



Q.1.a) Ans

S.N.	Computer Architecture	Computer Organization
1)	Computer architecture refers to those attributes of a system that have direct impact on logical execution of a program.	Computer organization refers to the operational units and their interconnections that realize architectural specification.
2)	It acts as an interface between hardware and software.	It deals with the units of a connection in a system.
3)	It helps us to understand the functionalities of a system.	It tells us how exactly all the units in a system are arranged and interconnected.
4)	Eg:- instruction sets, memory addressing techniques, I/O mechanism, etc.	Eg:- control signals, interfaces between computer and peripherals, memory technology being used, etc.

The future trends in computer system are explained as below:-

- Artificial Intelligence and Machine Learning:
Involves the development of systems capable of

performing tasks that normally require human intelligence, while machine learning is a subset of AI focused on the ability of machines to learn from data.

b) Internet of Things (IoT):

IoT involves connecting everyday objects to the Internet, allowing them to send and receive data.

c) Augmented Reality (AR) and Virtual Reality (VR):

AR overlays digital information onto the real world, while VR immerses users in a completely digital environment, helping in training and education and also enhancing gaming experience.

Q.1.b) Ans

The uses of shift micro-operations in computer arithmetic are:-

- 1) Left shifts multiply a number by 2 per shift position.
- 2) Right shifts divide a number by 2 per shift position.
- 3) Maintain the sign in signed binary numbers with arithmetic shifts.

- 1) Adjust bit positions for data alignment.
- 5) Create bit masks for setting or clearing specific bits.
- 6) Perform circular shifts for efficient bitwise operations.
- 7) Normalize floating-point numbers.
- 8) Test specific bits by shifting them to designated positions.

A micro operation specifies an operation whose result is stored typically in a register or memory location. The operation can be copying data from one register to another or adding the contents of two registers and storing the result in third register.

The register transfer micro operations between two 4-bit register via bus connection can be explained as below:-

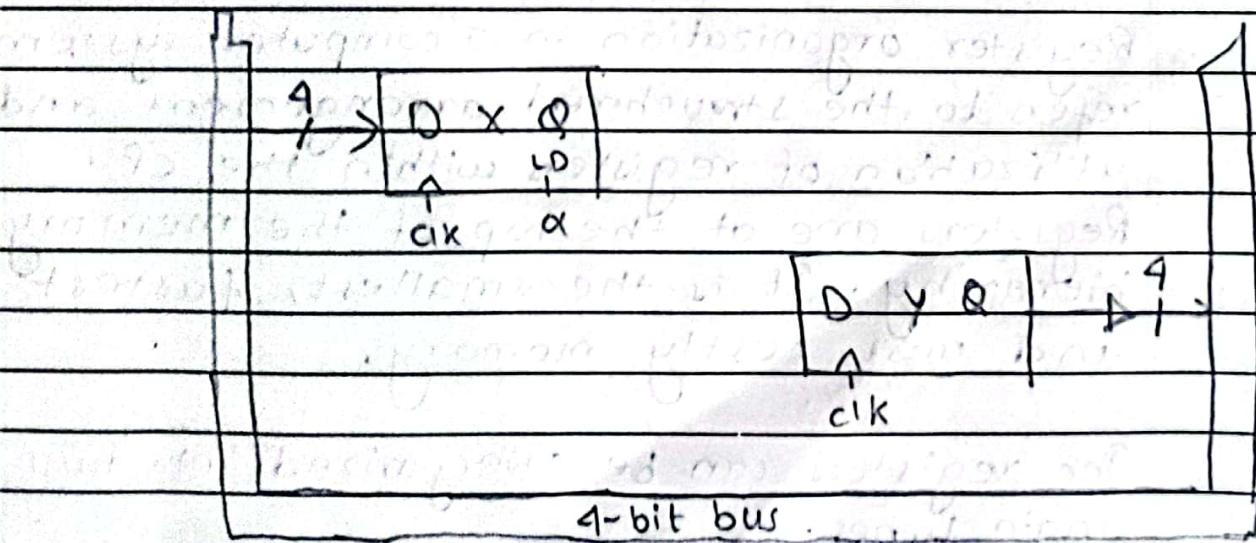


Fig:- Using Bus.

When data transfer takes place between register of same size and between corresponding bits, it is not necessary to specify individual bits i.e,

$\alpha: x \leftarrow y$ can be written,

if $x_3 \leftarrow y_1$, $x_2 \leftarrow y_2$, $x_1 \leftarrow y_1$, and $x_0 \leftarrow y_0$

- 1) User visi
- a) General
- for vari
- hold o
- addressi
- b) Data:

for individual bits to be transferred, x_3, x_2, x_1, x_0 can be written as $x[3-0]$ or $x(3:0)$

Eg:-

if $x_3 \leftarrow y_2$, $x_2 \leftarrow y_1$, $x_1 \leftarrow y_0$, $x_0 \leftarrow y_3$

then,

$\alpha: x(0:3) \leftarrow y(2:0)$

$\beta: x(0) \leftarrow y(0)$

- c) Address
- d) Conditi

Q.2-a) Ans

Register organization in a computer system refers to the structured arrangement and utilization of registers within the CPU.

Registers are at the top of the memory hierarchy. It is the smallest, fastest and most costly memory.

The registers can be categorized into two main types.

- 2) Contro
- a) Progr

uction

- b) Instr

used

1) User visible registers:

a) General purpose:

These registers can be used for variety of functions. They can be used to hold operand. They can also be used for addressing functions.

b) Data:

These registers are used to hold data and are not used for the calculation of an operand address.

c) Address:

These are used for addressing modes.

d) Condition codes:

Conditional codes (flags) are bit set by CPU hardware as the result of operation. Condition codes are collected in one or more register.

2) Control and status registers:

a) Program Counter (PC):

Contains address of an instruction to be fetched.

b) Instruction Register (IR):

Contains most recently used instructions.

ctd

c) Memory Address Register (MAR) :

Contains

address of a location in memory.

d) Memory Buffer Register (MBR) :

Contains

a word of data to be written to memory or
the most recently used word.

e) Program Status Word (PSW) :

Contains status

information. PSW contains condition codes
plus other status informations.

Q.2.b) Ans

Booth's algorithm is a multiplication off
algorithm used for multiplication of
two number in 2's complement notation.

Given:

$(-11)_{10}$ and $(13)_{10}$. $M = -11$ and $Q =$

The algorithm i.e, Booth's algorithm used is :-

- i) Multiplier is stored in Q-register, and multiplicand in M.
- ii) Q-1 is 1-bit register and the result is stored in A and Q.

- Date _____
Page _____
- iii) Initially A and Q_{-1} are zero.
 - iv) Bits are checked in $Q_0 Q_{-1}$.
 - v) If 11 or 00 occurs, then bits of AQ and Q_{-1} are shifted to 1-bit right.
 - vi) If 10 or 01 occur, then multiplicand is subtracted or added from A depending upon whether the bits are 01 and 10 [$01 = A \leftarrow A + M$]
 $10 = A \leftarrow A - M$. After this, shift is performed 1-bit right in AQ Q_{-1} .
 - vii) shifting is such that left most bit of A ie, A_{n-1} is shifted to A_{n-2} remains in A_{n-1} in both cases i.e., arithmetic shift.

A	Q	Q_{-1}	M
00000	01101	0	10101
01011	01101	0	10101 [A $\leftarrow A - M$]
00101	10110	1	10101 [shift]
11010	10110	1	10101 [A $\leftarrow A + M$]
11101	01011	0	10101 [shift]
01000	01011	0	10101 [A $\leftarrow A - M$]
00100	00101	1	10101 [shift]
00010	00010	1	10101 [shift]
10111	00010	1	10101 [A $\leftarrow A + M$]
11011	10001	0	10101 [shift]

Result = AQ

$$= (1101110001)_2, [-193]_{10}$$

Ans

Q.3. b) Ans

Single

~~One~~ address, two address and variable format differ from each other as:

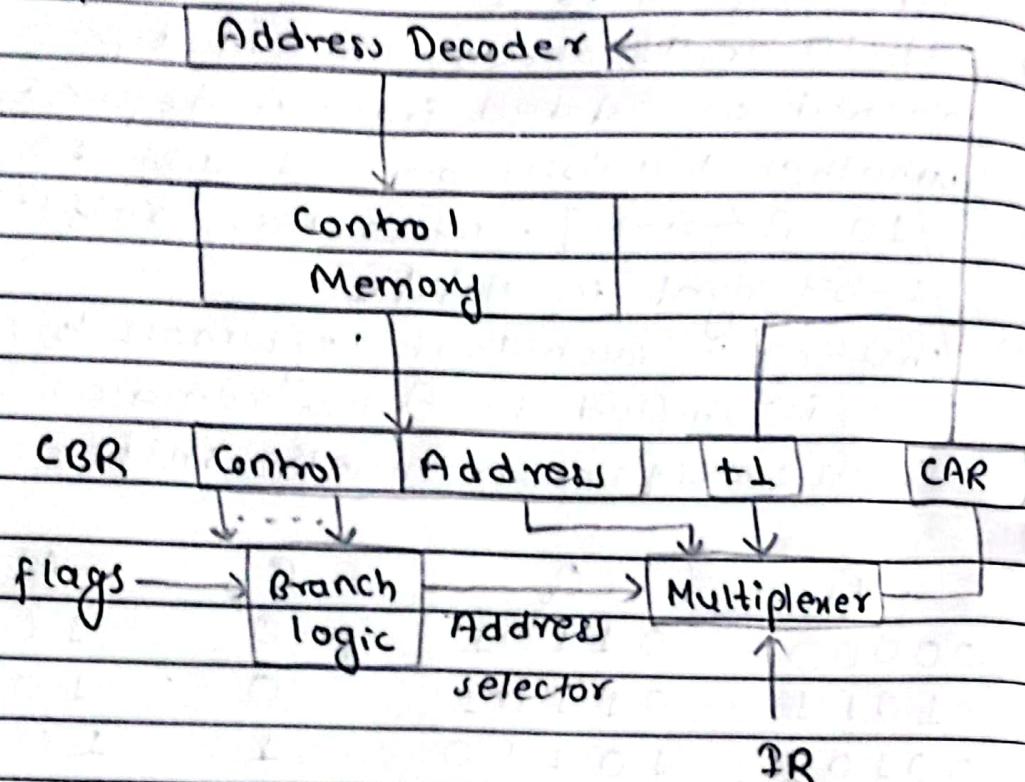


Fig:- Single address format.

With single address field the option of next address are as follows:

- i) address field
- ii) instruction field
- iii) next sequential address.

The address selection signal determines which option is selected. This approach reduces the no. of address field to one.

ctd
→

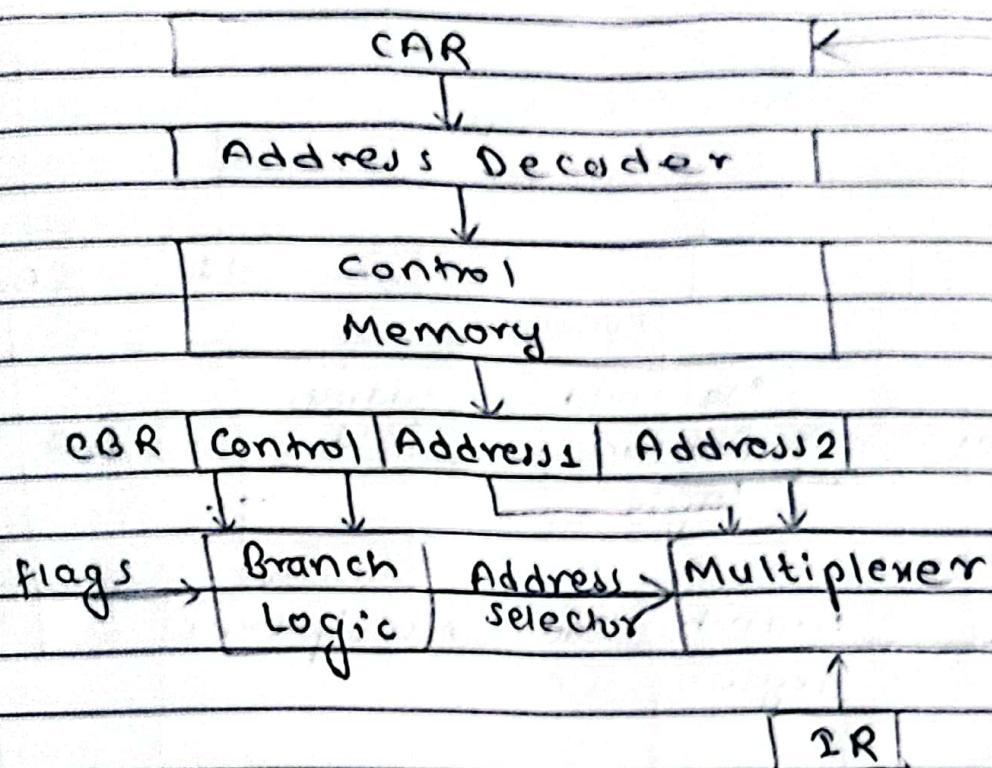


fig:- Two Address field format

This is the simplest approach. There is two address field in each microinstruction. A multiplexer is provided that serves as a destination for address field and instruction register. Based on the address selection input, multiplexer transmits either opcode or one of two addresses to the CAR. CAR is decoded to produce next microinstruction address. Address selection signals are provided by branch logic module whose inputs are from flags and control part of the CBR.

Ch

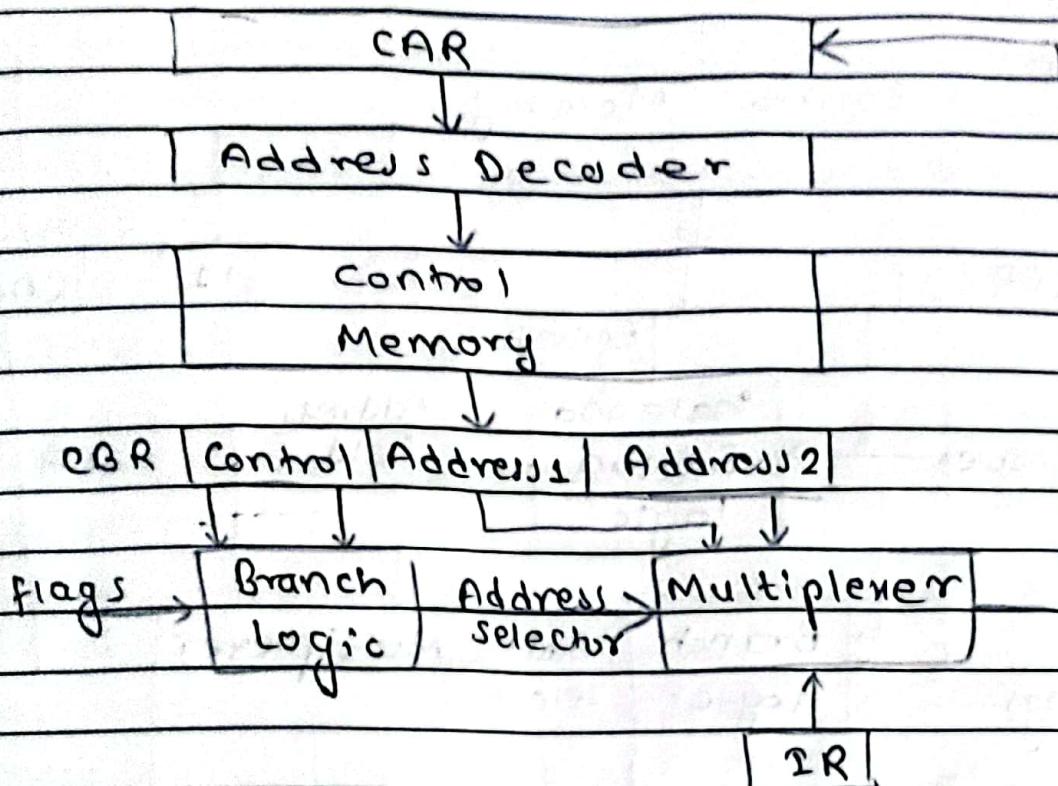


fig:- Two Address field format

This is the simplest approach. There is two address field in each microinstruction. A multiplexer is provided that serves as a destination for address field and instruction register. Based on the address selection input, multiplexer transmits either opcode or one of two addresses to the CAR. CAR is decoded to produce next microinstruction address. Address signal selection signals are provided by branch logic module whose inputs are from flags and control part of the CBR.

CS

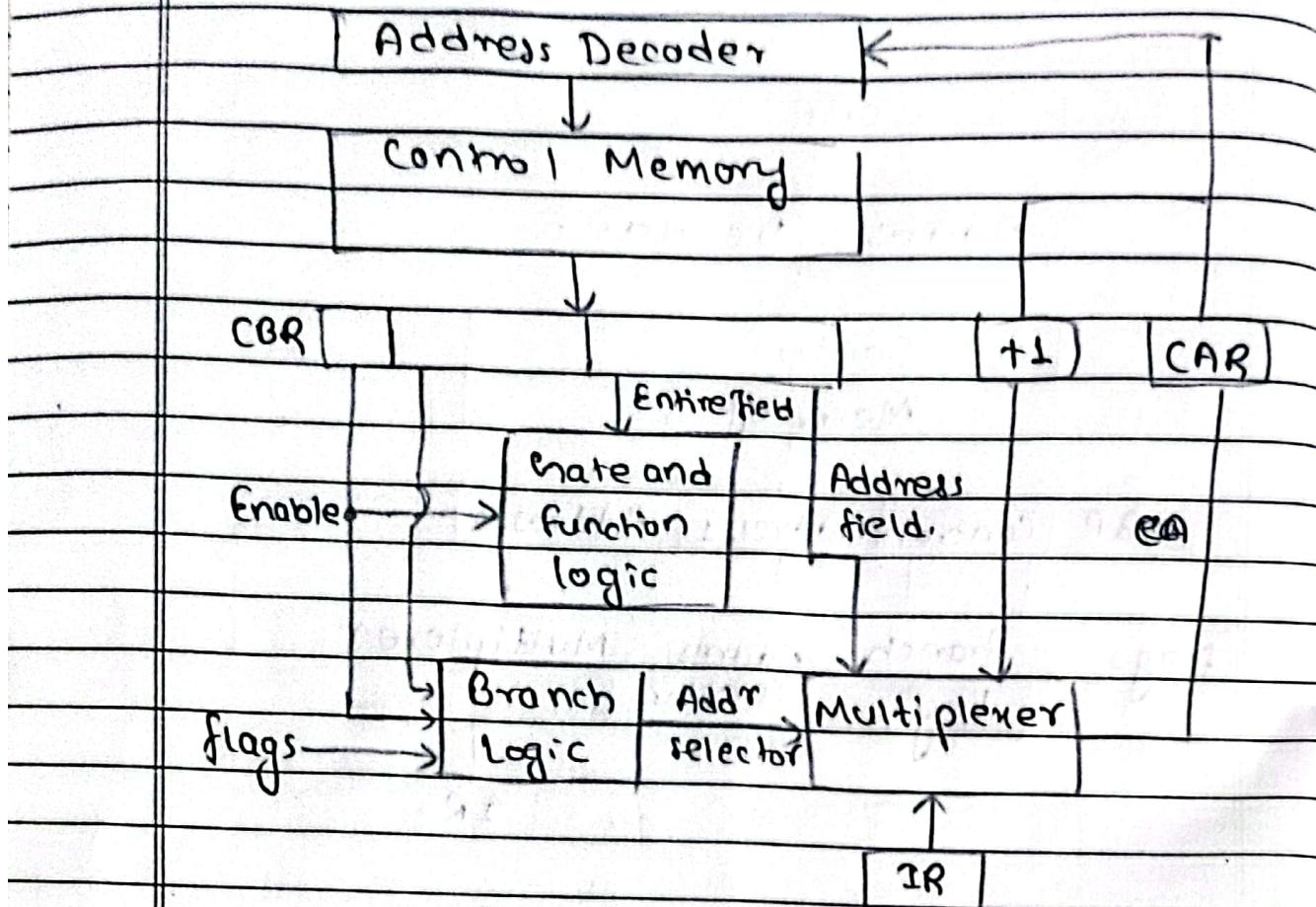


fig:- Variable format .

Another approach is to provide two different microinstruction format. Here one bit decides which format is being used. In one format, remaining bits are used to activate control signals. In other format, some bits derive the branch logic and remaining bits provide the address. In first format, next addr is either next sequential addr or an addr by IR. In second format, either a conditional or unconditional branch is specified.

Q.1(a) Ans

The transformation of data from main memory to cache memory is referred as mapping process. Three types of mapping techniques are:

- i) Direct mapping
- ii) Associative mapping
- iii) Set-Associative mapping

The direct mapping techniques can be explained as below:-

In direct mapping, each main memory block is assigned to a specific line in the cache.

Mapping is expressed as:

$i = j \text{ modulo } C$, where i is the cache line number assigned to main memory block j .

If $M = 64$, $C = 4$

line 0 can hold blocks 0, 4, 8, 12, ...

line 1 can hold blocks 1, 5, 9, 13, ...

line 2 can hold blocks 2, 6, 10, 14, ...

line 3 can hold blocks 3, 7, 11, 15, ...

Direct mapping cache treats a main memory address as 3 distinct fields:

- 1) Tag identifier: The tag is stored in the cache along with the data words of the line.
- 2) Line number identifier: line identifier specifies the physical line in cache that

will hold the referenced address.

- 3) Word identifier: Word identifiers specifies the specific word in a cache line that is to be read.

for every memory reference that the CPU makes, the specific line that would hold the reference is determined. The tag held in that line is checked to see if the correct block is in the cache.

The pros and cons of direct mapping are:-

Advantages (Pros):

- i) Easy to implement.
- ii) Relatively inexpensive to implement.
- iii) Easy to determine where a main memory reference can be found in cache.

Disadvantages (Cons):

- i) Each main memory block is mapped to a specific cache line.
- ii) Through locality of reference, it is possible to repeatedly reference to blocks that map to the same line number.
- iii) These blocks will be constantly swapped in and out of cache, causing the hit ratio to be low.

Qd

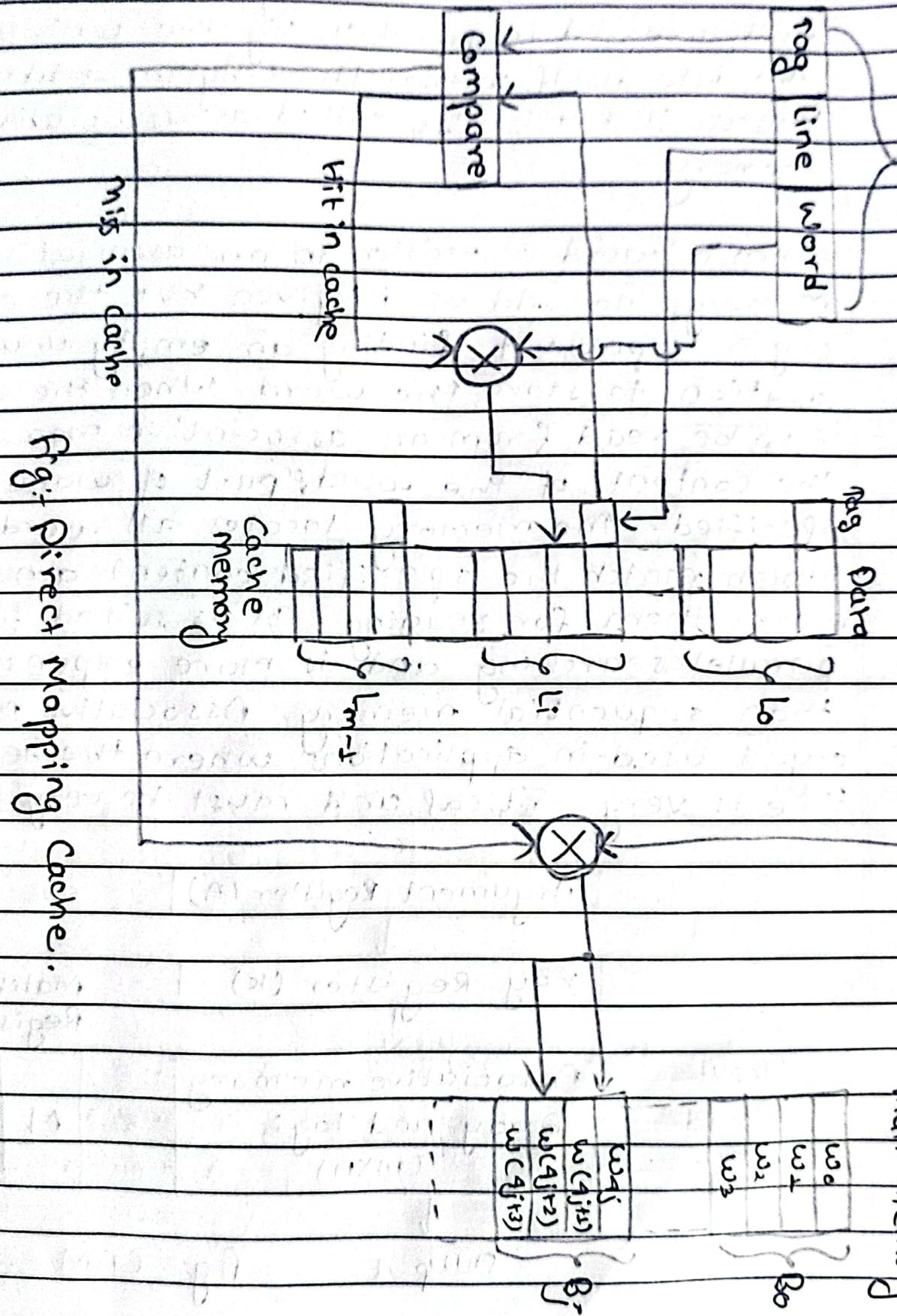
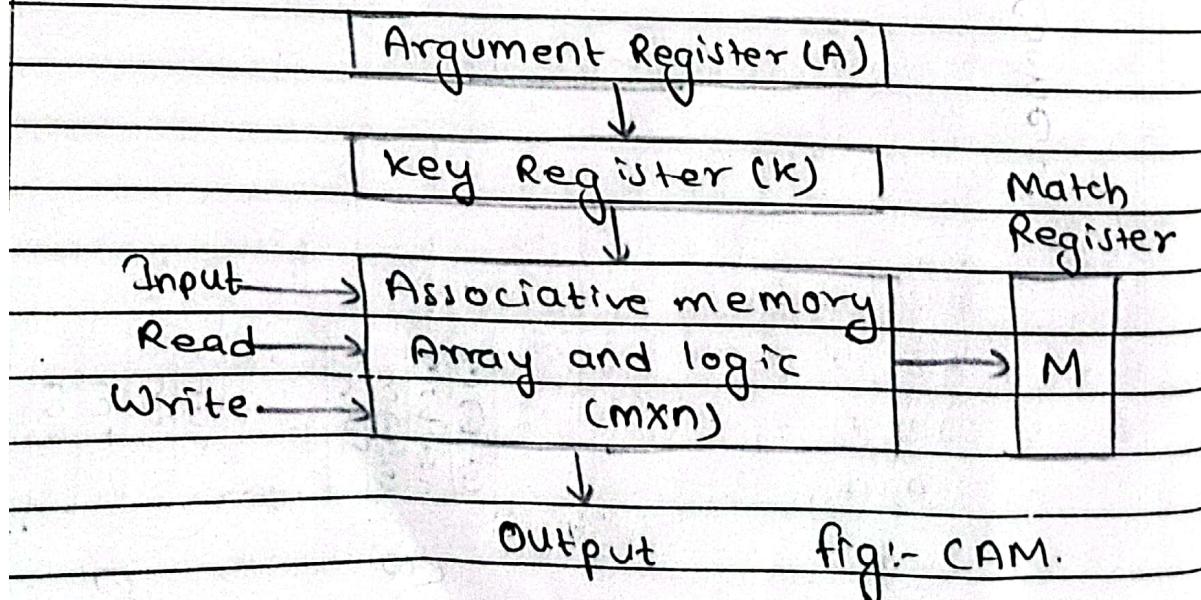


Fig: Direct Mapping Cache.

Q.4.b) Ans

Content addressable memory is a memory unit accessed in parallel by the content of the data itself rather than by an address. Hence it is also called as associative memory.

When a word is written in an associative memory, no address is given but the memory is capable of finding an empty unused location to store the word. When the word is to be read from an associative memory, the content of the word, or part of word is specified. The memory locates all words which match the specified content and marks them for reading. It is suited for parallel searching and is more expensive than sequential memory. Associative memory is used in applications where the search time is very critical and must be very fast.



The matching logic in CAM can be explained as :-
address

CAM consists of a memory array and logic for ' m ' words with ' n ' bits per word. The argument register 'A' and key register 'K' each have ' n ' bits, one for each bit of a word. The match register 'M' has m bits, one for each memory word. Each word in memory is compared in parallel with the content of argument register. The words that match bits of argument register set a corresponding bit in ~~the~~ the words whose corresponding bits in the match register have been set.

The key register provides a mask for choosing a particular field or key in argument word. The argument is compared with each memory word if key register contains all 1's. Otherwise, only those bits that have 1's in their corresponding position of key register are compared. Thus, key provides a mask for identifying piece of information.

Q. 5. a) Ans

Pipelining is a technique of decomposing a sequential process into sub operations, with each sub process being executed in a special dedicated segment that operates concurrently.

with all other segments.

It is a collection of processing segments through which binary information flows. Each segment processes partially the partitioned task. The result from each segment is transferred to next segment in pipeline. The final result is obtained after the data have passed through all segments.

The differences between RISC and CISC pipelining are:-

RISC pipelining	CISC pipelining
It uses a small, fixed-length instruction set.	It uses a large, variable-length instruction set.
Instructions are simple and typically execute in a single cycle.	Instructions can be complex and may require multiple cycles.
Decoding is straightforward due to uniform instruction size.	Decoding is more complex due to variable instruction length.
The control unit is simpler and focused on optimizing throughput.	The control unit is more complex to manage diverse instruction formats.

5) Often has more pipeline stages due to simpler instructions.	Typically has fewer pipeline stages but handles more complex instructions.
6) Data hazards are managed with techniques like forwarding and stalls.	Data hazards are more complex due to varied execution times.
7) Execution time per instruction is consistent, leading to higher performance.	Execution time varies, which can impact pipeline efficiency.

Q. S.b) Ans

Register renaming is a technique used in computer architecture to eliminate hazards and improve instruction-level parallelism in pipelined processors. It is used to add flexibility to the idea of register windowing.

Register window is beneficial when CPU calls a sub-routine. When calling process takes place, processor activates register window is moved down one position.

Register window handles subroutine call and return inside CPU itself in RISC processor as

explained below:-

Let us consider a CPU with 48 registers with 3 windows. Each window have 16 registers and a overlap of 4 registers between windows.

window pointer register (first window active)	100 ↴ 1000	param#1 param#2 param#3	12 13 14	0 } 1 } window #1 2 } 3 } param#4 4 } param#5 5 } param#6 6 } param#7 7 } param#8 8 } param#9 9 } param#10 10 } param#11 11 } param#12 12 } param#13 13 } param#14 14 } param#15 15 } param#16 16 }
window mask register (only first window has valid data)	1100		15 16 17	
(a)			17	

window pointer register (second window active)	101 ↴ 1100	param#1 param#2 param#3 Result	12 13 14 15	0 } 1 } window #1 2 } 3 } param#4 4 } param#5 5 } param#6 6 } param#7 7 } param#8 8 } param#9 9 } param#10 10 } param#11 11 } param#12 12 } param#13 13 } param#14 14 } param#15 15 } param#16 16 }
window mask register (first two windows have valid data)			16 17	
(b)			17	

window pointer

register 00
(first window active)

window 1000
mask register
(only first window have valid data)

	Result	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		window 1/2								window 1/2									

(c)

Initially, CPU is running on the first window as in fig(a). Here, a subroutine is called and three parameters are to be passed. CPU stores these parameters in three of overlapping registers and calls the subroutine. Subroutine accesses the parameter, calculates and result to be returned to the main program is again stored in the overlapping register. Now second window is deactivated and CPU works with first window as shown in fig(c). Register windowing is not linear but circular. The last window overlaps window 1.

Q-6-a) Ans

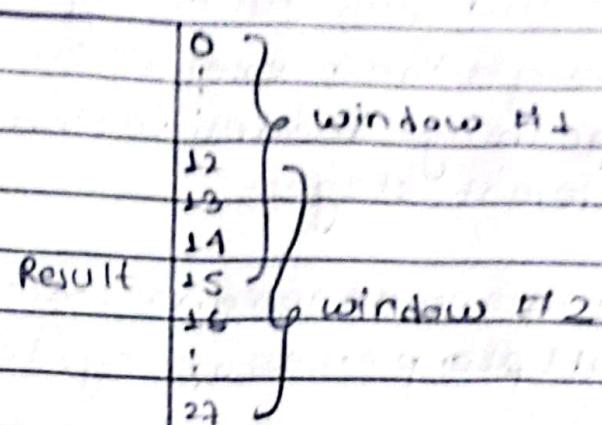
Parallelism occurs in uniprocessor through SLP and pipelining.

SLP allows multiple instructions to be

window pointer

register 00
(first window
active)

window mask
register
(only first window
have valid data)



(c)

Initially, CPU is running on the first window as in fig(a). Here, a subroutine is called and three parameters are to be passed. CPU stores these parameters in three overlapping registers and calls the subroutine. Subroutine accesses the parameter, calculates and result to be returned to the main program is again stored in the overlapping register. Now second window is deactivated and CPU works with first window as shown in fig(c). Register windowing is not linear but circular. The last window overlaps window 1.

Q-6-a) Ans

Parallelism occurs in uniprocessor through SLP and pipelining.

SLP allows multiple instructions to be

executed simultaneously by overlapping through their execution phases, while pipelining divides instruction execution into different stages.

The interconnections structure possible in multiprocessor system are:-

1) Time shared common bus:

A common bus multiprocessor system consists of a number of processor connected through a common path to memory unit. Only one processor can communicate with the memory at any given time.

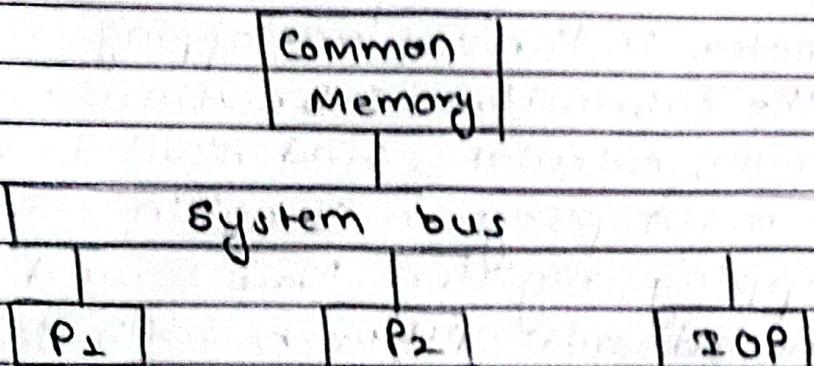


Fig:-Time shared common bus.

2) Multiport memory:

A multiport memory system employs separate buses between each memory module and each CPU. Each processor bus is connected to each memory module.

Ctd
→

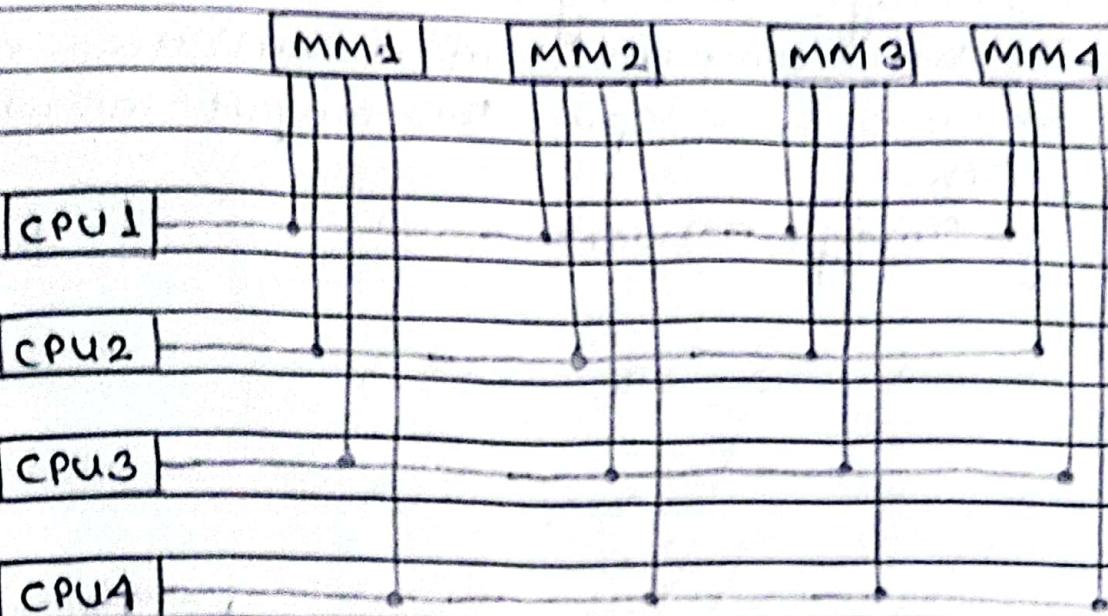


Fig:- Multiport Memory

3) Crossbar Switch:

The crossbar switch organization consist of a number of cross points that are placed at intersections between processor bus and memory module paths. Each switch determines the path from a processor to memory module.

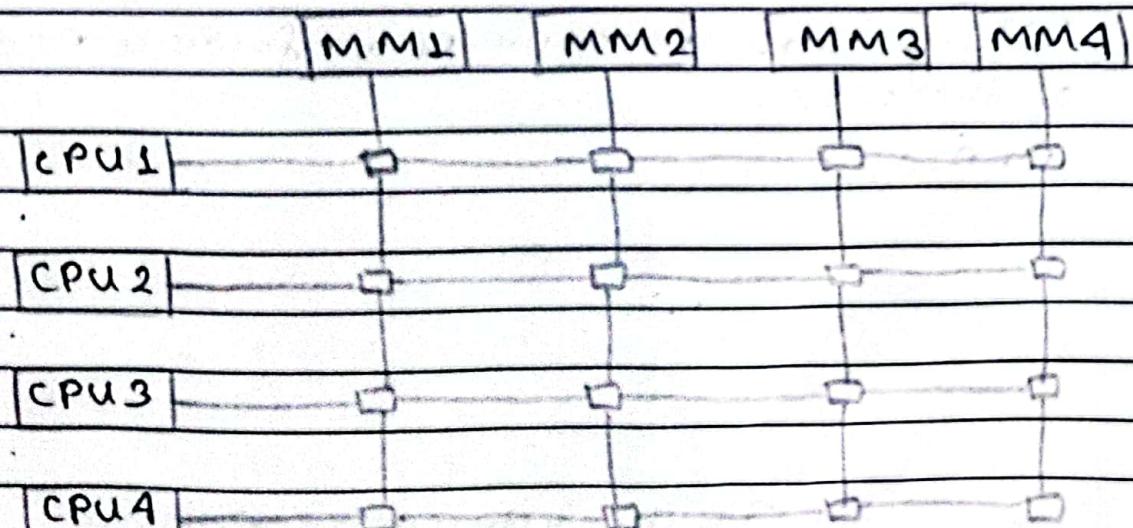


Fig:- Crossbar switch.

1) Multistage switching network:

The basic component of a multistage network is a two-input, two-output interchange switch.

Eg:- Omega network.

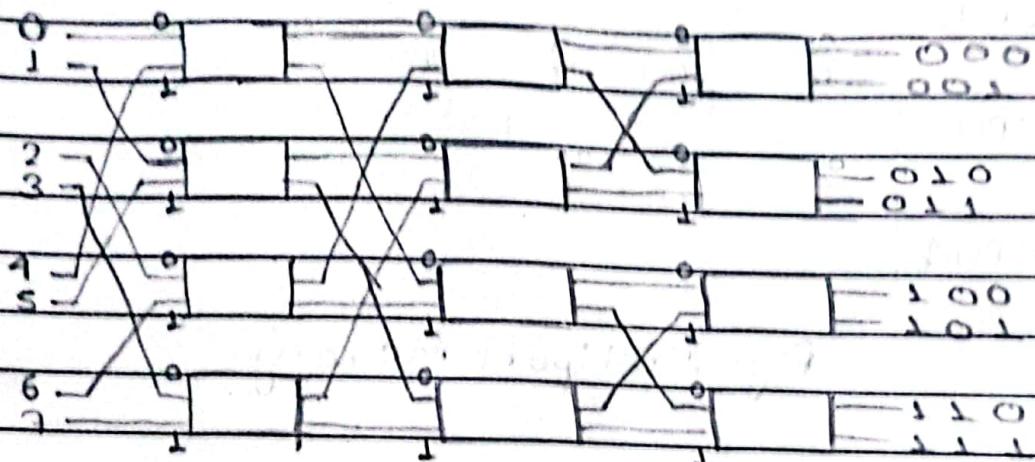


Fig:- Omega switching Network

5) Hypercube interconnection:

The hypercube or binary n -cube multiprocessor structure is loosely coupled system composed of $N = 2^n$ processor in an n -dimensional binary cube. Each processor forms a node of the cube.

for $n=3$;

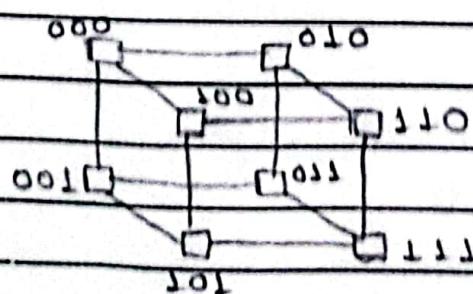


Fig:- 3-cube interconnection.

Q. 6.b) Ans.

The hardware issues performance issues that arise in multicore organization are :-

1) Increase in parallelism & pipelining :-

Instructions are executed through pipeline of stages so that while one instruction is executing in one stage of pipeline, another instruction is executing in another stage of pipeline. Simple three-stage pipelines were replaced by pipelines with five stages, and many more. There is a limit in increasing pipelining stages as increased stage require more logic, interconnections and control signal.

2) Alternative chip organization

Superscalar :-

Multiple pipelines are constructed replicating the resource. This enables parallelism in parallel pipeline. Performance can be increased by increasing number of parallel pipelines.

Simultaneous Multithreading (SMT) :-

Registers are replicated so that the multiple threads can share the use of pipeline resources. Managing multiple threads over a set of pipelines limits the number of threads and number of pipelines that can be effectively utilized.

3) Power Consumption:

As the number of transistors per chip and clock frequencies increased power requirements grown exponentially. Power density can be minimized by using multiple chips on cache memory as memory transistors are smaller and have a power density lower than that of logic transistors.

The software performance issues that arise in multicore organization can be explained as :-

Performance benefits of multicore organization depend on effective exploitation of parallel resources. Software suffers overhead as a result of communication, distribution of work to multiple processors and cache coherence overhead. This results in high performance at the beginning and goes on decreasing as burden of overhead of using multiple processors increases.

Q.7(a) Ans

In floating-point representation, numbers can be represented as;

$$\text{S} \times \text{F}$$

where,

$s \rightarrow$ Significand / Mantissa

$B \rightarrow$ Base

$e \rightarrow$ Exponent

Sign bit \rightarrow plus or minus.

The base B is implicit and need not to be stored because it is same for all numbers.

Radix point is right of the left most or MSB of significand i.e., there is one bit to the left of radix point.

Eg:-	sign	Biased Exponent	Significand / Mantissa.
	← 1-bit ←	8-bit →	23-bit →

fig:- 32-bit floating point representation.

The left-most bit stores sign of number (0 \rightarrow +ve, 1 \rightarrow -ve). The exponent value is stored in next 8-bits. This is known as biased representation.

A fixed value called the bias is subtracted from the field to get true exponent value. The bias equals $2^{k-1} - 1$, where k is the number of bits in binary exponent.

Significand is stored in normalized form.

A normalized number is one in which the most significant digit of the mantissa is non-zero.

Q.7.b) Ans

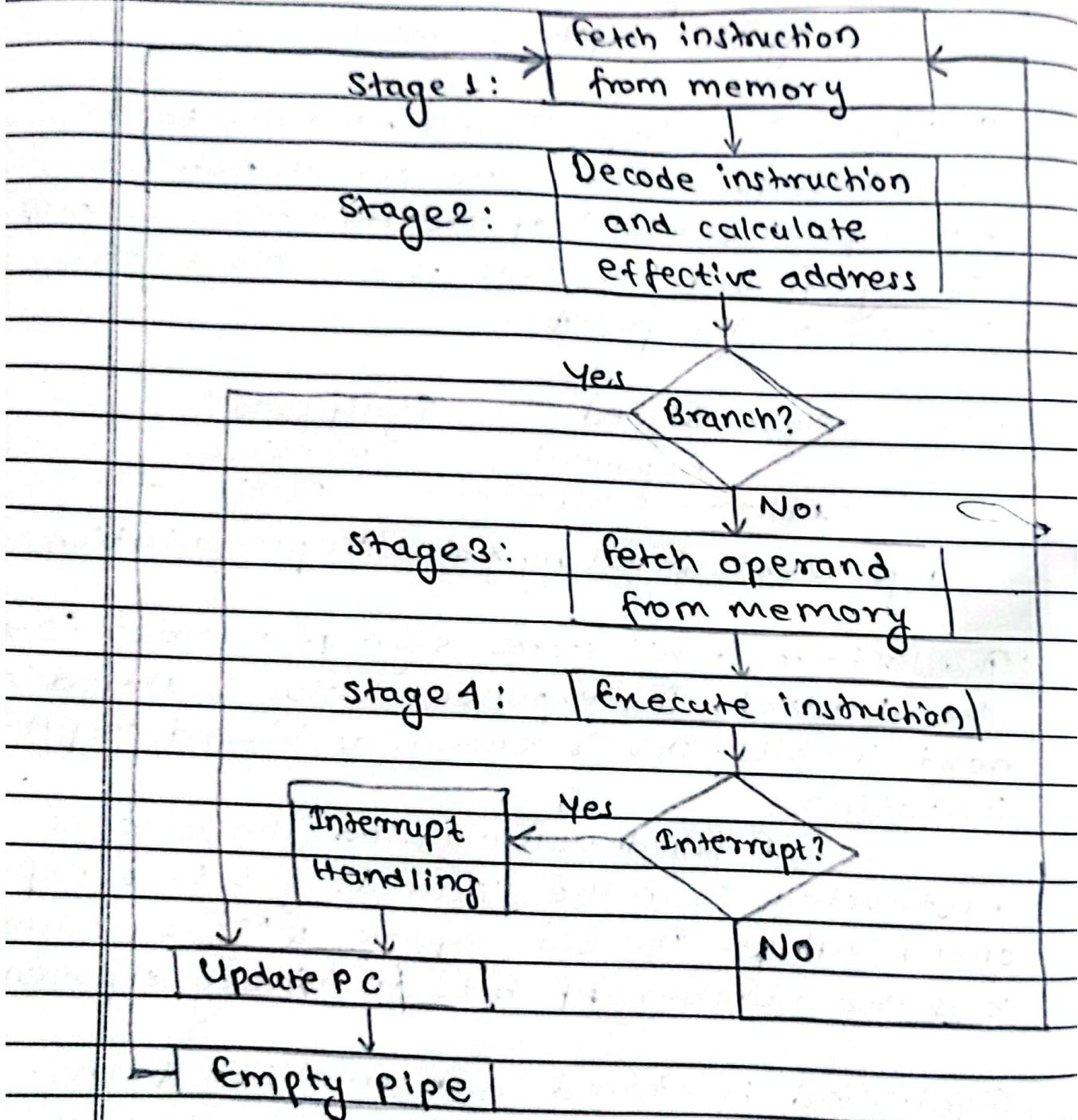


fig:- four stage instruction pipeline

An instruction pipeline reads consecutive instructions from memory while previous

instructions are being executed in other segment.

The sequences of steps to process an instruction in four-stage instruction pipeline are:-

- i) fetch the instruction from memory.
- ii) Decode instruction and calculate effective address.
- iii) fetch operand from memory.
- iv) Execute and store.

Q.3-a) Ans

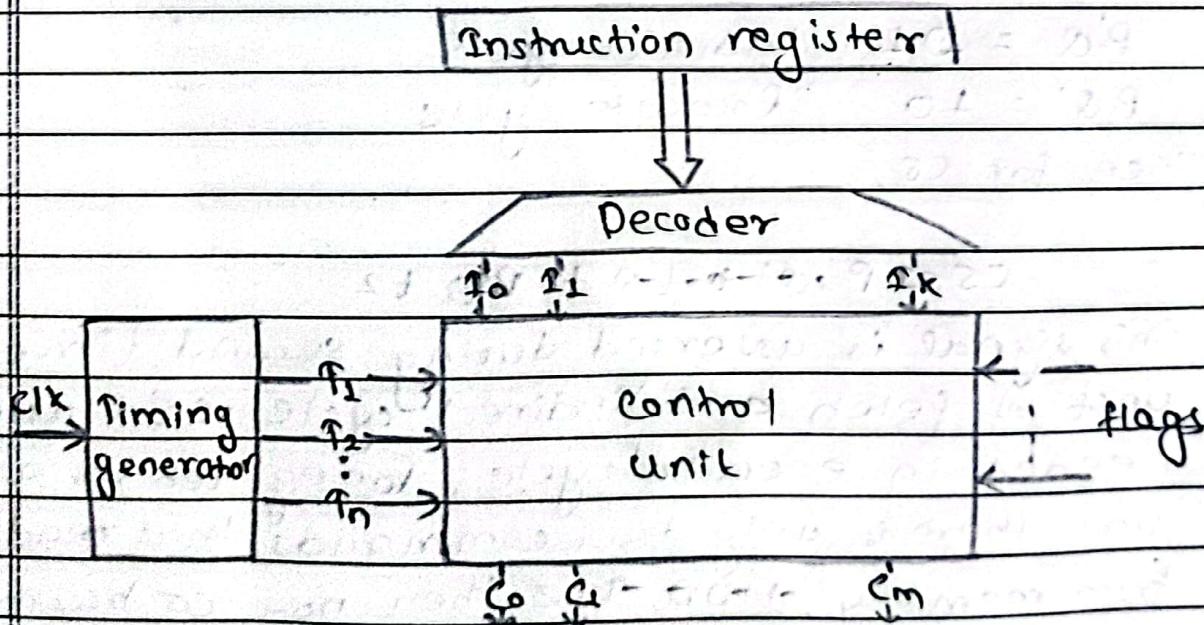


fig:- Hardwired control unit.

In hardwired implementation, control unit is combination of circuit. Its input signal is transferred

into set of output signals, which are control signals.

Let us discuss internal logic of control unit that produces output control signals as a function of input signals. A boolean expression is derived for each inputs.

Let us consider a single control signal CS from micro operations of instruction cycle. CS causes data to be read from memory into MBR. Let us consider two new control signals, P and Q with the following interpretation.

$P \cdot Q' = 00$ fetch cycle

$P \cdot Q = 10$ Indirect cycle

$P \cdot Q' = 10$ Execute cycle

Then for CS;

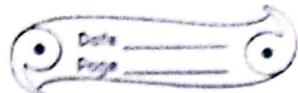
$$CS = P' \cdot Q' \cdot t_1 + P \cdot Q \cdot t_2$$

This signal is asserted during second time unit of fetch and indirect cycle. CS is also needed in execute cycle. For eg. let us assume there is only two commands that read from memory ADD, ISZ then now CS become

$$CS = P' \cdot Q' \cdot t_1 + P \cdot Q \cdot t_2 + P \cdot Q' (ADD + ISZ) \cdot t_2$$

This process is repeated for every control signal generated by processor, defining the logic of control unit.

2023 Spring.



Q.1.a) Ans [2022 Fall]

Computer architecture refers to those attributes of a system that have direct impact on logical execution of a program.

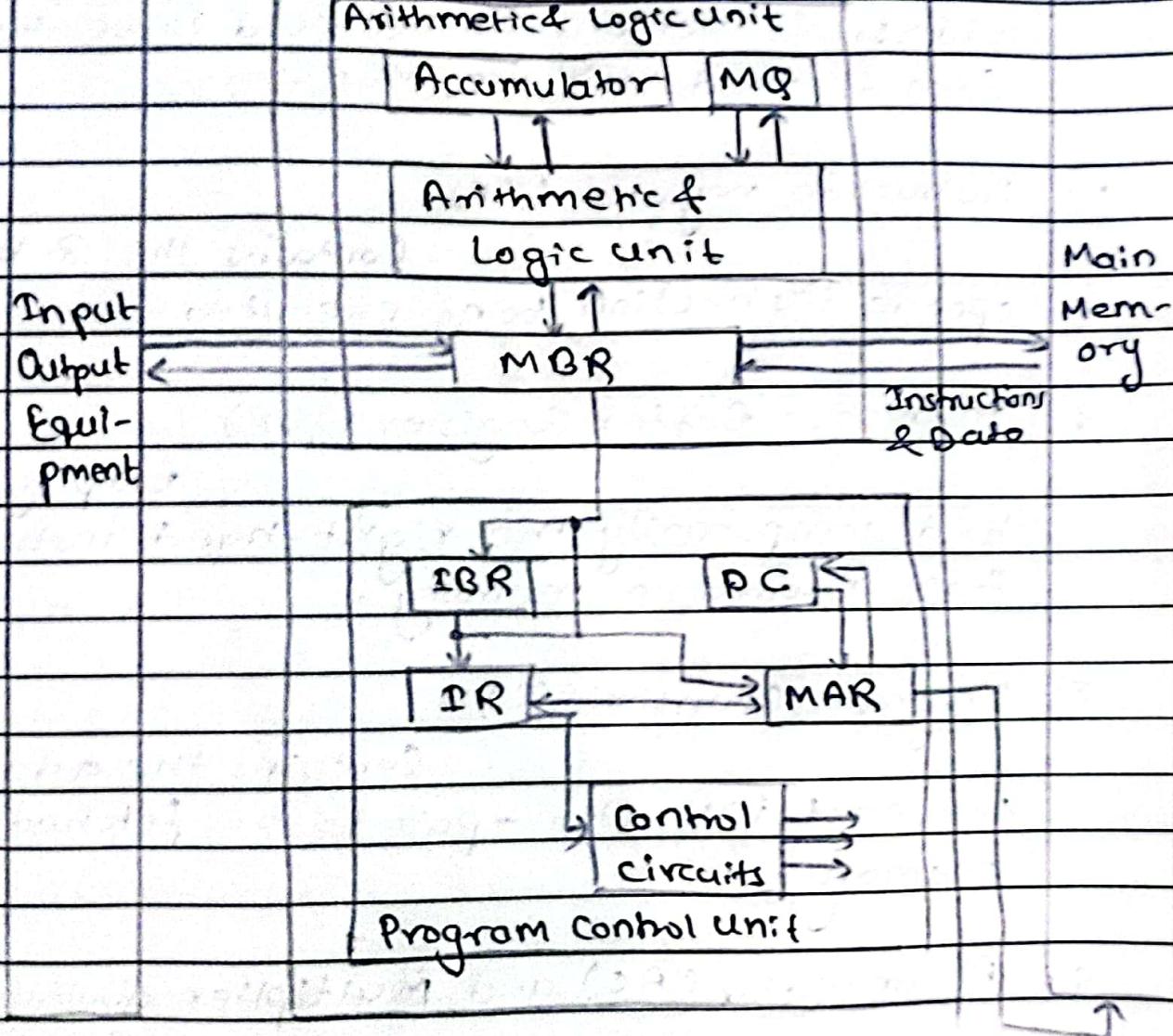


Fig:- Extended IAS structure

The components of extended IAS structure are explained as below:-

1) Memory Buffer Register (MBR) :

Contains a word to be stored in memory or sent to the I/O unit, it is used to receive a word from memory or from the I/O unit.

2) Memory Address Register (MAR) :

Specifies the address of memory in the word to be written from or read into the MBR.

3) Instruction register (IR) :

Contains the 8-bit opcode instruction being executed.

4) Instruction Buffer Register (IBR) :

Employed to hold temporarily the right-hand instruction from a word in memory.

5) Program Counter (PC) :

Contains the address of the next instruction-pair to be fetched from memory.

6) Accumulator (Ac) and Multiplier Quotient (MQ)

Employed to hold temporarily operands and results of ALU operations. For eg.: the result of multiplying two 40-bit numbers is an 80-bit number, the most significant 10 bits are stored in the A

and the least significant in the memory.

7) Central Processing Unit (CPU):

Controls the operation of the computer and performs its data processing functions, often simply referred to as processor.

8) Input Output:

Moves data between the computer and its external environment.

Q. 1.b) Ans

RTL stands for Register Transfer Language. It is a level of abstraction used in digital design and computer engineering to describe the behavior and functionality of a digital circuit or system.

The operation can be copying data from one register to another or adding data from one register to another or adding data from two registers and placing result in another register.

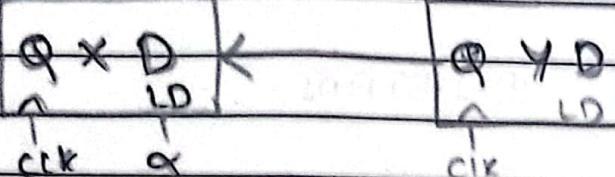
Data transfer operation could be either direct path or bus. Both are valid but system designer should take the best implementation. The set of 14-operation for a system is sufficient for data path design. The component connections are used for data

transfer but they do not provide the conditions for data transfer.

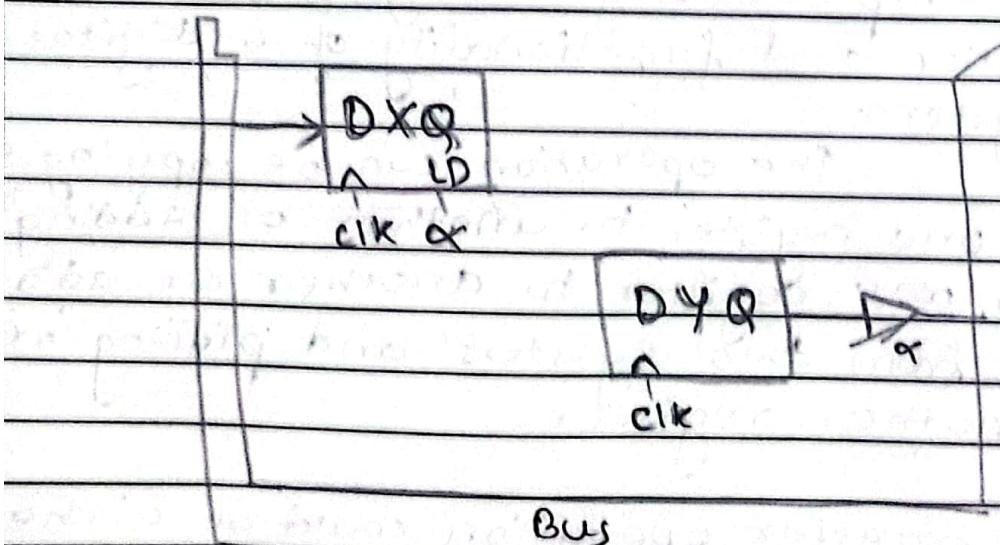
Let us assume a condn, the transfer to take place when control lfp α is high. The transfer could be written as:

If α , then $X \leftarrow Y$ [RTL]

or
 $\alpha: X \leftarrow Y$



direct path:



- Using Bus.

Fig:- Implementation of M-ops using hardware



In direct path, α is used to load register X.
In bus based, tri-state buffer is enabled
by α to place the contents of Y on bus.

Q. 2. a) Ans

The different types of registers used in computer system are:-

- 1) User visible registers
- a) General purpose:

These registers can be used for variety of functions. They can be used to hold operand. They can also be used for addressing functions.

- b) Data:

These registers are used to hold data and are not used for the calculation of an operand address.

- c) Address:

These are used for addressing modes.

- d) Conditional codes:

Conditional codes (flags) are bit set by CPU hardware as the result of operation. Condition codes are collected in one or more register.

2) Control and Status Registers

a) Program Counter (PC) :

Contains address of an instruction to be fetched.

b) Instruction Register (IR) :

Contains most recently used instructions.

c) Memory Address Register (MAR) :

Contains address of a location in memory.

d) Memory Buffer Register (MBR) :

Contains a word of data to be written to memory or the most recently used word.

e) Program Status Word (PSW) :

Contains status information. PSW contains condition codes plus other status informations.

The uses of registers in computer system are -

1) Data storage and Manipulation

2) Instruction Execution

3) Data Movement

4) Addressing

5) Control Flow

6) Performance Optimization

Q.2.b) Ans

SOLN;

Given

$$-15 \times 6$$

$$M = -15 \quad \text{and} \quad Q = 6$$

A	Q	Q-1	M
000000	00110	0	10001
000000	00011	0	10001 [shift]
01111	00011	0	10001 [AGA-M]
00111	10001	1	10001 [shift]
00011	11000	1	10001 [shift]
10100	11000	1	10001 [AGA-M]
11010	01100	0	10001 [shift]
11101	00110	0	10001 [shift]

Result = A ~~and~~ Q

$$= 1110100110 \quad (-90)$$

Ans

Q.3.a) Ans

The restoring division algorithm can be explained as below:-

- Divisor is loaded in M and dividend in A and Q register.

- Dividend must be expressed as two n-bit 2's complement number.
(If +ve, A = 0000, if -ve, A = 1111)
- Shift A(Q) left 1-bit position linearly.
- If M and A has same sign, $A \leftarrow A - M$
otherwise $A \leftarrow A + M$.
- Above operation is successful if the sign of A is same before and after operation.
i) If operation is successful, then set $Q_0 = 1$.
ii) If operation is unsuccessful, then $Q_0 = 0$ and restore previous value of A.
- Repeat above steps for the number of bits in Q.
- Remainder is in A.
- If the sign of divisor and dividend were same, then quotient is in Q, otherwise current quotient is Q's complement of Q.

Q.3.b) Ans

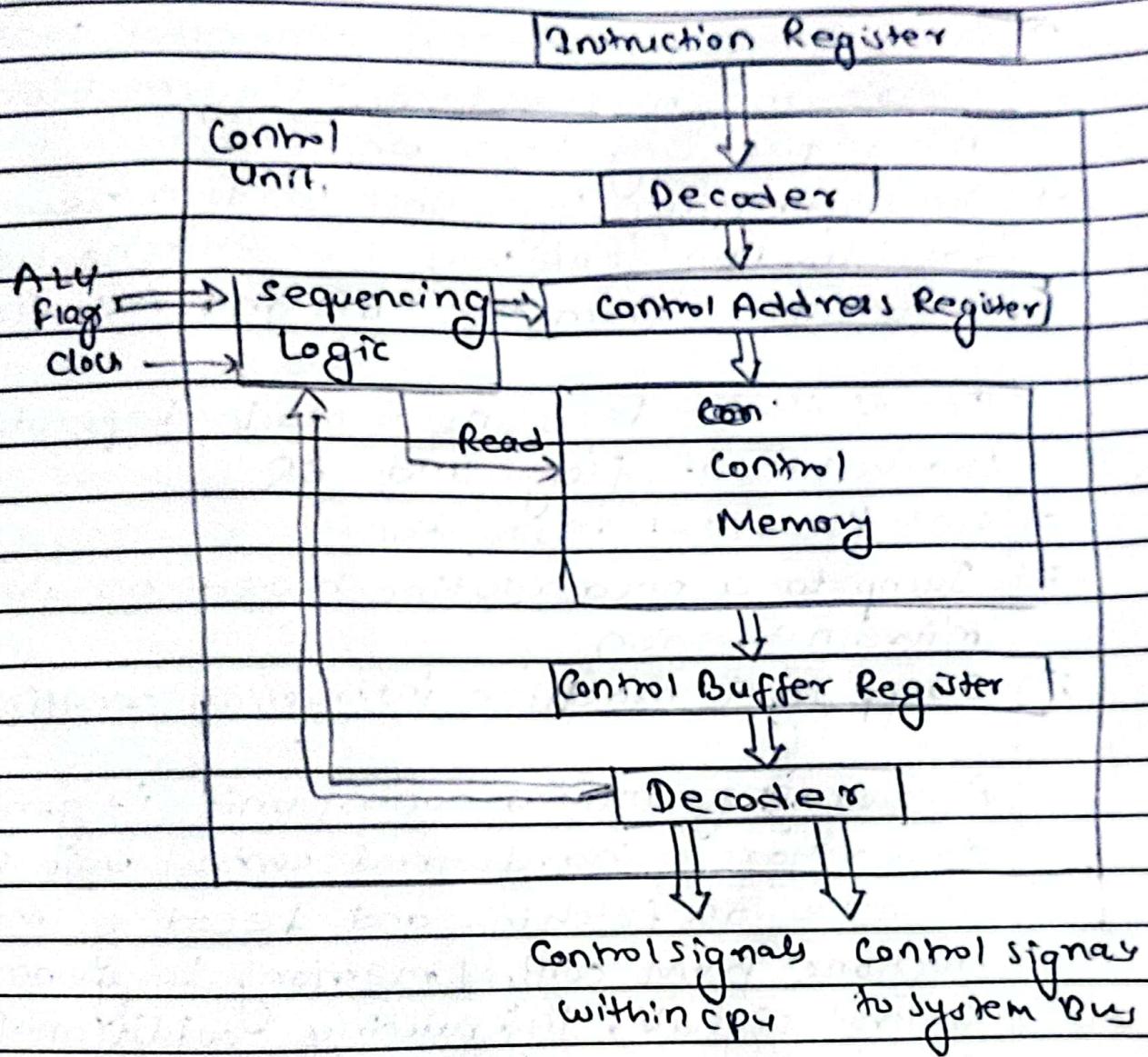


fig:- functioning of microprogrammed control unit.

The control unit functions as follows in a single clock pulse:

- To execute an instruction, the sequencing logic

unit issues a READ command to the control memory.

- The word whose address is specified in the CAR is read into the CBR.
- The content of the CBR generates control signals and next-address information for the sequencing logic unit.
- The sequencing logic unit loads a new address into the CAR based on the next address information from the CBR and ALU flags.

One of three decisions is made depending on the value of flags and CBR

- i) Get the next instruction.
- ii) Jump to a new routine based on jump microinstruction
- iii) Jump to a machine instruction routine.

A microprogrammed control unit is generally slower than a hardwired control unit because it relies on fetching and decoding microinstructions from control memory to generate control signals, introducing additional delay. In contrast, a hardwired control unit uses fixed combinational logic circuits that provide direct and immediate control signal generation, resulting in faster execution. While microprogrammed control units offer greater flexibility for modifying control logic, this comes at the cost of speed compared to the more streamlined and efficient hardwired approach.

sem

Department of BE.com

Submission Date : 1, M. June,
2024

Q.9.0 Ans.

Associative memory is a memory unit accessed in parallel by the content of the data itself rather than by an address. Hence, it is also called as ~~content addressable memory~~ content addressable memory.

Read Operation in Associative Memory:

If more than one word in memory matches the argument field, all the matched words will have '1' in the corresponding bit of position of match register. The bits of match register is scanned one at a time. The matched words are read in sequence by applying read signal to each word whose bit is '1'. In most cases, associative memory stores table with no identical items. Thus, no read signal is needed, output is connected directly to M_i .

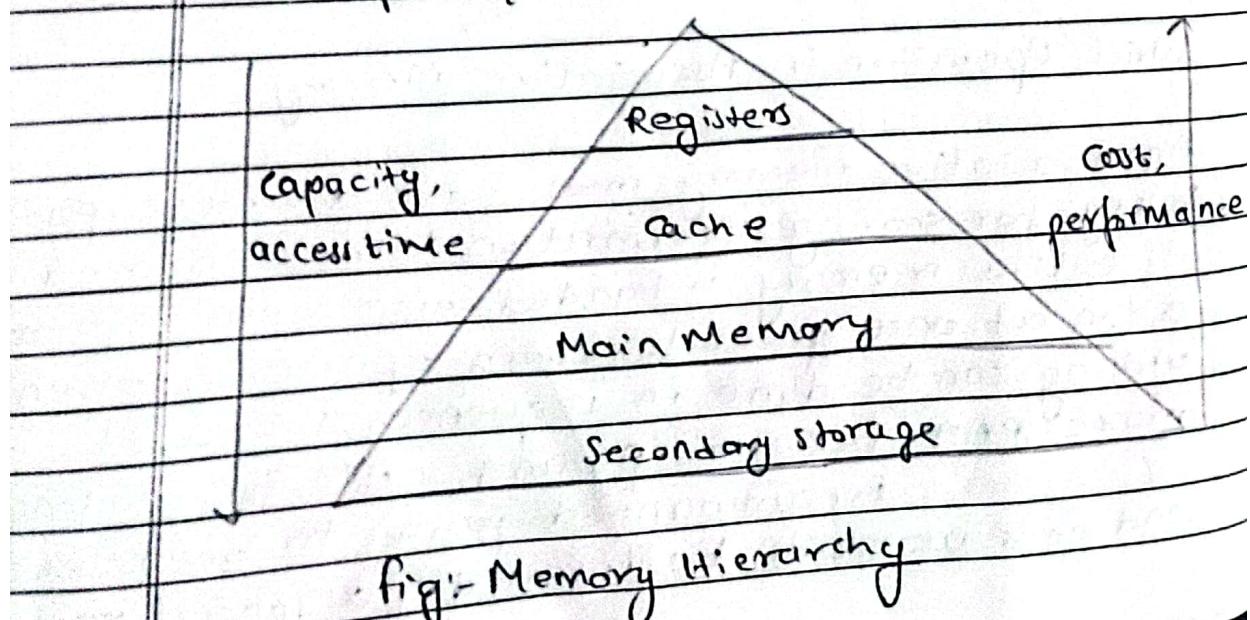
Write Operation in Associative Memory:

An associative memory must have a write capability for storing information to be searched. If entire memory is loaded with new information at once prior to search operation, then writing can be done in sequence. This will make RAM for writing and CAM for reading. For unwanted words to be deleted and new words to be inserted, a special register

is required to distinguish between active and inactive words. This register, also known as tag register would have as many bits as there are words in memory. All the active words are set to bit '1'. Word is cleared in memory by clearing the tag bit to '0'. Word is deleted by scanning the tag register until first '0' is encountered. This is the position to delete and insert new word.

Q.4.(b) Ans (2022 fall)

Memory hierarchy is a structure that optimizes the trade-off between speed and cost in a computer system. It organizes different types of memory and storage based on their access speeds, sizes, and costs.



Date _____
Page _____

The reasons that we need multilevel hierarchy are explained below :-

1) Speed vs cost :

Fast memory technologies (like cache) are very costly and have limited capacity, while slower technologies (like hard drives) are much cheaper and can store vast amount of data. A multilevel hierarchy balances these factors by using fast memory for critical, frequently accessed data and slower memory for larger, less frequently accessed data.

2) Performance :

High-speed memory levels, such as caches, store copies of frequently accessed data and instructions close to the CPU. This reduces the time it takes for the CPU to retrieve data, significantly improving overall system performance by minimizing delays caused by slower memory access.

3) Efficiency :

Different memory levels store data based on how often it's accessed, optimizing performance and access speed.

4) Cost Efficiency :

Combining fast, expensive memory with slow, cheap memory balances high performance with affordable costs.

5) Data Locality:

The hierarchy improves access times by storing frequently accessed data in faster memory, leveraging patterns in how data is used.

6) Scalability:

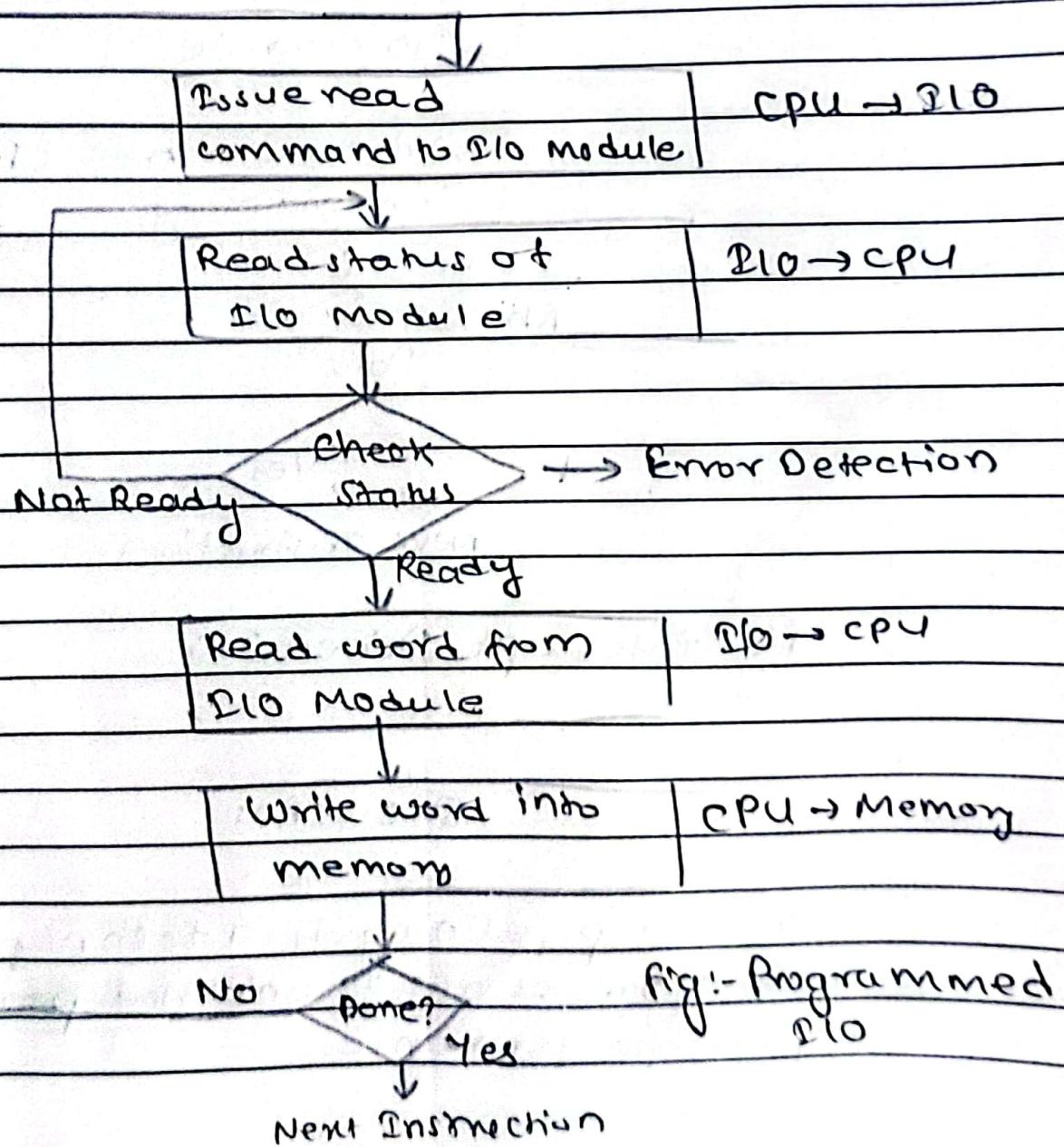
The hierarchy improves accommodates new memory technologies, allowing systems to integrate advancements without major overhauls.

Q.5(a) Ans

Interrupt driven I/O differ from programmed I/O as:

SN	Interrupt Driven I/O	Programmed I/O
1)	The I/O device signals the CPU when it is ready, causing the CPU to stop its current task and handle I/O.	The CPU continuously checks the status of the I/O device at regular intervals.
2)	The CPU can execute other instructions while waiting for an interrupt.	The CPU is actively engaged in checking the device status, which can lead to wasted cycles if the device is not ready.

- 3) Quick handling of I/O operations as the CPU responds to interrupts immediately.
- Slower handling of I/O operations due to constant polling.
- 4) Involves managing interrupts and context switching which adds overhead.
- Simpler implementation with less overhead.
- 5) Ideal for scenarios with time-sensitive I/O operations.
- Suitable for simpler systems.



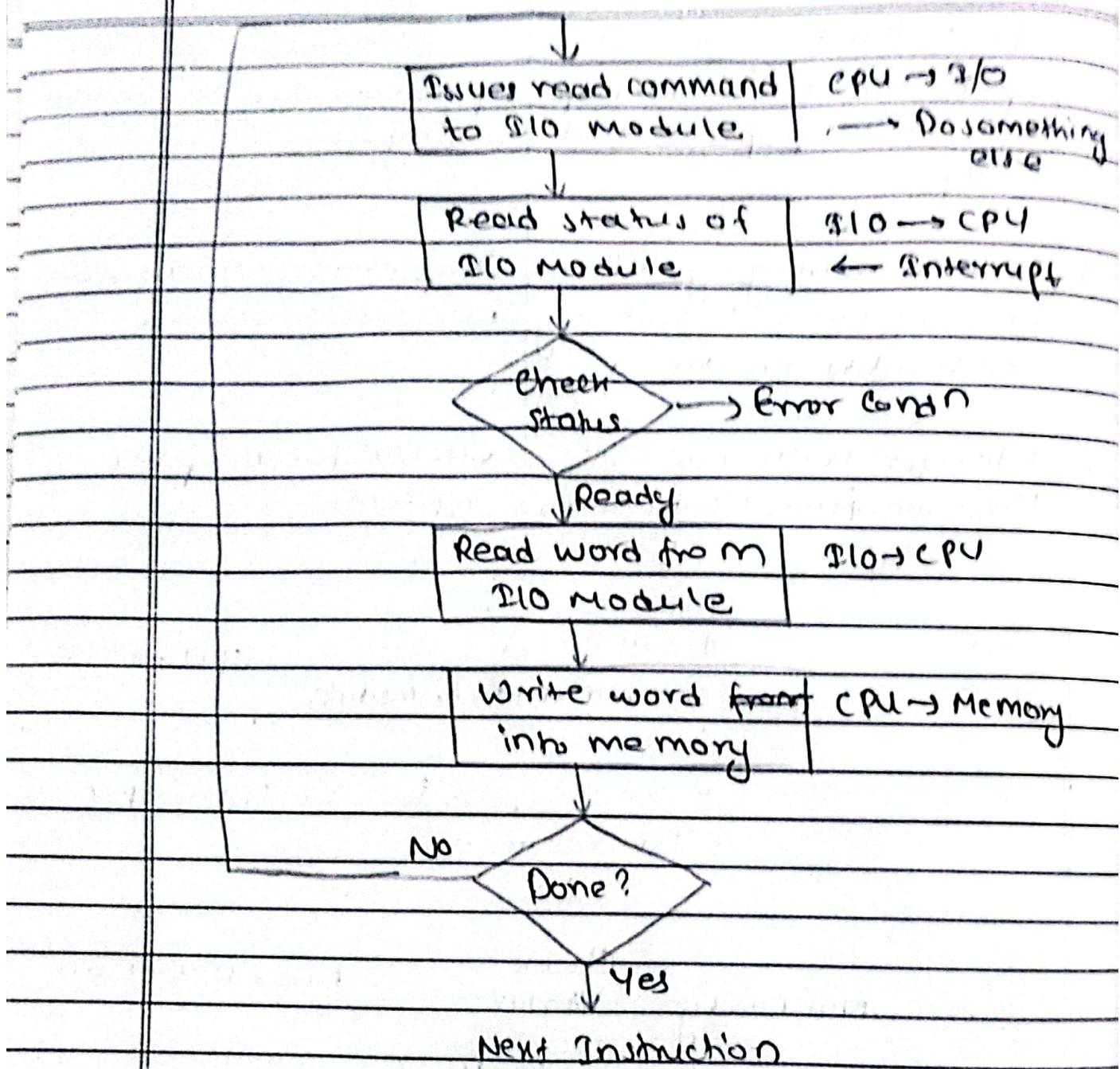


Fig:- Interrupt Driven I/O.

(Q. 5.b) Ans

Instruction pipelining is a technique used in modern processors to enhance parallelism and improve performance.

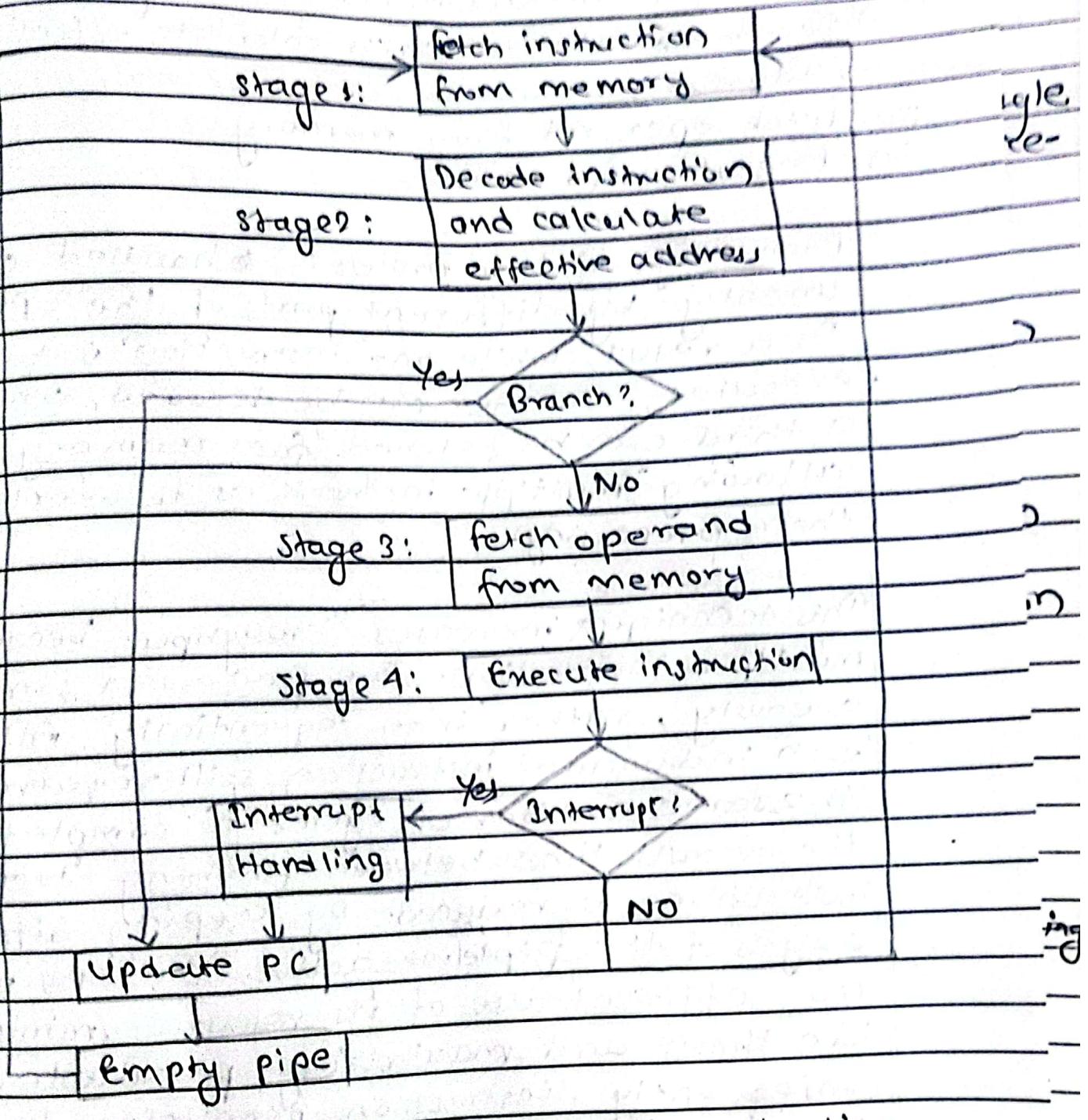


fig.. four stage instruction pipeline.

Instruction pipelining boosts processor performance by dividing the instruction execution process into several distinct stages such as :-

C11

- i) Fetch the instruction from memory
- ii) Decode the instruction and calculate effective address.
- iii) fetch operand from memory.
- iv) Execute and store.

Each stage of this process is handled concurrently by different parts of the CPU. As a result, while one instruction is being executed, another can be decoded, and a third can be fetched from memory, allowing multiple instructions to overlap their processing.

This technique increases throughput because multiple instructions are processed simultaneously, rather than sequentially. Although each individual instruction still requires the same number of cycles to complete, the overall time between finishing successive instructions is reduced. By keeping different stages of the pipeline active, the CPU makes more efficient use of its resources, minimizing idle times and maximizing parallelism. Consequently, instruction pipelining leads to faster processing speeds and improved overall performance, making better use of the CPU's capabilities.

Q. (c-a) Ans

In multiprocessor systems, cache coherence becomes a critical issue when multiple processor have their own local caches.

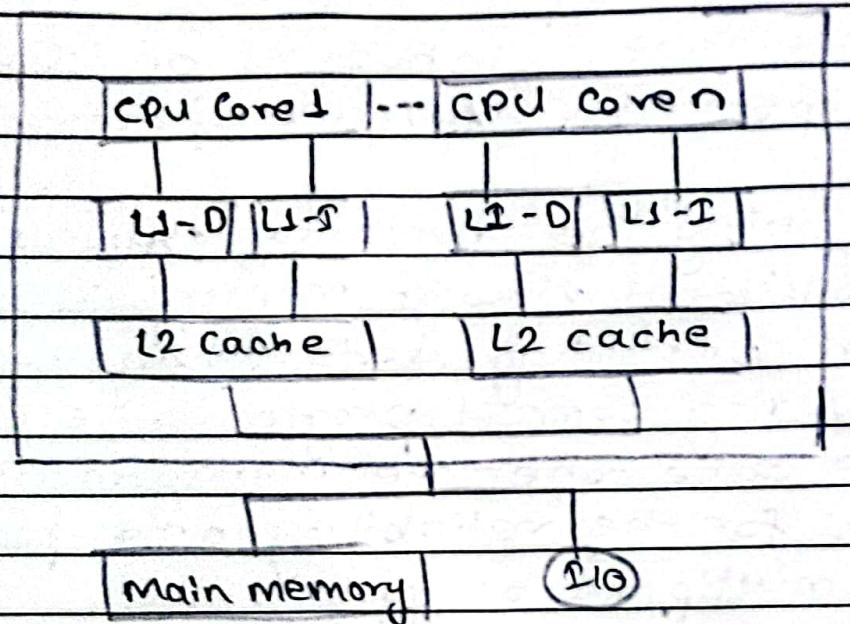


fig:- multiple processors having own local caches

Here, multiple processors ~~have~~ cache copies of the same memory location. If one processor updates its cached data, other processor might still hold outdated or inconsistent copies, leading to discrepancies and incorrect results. This inconsistency can cause significant issues in parallel computation and multi-threaded applications, where accurate and synchronized data access is crucial.

To address this issue, cache coherence protocols are implemented. These protocols, such as MESI,

(Modified, Exclusive, Shared, Invalid) and MESI (Modified, Owner, Exclusive, Shared, Invalid), ensure that all caches reflect the most recent updates. MESI manages cache line states and coordinates read and write operations to maintain consistency, while MESI introduces an "Owner" state to optimize performance further.

By using these protocols, systems can maintain a consistent view of memory across all processors, preventing data integrity issues and ensuring correct program execution. Effective cache coherence management is essential for the reliability and performance of multiprocessor systems.

Q.7.c) Ans

Flynn's classification consists of four groups.

- 1) Single instruction stream, single data stream (SISD)
- 2) Single instruction stream, multiple data stream (SIMD)
- 3) Multiple instruction stream, single data stream (MISD)
- 4) Multiple instruction stream, multiple data stream (MIMD)

They are explained as below:

SISD represents the organization of a single computer containing a control unit, a processor, and a memory unit. Instructions are executed simultaneously and the system may or may not have internal parallel processing. Parallel processing is through multiple functional unit.

SIMD has organization including many processing units under the classification supervision of a common control unit. Receives same instructions but operates on different items of data.

MISD is only theoretical application; no practical system has been constructed using organization.

MIMD refers to computer capable of organizing processing several programs at the same time. Most multiprocessor and multiple computer systems is classified in this category.

(Q.7-a) Ans(A) (2020 Fall)

Shift micro-operations :

1) Linear shift:

shift takes place one position to the left or right. The end bit is discarded and '0' takes the vacant position

Eg:-

$$\begin{array}{r} \times 1001 \\ \text{left shift} \quad \swarrow \swarrow \swarrow \\ 0010 \end{array} \qquad \begin{array}{r} 1001 \\ \text{Right shift} \quad \searrow \searrow \searrow \\ 0100 \end{array}$$

2) Circular shift:

It works similar as linear shift but the difference is, the bit that is discarded in linear shift takes the vacant position.

Eg:-

$$\begin{array}{r} 1001 \\ \text{left: } \swarrow \swarrow \swarrow \quad \curvearrowright \\ 0011 \end{array} \qquad \begin{array}{r} 1001 \\ \text{Right: } \quad \curvearrowright \quad \curvearrowright \quad \curvearrowright \\ 1100 \end{array}$$

3) Decimal shift:

It was developed for BCD representation. It acts like linear shift except it shift one digit or 4-bit

Eg:-

$$\begin{array}{r} 01101100 \\ \text{left} \quad \swarrow \\ 11000000 \end{array} \qquad \begin{array}{r} 01101100 \\ \text{Right} \quad \searrow \\ 00000110 \end{array}$$

.. www.com
Submission Date: 11th June, 2024

A) Arithmetic shift:

This was developed for signed number notations - Here, left most bit is the signed bit and remains unchanged by shift operation.

Eg:-

left: $\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ \downarrow \quad \swarrow \quad \searrow \\ 1 \ 0 \ 1 \ 0 \end{array}$ Right $\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ \downarrow \quad \searrow \quad \searrow \\ 1 \ 1 \ 0 \ 0 \end{array}$