7.12.2024

Computer Organisation - study of diff. hardware & their interconnection to built a computer.

Computer Architecture :- study of the behavior of the computer with the user along with system softwares like Operating Systems, ISA (instruction set architecture), compilers & device driver software.

→ most popular microprocessor → 8085 µp (first gen) (8 bit)

8086 µp (2nd) (16 bit)

dual core → 2 layers → *2 ALU

* supports parallel processing

Register is the smallest unit of storage in a processor.

→ 1 Byte

Registers

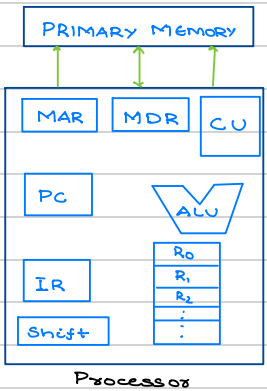
it can be 16 bits, 32 bits or 64 bits

* for specific storage.

* holds only intermediate data.

09.12.2024

imp * **Basic Operational Concept :-**

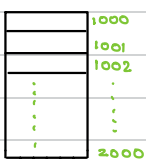


↕ → vice versa

	opcode	operands
1000.	ADD	R_0, R_1, R_2
1004.	SUB	A, B, C
1008.		
1012.		
1016.		

* registers are denoted by R. (GPR)

* A, B, C means data is stored in Memory.



→ C.U (control unit) → only generates clock pulse ; by sending these clock pulse, it controls all hardware.

→ A.L.U (arithmetic & logic unit) → it performs all the arithmetic & logic operations.
* only combinational ckts are present.

Registers – are smallest storage unit of a comp. & are present inside the processors

– 2 types :-

(i) General Purpose Registors (GPR)

(ii) Special Purpose Registors (SPR)

General Purpose Resistor → are used to hold data for temp. period.
→ it is a memory.

Special Purpose Resistor → also known as PC (Program Counter)
(Program Counter) PC → PC is a SPR which holds the address of the next instruction to be executed inside the CPU.

Instruction Register (IR) → IR holds the currently executing instruction.

Memory Address Register (MAR) → MAR is a SPR which holds the address of memory location from where we need information & bring it to CPU.

* Memory Data Register (MDR) → MDR is a SPR which holds the data to be sent from memory to processor or vice-versa. (processor to memory)

2000	ADD	R_0	R_1	R_2
2004	SUB	A	B	C
2008				
⋮				

memory access

1

1 + 1 + 1 + 1

read inst^r | A | B | write in C

*imp

Instruction Execution Steps :-

① Fetch

1st instruction's address is stored in PC.

add stored in PC ⟶ OS gives instruction to PC

sends add to MAR.

Any add. proc. wants to send to memory ; it will send first to MAR ; then MAR sends to Primary memory.

then 1° memory checks & Instruction (data) transferred to MDR then MDR to IR.

While reading the 1st instruction, PC has the add of 2nd inst.

PC → MAR → PM (Primary mem.) → MDR → IR

Searched inst. on that add

then PC increment to store add. of next instr.

② Decode

1st when CPU gets instruction, it will understand how to process it.

CU does the decoding of the work.

decode → ADD

it'll understand - Source R_0, R_1 (inside CPU)

" " - destination R_2 (inside CPU)

③ Execute

ALU gets the R_0, R_1 values & operations Add & performs the operation

④ Write the result

ALU writes the result in the destination R_2.

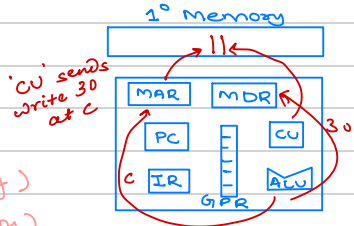
SUB A, B, C

① fetch

② Decode - Subtract

source - A, B (memory)

Destination - C (memory)



CU $\xrightarrow{A}$ MAR $\xrightarrow{A}$ 1° mem $\xrightarrow{\text{let A=50}}$ MDR $\xrightarrow{50}$ ALU

CU $\xrightarrow{B}$ MAR $\xrightarrow{B}$ 1° mem $\xrightarrow{20}$ MDR $\xrightarrow{20}$ ALU

ALU operates & A - B = 30 sends 30 to MDR & C to MAR

ALU $\xrightarrow{C}$ MAR $\xrightarrow{C}$ 1° mem $\longrightarrow$ C = 30

* CPU sends a control signal to memory to signify whether read or write operation.

Interrupt :- external signal which stops the currently executing process.

It uses a program called 'Interrupt Service Routine'

↓

it serves the interrupt

Whenever interrupt come in CPU

"it" stove all immediate information in a stack.

then allow the interrupt program to execute.

→ then after completion of the instruction, it loads the instruction intermediate information to the desired location & all the process to execute from where it stops.

18.12.24

series of wire are used to transfer data
16 bit = 16 wire
8 " = 8 "

BUS - is a series of wire used to establish the connection + communication betⁿ diff. hardwares present in the comp.

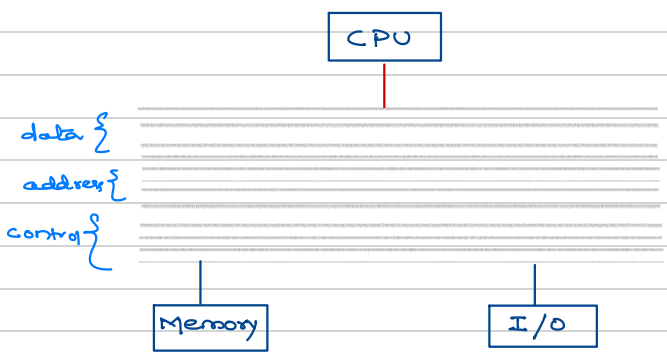
2 types :- (I) Internal BUS - internal connᶜ (inside RAM)
(II) External BUS - external

CLASSIFICATION :-

Data & Address are transferred through BUS.

So ; types of BUS are :-

(i) Data BUS - to transfer data
(ii) Address BUS - to transfer address
(iii) Control BUS - to transfer control signals.



→ single BUS organisational data

- to transfer data using same wires.

Disadvantages of single BUS :-

* only one way of communication
* it can't communicate with other devices.

Buffer

* very small memory

* is a small memory basically attached with low speed hardwares to facilitate the data transfer betⁿ two diff. hardwares of dissimilar speed.

14.12.2024 Basic Performance Equation :-

$$T = \frac{N \times S}{R}$$

T → Time taken to execute a task.
N → Number of instructions in a task.
S → Average Number of clock cycles per instruction.
R → Clock rate / frequency

(N x S → Total no of clock cycles for 'x' no of instruction)
(Clock rate means how many clock pulse generated in 1 sec)

$$\text{Clock Time } (P) = \frac{1}{R}$$

or $T = N \times S \times P$

R	P
1 KHz = 10^3 Hz	1 MS
1 MHz = 10^6 Hz	1 μs
1 GHz = 10^9 Hz	1 ns

Q → If R is 2.4 GHz ; then calc P.

$$2.4 \text{ GHz} = 2.4 \times 10^9 \text{ Hz}$$

$$P = \frac{1}{R} = \frac{1}{2.4} \times 10^{-9} = 0.41 \text{ ns}$$

Q - A task contains 200 instructions & each instruction will take on an average 8 clk cycles to execute. If the system operate with a clock rate of 400 MHz. Calc the time req. to execute a task.

$$T = \frac{N \times S}{R}$$

$N = 200$

$S = 8$

$R = 400$ MHz

$$T = \frac{200 \times 8}{400}$$

$$T = 4 \text{ μs}$$

Q- Let a processor operates by a freq. of 10 MHz & it executes a program having 90 instructions, out of which 50% of register referan. inst. & 30% are memory trans inst. & 20% are branch inst. Register ref. inst, Memory & Branch takes 4, 8 & 6 clk cycles respectively. Find out the total time taken by the processor to execute the program.

Ans → $R \rightarrow 10$ MHz

$N = 90$

register. ve ins = 50% of 90 = 45 = N_1

memory " " = 30% of 90 = 27 = N_2

branch " " = 20% of 90 = 18 = N_3

$$S = \frac{45(4) + 27(8) + 18(6)}{90}$$

$$= 5.6$$

$$T = \frac{N \times S}{R} = \frac{5.6 \times 90}{10} = 50.4 \text{ μs}$$

Or

$R = 10\ MHz$

$P = 0.1\ \mu s$

$T_1 = \{45 \times 4 \times 0.1\}\ \mu s$ $\qquad T = N \times S \times P$

$= 18$

$T_2 = \{27 \times 8 \times 0.1\}\ \mu s$

$= 21.6$

$T_3 = \{18 \times 6 \times 0.1\}\ \mu s$

$= 10.8$

$T = T_1 + T_2 + T_3$

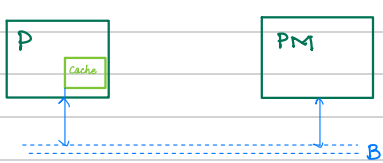
$= 50.4$

$$\downarrow T = \downarrow \frac{N \times S \downarrow}{\uparrow R}$$

16.12.2024 **Criteria for Performance Enhance :-**

Perf. can be optimised by doing some hardware & software modifications :-

1) Inclusion of cache memory in the processor.



P → processor

PM - Primary Memory

speed of Processor is much faster than PM ; so execution takes time .

Cache is a very small memory present inside the processor to reduce the memory excess time

(if PM is 8 Gb ; then cache is almost 4 mb)

There are 2 reasons for reduction in fetch time.

(i) Cache memory is very fast.

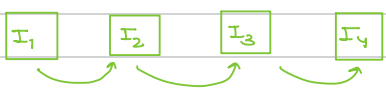
(ii) Cache memory is present inside the processor.

Locality of Instruction (Properly)

90% of instruction of the program will execute for the 10% of time & 10% of instruction of program will execute for 90% of time.

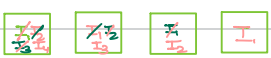
2) Pipeline of Processor :-

* overlapping is done



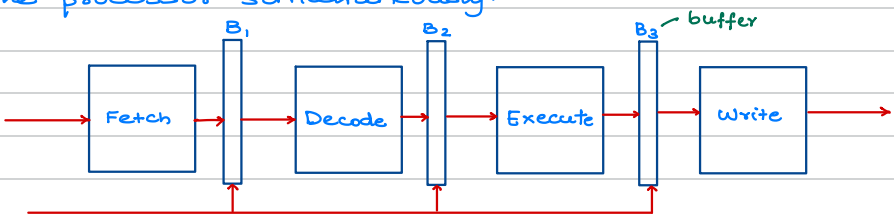
⇓ efficient way

18.12.24

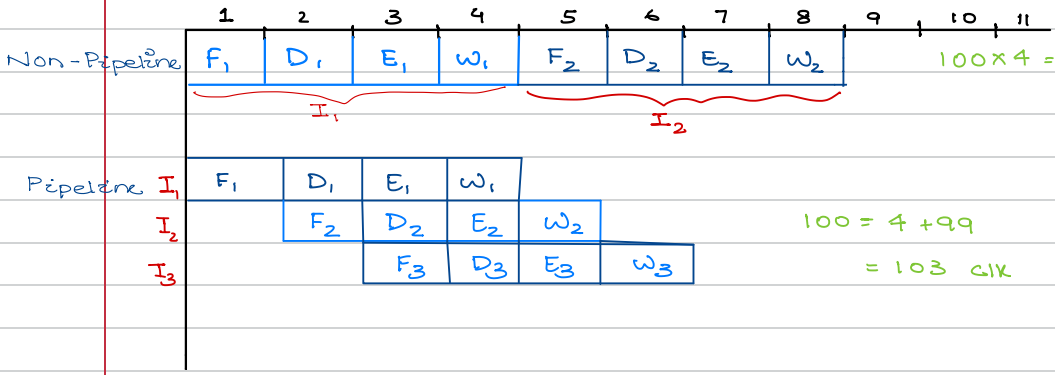


Pipeline is a hardware enhancement to achieve parallel processing in a single processor by logically dividing the processor into diff modules & executing multiple instruct-ions simultaneously in a overlapped fashion.

So any particular time, multiple instructions are executed in the processor simultaneously.



Synchronous pipeline diag



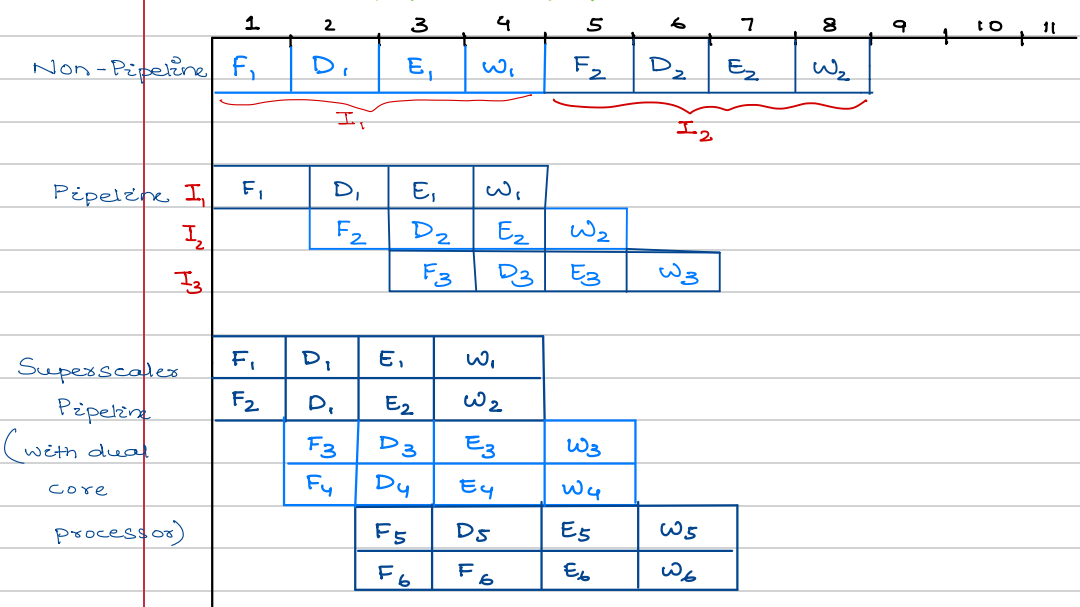
$$\text{Speedup} = \frac{400}{103} \simeq 4$$

$$\text{Speed up} = \frac{n-K}{K+(n-1)}$$

K = no of stages in the pipeline
processes

Super Scales pipeline :-

(pipeline of pipeline)



Performance Eqn for Pipeline :-

$$T = \frac{N \times S\downarrow}{R}$$

$S \rightarrow 1$ (1 clk pulse)

So eqn becomes

$$T = \frac{N}{R}$$

③ Clock rate

* hardware enhancement
* enhancement in 'T'. (time taken)

④ Compiler

* Software enhancement.
* 'N' will be reduced.
  ↳ by using advanced & efficient compiler; the no of instructions per program (N) is reduced.

19-12-24

Types of Instruction Set Architecture / Assembly Language Instruction :-

Every system has 2 different types of ISA :-

	RISC	CISC
	* Reduced Instruction Set Computer	* Complex Instruction Set Computer
C = A + B	* LOAD A, R1 LOAD B, R2 ADD R1, R2, R3 STORE R3, C	* ADD A, B, C } only one instruction
	* also known Load - Store Interaction bcoz Memory Interaction is only done by Load & store	* no such name
	* supports maximum one memory interaction.	* supports any no of memory interaction.
	* Size of instruction is less. (it deals with resisters)	* Size of instruction is more. (it deals with address)
	* supports pipeline	* doesn't supports pipeline

H-W 10 diff. bet^n RISC & CISC *imp

	Feature	RISC	CISC
1.	Instruction Format	fixed instruction size	variable instruction set
2.	Pipelining	Easier to implement	Difficult to implement
3.	Registers	more registers	fewer registers
4.	Memory Usage	needs more RAM	less RAM
5.	Execution Time	one instruction per clock cycle	multiple cycles per instruction
6.	Power efficiency	consumes less power	consumes more power
7.	Cost	cheaper to design	expensive
8.	Compiler usage	relies heavily on compiler organization	less dependent on compiler organization
9.	Addressing Mode	fewer addressing mode	many addressing mode.
10.	Example	ARM, MIPS	x86, Intel 8086

ARM → Advanced RISC Machine

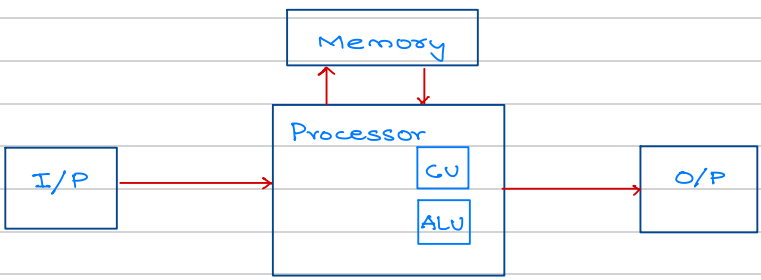
MIPS → Microprocessor without interlocked pipeline stages

Classification of Computer :-

1) Von-Neumon Architecture
2) Havard's Architecture

Von-Neumon is the 1st scientist who proposed memory in a computer.

Von-Neumon Architecture



But Havard divided the memory into 2 parts -

(i) Instruction Memory

(ii) Data Instruction