DAY 5 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Processor Pipelining**

Homework: Understand Processor pipelining in detail. There are 5 basic steps – Fetch, Decode, Execute, Memory Access, Write back.

Day 5 Challenge:

1. Why do we need processor pipelining? What will happen if we don't use any processor pipelining at all?
2. Draw 5 stage pipeline & explain operation of each stage.
3. Consider a pipeline having 5 phases with duration as below:
   Fetch – 60 ns, Decode – 50 ns, Execute – 90 ns, Memory Access – 100 ns and Write back – 150 ns. Given latch delay is 25 ns.
   Calculate: -
   - Pipeline cycle time (Tip: Pipeline Cycle Time = Sum of time taken by each stage + Latch delay)
   - Non-pipeline execution time (Tip: Non-pipeline execution time = Sum of time taken by each stage)

Date
Page

DAY:6

Topic : PROCESSOR PIPELINING

→ what is pipelining ?

- pipelining is the process of arrangement of hardware element of CPU such that its overall performance is increased.
- means A way of speedup execution of instructions.

∴ So pipelining is not a process in which we are puchasing new hardware and implement them. NO we are already arranging existing hardware and CPU in a way so that performance should increase

- But now will the perfomance actually increase ?

∴ Simultaneous execution of more than one instruction takes place in pipelined processor. (means at one time not one instruction will be executed, multiple instruction will be taking place).

Key idea :

Overlap execution of multiple instructions.

∴ So what is the maning of overlapping ?

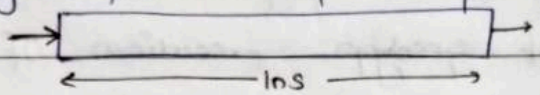
∴ at one time one process is executed completely after that the second process starts, that is know pipeline non pipeline (overlapping). (Non pipeline means one process started till it is not Complete, the second process connot even start.

- what is in pipeline is that if one process is running on a particular stage then the second process will run on any other stage in overlapping state and we call this thing Overlapping.
- if there is no stage then one process will start and when it is Complete then we will get the second process input. But if we divide multiple stage, stages we also call them Segments.
- then one process can be running on any stage 2, second process can run on stage 1, third process on stage 3 means different processes are running on different stages. So feel that not one process at one time but multiple processes executed due to that performance increases.

⊙ How to define these stages? How many stages can be there?

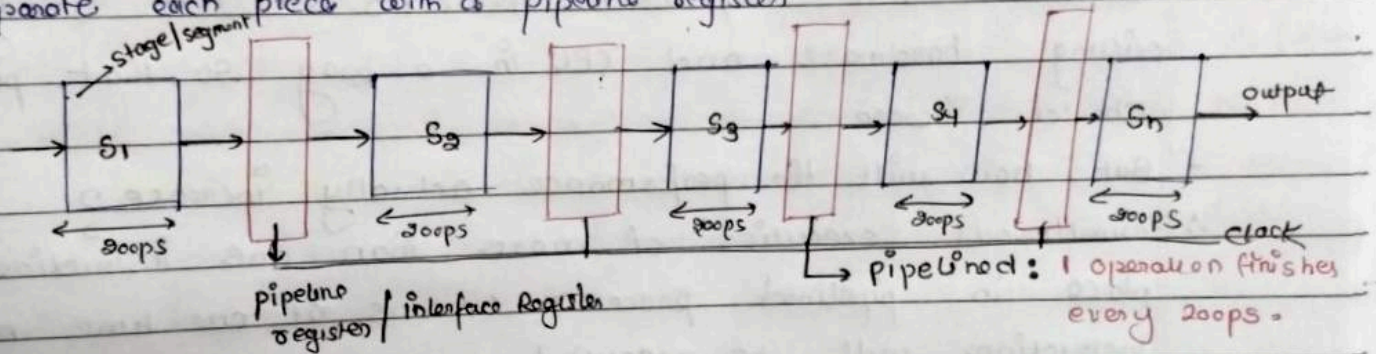
∴ It has simple example of this RISC (Reduced instruction set where we can use pipeline in 5 stages and in 5 stages instructions are executed.

- Break big computations up into pieces



Non-pipelined: 1 operation finished every 1ns.

→ Separate each piece with a pipeline registers



→ Benefit of interface register, it stores intermediate result. because the o/p one stage is input of the second stage.

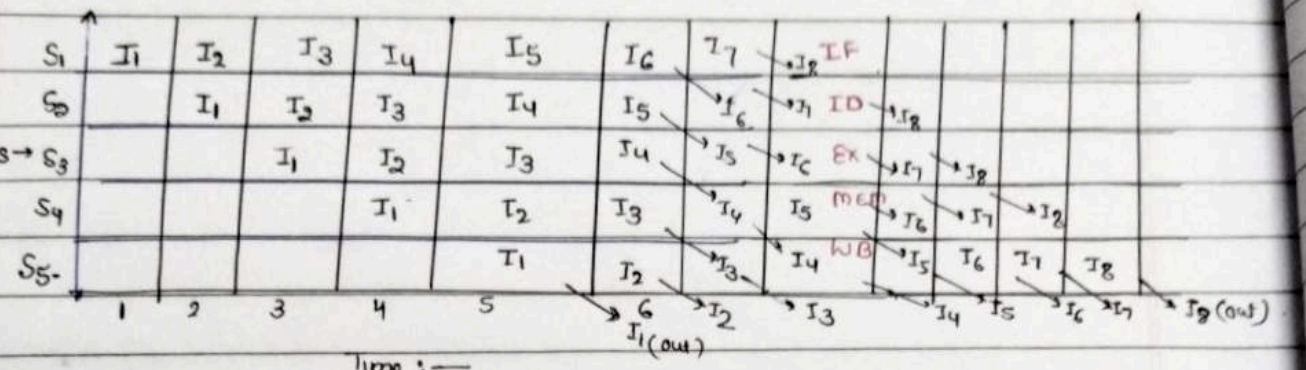
- The result ~~perform~~ obtained from each segment is transferred to next segment.

Sol^n ① why do we need processor pipelining? what will happen if we don't use any processor pipelining at all?

- pipeline was developed to help improve the efficiency of CPU chip. Each CPU is clock driven. or pipeling its basic sence is running multiple instruction at the same time through the CPU at the same time.
→ if we don't use this, longer execution time it quite low. yours CPU performances one instruction percycle, but each cycle is very long. what's more, most of hardware stays doesn't have anything to do at given time because the instruction has'nt finished being processed.

Soln ② Draw 5 stage pipeline & explane operations of each stage.

→ Every instruction goes through following steps when it executes in CPU.



→ and generally the RISC architecture works on 5 stages pipelining.

→ we have 5 stage, to complete every stage we have assumed that one clock cyb. is needed.

∴ 5-stages pipeline working :-

→ in first clk cycle, Instruction (I_1) will Complete stage 1. Stage 1 means we

Stage 1 : fatched instruction (I_1). at this time we can't do any other work because Instruction 1 is fetched.

Stage 2 : next clk cycle, we put it on decoding and at this time we fetched instruction 2. as soon as we put Instruction 1 on decoding in clk cyb 2. means its working on the second stage. so stage 1 is free.

Stage 3 : I_1 will reach, means in stage 3 execute. and at that time I_2 will reach in fetch. in decoding and I_3 Fatch

Stage 4 : next clock cycle I_1 in memory, I_2 is getting executed, we put I_3 on decoding & I_4 we fetched the next instruction.

Stage 5 : In next clock cycle, In we are writing Back I_1, I_2 come to we in memory & I_3 put on executing & I_4 getting decoded, I_5 we are fetching here. as soon as our time reached 5 clock cycle as the end of the 5th clock cycle is over I_1 came out by getting executed.

Consider a pipeline having 5 phase with duration as below.

Fatch → 60ns , Decode → 50ns , Execute → 90ns,

memory access → 100ns & write back → 150ns

Given latch delay is 25 ns.

Calculate :-

i.) pipeline cycle time (Tip: pipeline cycle Time = max of time taken by each stage + latch delay)

ii.) Non-pipeline execution time (Tip: Non-pipeline execution time = Sum of taken by each stage)

i.)

Stage①	IF	ID	EX	MEM	WB	Fetch Instruction
Stage②		F	D	Ex	MEM	WB Decode Instruction
Stage③			F	D	Ex	MEM execution WB
Stage④				F	D	Ex memory MEM WB
Stage⑤	←60ns→	←50ns→	←90ns→	←100ns→	←150ns→	write back

stage

← clock →

i.) pipeline cycle time = max (IF + ID + EX + mem + WB) + latch delay

⇒ max (60 + 50 + 90 + 100 + 150) + 25

⇒ max (150) + 25

Cycle time ⇒ 175nS

ii.) Non-pipeline execution time = (Sum of taken by each stage)

= (60 + 50 + 90 + 100 + 150)

execution time = 450nS

DAY 6 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Pipelining Hazards**

Homework: Deep dive into pipelining hazards, ways to resolve hazards & some examples of instruction sequences leading to such hazards. Also, understand different types of each hazard. E.g., Data hazard mainly has 3 types – RAW, WAW, WAR. Now, go through various techniques to resolve each of the pipelining hazard.

Day 6 Challenge:

1. What are different types pipelining hazards. Explain with examples?
2. What are ways to resolve different pipeline hazards? What are pros & cons of these resolution techniques?
3. Identify the pipelining hazards (also tell the sub-category e.g., Data hazard – WAW) in below sequence of instructions. Explain why these instructions are resulting in pipelining hazard.
    a. SUB R1, R4, R3
       ADD R1, R2, R3
    b. ADD R1, R2, R3
       SUB R4, R1, R3
    c. SUB R4, R1, R3
       ADD R1, R2, R3

antiyogi_CAO

DAY: 6

Topic : pipelining - Hazards :

① what are different types pipelining hazards. Explain with examples?

Hazards means problem. problem with pipeling we always talk about that the CPI should be 1. (CPI mean Clock per instruction).

i) Data Hazards → RAW (TRUE dependency)
→ WAR (anti dependency)
→ WAW (output dependency)

ii) Structural Hazards

iii) Control Hazards

∴ because of three Hazards, delay occurs and because of delay, our CPI = 1 does not come

① data hazards : attempt to use data before it is ready.
- An instruction's Source operand(s) are produced by a prior instruction still in the pipeline.

② Structural hazards : attempts to use the same resource by two different instructions at the same time.

③ Control Hazards : attempt to make a decision about program control flow before the condition has been evaluated and the new pc target address calculated.
- Branch and jump instructions, exceptions.

i) data hazards :- data Hazards are of three types:

i) RAW [Read after write] OR [TRUE / Actual DEPENDENCY]

example :- lets take we have instructions,

Instruction 1 ∴ here we write R2 → R2 = R2 + R3

∴ means R2 and R3 addition is saved in R2.

Instruction 2, R5 = R2 + R4 ∴ here R2 is Read

∴ here I am use R2 and add it from R4 finally result is in R5.
- here output of first instruction that will be depending on input of next instruction

II) WAR (write after Read) Hazard :- [also reffered as anti dependency].

- lets say we have an instruction I_1, in R_1 lets say we multiply R_2 and R_3. means we have multiple registers so that is based on addressing mode, that we have direct addressing mode or indirect addressing mode whatever we have
- in I_2, means whatever operand we will get here by adding them $R_4 + R_5$ and we are saving in R_2.

I_1 : $R_1 \leftarrow R_2 * R_3$ (Read)

I_2 : $R_2 \leftarrow R_4 + R_5$

∴ write after Read hazard means, I_1 Read the R_2 & R_3 data after read it will multiply that data and save in R_1.

- my input operand is depending on output of next instruction so that is anti-dependency. generally anti dependency does not create any issue. [means input of first instruction that is depending on output of next instruction that is anti dependency].

III) WAW - write after write [output dependency] :

Add : R_1, R_2, R_3

Add : R_1, R_4, R_5

↑ output, R_4, R_5 → input operand

∴ If you observe my output of both instruction that is depending on each other, what it means this is WAW (write after write dependency so as if output ~~depend~~ operand of first instruction that is having something common with output operand of next instruction then you can say there is WAW dependency.

- output of first instruction is depending on output of next instruction.

Q(5) what are ways to resolve different pipeline hazards? what are pros & cons of these resolution technique?

- Hazard & their resolution:-

• STRUCTURAL Hazards:

Arises from resource conflicts among instruction executing Concurrently.

- Same resource is required by two (or more) Concurrently executing instructions at the same time.

- Easy way to avoid Structural hazards:
  - duplicate resources (Sometime not practical)
- Example of Resolution of structural Hazards.
- A' ALU to perform an arithmatic operation & an adder to increment PC.
- seperate data cache & instruction cache accessed simultaneously in the same cycle.

- A Solution to Compiler Technique to Take data hazards
  ∴ A compiler can help eleminate some stalls caused by data hazards
  eg:- an instruction that use result of a load destination register should not Immediatly follow the load instruction.
- The Technique is called:
  Compiler - based pipeline instruction scheduling.

→ Hardware Techniques to deal with Hazards;

• Simple Solution
  - Stall pipeline

• pipeline Stall:

- A pipeline can be stalled by inserting a "bubble" or NOP.

Soln (3) Identify the pipelining hazards (also tell the sub-category e.g Data hazard - WAW) in below sequence of instructions. Explain why these data instructions are resulting in pipelining hazard.

(a) SUB R1, R4, R3 ← This is a WAW hazards.
ADD R1, R2, R3

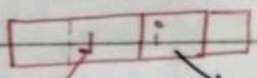
i: SUB R1, R4, R3 (write R1)
j: ADD R1, R2, R3 (write R1)

- instruction j tries to write an operand before instruction i write it.
- write are performed in wrong order.

(b) ADD R1, R2, R3 ← This is a RAW hazards.
SUB R4, R1, R3

- Assume instruction i is issued before j. j tries to read its operand before instruction i write it.

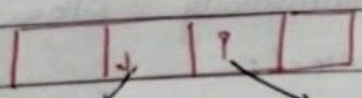
i: ADD R1, R2, R3 (write R1)
j: SUB R4, R1, R (Reading R1)



Instruction j is a read instruction issued after i.

Instruction j is a write instruction issued before j.

(c) SUB R4, R1, R3
ADD R1, R2, R3 ← This is a WAR Hazards.



instruction j is a write instruction issued after i

Instruction i is a read instruction issued before j.

i: SUB R4, R1, R3
j: ADD R1, R2, R3

∴ instruction j tries to write an operand before instruction j read it.

- instruction i instead of getting old value receive newer, undesired value.

DAY 7 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Memories**

Homework: Different levels of memory (Register, Cache, Main memory, Secondary memory, etc.), virtual memories, memory addressing & numerical, addressing modes, RAID system in SSDs, "Wait" memory state

Day 7 Challenge:

1. What are the different levels of memory? List the pros & cons of each level.
2. What is memory addressing technique? Explain the various addressing modes?
3. Address bus consists of 20 bits in a byte-addressable system. Find the size of memory.
4. Memory size is 8 GB in a 4-byte addressable system. Find the no. of address bits.
5. What is a **RAID** system (used in SSDs)? Explain with diagram.
6. Explain virtual memory with example? Why do we need them?
7. What is a ***wait*** state in memory? How can we deal with this state?

DAY : ①

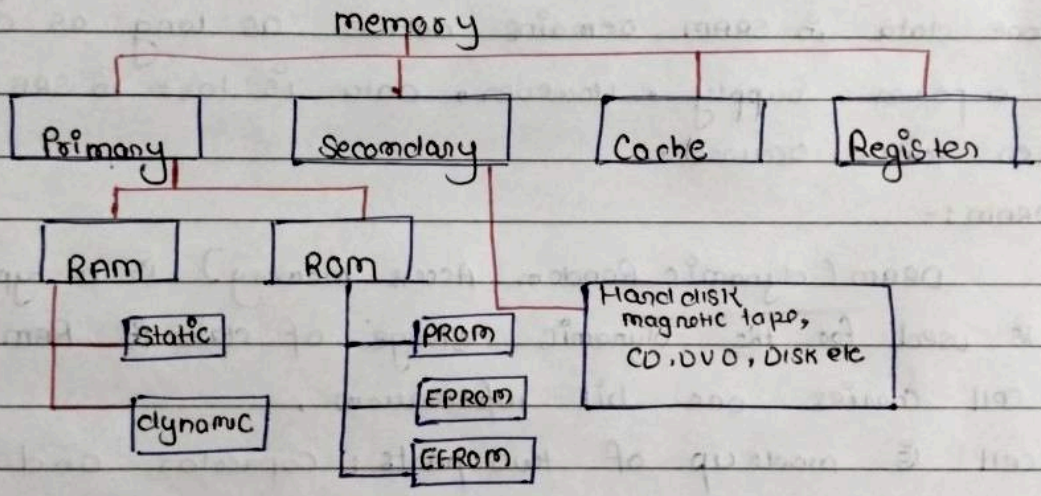
Topic : MEMORIES

⊙ MEMORIES :-

The Computer takes data as input and stores the data in the memory. as Computer memory is just like the human brain. It is used to store information and instruction.

Soln ① what are different level of memories? List the pros and cons of each level.

classification of memory :-



→ in general, Computer memory is of three types:
① Primary memory ② Secondary memory ③ Cache memory

① Primary memory:

→ primary memory is also known as the main memory that Communicates directly with the CPU.
→ it is used to store data and program or instructions during Computer operation.
→ primary memory is an essential Component of a Computer System.
→ program and data are loaded into to primary memory before processing.
→ CPU interacts directly with the primary memory to perform read and write operations.

There are two types of primary memory:
① RAM (Random Access memory) ② ROM (Read only memory).

1) RAM (Random Access memory):

- RAM is volatile memory that temporarily stores the file you are working on.
- when the computer is shut down, the memory is cleared until the process begins again.

∴ There are two Type of RAM:
① SRAM ② DRAM

① SRAM :- SRAM (static Random-Access Memory) is a type of RAM used to store static data in the memory. It means to store data in SRAM remains active as long as computer system has a power-supply. However, data is lost in SRAM when power failure occurred.

② DRAM :-

DRAM (dynamic Random-Access memory) is a type of RAM that is used for the dynamic storage of data in RAM. in DRAM, each cell carries one bit information.

- The cell is made up of two parts: capacator and transistor.
- The size of the capacitor & transistor is so small, require milions of them to store a single chip.
- DRAM is volatile. If Power is switched off, the data store in memory is lost.

2) Rom :-

- Rom is non-volatile memory, which means the information is permanently stored on the chip.
- Turning of the Computer does not have any effect on Rom.
- non-volatile memory cannot be changed by user.
- Information stored in the Rom in Binary format. It is also know as permanent memory. Rom is of four types.

① MROM ② PROM ③ EPROM ④ EEROM

i) MROM [masked Rom]: Hard-wired devices with a pre-programmed collection of data or instruction were the
- FIRST ROMs. masked Rom are a type of low-cost Rom.

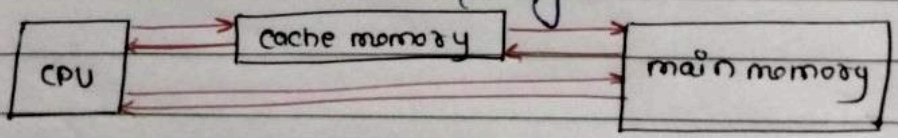
ii) PROM (Proggrammable real only memory):
- This read-only memory is modifiable once by the user.
- The user purchace a Blank PRom & use a PRom progam to put the required contents into the PROM. its content can't be erased once written

iii) EPROM (Erasable programmable Read only memory):
EPROM is an extension to PROM where you can erase the content of Rom by exposing it to Ultraviolet rays for nearly 40 minuts.

iv) EEPROM [Electrically Erasable programmable Read only memory):-
here the written content can be erased electrically. you can delet and reprogramme EEPROM upto 10.000 times.
- Erassing & programming take very little time.

② SECONDARY memory:-
- Secondary memory is a permanent storage space to hold a large amount of data.
- Secondary memory is also know as external memory that represent the storage media [hard drives, USB, CDs, DVDs] on which data the computer data & program can be Saved on a long term basis.
- it is slower and cheaper than the main memory.

③ Cache memory:
- Cache memory is a special very high-speed memory.
- It is serve as a Buffer. it is used to store the data & progra that the CPU uses the most frequently.



◉ PROS AND CONS OF each level:-
Cache memory is the fastest and most expensive, but it has a limited capacity & only used to store different frequently accessed data.

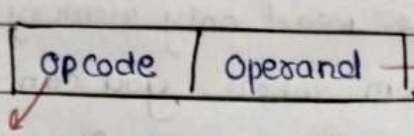
Ashi Tyagi

⑤ What are memory addressing Technique ? Explain The Various addressing modes?

- The addressing mode is the method by which an instruction operand is specified. The data stored in the operation code is operand value or the result.

i) The administrator or opcode that indicate what to do.

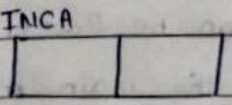
ii) The operands that portray the information to be utilized in the technique.



⑥ Types of Addressing modes:

i. implied mode ii) Immediate mode iii) Register mode
iv. Register indirect mode v) Autoincrement or Autodecrement mode
vi) direct Address mode vii) Indirect address mode viii) Relative mode
ix) Indexed Addressing mode

i) Implied mode: Implied mode is also called implicit mode
i) Operand is specified implicitly in the definition of instruction.
ii) Used for zero address and one address instructions.
iii) The operant is hidden / fixed inside the instruction.



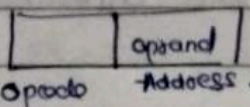
eg: Add R_1 R_2
$R_1 \leftarrow R_1 + R_2$

$AC \leftarrow AC + 1$

- Advantage: Benefit of implied mode is the size of instruction will obviously be less.

⑦ Immediate Addressing mode: In this mode is operand is specified in instruction itself.

- operand is directly provided as constant
- No Computation required to Calculate



Date
Page

This is constant

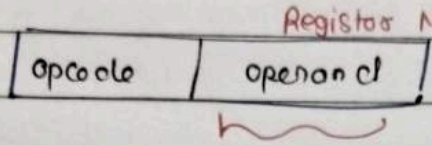
eg: add R_1, #3 , data

$R_1 \leftarrow R_1 + 3$

◉ Register mode : The operand is specified with in one of the processor register.

- instruction specified the register in which the operand is stored.
- means operand is present in the register.

Register No.

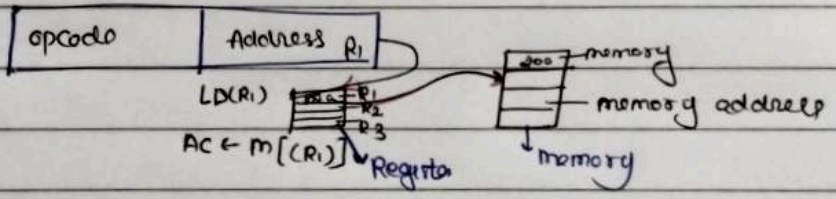


eg: LD R_1

$AC \leftarrow R_1$

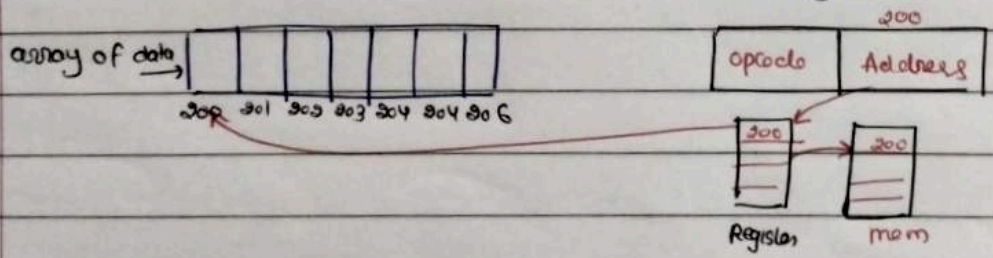
- Benefit is size of instructions will be small.

◉ Register indirect mode :- Register contain address of operand rather than operand itself.



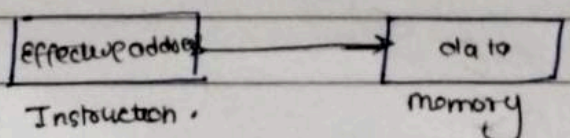
◉ Auto increment or Auto decrement Addressing mode :-

- Special case of Register indirect addressing mode.



◉ direct addressing mode :- [Absolute Addressing mode]

- The Balence of the operand is stated as an address or 16 cycle removal Component in the guidance. The 16-cycle usable location of the info is important for the guidance in this mode.



- use to access variable.

Sol^n ③ Address Bus Consists of 20 bits in a byte - addressable System. find size of memory.

Address Bus = 20 bits

Size of memory = $2^{\text{address Bus}}$

$= 2^{20}$ is 1MB

∴ So 20 bit address lines can address 1MB locations.

Sol^n ④ Memory size is 8GB in a 4-byte addressable System. find the no. off address bits.

memory size = 8GB ⇒ 8 * 1024 * 1024 * 1024 bytes

Address line = $\text{clog}_2(\text{DEPTH})$ OR $\text{clog}_2\left(\frac{\text{memory size}}{\text{word}}\right)$

$= \text{clog}_2\left(\frac{8 * 1024 * 1024 * 1024}{4}\right)$

$= \log_2(2 * 2^{30})$

$= \log_2(2^{31})$

Address line = 31

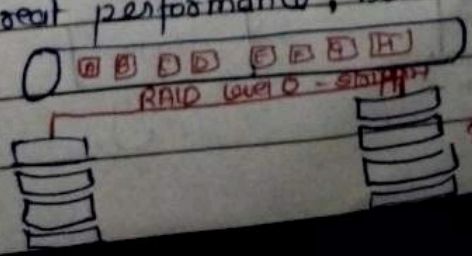
Sol^n ⑤ What is a RAID System (used in SSDs)? Explain with diagram.

- RAID is a technology that is used to increase the performance and/or reliability of data storage. The abbreviation stands for either Redundant Array of Independent Disks or Redundant Array of Inexpensive Disks, which is older & less used.

→ RAID level 0 - striping

→ in a RAID 0 System data are split up into Blocks.

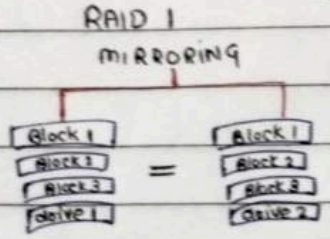
→ RAID 0 offers great performance, both in read & write operations.



- RAID 0 does not provide redundancy or fault tolerance.

② RAID 1 (mirrored disks) :-

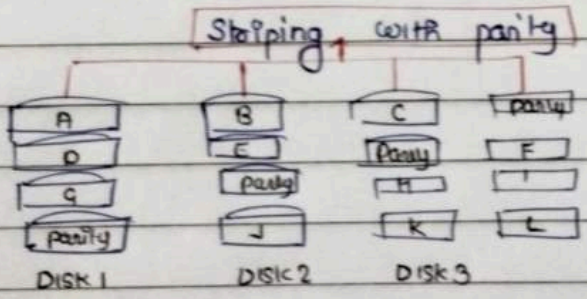
- you need at least 2 drives for a RAID 1 array.
- Data are stored twice by writing them to both the data drive (or set of data drives) & mirror drive (or set of drives.)
- disk are store exactly the same data, at the same time, & at all times. data is not lost as long as one disk survives.



③ RAID 5 (Striped disks with Single parity) :-

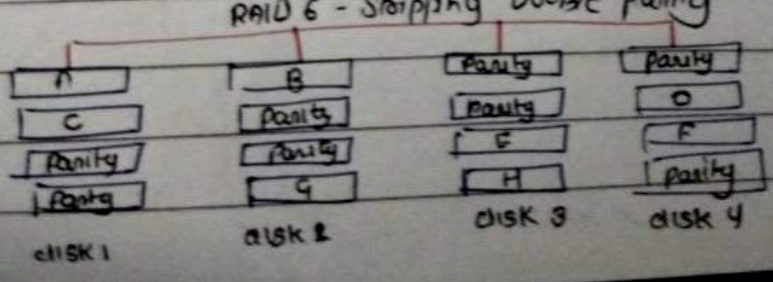
- Its requires the use of at least three drives but work up to 16.
- It Combines these disks to protect data against loss of any disk.
- The array storage capacity is reduced by one disk.

→ RAID 5 is the most common secure RAID level.



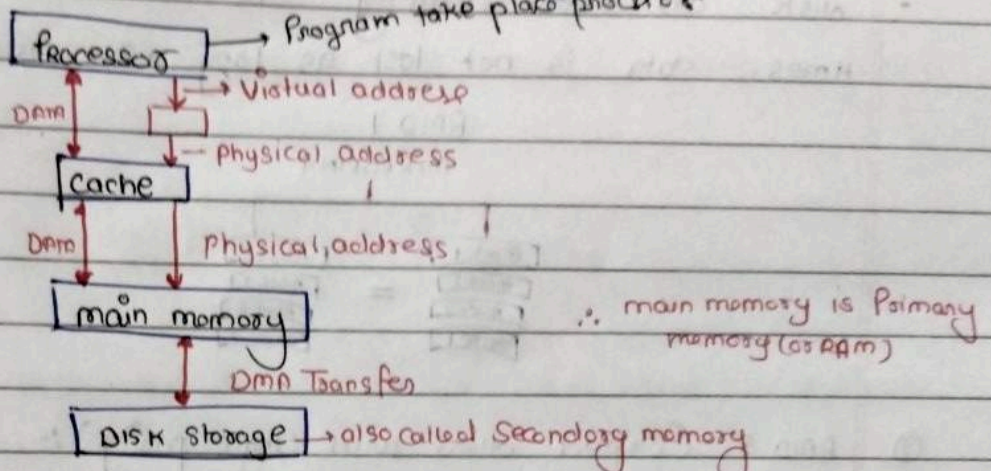
④ RAD level 6 - Striping with double Parity.

- Raid 6 is like RAID 5, but the parity data are written to two drives. That means it requires at least 4 drives & can withstand 2 drives dying simultaneously.
- If two drives fail, you still have access to all data, even while the failed drives are being replaced. So RAID 6 is more secure than RAID 5.

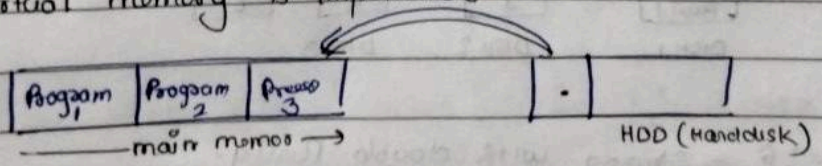


Q⑥ Explain Virtual memory with example? why do we need them?

- VIRTUAL Memory :- Virtual memory is partition of logical memory from Physical memory. This partition supports large virtual memory for programmers when only limited physical memory is available.

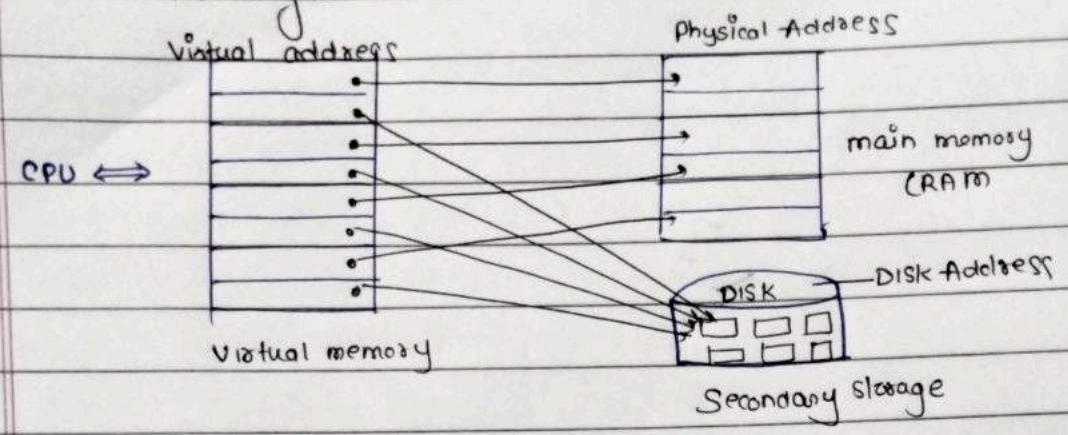


- It appears to be present but actually it is not. It provides illusion of large memory.
- Virtual memory Technique allows user to use more memory for a program than the real memory of a Computer.
- Virtual memory is the Concept that gives the illusion to the users they will have main memory equal to the capacity of secondary storage using.
- How the virtual memory is implemented :-



- To implement virtual memory, a portion of Harddisk (HDD) is reserved by the System. This portion can be either file or separate partition.
- When the System needs more memory (RAM), then it transfers some of its data from main memory (RAM) to hard disk drive.
- To extra memory does not actually exist in RAM, but its the storage space on the disk space, which is implemented with the help of swapping between main memory & hard disk.

⇒ Virtual memory :-



- Virtual address ⇒ address used by the programmers
- Physical address ⇒ actual physical (or main) memory address (or locat

→ CPU access virtual address which is translated into physical address, using a mapping table.

Need of virtual memory :-

- Virtual memory is a imagenary we are assumming if we have a program that exceeds the size of main memory then we need to use the Concept of virtual memory.
- It is a temporary memory which is used along with the RAM (or main memory) of the system.
- we want to run multiple processes at the same time.
- Virtual memory might Contain twice as many address as main memory.

Soln ① what is a wait state in memory ? How can we deal with this state ?

→ A wait state is a delay experienced by a computer processor when accessing external memory or another device that is slow to respond. Each of the cycles spent waiting is called wait states.

- during this period, the computer slows down and it can create a Bubble, where a glitch in communication delay all commands until the processor can sort it out.
- To deal with this, Some modern CPU's have integrated memory Controller that offer improved memory access & reduced the wait state.
- To minimize wait state cache, instruction pre-fetch and pipelines can be used.

DAY 8 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Different Computer Architectures**

Homework: Types of computer architectures, Von Neumann and Harvard architecture, RISC, CISC, RISC-V, SIMD Architecture for GPU.

DAY 8 Challenge:

1. Explain 3 major types of computer architecture – System Design, ISA (Instruction Set Architecture) & Microarchitecture.
2. Explain Von Neumann and Harvard architecture in detail. List down pros & cons of Von Neumann and Harvard architecture.
3. Explain RISC & CISC Architecture in detail. List down pros & cons of RISC Vs CISC?
4. What is a RISC-V processor? Why do we use RISC-V?
5. What is SIMD architecture in GPU? What are the advantages of using SIMD over SISD?

Adityagi - COA - 2023

#100 DAYS VERIFICATION CHALLENGE

DAY : (8)

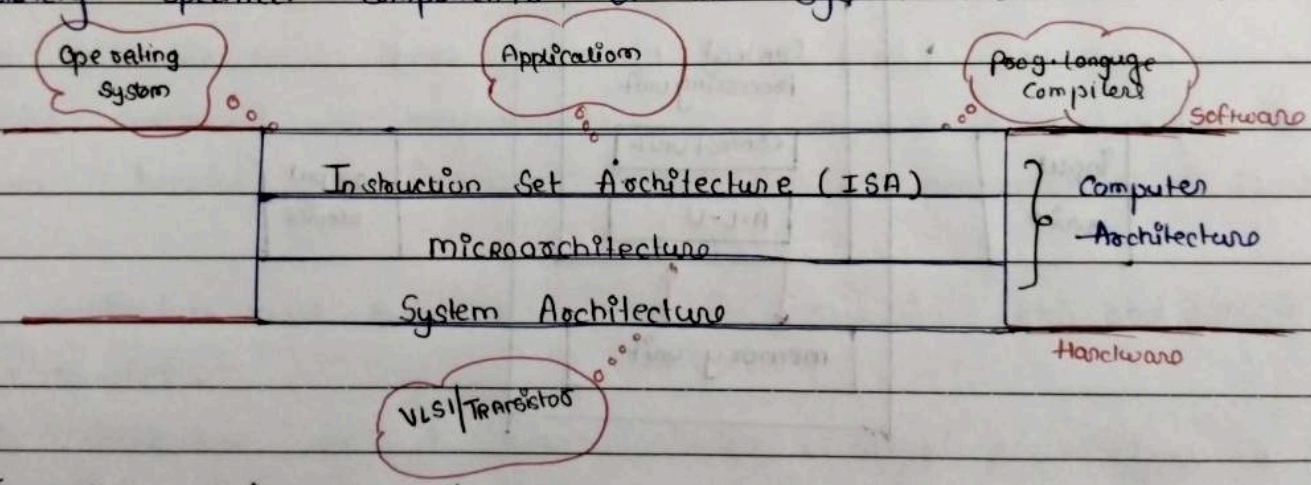
Topic : DIFFERENT COMPUTER ARCHITECTURES

① Explain 3 major types of COMPUTER ARCHITECTURE - System design, ISA (Instruction Set Architecture) & microarchitecture.

→ COMPUTER ARCHITECTURE :-

→ It is set of Rules and methods that describe the functionality, organization, and implementation of Computer Systems.

→ The architecture of a system refers to its structure in terms of separately specified components of that system and their interrelationship



i.) Instruction Set Architecture (ISA) :-

- ISA is set of computer instruction provided to programmers to implement software
- An abstract model of a computer
- ISA is implemented in microarchitecture.

ii.) Microarchitecture (uarch) :-

- The actual implementation of ISA in processors. e.g adders & multipliers.
- uarch is about ensuring instruction run correctly and quickly.
- Intel and AMD have to implementation of x86 ISA, but two uarchs
- uarch is implemented with transistors.

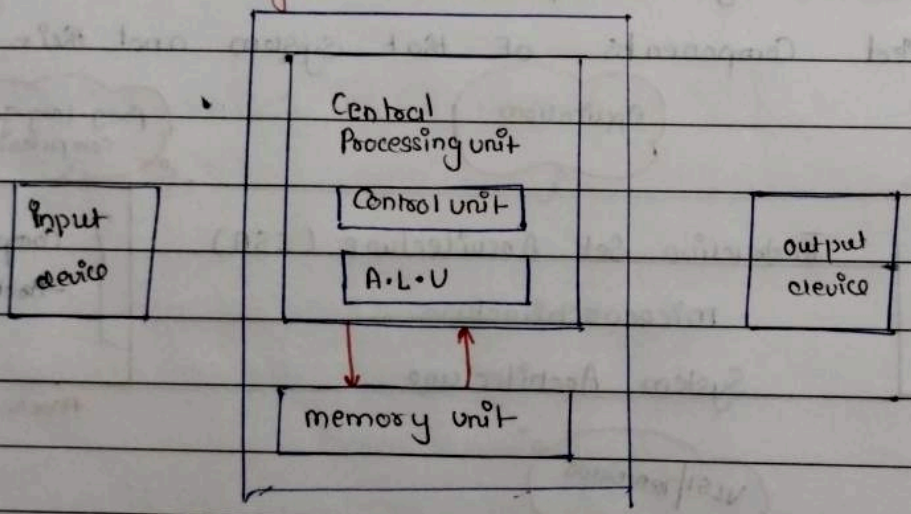
iii.) System architecture (Sys-arch):-

- Any hardware implementation beyond processors, such as memory and I/O devices.
- Sys-arch is about the connecting processors and then devices.

Sol^n→ ② Explain Von Neumann and Harduard architecture in detail. List down pros & cons of von neumann & Harvard architecture.

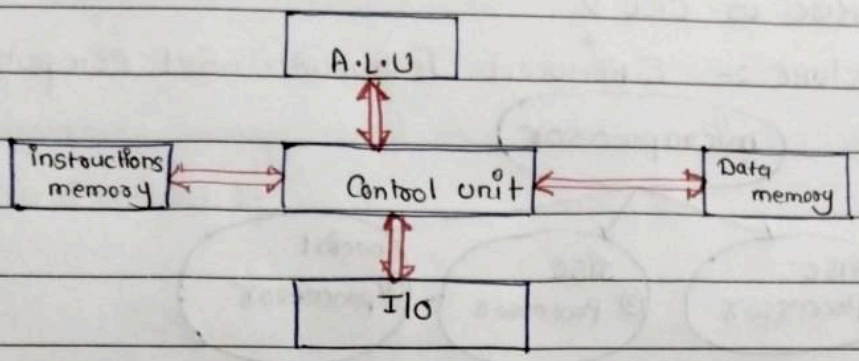
① VonNeumann Architecture :-

Block diagram :-



- Von neumann Architecture is a Theoritical Computer design based on the Concept called Stored-program Concept.
- it is used the same physical memory address for instruction and data.
- processor need two clock cycles to execute execute an instruction.
- Simple Control unit design
- Data transfer and instruction fetches cannot be performed simultaneously.
- This architecture used only one bus that is used for both instruction fetches & data transfer.
- This architecture used in personal Computer, laptop & work stations.

→ Harvard Architecture :-



- Harvard Architecture is a modern Computer Architecture based on the harvard mark - I relay - based computer model.
- It uses seprate memory address for instruction and data.
- processor needs one clock cycle to execute an instruction.
- Control unit for two buses is more complicated.
- Costlier Compared to VonNeumann.
- data transfer and instruction fetches can be performed at the same time.
- This architecture we two buses one for instruction fetch and other for data transfer.
- This architecture used in micro-Controller, signal processing.

⊙ Pros & Cons of Harvard Architecture :-

Pros :-

- Since there two diffrent buses, there is no need of scheduling the instruction & data like in von-Neumann architecture.
- Harvard Architecture allows a Simultaneous access to both the code as well as data.

Cons :- Complexity in Architecture.

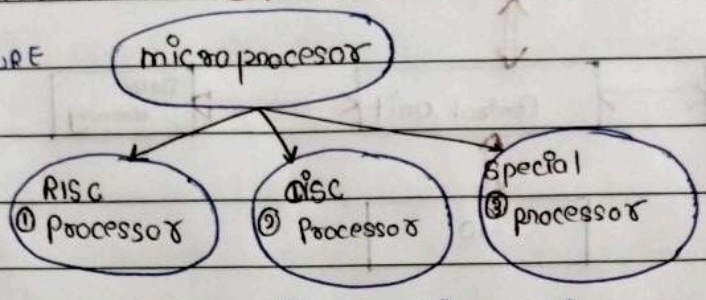
- This architecture is not widely used but modified (variation of Harvard Architecture).

Sol^n ③ Explain RISC and CISC Architecture in detail. List down pros & Cons of RISC vs CISC ?

: RISC Architecture :- [Reduced Instruction Set Computer]

:- micro processor Architecture

Types :→



RISC : It is design to reduce the execution time by simplifying the instruction set Computer.

∴ each instruction requires → only 1 clock cycle to execute the result in uniform execution time.

(fetch, decode, execution the instruction) this is called as 1 clock cycle.

one clock cycle = fetching + decode + execution

⊙ Why RISC ? :- ⊙ Reduce processor efficiency

..... more line of code.

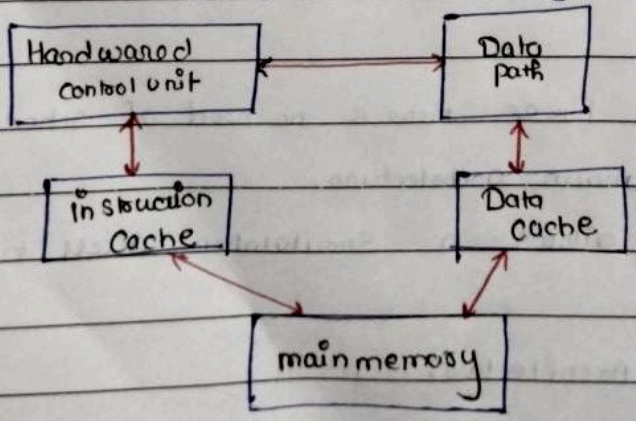
-- -- more RAM needed to store code

⊙ Compiled load increases

---- need to work more to high level language to machine level language

⊙ used, apple i-pad, tablets, smartphone.

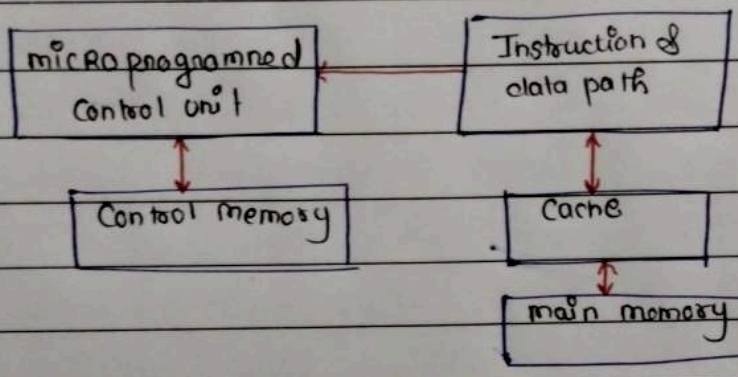
∴ RISC Architecture :- It used highly optimized set of instruction it used in Portable device like smartphone & tablets.



∴ datapath means where you have data & where you have fetch the data.

⊙ CISC PROCESSOR : [Complex Inst Set Computer]

- CISC is design to minimize to the no. of instruction per program and ignoring the no. of cycles per instruction.
- CISC processor mostly used in desktop, laptop Computer.
- CISC Architecture :-



- Very little RAM is required to store instruction because here the code length is relatively short that's why little RAM Required.

: characteristics of CISC :-

i.) Variety of addressing mode used
ii.) large no. of most required
iii.) Several cycles may be required to execute 1 inst.

⊙ PROS & CONS of RISC & CISC :-

⊙ FOR RISC PROS :	FOR RISC CONS
i.) RISC architector set of instruction, high level language compile can produce more efficient code. ii.) Speed operation maximized & execution time minimized.	i.) most of the performance depend on the programmer. ii.) while rearranging the CISC code to RISC code terms as code expension will increase the size.
⊙ FOR CISC PROS	FOR CISC CONS
i.) micro programming is easy assembly language to implement. ii.) Easy of micro coding new instruction.	i.) Performance of the machine slow down due to amount of clock time. ii.) only 20% existing instruction used in a typical programming event.

Sol^{n} ⑷ What is RISC-V processor? Why do we use RISC-V?

⊙ RISC-V:

- open-Source Hardware
- Instruction Set Architecture
- 32, 64, 128 Bits. CPU family depending on the ~~instructio~~ industrial standard IS.

:- RISC-V is an open source instruction set architecture (ISA) that can be used to design a wide range of processors for different applications. The ISA is designed to be simple, extensible which allows for easy customization and adaption to specific needs.

- The RISC-V ISA is available to anyone under a free & open-source licence, which means that anyone can design, manufacture & sell processors based on the RISC-V architecture without paying any licensing fees.

:- • RISC-V is used in embedded systems, artificial intelligence & machine learning.

• RISC-V is appropriate to used in some particular fields like edge computing AI & storage applications.

Sol^{n} (5) what is SIMD architecture in GPU ? what are advantage of using SIMD over SISD ?

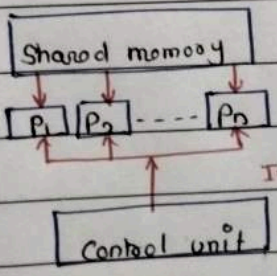
- SIMD stand for Single Instruction, multiple data elements.

∴ Single instruction is all processing unit execute same instruction at any given clock cycle.

∴ multiple data, each processing unit can operate on a different data element. (means single instruction operates on multiple data elements.

- It is used in Vector Super Computer.
- SIMD Computer has single control unit which issue one instruction at a time but it has multiple ALU's or processing unit to carry out on multiple data set simultaneously.

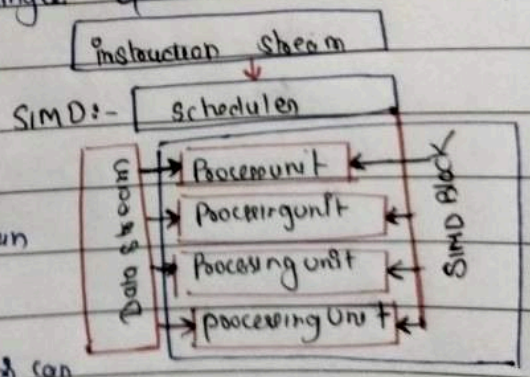
SIMD Architecture :-



- as opposed to SISD ie single instruction, single data corresponding to the traditional von Neumann architecture. its a parallel processing technique exploiting data level parallelism by performing a single operation across multiple data elements simultaneously.

: advantage of using SIMD over SISD :

- SISD, This is the simplest & most common type of computer architecture it is easy to program & debug & can handle a wide range of applications.
- SIMD, this type of architecture is highly parallel & can offer significant performance gains for application that can be parallelized.



DAY 9 – 100 DAYS VERIFICATION CHALLENGE

Topic: Instructions & Buses

Homework: Deep dive into instruction – various fields, rules of assembly language. Three buses – address, data & control.

Day 9 Challenge:

1. Can you tell at least 5 common rules of assembly language?
2. Explain 3 major fields in an instruction: the operation field, the address field, and the mode field.
3. Explain types of micro-operations (Register transfer, Shift, Logic, Arithmetic) with example?
4. What is a horizontal microcode? Why do we use it?
5. Explain 3 major types of buses – address, data & control bus with an example.
6. Give one example of below instructions: -
    a. Arithmetic Instructions.
    b. Branch Instructions.
    c. Data Transfer Instructions.
    d. Logic Instructions.
    e. Bit-oriented Instructions.

anitityagi_COA2023
#loadaye Verification

DAY : (9)

Topic : Instructions & BusES.

Sol^n ① Can you tell at least 5 Common rules of assembly language ?

: Assembly languge is also called middle level language

:- **Rules of the assembly language :**

i.> each line of assembly language Program is arrange three colums called fields.
i.> Label field ii.> instruction field iii.> Comment field.

The basic unit of Assembly language program is a line of code

ii.> It must consist of statements, one per line

iii.> Statement are either instructions or assembly directive.

iv.> each assembly language may be have specific directives to control the assembly process, define controls & allocate memory.

v.> Address mode: it use different addressing modes to access memory or ~~Registers~~ registers which specify the location of the data that is being operated upon.

Sol^n ② Explain 3 major fields in an instruction: The operation field, The address field, & The mode field.

:. **what is instruction :-** Progoam means A Sequence of Instructions.

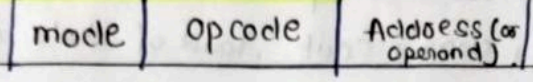
:. So instruction is a Command given to the Computer to perform a Specific operation.

:. An instruction is a group of bits (binary code) that instructs the computer to perform a specific operation.

- The purpose of an instruction is to specify both operation to be peformed by CPU and the set of operands (or data).

- operation : include add, sub, mul, shift etc. Operations are performed on data (operands).
- Operands include the data and the result that are produced.

→ The bits of the instructions are divided in to groups called fields.



Instruction format (with mode field)

⇒ The most common fields in instructions format are :-

① Opcode (operation code):

An opcode (Operation Code) field specified the operation to be performed such as add, subtract, multiply, shift, complement etc.

② Address field:

An address field specifies a memory address or processor register, where operand is stored.

③ Mode field :-

A mode field specifies the way the operand or the effective address of the operand is determined (or located).

⊙ NOTE : what is Instruction format :

∴ an instruction format is a binary format which specifies a Computer instruction.

Soln ⑧ Explain Types of micro-operations (Register transfer, shift logic, Arithmatic) with Example ?

- **micro-operations:**

micro-operations executed on data stored in registers are knows as microoperation.

eg: shift, clear, cout, load of data means any operation.

:- **TYPES OF MICRO-OPERATION:**

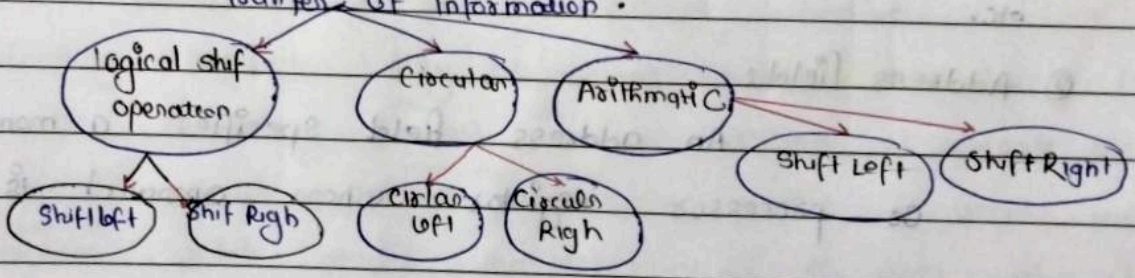
i) DATA Transfer → Register / memory

iii) Logical micro Operation

ii) Arithmatic micro-opration

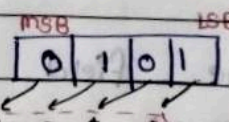
iv) Shift micro-operation

i) **shift micro-operation** :- shifting of bits. OR that are used for serial transfer of information.



i) **logical shift operation :-**

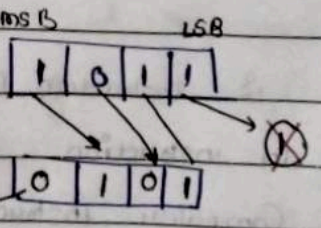
Logical shift left :-



← here bit are moving left side.

discarded → 1 0 0 (filled with 0 (LSB))

Logical Right shift:



⑨ **Registor Transfer :** Transfer data from reg to reg.

eg: $R_2 \leftarrow R_1$ (transfer data one Register to another

② logic micro operation :

∴ List of logic micro operations : 16 different logic operation with 2 binary variable.

16 logic micro operations obtained :

Boolean function	microoperation	Name
$F_0 = 0$		Clear
$F_1 = x \cdot y$	← AND →	$R_3, R_1 P_2$.

Similar Performed like AND, OR, NOT, EXOR etc,

iv.) Arithmatic microoperation :-

There are several operation, addition, Sub, mul, division, increment, decrement. These micro-operations are use in Combination with logical micro-operation likeas AND, OR and NOT to perform more Complex Calculations & manipulate data with CPU.

eg: for Subtraction $R_3 \leftarrow R_1 - R_2$, increment $R_1 \leftarrow R_1 + 1$, Complement $R_1 \leftarrow \bar{R}_1$

Sol^n ④ what is horizontal microcode ? Why do we use it ?

→ in horizontal microcode, each microoperation is represented by one bit in each microinstruction.

- Horizontal microcode is generally included a fairly wide Control Save it is not exceptional for each word to be 56 bits or more. on each micro click of a Sequencer clk, a microcode word is read, decoded & used to Control the functional Components which is create up the CPU.

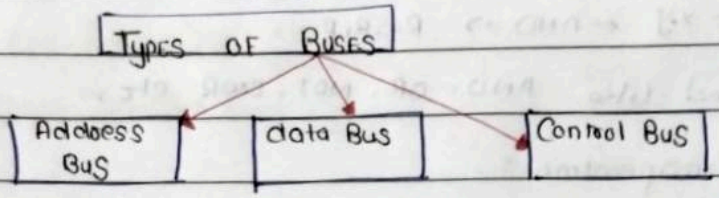
:- used for : • Faster execution.

• Provides direct Control over CPU's Components

Sol⁷ ⑤ Explain 3 major types of Buses : Address, data & Control Bus with example.

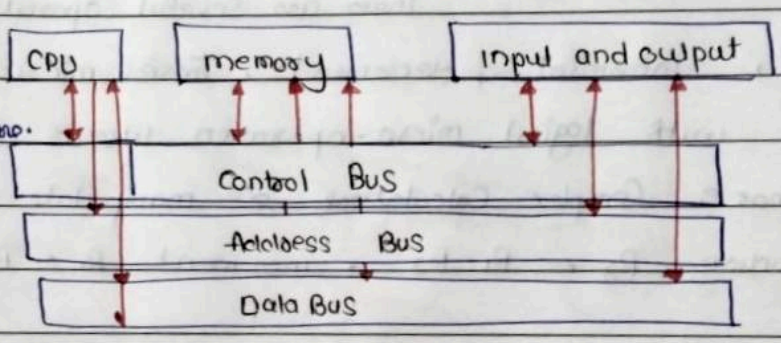
BUS STRUCTURE: • Bus is a group of wires (or lines) that connects several devices with in a Computer System.

- Each wire/line can transfer one bit (1/0) of information.
- Bus carries data, address & Control information.



BUS STRUCTURE:-

- Single Bus Structure
- Bus - multi Busstructure.



① **Address Bus : Unidirectional** : group of wires which carries address information bits from processor to peripherals.

- Address bus width determines the maximum memory capacity.
- for eg: if address line = 8bit $2^3 = 8$. ie. 3 address line is required to select 8 location.

in general $2^x = n$ where x number of address lines (address bit) & n is location.

② **Data Bus:- Bidirectional** : group of wires which carries data information bit from Processor to peripherals & vice-versa.

- move data/instruction B/w CPU & Peripherals.

③ **Control Bus:- Bidirectional**: Group of wires which carries Control Signals from Processor to peripheral & vice. versa.

- Control the access to and the use of address line & data lines.
- Control Signals like memory read/write/ I/o read/write etc.

⑥ Give one example of below instructions :-

ⓐ Arithmatic Instructions ⓑ Branch Instructions ⓒ data transfer ⓓ logic instructions

ⓔ Bio-oriented instructions.

⊙ An instruction is a command given to the microprocessor to perform a specific operation on data. Basically we are having five groups of instructions.

① ARITHMATIC Instructions : Arithmatic Instruction perform arithmatic operation like

Instruction		Operation	Instruction	Operation	Instruction		operation
① ADD	R (Register)	A ← A+R (Accumulator)	ADC R	A ← A+R+CY	INR	R	R ← R+1
② ADD	M (memory)	A ← A+M	CY=1	A=51H	INR	M	M ← M+1
Ex: A=30H	M=HL	A H + 30 → A	ADC M	A ← A+M+CY	DCR	R	R ← R-1
③ ADI	8bit data	A ← A+8bit data	CY=1	A ← 8+8bit data+CY 8bit data	DCR	M	M ← M-1
↓ Add immediate data		↓ accumulator A2H	ADC		INX	sp	sp ← sp+1
Ex = A30H	A2H	30 / 02	CY=1		DCX	sp	sp ← sp+1

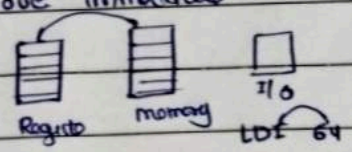
② data transfer instruction :- These instruction transfer or copy data from one register to another regeter or from memory to register & vice-versa.

eg: MVI R, 8bit data : move immediate

MVI B 64H

mov R1 R2

mov R, 64



③ Logic Instruction : Computer works on 0 & 1 when we want to work on particular bits & set of Bit, we us logical instructions or

- Thes instruction perform various logical operations with the content of accumula like, AND, OR, XOR, Rotate, Compare, Complement.

eg ANA R (AND, Accumulator, Register)

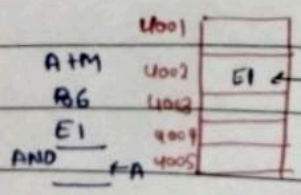
A → 10110110 (B6)H

C → 01001010 (4A)H

A ← 00000010

Store in accumulator

ANA M (AND, Accumulator, memory)



ANI 8bit

ANI 05H

A = 10110110

00000101

00000100 (04)H

④ BRANCH CONTROL Instructions:-

This group of instructions alters of the sequence of Program execution either Conditionally or unconditionally

Jump Instruction:- JMP, JZ, JNZ, JC, JNC, JAE, JPE, JPO, JP

Call, Return & Restent: CALL, RET

Conditional jump instructions.

⑤ Bio - oriented Instruction:-

→ Bio - oriented instruction are instruction that perform logic operations on Single bits. They are similar to logic instruction, but perform operations on Single bits.

ANL C, : AND Complements of direct bit to the Carry flag.

CLR C: Clear the Carry flag too.

SETB 0D5H : Set to 1 bit.

DAY 10 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Microprocessor, Microcontroller, Interrupts**

Homework: Deep dive into Microprocessors, Microcontrollers & Interrupts

Day 10 Challenge:

1. Explain the common components of a microprocessor (IO unit, Control Unit, ALU, Register, Cache)?
2. Difference between microprocessor & microcontroller?
3. What distinguishes a microcontroller's timer from its counter?
4. What is program counter?
5. Explain Tri-state logic.
6. What are the different types of interrupts in a microprocessor system?
7. What's the difference between interrupt service routine and subroutine?

Adityagt
#100daysVerification

DAYS : (10)

Topic : MICROPROCESSOR , MICROCONTROLLER , INTERRUPTS.

Sol^{n} ① Explain the common components of a microprocessor (IO unit, Control unit ALU, Register, Cache)?

⇒ microprocessor typically consists of :

① Register : Temperary storage location for performing instruction or data.

② ALU : Arithmatic Igicc unit performs arithmatic & logical operation

③ Timing & Control unit : Keeps all other parts of System (reg, ALU, memory & I/O) working together in right time the Seq.

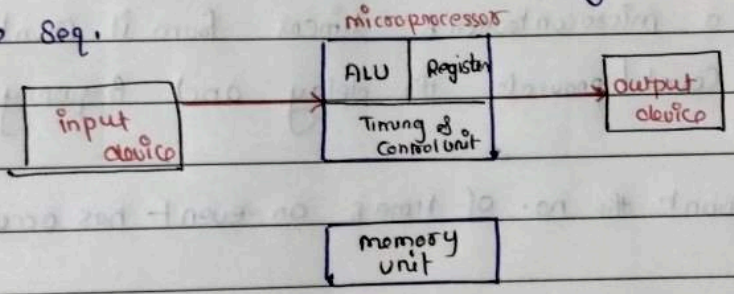


fig: microprocessor Component

④ I/o unit : The processor needs to be able to Communicate with the rest of the Computer System. The Communication occurs through the I/o ports. The I/o ports will interface with the System memory (RAM).

⑤ Cache :-

most CPUs manufactured do not have any cache. Cache is memory that is located on the chip, but that is not Considered registers. The cache is used because reading external memory is very slow. & reading a local cache is much faster.

Soln ② Difference Between microprocessor & microcontroller ?

microprocessor	microcontroller
① microprocessor are widely used in Computer Systems	① microcontroller is widely used in embedded System.
② it consume more power	② it consume less power than a uprocesor
③ has a high clock Speed	③ has a lower clock speed.
④ uses USB, UART, & high speed Ethernet as peripheral interface.	④ uses I2C, UART & SPI for the peripheral interface.

Soln ③ what distinguishes a microcontroller's timers from it. Counter ?

→ Timers are uses to count / provide the delay and frequency of operation in a micro controllers.

→ Counter are used to count the no. of times an event has occurs in a micro controller.

→ Timers used frequency of internal clock to generate the delay.

→ Timer is used to measure internal clock cycles, whereas the counter counts external events.

Soln ④ ~~Prog~~ what is program Counter ?

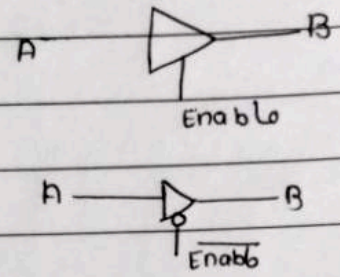
→ A program Counter is a register in a Computer's central processing unit (CPU) that contains the address of the next instruction to be executed from memory. its also known as instruction counter.

Sol^n ⑤ Explain Tri-state logic.

⇒ Tri-state logic used in digital Electronic that allows an input or output have 3 state; ① Logic high ② logic low ③ High impeance state.

- high impedance state effectively removes the output from the circuit.

input		output
A	B	C
0	0	high impedance
1	0	high impedance
0	1	0
1	1	1



Sol^n ① what are difference Between Interrupt Service routine & Subroutine?

⇒ • Subroutine is function that runs when you call it.

• A subroutine performs a task that is required by the calling program.

• Interrupt Service routine may not have anything in common with the program it interrupts.

• initialized by execution of some instruction in subroutine.

• in Interrupt, initialized by some external or internal signal.

• The CPU hardware automatically saved & restores all register.

• with Subroutines, the programmer has to write code for saving & restoring in general, not all registers are saved.

DAY 11 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Computer Architecture Miscellaneous**

Homework: DMA, MESI protocol, Synchronization, Paging

DAY 11 CHALLENGE:

1. Explain DMA (Direct Memory Access) with diagrams.
2. Explain operation of MESI Protocol (Cache Coherency)
3. What is meant by Synchronization in computer architecture?
4. What is Paging? What is a Page Table?
5. Find the total number of frames. If a system, size of the main memory is 230 bytes, the page size is 4 KB and the size of each page table entry is 32-bit.
6. Consider a system with page table entries of 8 bytes each. If the size of the page table is 256 bytes, what is the number of entries in the page table?
7. Consider a machine with 32-bit logical addresses, 4 KB page size and page table entries of 4 bytes each. Find the size of the page table in bytes. Assume the memory is byte addressable.

Abhilyagi_COA_2023
#100days Verification

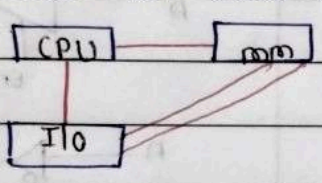
DAY : (11)

Topic : Computer Architecture Miscellaneous

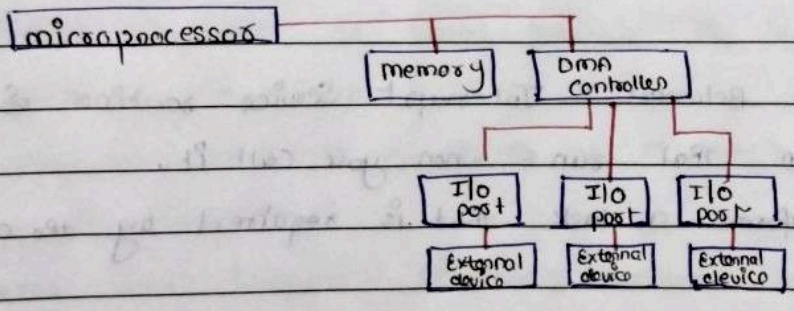
Sol^{n} (1) Explain DMA (Direct memory Access) with diagrams.

i.) In modern computers, DMA is used to transfer large Block of data of high speed between the I/O devices & main memory.

ii.) DMA increses the data transfer rate.



iii.) DMA allows I/O device to transfer data without CPU's intervention.



Sol^{n} (2) Explain operation of MESI Protocol (cache Coherency).

This indicates that the ~~Cache~~ most widely used cache coherence protocol.

- MESI protocol : Modified Exclusive Shared Invalid
- Dragon update - based protocol
- Firefly update - based protocol

	Invalidate	update
3 State	MSI	Firefly
4 State	MESI	Dragon

modified : it means that the Cache has the only copy which is correct in the whole system. The data which are in the main-memory are wrong.

Exlusive :- This cache is the only one that has the correct value of the Block.

Shared :- Another process can have the data in the cache memory & both copies are in their current version.

Invalid :- It is a non-valid state. The data you are looking for is not in the Cache & the local copy is not correct because another processor has updated the corresponding memory location.

Soln ③ what is meant by Synchronization in Computer architecture?

→ Synchronization in Computer architecture it refers to the coordination and control of parallel process, threads, or components to ensure the proper & orderly execution of a program

→ It is crucial to maintain data consistency & prevent issues such as data race, deadlocks etc.

→ Synchronization mechanisms are used to control access to shared resources and coordinate the execution of multiple tasks.

→ Example of Synchronization mechanism are semaphores, Condition Variable, Atomic operation, mutual exclusion etc.

Soln ④ what is paging? what is a page table?

→ paging is a memory management technique. It is a process for data transferring between main memory & secondary.

→ secondary and main memory are divided into equal size of fragments. Then data from Secondary memory to main memory is transferred & mapped.

Soln ⑤ find the total number of frames. if a System, size of the main memory is 230 bytes, The page size is 4KB & The size of each page table entry is 32-bit.

Total size of main memory is 230 bytes

The page size is 4KB

→ frame size = page size = 4KB = $(2^2 \times 2^{10})$

Number of frame = $\frac{230 \text{ bytes}}{2^{12}} = \frac{230}{4 \times 1024}$

≈ 1 frame

Soln ⑥ Consider a System with page table entries of 8 bytes each. If the size of the page table is 256 bytes, what is the number of entries in the page table?

table Entry in page = 8 bytes each

Size of page table = 256 B

Total page table = $\frac{256}{8}$ = $\frac{32}{\text{Size of page table}}$ / Table entry in page

total Page table = 32 entries

Soln ⑦ Consider a machine with 32-bit logical address, 4KB page size & page table entries of 4 Bytes each, find the size of the page table in Bytes, Assume the memory is Byte addressable.

Byte Addressable memory :- logical address = 2^{32} B

page size = 4KB $2^2 \times 2^{10} = 2^{12}$ B

Page table size = number of pages × Entry size

number of pages = $\frac{2^{32}}{2^{12}} \times 4 = 2^{20} \times 2^2$

$= 2^{22} \Rightarrow$ 4MB