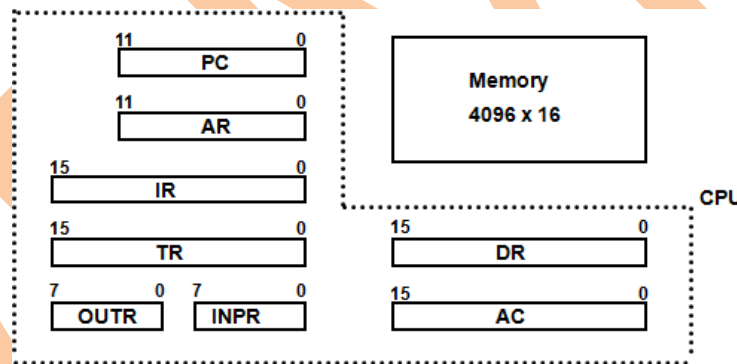


Figure 1: Basic Computer Register and Memory

Register Symbol	Bits	Register Name	Function
DR	16	Data register	Holds memory operand
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

Fig 2: list of basic computer register with size and range

- The data register (DR) holds the operand read from memory.
- The accumulator (AC) register is a general purpose processing register.

- The instruction read from memory is placed in the instruction register (IR).
- The temporary register (TR) is used for holding temporary data during the processing.
- The memory address register (AR) has 12 bits.
- The program counter (PC) also has 12 bits and it holds the address of the next instruction to be read from memory.
- The input register (INPR) receives an 8-bit character from an input device. The output register (OUTR) holds an 8-bit character for an output device.

Q-10 Draw and explain Control unit of Basic computer system.

Ans:

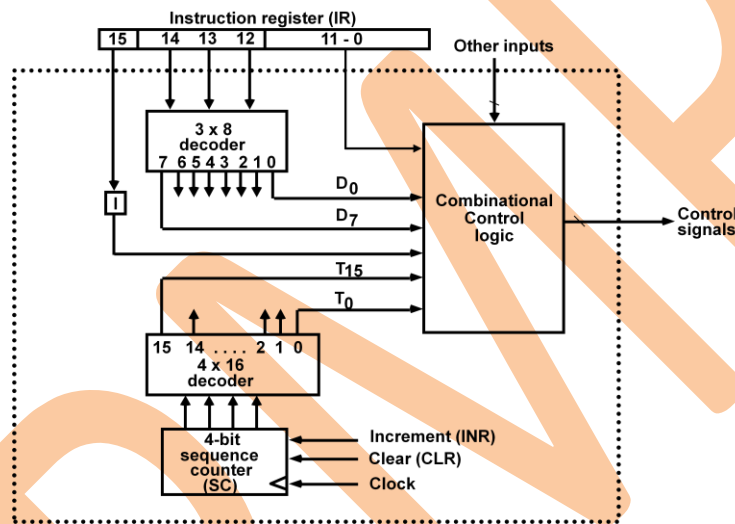


Fig: Block diagram of the control unit of basic computer

The block diagram of the control unit is shown in above figure.

- Components of Control unit are
 - Two decoders
 - A sequence counter
 - Control logic gates
- An instruction read from memory is placed in the instruction register (IR).
- In control unit the IR is divided into three parts: I bit, the operation code (12-14) bit, and bits 0 through 11.
- The operation code in bits 12 through 14 are decoded with a 3 X 8 decoder.
- Bit-15 of the instruction is transferred to a flip-flop designated by the symbol I.
- The eight outputs of the decoder are designated by the symbols D0 through D7. Bits 0 through 6 are applied to the control logic gates.
- The 4 bit sequence counter can count in a binary from 0 through 15. The outputs of counter are decoded into 16 timing signals T0 through T15.
- The sequence counter SC can be incremented or cleared synchronously. Most of the time, the counter is incremented to provide the sequence of timing signals out of 4 X 16 decoder.

If asked with timing diagram:

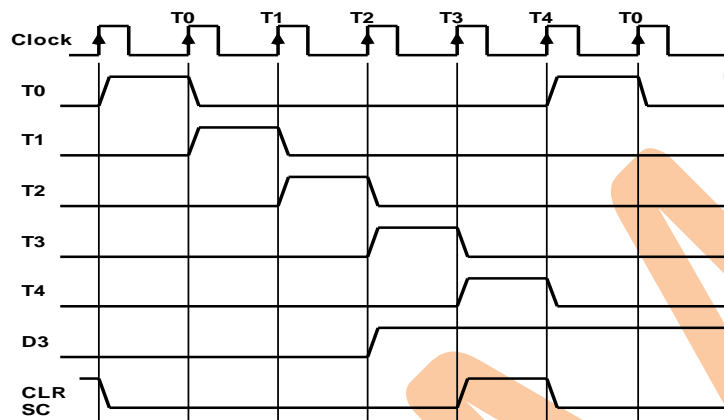
TIMING SIGNALS:

Generated by 4-bit sequence counter and 4×16 decoder

- The SC can be incremented or cleared.
- Example: $T_0, T_1, T_2, T_3, T_4, T_0, T_1, \dots$

Assume: At time T_4 , SC is cleared to 0 if decoder output D_3 is active.

$$D_3T_4: SC \leftarrow 0$$



- The last three waveforms shows how SC is cleared when $D_3T_4 = 1$.
- Output D_3 from the operation decoder becomes active at the end of timing signal T_2 .
- When timing signal T_4 becomes active, the output of the AND gate that implements the control function D_3T_4 becomes active.
- This signal is applied to the CLR input of SC. On the next positive clock transition the counter is cleared to 0. This causes the timing signal T_0 to become active instead of T_5 that would have been active if SC were incremented instead of cleared.

Q-11 Draw and explain the flowchart for instruction cycle.

Ans

In the basic computer each instruction cycle consists of the following phases:

1. Fetch an instruction from memory.
2. Decode the instruction.
3. Read the effective address from memory if the instruction has an indirect address.
4. Execute the instruction.

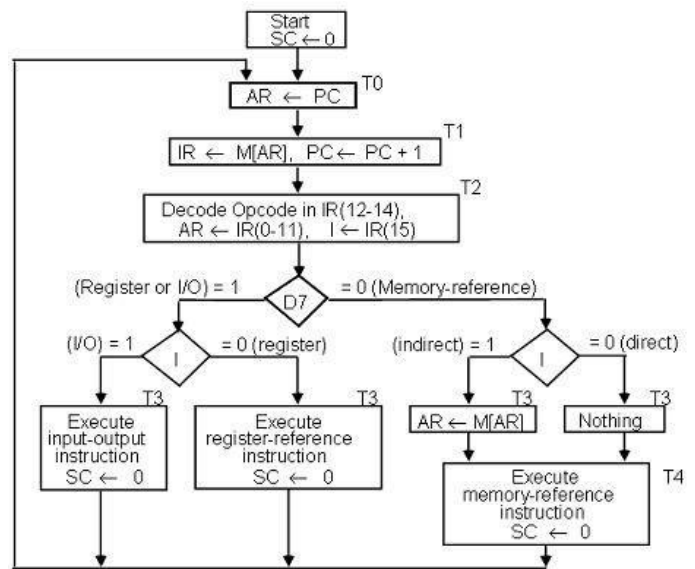


Fig. Flowchart for instruction cycle

- If $D7 = 1$, the instruction must be register-reference or input-output type. If $D7 = 0$, the operation code must be one of the other seven values 110, specifying a memory-reference instruction.
- If $D7 = 0$ and $I = 1$, we have a memory-reference instruction with an indirect address.
- $D7 I = 1$: Execute register-reference instruction
- $D7 I = 0$: Execute an input-output instruction

Q-12 Give the list of Memory Reference Instruction. Explain any THREE of it.

When the memory-reference instruction is decoded, $D7$ bit is set to 0.

Symbol	Operation Decoder	Symbolic Description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, PC \leftarrow AR + 1$
ISZ	D_6	$M[AR] \leftarrow M[AR] + 1, \text{ if } M[AR] + 1 = 0 \text{ then } PC \leftarrow PC + 1$

Above is a List of memory-reference instruction.

AND to AC

This is an instruction that performs the AND logic operation on pairs of bits in AC and the memory word specified by the effective address. The result of the operation is transferred to AC.

$D_0 T_4$: $DR \leftarrow M[AR]$ Read operand
 $D_0 T_5$: $AC \leftarrow AC \wedge DR, SC \leftarrow 0$ AND with AC

ADD to AC

This instruction adds the content of the memory word specified by the effective address to the value of AC. The sum is transferred into AC and the output carry Cout is transferred to the E (extended accumulator) flip-flop.

D₁T₄: DR ← M[AR] Read operand

D₁T₅: AC ← AC + DR, E ← C_{out}, SC ← 0 Add to AC and stores carry in E

LDA: Load to AC

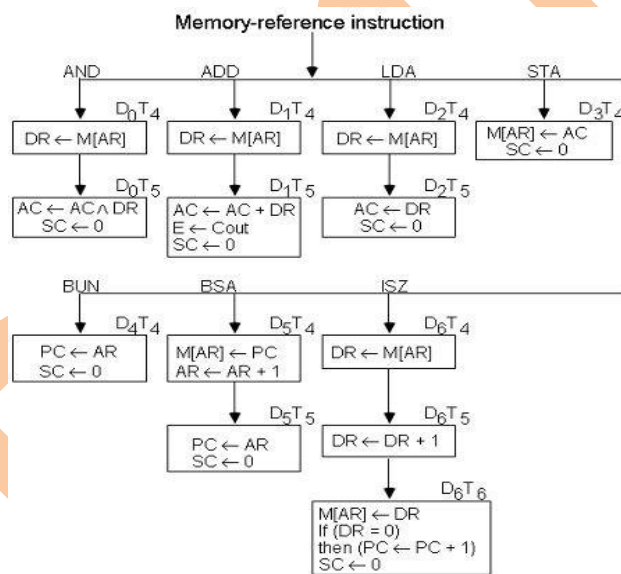
This instruction transfers the memory word specified by the effective address to AC.

D₂T₄: DR ← M[AR]

D₂T₅: AC ← DR, SC ← 0

Q-13 Draw the flowchart of Memory Reference Instruction.

Ans



Q-14 Explain AND and BSA instructions.

AND to AC

This is an instruction that performs the AND logic operation on pairs of bits in AC and the memory word specified by the effective address. The result of the operation is transferred to AC.

D₀T₄: DR ← M[AR] Read operand

D₀T₅: AC ← AC ∧ DR, SC ← 0 AND with AC

BSA: Branch and Save Return Address

This instruction is useful for branching to a portion of the program called a subroutine or procedure. When executed, the BSA instruction stores the address of the next instruction in sequence (which is available in PC) into a memory location specified by the effective address.

D₅T₄: M[AR] ← PC, AR ← AR + 1

D₅T₅: PC ← AR, SC ← 0

Q-15 Draw and Explain General registers organization.

OR

Draw Block Diagram of Bus Organization for Seven CPU Register & Explain with format of Control Word.

Ans

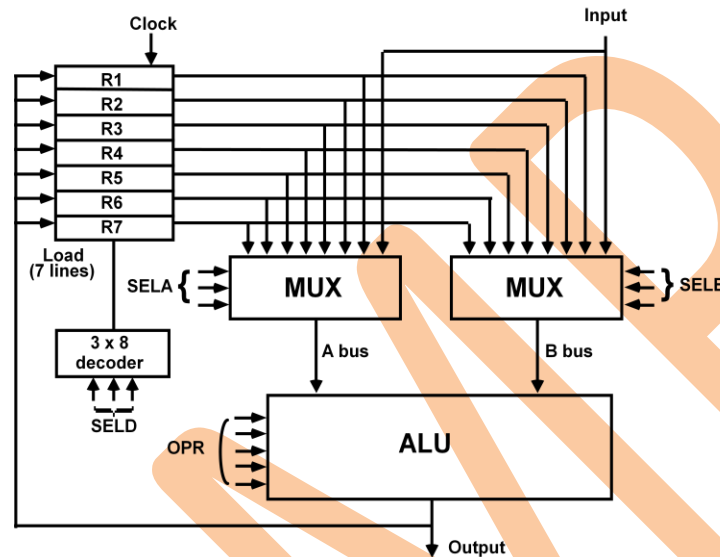


Fig. General registers organization

- The output of each register is connected to two multiplexers(MUX) to form the two buses.
- The selection line in each multiplexer select one register or the input data.
- A and B buses form the input to the ALU.
- ALU determines the arithmetic or logic microoperation.
- The result of microoperation is available for output data and also goes into the inputs of all the registers.
- The decoder select one register.
- Register receive the information from the output bus.

Control Word: It is 14-bit binary word. (Q- Explain Control word with example.)

3	3	3	5
SELA	SELB	SELD	OPR

- There are 14 binary selection inputs in the control unit and their combine value specifies a control word.
- It consists of 4 fields.
- The three bits of SEL A, select a source register for the A input of the CPU.
- The three bits of SEL B, select a source register for the B input of the CPU.
- The three bits of SEL D, select a destination register using the decoder.

- The five bits of OPR select one of the operations in the ALU.

Example:

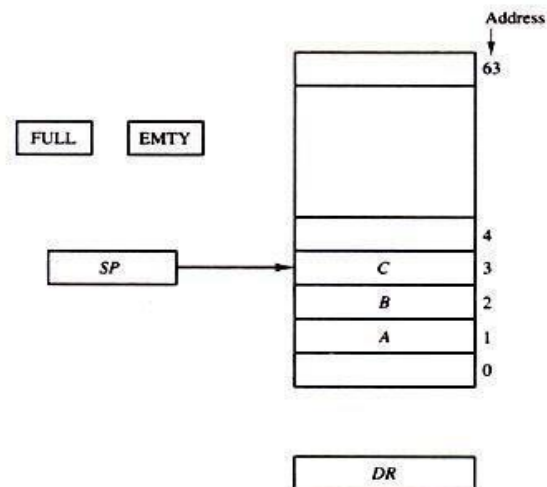
- $R1 \leftarrow R2 + R3$
 1. MUX A Selector (SEL A) : $BUS A \leftarrow R2$
 2. MUX B Selector (SEL B) : $BUS B \leftarrow R3$
 3. ALU operation selector(OPR) : ALU to ADD
 4. Decoder destination selector(SEL D): $R1 \leftarrow Out\ bus$

Field:	SEL A	SEL B	SEL D	OPR
Symbol:	R2	R3	R1	ADD
Control Word:	010	011	001	00010

Q- 16 What is Stack? Give the organization of register stack with all necessary elements and explain the working of push and pop operations. (OR Explain Register stack).

Ans

- **Stack :**
A stack is a storage device that stores information in such a manner that the item stored last is the first item retrieved.
- It follows LIFO.
- In a 64-word stack, the stack pointer contains 6 bits because $2^6 = 64$.
- The register that holds the address for the stack is called a stack pointer (SP). SP always points at the top item in the stack.
- Two operations of a stack are:
 1. Insertion of items (PUSH)
 2. Deletion of items (POP)



Block diagram of a 64-word Register stack

- The stack pointer register SP contains a binary number whose value is equal to the address of the word that is currently on top of the stack.
- DR is the data register that holds the binary data to be written into or read out of the stack.

- The one-bit register FULL is set to 1 when the stack is full, and the one-bit register EMTY is set to 1 when the stack is empty of items.

PUSH:

If the stack is not full (FULL = 0), a new item is inserted with a push operation. The push operation consists of the following sequences of micro operations:

$SP \leftarrow SP + 1$ Increment stack pointer
 $M[SP] \leftarrow DR$ WRITE ITEM ON TOP OF THE STACK

IF (SP = 0) then FULL ← 1 Mark the stack not empty
 EMTY ← 0 Check is stack is full

POP:

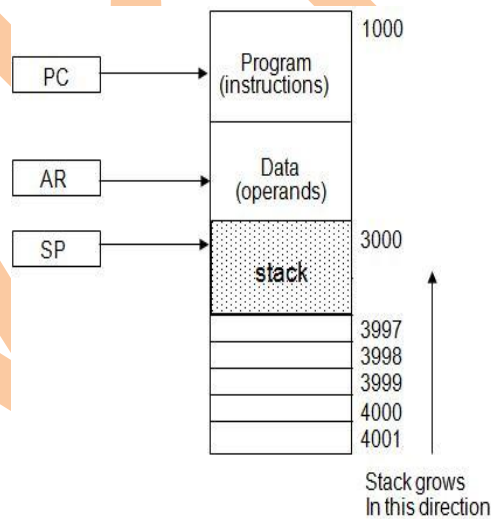
A new item is deleted from the stack if the stack is not empty (if EMTY = 0). The pop operation consists of the following sequences of micro operations:

$DR \leftarrow M[SP]$ Read item from top of the stack
 $SP \leftarrow SP - 1$ Decrement stack pointer
 IF (SP = 0) then EMTY ← 1 Check if stack is empty
 FULL ← 0 Mark the stack not full

Q-17 Explain Stack organization of CPU. OR Explain Stack organized CPU.

OR Explain Memory Stack.

Ans



Computer memory with program, data, and stack segments

- Computer memory is partitioned into three segments: program, data, and stack.
- The program counter PC points at the address of the next instruction in the program which is used to read an instruction.

- The address registers AR points at an array of data which is used to read an operand.
- The stack pointer SP points at the top of the stack which is used to push or pop items into or from the stack.
- PC, AR and SP are connected to a common address bus.
- DR is a data register.

PUSH

- A new item is inserted with the push operation as follows:

$SP \leftarrow SP - 1$

$M[SP] \leftarrow DR$

- The stack pointer is decremented so that it points at the address of the next word.
- A memory write operation inserts the word from DR into the top of the stack.

POP

- A new item is deleted with a pop operation as follows:

$DR \leftarrow M[SP]$

$SP \leftarrow SP + 1$

- The top item is read from the stack into DR.
- The stack pointer is then incremented to point at the next item in the stack.

Q-18 Explain any four addressing mode.

1) Implied mode:

- Operands are specified implicitly in the definition of the instruction.
- Example: The instruction, “Complement Accumulator “ is implied mode instruction, because the operand in accumulator register is implied in the definition of the instruction.
- All register reference instruction that use accumulator are implied.
- Zero address instruction are implied mode instruction.

2) Immediate Mode:

- In this mode the operand is specified in the instruction itself.
- Immediate mode instruction has an operand field instead of an address field.

3) Register Mode:

- In this mode the operands are in registers that resides within the CPU.
- The particular register is selected from a register field in the instruction.

4) Register Indirect Mode:

- The instruction specifies a register in the CPU, whose contents give the address of the operand in memory.
- The advantage of this mode is that address field of the instruction uses fewer bits to select a register than would have been required to specify a memory address directly.

5) Direct Address Mode:

- In this mode the effective address is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction.

6) Indirect Address Mode:

- In this mode the address field of the instruction gives the address where the effective address is stored in memory.

Q-19 Give characteristics of RISC and CISC.

Ans

Characteristics of RISC:

1. Relatively few instructions
2. Relatively few addressing modes
3. Memory access limited to load and store instructions
4. All operations done within the registers of the CPU
5. Fixed-length, easily decoded instruction format
6. Single-cycle instruction execution
7. Hardwired rather than microprogrammed control
8. A relatively large number of registers in the processor unit
9. Use of overlapped register windows to speed-up procedure call and return
10. Efficient instruction pipeline

Characteristics of CISC:

1. A larger number of instructions – typically from 100 to 250 instructions
2. Some instructions that perform specialized tasks and are used infrequently
3. A large variety of addressing modes – typically from 5 to 20 different modes
4. Variable-length instruction formats
5. Instructions that manipulate operands in memory

RISC	CISC
REDUCED INSTRUCTION SET COMPUTER	COMPLEX INSTRUCTION SET COMPUTER
Relatively few instructions	A larger number of instructions – typically from 100 to 250 instructions.
Relatively few addressing modes	A large variety of addressing modes – typically from 5 to 20 different modes
Fixed-length, easily decoded instruction format	Variable-length instruction formats
All operations done within the registers of the CPU	Instructions that manipulate operands in memory

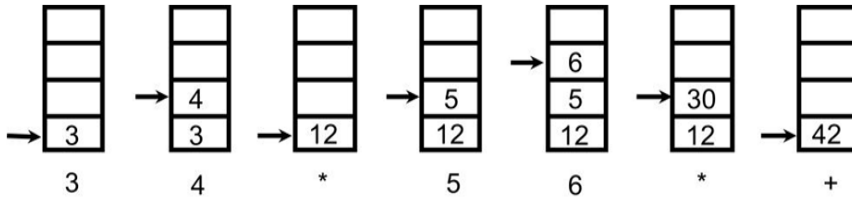
Q-20 Convert into RPN and show stack operations: $(3*4) + (5*6)$.

Ans

Evaluation of Arithmetic Expressions

Any arithmetic expression can be expressed in parenthesis-free Polish notation, including reverse Polish notation

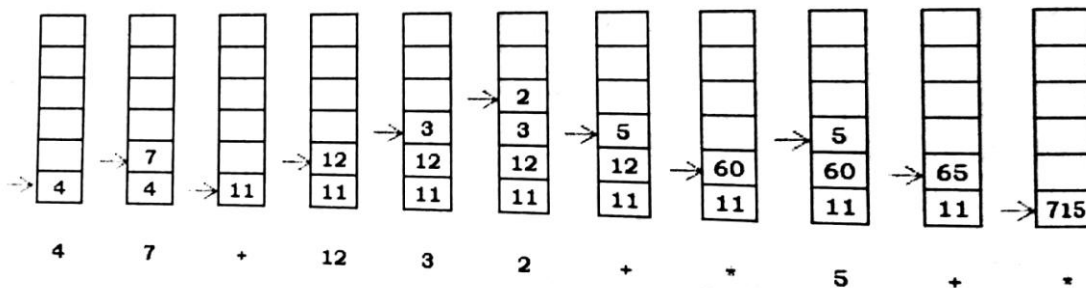
$$(3 * 4) + (5 * 6) \Rightarrow 3 4 * 5 6 * +$$



Example 2: $(4+7) * [12 * (3 + 2) + 5]$

Conversion of given expression into RPN :

$$\begin{aligned} &= (4 + 7) \times [12 \times 3 2 + + 5] \\ &= (4 + 7) \times [12 3 2 + * + 5] \\ &= (4 + 7) \times [12 3 2 + * 5 +] \\ &= (4 7 +) \times [12 3 2 + * 5 +] \\ &= 4 7 + 12 3 2 + * 5 + * \end{aligned}$$



Q-21 A) Explain Data Transfer Instruction.

- Data transfer instructions move data from one place in the computer to another without changing the data content.
- The most common transfers are between memory and processor registers, between processor registers and input or output, and between the processor registers.

Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT

Push	PUSH
Pop	POP

- Load: To perform Data transfer from memory to processor register.
- Store: To perform data transfer from processor register into memory location.
- Move: To perform data transfer from one register to another.
- Exchange: Swaps information between two registers or register and a memory word.
- Input and Output: Data transfer between processor register and input-output terminal.
- Push and Pop: Data transfer between processor register and memory stack.

B) Explain Arithmetic data manipulation instruction.

Ans

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate (2's complement)	NEG

- The four basic arithmetic operations are addition, subtraction, multiplication and division.
- Increment instruction adds 1 to the value stored in a register or memory word.
- Decrement instruction subtracts 1 from a value stored in a register or memory word.
- Add with carry performs the addition on two operands plus the value of the carry from the previous computation.
- Subtract with borrow instruction subtract two words and borrow which may have resulted from a previous subtract operation.

Q- List types of Interrupt.

- There are three major types of interrupts that cause a break in the normal execution of a program.
- They can be classified as:
 1. External interrupts

2. Internal interrupts
3. Software interrupts

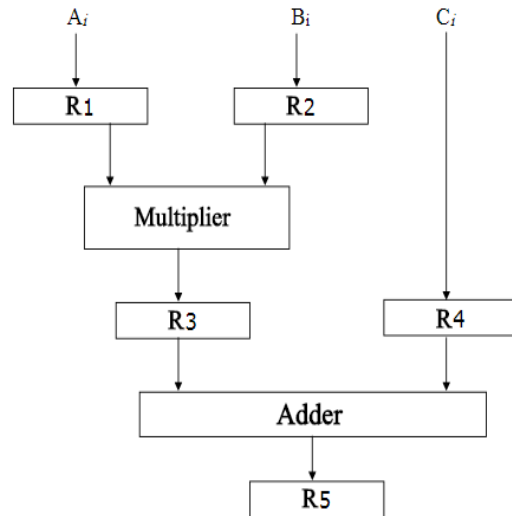
Q-22 Explain Parallel Processing.(OR Draw pipeline processing for instruction: $A_i * B_i + C_i$)

- Parallel Processing is a term used to denote a large class of techniques that are used to provide simultaneous data processing tasks for the purpose of increasing the computational speed of a computer System.
- A Parallel Processing system is able to **perform concurrent data processing to achieve faster Execution time.**

The purpose of parallel processing is **to speed up the computer processing.**

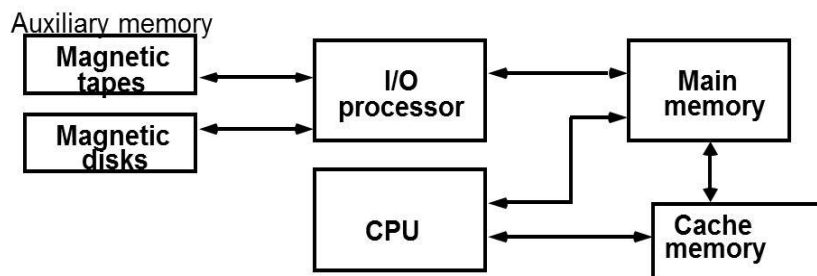
Example:

- $R1 \leftarrow A_i, R2 \leftarrow B_i$
Input A_i and B_i
- $R3 \leftarrow R1 * R2, R4 \leftarrow C_i$
Multiply and input C_i
- $R5 \leftarrow R3 + R4$
Add C_i to product



Q-23 Explain Memory Hierarchy.

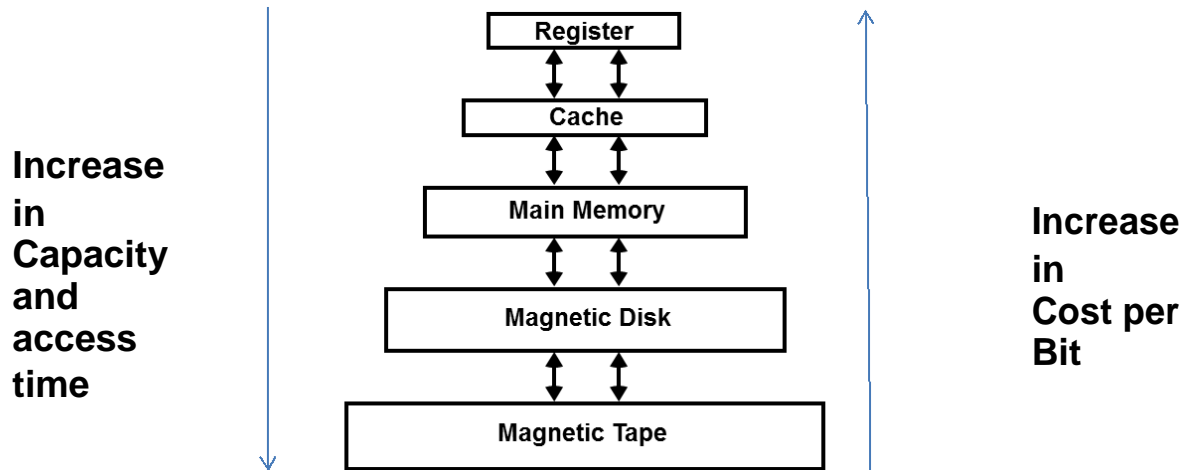
- The memory unit that communicates directly with CPU is called Main Memory.
- Devices that provide backup storage are called Auxiliary Memory.



- Only program and data **currently needed** by the processor reside in main memory.
- The **Main Memory communicates directly with CPU and with Auxiliary Memory Devices through an I/O Processor.**
- Cache is used to increase the speed of processing by making current programs and data available to CPU.
- Auxiliary Memory → Large storage capacity, Relatively Inexpensive, Low access speed compared to main memory.
- Cache Memory → Very small, Relatively expensive, very high access speed.

- **Advantage of Memory Hierarchy:**

Increase capacity
 Improve average access time
 Decrease cost/bit



Q-24 A) Explain Main Memory. OR How main memory is useful in computer system? OR Explain RAM and ROM.

Ans:

Main memory is divided into two subcategories RAM and ROM.

RAM

- RAM loses its contents when the power is turned OFF. So, RAM is called volatile memory.
- User saves 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.

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
 Flash Memory

Q- 24 B) Explain PROM, EPROM and EEPROM.

PROM: Programmable Read only memory

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

EPROM: Erasable Programmable Read only memory

- 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.

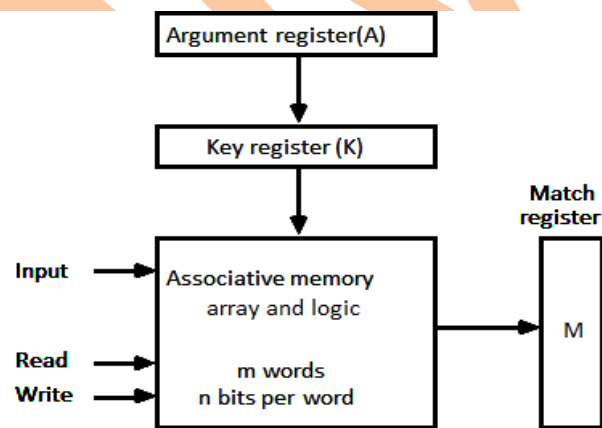
EEPROM: Electrically Erasable Programmable Read only memory

- 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.
- Functionally similar to EPROM.

Q- 25 Explain Content Addressable Memory (CAM) OR

Explain Associative Memory.

- 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

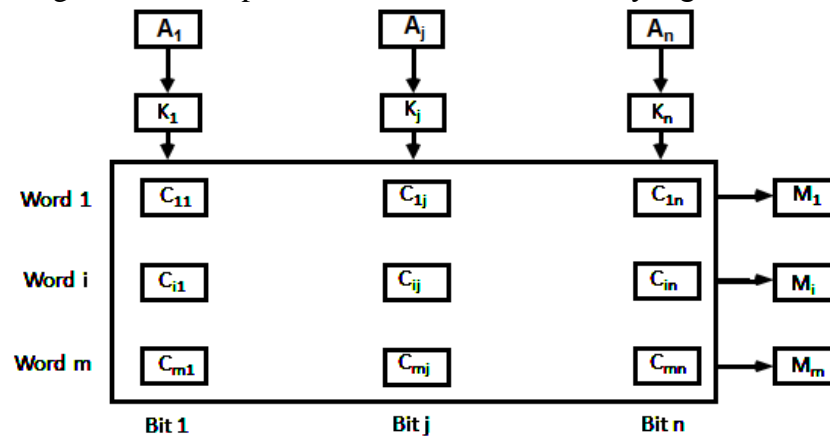
Word1 100 111100

No match

Word2 101 000011

Match Occur

- Here only leftmost three bits are compared with memory words because k has 1's in three positions.
- Word 2 is match with key register, because leftmost three bits of the argument and word are equal.
- The entire argument is compared with each word if the key register contains all 1's.



Q-26 Explain Auxiliary Memory.

Magnetic Disk

- Bits are stored in magnetized surface along concentric circles are called Tracks.
- Tracks are divided into sections called Sectors.
- A disk system is addressed by address bits that specify the Disk number, Disk surface, Sector number and the Track within the sector.

How To Read/Write?

- First Read/Write heads are positioned in the specified **Track**.
- The system has to wait until the disk reaches the **specified sector** under Read/Write head.
- Information **transfer is very fast** once sector has been reached.
- Storage capacity of disk depends on **bits per inch of track** and **Track per inch of surface**.

Floppy Disk

- A disk drive with removable disk is called a Floppy Disk.
- It is made of plastic coated magnetic material.
- Slower access than Hard Disk
- Less storage capacity
- Less Expensive
- Portable

Magnetic Tape

- It consists of **magnetic**, coating on a thin plastic strip.
- Example: Recording Tap(used for audio, video, general purpose digital data storage using a computer)
- Secondary Storage Device

Q-27 Explain Cache Memory OR Write short note on 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. (Q- Explain Performance Parameter of Cache.)

1. Hit: When the CPU refers to memory and find the word in cache , it is said to produce a HIT.
2. Miss: The requested data is not found in the cache memory, then it counts as a MISS.
3. Hit Rate: Performance of cache memory is measured in terms of quality, is called Hit Rate(Hit Ratio)

$$\text{Hit Rate} = \text{No. of Hits} / \text{Total CPU references to Cache memory}$$

$$\text{Hit Rate} = \text{No. of Hits} / (\text{Hits} + \text{Misses})$$

Q-28 Explain Write-through and Write back cache write method.

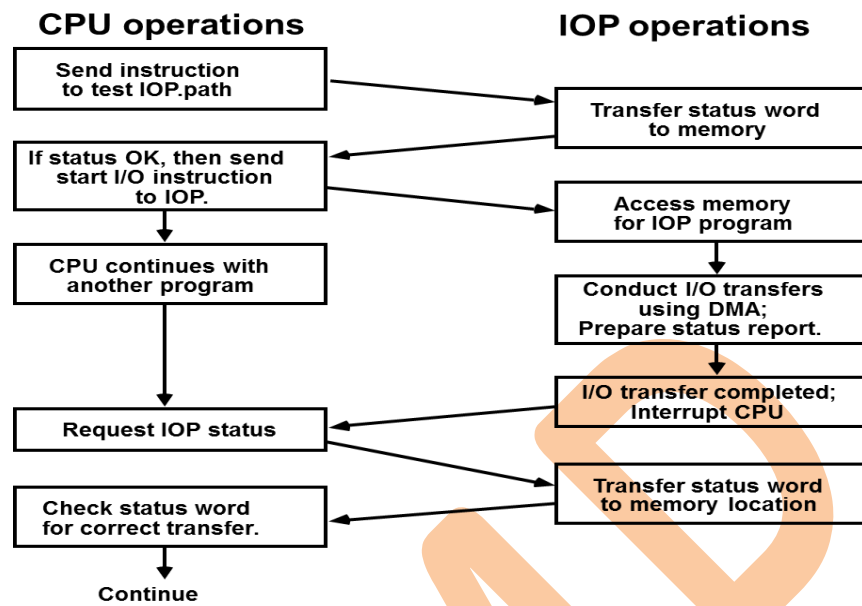
Write Through Method:

- 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.

Q- 29 Explain CPU-IOP communications.



- 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.

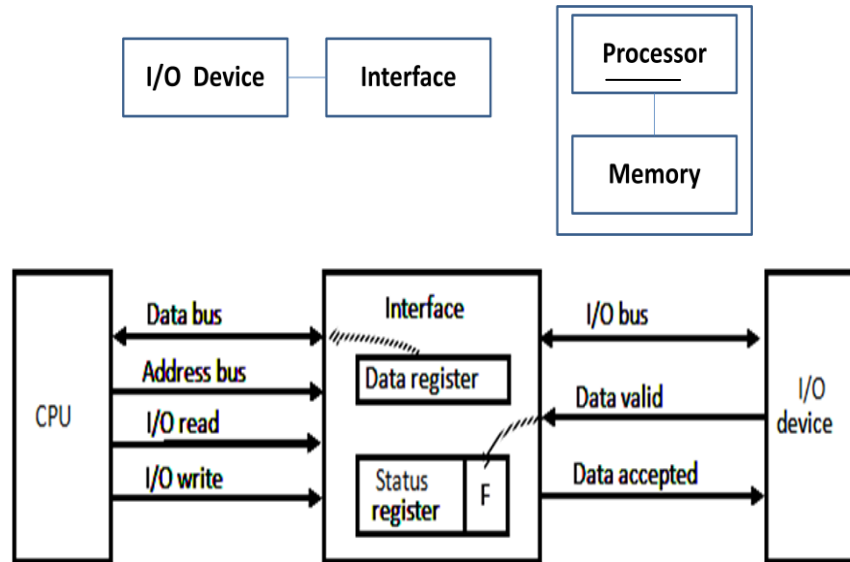
Q-30 List modes of data transfer and explain modes of data transfer.

Ans 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)

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 disable 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 on IC)
- Then Interface disable the data accepted line and the device can transfer the next data byte.

Interrupt Initiated I/O

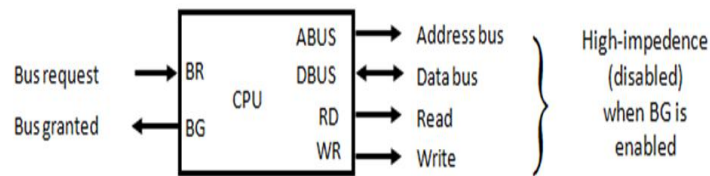
- In programmed initiated, CPU stays in a program loop until the I/O unit indicates that it is ready for data transfer.
- It keeps the processor busy without need.
- It can be avoided by using an interrupt facility.
- When the data are available from devices, interface issues an interrupt request signal.
- 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.

DMA(Direct Memory Access)

- 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.

Q- 30 A) Explain DMA(Direct Memory Access)

- 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.
- 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.

Q-31 What do you mean by Asynchronous data transfer? Give the methods of Asynchronous data transfer.

Ans:

- **Asynchronous data transfer :**
Data transfer between two independent units, where internal timing in each unit is independent from the other. Such two units are said to be asynchronous to each other.
- **Methods of Asynchronous data transfer.**

- 1) **Strobe Control method**
- 2) **The handshake method**

Strobe Control method:

- The Strobe control method of asynchronous data transfer employs a single control line to time each transfer.

The handshake method:

- The handshake method solves the problem of Strobe method by introducing a second control signal that provides a reply to the unit that initiates the transfer.
