Computer Architecture (EC 504)

Credit 3

3rd year, Batch 1,2,3

Suman Paul
Assistant Professor, Dept of ECE,
Haldia Institute of Technology
(An Autonomous Institute under MAKAUT)
Online classes during COVID 19 pandemic

Roadmap to subject, Syllabus and course contents, online Class Lectures mechanisms and protocols , Introductory and consecutive Class Lectures

Batch 1,2,3

Contents :

Review of previous concepts and interactions to get ready with this subject

Rules and practices in online class for effective learning considering online platform and various QoS/QoE parameters/ issues and concerns. How to ask a query ? Submission of day to day assignments

Attendance requirements

Why we should study this subject ?

Detailed illustrations/ Overview of module wise course contents/ syllabus : How to proceed ? Including learning methodology/outcome

Text / Reference book suggested including recommended MOOCs

How to prepare as per examination point of view during online/ offline (flipped classroom or Blended learning concept.

Any queries/ issues as suggested by students/ Instructor or Chair

Prepared by Suman Paul, Assist Prof, Dept of ECE, Haldia Inst. of Tech., Haldia, WB 721657

Most interesting is parallel processing, also ongoing research :fundamental of multi core, vital for modern processor architecture

Previous concepts

Detailed illustrations/ Overview of module wise course contents/ syllabus : How to proceed ? Including learning methodology

Basic Structure of Computers, Functional units, software, performance issues software, machine instructions and programs, Types of instructions, Instruction sets: Instruction formats, Assembly language, Stacks, Ques, Subroutines.

No. formats --digital

Processor organization, Information representation, number formats.

Multiplication & division, ALU design, Floating Point arithmetic, IEEE 754 floating point formats

Concept

Control Design, Instruction sequencing, Interpretation, Hard wired control - Design methods, and CPU control unit. Microprogrammed Control - Basic concepts, minimizing microinstruction size, multiplier control unit. Microprogrammed computers - CPU control unit

Memory organization, device characteristics, RAM, ROM, Memory management, Concept of Cache & associative memories, Virtual memory

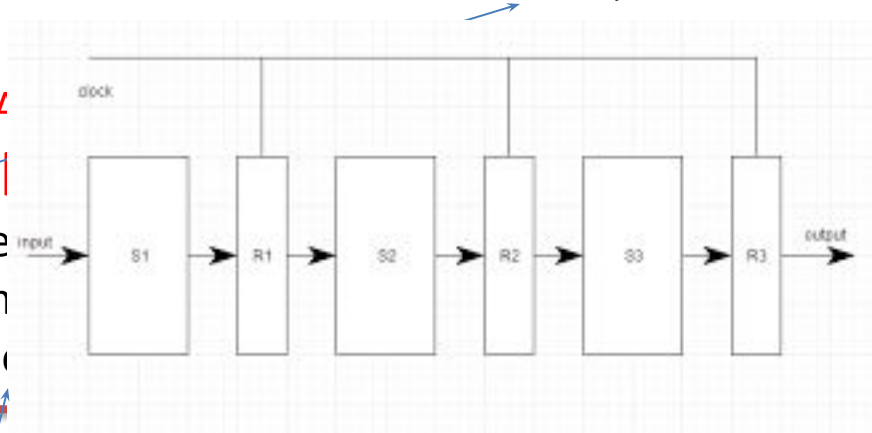
System organization, Input - Output systems, Interrupt, DMA, Standard I/O interfaces

Concept of parallel processing, Pipelining, Forms of parallel processing, interconnect network

Details of Memory organization

But we must too concentrate on the following vital topics in details :

L1, L2..

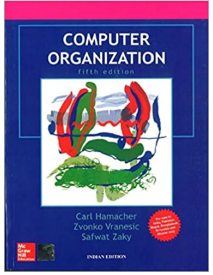


Memory organization, device characteristics, RA
and management techniques , Concept of **Cac**
Cache memory organizations, Techniques for re
Hierarchical memory technology: Inclusion, Coh
properties, **Virtual memory**, memory replacem

Details of Pipelining, instruction and arithmetic pip
control hazards and structural hazards, techniques for
hazards, Pipeline optimization techniques. Flynn's class
MISD, MIMD architectures, VLIW processor architectures, Array and Vector processors.

Modern Multi-core processor and load balancing (basic concepts)

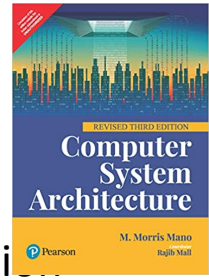
Suggested Text / reference books



1. V.Carl Hammacher, “Computer Organisation”, Fifth Edition. McGraw Hill

2. A.S.Tanenbum, “Structured Computer Organisation”, PHI, Third edition

3. Y.Chu, "Computer Organization and Microprogramming”, II, Englewood Chiffs, N.J., Prentice Hall Edition



4. M.M.Mano, “Computer System Architecture”, 3rd Edition, Pearson Education

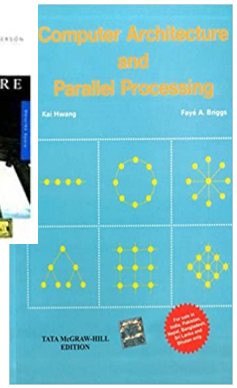
5. C.W.Gear, “Computer Organization and Programming”, McGraw Hill, N.V. Editi

6. Hayes J.P, “Computer Architecture and Organization”, Mc Graw Hill, Second edition

Cache memory problems

7. Computer architecture, *A Quantitative Approach*, John L Hennessy and David A Patterson, Elsevier (Morgan Kaufmann), 5th Edition.

8. Computer Architecture And Parallel Processing , Kai Hwang, Tata Mc GrawHill



Parallel processing, pipelining, Cache..

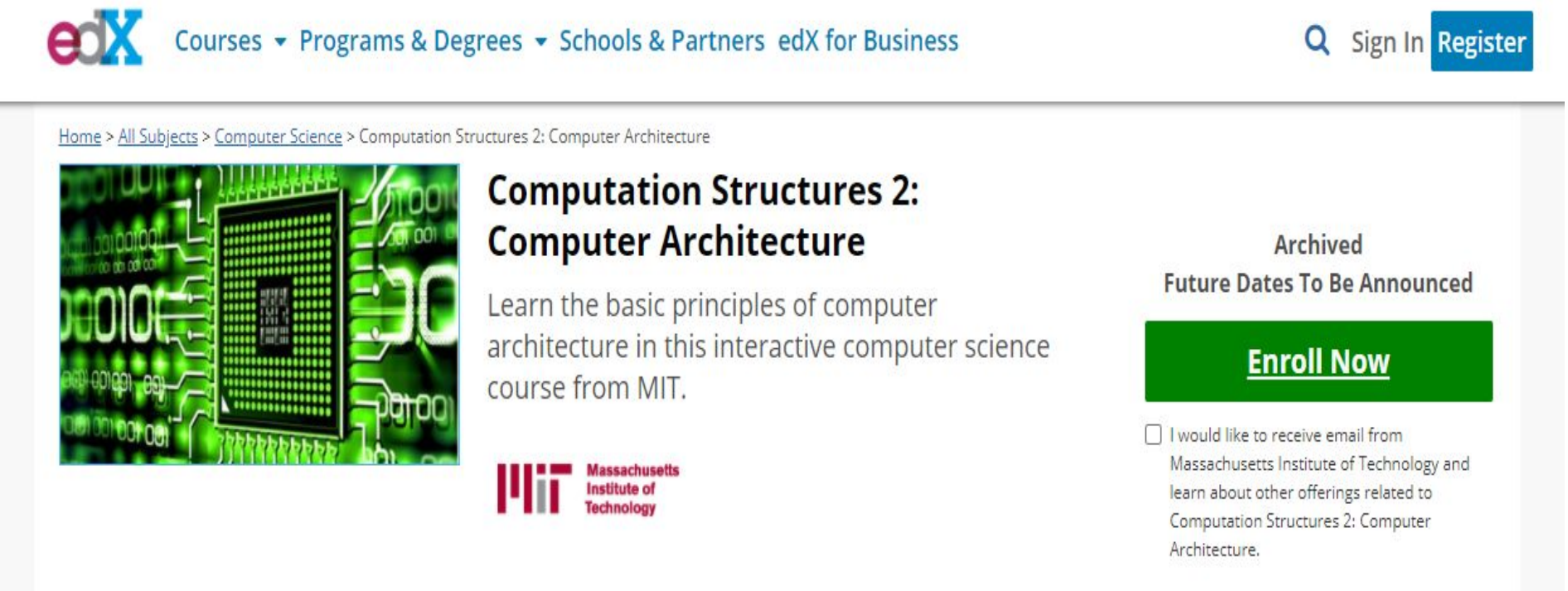
Recommended online resources / MOOCs

1. MOOC of "Computer Architecture" by Prof. David Wentzlaff offered by Princeton University through Coursera, Available : https://www.coursera.org/learn/comparch

Recommended online resources / MOOCs

2. MOOC of “Computation Structures 2: Computer Architecture” through edX

Available:
https://www.edx.org/course/computation-structures-2-computer-mitx-6-004-2x



Let's Begin our journey to Computer Architecture ...

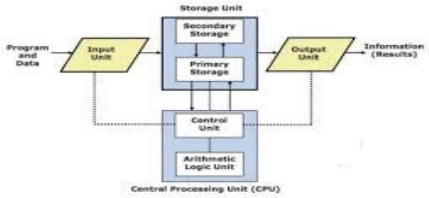
What is Computer Architecture ?

CA

- **Computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems. Some definitions of architecture define it as describing the capabilities and programming model of a computer but not a particular implementation.**

>>conceptual design and basic overview of the computer system. It defines the parameters of the computer system those are visible to the user. It deals with the attributes that have direct impact on the execution of a program. It includes : instruction formats, instruction addressing modes, instruction sets, I/O mechanism, etc.

This is the study of structure, behaviour and design of the computers/ computer systems.



Organization refers to "operational units and their interconnections"

What is Computer organization ?

CO

- **Computer organization is the operational units/ constituent parts and their interconnections that realize the architectural specifications.**

  : hardware details, system interconnect components, interfacing between CPU and peripherals , memory technology.

So, CA VS CO ??

Structure and behaviour of various functional units => CA include whether multiply / other instruction in its instruction set ? Is an archi.. issue

The way the hardware components are connected => CO whether that specific instruction will be implemented by a special hardware unit ..? Is an organizational issue.

Therefore,

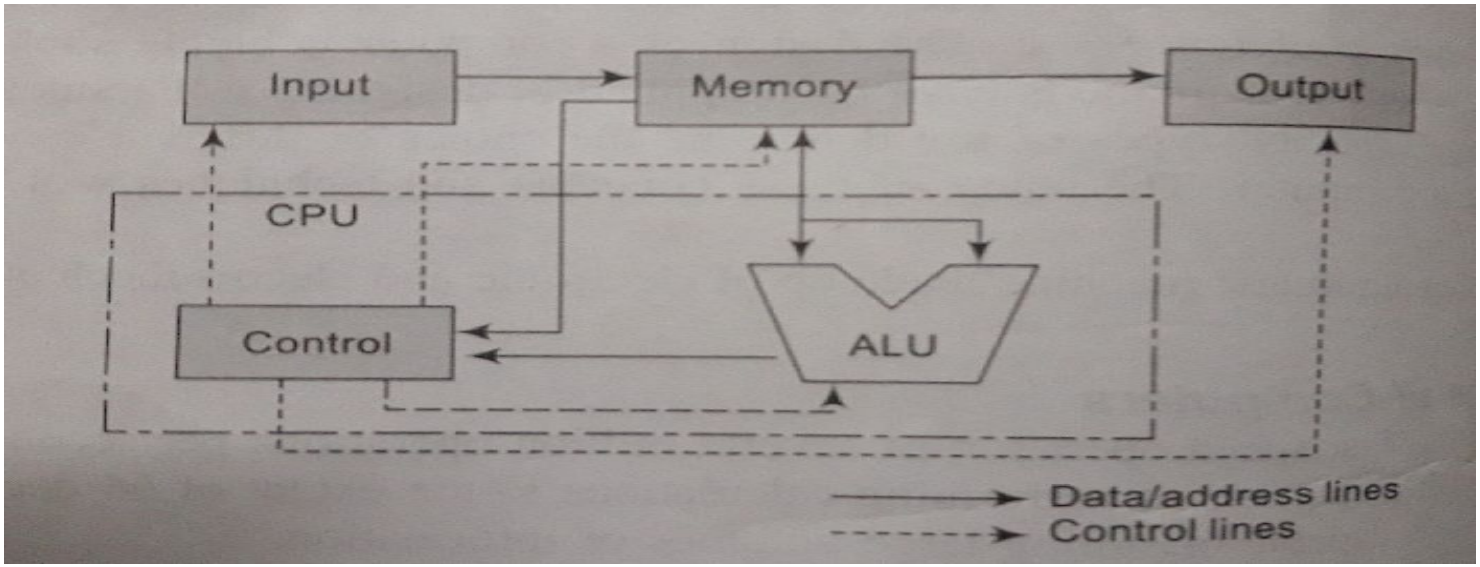
Basic concepts

Characteristics of a computer/ digital computer

Batch 2

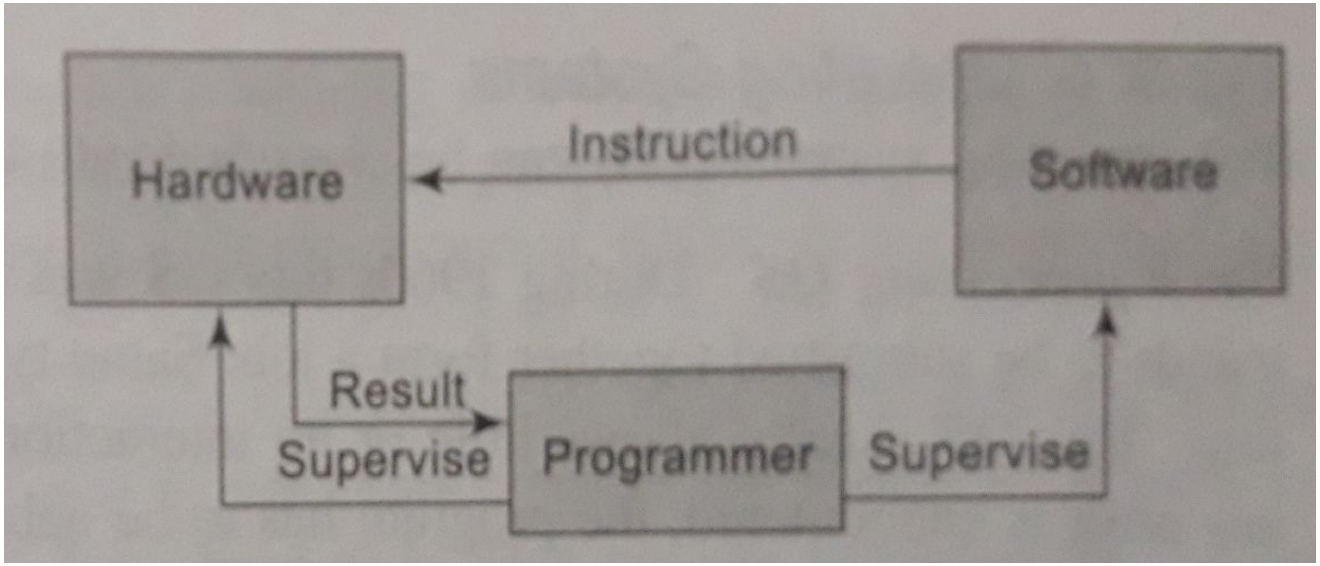
- **Speed output >>>no of operations to be completed in a faster way..**
- **Accuracy >>> capability of performing calculation/ computation to extent of decimal accuracy**
- **Storage >>> capacity of storage**
- **Decision making >>> capability of decision making as per instructions and supplied information**

A Digital computer Basic building block



Basic concepts

Block diagram of software –hardware-programmer interface



Various types of Computers

Mainframe : Large systems having many ICs, physically distributed, intensive computational tasks, shard by multiple users connected to the computer through multiple terminals

Example: IBM system 360, UNIVAC 1100/ 2200 series

Minicomputer: Slower and smaller ver. of mainframe computer, Accepts multiple users simultaneously

Example: CDC 1700, HP 3000 series

Microcomputer: With the invention of microprocessor, normally single user machine

Example: IBM PC of Intel 8086 family

Supercomputer: Expensive and powerful computer, performs multiprocessing and parallel processing, used and designed complex scientific applications

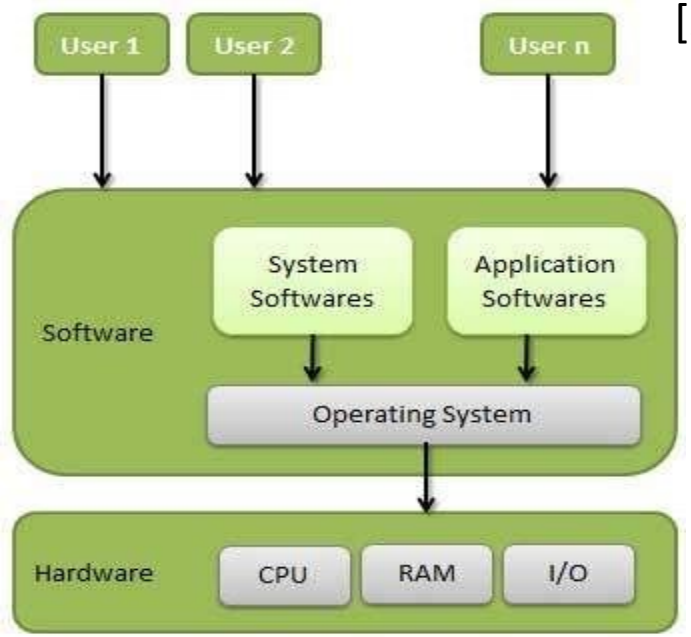
Example: Cray 1, Power PC

Operating System

Operating System

[1]

[2,3]

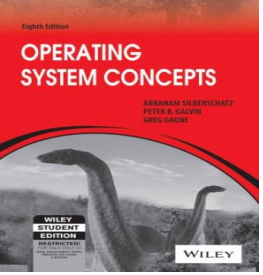


Resource allocation>> OS as resource allocator, Program execution, Process management, Memory (main) management, I/O system management and operation, File management and File-system manipulation, Communication including IPC, Error detection, accounting, protection, networking.....

Available: 1.https://www.howtogeek.com/361572/what-is-an-operating-system/

2. Operating System Concepts by Abraham Silberschatz (Author), Peter B. Galvin (Author), Greg Gagne

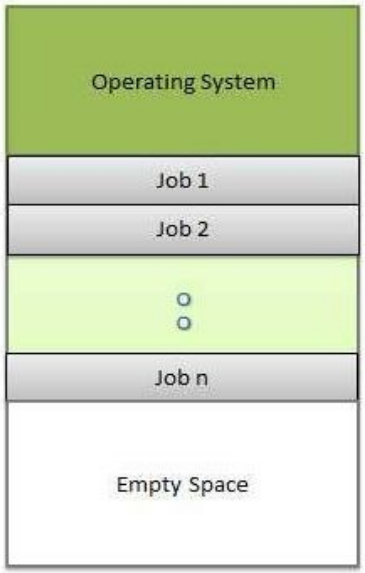
3. https://www.tutorialspoint.com/operating_system/os_overview.htm



Suggested book : **Operating System Concepts by Abraham Silberschatz (Author), Peter B. Galvin (Author), Greg Gagne...suggested Reading ..Ch 1 and Ch2**

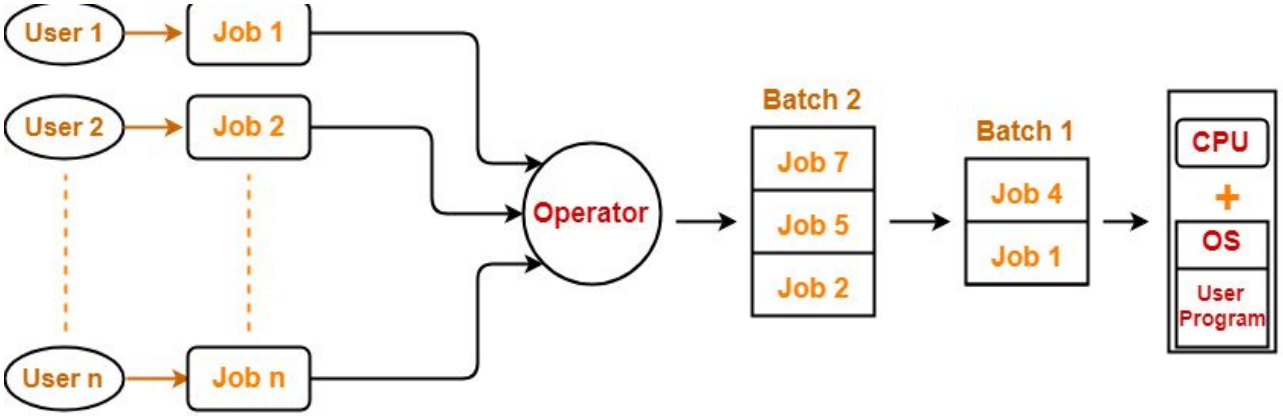
OS Revisited

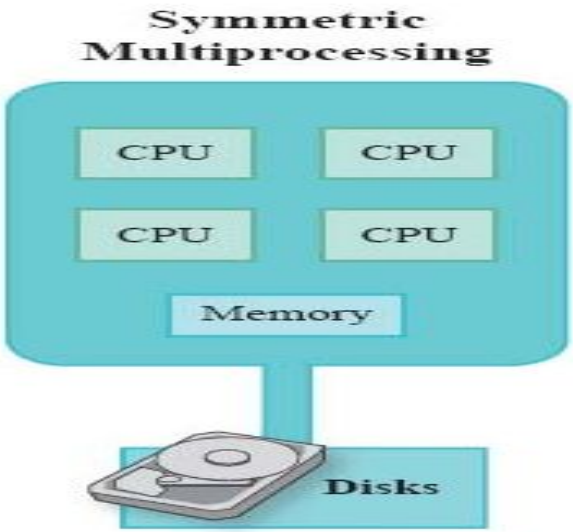
Operating System : Various Types

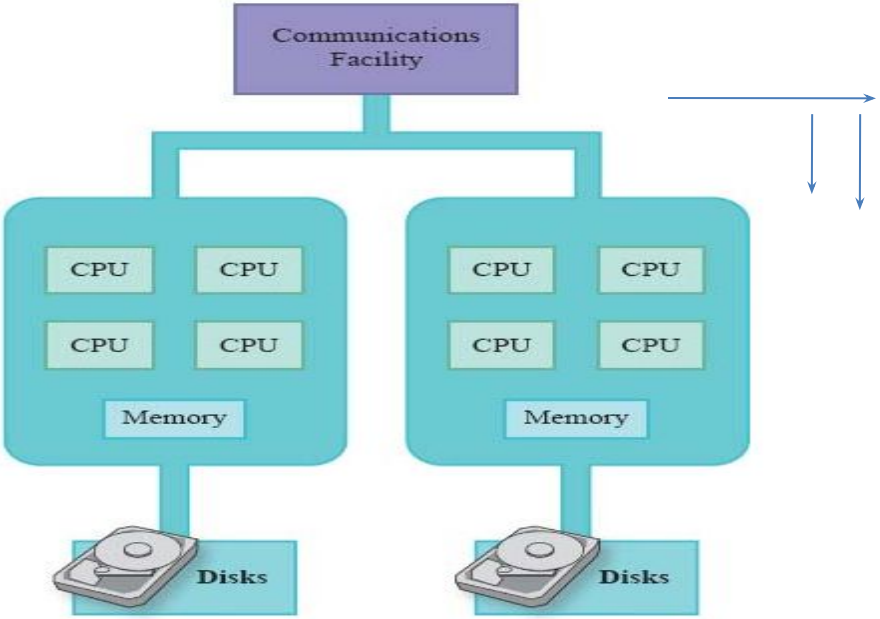


- **Bath processing system / OS**
- **Multiprogramming OS**
- **Time-sharing or Multi-tasking OS**
- **Multithreading OS: Diff. Parts of a program /thread (light weight process) to run or execute concurrently. .. Unix, and Linux**
- **Real time system ....> Hard & Soft .. Real-time OS Pre-defined timing constraint**
- **Distributed OS ... Program codes may be shared to execute task using high speed n/w. CPU s have attached main memory...computation speed high, resource sharing**
- **Multiprocessing OS ...having 2 or more CPUs controls the function, each CPU contains a copy of OS, communicate with one another to coordinate operations. The use of multiple processors allows the computer to perform calculations faster, since ....multiprocessing OSs divide the work up into various subtasks and then assign these subtasks to different central processing units (CPUs). .. Or tasks can be divided up between processors..... Normally a large shared memory Linux, Unix**

Batch processing







Symmetric multiprocessing

Clustered symmetric multiprocessing

IAS/ Von-Neumann Computer and its architecture

Von Neumann architecture was first published by John von Neumann in 1945

Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory.

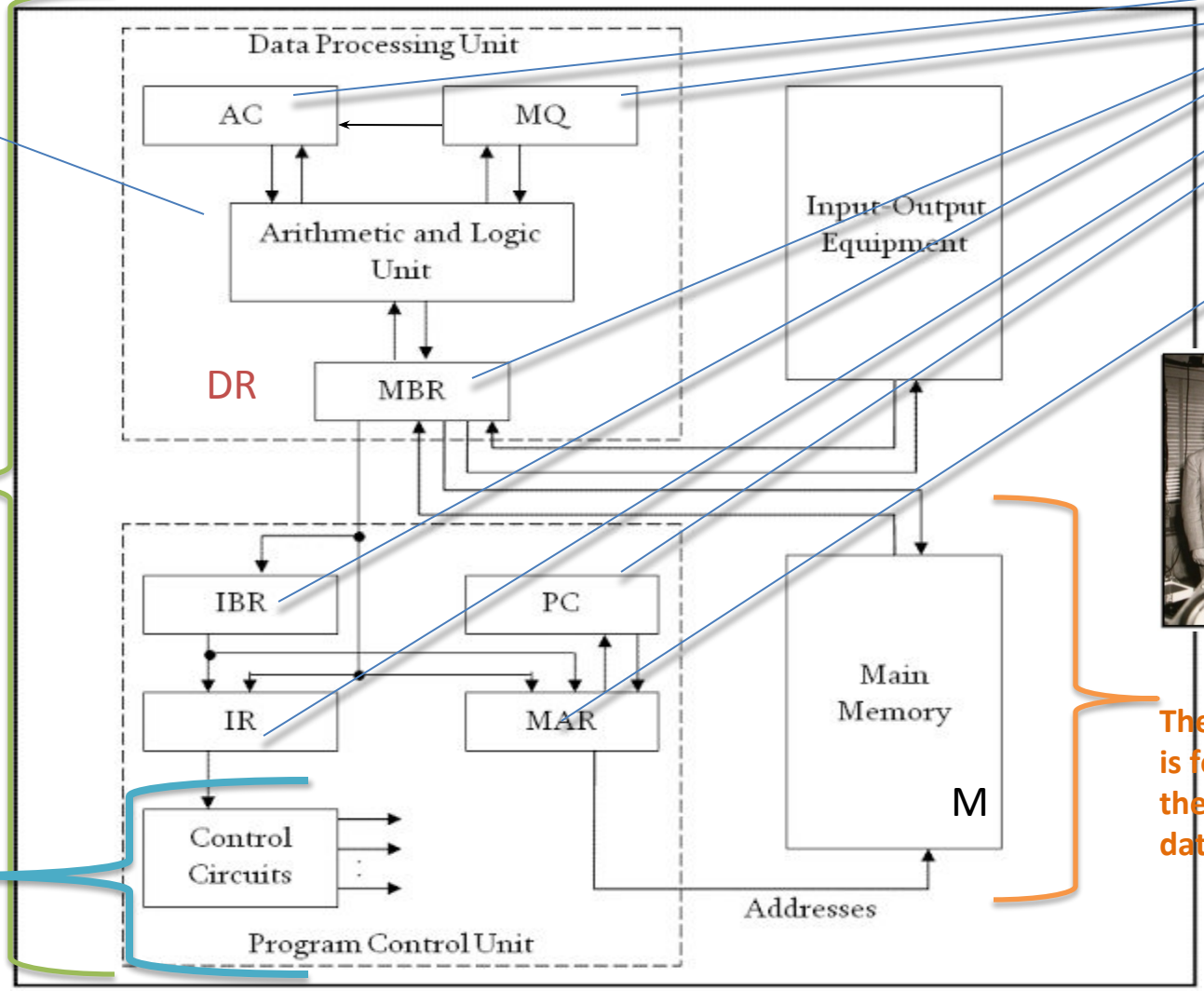
High speed register

Actions specified by the instructions

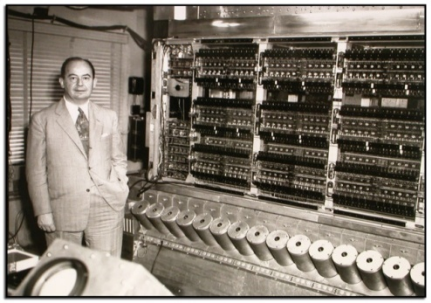
Consists of DPU and PCU

CPU

Responsible for 1.Fetching instructions, decoding of instructions, fetching data(operands) from the memory, 2. Providing necessary control signals



These High speed Registers are used to temporary storage of instructions/ data/ memory address.



The main memory is for the storing of the programs or data

Figure: Expanded structure of Von Neumann Architecture or IAS computer

Prepared by Suman Paul, Assistant Prof, Dept of ECE, Haldia Inst. of Tech., Haldia, WB 721657

- The Main memory is used for storing programs and data. A word transfer can take place between DR of the CPU and any location M(X) with the specific address X in M.
- The address X to be used is stored in MAR.
- The DR is used to store an operand during execution of instruction.
- Memory buffer register (MBR) >>Contains a word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.
- **AC Accumulator**
- **MQ Multiplier Quotient Register**

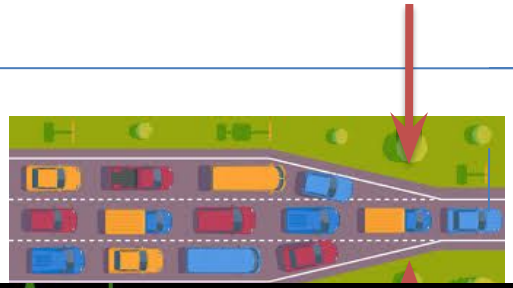
For the temporary storage of the operands and the results.

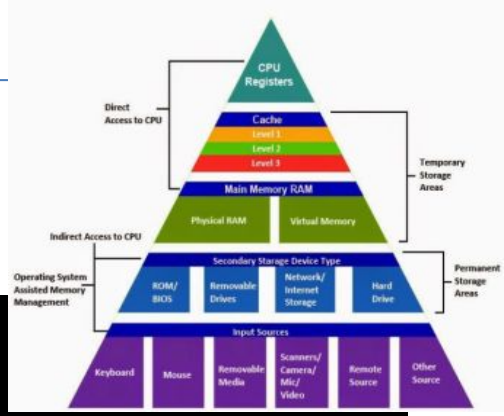
To hold temporarily operands and results of ALU operations. The most significant bits are stored in the AC and the least significant in the MQ.

- **2 instructions are fetched simultaneously from 'Main Memory'**
- **IBR Instruction Buffer Register >>The instruction that is not to be executed immediately is placed in an instruction buffer register.**
- **IR Instruction Register >> The op-code of the other instruction is placed in IR...where it is decoded.**
- **MAR Memory Address Register >>**Specifies the address field of the current instruction is transferred to the memory address register.

- **PC Program Counter >> Is used to store the address of the next instruction to be executed ..... run**

Von –Neumann Bottleneck



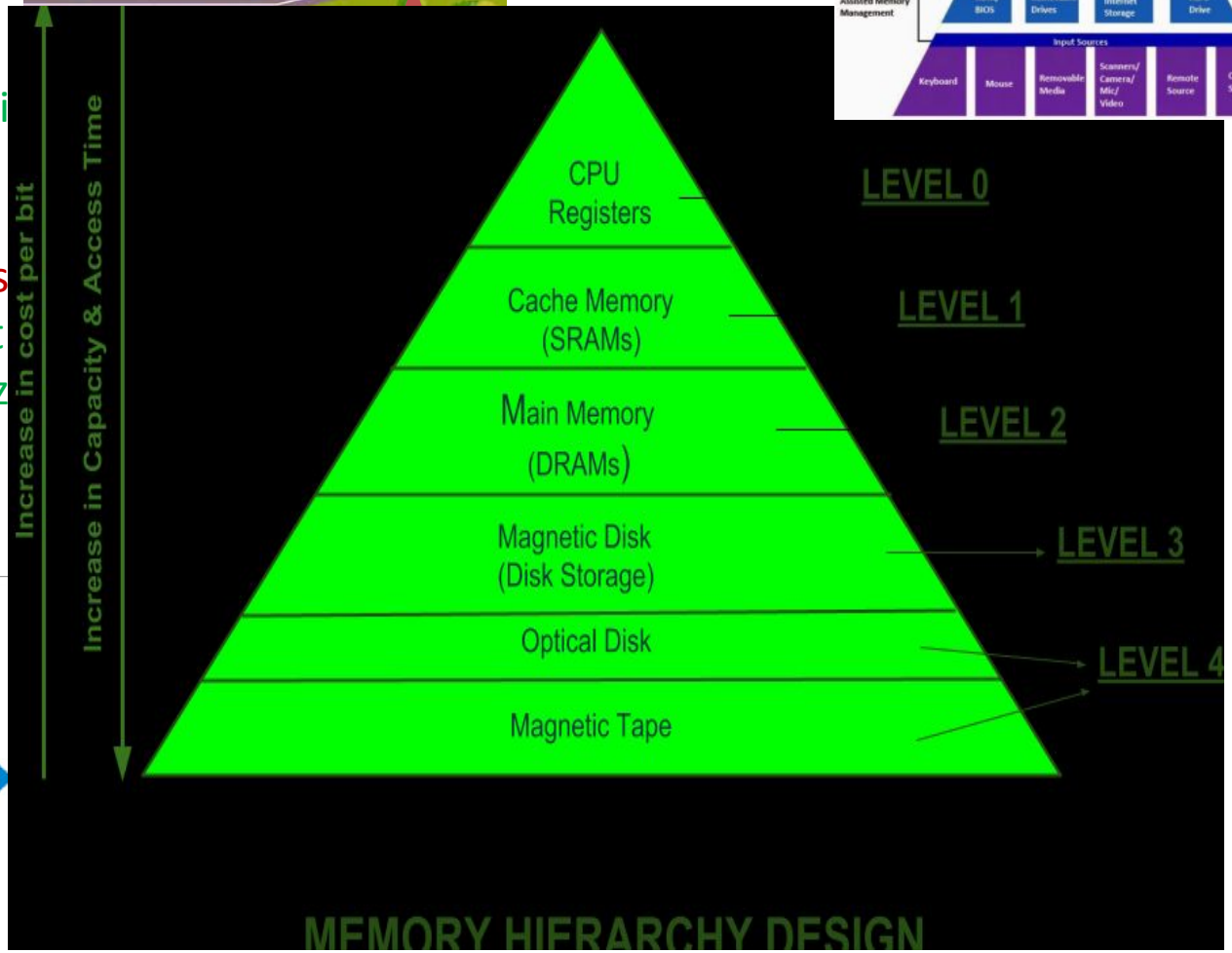


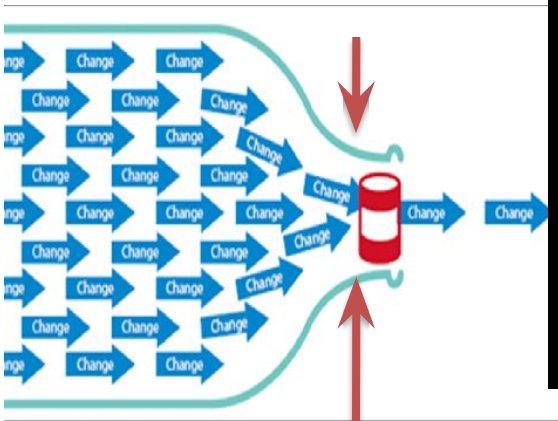
CPU – memory speed dispari

Von –Neumann Bottleneck

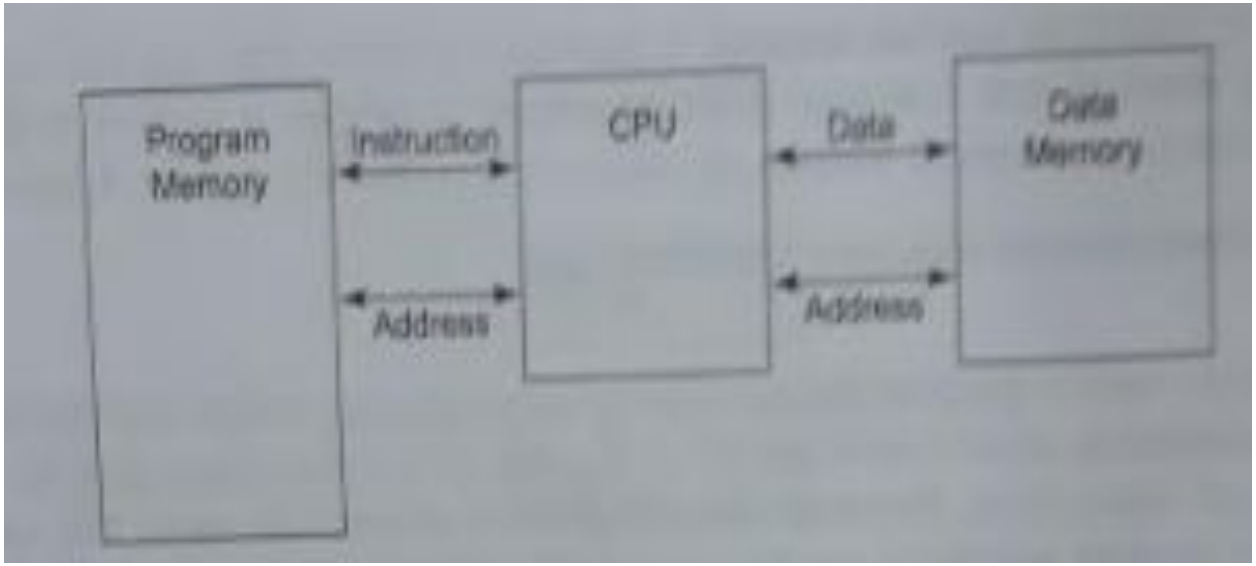
How to solve/ overcome this

1. Incorporate / implement
2. Use of RISC ...which utiliz





▪Harvard Architecture



Howard Aiken of Harvard University developed .....named as Automatic Controlled Calculator ....>>>Later Harvard Mark I

Two physically separate memories , one is for instructions and other is for Data..>>> Dedicated buses for each of them.

The instructions and the data can be fetched simultaneously. Both the memories can be accessed at the same time using separate buses. >> DSP,Microcontrollers

▪Von Neumann Vs. Harvard Architecture key differences ?? [1]

Von Neumann Vs. Harvard Architecture In Tabular Form

BASIS OF COMPARISON	VON NEUMANN ARCHITECTURE	HARVARD ARCHITECTURE
Description	The Von Neumann architecture is a theoretical design based on the stored-program computer concept.	The Harvard architecture is a modern computer architecture based on the Harvard Mark I relay-based computer model.
Memory System	Has only one bus that is used for both instructions fetches and data transfers.	Has separate memory space for instructions and data which physically separates signals and storage code and data memory.
Instruction Processing	The processing unit would require two clock cycles to complete an instruction.	The processing unit can complete an instruction in one cycle if appropriate pipelining plans have been set.

Available https://vivadifferences.com/5-major-difference-between-von-neumann-and-harvard-architecture/

▪Von Neumann Vs. Harvard Architecture key differences ?? [1] contd..

Use	Von Neumann architecture is usually used literally in all machines from desktop computers, notebooks, high performance computers to workstations.	Harvard architecture is a new concept used specifically in microcontrollers and digital signal processing (DSP).
Cost	Instructions and data use the same bus system therefore the design and development of control unit is simplified, hence the cost of production becomes minimum.	Complex kind of architecture because it employs two buses for instruction and data, a factor that makes development of the control unit comparatively more expensive.

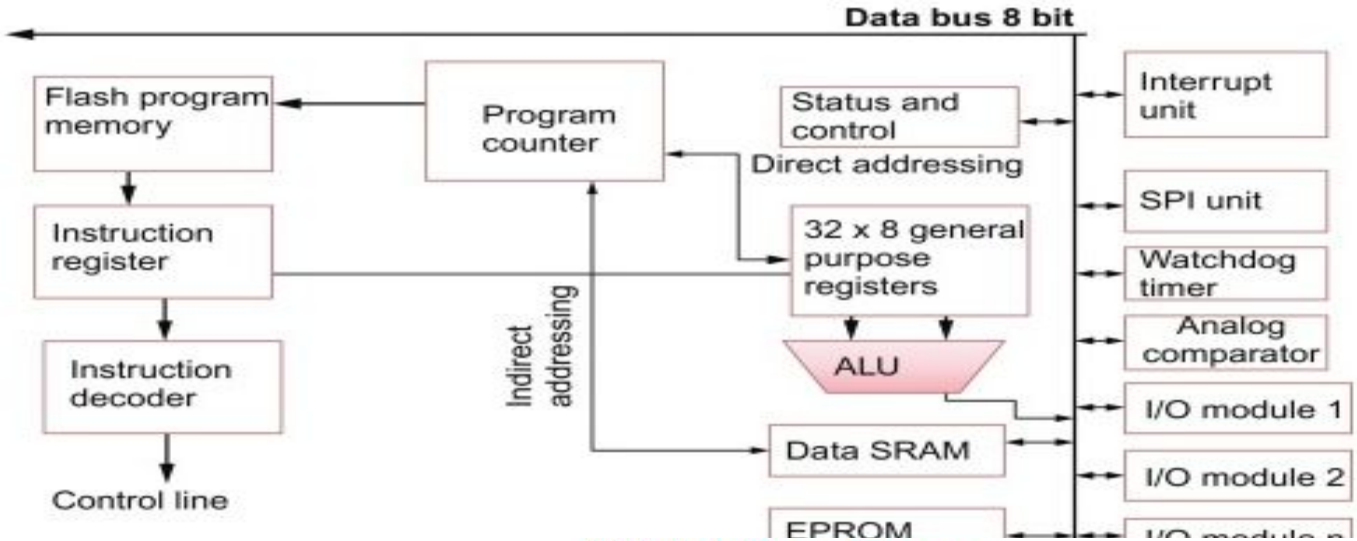
Duly Acknowledged (Available) https://vivadifferences.com/5-major-difference-between-von-neumann-and-harvard-architecture/

Arduino Architecture

Detailed illustrations:

https://www.sciencedirect.com/topics/engineering/harvard-architecture

Basically, the processor of the Arduino board is based on the Harvard architecture, where the program code and program data use separate memory. It consists of two separate memories, program memory and data memory. In this architecture, the data is stored in data memory whereas the code is stored in the flash program memory. The Atmega328 microcontroller has 32 kB of flash memory, 2 kB of SRAM, 1kB of EPROM, and operates with a 16-MHz clock speed.



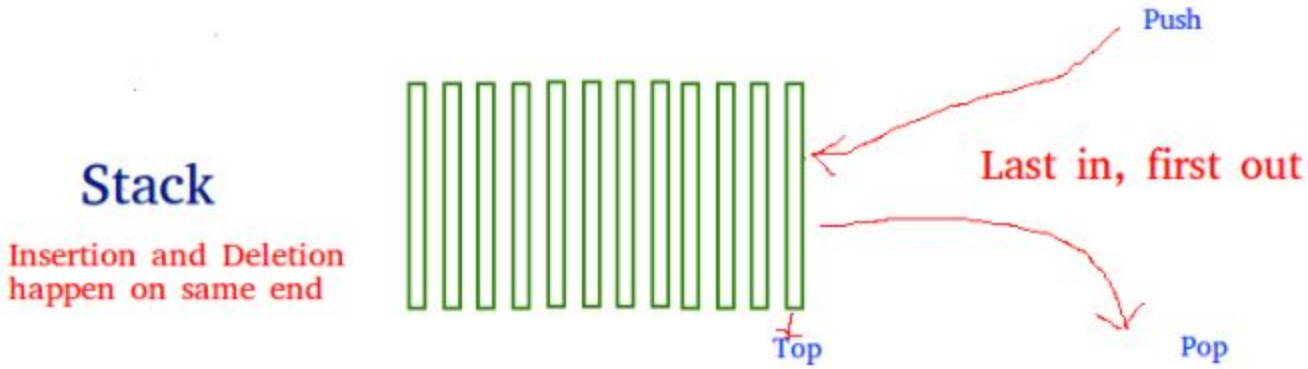
WatchDog Timer

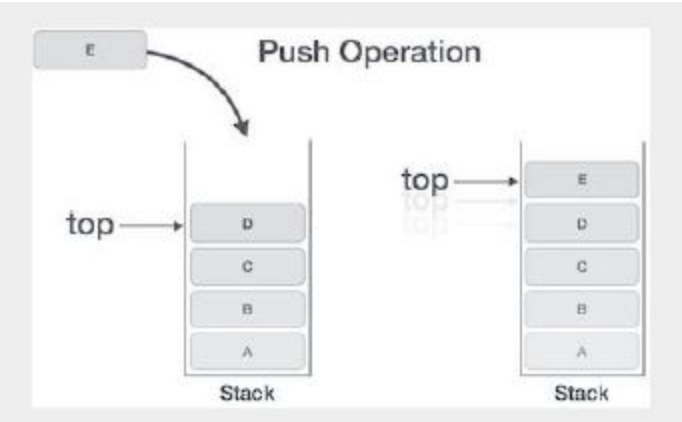
A watchdog timer (WDT) is a hardware timer that automatically generates a system reset if the main program neglects to periodically service it. It is often used to automatically reset an embedded device that hangs because of a software or hardware fault. Some systems may also refer to it as a computer operating properly (COP) timer. Many microcontrollers including the mbed processor have watchdog timer hardware.

Prepared by Suman Paul, Assist Prof, Dept of ECE, Haldia Inst. of Tech., Haldia, WB 721657

- **Stack**

- **A linear Data structure which follows a particular order in which the operations are performed....>>> LIFO or FILO**





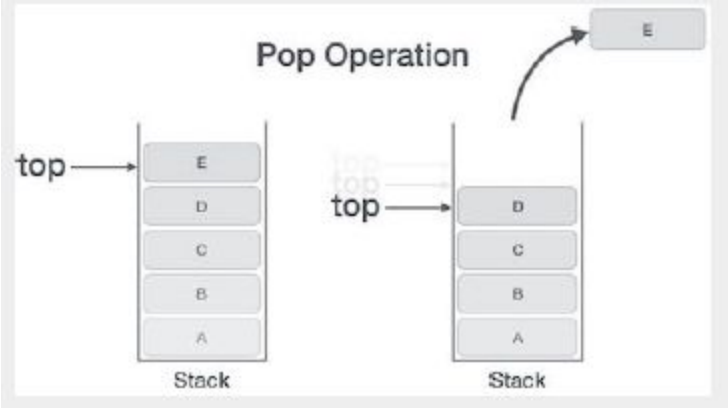
Push Operation

The process of putting a new data element onto stack is known as a Push Operation. Push operation involves a series of steps –

- **Step 1** – Checks if the stack is full.
- **Step 2** – If the stack is full, produces an error and exit.
- **Step 3** – If the stack is not full, increments **top** to point next empty space.
- **Step 4** – Adds data element to the stack location, where top is pointing.
- **Step 5** – Returns success.

- **Stack**

- **A linear Data structure which follows a particular order in which the operations are performed....>>> LIFO or FILO**



A Pop operation may involve the following steps –

- **Step 1** – Checks if the stack is empty.
- **Step 2** – If the stack is empty, produces an error and exit.
- **Step 3** – If the stack is not empty, accesses the data element at which **top** is pointing.
- **Step 4** – Decreases the value of top by 1.
- **Step 5** – Returns success.

▪QUEUE

Queue is an abstract data structure, One end is always used to insert data (Enqueue) and the other is used to remove data (Dequeue). Queue follows First-In-First-Out methodology.

▪QUEUE

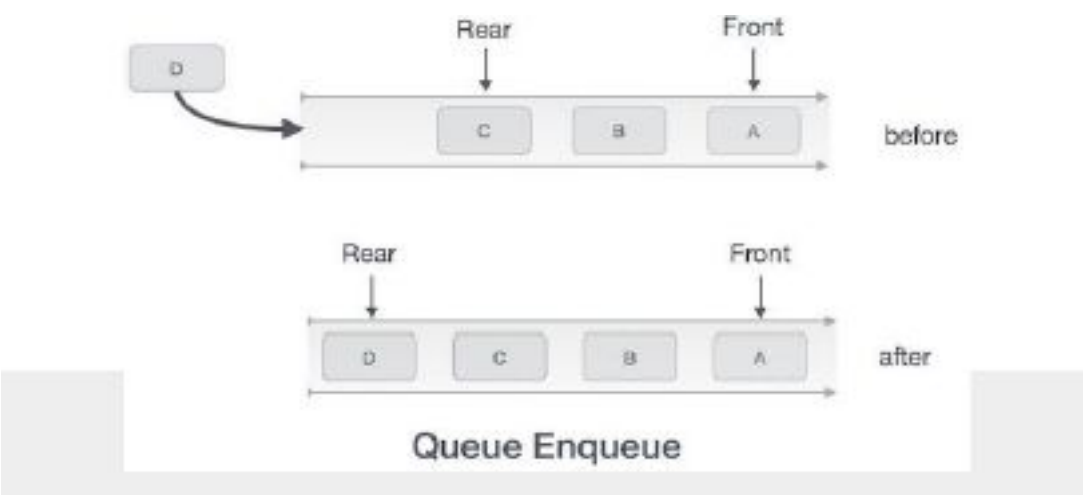
Queue is an abstract data structure, One end is always used to insert data (Enqueue) and the other is used to remove data (Dequeue). Queue follows First-In-First-Out methodology.

Enqueue Operation

Queues maintain two data pointers, **front** and **rear**. Therefore, its operations are comparatively difficult to implement than that of stacks.

The following steps should be taken to enqueue (insert) data into a queue −

- **Step 1** − Check if the queue is full.
- **Step 2** − If the queue is full, produce overflow error and exit.
- **Step 3** − If the queue is not full, increment **rear** pointer to point the next empty space.
- **Step 4** − Add data element to the queue location, where the rear is pointing.
- **Step 5** − return success.



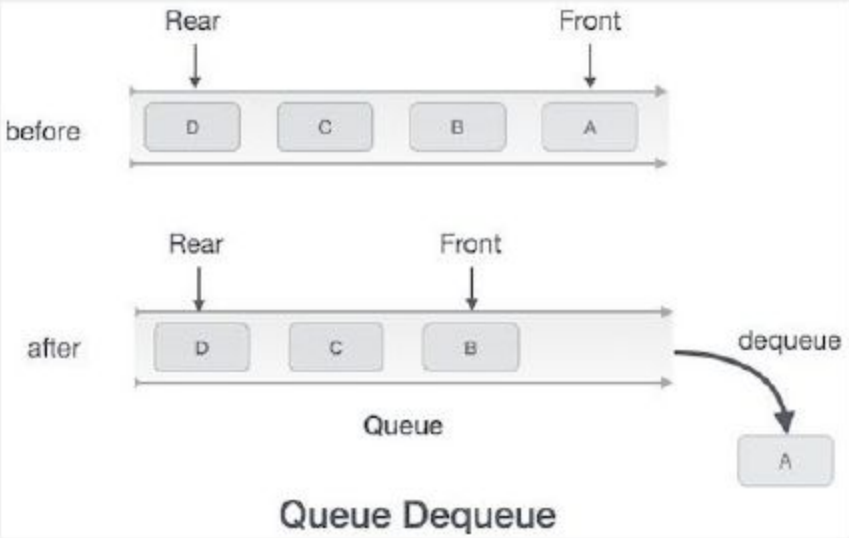
▪QUEUE

Queue is an abstract data structure, One end is always used to insert data (Enqueue) and the other is used to remove data (Dequeue). Queue follows First-In-First-Out methodology.

Dequeue Operation

Accessing data from the queue is a process of two tasks – access the data where **front** is pointing and remove the data after access. The following steps are taken to perform **dequeue** operation –

- **Step 1** – Check if the queue is empty.
- **Step 2** – If the queue is empty, produce underflow error and exit.
- **Step 3** – If the queue is not empty, access the data where **front** is pointing.
- **Step 4** – Increment **front** pointer to point to the next available data element.
- **Step 5** – Return success.



Queue Dequeue

- **Subroutine**

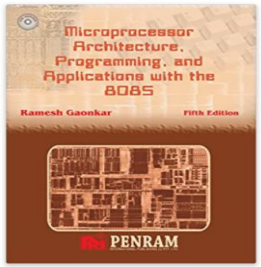
A **subroutine** (also called a **subprogram**) is an abstraction of a process that is **called**.

- Subroutines are blocks of code that may be repeatedly called by the main program to serve a given function.

Book of: Microprocessor architecture, Programming , and Applications with the 8085 by Ramesh Gaonkar, Penram Publisher

- **Program**—a set of instructions written in a specific sequence for the computer to accomplish a given task.
- **Machine language**—the binary medium of communication with a computer through a designed set of instructions specific to each computer.
- **Assembly language**—a medium of communication with a computer in which programs are written in mnemonics. An assembly language is specific to a given computer.
- **Low-level language**—a medium of communication that is machine-dependent or specific to a given computer. The machine and the assembly languages of a computer are considered low-level languages. Programs written in these languages are not transferable to different types of machines.
- **High-level language**—a medium of communication that is independent of a given computer. Programs are written in English-like words, and they can be executed on a machine using a translator (a compiler or an interpreter).
- **Source code**—a program written either in mnemonics of an assembly language or in English-like statements of a high-level language (before it is assembled or compiled).
- **Compiler**—a program that translates English-like words of a high-level language into the machine language of a computer. A compiler reads a given program, called a source code, in its entirety and then translates the program into the machine language, which is called an object code.
- **Interpreter**—a program that translates the English-like statements of a high-level language into the machine language of a computer. An interpreter translates one statement at a time from a source code to an object code.
- **Assembler**—a computer program that translates an assembly language program from mnemonics to the binary machine code of a computer.

Suggested Book



OPCODES AND OPERANDS

An opcode is short for 'Operation Code'.

An opcode is a single instruction that can be executed by the CPU. In machine language it is a binary or hexadecimal value such as 'B6' loaded into the instruction register.

In assembly language mnemonic form an opcode is a command such as MOV or ADD or JMP.

For example

```
MOV AL, 34h
```

The opcode is the MOV instruction. The other parts are called the 'operands'.

Operands are manipulated by the opcode. In this example, the operands are the register named AL and the value 34 hex.

Model Questions/ assignment 1 :

What is Computer organization and architecture ?
With detailed illustrations discuss Von Neumann architecture.
What is Von Neumann bottleneck ? How to solve this problem ?

State the working principle of Harvard architecture.

Write short notes on: (i)Multiprogramming OS, (ii) RTOS

Thank you!!