

## \* Computer Architecture $\Rightarrow$

→ Computer को design कैसे किया गया है।

→ It includes  $\Rightarrow$

A. Instruction Set Architecture (ISA) = Instructions का set जो CPU समझ सकता है।

B. Data Types = int, float, double

C. Registers का design = कितने Registers हैं & उनका size क्या है।

D. Addressing Modes.

$$\text{कल} = \frac{\text{पर्फ}}{\text{प्रप}} = \frac{\text{पर्फ}}{\cancel{\text{प्रप}}} \checkmark$$

## "Computer Organization & Architecture"

EMRS  
↓  
Live

### \* Computer Organization →

- Computer की Internal working की व्याख्या |
- प्रयोग कौन सा hardware किस काम की मार्फत करेगा |
- It includes =
  - A. Data का flow कैसे होता है | (Between CPU, Memory & Input/Output Devices)
  - B. Control Signals कैसे generate होते हैं, जो Computer के उभी parts को संचालित करते हैं ताकि लिए बोलते हैं।
  - C. Memory का structure कैसा होता है | (RAM, ROM, Cache etc.)
  - D. Processor की working ⇒ Instructions कैसे execute होते हैं।

## \* Organization v/s Architecture

| Difference                | Organization                                  | Architecture                                    |
|---------------------------|---|---|
| 1. Focus                  | Implementation (काम करके करेगा)               | Design (काम होना चाहिए)                         |
| 2. User से Related ?      | नहीं (internal hardware level पर काम करता है) | हाँ (User या programmer के लिए Visible होता है) |
| 3. Example                | Control signals, Memory layout, Data path.    | Instruction set, Registers                      |
| 4. Features Supported for | Hardware engineers, System designers.         | Software developers, Computer designers.        |

## ★ Types of Computer Architecture ⇒

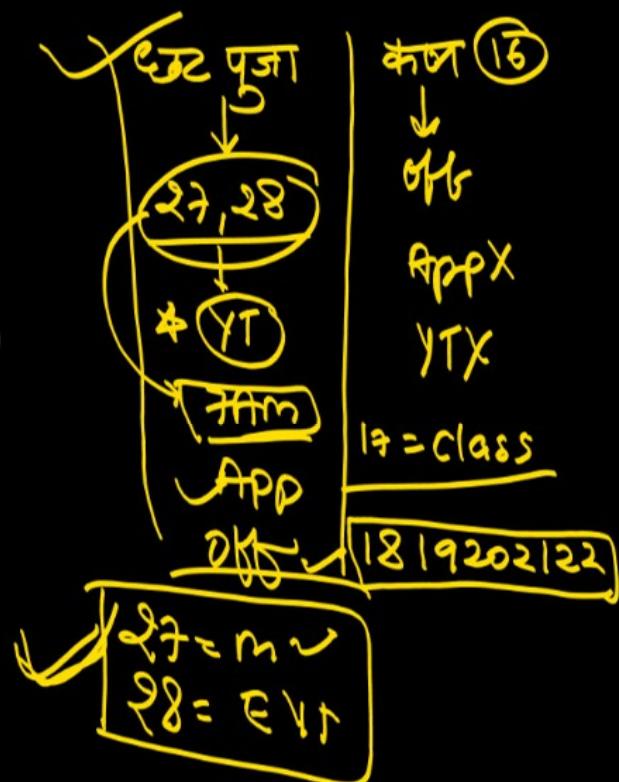
1. Von Neumann Architecture
2. Harvard Architecture
3. Instructions Set Architecture (ISA)
4. Micro architecture
5. System Design Architecture

## I. Von Neumann Architecture

- लब्धे पुराना & widely used architecture जिसमें Data & Instructions को एक ही Memory से Access किया जाता है।
- 1945 में John Von Neumann ने propose किया।

### \* features

- Single Memory for program & Data.
- Instructions sequentially execute होते हैं। (एक के बाद दूसरा)
- Simple design
- Program Counter Instructions का address बनाता है।



#### \* Working $\Rightarrow$

- CPU, Memory से Instructions की Fetch करता है।
- Instructions को Control unit द्वारा decode किया जाता है।
- Instructions को ALU द्वारा execute किया जाता है।
- Results, Memory में store होते हैं।

#### \* Problem $\Rightarrow$

- इसमें Data & Instructions एक ही Bus share करते हैं, इसलिए CPU को हर बार wait करना पड़ता है जब तक Data & Instructions Memory से नहीं मिल जाते।
- इससे performance slow हो जाती है।

#### \* Ex $\Rightarrow$ Intel 8085 & 8086.

## 2. Harvard Architecture ⇒

- invented at Harvard University
- इस Architecture में Data & Instructions के लिए मलग-2 Bus & Memory होती हैं

### \* Working ⇒

- CPU एक ही समय में Memory से Data read कर सकता है और Memory से Instructions को fetch कर सकता है।
- दोनों tasks parallel होने से performance fast होती है।

|                   |    |               |
|-------------------|----|---------------|
|                   |    | 3 NOV         |
|                   |    | ↓             |
| Thulank<br>Battle | 24 | Sunday        |
|                   | 25 |               |
| (26)              |    |               |
| 27 -              |    |               |
| 28 -              |    | 7 Manu<br>Kum |
| 29 -              |    |               |
| 30 -              |    |               |
| 31 -              |    |               |
| 1 -               |    |               |
| 2 -               |    |               |

### \* Features ⇒

- Separate Memory & Buses.
- Parallel Processing possible.

### \* Drawback ⇒

- Design complex & costly.
- Data & Instructions की Memory space  $\overline{\text{अलग}}^2$  होने से flexibility कर देती है।

### \* EX ⇒ ARM Cortex-M Series, PIC Microcontroller, DSP(Digital Signal Processors)

↓  
Advanced RISC Machines

## \* Modified Harvard Architecture =>

- Modern CPUs में Von Neumann & Harvard Architecture के features combine किए जाते हैं।
- Memory common होती है। (Like Von Neumann)
- लैंडिङ Data & Instructions के लिए मलग-<sup>2</sup> Cache होती है। (Like Harvard)
- इससे Speed भी गिरती है & Design भी efficient होती है।
- Ex- Intel i7, AMD Ryzen, ARM Cortex-A Series.

watch  
✓ कनेक्ट > 07 AM

### 3. ISA =

- Instruction Set Architecture
- Hardware (CPU) & Software (Program) के बीच का Interface.
- ISA कहाता है कि processor कोनसे instructions, understand & execute कर सकता है।

## \* Types of ISA =>

A. CISC

B. RISC

### A. CISC =>

- Complex Instructions Set Computing
- इसमें बहुत सारी Complex instructions होती हैं।
- एक instruction multiple low level operations के कर सकती है।
- जैसे - Intel X86.
- इसकी Code size small होती है, क्योंकि कम instructions लिखने पड़ते हैं।
- Main disadvantage यह है कि इसकी execution speed complex decoding के कारण slow होती है।

### B. RISC ⇒

- Reduced Instructions Set Computing.
- Simple & fixed length instructions.
- प्रत्येक instruction एक clock cycle में execute होता है।
- Ex = ARM.
- faster execution.
- Easy to design pipeline.
- Main disadvantage = code size बहुत बड़ा हो जाता है।

## \* Type of instructions in ISA ⇒

1. Data Transfer ⇒ MOV, LOAD, STORE (Data को एक जगह से दूसरी जगह Move करता)
2. Arithmetic ⇒ ADD, SUB, MUL (Maths Operations perform करता)
3. Logical ⇒ AND, OR, XOR (logical operations)  
✓ Return
4. Control flow ⇒ JMP, CALL, RET (Program के flow को change करता)
5. Input/Output ⇒ IN, OUT (Input & Output devices के साथ data exchange करता)
6. Comparison ⇒ CMP (2 operands के बीच compare करता)

## \* Examples of Instructions $\Rightarrow$

1. MOV R1, R2  $\Rightarrow$  Move data from R2  $\rightarrow$  R1
2. ADD R1, R2  $\Rightarrow$  Add R1 + R2 & Move to R1
3. AND R1, R2  $\Rightarrow$  R1 AND R2
4. JMP 420H  $\Rightarrow$  Jump to address 420H
5. CMP R1, R2  $\Rightarrow$  Compare R1 & R2
6. PUSH R1, POP R2  $\Rightarrow$

$$R1 = 5, R2 = 7$$

AND R1, R2

$$\text{ADD } \overbrace{R1, R2, R3}^{\substack{R1 = \\ R2 + R3}}$$

## \* Pipelining ⇒

- यह एक Technique है, जो CPU की performance को improve करती है।
- इसमें instructions को step by step process किया जाता है।
- Normal CPUs (without pipeline) किसी एक instruction को complete करके next को start करते हैं।
- But pipelining में जब एक instruction decode हो रहा होता है, हो next instruction को fetch किया जा सकता है।
- Parallel execution allowed.
- pipelining से CPU की speed increase हो जाती है & throughput increase हो जाता है।

#### 4. Micro Architecture →

- मास्टर अर्किटेक्चर CPU का internal structure के से बना है और इसके components internally कैसे काम करते हैं, मास्टर बताता है।
- यहाँ system architecture का Map है, तो Micro Architecture उस Map का detailed implementation है।

#### \* Main Components of Micro Architecture →

##### A. Registers →

- Fastest, Smallest & Costliest memory of Computer.
- Ex = Program Counter, Accumulator, Instruction Register.
- Inbuild memory of CPU.

### B. ALU =>

- Arithmetic & Logic Unit
- performs arithmetic & logical operations like AND, OR, NOT, Add, Sub etc.
- मध्य Calculations करते time Rough work के लिए Register का use करती है।

### C. Control Unit =>

- Controls & manages all the devices & resources that's why it is known as central Nervous System.
- Instructions को decode कर Time & control signals generate करता है।

### D. Cache =>

- Buffer memory between CPU & RAM.
- मध्य Memory CPU & RAM के बीच Speed management करती है।
- 4 Types = L1, L2, L3, L4 (L1 = Fastest, smallest & nearest to CPU)

### E. Pipeline ⇒

- Multiple instructions को सुक्र साथ process करने की तकनीक |
- 4 Steps = fetch → Decode → execute → writeBack

### 5. System Design Architecture ⇒

- It architecture system के major components के arrangement & उनके बीच Data flow को बताता है।
- High level design जो बताता है कि System कौनसे काम करेगा & उसके major units (CPU, Memory, I/O) केसे Connected हैं।

•  X

## \* Error Detection & Correction Codes ⇒

→ एक Computer से दूसरे Computer तक Data Transmit करते समय Noise, Signal loss या Interference की वजह से Data Corrupt हो जाता है, जिसे Handle करने के लिए इसका Use होता है।

## \* Types of Errors ⇒

### 1. Single Bit Error ⇒

→ एक Bit का गलत हो जाना।

→ Ex ⇒  $1010101 \rightarrow 1010001$

↓  
1 Bit change

### 2. Burst Error ⇒

→ एक के बाद एक Multiple Bits का गलत हो जाना।

→ Ex ⇒  $11001100 \rightarrow 11111100$

↓  
4 Bits change

## \* Error Detection Codes ⇒

### 1. Parity Bit Method ⇒

- सबसे Simple & Basic Method.
- इसमें एक extra bit (Parity Bit) Data के साथ Transmit की जाती है।
- यह ensure करता है कि Data में Total 1's को Counting either even or odd होनी चाहिए।
- Types = A. Even Parity  
B. Odd Parity

#### A. Even Parity ⇒

- Total number of 1's must be even (including parity bit).
- Ex ⇒ 1010001 (Total 3 1's)  
$$\begin{array}{r} 1010001 \\ \downarrow \\ 10100011 \\ \swarrow \end{array}$$

### B. Odd Parity $\Rightarrow$

→ Number of 1's must be odd (including parity bit)

→ Data Bits = 1010001



10100010 (1's are already odd)

\* Parity bit Method का Use केवल single bit error detect करने में किया जाता है।

## 2. Checksum Method

- Network Data Transmission में Mostly used.
- Data को equal size के blocks/segments में divide किया जाता है।
- सभी blocks का sum generate करते के लिए checksum का use किया जाता है।
- ये checksum receiver end पर भी check किया जाता है।

→ Ex :-

Data Blocks:

|          |      |
|----------|------|
| 1 = 1001 |      |
| 2 = 1100 |      |
| 3 = 0101 | Even |

Sum = 1001 + 1100 + 0101 = 11010  
                                    ↓  
                                    Carry Ignore

= 1010  
Sender

|       |
|-------|
| 1001  |
| 1100  |
| <hr/> |
| 10101 |
| 0101  |
| <hr/> |
| 11010 |

## \* Error Correction Code $\Rightarrow$

- इन codes का objective error को detect करके automatically correct करना होता है, पिछा data retransmission के।
- इसका use more reliable systems like memory, satellite communication में किया जाता है।

### A. Hamming Code $\Rightarrow$

- Richard Hamming ने दिया था।
- यह Single Bit Error Detection & Correction करता है।
- इस code में some redundant bits ( $r$ ) और Data bits ( $d$ ) के बीच add कर दिया जाता है, ताकि दर bit position एक unique binary number को represent करे।
- Find no. of redundant bits =  $2^r \geq (r+d+1)$  | 4 Data bits =  $2^4 \geq \frac{(r+d+1)}{5} = \boxed{r=3}$

## \* BUS \*

→ एक communication path / Set of wires / Set of lines जिसके through Computer Systems के Multiple Parts के बीच Data, Address & control signals का exchange होता जाता है।

→ Types = 3  
↓

A. Address Bus

B. Data Bus

C. Control Bus



### A. Address Bus →

- Unidirectional
- यह Bus उस Memory locations का Address Carry करती है, जहाँ से Data को Read/write करना होता है।
- जब CPU को Memory से Data को Read करता है, तो यह उस Memory locations का address, Address Bus के Through मेंजाता है।
- Ex = यदि CPU 16-Bit Address Bus का use करता है, तो वह  $2^{16} = 65536$  (64KB) Memory locations को Access कर सकता है।

### B. Data Bus $\Rightarrow$

$\rightarrow$  BiDirectional

$\rightarrow$  मद Bus CPU, Memory & I/O के बीच Transfer हो रहे Actual Data को Carry करती है

$\rightarrow$  CPU  $\xrightarrow{\text{Data Write}}$  Memory

$\rightarrow$  CPU  $\xleftarrow{\text{Data Read}}$  Memory

$\rightarrow$  Ex-> मडि CPU ने Data Bus की width 16 Bit है, तो मडि 2 Byte Data Per Cycle Carry कर सकती है



### C. Control Bus →

- Carries control & Time signals
- मध्य बस जाता है कि किस Type का operation perform करता है। (Read | write | interrupt)
- Direction = Both
  - ↓
  - (Based on control signal)

## ★ Microprocessor & Microcontroller ⇒

### A. Microprocessor ⇒

→ Invention = Year 1971



Intel 4004 by Intel Incorporation



1<sup>st</sup> Smallest & Complete CPU (4 Bits)

→ 1971 के बाद Intel 8008, Intel 80386 etc. processors में से & Currently Intel i Series

(Itanium) & AMD Ryzen Series के Complex & MultiTasking CPUs Available हैं।

→ Use = Personal Computers, laptops, Servers, Gaming Consoles (Playstations & Xbox)

→ Intel X86 1978 में आया था।

↓

Intel 8086 Architecture पर based.

AMD



Advanced Micro  
Devices

→ Microprocessor  $\xrightarrow{?}$  Types ⇒

- a. RISC
- b. CISC
- c. VLIW (Very Long Instructions Word) - Used for parallel processing
- d. EPIC (Explicitly Parallel Instructions Computing) - Advanced form of VLIW

## B. Microcontroller $\Rightarrow$

- इसमें Microprocessor के साथ-<sup>2</sup> Memory (RAM & ROM) & Peripherals (Input-Output & Timers) का भी Use किया जाता है, इसलिए इसका Use Embedded Systems में किया जाता है।
- First Microcontroller = Intel 8031  
1980s
- इसके बाद PIC, ARM, AVR popular Microcontrollers आये।
- Use = Embedded Systems (Microwave oven, Washing Machine), Robotics, Automobiles, Medical devices (Pacemakers, Insulin pumps), Consumer Electronics (TV, AC, Mobile)

## \* Types of Microcontroller ⇒

### a. PIC ⇒

- Peripheral Interface Controller
- Developed by = Microchip Technology
- Size = 8, 16, 32 Bit variants
- Use = Consumer Electronics, Embedded Systems & Automotives.

### b. AVR ⇒

- Advanced Virtual RISC
- Developed by = Atmel (Now Microchip)
- Based on RISC
- Used in Small Embedded devices

### C. ARM $\Rightarrow$

- Advanced RISC Machine
- RISC based powerful microcontroller Family.
- Low power consumption yet high performance के लिए well known.
- Smartphones & Embedded systems में widely used.

कठोर  
CoA  
Finish

Monday



## \* Parity Bit Method v/s Checksum Method v/s Hamming Code =>

| Comparison                 | Parity Bit Method  | Checksum Method  | Hamming Code   |
|----------------------------|--|--|--|
| 1. Definition              | Simple Error Detection Technique जिसमें Single Bit Parity को Add किया जाता है। | Data Blocks को Add कर 1's complement से Checksum generate होता है। | Data + Redundant Parity Bits का use कर Single Bit error correct करता है। |
| 2. Main Goal               | Error Detection  | Error Detection  | Error Detection + Correction   |
| 3. Category                | Error Detection Code   | Error Detection Code   | Error Correction Code  |
| 4. Added Redundant Bits    | Only 1 Bit   | 16 Bit / 32 Bit checksum   | Multiple Parity Bits (3,4,5 Depend on Data size)                         |
| 5. Placement of Extra Bits | End of Data frame  | Data के बाते checksum Block  | Parity Bits at positions 1, 2, 4, 8                                      |

|                              |                         |                                     |  |
|------------------------------|-------------------------|-------------------------------------|--|
| 6. Technique used            | Count 1's = Even or Odd | Binary addition + wrap around carry | Parity bits calculate using multiple overlapped bits group.            |
| 7. Detects Single Bit Errors | YES                     | YES                                 | YES  |
| 8. Corrects Single Bit Error | NO                      | NO                                  | YES  |
| 9. Detects Double Bit Error  | NO                      | NO<br>(Sometimes)                   | YES (Detect only, not correct)<br>→ Correct by Reed Solomon & BCH code |
| 10. Reliability              | Low                     | Medium                              | Very High  |

|              |                             |                        |  |
|--------------|-----------------------------|------------------------|--|
| 11. Use      | Simple Serial Communication | TCP, UDP, IP, Ethernet | ECC RAM, Satellite, Mission critical Systems |
| 12. Pros     | Simple, Cheap, Fast         | Large Error Detection  | Detect + Correct = Very accurate             |
| 13. Cons     | Only Detection, Weak        | No Correction          | Complex & More Overhead                      |
| 14. Use case | Low Risk Communication      | Networking Protocols   | High Reliability Systems                     |