

## Computer language

- Instructing the computer to act as we want.
- Keywords are predefined words with predefined meanings in a comp. lang.
- Set of rules to communicate in computer language is called Syntax.

## Architecture

Computer architecture refers to the structure of the computer system which determines how its components interact with each other in order to execute a program.

- Silicon Carbide (SiC) & Gallium Nitride (GaN) are the modern choices for the purpose of semiconductor fabrication.
- MOSFETs are chosen over BJT in case of switching because, MOSFETs are voltage controlled devices while the BJT is current controlled device and it is easier to control voltage controlled than the current controlled.

## Analog v/s Digital Signals

### Analog

- ① the signal is continuous
- ② range of the signal is continuous
- ③ bandwidth is low
- ④ Analog hardware doesn't provide flexible implementation.
- ⑤ Suited more for audio & video transmission.

### types of languages

- ① High level
- ② Assembly level
- ③ Machine level

### Digital

- ① the signal is time separated.
- ② the signal only has two levels '0' & '1' to represent an information
- ③ bandwidth is higher.
- ④ digital hardware provides flexibility in implementation.
- ⑤ suited more for computing purposes.

### ① High level language

#### Advantages

- ① easy to write & debug
- ② easy to learn
- ③ one code can be run on various processors i.e., portability.

#### disadvantage

- ① memory inefficient
- ② slower to execute,
- ③ cannot communicate directly with hardware.

→ e.g.: Java, Python (due to AI/MU)

→ portability for high level language is achieved by a compiler or an interpreter.

→ a compiler compiles the whole code and then executes it while an interpreter reads the lines of the code one at a time.

## Computer language

- instructing the computer to act as we want.
- Keywords are predefined word with predefined meanings in a comp. lang.
- Set of rules to communicate in computer language is called Syntax.

## Architecture

Computer architecture refers to the structure of the computer system which determines how its components interact with each other in order to execute a program.

- Silicon Carbide (SiC) & Gallium Nitride (GaN) are the modern choices for the purpose of Semiconductor fabrication.
- MOSFET are chosen over BJT in case of switching devices, MOSFETs are voltage controlled devices while the BJT is current controlled device and it is easier to control voltage controlled than the current controlled.

## Analog vs Digital Signals

### Analog

- ① the signal is continuous
- ② range of the signal is continuous
- ③ bandwidth is low
- ④ Analog hardware doesn't provide flexible implementation.
- ⑤ Suited more for audio & video transmission.

### types of languages

- ① High level
- ② Assembly level
- ③ Machine level

### Digital

- ① the signal is time separated.
- ② the signal only has two levels '0' & '1' to represent an information
- ③ bandwidth is higher.
- ④ digital hardware provides flexibility in implementation.
- ⑤ Suited more for computing purposes.

### ① High level language

#### Advantages

- ① easy to write & debug
- ② easy to learn
- ③ one code can be run on various processors i.e., portability.

→ e.g.: Java, Python (due to AI/ML)

→ portability for high level language is achieved by a compiler or an interpreter.

→ a compiler compiles the whole code and then executes it while an interpreter reads one line of the code at a time.

#### disadvantage

- ① memory inefficient
- ② slower to execute,
- ③ cannot communicate directly with hardware.

## ② Assembly level language

### Advantages

- ① almost as faster as high level language.
- ② allows developer to have a greater control over the hardware and memory layout of their program.
- ③ programmes are smaller.

### Mnemonics

→ they are like keywords, where a particular name has a fixed functionality.  
e.g: ADD A,B,C  
MOV R1,R2

### Opcode

→ it is a single instruction that can be executed by the CPU.

→ in computer machine language it is binary or hexadecimal in value.

### Operand

→ it is the data or memory location used to execute the instruction.

## ③ Machine language

### Advantage

- ① no translation of codes is required. i.e., no compiler or assembler reqd.
- ② the CPU can directly read the set of instructions.
- ③ execution is very fast.

→ 4 bits are used so that Hexadecimal notations could be easily denoted, for the opcode.

9:	2	5	7	1	← LSB
	2	2	8	1	
	2	1	4	0	
	2	7	0		
	2	3		1	
	2	1		1	← MSB
	0			1	

④ ARM : Advanced RISC machine.

MIPS : million instruction per second.

### Disadvantages

- ① position of data is not specified at the code and the programmer must follow the data sheet.
- ② for programming, architecture has its mnemonic of every processor & data sheet of the processor is required, to be known.
- ③ each processor

### Assembler

→ an assembler is a program which converts codes written in assembly language to machine code (in '0' & '1')

### Binary

format

### disadvantages

- ① lacks portability
- ② hard to debug
- ③ hard to learn.

### Conversion

$$\begin{array}{r} 8 | 57 \\ 8 | 7 \quad . \\ 8 | 0 \quad 1 \\ \hline 0 \quad 11 \end{array} \quad (0.125)_{10} = (0.1)_2$$

$$\therefore (57)_{10} = (111001)_2 \quad \rightarrow 0.125 \times 2 = 0.250$$

$$0.250 \times 2 = 0.500$$

$$0.500 \times 2 = 1.000$$

$$\therefore (0.001)_2 = (0.125)_{10}$$

1001

1021

$$= 2^0 \times 1 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 \times 8^0 + 2 \times 8^1 + 0 \times 8^2 + 1 \times 8^3 = 8 + 16 + 0 + 8^3 = 536$$

= 9

Hexa decimal to binary

$$(A6)_{16} = (1010 0110)_2$$

 $\rightarrow 8$  convert876<sub>10</sub> to Binary, Hex, Oct.24A<sub>H</sub> to Bin, Dec, Oct.101101<sub>2</sub> to Dec, Hex, Oct

ASCII code?

Bin?

 $\rightarrow (876)_{10}$  to binary

2	1	876	0
2	438	0	
2	219	0	
2	109	+	
2	54		
2	27	0	
2	13	1	
2	6	1	
2	3	0	
2	1		
2	0	1	

$$\therefore (876)_{10} = (1101101100)_2$$

$$\text{or } (00111001100)_2 = (36C)_{16}$$

$$\text{or } (001101101100)_2 = (1554)_8$$

not enough $\rightarrow (24A)_{16}$ 

$$(24A)_{16} = (0010 0100 1010)_2$$

$$\therefore (001001001010)_2 = (1112)_8$$

$$\text{and } (1112)_8 = 8^3 \times 1 + 8^2 \times 1 + 8 \times 1 + 8^0 \times 4 = (586)_{10}$$

 $\rightarrow (101101)_2$ 

$$\text{or, } (001101)_2 = (2D)_{16}$$

$$\text{or, } ((01101)_2 = (55)_8)$$

$$\text{or, } (55)_8 = 8 \times 5 + 8^0 \times 5 = 45$$

## ASCII Codes

- it stands for American Standard Code for information interchange.
- it is an alphanumeric code to represent numbers, alphabets, special symbols.
- the ASCII codes are of 7-bits with MSB set to '0' but the extended set of ASCII codes contain MSB as '1' and 128 more characters are added to it.

## Advantages

- ① Takes less space
- ② Stores all alpha-numeric characters
- ③ ASCII code is standardize

## Application

- In every modern communication device.

## Binary Coded Decimal (BCD)

- It is a type of conversion of decimal number to binary system where each decimal number is represented by a 4-bit binary number.
- The 4-bit binary represents from '0' to '9'.
- '10' to '15' are not used in 4-bits in case of BCD.

Ex:  $(2)_{10} = (0010 \text{ 0001})_{BCD} = (0010 1001 0001 1000)_{10}$

## Advantages

- ① easy to represent decimal numbers.
- ② BCD is easily human readable

Ex: 7-segment display.

- 4 bits are called a nibble.
- 8 bits are called a byte.

∴ There are 2 nibbles in a byte.

- if two BCD codes are there in a byte, it is called a packed BCD.
- if only one BCD code is there in a byte, it is called an unpacked BCD.

∴ 

8	6
---	---

  
→ packed BCD  
→ both nibbles are paired

0	6
---	---

  
→ unpacked BCD  
→ only one nibble is packed  
→ unpacked BCD

→ Intel 4004 was the first processor (4-bit).

→ n-bit processor means, the processor processes n-bit of data at a time and it is called word size of the processor.

## Disadvantages

- ① Due to 8 bits, only a max<sup>n</sup> of 256 characters can be represented.

## Moores Law

- Every 2nd year the no of transistors on a micro-processor gets doubled.
- first computer was Analytical Engine by Charles Babbage made of gears.
- INTEL stands for Integrated Electronics.
- electrical equivalent of gears (variable speed) is a transformer with their turn ratio.
- electro-mechanical device: Relays.
- Colossus? → UK → Z3 is the first electromechanical general purpose programmable controlled computer made by Germans.
- Enigma? → Germany
- ENIAC? → USA → Electrical Numerical Integrator and Computer (ENIAC).
- In vacuum tubes the flow of free e<sup>-</sup> can be controlled by the heating of coil.
- An IC is an integration of many transistors.
- History of micro-processor?

→ Fairchild Semiconductors invented the first integrated circuit in 1959.

→ A micro-processor is a central processing unit on which a number of transistors are fabricated on a single chip.

→ the chip contains:

- ① an ALU
- ④ bus systems
- ② a control unit
- ⑤ clock
- ③ registers

→ first microprocessor was intel 4004 with 4 bits of data computing capability at a time.

## Generations of Microprocessors

### ① First Generation [1971 to 1972]

- from 1971 to 1972
- 4004 was invented in 1971
- 4004 had a clock freq of 740 KHz
- it has 4-bit word size
- could represent signed number b/w -8 to 7 only, hence only application was in controlling devices.

### ② Second Generation [1973 to 1978]

- creation of 8-bit microprocessors marked the starting of 2nd generation of microprocessors.
- Intel-8008 was first 8-bit microprocessor invented in 1972, with 500 K.
- Since, the bit size was only 8, it could represent -128 to +127 only, hence application was again in controlling devices.

→ Intel 8080 was first commercially popular 8-bit microprocessor.

→ 6800 from Motorola and 8080 from Zilog were other famous microprocessors but were costly due to NMOS technology.

### ③ Third Generation [1979 to 1980]

→ the third generation of microprocessors started with Intel releasing Intel 8086 which had 16-bit word size.

→ Hence -32768 to 32767 could be represented.

→ So basic arithmetic & controlling operations could be done.

→ Other than Intel 8086, Motorola 68000 and Zilog 8000 were other 16-bit microprocessors of 3rd gen.

#### ④ 4<sup>th</sup> Generation [1981-1995]

- the 4<sup>th</sup> generation of micro-processors started with Intel 80386, which had a word size of 32-bits, made using 1 μm fabrication.
- here the signed numbers range between  $\pm 2 \times 10^9$ .
- this microprocessor was suitable for arithmetic as well as controlling devices.
- the release of Intel 80386 popularly known as Pentium was the key stone in processors making fault executions.
- Pentium 4 was released in 2000 and had 42 million transistors with 1.5 GHz clock and 1500 MIPS capability.
- Motorola 68030 was another 32-bit type micro-processor.

MIPS = million Instructions per Second.

- Ted Hoff formulated the IC for the Japanese calculator. There he came with an IC with four parts.

RAM to keep data 4002	ROM to keep Instruction 4002
Executor circuits 4004	for I/O 4001

- ROM [read only memory] because → cannot write into it → why read only.
- it is called read only memory because, it permits the burning of instructions only once and then after only reading could be done on that memory.
- it is a kind of non-volatile memory.

#### PROM

- Programmable ROM
- instructions can be stored only once.
- error in instructions cannot be rectified.

#### EPROM

- Erasable programmable ROM
- Instructions can be stored multiple times.
- written instructions can be erased & change with the help of ultra violet ray.

#### ⑤ 5<sup>th</sup> Generation [1995 - onwards]

- the mark of 5<sup>th</sup> generation microprocessors started in 1995.
- these microprocessors have a word size of 64-64 bits.
- these microprocessors are faster, have more computational power and are cheaper than their previous Counter-parts.

#### → Generations of computer?

- IC packaging types
- Hierarchy of memory.

Why is it called ROM?

- 4004 had 23000 transistors with 60,000 instructions per second capability.
- Clock size of 740 KHz.
- Clock → is a train of pulses.

EEPROM

- Electrically erasable & programmable ROM
- Re-writing of instructions can be done multiple times with the help of a high voltage circuit.

#### → Types

- ① Serial EEPROM → allows individual bytes of data to be erased hence also called byte erasable chip.
- ② Parallel EEPROM

→ We need ~~to~~ memory to store digital information.

→

- ↳ data
- ↳ instructions
- ↳ configuration ~~to~~ Instructions

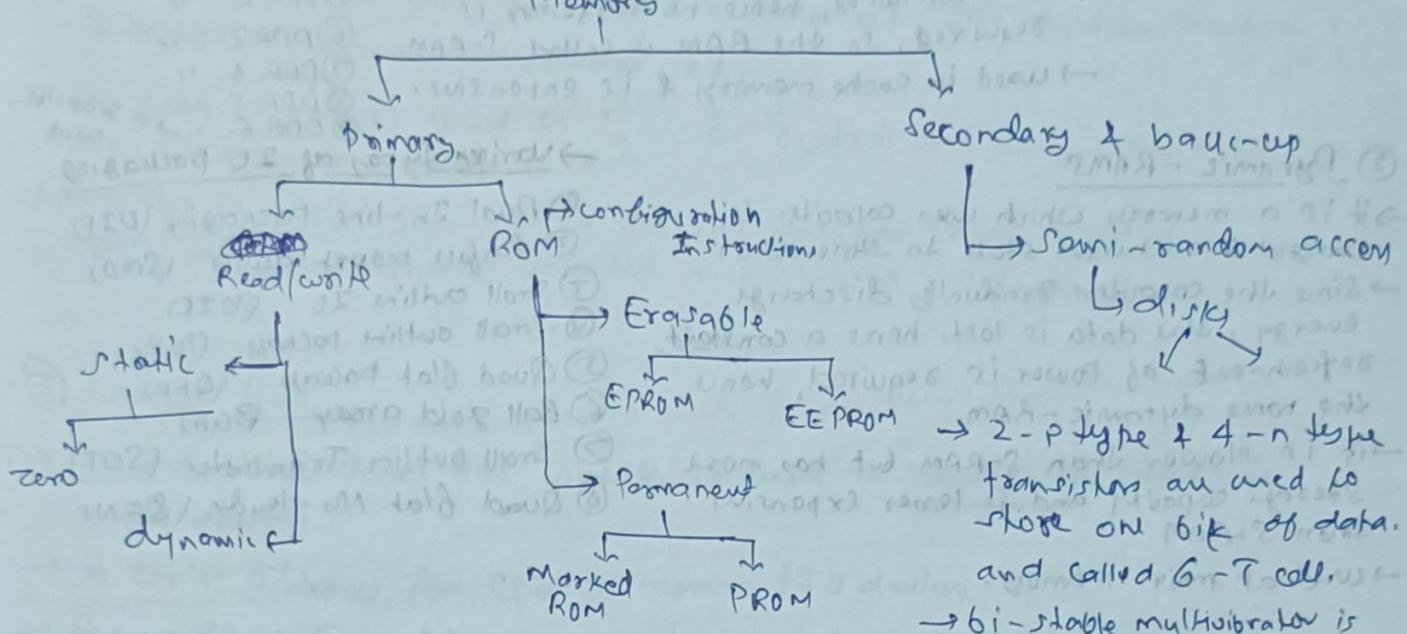
→ Primary memory is always attached to the system.

→ Secondary memory is removable to the system.

BIOS

↳ basic input output system.

### Memory



### Flash memory

- Special type of EEPROM.
- data can be serially erased & modified.
- requires PC level voltage for re-writing.

### Masked ROM

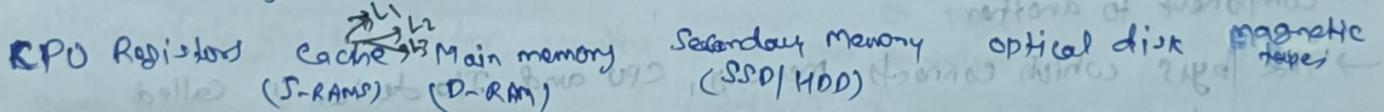
- data is burnt into the chip at the manufacturing time itself.
- data is hard-wired into permanent arrays of 0 & 1 using photo-lithographic process.

→ 2-p-type & 4-n-type transistors are used to store one bit of data and called G-T cell.

→ bi-stable multivibrator is used in S-RAM.

### Memory Hierarchy

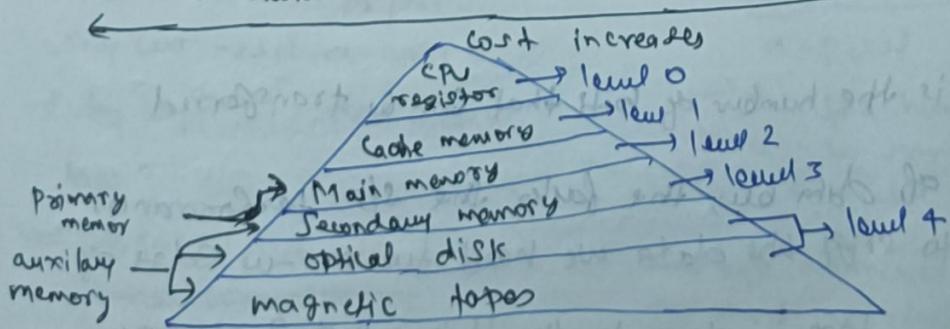
→ it is arranging different kinds of storage present on a computing device based on speed of access.



→ Read-write speed & access time increase.

→ Size increase

→ Bandwidth increases



→ 8085 has 16-bit word size.

→ 8086 has 32-bit word size.

## RAM

→ Random Access Memory is a type of primary storage which allows the user to randomly access any part of data regardless of its position in almost same time.

(Study all)

### types

- ① S-RAM
- ② D-RAM

#### ① Static - RAMs

→ it is a semi-conductor memory which uses flip-flop to store data.

→ Since semiconductor switches are used to store the data, a constant power flow is maintained, hence no refreshing is required, so the RAM is called S-RAM.

→ Used in cache memory & is expensive.

#### ② Dynamic - RAMs

→ it is a memory which uses capacitor in charged & uncharged state to store data.  
→ Since the capacitors gradually discharge energy, the data is lost, hence a periodic refreshment of power is required, hence the name dynamic - RAM.

→ it is slower than S-RAM but has more storage capacity and is lesser expensive than S-RAM

→ used at main memory.

→ different busses in a system and their need and what do they decide  
→ virtual memory.

### Types of System buses

- ① Data bus
- ② Address bus
- ③ Control bus

→ a bus is a set of electrical wires which connect various hardware components.

→ it works as a communication highway at which information flows from one hardware component to another.

→ The bus which connects memory, CPU and I/O devices is called a system bus.

#### ① Data Bus

→ it is a bi-directional bus which transmits data from memory to CPU and vice-versa.

→ the width of data bus is the number of bits that can be transferred parallelly.

→ the wider the width of data bus, the faster the S/I performance.

→ we need clock because, to shift the data we need the rising or falling edge of clock pulse.

→ if CPU & memory have same triggering clock edge then it is called synchronous.

### Types of D-RAM

- ① FPM
- ② EDO & BEDO
- ③ SDRAM & RDRAM
- ④ DDR & SDRAM
- ⑤ DDR2 ..
- ⑥ DDR3 ..
- ⑦ DDR4 ..
- ⑧ DDR5 .. in graphic card
- ⑨ DDR6 ..

### Various types of IC Packaging

- ① Dual In-line Packaging (DIP)
- ② Surface mount device (SMD)
- ③ Small outline IC (SOIC)
- ④ Small outline package (SOP)
- ⑤ Quad flat package (QFP)
- ⑥ Ball grid array (BGA)
- ⑦ Small outline Transistor (SOT)
- ⑧ Quad flat No-leads (QFN)

### ② Control bus

→ a burst is a sudden flow.

- control signals are generated at control unit of CPU.
- timing signals are used to synchronize the memory & I/O devices with clock.
- e.g.:
  - memory read → data from memory to be placed on data bus.
  - memory write → data from data bus to be placed on memory address location.
  - I/O read → data from I/O address location to be placed on data bus.
  - I/O write → data from data bus to be placed at I/O address location.
  - Interrupts, interrupt acknowledge, etc.
- the control bus is used to transfer control and timing signals from one component to another.
- the bus is bidirectional where the CPU generates various types of control signals and the peripherals connected to it respond accordingly.

### ③ Address Bus

- it is a unidirectional bus which is used to identify the addresses in the memory.
- it carries addresses from CPU to memory (I/O devices).
- the width of the address bus determines the amount of physical memory addressable by the processor.
- Hence wider the bus width, the more memory it can use at a time.
- also, the number of memory locations the processor can address =  $2^{\text{No. of add bus}}$ 
  - i.e. a 32 bit address bus addresses  $2^{32}$  memory locations.

### FPM

- allows data access faster, in the same row <sup>or page</sup>.

SATA ?  
PCI ?  
PCIe ?  
PCI Express ?  
PCIe Gen 3  
PCIe Gen 4

→ displays are totally digital.

Booleans		
R	G	B
0	0	0

a-Pixel

→ due to humongous process to handle we require GPU.

for  $1024 \times 960$   
→  $\approx 9$  million pixels

with 4 stars to maintain with  $60\text{ Hz} / 112$  refresh rate

ISVs with OpenGL or OpenCL.

GPU is used in AI/ML.

Why we need a GPU?

→ why comp. S/Pi are active low?

~~8085 / 8086 also!~~

~~How many bits no~~

- byte addressed memory? → word size 3
- word addressed memory?

→ Why memory is byte addressed?

→ linear addressing and other types of addressing, their application and benefits?

→ microprocessors

→ graphical processors

→ microcontrollers

→ DSP

→ FPGAs

→ ASIC

→ ASIP

→ TPU

→ NPU

→ PPU

→ ISP

→ SPU

→ types of microprocessor?

→ diff b/w microprocessors & DSP?

basis of selection of a

DSP

① Speed ④ no. of pins

② Price ⑤ Power consumption

③ Memory ⑥ Support from manufacturer

→ What steps take place after starting the processor.

→ Memory architectures are classified into two categories

① byte addressable ② Word addressable.

### ① Word Addressable

→ It is a type of memory architecture, in which data is stored and accessed in blocks called word, where each word size has same word size of CPU.

→ the size of a word can be 2, 4, 8 or bytes, and each word has a unique address.

→ Since more data can be accessed at a time by word addressed architecture, it is used in high level languages like C.

→ Here a word can be modified & not a single byte of data.

### ② Byte Addressable

→ It is a type of memory architecture in which data is stored and accessed as individual bytes, where each single byte has its own unique address.

→ Here each byte of data can be modified giving the programmer a direct memory access.

→ This architecture is used for microcontrollers, embedded systems and other low level systems.

→ Since one byte of data can be modified, the programmer doesn't have to align the data to word boundaries.

→ In computer design, byte addressable memory is the default memory assignment because:

- ① it eliminates the need for the programmer to align the data into word boundaries.
- ② it provides direct memory access to the programmer, where each byte can be individually modified.
- ③ At word addressed memory, each byte of data cannot be accessed individually hence efficient use of memory is restricted.

## SATA

→ Serial Advanced Technology Attachment

→ SATA is used over PATA, although it is considered that parallel transmission of data is faster than serial transmission.

→ It is so because, in parallel transmission synchronization of all the buses cannot be guaranteed to a greater extent, hence a change in bus length changes the transmission speed of each bus, making other buses to halt till all the data is not fully transmitted.

→ Also, due to crosstalk i.e., noise interference with other bus, SATA is preferred over PATA.

RTC = Real Time Clock

PCI = Peripheral Component Interconnect.

→ It was used for connecting extensions to the computer like sound cards, etc.

→ Computer signals are active low, because, digital devices are present in the S/I.

→ The digital S/I prefer active low S/I so that, less power consumption could take place, and any noise couldn't interfere with the circuit such that, at any interference, the device will turn-off, hence no false output.

## DSP & Microprocessor

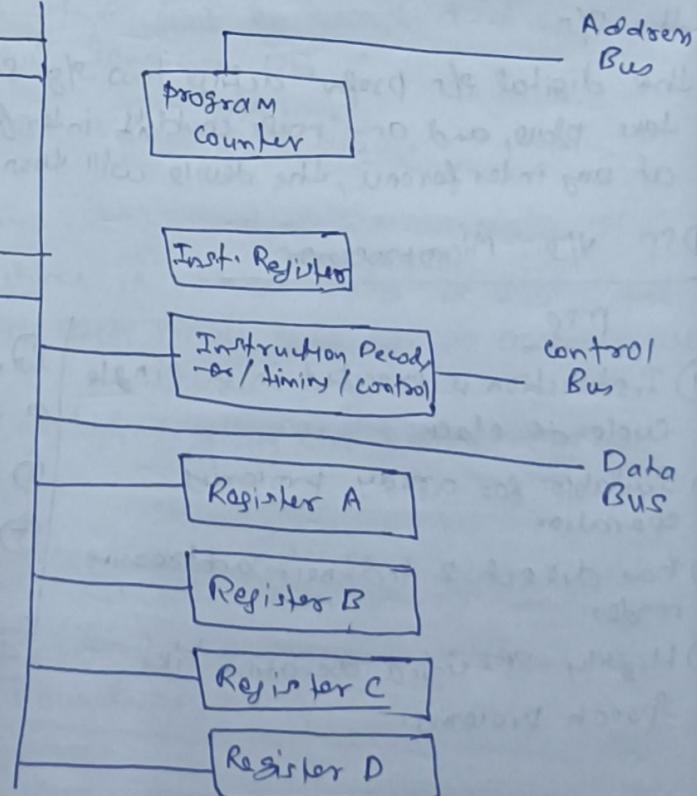
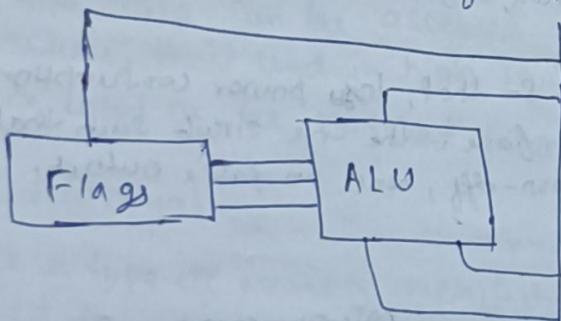
### DSP

- ① Instruction is executed in a single cycle of clock
- ② Suitable for array processing operation
- ③ Has direct & indirect addressing modes.
- ④ Highly specified operation like speech processing.

### Microprocessor

- ① Multiple clock cycle is reqd for a single instruction execution.
- ② Suitable for general purpose processing
- ③ Has many addressing modes including direct, indirect, immediate, etc.
- ④ Highly generalized application.

- Steps after starting of a processor
- ① Fetch instruction
    - instruction is fetched from main memory.
    - the instruction at current program counter is fetched & stored at instruction register.
  - ② Decode instruction
    - Here the decoder interprets the instruction present in IR.
  - ③ Perform ALU operation
    - Here the arithmetic operation takes place and output is generated.
  - ④ Store
    - the result/output so generated is stored back to memory.
- Intel 8051 was first micro-controller.  
 → Embedded processor is another name for micro-controller.
- Set of instructions is called a program.  
 → an instruction set is mnemonics. e.g.: ADD  
 → programs are a subset of instruction set.  
 → general purpose registers are like scratch pads.  
 → Assembler converts mnemonics to opcodes for the s/s to understand.  
 → flags determine the status of last executed program.



$\overline{CE}$  = chip enabled

- Intel uses HLT to end a program.
- \_\_\_\_\_ uses ENQ to end the program.
- chip select  $\overline{CE}/\overline{CS}$  selects the module of the memory to be selected
- the default value of the program counter is such that it either contains the address of the first instruction or the default address redirect to the address of the first instruction.
- Why we use < > in #include <stdio.h>
- Exact operation of operation of a processor!
- < > is written in #include <stdio.h>  
stdio.h is a file name and #include is a pre-processor directive, if we don't use < >, the command would look for the mentioned file at current as well as specified directory but with < >, it would only look at the specified directory only.
- # is a pre-processor directive, it acts as to distinguish and tells the controller to include the contents of given header file just before the starting of the program.

Inside the Processor :

- ① It contains an ALU
  - ② Many registers
  - ③ general purpose
  - ④ special purpose
- ↳

\* program counters

- ↳ register which contains address of next executable instruction  
→ is a pointer register

\* process of getting inst & memory to CPU is called fetching.

\* Instruction register contains the next instruction to be executed.

→ Program as mnemonics is converted to OPcode by inst. decoding.

→ decoders use many transistors

### Flags

→ there are registers which mould the states of the outcome.

#### ① Sign flag (S)

→ it shows whether the outcome of the operation is negative or positive.  
at -ve it is 1  
at +ve it is 0

#### ② Zero flag (Z)

→ it shows the output of the operation is zero.  
ie: if output is zero, the flag will be set ie: 1

#### ③ Auxiliary Carry flag (AC)

→ it is used in BCD number sys when the first nibble generates a carry which is to be used in the 2nd nibble.

ie: ↘ 0

→ this flag is not accessible to programmer.

## ④ Parity flag (P)

- it shows even no of 1's
- if CP has even no of 1's  
∴ P will be set to 1.

## ⑤ Carry flag (CY)

- if the output of n-bit operation is of (N+1) bits, a carry is generated and the carry flag is set.

## ⑥ Overflow flag (O)

- this flag is set to 1, if the output of the operation is too large to fit in the range of the memory.

e.g.:  $100 + 50$  doesn't lie b/w  
 $-128$  to  $127$ .

∴ O is set to 1. \* Inconclusive

## Operation of a microprocessor

- Program counter contains the default value.

- Corresponding to the default value of PC address of first instruction should be put.
- The address will put the value in address bus

- After that instruction is in data bus and reaches to instruction register

- this is called fetching.

- Then instruction passes to instruction decoder, opcode corresponding to the decoded instruction is activated.

- The output of operation will get stored in registers after the operation in ALU at an address.

- this process is called execution.

- Control unit then automatically points towards the address of the next fetched instruction in Program Counter.

- Till END/HALT is provided, the cycle is repeated.

0111 0011

## Directive

- it is an instruction to compiler only. e.g.: ORG, #

- ORG is a directive, to correspond the address of the first instruction to the default value of the program counter.

## Decoder

- it is a digital circuit, which decodes the instruction set so that ALU could understand.

- Little Endian? → big Endian

OPCODE

lower byte

LSB is stored

in the highest

address &

MSB is stored

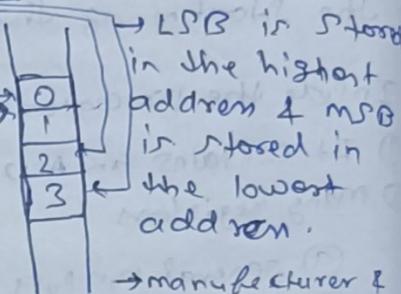
in the lowest

address &

MSB is stored

in the highest

address.



→ manufacturer & programmer must follow the same way.

→ MOTOROLA follows big Endian

→ Intel follows little Endian

→ TX Int follows little Endian & can convert to big Endian.

→ problem with Von Neumann Architecture

→ how it was solved?  
↓  
processor stays idle.

→ problem with Harvard architecture is that due to ~~the state of~~ overburdening of data bus, the processor has to stay idle.

→ Super Harvard architecture?

→ Secondary memory ~~is~~ is the data whose value is fixed, this data is present in program memory itself for Super Harvard architecture.

RISC & CISC ?  $\rightarrow$  general purpose registers in RISC and  
4096  $\rightarrow$  registers in CISC controller.

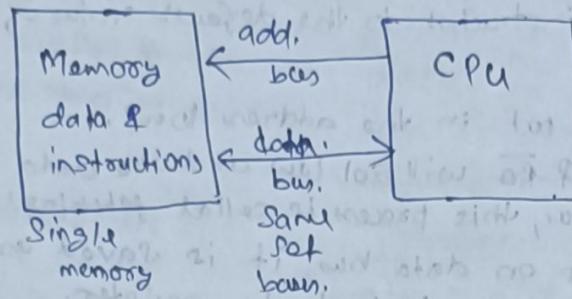
\* complete

## → operation of a microprocessor

- ① program counter contains the default value.
- ② corresponding to the default value of PC, address of the first instruction should be put.  
 $\hookrightarrow$  the ORA directive is an instruction to assembler to correspond the address of the first instruction to the default value of the program counter.
- ③ Then this address will be put in the address bus.
- ④ Now control signals  $\overline{CS}$  &  $\overline{RD}$  will go low and the data and instruction will be put in the data bus, this process is called fetching.
- ⑤ Once the instruction is put on data bus, it is saved as opcodes corresponding to that mnemonic, in instruction register.
- ⑥ The opcode will now be passed to the instruction decoder, a decoder is a digital circuit which decodes the instruction set into series of 0's & 1's so that the ALU could understand.
- ⑦ The ALU will understand the instruction and will execute the operation, and the output of the operation is saved in registers.  
 $\hookrightarrow$  this process is called execution.
- ⑧ Now, the control signals will increment the program counter to the address of the next instruction to be fetched.
- ⑨ Now again the incremented address will be put into the address bus, the  $\overline{RD}$  &  $\overline{CS}$  will go low and fetching of instruction & data takes place.
- ⑩ the instruction as in the form of opcodes go to instruction register which redirects it to decoder.
- ⑪ The decoder decodes the information and passes it on the ALU and execution takes place.
- ⑫ Again the output is saved into the registers and the program counter is incremented.
- ⑬ This process will continue until opcodes corresponding to END / HALT is reached to instruction register.
- ⑭ Then decoder will decode the opcode to ALU and ALU will execute the ending of the program.

## Von Neuman Architecture

- It is a computer architecture used in personal computers, proposed by John Von Neuman.
- Here data and instructions are stored in the same memory and are accessed by same set of address and data bus.
- ie, same address bus carries the address of both data & instruction.

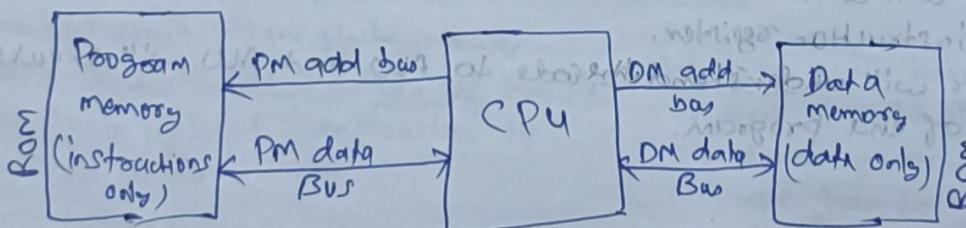


## Von Neuman Bottleneck

- Since, same set of buses are used for fetching both data and instructions, the processor has to stay idle for most of the times.
- Eg: first the instruction is being fetched, the processor will be sitting idle. Then data will be fetched, again the processor will be sitting idle. Once all the information is reached, then only execution will take place.
- due to this speed of the processing slows.

## Harvard Architecture

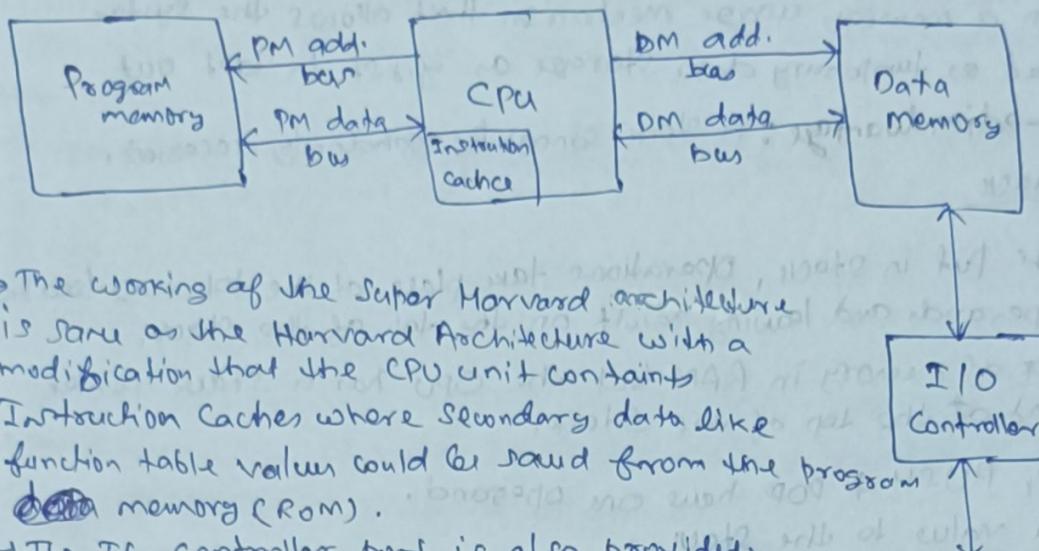
- it is a computer architecture used in DSP & microcontrollers.
- Here the dual memory contains the data and instructions and are accessed by separate set of address and data bus.
- ie, instructions will have their own address and data bus.  
data will have their own address and data bus.
- Since, separate buses are used for fetching different information, the amount of time the microprocessor stays idle is lesser.



## Problem

- Since more data read & write is required, the data bus is overburdened and hence the speed gets lowered.

## Super Harvard Architecture



- The working of the Super Harvard architecture is same as the Harvard Architecture with a modification that the CPU unit contains Instruction Caches where Secondary data like function table values could be read from the program ~~ROM~~ memory (ROM).
- The I/O controller part is also provided.
- Since Secondary data is already fetched to the instruction cache, less time is taken by the data bus to fetch the data and the microprocessor stays idle for less amount of time.

## RISC Architecture

### Reduced Instruction Set Architecture Properties

- ① Instructions are of same size.
- ② Large no of registers and hence no stack is reqd.
- ③ less no of instructions
- ④ Single cycle execution.
- ⑤ Harvard architecture
- ⑥ Micro instructions are hard wired instead of microcodes, hence faster execution.
- ⑦ Load/Store architecture, hence reduced pipeline stalling.
- ⑧ Low power

### Disadvantages

- ① costly
- ② requires more instructions to perform complex task due to limited no of instructions
- ③ requires more memory to perform complex task.

## ISA

- Instruction Set Architecture is the part of the processor visible to only programmer or compiler writer and defines how CPU is controlled by the software.
- It serves as the boundary b/w Software & hardware.

## Stack Architecture

→ Stack memory is a memory usage mechanism that allows the system memory to be used as temporary data storage as first in last out scheme.  
→ disadvantage: A stack cannot be randomly accessed.

### \* Operation in Stack

- ① The operands are put in stack, operations take place at the top two locations destroying the operands and leaving result on the top of the stack.
- ② Stack is a block of memory in RAM but the CPU has a Stack pointer register to point at the top of the Stack.
- ③ Two operations, PUSH & POP have one operand.
  - i.e., ④ Push a value to the stack.
  - ⑤ Pop a value from the top of the stack to a destination.
- Other operations have implicit operands which are at the top of the stack.
- ④ To perform an operation, the operands are first pushed into stack, then operation on it is performed, then those operands are deleted from the stack and the result of the operation is stored at the top of the stack.
- ⑤ The POP command, then stores the result back to the memory.

## Accumulator Architecture

- A CPU with an accumulator-based architecture stores the intermediate results into the special purpose register called accumulator.
- ### \* Operation
- ① The operand in the accumulator is loaded from memory using LOAD command.
  - ② Then the operation is done with other operand which is present in register or memory.
  - ③ After the calculation, the result is stored in the accumulator, since accumulator is the default saving address.
  - ④ Then the result of the accumulator is stored back to memory using the command opcode STORE.

→ It is still used in DSP.

Disadvantage: the accumulator is only temporary storage, so memory traffic is very high for accumulator approach.

## Register - Set Architecture

{ Study in detail later }

→ General Purpose Registers are extra registers which are present in the CPU and are utilized for storing data or memory locations.  
they hold:

- ① Operands for logical and arithmetical operations
- ② memory pointers
- ③ Operands for address calculation.

→ It is the most commonly used in modern computers.  
→ it allows fast access to temporary values, permits clever optimization of the compiler but the instruction sets are longer than accumulator designs.  
→ operations here, require to specify all the operands explicitly.  
→ this architecture is of 3 types

- ④ register - register
- ⑤ register - memory
- ⑥ memory - memory.

## Pipelining

→ it is a technique of decomposing a sequential process into sub-operations such that each sub-operations could be executed in a dedicated segment that operates concurrently with all other segments.  
ie; Several computations can be in progress in distinct segments at the same time.  
→ the overlapping of the computation is made possible by interface registers which hold the output between two stages. These interface registers are called latch or buffer.  
→ All the stages of the pipeline is connected by the same common clock including the interface register.

## Pipeline Stalling

→ Pipeline requires instructions to be executed in a specific order. If a processor is not designed properly, the pipelining will be affected and hence processing of an instruction will be delayed, this delay is called pipe-line Stalling.

## CISC Architecture

→ complex instruction set architecture.

Here, a single instruction does the all, loading, evaluating and storing operation, hence complex instruction.

## Characteristics

- ① complex instruction & its decoding is complex.
- ② instruction size is larger.
- ③ instructions take more than one cycle to complete.

④ less no of general purpose registers are required as operations might get performed in the memory itself.

### advantage

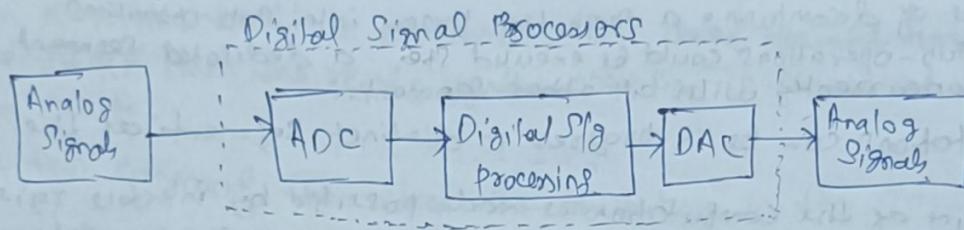
- ① Since complex instruction set, hence reduced code size.
- ② more memory efficient due to complex instruction set as fewer instructions perform complex tasks.

### disadvantages

- ① Since complex instruction set, hence it might take more time to execute.
- ② design is complex
- ③ consumes more power.

### DSP

→ A digital signal processor is a specialised microprocessor designed specifically for digital signal processing algorithms, generally in real time computing.



### benefits of digital s/s

#### → Noise Rejection

Expected value = 4.5 V

~~got value~~ got value = 4.2 V  $\therefore$  noise = 0.3

got value = 3.8 V  $\therefore$  noise = 1.7

→ Since analog domain is continuous, a noise is evident while the digital domain has only states, 0 or 1, hence therefore, 4.2 gets approximated to 5, hence noise gets rejected.

→ off-line process

→ Environmental effects resistant

→ Ageing Resistant

→ implementation is easy

→ modification is easy

→ saving in memory is easy.

→ compact size.

Here the entire signal resides at the computer at the same computer at the same time.

e.g.: MRI scan, data is acquired & stored and image generation takes time.

## Thermal Runaway in BJT

- Temperature coefficient: State the variation of resistance with change in temperature.
- for a device with negative thermal coefficient, the resistance decreases with increase in temperature.
- for NTC devices like diode, BJT, parallel operation is prohibited because that could lead to thermal runaway.
- As we connect two NTC devices in parallel, the current through them will increase, this increase in current will cause heat to lower down the resistance of the device, resulting in further increase of current causing more heat and further lowering of resistance, ultimately this will lead to breaking down of the system and is called Second Breakdown or Thermal runaway.

### For BJT

- mostly BJT is connected in CE configuration because in this configuration both current and voltage gain is higher while CC gives just good current gain and CB gives just good voltage gain.

### for CE configuration

$$I_C = \beta I_B + (\beta + 1) I_{CBO}$$

$I_{CBO}$  is due to minority charge carriers

- Since the concentration of minority charge carriers are dependent on temperature, a rise of temperature increases their concentration.
- with rise in concentration of minority charge carriers,  $I_{CBO}$  increases due to which  $I_C$  increases.
  - flow of  $I_C$  causes heat which further increases the minority carriers leading to more  $I_{CBO}$  & hence more  $I_C$ .
  - Due to this cycle the transistor will get damaged.
  - ∴ Self destruction of transistor is called thermal runaway.

### Mitigation of thermal runaway

#### ① by negative feedback

i.e., providing emitter resistance due to which

if  $I_E \uparrow$ , then  $I_B \downarrow$  which will cause  $I_C \downarrow$  or  $I_C = \beta I_B$

#### ② with heat sinks

### Real time

A real time system is the one in which responses should be guaranteed within a specified timing constraint or the system should meet the specified deadline. Here output signal is produced at the same time that the input signal is being acquired."

MAC → multiplication & addition calculation

→ clearly the DSP requires most of its functionality in terms of addition & multiplication, hence if the hardware is more equipped to do such it would be easier to get the result. Clearly the SIS should be such that multiplication & addition could take place & their output could be stored simultaneously.

SIMD? MMX?

## Parallel Computing

→ Here the job is broken into discrete parts which can be executed concurrently. Each part is broken down into a series of instructions and instructions from each piece will execute on different CPU's simultaneously.

→ Flynn's taxonomy is a classification scheme for computer architectures based on the number of instruction streams & data streams which can be processed simultaneously by a computer architecture.

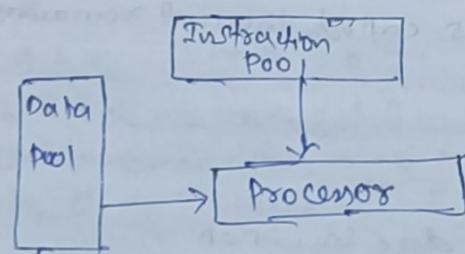
### Four categories in Flynn's taxonomy

- ① SISD (Single instruction single data)
- ② SIMD (Single instruction multiple data)
- ③ MISD (Multiple instruction single data)
- ④ MIMD (multiple instruction multiple data)

#### ① SISD

→ it is a uni-processor machine that is capable of executing a single instruction operating on a single data stream, where instructions are processed in a sequential manner.

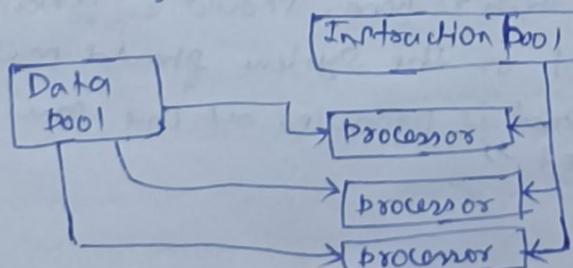
→ Here the speed is dependent on rate of internal data transfer.



#### ② SIMD

→ it is a multi-processor machine which executes same instruction on different data streams.

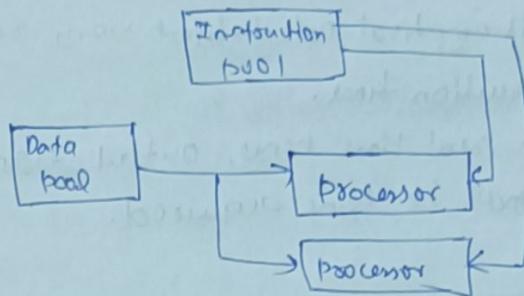
→ Application in scientific computing like DSP.



### ③ MISO

→ It is a multiprocessor machine capable of executing different instruction sets on same data set.

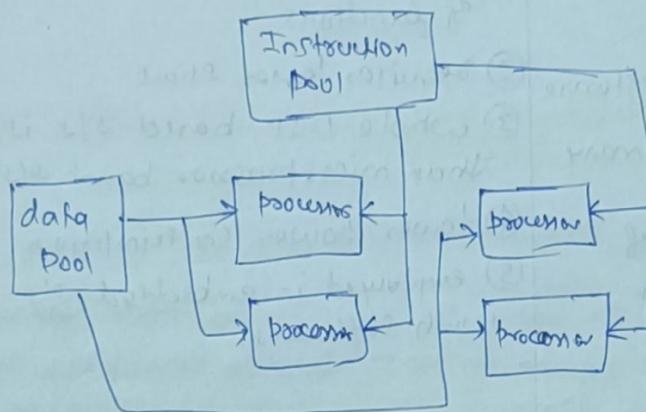
e.g.:  $\sin(\omega) + \cos(\omega) + \tan(\omega)$



### ④ MIMO

→ It is a multiprocessor machine capable of executing different instructions sets on different data sets.

→ Program executions are executed here asynchronously unlike SIMD & MISO.



## Digital Signal Processors

### → About DSP

- ① easy filtering
- ② remove unwanted interference
- ③ amplify certain frequency
- ④ low cost, small and consumes less power
- ⑤ works in real time & captures & processes data.

### → Advantages

- ① ease of processing
- ② thermal drift and reliability
- ③ Repeatability
- ④ Immune to noise
- ⑤ Programmability

### → DSP Characteristics

- ① real time processing
- ② optimum performance with streaming data.
- ③ Harvard architecture
- ④ Special instructions for SIMD operations

### → Uses of DSP

- ① FFT
- ② FIR/IIR digital filters
- ③ Convolution
- ④ Moving average
- ⑤ Wavelet transform.

→ All processors are capable of data manipulation & mathematical calculation but optimization of a processor to do both is not economical.

→ Normal regular microprocessors are designed for data manipulation but DSP are designed to perform mathematical calculations.

↳ storing &  
sorting of  
data

→ for DSP

Execution speed is limited by number of multiplication & additions reqd.

→ to perform mathematical calculations very rapidly, DSP must have a predictable execution time.

→ DSP work in real time hence, output signal is produced at the same time input signal is being acquired.

### Microprocessors

- ① range of general purpose functions
- ② run large blocks of software
- ③ are fast and handles many tasks at once.
- ④ controls huge amount of memory, data & computer peripherals
- ⑤ larger size, cost & power consumption

### DSPs

- ① dedicated, fast at smaller range of functions
- ② requires lesser space
- ③ whole DSP based S/R is cheaper than microprocessor based S/R.
- ④ lower power consumption
- ⑤ employed in embedded S/R accompanied with software.

### \* Circular Buffers

→ To calculate the output sample, we must have access to 9 certain numbers of the most recent samples from the input.

Eg: Suppose we have eight coefficients in a filter,  $a_0, a_1, \dots, a_7$ . This means we must know the value of eight most recent samples from the input signal i.e.,  $x[n], x[n-1], \dots, x[n-7]$ .

→ The best way to manage these stored samples is by circular buffering.

→ The idea of circular buffering is that the end of the linear array is connected to its beginning i.e., 20041 is being viewed as next to 20048 or 20045 to 20044.

→ Tracking of the array is done by a pointer which indicates the most recent sample besides.

Eg: in ① the pointer contains the address 20049 while in ② it contains 20044.

→ Here, as a new sample is acquired, it replaces the oldest sample in the array and the pointer is moved one address ahead.

→ Circular buffers are efficient because only one value needs to be changed when a new sample is acquired.

→ More buffer means to keep my ~~data~~ dedicated data in dedicated space i.e., dedicated memory area.

20039	
20040	0.910 ← $x[n-3]$
20041	0.1762 ← $x[n-2]$
20042	-0.761 ← $x[n-1]$
20043	-0.123 ← $x[n]$
20044	0.124 ← $x[n-7]$ → 0
20045	0.178 ← $x[n-6]$
20046	0.167 ← $x[n-5]$
20047	-0.891 ← $x[n-4]$
20048	
20049	

20039	0.910 ← $x[n-9]$
20040	0.762 ← $+x[n-3]$
20041	-0.761 ← $x[n-2]$
20042	-0.123 ← $x[n-1]$
20043	0.124 ← $x[n]$
20044	0.198 ← $x[n-7]$ → 0
20045	0.167 ← $x[n-6]$
20046	-0.891 ← $x[n-5]$
20047	
20048	
20049	

already ① idea develops economy of buffer if collected old → making add first & then

newly arrived samples into buffer after removing old to free

Oldest 20049

Newest 20043

Oldest 20045

Newest 20049

### → Advantages of Circular Buffer

- ① Efficient use of memory
  - ② Can be used for LIFO & FIFO
- Advantages of Circular Buffer over Linear Buffer
- ① It is not possible to fill the data if the pointer has reached the rear location in a linear buffer but is possible in circular buffers.
  - ② Data can be easily added & removed.

Memory Management

### → for configuration of Circular Addressing

- ① A pointer that indicates the start of the circular buffer in the memory.  
Eg: in this case 20040.
- ② A pointer indicating the end of the array Eg: 20047 or a variable that holds the length Eg: 8 bit
- ③ The step size of memory addressing must be specified.  
→ These three values define the size and configuration of the circular buffer and will not change during the program operation.
- ④ The pointer to the most recent sample must be modified as each new sample is acquired.  
→ There must be program logic that controls how this pointer to the most recent sample is modified which would be based on the first three parameters.

## Hardware Units in DSP Processors

- ① modified bus structures and memory access schemes
- ② multiple access memory & multipported memory
- ③ Instructional level parallelism
- ④ VLIW or Very long instruction word architecture.

### ① Modified Bus Structures and memory access schemes

→ Here we could use modified Harvard architecture where filter coefficients would be put in program memory and the input signal in data memory.

#### Memory Access Schemes

- I/O controller is connected to data memory through which signals enter & exit the system.
- most of the processors contain both serial and parallel communication ports
- dedicated hardware allows these data streams to be transferred directly into memory without having to pass through the CPU registers.  
i.e., Direct memory allocation or DMA.
- \* these types of high speed I/O is a key characteristic of DSP.
- Some DSPs have onboard analog to digital & digital to analog converters, a feature called mixed signals.

### ② Multi Access & multipported memory

#### a) multi access memory

→ the number of memory access per clock cycle can also be increased by using a high speed memory that permits more than access per clock cycle. e.g.: DARAM.

→ Multiple access RAM can be connected to the processing unit of the Harvard architecture.

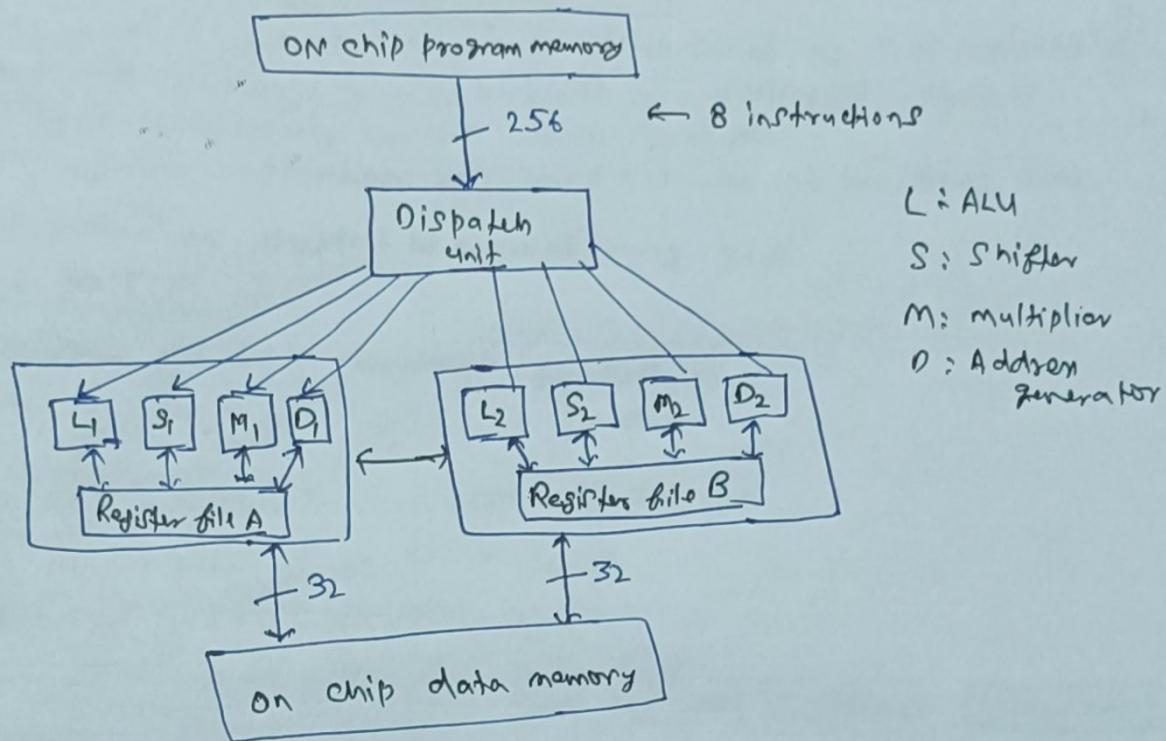
#### b) multipported memory

→ Here data & program is stored in the same memory chips but are accessed via different set of buses.

## VLIW

→ very long instruction word architecture

- \* Here the processor has a number of functional units  
ie; independent ALU, MAC units, shifters, etc
- \* The multiple functional units share the common multipported register file for fetching the operand and storing the results.
- \* the compiler determines the ILP and schedules the functional unit.



→ for 8-instructions of 32-bits each, 256 bytes are laid out between chip program memory and the dispatch unit.

→ The dispatch unit decides which instruction will go to which functional unit.

### \* Characteristics of VLIW

- ① simple & regular instruction set
- ② Instruction Scheduling ie, the order in which instruction to be executed is determined at compile time.
- ③ Order of execution doesn't get changed while program execution

## Instruction level Parallelism

- It is the kind of architecture in which multiple operations can be performed parallelly in a particular process, having its own set of resources, like address space, registers, program counters.
- Here multiple operations are performed in a single cycle, either by executing them simultaneously or by utilizing gaps between two successive operations at which processor stays idle.
- Here the compiler decides when to execute an operation.

## Difference b/w ILP & Pipelining

- In Pipelining the sequential program is decomposed of sub-operations such that sub-operations could be executed in a dedicated segment that operates concurrently with all other segments.
- In ILP, multiple instructions are being executed at the same time.
- Dependent instructions shouldn't be executed using ILP.

## Requirement for ILP

- ① Independent set of instructions for software.
- ② Independent functional units in hardware.

\* Available ILP : Inherent in a region of the code.

\* Achievable ILP :: Provided by the hardware

DSK Available ILP  $\geq$  Achievable ILP

e.g: ADD 6,7  
ADD 14,10  
ADD 11,12  
Shift A,B

Available ie, 4

ADD	Shift	Multiplication	Load/Store
-----	-------	----------------	------------

Achievable  
only 2, because ADD is being used multi-times.  
L 3 clock cycle reqd.

Cycle	OPERATION
1	$y_1 = x_1 \times 1010$
2	nop
3	nop
4	$y_2 = x_2 \times 1100$
5	nop
6	nop
7	$z_1 = y_1 + 0100$
8	$z_2 = y_2 + 0101$
9	$t_1 = t_1 + 1$
10	$p = q \times 1000$
11	$clr = clr + 0010$
12	$r = r + 0001$

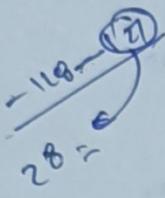
Sequential operations  
no of cycles = 12.

Cycle	INT ALU	FLOAT ALU	INT ALU	FLOAT ALU
1	$t_1 = 2t_1 + 1$	$clr = clr + 0010$	$y_1 = x_1 \times 1010$	$z_2 = z_2 \times 1100$
2	$r = r + 0001$			$p = q \times 1000$
3	nop			
4	$z_1 = y_1 + 0100$	$z_2 = y_2 + 0101$		

ILP improves the performance  
no of cycles = 8

→ word size means how many bits a CPU can process simultaneously. e.g., 8085  $\rightarrow$  8 bits  
8086  $\rightarrow$  16 bits

- Hardware point of view of meaning of word size
    - data bus size = word size
    - no of registers = word size.
    - ALU capable of handling that much data simultaneously



L = add/Sub

→ high band width  
mean in one cycle

S = shift

8 ~~tele~~ Inst.

M = mult

D = Load/store

L, each has p & d means are separate

or both have same memory.

→ guard bits,

extra bits in large register  
to make sure no bits in q10  
is lost.

## TMS320C6000 Architecture

6713

MANARD

→ different level on org & for speed  
or size & speed &

Bit reversal addressing for convolution

bitellin - 15 show  
4/10 exec in 6713

fidelity → how accurately analog signal is represent..

→ unclear how accurate.

int → no decimal available

5.75 → 5

float → decimal available 5.78 → 5.78

↪ at 32 bit

1 ~~for~~ for sign

8 for ~~exponent~~ exp

23 for ~~frac~~ fract

↪ 1 at 04 bit

↪ 11 exp

↪ 52

Single precision

char size 16 bit

INTL

reason for having conditional execution

→ to reduce the int time reqd

to check condn, with faster speed.

→ regarding pipeline?

→ polling

→ interrupt