Computer organisation and Architecture.

1. what is computer Architecture and organisation? write difference between them.
2. Draw the block diagram of vonn-neuman computer (vonn-neuman base Architecture). Draw the block diagram of structure of an IAS computer.
3. what is vonn-neuman concept? what are its component?
4. what are the advantages of vonn-neuman architecture.? what is vonn-neuman bottleneck? How to improve it?
5. How many cpu-registers are there? discuss special purpose registers (structure of an IAS comp)
6. what is instruction format? what is field? discuss ~~its fields?~~ commonly used fields.
7. classify cpu-organisation. give examples of each.
8. what are the advantages of using different number of addresses in instructions?
9. what are the factors that affects the length of the instruction.
10. evaluate the expression using 3-address, 2-address, 1-address and 0-address instruction formats. - $(A+B*C)/(D-E)$.
11. what is addressing modes? classify addressing modes with examples.
12. Draw the state diagram of instruction cycle. Discuss it shortly.

13. what is Machine cycle?
what is T-state?
14. Draw the Memory hierarchy.
Discuss each Memory.
15. what are the different access Methods?
16. what is access time?
what is bandwidth? (data transfer rate)
what is latency?
17. what is instruction? discuss its type.
18. what is instruction set and its characteristics. Draw block diag. of basic instr.
19. write difference between RISC and CISC
20. calculate effective address for each, LDA 2500, where PC = 2000, R1 = 2300, XR = 20
21. what is Main Memory?
discuss its types.
22. write difference between RAM & ROM.
23. what is locality of reference?
24. A hierarchial cache-main Memory subsystem has the following specification-
i) cache access time = 50 ns
ii) main Memory " " = 500 ns
iii) 80% of the Memory request are for read.
iv) hit-ratio of 0.9 for read & write-through scheme is used.

calculate:
i) Average access time of the Memory system considering only Memory read cycle
ii) average access time of the system for both read and write req.

25. What is ROM? Discuss its types.
What is the difference between SRAM and D-RAM.?

26. Draw block diagram of associative memory. write its definition? what are its advantage & disadvantage.

27. write organisation of associative memory.

28. A cache have 64 KB, capacity, 128 byte line & is 4-way set associative. The cpu generates 32-bit address for accessing data in the memory.

i) How many entries are required in the tag field.

ii) How many bits of tag are req. in each entry of the tag array.

iii) How many lines and sets does the cache have?

29. A computer has main memory of size 64K×60 & a cache use direct mapping with the block size of 4 words.

i) How many bits are there in the tag, block & word fields of the address format.

ii) How many bits are there in each word of the cache.

iii) How many blocks can the cache accommodate.

30. represent (-0.75) in IEEE. single Precision format.

class test:

31. Give an example and explain base-indexed addressing mode (Q11). (32)
33. Explain diff b/w 3, 2, 1 addressing modes with example. (Q10).
34. what is stored Program concept? (Von-Neumann concept).
35. Explain how a RAM of capacity 2 KB can be built using RAM chips of size 512 x 8. (Q-14)
36. Why do we require Memory hierarchy.
37. compare RISC & CISC. (Q-19).
38.

google classroom

38. write difference between hardwired and microprogrammed control unit.
39. what are the various data transfer modes? with examples.
40. write working Principle of DMA with block diagram and write its advantages.
41. write booth's Algo & find 7 x 3 in booth's Algo.
42. convert -32.75 in IEEE.
43. Explain how pipelining improves Performance in computer Architecture.
44. A non-pipelined Processor takes 5 cycles to execute an instruction, if pipelining is introduced with 5 stages and ideal conditions. calculate the speedup & efficiency for 100 instruction.

FAQ's

45. what is translate buffer and its purpose.
46. Explain the Merits & demerits of the floating point & fixed point representations for storing real number. what is biased component?
47. Discuss the asynchronus mode of data transfer with its Merits and demerits. what are the different modes of data transfer b/w cpu & I-O devices. (Explain them with their Merits and demerits). (Q-39)
48. convert hexa to octal '68 BE'.
49. what is hit-ratio?
50. what is write-through & write-back Mechanism.
51. calculate max capacity of a Memory which uses an address bus of size 12 bit.
52. How Many address bits are required for a 512 X 4 Memory.
53. what is bus and its types.
54. Subtraction by 10's complement 5864 − 2378. of subtrahend.

P4Q-24.

1). Computer Architecture:-

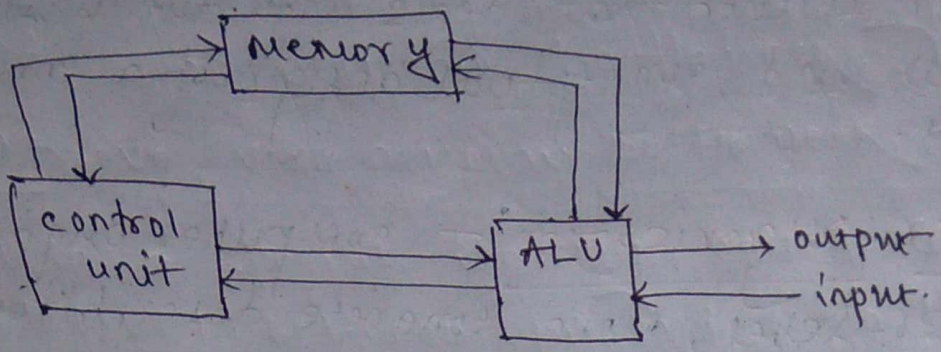
Computer Architecture is a branch of study in computer science that deals with the basic overview and conceptual design of a computer system.

Computer organisation:- computer organisation is lower-level, more concrete description of the computer system that deals with the interconection and interoperation between the constituent part of the computer system.

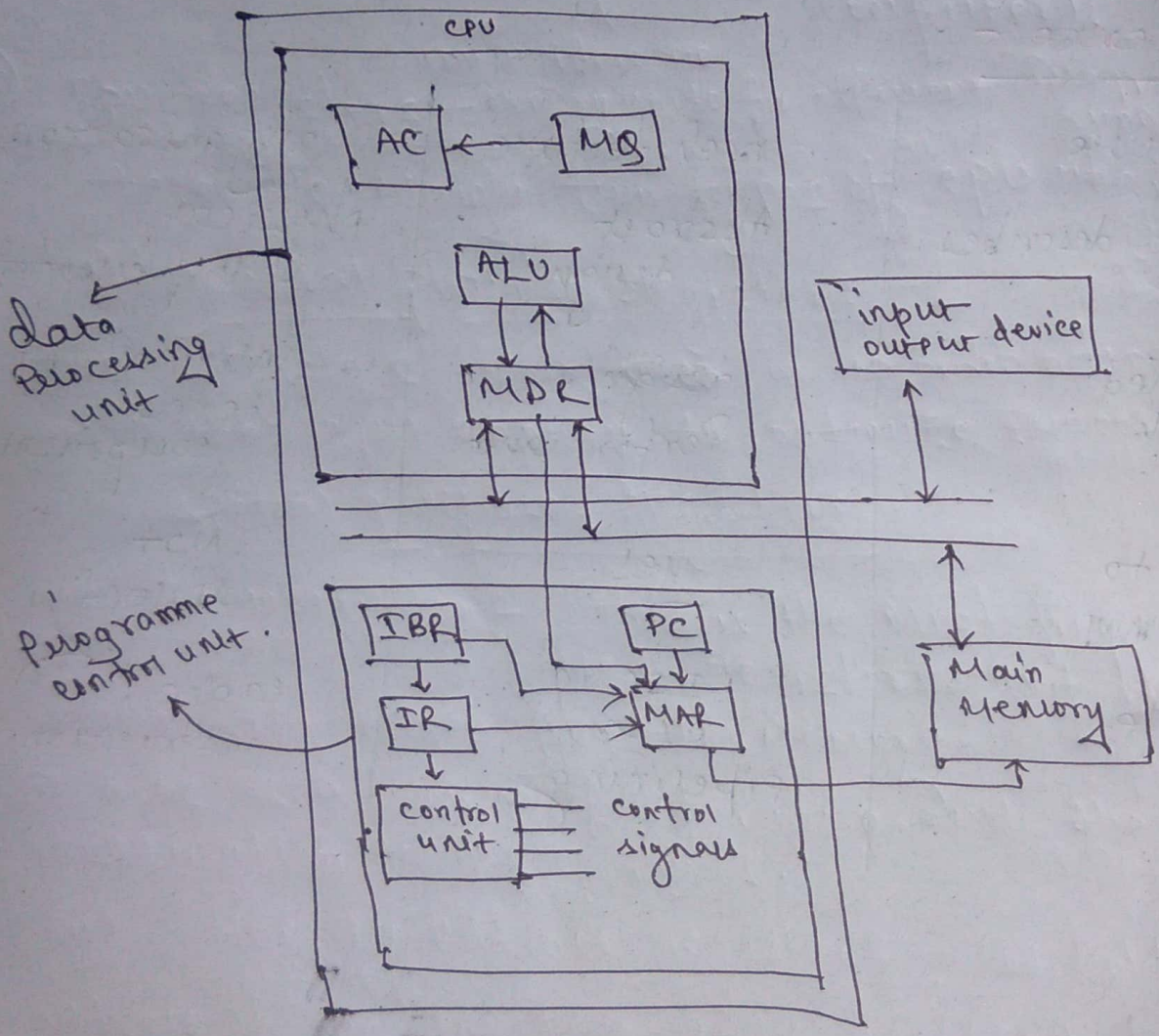
Difference:-

Feature	Architecture	organisation
what it describes	Abstract design	physical implementation
concerned with	design, Performance	internal ckt, internal component.
visible to Programmar	yes	Not directly
example topic	RISC vs CISC, Pipelining	cache, instruction

8) Block diagram of von-neuman base Architecture



von-neuman base Architecture



structure of IAS computer

3) Vonn-neuman concept or stored Program Concept

The vonn-neuman concept proposed the stored Program concept that states that:-

(i) Program and data can be stored in same memory.

(ii) A computer executes the Program sequentially as directed by the instructions in the Program.

(iii) A ~~computer~~ Program can Modify itself when the computer executes the Program.

The Components of the Vonn-neuman Computer

(i) control unit
(ii) ALU
(iii) Memory unit
(iv) system bus.

4) the Advantages of vonn-neuman architecture:-

(i) Reducing hardware complexity:- uses same memory for storing data & instructions, thus it reduce hardware complexity.

(ii) flexible:- can run various programme.

(iii) ease of Programming:-

5) ~~Data~~ vonn-neuman bottleneck:-

one of the major factor contributing to computer's performance is the time required to execute instructions given in programme and the flow of the data between cpu and memory. But in vonn-neuman architecture, as the data and the programme stored together inside same memory the cpu has to wait longer to obtain a data word from the memory than from its registers

because registers are fast and logically placed inside the cpu.
so this cpu-memory speed disperaty is called vonn-neuman bottleneck.

To improve vonn-neuman bottle neck :-

(I) Use a special memory called cache memory between cpu and main memory.

(II) using a special type of computer that is RISC (Reduced instruction set computer).

5) **cpu registers :-** cpu registers are memory units for efficient Programm running.

there are mainly two types of cpu registers

a) **general purpose registers :-**
these registers work in co-ordination with the cpu memory to optimize processing.

b) **special purpose registers :-**

(i) **Accumulator :-** This is the most frequently used registers. used to store data taken from the memory.
It acts as a source of operation and the destination to the operands.

(ii) **Memory Address registers :-** It holds the address of the location to be accessed from the memory.

(iii) **Memory data registers :-** It contains data to be written into or read out from the addressed location.

{ MAR & MDR facilitate Communication b/w cpu and Memory }

(iv) Program counter:- • PC used to keep track of the execution of the program.
• It contains the address of the next instruction to be fetched from the main Memory, when the previous instruction has successfully Completed.

(v) Instruction registers:- • the IR holds the instruction which just about to be executed.

(vi) stack pointer:- • the stack pointer points to the top of the stack, which is a part of the memory.
• stack pointer used to store function calls & other operations.

(vii) Flag registers | status register | Condition code register:- It is used to indicate status of the CPU or status of the out comes of various operations.

(viii) Instruction Buffer registers:- IBR used to hold right handed instructions temporarily from a word in memory.

(ix) Multiplier Quotient:- MQ used in ALU to store results of multiplication and division operations.

6) Instruction format:- Instruction formats refers to the way the ~~informations~~ instructions are encoded and represented in machine language formats.

~~Fields~~:- Fields:- Instructions in Computer are divided into groups called fields.

Commonly used fields:-

(i) **op.code:-** this field state the operation to be performed.
e.g. Add, substract, Multiply.

(ii) **operand:-** An operand can be a value or an address.

(iii) **Mode:-** this field specifies the method to get the operand or the effective address of the operand.

(7) **cpu organisation:-**

there are three ways in which cpu can be organised -

(1) **single Accumulator based cpu organisation:-**

- In this type of cpu org. Computers have single Accumulator based cpu.
- It has a simple cpu, in which the AC register used ~~to store~~ implicitly for processing all instructions of a program.
- The instruction format in this cpu org. uses one address field. that is why it is known as one-address machine.

e.g. ~~the instructions~~ PDP-8 processors.

(ii) general registers based cpu org.

- Multiple cpu registers are used in this type of cpu org.
- This type of computer uses two-three address field in their instruction format.

e.g. IPN-360.

Adv- ↓time, ↓mem
dis Ad 1 : 1

(iii) stack based cpu org:-

- stack based computers operates instructions based on a data structure called stack.
- operands are stored in stack (top of the stack)
- two operations can performed on the operands push & pop.
- the register that hold the address of top most operand in the stack is called stack pointer.
- In this type of cpu org., the instruction format do not need any field that is zero-address field.

e.g. INTEL-8085

(8) Advantages of using different number of addresses in the instruction:-

(i) the fewer the address, the shorter length of instruction, results less complex decoding ckts.

(ii) limiting number of addresses limits the range of operations each instruction can perform.

(iii) long Program (large number of address) requires long execution time

(9) What are the factors that effect length of Instruction:

(i) Memory size
(ii) Memory organisation
(iii) Bus structure
(iv) cpu complexity
(v) cpu speed.

3-Address OP, dest, src1, src2
2 OP dest, src
1 OP operand

(10) $X = \frac{(A + B * C)}{(D - E)}$

3-Address

```
MUL  T1, B, C  ; T1 = B*C
ADD  T2, A, T1 ; T2 = A+T1
SUB  T3, D, E  ; T3 = D-E
DIV  X, T2, T3 ; X = T2/T3
```

2-Address

```
MOV  T1, B  ; T1 = B
MUL  T1, C  ; T1 = T1*C
ADD  T1, A  ; T1 = T1+A
MOV  T2, D  ; T2 = D
SUB  T2, E  ; T2 = T2-E
DIV  T1, T2 ; T1 = T1/T2
```

1-Address

```
LOAD  B  ; AC = B
MUL   C  ; AC = B*C
ADD   A  ; AC = A+B*C
STORE T1 ; T1 = AC
LOAD  D  ; AC = D
SUB   E  ; AC = D-E
STORE T2 ; T2 = AC
LOAD  T1 ; AC = A+B*C
DIV   T2 ; AC = (A+B*C)/(D-E)
STORE Result.
```

0-Address

Postfix ABC*+DE-/

```
PUSH A ;
PUSH B ;
PUSH C ;
MUL    ; B*C
ADD    ; B*C+A
PUSH D ;
PUSH E ;
SUB    ; D-E
DIV    ; (A+B*C)/(D-E)
```

(11) Addressing Modes:-

Addressing Modes are the different ways in which the location of an operand is specified in an instruction

Types of Addressing modes:-

(a) When No address is specified as a part of instruction

(I) Implied / inherent Addressing modes:-

- In this mode the operands are indicated implicitly by the instruction.
- The AC is generally used to store operands and the intermediate results (AC → Accumulator).

↳ Advantage (use)

e.g. CMA , RAL

CMA → Complement accumulator

RAL → Rotate accumulator left.

(II) Immediate addressing modes:-

- In this mode the operands are mentioned explicitly by the instruction.
- An Immediate Mode instruction contain the operand rather than address of operand.

e.g. MVI A , 06H , ADI 05H

MVI A , 06H → Move immediate

ADI 05H → Add immediate

- this mode is very useful for initializing the register to a constant value.

↳ Advantage (use)

(iii) Stack Addressing mode :-

- The stack organisation based computer uses this type of mode
- In this addressing mode all the operations are taken from top of the stack (Advantage / use)
operands
- The instruction ~~have~~ does not have any operand field. (uses zero-address instruction format).

e.g. ADD , SUB

(B) Address field is a Memory Address

(i) Direct / Absolute Addressing mode :-

- In this mode the instruction contains memory address of the operand explicitly.
- Thus the address field of the instruction is called effective address

e.g. STA 2050H , LDA 2050H

STA 2050H → store accumulator direct

LDA 2050H → load AC direct.

use: Op. Address is directly available no need to calculate EA & in one clock time

(ii) Indirect addressing mode :-

- In this type of mode, the instruction contains a memory address, that holds the effective address of the operand

e.g.

STAX B , LDAX B

STAX B → store AC indirect

LDAX B → load AC indirect

there is scope of changing the address during run-time gives flexibility, in programming used in C-lang & in pointer

(c) Address field is a cpu register :-

(I) Register direct addressing mode :-

- In this mode the processor's registers hold the operand

e.g. `ADD R1, R2` // R1 ← R1+R2

use: Fast execution time & storing intermediate results in reg.

(II) Register Indirect Addressing mode :-

- In this mode the instruction specify an address of cpu register that holds the address of the operand.

useful for rapid access of main memory, such as arrays,

e.g. `STAX B` , `LDAX B`

(d) Address field do not contain effective address :-

(I) Relative addressing mode :-

- In this mode the effective address of the operand is obtained by adding the contents of the program counter with the address part of the instruction (also called offset).

i.e. operand address = contents of PC + offset.

e.g. `JZ +05`

Jump if zero.

this mode is used to specify the branch address of the branch instruction

(ii) Indexed Addressing Mode :-

- In this mode the effective address is determined by adding contents of Index register with address part of the instruction

i,e. operand address = contents of index register (XR) + offset

this mode is useful for accessing operand array

e,g. MOVC AC

Move code

(iii) Based Addressing mode :-

- In this type of mode the effective address is calculated by adding contents of base register with address part of the instruction.

i,e. operand address = contents of Base register (BR) + offset.

this mode is useful for relocation of Programs in Memory

e,g. MOV AX

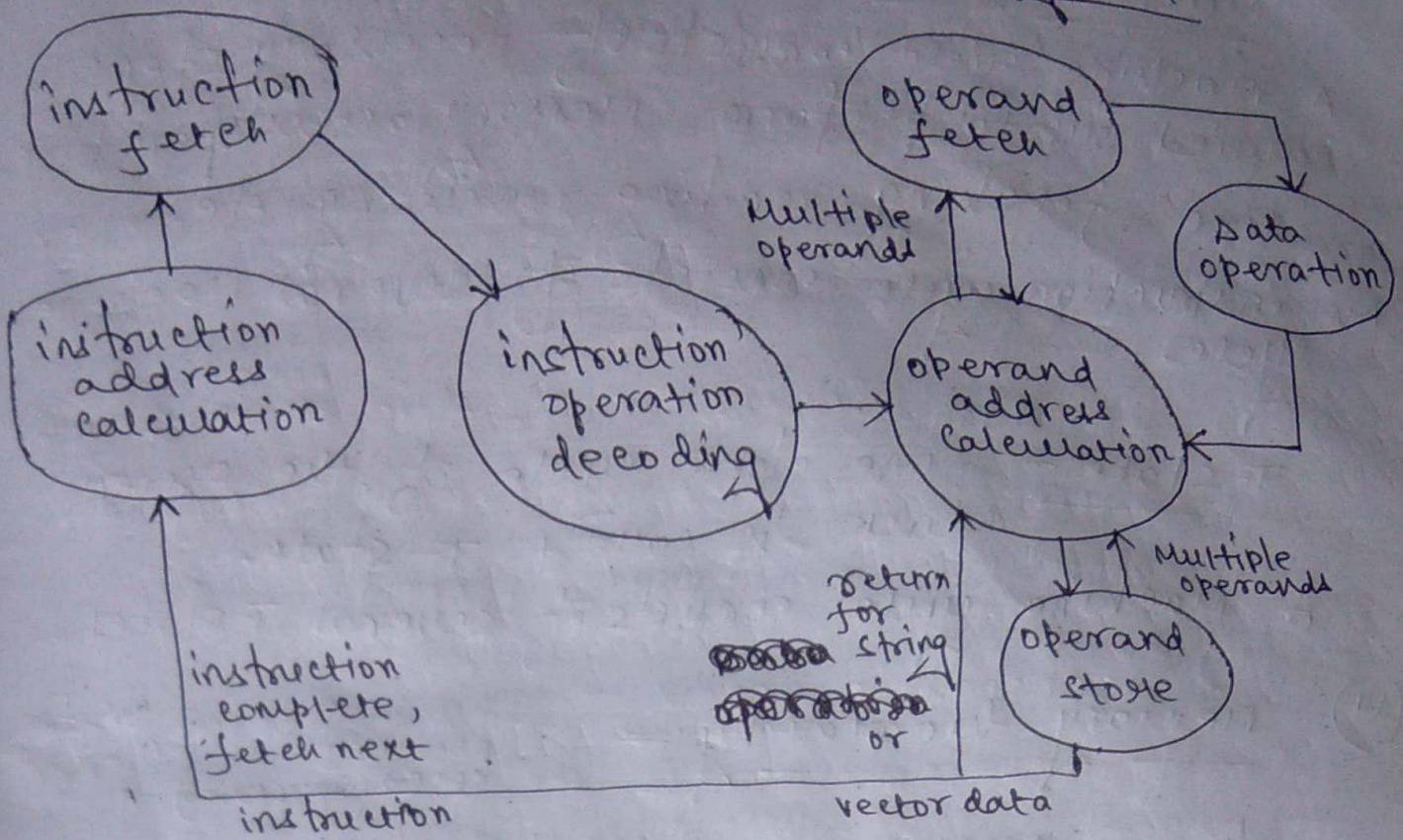
direct AM
MOV A

PYQ. 24.

* Explain the use of addressing modes with ex.

=> Addressing modes specify how the operand of an instruction is accessed during program execution. They allow flexible way to refer to data - either directly, indirectly etc. enabling efficient Memory usage and code reuse.

(12) State diagram of instruction cycle:-



the states can be described as follow:-

(i) Instruction address calculation:- determine the address of the next instruction to be executed

(ii) Instruction fetch:- read instruction from its memory location.

(iii) instruction operation decoding:- Analyse instruction to determine type of instruction.

(iv) operand address calculation:- determine the address of the operands.

(v) operand fetch:- fetch the operand from memory.

(vi) Data operation:- perform the operation indicated in the instruction.

(vii) operand store:- write or read out the result in the memory.

- Instruction cycle is a process by which a computer fetches, decodes and executes I/N."

(13) **Machine cycle:-**

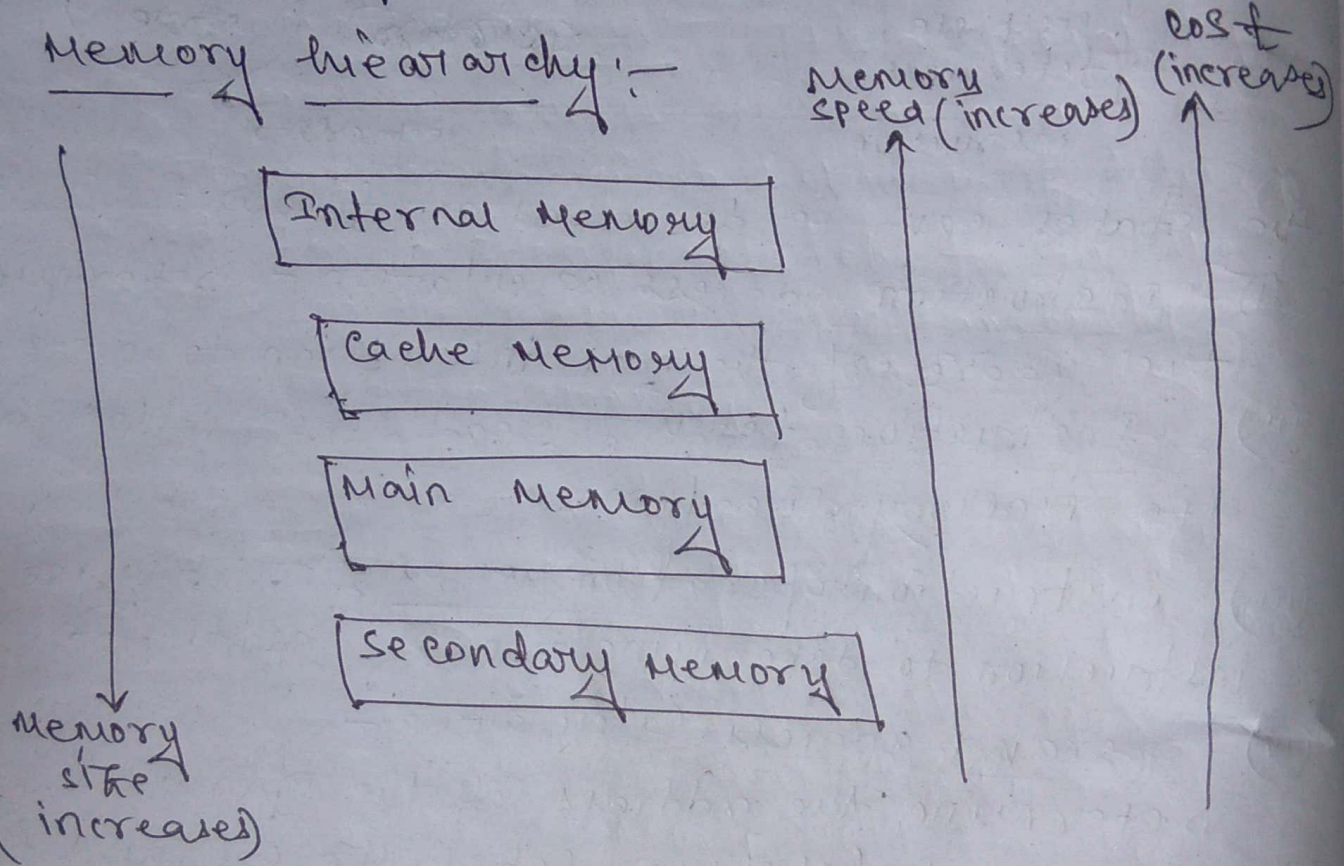
A machine cycle consists of necessary steps carried out to perform memory access operations like fetching, reading, writing etc.

An instruction cycle consists of several machine cycle.

T-state:- one clock cycle of the system clock is called T-state.

A machine cycle consists of several T-states.

(14) **Memory hierarchy:-**



(I) Secondary Memory :- the slow speed, low cost devices that provides back up storage are called secondary memory.

- Programmes that are not currently needed by the main memory are transferred here to provide space for currently used programmes.

e.g. Magnatic tapes

(II) Main Memory :- this is the memory that communicates directly with the cpu.

- only programes that are currently needed, reside here.

e.g. RAM, ROM

(III) Cache Memory :- special high speed Memory used to increase speed of processing.

e.g. L1 cache

(IV) Internal Memory :- this refers to high speed registers used inside cpu, this registers hold temporary results when the computation is in progress.

(15) Different types of access Methods :-

(i) sequential access :-

In this Method the Memory is accessed in a specific linear sequential manner.

e.g. Magnatic tapes.

(ii) Random access :- In this ~~Mode~~ Method any location of the Memory can be accessed randomly. e.g. RAM, ROM

(iii) Associative access :- It is a special type of random access method that performs comparison ~~between~~ of desired word for a specific match for all the boards in the Memory simultaneously.

e.g. Associative Memory

(16) Access time :- Access time is the time taken by the Memory to complete a read operation.

Bandwidth/ Data transfer rate :- the max. amount of information that can be transferred to or from the Memory unit time is called the Bandwidth.

expressed as words/sec.

Latency :- It is the time taken for the data to travel from one point to another.

or

delay b/w the ~~data~~ request for data & start of the response.

17. **Instruction:-** An instruction is a group of bits that instructs the computer to perform a specific operation.

there are five types of instructions:-

(I) **Data transfer instructions:-** these instruction copy information from one location to another

e.g. MOV, LOAD, PUSH etc.

(II) **Arithmatic instructions:-** these instructions performs operations on numerical data.

e.g. ADD, SUB etc

(III) **Logical instruction:-** these instructions include boolean operation

e.g. AND, OR etc.

(IV) **Program control instruction:-** these instructions change the sequence ~~of the program~~ in which the programs are executed.

e.g. JUMP, WAIT

(V) **Input-output operations:-** these instructions causes transfer of information b/w processor or memory to external i/o devices

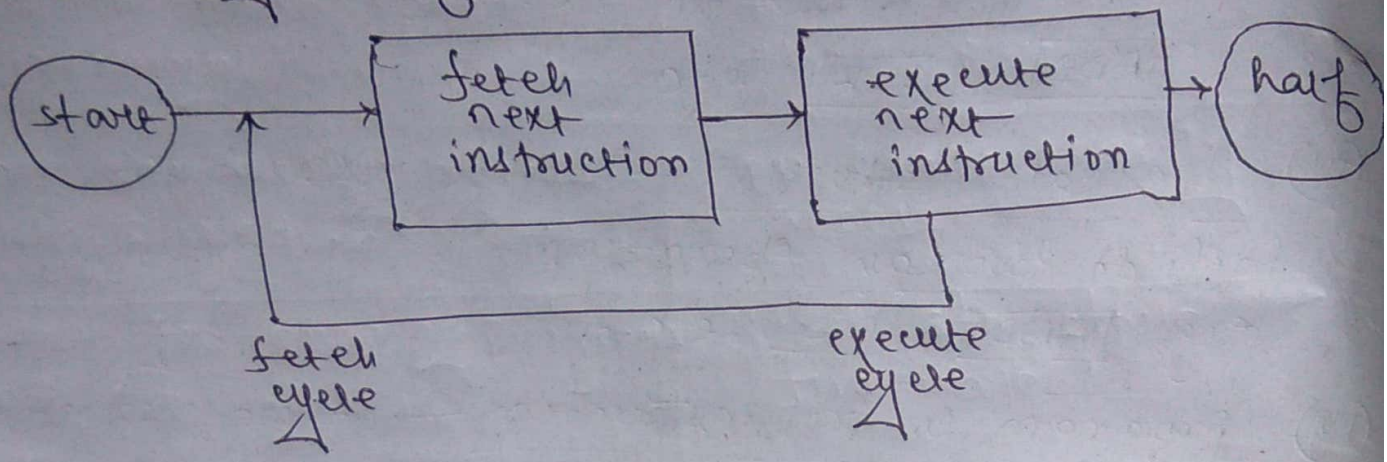
e.g. IN, OUT

(18) Instruction set :- An instruction set is a collection that defines all the instructions. A complete instruction set is refer to as instruction set Architecture. (ISA).

The characteristics of instruction set :-

(i) completeness
(ii) efficiency
(iii) Regularity
(iv) Compatibility

Block diagram of basic instruction set :-



(19) Difference between RISC & CISC

RISC	CISC
Lesser number of instructions (<100)	Greater number of instructions (>100).
single fixed length instructions.	Variable length instruction
Relatively few Addressing modes	A large number of addressing modes
Mostly hardwire control unit	Mostly microprogrammed control unit
A large number of registers are used	A small number of registers are used.

(20) PC = 2000 & LDA 2500
R1 = 2300
XR = 20

Effective address for relative addressing mode = 2000 + 2500
= 4500

E.A for Indexed Addressing modes = 20 + 2500
= 2520

E.A for Based Addressing mode = 2300 + 2500 = 4500.

(21) Main Memory :- The main memory of a computer system is a central storage unit which is directly accessible by the CPU.

Main memory are of two types :-

(i) RAM :- The RAM memory consists of IC chips and are volatile. that is if power goes off, the stored information is lost.

- used to store data & Programme that are being used by the CPU in real time.

there are two types of RAM :-

a) SRAM
b) DRAM

(ii) ROM :- ROM is non-volatile, means data stored in ROM is written after the computer is turned off. even

ROM are three types

a) Programmable ROM
b) erasable programmable ROM
c) electrically erasable Programmable ROM.

22) Difference b/w RAM and ROM

RAM	ROM
RAM is volatile, that could store the data, as long as the power is supplied	ROM is non-volatile, that could retain data even after the power is turned off
Read & write operations are supported	only read operations are supported
It is high speed memory	It is much slower than RAM
CPU can easily access data stored in RAM	CPU can't easily access
used for running programs and processes	used to store firmware

23) Locality of reference :-

Locality of reference is the tendency of a processor to access the same set of memory locations repetitively over a short period of time during program execution.

It is of two types (i) temporal locality (ii) spatial locality

```
e.g. for (int i=0; i<n; i++){
        sum += arr[i];
     }
```

'sum' is used again and again

4). given

$t_c = 50$ ns

$t_m = 500$ ns

Probability of read $(P_r) = 0.8$

Hit ratio $(h) = 0.9$

write through scheme is used.

⇒ Average access time (t_{av}) [considering only memory-read cycle]

$= h * t_c + (1-h)(t_c + t_m)$

$= 0.9 \times 50 \times 10^{-9} + (1-0.9)(\cdot 550 \times 10^{-9})$.

$= 100$ ns.

⇒ Average access time (t_{av}) [considering both read & writes].

$= p * t_{av} + (1-p) * t_m$

$= 0.8 * 100 + (1-0.8) * 500 = 180$ ns.

(22) ROM :- (Read only Memory) :- A ROM is a main memory which is used to store permanent system program (bootstrap program) & system data.

Types of ROM :-

(i) PROM :- The ROM that allow data to be loaded into the cell by the user.

(ii) EPROM :- The ROM whose contents can be loaded by user and can be erased by burning the device under an UV Ray.

(iii) EEPROM :- (electrically erasable PROM)

A ROM whose contents can be selectively erased by writing once.

e.g. Pen drive

(iv) flash memory :- is a type of EEPROM in which the density of data storage is higher.

e.g. USB device

difference b/w SRAM and DRAM

SRAM	DRAM
made with transistor that is costly	made with capacitor that is cheaper.
It is faster than DRAM	slower than SRAM.
It requires constant power supply	It does not require constant power supply
costlier	cheaper
retains stored information as long as power is supplied	Tends to loose information over a period of time, capacitor needs to charge

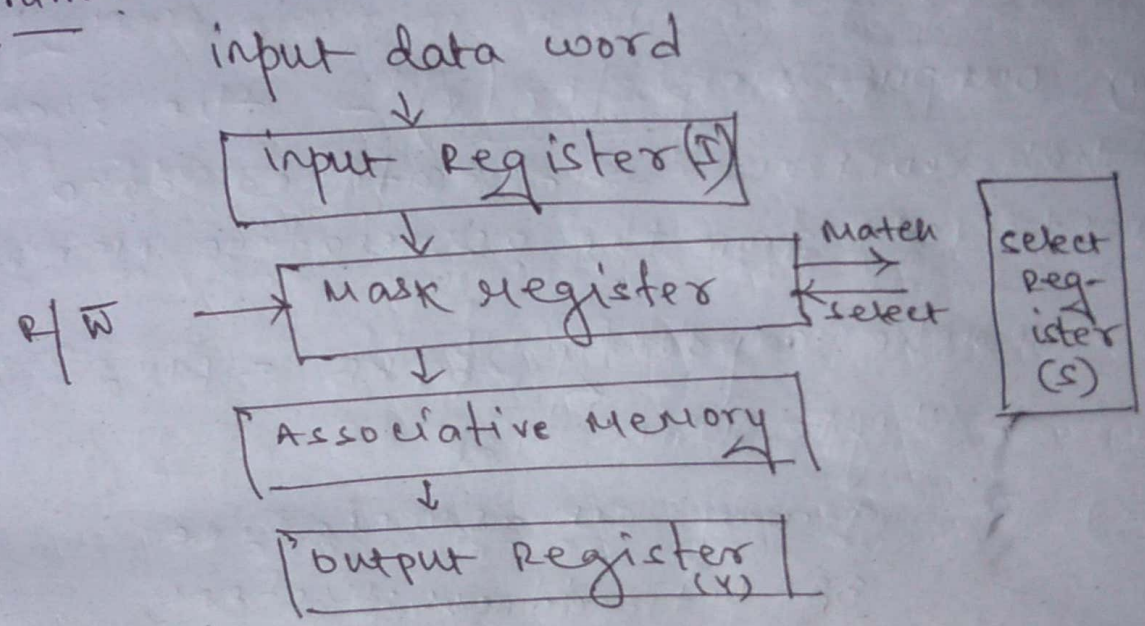
(26) Associative Memory :-

Associative Memory also called content addressable memory is a type of memory where data is accessed based on content rather than a specific memory address.

Advantage :- Associative Memory allows searching data by content not by address it is fast and efficient for matching operations.

disadvantage :- It is more expensive than RAM.

Block diagram :-



27) organisation of associative memory :-

the associative memory consists of a memory array and a logic for m words with n-bits/word.

It has four types of registers :-

(I) Input register :- the 'I' is used to hold the data to be written into or searched from associative memory. (I)

(II) mask register :- (M) :- the 'M' provides a mask for choosing a particular field.

(III) select register (S) :- if match is found after comparing input data in 'I' with the key field in 'M', then the corresponding bits in the select register is set.

(IV) output register (Y) :- the register that contain the matched data word retrived from the associative memory.

28). cache capacity 64 KB = 64×2^{10} bytes

byte line = 128

4-way set associative

32-bit address system

(a) lines = $\frac{\text{cache size}}{\text{line size}} = \frac{64 \times 2^{10}}{128} = 512$

set = $\frac{\text{number of lines}}{\text{set-way associative}} = \frac{512}{4} = 128.$

(b) each cache line has one entry
so total entries
= total lines = 512

(c) offset bits = $\log_2$ (line size)
$= \log_2 (128) = 7$

Index bits = $\log_2$ (number of sets)
$= \log_2 (128) = 7$

Tag bits = (system bits) − (index bits + offset bits)
$= 32 - (7+7)$
$= 18.$

(29) size of main Memory = 64K X 60 bits
$= 64 \times 1024 \times 60$
each site = 1K = 1024 words
block site = 4 words
cache mapping = direct mapping

⇒ number of blocks in main memory $= \frac{64K}{4} = \frac{65536}{4} = 16384$ blocks

⇒ number of blocks in cache $= \frac{1K}{4} = \frac{1024}{K} = 256$

(i) Bits in tag, block & word

- word offset (within a block)

as block size = 4 = 2^2

so bits in word = 2. (Ans)

- cache index / block number in cache

as 256 blocks total i.e. 2^8

so bits in blocks = 8 (Ans)

- Tag bits = 14 − 8 = 6 bits

as total blocks in main memory is 16384 = 2^{14}

and 8 is bits in blocks.

(ii) From the question: each word in main memory is 60 bits

so in cache (60) bits (Ans)

(iii) cache can accommodate

$$= \frac{1024 \text{ words}}{4 \text{ words/block}} = 256 \text{ blocks}$$

(Ans)

word	block	tag
2	8	6

30. (0.75) in IEEE single precision :-

step-I : converting into binary

$(0.75)_{10} = (0.11)_2$

$0.75 \times 2 = 1.50$
$.50 \times 2 = 1.0$
$0 \times 2 = 0.0$ ↓

step-II :- Normalization :-

$0.11 = 1.1 \times 2^{-1}$

so, sign bit = 0 (positive number)

Exponent = -1

step-III : calculate Biased component :-

IEEE 754 single precision uses bias = 127

Biased component $= -1 + 127 = 126$

$= (0111111)_2$

step-IV : Mantissa :-

from (1.1), we take the fractional part

so Mantissa = 10000000000000000000000

Final representation :-

0 01111111 10000000000000000000000

sign bit | exponent bias 8-bits | Mantissa (23 bits)

(Ans.)

31. (Q 11)

32. (Q 11).

33. (Q-10)

(34) (Q-3) Part-1

(35) RAM capacity = 2 KB = $2 \times 2^8 \times 8$ bits

size of each RAM chips = 512×8 "

the number of RAM chips = $\frac{2 \times 2^8 \times 8}{512 \times 8}$

$= 4.$

Design :-

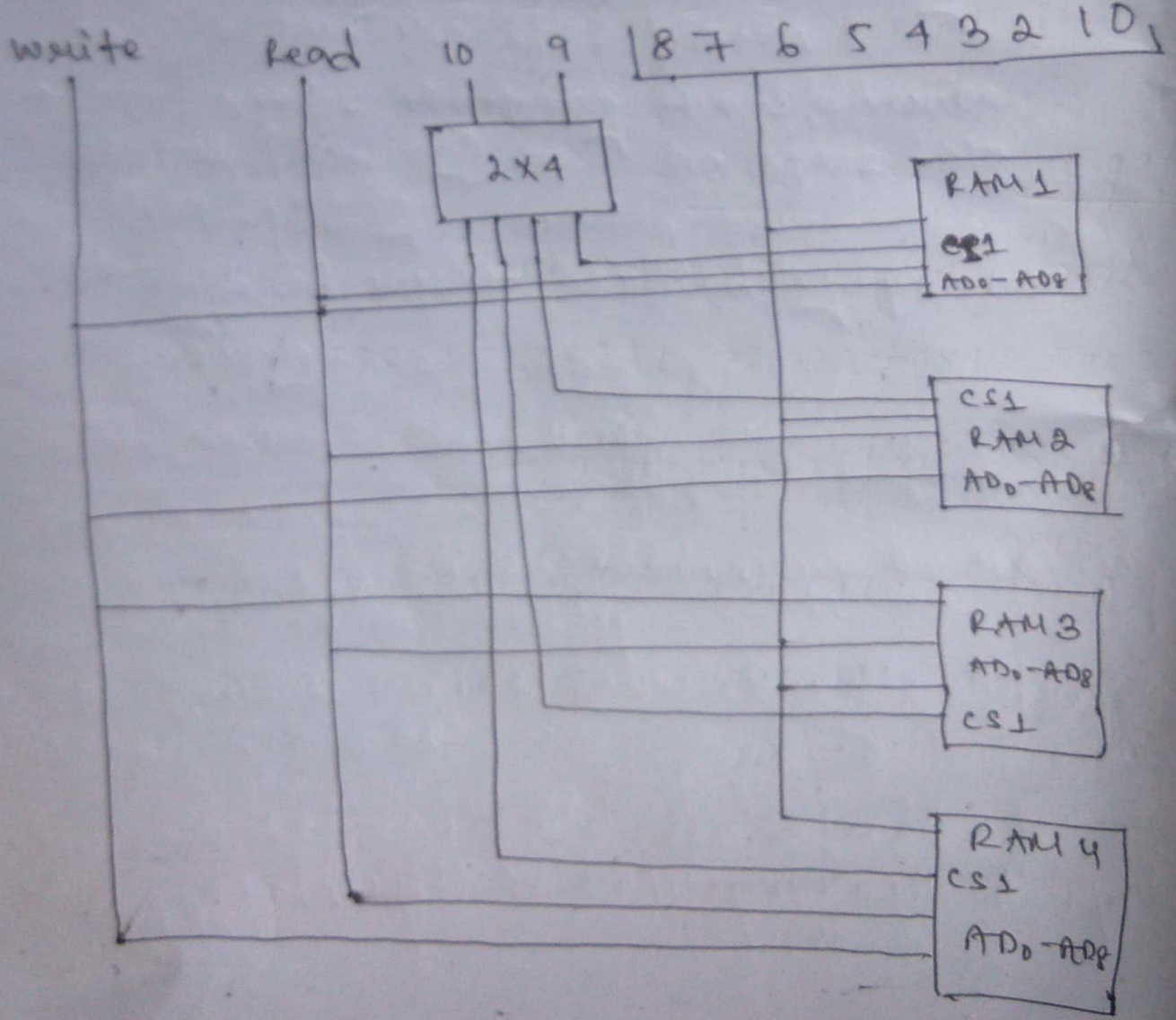


table:-

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex range
RAM1	x	x	x	x	x	0	0		0	0	0	0	0	0	0	0	0000 H
									1	1	1	1	1	1	1	1	01FF H
RAM2	x	x	x	x	x	0	1		0	0	0	0	0	0	0	0	0200 H
									1	1	1	1	1	1	1	1	03FF H
RAM3	x	x	x	x	x	1	0		0	0	0	0	0	0	0	0	0400 H
									1	1	1	1	1	1	1	1	05FF H
RAM4	x	x	x	x	x	1	1		0	0	0	0	0	0	0	0	0600 H
									1	1	1	1	1	1	1	1	07FF H

36. (Q-14)

37. (Q-19)

38. Difference b/w Hardwired control unit & Microprogrammed control unit :-

	Hardwired control unit (simple design)	Microprogrammed control unit (complex design)
Implementation	Fixed set of logic gates & circuits	Micro code stored in Memory
Flexibility	less flexible	more flexible
instruction set	supports limited Is	supports complex IS
speed	Fast operation	slower operation
size & cost	smaller size, lower cost	larger size, higher cost

39. Data transfer modes b/w the cpu and I/O devices are of three types:

1. Programmed I/O
2. Interrupt-Initiated I/O
3. Direct Memory access.

Programmed I/O:-

- Programmed I/O instructions are the result of I/O instructions written in a computer program.
- Each data item transfer is initiated by the instruction in the program.

- How it works:-

→ cpu actively controls and manages the entire data transfer process
→ cpu keeps checking the I/O device status (polling) and transfers data when ready.

e.g. keyboard input handled by cpu via polling.

Interrupt-Initiated I/O:-

- How it works:-

→ I/O devices send an interrupt to the cpu when it is ready
→ cpu pauses current task handle I/O, then resumes

e.g. mouse/keyboard sends an interrupt to notify the cpu of input

DMA :-

How it works :-

A DMA controller handles data transfer directly b/w Memory & I/O device, bypassing the cpu for each byte.

e.g. transferring a large file from disk to RAM.

40. working Principle of DMA :-

DMA uses hardware for accessing the memory, that hardware is called Dynamic memory access controller.

DMA controller is a type of control unit that works as an interface for the data bus & I/O devices

The DMA controller registers have three registers :-

(i) Adress register :- It contains the address to specify the desired location in memory

(ii) word count register :- It contains the number of words to be transferred.

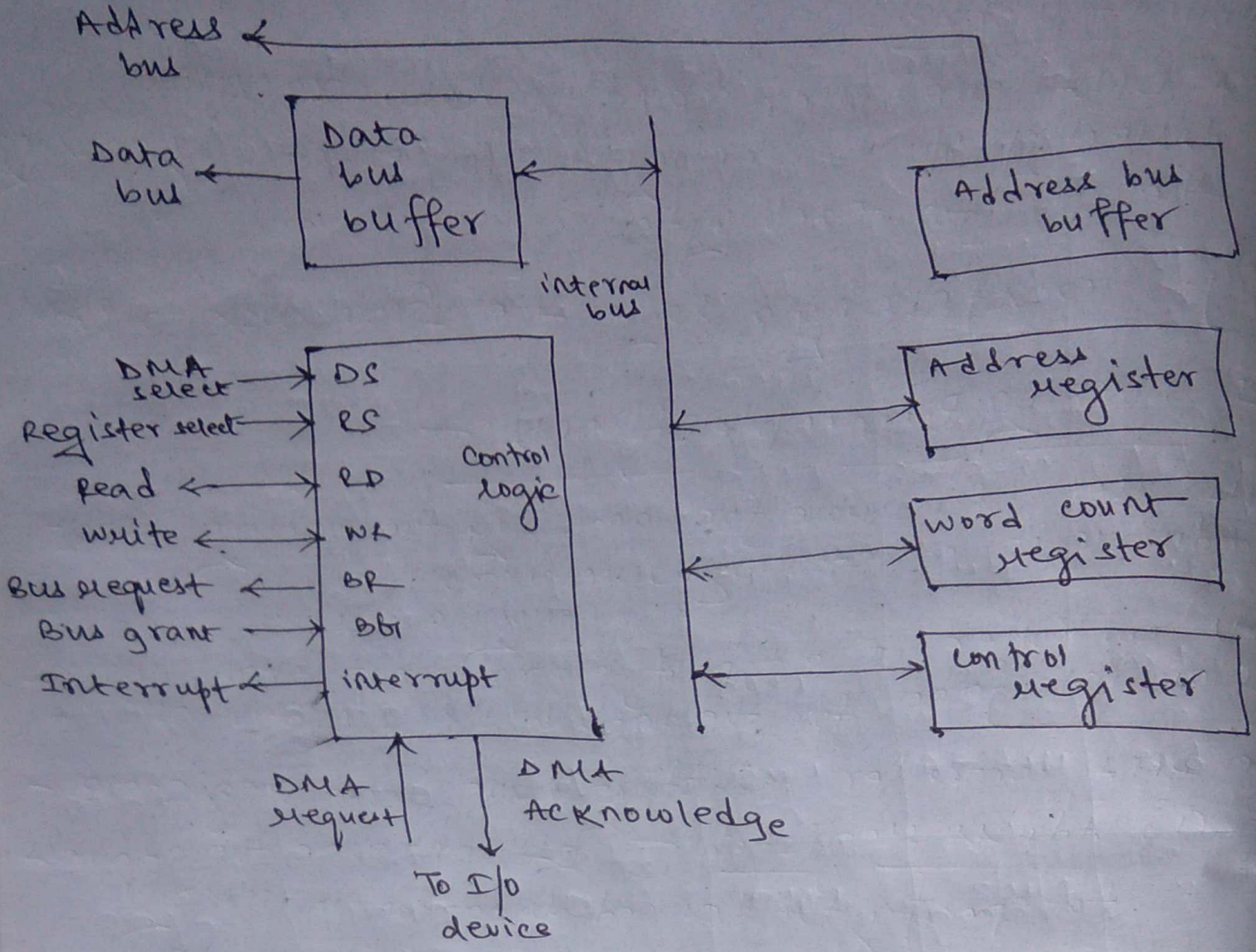
(iii) control register :- It specifies the transfer mode

Modes of data transfer in DMA :-

i) Burst mode :- Buses are handed over to cpu by DMA if the whole data is completely transferred

(ii) cycle stealing mode :- Buses are handed over to cpu by DMA after transfer of each byte

(iii) Transparent mode :- It does not require any bus in transfer of the data as it works when the cpu is executing the transaction.



Block diagram of working principle of DMA.

Advantages of DMA controller

(i) DMA speeds up memory operations and data transfer

(ii) cpu is not involved while transferring the data

(iii) DMA requires very few clock cycles while transferring data

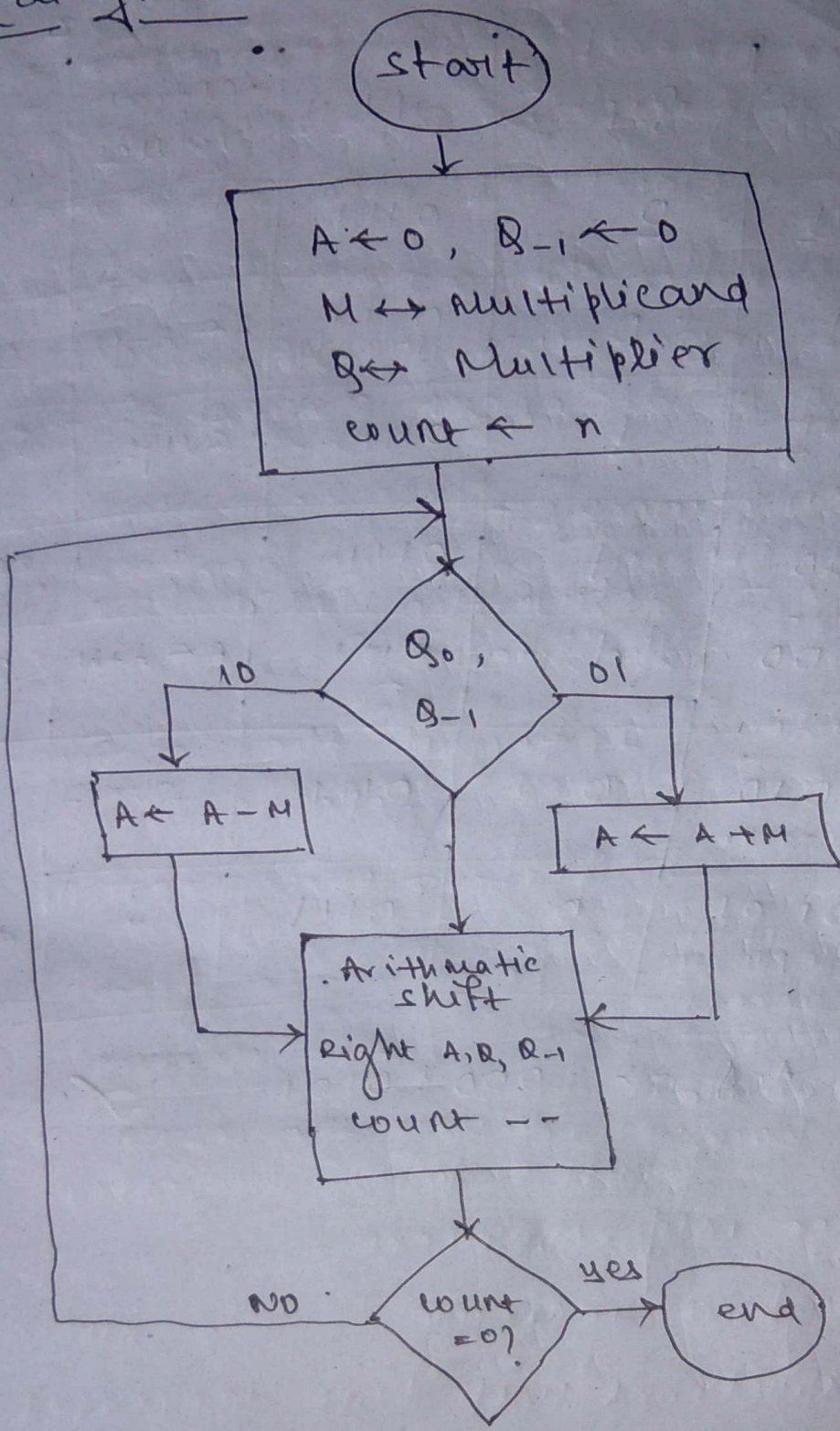
(iv) DMA distributes workload very appropriately

(v) DMA helps cpu in decreasing its loads.

41. Booth's Algorithm :-

Booth's Algo. is used to multiply two signed binary numbers.

⇒ Block diagram :-



7 × 3 in booths Multiplication

Multiplier (Q_0) = $(3)_{10}$ = $(0011)_2$

Multiplicand (M) = $(7)_{10}$ = $(0111)_2$

A	Q_0	Q_{-1}	M	initial values	cycles
0000	0011	0	0111		
1001	0011	0	0111	A ← A-M	1st cy
1100	1001	1	0111	shift	"
1110	0100	1	0111	"	2nd "
0101	0100	1	0111	A ← A+M	3rd "
0010	1010	0	0111	shift	"
0001	0101	0	0111	"	4th "

Product in A, Q → 00010101 ✓

42. (-32.75)

converting into binary

32 16 8 4 2 1

1 . . .

.75 × 2 = 1.50

.50 × 2 = 1.0 ↓

$(-32.75)_{10} = (100000.11)_2$

Normalization:—

$(100000.11)_2$

$= (1.0000011) \times 2^5$.

converting into IEEE

- biased exponent $= (127 + 5)$

$= 132$

$= (10000100)_2$

- Mantissa: 00000110000000000000000

Representing into IEEE: 1 10000100 00000110000000000000000

Ans.

sign bit

13. Pipelining improves performance in Computer architecture by allowing Multiple instructions to overlap in execution, much like a assembly line in factory. Rather than processing one instruction at a time, from start to finish, pipelining breaks the execution of instructions into separate stages Each stage handles a part of instruction, and multiple instructions can be processed simultaneously at different stages of the pipeline

* How pipelining works :-

(i) Feteh :- Retrive the Instruction from the MEMORY

(ii) Decode :- Interpret the instruction and feteh operands

(iii) Execute :- perform the operation

(iv) MEMORY access :- Read/write from/to the MEMORY

(v) write back :- write the result back to the register.

* performance improvement :-

(i) Throughout increase :- The number of instructions/unit time increases.

(ii) Improved cpu utilization :-

eg. If a processor has 5 stage pipeline and each stage take 1 clock cycle

- Non-pipelined execution of 5 instruction :- 5 instruction x 5 cycles = 25 cycles

- Pipelined execution :-
  - first instruction finishes in 5 cycles
  - each subsequent instruction completes every cycle after that.

total = 5 + 4 = 9 cycles

44. Non pipelined processor takes 5 cycles / instruction

pipelined processor : 5 stages
so 1 instruction / cycle.

Number of instruction = 100

(i) Non-pipelined total time :-
- total cycles = 100 x 5 = 500 cycle

as each instruction takes 5 cycles.

(ii) Pipelined :-
- the first instruction takes 5 cycles
- After that 1 instruction completes every cycle.

so for 100 instructions
total cycles = 5 + (100 - 1) = 104 cycles.

(iii) Speedup = $\frac{\text{time without pipeline}}{\text{time with pipeline}} = \frac{500}{104} = 4.81$

(iv) efficiency = $\frac{\text{speedup}}{\text{number of stages}} = \frac{4.81}{5} = 0.962$

$= 96.2\%$

Ans. Transient buffer:- a small, temporary storage area used to hold data in transit. - this data is being transferred b/w two locations such as-

- b/w memory and device
- b/w cpu and I/O
- b/w cache and memory.

purpose of transient buffer:-

(i) smooth data transfer
(ii) temporary storage
(iii) Avoid data loss
(iv) supports Asynchronous communication

46. Floating point representation:-

$$\text{Number} = M \times 2^E$$

M → Mantissa, E → Exponent.

Merits:-
(i) wide range
(ii) high precision
(iii) standardized: follows IEEE 754

demerits:- (i) more complex
(ii) slower processing
(iii) Rounding errors

Fixed point representation

Real numbers are stored with a fixed number of digits after the decimal point (e.g. 123.45)

Merits:- (i) fast & simple
(ii) efficient for embedded system
(iii) less power usage

Demerits:- (i) limited Range
(ii) low precision
(iii) Manual scaling

Biased exponent:- In floating point representation (like IEEE) the exponent is stored in a biased form rather than as a signed number

e.g. for IEEE 754 single precision

Exponent bits = 8

Bias = 127 $\left(\text{i.e. } 2^{8-1}-1\right)$.

$= 128-1$

$= 127$

47. **Asynchronous mode of data transfer:-**

Data is sent one byte or character at a time.

How it works:-

each char. (usually 8-bits) is transferred with -

- 1 start bit (always 0)
- Data bits (usually 7 or 8 bits)
- optional parity bits (for error check
- 1 or more stop bits (always 1)

Merits

- simple
- cheap
- flexible

Demerits

- slower
- limited speed
- low efficiency

48. $(68BE)_{16}$ 8121

$= (0110\ 1000\ 1011\ 1110)_2$

$= (064276)_8$

49. **Hit ratio:-** is a measure used in cache memory system to indicate how often the requested data is found directly in the cache, without needing to access slower main memory.

$$\text{Hit ratio}(h) = \frac{\text{number of cache hits}}{\text{total number of memory accesses.}}$$

50. **write-through:-** when the cpu writes data to the cache, the same data is immediately written to the main memory as well.

so cache & main memory are always consistent

write-back mechanism:-

- when the cpu writes data to the cache, it only updates the cache.
- the modified cache is marked as dirty
- The data written back to main memory only when that cache block is replaced.

51.

Max. Memory capacity = 2^(number of address bits) x word size.

→ Address bus size = 12 bits

→ each address points to 1 word

Max Memory capacity = $2^{12} \times 1$ = 4096 bytes

= 4 KB

52.

No: of address bits = $\log_2$(number of locations)

= $\log_2(512) = 9$

9 address bits are req.

53. what is bus:- is a communication system that transfers data b/w components inside a computer (like cpu, I/O devices, Memory).

It consists of wires or traces on a motherboard that carry data, addresses, and control signals.

Main types

(i) Data Bus : Carries actual data, Bidirectional

(ii) Address bus : Carries address, unidirectional

(iii) control bus : Carries control signals, both

54. 5869
- 2378

using 10's complement.

the subtrahend is 2378

10's complement of 2378 + 1

= 7621 + 1

= 7622

2 3 7 8

9-2 =7, 9-3 =6, 9-7 =2, 9-8 =1

Adding with minuend

5869
2378
13982

drop. (as overflow) = 3982