Name - Ipsha Ghimire Subject - CA
Semester - V 'Day'
Roll - 181313

2018 fall

a) Explain different stages of computer evolution with reference of generation of computers.

Ans Computers are devices that accomplish tasks or calculations in accordance to a set of directions or programs. The first fully electronic computers, introduced in the 1940s were voluminous devices that required teams of people to handle. In comparable with those machines, today's computers are astounding. They are not thousands of times more expeditous, but also they can fit on your desk, on your lap, or even in your pocket. Computers are such an integral part of our everyday life.

Computer work through an interaction of hardware and software. The whole picture of the computer goes back to decades. However there are five apparent generations of computers. Each generation is defined by a paramount technologies development that changes necessarily how computers operate leading to more compressed, inexpensive but more dynamic, efficient and

booming machines

first Generation - Vaccum Tubes (1940-1956)

these ancient computers utilized vacuum tubes as circuitry and magnetic drums for recollection. As a result they were huge, actually taking up entire rooms and costing fissures to run. These were ineffective materials which produce a huge amount of heat, sucked enormous electricity and subsequently engendered an abundance of heat which caused perpetual breakdowns.

These first generations computers relied on machine language. these computers were limited to solving one problem at a time. Input was produced on punch cards and paper tape. output emerged on print outs. the two eminent machines were the UNIVAC and ENIAC machines. the UNIVAC is first ever commerical computer which was purchased in 1951 by a business named the US Census Bureau.

* Second Generation - transistors (1956 - 1963)

the supersession of vaccum tubes by transistors, visualised the onset of

second general of computing. Although first invented in 1947, transistors weren't used considerably in computes until the cessation of 1950's they were a huge development over the vaccum tube, despite the fact still subjecting computers to destroying diffent levels of heat However they were extremely superior to the vaccum tubes making computers smaller, more expeditious, even pensive and less burdersome on electricity use. They still count on punch cards for input / printouts ~~The two eminent machines of this era were the~~

Third generation - Integrated circuits (1960-1971).

By this phase, transistors were now being minimized and placed on silicon chips called semiconductors, which drastically increased the speed and efficiency of computers.

Instead of punched cards and printouts users interacted with an ~~OS~~ operating system, which allowed the device to run many different applications at one time with a central program that monitored the memory.

Computers for the first time became accessible to a mass audience because they were smaller and cheaper than their predecessors.

* Fourth generations :- (microprossors ~~and present~~ (1971 - Present)

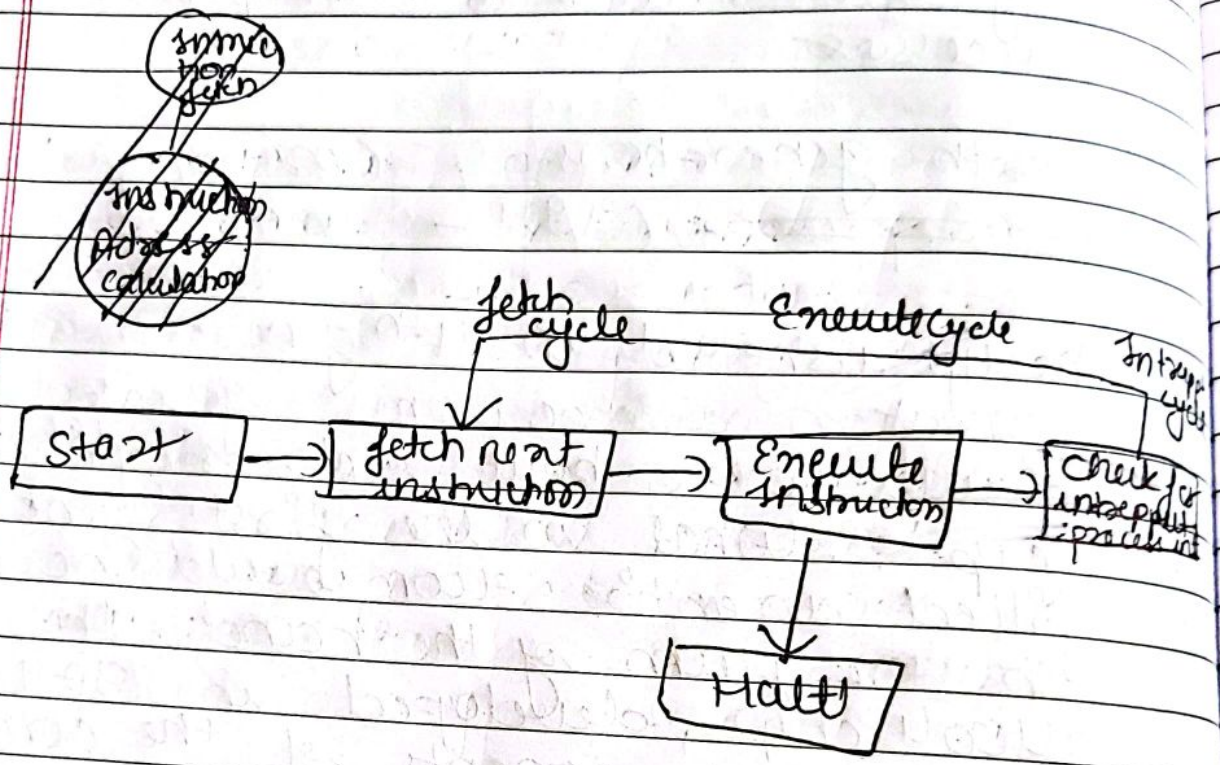
The up ushered in 4th generation of computers as thousands of integrated circuits were built onto single silicon chip. What in the first generation filled an entire room could now fit in the palm of the hand. The intel 4004 chip, developed in 1971, located all the components of the computer from the central processing unit and memory to input/output controls on a single chip.

* Fifth generation :- AI (Present and beyond.

fifth generation computing devices based on AI, as still in development, though there are some application such as voice recognition that are being used today. The use of parallel processing and super conductors is helping to make AI a reality.

Scanned with CamScanner

b) Explain instruction cycle state diagram with interrupt.



(1) Instruction address calculation (iac):-
Read/ instruction from its memory locaho into the processor

Determine the address of the next instruction to be executed. Usually, this involves adding a fixed number to address of the previous instruction.
for eg if each instruction is 16 bits long and memory is organized into 16 bit words, then add 1 to previous address.

(ii) Instruction fetch (if)

Read instruction from its memory location into the processor

(iii) Instruction operation decoding (iod):
Analyze instruction to determine type of operation to be performed and operand to be used.

(iv) Operand address calculation (oac)
If the operation involves reference to an operation in memory or available via I/O, then determine the address of the operand.

(v) Operand fetch (of):- fetch the operand from memory or read it in from I/O

(vi) Data operation (dt): Perform the operation indicated in the instruction

(vii) Operand store (os):- Write the result into memory or out to I/O.

(viii) fetch:- Read the next instruction from memory into the processor.

(ix) Execute: Interpret the op-code and perform the operation.

Scanned with CamScanner

(x) Interrupt:- If ~~to~~ interrupts are enabled and an interrupt has occured save the current process state and service the interrupt.

The main line of activity consists of alternating instruction fetch and instruction execution activities. After an instruction is fetched, it is examined to determined if any indirect addressing is involved. If so, the required operand are fetched using indirect addressing. Following execution an interrupt may be processed before the next instruction fetch.

Q.2a) How overflow be detected in computer. Verify the operation (-7) / (3) using signed 2's complement method.

Ans When two numbers of n digits each are added and the sum occupies n+1 digits, we say that an overflow occured. An overflow is a problem in digital computers because the width of registers is finite. A result that contains n+1 bits cannot be accommodated in a register with a standard length of n bits.

Scanned with CamScanner

eg → /3

for → 7

+7 = 0000 111

its 2's complement = 11111001

A	Q	M	
1111	1001	0011	shift A & Q
1111	0010	0011	A ← A + M
0010	0010	0011	Restore $A, Q_0 = 0$
1111	0100	0011	2nd cycle
1110	0100	0011	shift A & Q
0001	0100	0011	A ← A + M
1110	1000	0011	Restore A, $Q_0 = 0$
1100	1000	0011	3rd cycle
1111	1001	0011	shift A, Q
			A ← A + M
			$Q_0 \leftarrow 1$

2b) Compare and contrast betn microprogram and hardwired control unit. Explain microprogram sequencer with example

Hardwired control unit	Microprogrammed control unit
(1) Hardwired control unit generates the control signals needed for the processor using logic circuits	(1) Microprogrammed control unit generates the control signals with the help of microinstructions stored in control memory.

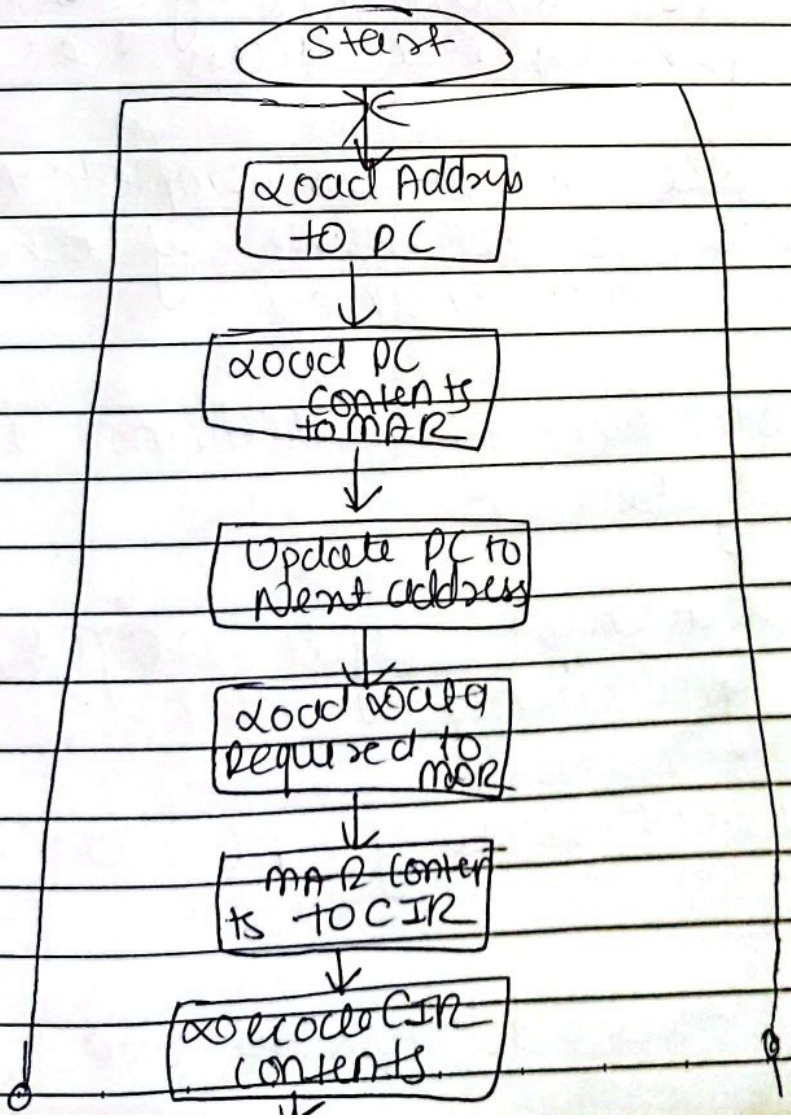
(2) Hardwired CU is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares.	(2) Ther slower than the other as micro instructions are used for generating signals here.
(3) Difficult to modify as the control signals that need to be generated are hard wired.	(3) Easy to modify as the modification need to be done only at the instruction level.
(4) More costlier as everything has to be realized in terms of logic gate	(4) Less costlier than hardwired control as only micro instructions are used for generating control signals

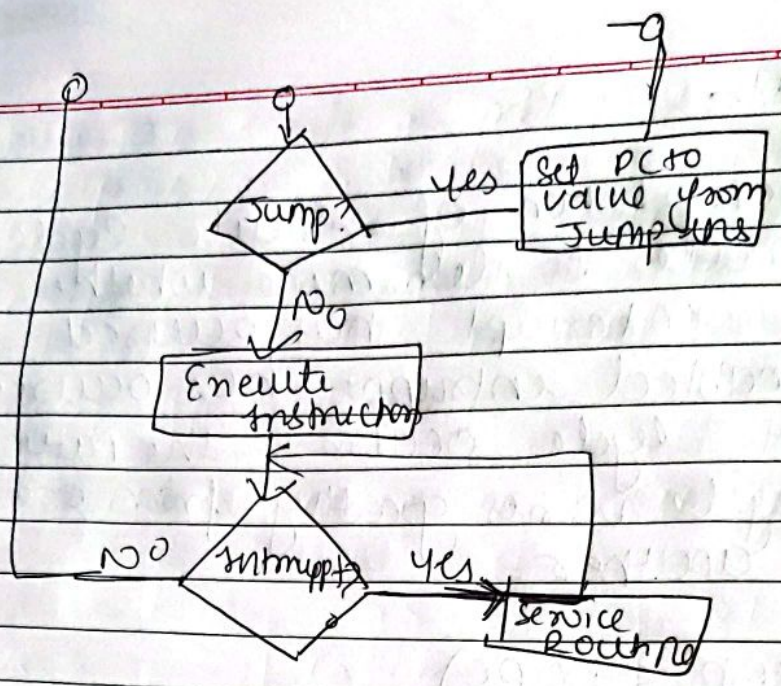
30) What is interrupt cycle? Explain how instruction be processed with interrupt in computer along with necessary diagram

~~Pm) An unsh~~

At the completion of Execute Cycle, a test is made to determine whether any enabled interrupt has occured or not. If an enabled interrupt has occured then interrupt cycle occurs. The nature of this cycle varies greatly from one machine to another.

t_1 : MBR ← (PC)
t_2 : MAR ← save address
PC ← Routine address
t_3 : Memory : (MBR)





interrupt cycle flowchart

* After the execute cycle is completed, a test is made to determine if an interrupt was enabled (eg. so that another process can access the CPU.

- After the execute cycle is completed, a test is made to determine if an interrupt was enabled.
- If not, instruction cycle returns to the fetch cycle.
- If so, the interrupt cycle might perform the following tasks.
- Save the current value of PC into MBR
- move the PC-save address to MAR

↳ move the contents of the address in MBR into indicated memory cell.

↳ Continue the instruction cycle with the interrupt routine

↳ after the interrupt routine finishes, the PC save address is used to reset the value of PC and program execution can continue.

b) What is m-way interleaving? Explain different types of memory interleaving

Ans It is a technique for compensating the relatively slow speed of DRAM. In this technique, the main memory is divided into memory banks which can be accessed individually without any dependency on the other.

In m-way interleaved, m memory blocks are accessed at same time for writing and reading operations.

Different types of memory interleaving are:-

① High order Interleaving

In high order interleaving, the most significant bits of the address select the memory chip the least significant

bits are sent as addresses to each chip. One problem is that consecutive addresses tend to be in the same chip. The maximum rate of data transfer is limited by the memory cycle time.

It is also known as memory Banking

Address bits Use	25-22 Bank select	21-0. Address to the chip.

2) Lower Order Interleaving
Here the MSB select the memory bank (module) In this, consecutive memory addresses are in different memory modules. This allows memory access at much faster rates than allowed by the cycle time.

Address bits Use	25-4 Address to the chip	3-0 Bank select

* What are the drawbacks of programmed I/O and interrupt driven I/O? Explain how DMA overcomes those drawbacks.

5a Both forms of I/O (Programmed and interrupt driven I/O suffer from two inherent drawbacks:-

1) I/O transfer rate is limited by the speed with which the processor can test and service a device
2) The processor is ~~tited~~ tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer

↓

Issues read block command to DMA module — CPU → DMA
- - → DO something else

Read status of DMA module ← - - - Interrupt DMA → CPU

↓

next instruction → CPU goes to next instruction
by DMA

DMA is a method that allows an input / output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations. The process is managed by a chip known as a DMA controller (DMAC)

A define portion of memory is used to send data directly from a phei-
pheral to the motherboard without involving the up, so that the processor does not interfere with overall computer operations.

In this way DMA overcomes the drawbacks of programmed I/O and Interrupt driven I/O.

Scanned with CamScanner

50) Assume that pipeline has $k = 8$ segment and execute $n = 125$ tasks in sequence. Let the time taken to process a sub-operation in each segment is 30 sec. Calculate the speed up ratio in the pipeline

Given
piplene has $K = 8$ segment
execute $= 125$ tasks (n)
and
$t_p = 30$ sec.

Then,
$t_n = 8 \times 30 = 240$ sec.
It will require
$k t_p =$

To complete the task, pipeline will require.
$(k - n + 1) t_p = (8 + 125 - 1) \times 30$
$= 3960$ sec.

Assuming $t_n = 8 \times 30$
$= 240$ sec.
It will require $k t_p = 125 \times 240$
$= 30,000$ sec.

$\therefore$ Speedup is $\frac{30,000}{3960}$
$= 7.57$.

5b what type of hardware and software performance issues are seen in multi-core computers? Explain

Ans. up systems have experienced a steady, exponential increase in execution performance. This increase is due to refinements in the organization of the processor on the chip and the increase in the clock frequency.

1) Pipelining

Individual instructions are executed through a pipeline of stages so that while instruction is executing in one stage of the pipeline, another instruction executing in another stage of pipeline

There is a practical limit to how far this ~~thread the~~ trend can be taken, because with more stages there is need for more logic (more ~~instruct~~) interconnections and more control signals

2) SMT

Register banks are replicated so that multiple threads can share the use of pipeline resources.

the increase in complexity to deal with all of the logical related to very long pipelines, multiple superscalar pipelines, and multiple SMT register banks means that increasing amounts of the chip area is ~~covered~~ occupied with coordinating and signal transfer logic.

This increases the difficulty of designing, fabricating and debugging the chips. Power issues are the other big challenges.

3) Cache Coherence.

Cache coherence is a concern in a multi core environment because of distributed α_1 and α_2 cache. Since each core has its own cache, the copy of the data in that cache may not always be the most up to date version. For example, imagine a dual core processor where each core brought a block of memory into its private cache, and then one core writes a value to a specific location. When the second core attempts to read the value from its cache, it will not have the most recent version unless its cache entry is invalidated and a cache-

must occur. this cache must force the 2nd core's cache entry to be updated. If this coherence policy was not in place, the wrong dated would be read and invalid results would be produced, possibly crashing the program or the entire computer.

7) Write short notes on:

a) RISC vs CISC.

CISC	RISC.
1) Here emphasis is given on hardware.	1) Emphasis is given on software.
2) Includes multiclock complex instructions	2) Single clock reduced instruction only.
3) Memory-to-memory	3) Register to register
4) Small code size	4) Low cycles per second, large code size.
5) Transistors used for storing complex instructions.	5) Spends more transistors on memory register

b) Hardware Description Language (HDL)

HDL is a programming language used to describe the behavior or structure of digital circuits (ICs). HDL are also used to stimulate the ckt and check its response. Many HDL's are available, but VHDL and verilog are by far the most popular. most CAD tools available in the market support these languages. VHDL stands for "Very high-speed integrated circuit hardware description language". Both VHDL and verilog are officially endorsed IEEE standards. Other HDLs include JHDL (Java HDL), and proprietary HDLs such as Cypress Semiconductor Corporation's ~~HDL~~ Active-HDL

2020 fall

Sol a)

* Differentiate between computer organization and architecture. Explain the various addressing modes.

Computer organization	Computer Architecture
a) Computer organization is concerned with the structure and behaviour of computer system as seen by the user.	(a) Computer architecture is concerned with the way hardware components are connected together to form a computer system.
(b) It deals with the components of a connection in a system	(b) It acts as the interface betn hardwares & softwares.
(c) Computer organization tells us how exactly all the units in the system are arranged and interconnected.	(c) Computer architecture helps us to understand the functionalities of a system

Addressing modes:-

1) Immediate
2) Direct
3) Indirect
4) Register direct
5) Register Indirect

range $= -2^{n-1}$ to $2^{n-1} - 1$
for 2's complement.

1) Addressing modes.

(1) Immediate Add Mode.

Instruction :-

	A :-

Operand = A

Here operand is present in the instruction

→ used to define and use constants or set initial values.

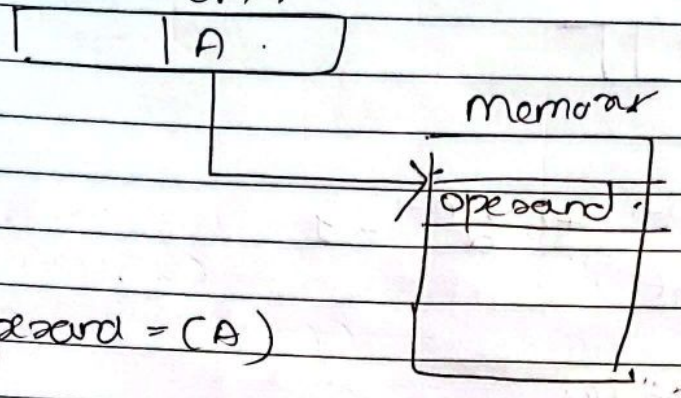
→ stored in 2's complement form & left most bit is sign-bit.

Adv.
→ no memory reference.
→ fast
Dis
The size of the number is restricted to the size of address field.

Scanned with CamScanner

(ii) Direct Addressing

Instruction.



Operand = (A)

E.A = Effective Address.
(A) = Content of A

Advantage :-

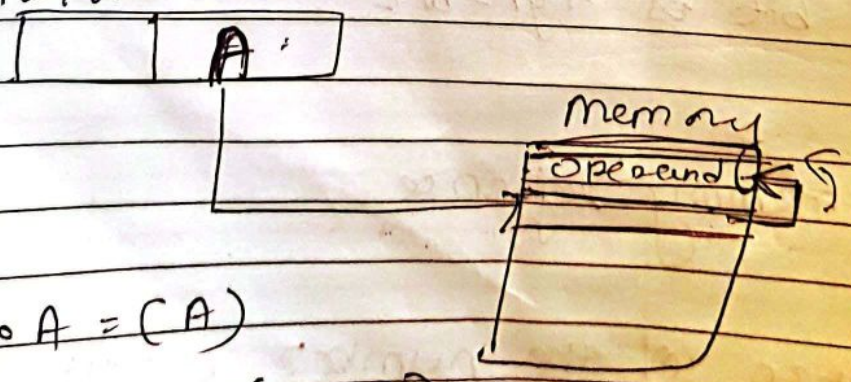
→ Single memory reference.

Disadvantage

→ Limited address space as length of address field is less than word length.

(iii) Indirect addressing

Instruction.



E.A = (A)

Operand = ((A))

Advantage.

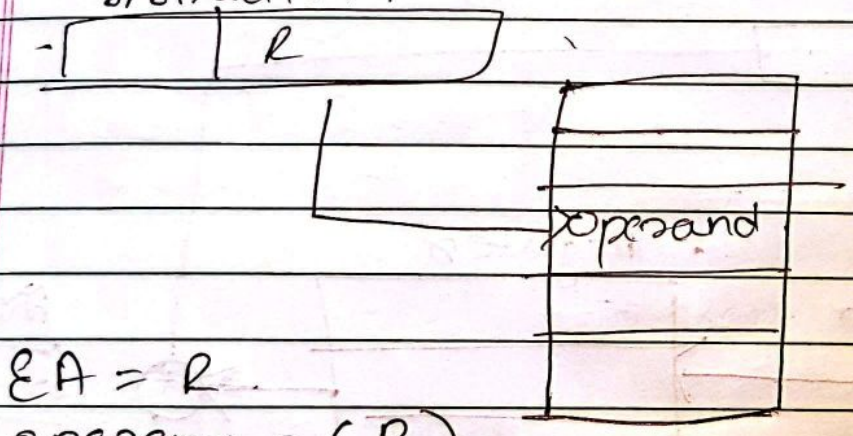
① For word length n, an address space of 2^n is available

Disadvantage

① Instruction execution requires two memory reference to fetch the operand, one to ~~get~~ get the address and 2nd to get its value.

(iv) Register addressing mode

Instruction.



EA = R.

operand = (R)

Advantage

① no memory reference.

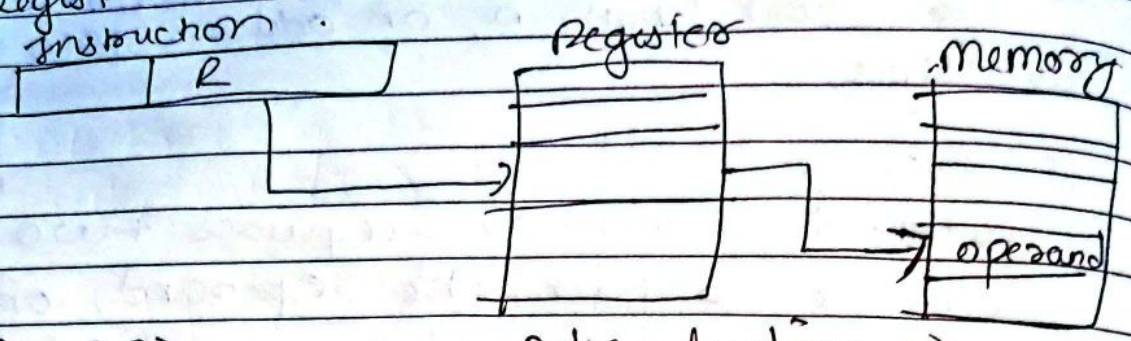
② small address field is needed in instruction

Disadvantage.

① Address space is limited to register.

Scanned with CamScanner

(g) Register Indirect

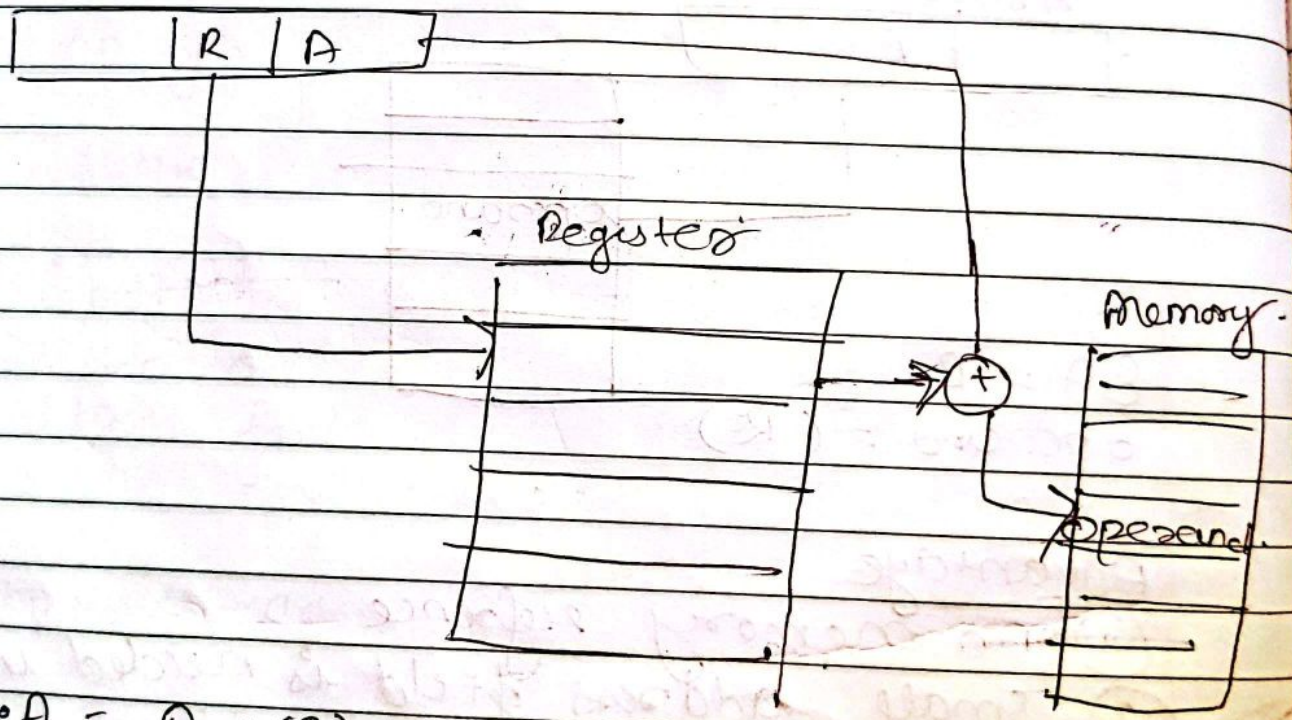


E.A = (R)

Operand = ((R))

Adv, disadvantage)
(Similar to indirect addressing)

(h) Displacement



E.A = A + (R)

Operand = (A + (R))

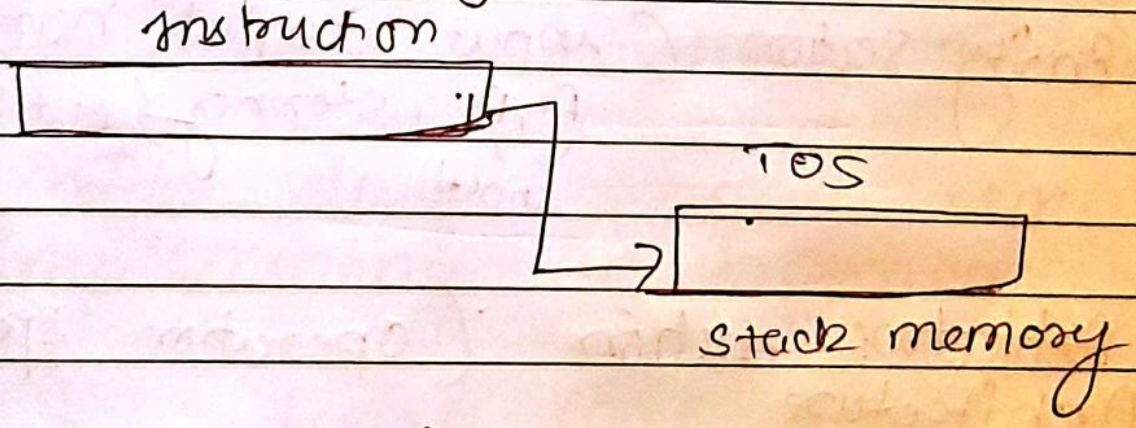
Addressing mode that combines the capability of direct addressing and register indirect addressing. It requires ~~the~~ that the instructions have

~~two~~ two address feild. Value in one feild is added directly and other refers to content of register that is added to produce effective address.

~~Stack~~. 3 types. bracket is content → R lai PC le replace

(1) Relative (PC).
(2) Base - register (BR) → similar
(3) Index - register (IR) → $EA = A + (IR)$

Stack addressing mode.



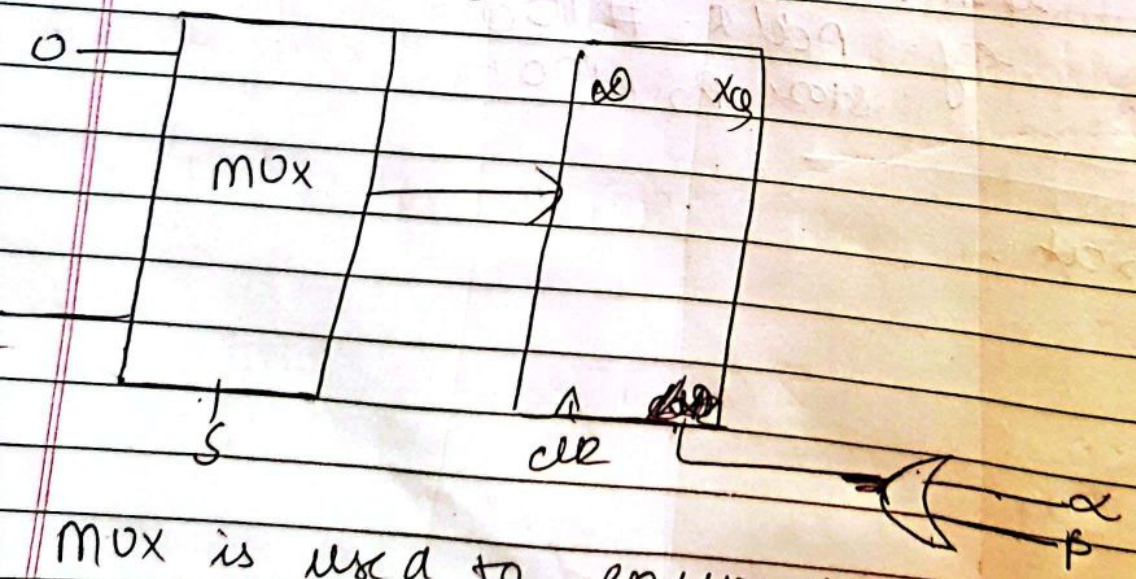
EA = Stack pointer.
Operand = Top of Stack

Q.1b Define RTL. Explain about BOS and memory transfer in RTL.

RTL.

The Register transfer language (RTL) is the symbolic representation of notations used to specify the sequence of micro-operations.

A Bus structure is more efficient for transferring information betn registers in a multi register configuration system. A bus consist of a set of common lines, one for each bit of register through which binary information is transferred one at a time. Control signals determines which register is selected by the bus during a particular register transfer



MUX is used to ensure connect β; x ← 1 I/P

The x X is set loading data for both

transfer either α or β can asserts the register load signals and mux is used to ensure the proper data transfer. If α = 0 and β = 0, x = 0. Since data value is low register x ignores these data and returns it's previous value. If α = 1 and β = 0, x is loaded as 0. If α = 0 and β = 1, x is loaded as (1). When α and β both are 1, the register will try to clear its value and load 1 which is not acceptable. This condition be resolved we can assure it that α and β is never 1 at same time or they can be modified as mutually exclusive.

Q.2(i) Explain different types of register used in computer system. What is the use of registers in computer system

Ans Register are a type of computer memory used to quickly accept, store and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as processor registers.

Scanned with CamScanner

Registers	Symbol	no of bits	function
1) Data Register	DR	16	Holds memory operand
2) Address Register	AR	12	Holds address for memory
3) Accumulator	AC	16	Processor Register
4) Instruction Register	IR	16	Holds instruction code
5) Program Counter	PC	12	Holds address of the instruction
6) Temporary Register	TR	16	Holds temp data
7) Input Register	INPR	8	Carries input character
8) Output Register	OUTR	8	Carries output character

Scanned with CamScanner

Registers are a type of computer memory used to quickly accept store and transfer data and instructions that are being used immediately by the CPU. The computer needs processor registers for manipulating data and a register for holding a memory address

Q.2b Perform multiplication -13 * 7 using Booth's Algorithm.

-13 * 7

A	Q	Q-1	M
00000	00111	0	10011
01101	00111	0	10011
00110	10011	1	10011
00011	01001	1	10011
00001	10100	1	10011
10100	10100	1	10011
11010	01010	0	10011
11101	00101	0	10011

Since -13*7 = -91 = 0001011011

2's Complement of 91 = = 1110100101

As result is stored in A & Q

Here A = 11101

Q = 00101

Combining both A & B = 11101 00101

Q.30 What are the various types of number system representation method in computer system.

(1) Integer
(2) Floating

(1) Integer :- Integer number can be represented in two forms.

1) sign number
2) unsign number

(1) In sign number :- If the sign bit is zero, the no is +ve and if the sign bit is one '1', the no is -ve. In a n-bit word, the right most (n-1) bit holds the magnitude of integer

eg
00010010 ⇒ +18
10010010 ⇒ -18.

In general case;

$$A = -\sum_{i=0}^{n-2} 2^i a_i \text{ for } a_{n-1} = 1 \Rightarrow$$

$$2^0 a_0 + 2^1 a_1 + 2^2 a_2$$

$$= 2^0 * 1 + 2^1 * 1 + 2^2 * 1$$

$= \sum_{i=0}^{n-2} 2^i a_i$ for $a_{n-1} = 0$.

(a) Unsign Number :-

Unsign notation doesn't have seperate bit to represent the sign of number. Number can be either +ve or -ve in some unsigned notations. Two commonly used unsigned notations are:-

a) Non negative notation :- It treats every number as either 0 or +ve value. And n-bit no has a value ranging from zero to $2^n - 1$. In general form for n-bit sequence number value of A is interpreted as

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

$$= 2^0 a_0 + 2^1 a_1 + 2^2 a_2 + 2^3 a_3$$
$$= 2^0 \times 1 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 1$$
$$= 1 + 0 + 0 + 8$$
$$= 9$$

(b) 2's complement method- Here both +ve & -ve numbers can be represented ie value of n bit no range from -2^{n-1} to $2^{n-1} - 1$

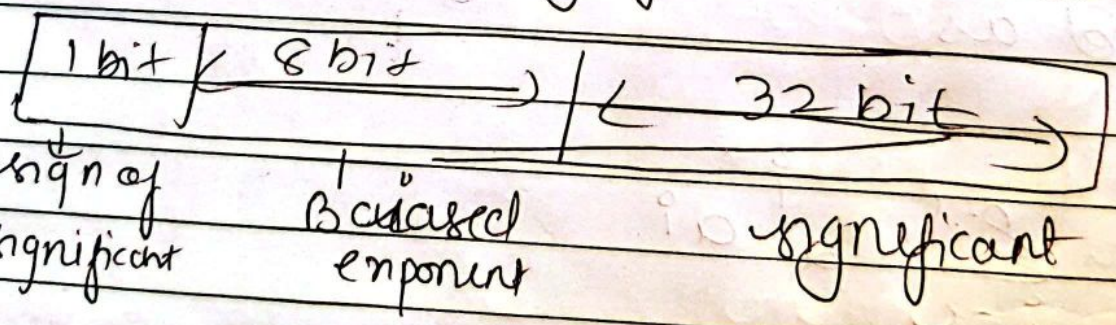
Example 4-bit
$= -8$ to $+7$

(2) floating:

In fixed point representation it is possible to represent a range of +ve and -ve integers centered on zero (0). For very large no of or fractional no integer point representation is not suitable. This can be represented in floating point representation. Number can be represented in the form

$$\pm S \times B^{\pm E}$$

where B = Base,
S = Significant E = Exponent



3D) Explain the block diagram of micro programed control organization and write down its advantages.

Scanned with CamScanner

... to be called specifies will cause ... seq

Micro instruction Execution

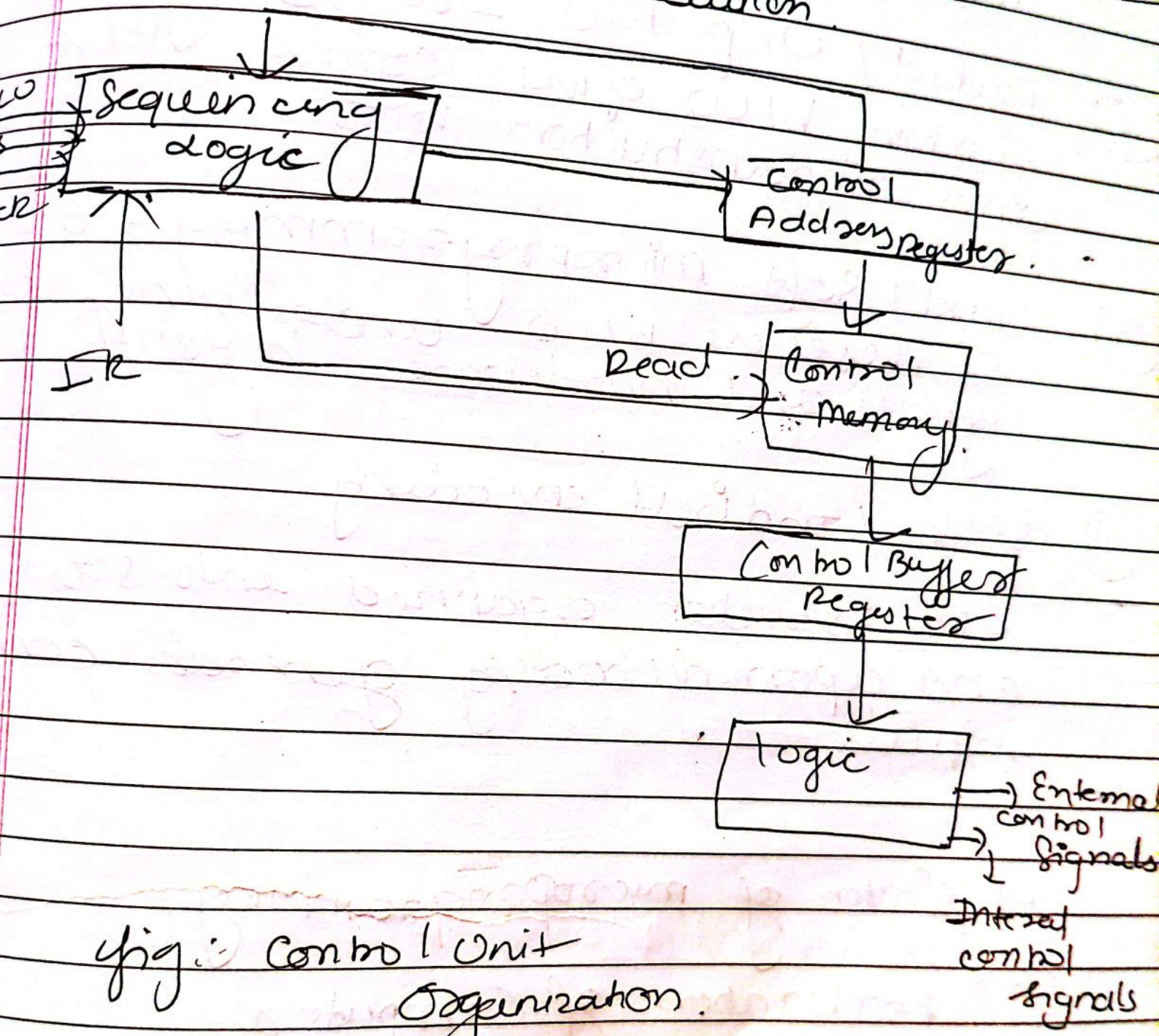


fig: Control Unit Organization.

Scanned with CamScanner

* control logic generate control logic that is determined by micro instruction format micro-instruction can be classified in following ways

(1) Vertical / horizontal - width of microinstruction

(2) Packed / Unpacked - degree of identification b/w given control field and microinstruction bits.

(3) Hard / Soft microprogramming - degree of closeness b/w underlying control signals and hardware layout.

(4) Direct / Indirect encoding

microstruction organized into sets and upon decoding generates control signals

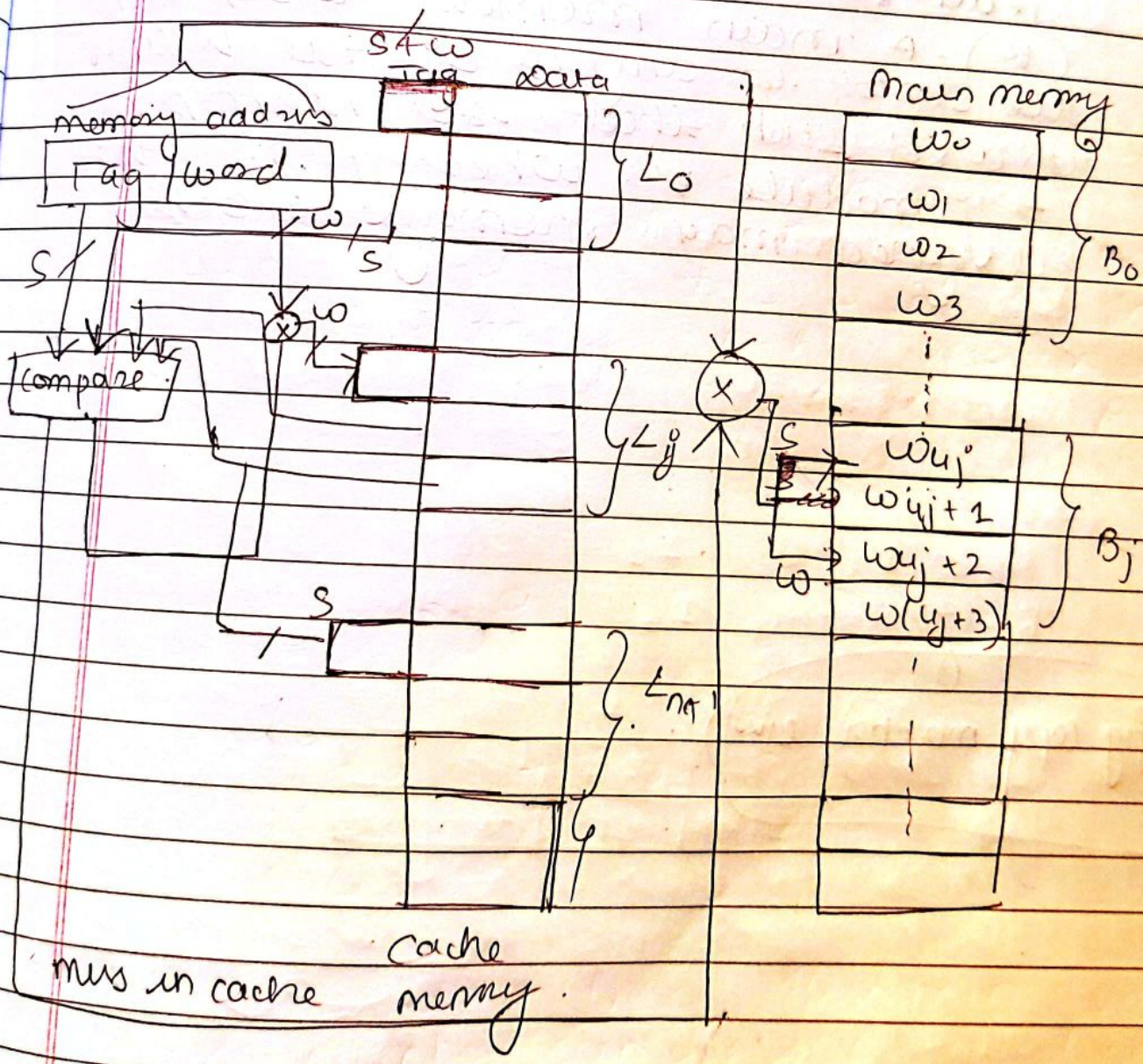
Application of microprogramming

(i) Realization of computer
(ii) Emulation
(iii) Operating system support
(iv) Realization of special purpose devices
(v) High level language support
(vi) Micro-diagnosis

Q.4(a) Explain associative and set associative mapping in cache memory?

Ans Associative mapping

Associative mapping removes the drawback of direct mapping by providing main memory block to be loaded into any line of cache.



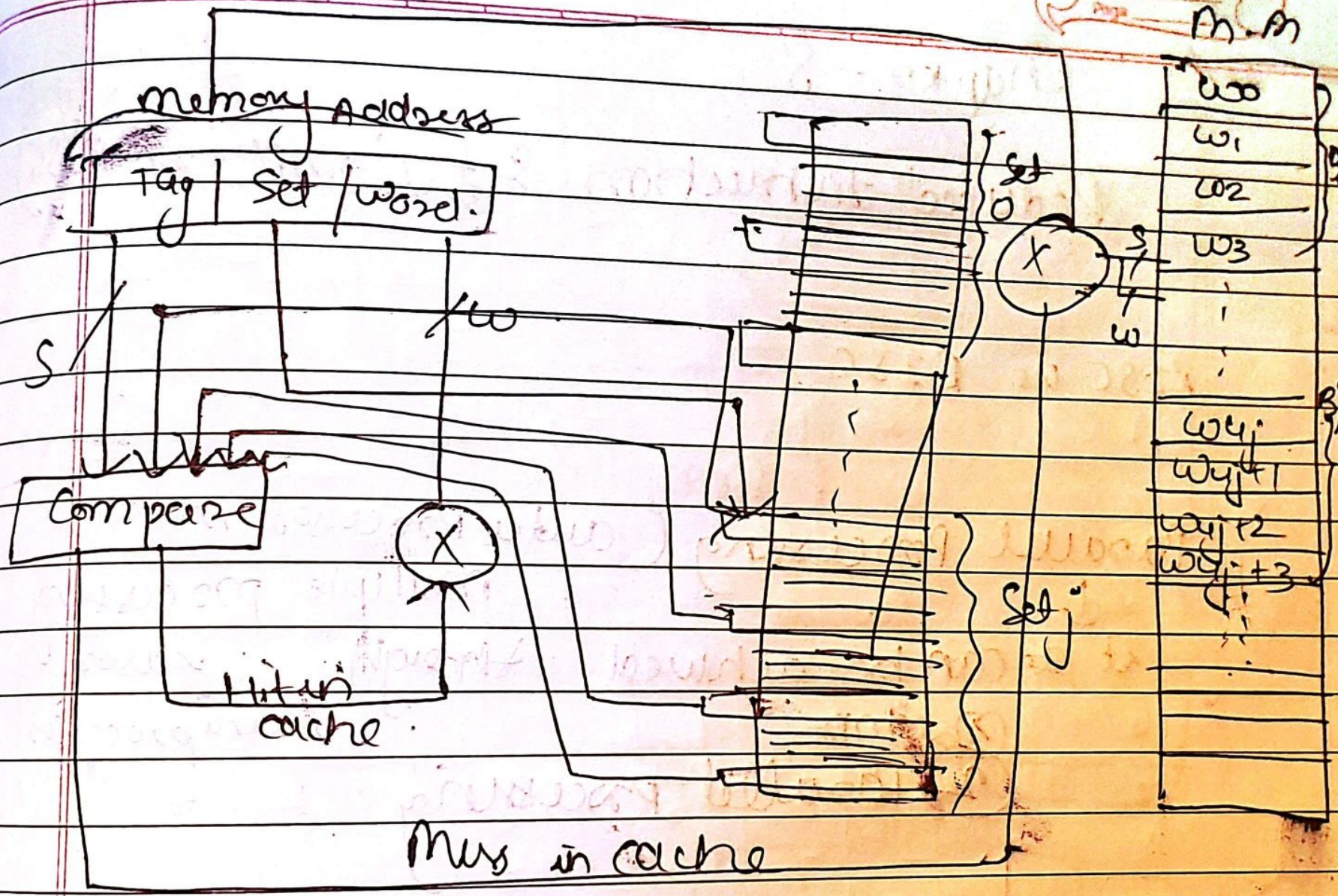
Hits in cache

fig Associative Memory

Advantages :- Fast due to associative memory
Flexible to replace block when new block is read into cache

* Set associative mapping

↳ Comprise both direct & associative mappings that builds on the strength of both.
Divide cache into a number of sets (V), each sets holding a no of lines (K). A main memory block can be stored in anyone of the K lines in a set such that set number = j modulo V. where j is the assigned main memory block



Set Associative Memory organization

Q.5a) What is pipelining hazard. Explain structural & data hazard and it's solution.

Ans// pipelining hazard's are the situations that prevent the next instruction in the instruction stream from executing in it's designated clock cycle. Hazards reduce the performance from the ideal speedup gained by pipelining.

A structural hazards occurs when two (or more) instructions that are already in the pipeline need the same resources. Structural ~~bo~~ hazards can be avoided by stalling, duplicating the resources or pipelining the resources or ~~renaming the resu~~ renaming the resources. Let illustrate it with examples:-

Clock cycle / instruction	1	2	3	4	5	6	7	8
I1	IF	ID	EX	MEM	WB			
I2		IF	ID	EX	MEM	WB		
I3			IF	ID	EX	MEM	WB	
I4				IF	ID	EX	MEM	WB

In clockcycle 4, in instruction I1, to load & store the value the value we are accessing memory and also in

instruction I_4, we are accessing memory to fetch the instruction. These two are occurring in the same memory. This results the conflicts. In this case either I_1 or I_4 will functions.

Data hazards occurs if an instruction reads a register that a previous instruction overwrites in a future cycle OR.

Data hazards occurs when an instruction depends on the result of previous instruction and that results of instruction has not yet been computed.

Consider four (4) stages of pipelining

Let

$R_3 \leftarrow R_1 + R_2$ I_1.

$R_5 \leftarrow R_3 + R_4$ I_2

(I_1) (R_1/R_2) + R_3(Result) / output

I_1	IF/ID	OF	EX	WB	
I_2		IF/ID	OF	EX	WB

IF/ID	OF	EX	WB

Here,

	Domain	Range
I_1	R_1 R_2	R_3

I_2 R_3 R_4 R_5.

now,

I_1 (Range) $\cap$ I_2 (Domain) $\neq \phi$ = Data hazards

$(R_3) \cap (R_3 R_4)$

R_3

Remove of Data Hazards

1) The 1st way is to give the responsibility software.

2) The 2nd way is to implement hardware to detect hazards if they exists and to insert a delay in the ckt.

3) The 3rd way is to notice that although R_2 does not contains valid, information all the time the next instruction wants to read it, the correct info is available

Q5b) What is cache cohesence problem? Explain ~~sft~~ software and hardware solution of cache cohesence problem.

Ans Already done at 1st set (same ques)

Q.4b) Define an interrupt with it's classes in detail. Also explain about interrupt driven I/O

Scanned with CamScanner

Ans Already done previously.

Q.6a Write about parallelism in computer system. Explain about symmetric multi-processor in detail.

Ans: A computer system achieves parallelism when it performs two or more tasks simultaneously. A system that processes two diff instructions simultaneously could be considered to perform parallel processing. If a system that performs different operations on same instruction is not considered as parallel processing. A parallel processing system can be achieved by having a multiplicity of functional units that perform identical or different operation simultaneously. The data can be distributed among various multiple function units.

The following diagram shows one possible way of separating the execution unit into eight functional units operating in parallel.

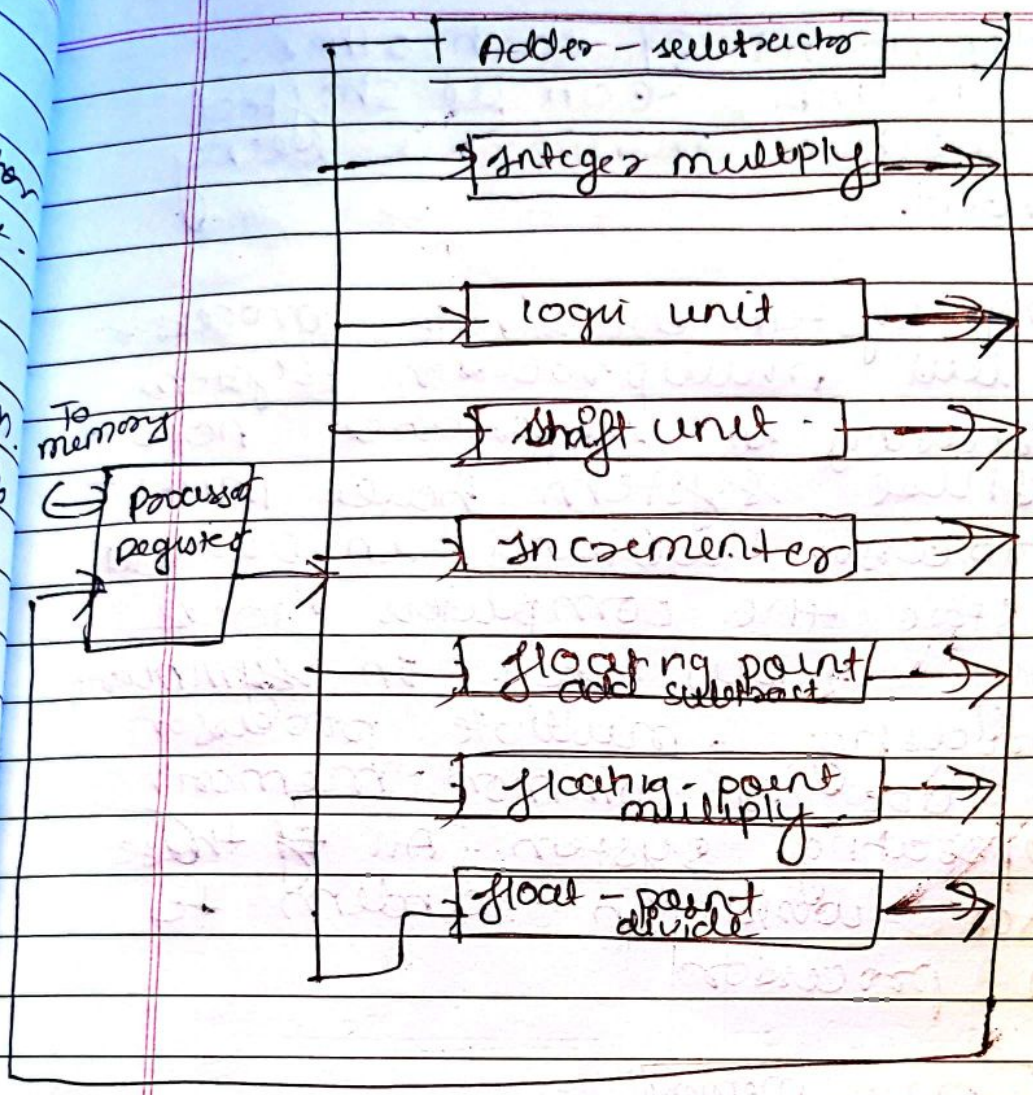
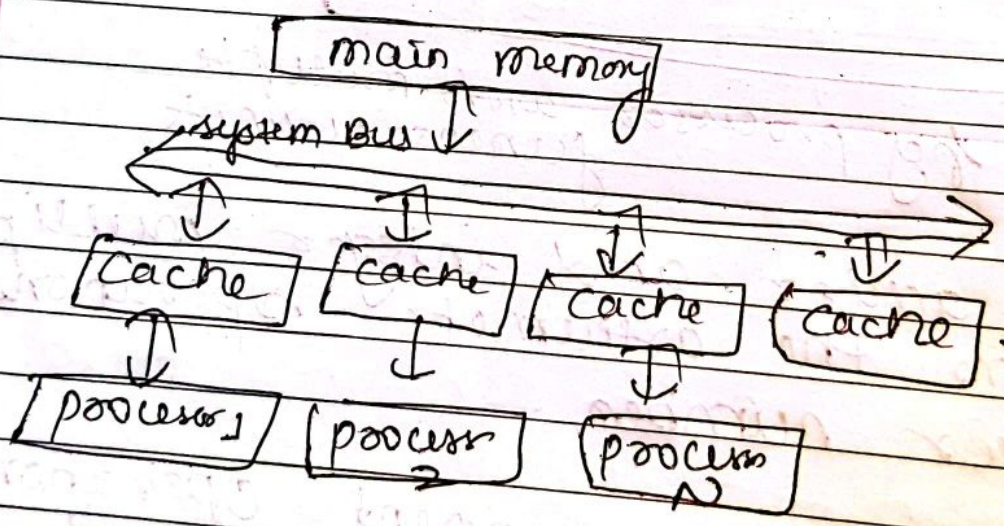


fig processor with multiple functional units

- The adder and integer multiplier performs the arithmetic operations with integer numbers.
- The floating point operations are separated into three circuits operating in parallel.
- The logic, shift, and increment operations can be performed concurrently on different data. All units

are independent of each other. so one number can be shifted while another number is being incremented.

most comp system are single processor system but multiprocessor system are increasing in importance now-days. These system have multiple processor working in parallel that share the computers clock memory, bus etc. In symmetric multiprocessing, multiple processor ~~share~~ share a common memory and operating system. All of these processor work in tandem to execute processor



Q.7 Write Short Notes on:-

(1) RISC vs CISC
→ done previously.

Scanned with CamScanner

② shift micro operations

(1) linear ~~microoperations~~ shift
(2) circular shift
(3) Arthmetic shift
(4) Decimal shift

① linear shift :- linear shift take place 1 position to the left or 1 posin to the right. the end bit is discarded and zero take place the vacant position example :- 1011
left shift 0110
Right shift :- 0101

② circular shift :- It works similar as linear shift but differnce is in the bit that is discarded in linear shift takes the vacant place
eg 1011
left shift 0111
Right shift 1101

③ Arthmetic shift
:- this was developed for the sign number notation. Here left most bit is sign bit that remains unchanged by shift operation

eg :- X = 1011

left shift = 1110
Right shift = 1101

(iv) Decimal shift :- It was developed for BCD representation. It acts like linear shift, accepts it shifts one digit ie 4 bits example

9

x = 1001
left shift = 0111
Right shift = 0000

7

0111
0000 = 70
1001 = 9

Scanned with CamScanner