

Introduction to Computer Architecture

■ Evolution of Computer Architecture

- first Generation: Vacuum tubes, slow and large

- Second Generation: Transistors, faster and smaller.

- Third Generation: Integrated circuit, more compact.

- fourth Generation: Microprocessors, personal computers.

- fifth Generation: AI, Parallel Processing

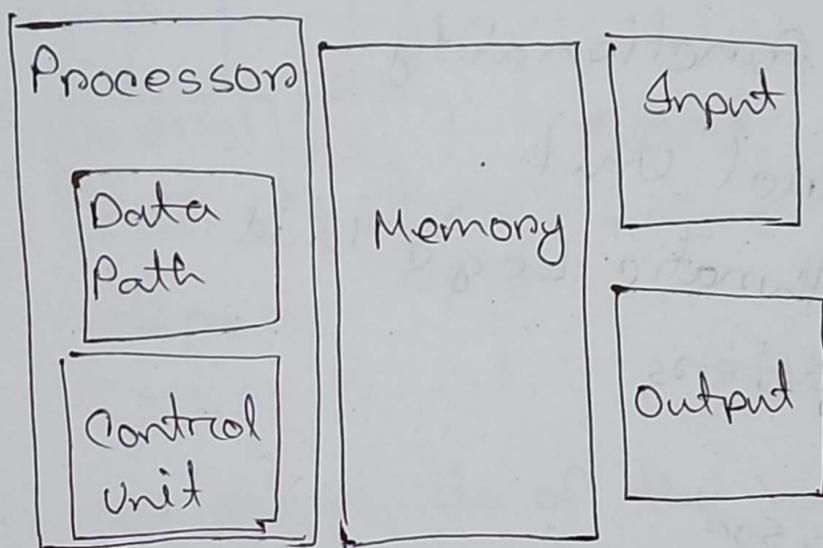
■ Computer Architecture refers to the design and organization of a computer's components. five classic components are -

1. Input 2. Output 3. Memory 4. Datapath

5. Control Unit

Simple architecture of a computer hardware -

Control unit + Data path = PROCESSOR



Buses: The pathways that carries signals.

- Data Bus : transfers data between components
- Address Bus : carries memory add for data retrieval
- Control Bus : to manage flow of data and instruction

- Datapath e.g. ALU, Registers, Bus
- control commands data path, memory and I/O devices

⊕ CPU functionality

- Control Unit
- Arithmetic Logic Unit
- Registers

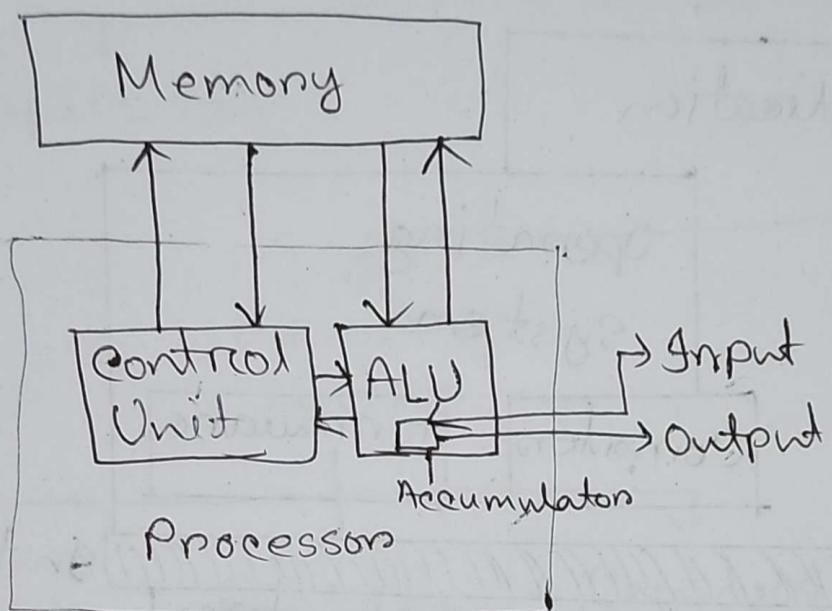
⊕ Processor

- CISC (Complex Instruction set computers)
- RISC (Reduced Instruction set Computer)
- DSP (Digital Signal Processor)

⊕ Memory Hierarchy

- Registers
- Cache
- Main memory (RAM)
- Secondary storage (Hard Disk, SSD)

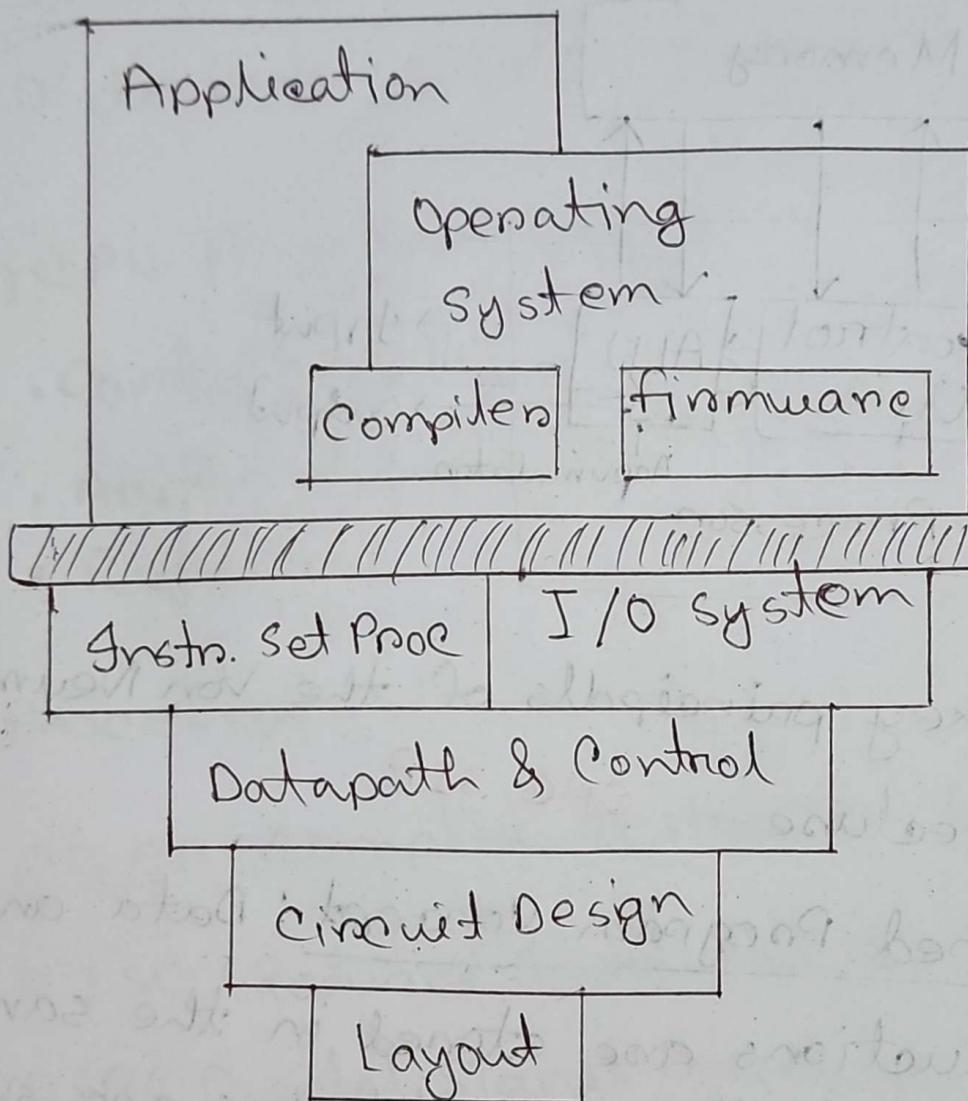
Von Neumann Architecture (stored Memory Program)



Two key principles of the Von Neumann Architecture - SPC, SIE

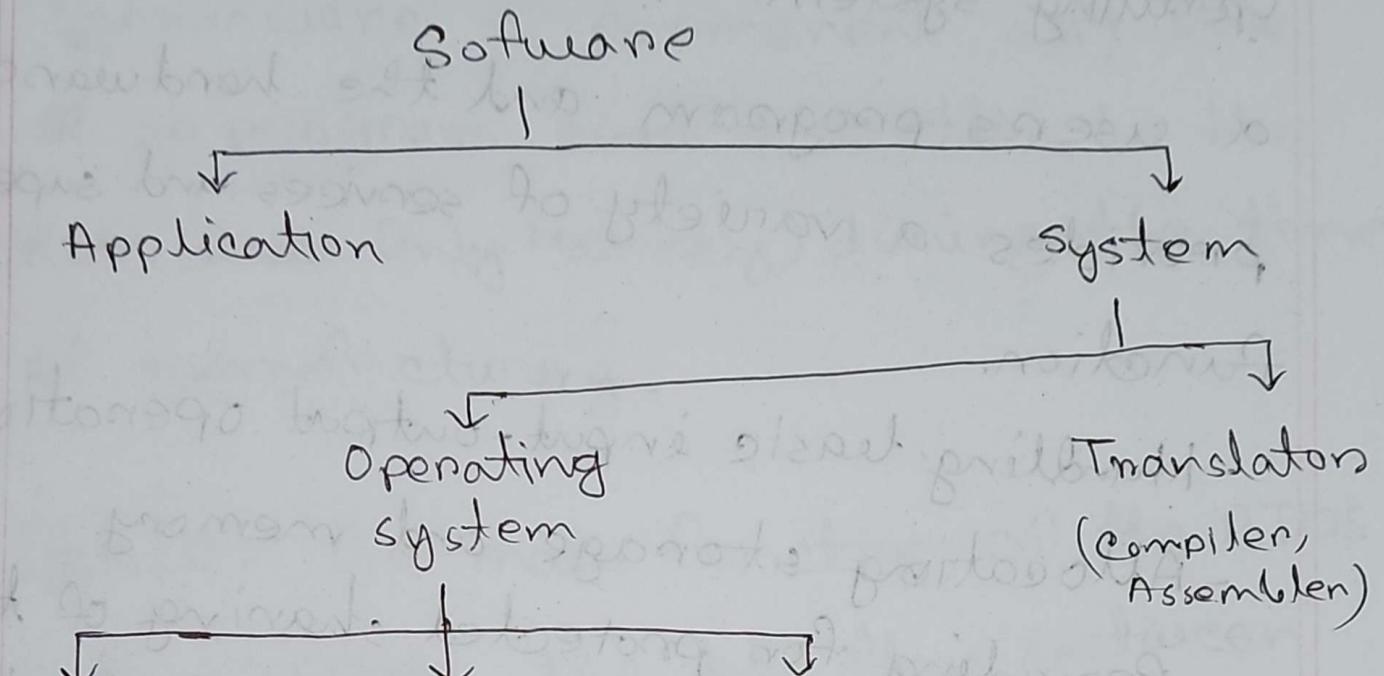
- 1) Stored Program Concept: Data and Instructions are stored in the same memory. Address and data bus are shared.
- 2) Sequential Instruction Execution: Ins. s are fetched and executed one at a time in a specific order.

Hardware and Software



Layers of Software





☞ OS Kernel: The kernel is the first program that is loaded after the bootloader because the kernel has ~~more~~ to handle all the earnest of the things

Computer System

Application \leftrightarrow OS Kernel \leftrightarrow Hardware

Operating system: An OS interfaces between a user's program and the hardware, provides a variety of services and supervisory function.

- Handling basic input-output operation
- Allocating storage and memory
- Providing for protected sharing of the computer and among multiple applications

Eg - Linus, iOS, windows

Instruction cycle (CPU)

- Fetch

- Decode

- Execute

Firmware is permanent software or program that is written in the ROM (Read Only Memory) during the time of manufacturing.

BIOS: The main function of the BIOS is to manage the data flow between computer's OS and attached devices like mouse, keyboard etc.

Bios check if everything is working (Mouse, Keyboard) before starting the OS

Instruction Set Architecture

Instruction set

Hardware

Well known ISA's

- X86 (64 bits)

- ARM (32 & 64 bits)

- MIPS (32 & 64 bits)

MIPS stands for Microprocessor without
Interlock Pipeline Stages. is a RISC
architecture

Registers and their alternative names

Register Numbers	Alternative Numbers	Description
0	zero	the value 0
\$	\$ at	assembler temporary
2-3	\$v0-\$v1	values
4-7	\$a0-\$a3	arguments
8-15	\$t0-\$t7	temporaries
16-23	\$s0-\$s7	saved values
24-25	\$t8-\$t9	temporaries
26-27	\$k0-\$k1	interrupt/trap handlers
28	\$gp	global pointer
29	\$sp	stack pointer
30	\$s8/\$fp	saved value/frame pointers

var1: .word 3

array1: .byte 'a', 'b'

array2: .space 40

Lecture 3

Types of Instruction Format, Instruction Format - MIPS

Different Types of Instruction

- 1) Data transfers - used to move data between register & memory
- 2) Arithmetic
- 3) Logic
- 4) Shift
- 5) Conditional Branch
- 6) Unconditional Branch

\$\$\$\$\$

~~lw~~ \$t0, 8(\$s0) } data transfer
sw \$t0, 12(\$s0)

add \$t0, \$t1, \$t2 \$t0 = \$t1 + \$t2

addi \$t0, \$t1, 5 \$t0 = \$t1 + 5

sub \$t0, \$t1, \$t2 \$t0 = \$t1 - \$t2

mult \$t0, \$t1 hi, lo = \$t0 * \$t1

div \$t0, \$t1 lo = quotient, hi = remainder

add

Let, $t_1 = g, t_2 = h, t_4 = i, t_5 = f$

{ add \$t0, \$t1, \$t2

add \$t3, \$t4, \$t5

sub \$t6, \$t0, \$t3

and \$t6, \$t1, \$t2

ft \$t11 \$t0, \$t1, \$t0

{ snt \$t0, \$t1, 5

arithmetic

conditional Branch

- beq, bne, bgtz, bltz, bgez, blez

beq \$t0,\$t1, target

All instructions are 32 bits length. Three formats -

① R-format

OP rd, rs, rt → add \$t0, \$t1, \$t2

Opcode	rs	rt	rd	shift	funct
6 bits	5	5	5	5	6 bits

② I-Instruction

OP rt, rs, IMM → addi \$t0, \$t1, \$t2

OP rt, IMM(rs) → sw \$t0, 125(\$t1)

OP code	rt	rs	IMM
6 bits	5	5	16

③ J-Instruction

OP label → branch 25

OP code	Pseudo Address
6 bits	20

Some common MIPS arithmetic instructions:

add, addi, addu, sub, subi, subu

mult, div

Logical:

and	or	nor	not
-----	----	-----	-----

Shift:

sll, srl, sra

Memory access:

lw, sw, lh, sh

lw = memory to register

lh = register to memory

Control flow instruction

beq, bne, j, jal, jb

before proceeding

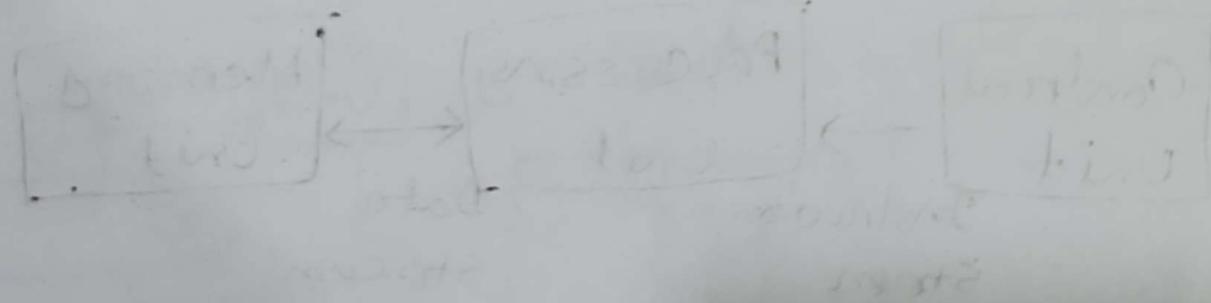
- most common for programs

OPI2

AM12

OC1M

AM1M



Flynn's Classification

It's a way of organizing multiple processor system

Categories of computer system-

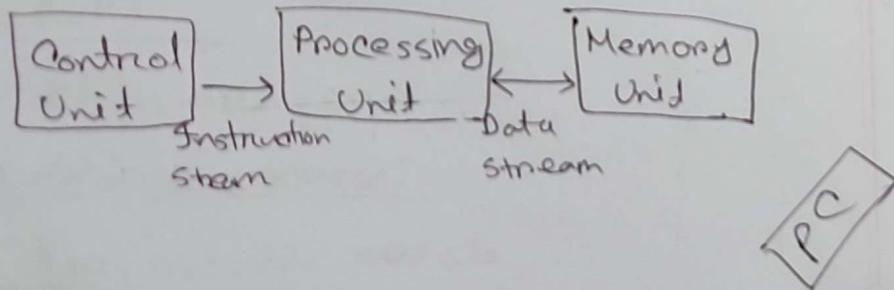
SISD

SIMD

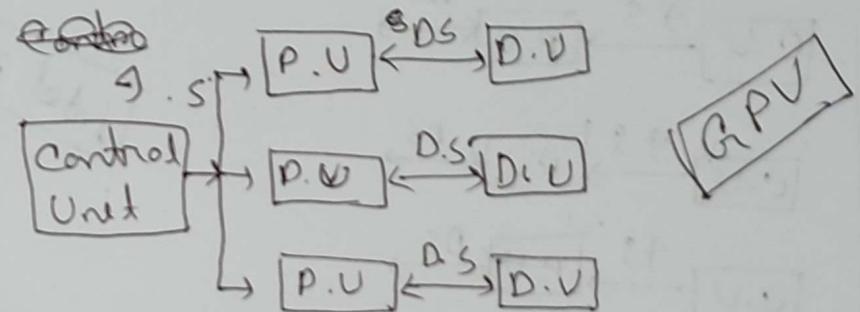
MISD

MIMD

SISD - single instruction single Data Stream



Single Instruction Multiple Data Stream (SIMD)



Eg - vectors, arrays

MISD-

