Processor Organization and Architecture by Prof. Ramchandra Mangrulkar

* Prerequisites include Digital Electronics and Operating Systems
* Registers are flip flop memory for the processors.
* After decoding the instructions, the processor has to find the variable values from diff registers. Has to perform operations and send the output.
* This instruction/processor cycle will be studied in this subject.
* First Course Objective is to go the basic execution structure. Read and Write operations.
* L1 (Fastest) -> L2 -> L3 (Slowest) caches. (Smallest to Biggest)
* L1 is inside the processor
* Virtual Memory is basically the empty space in secondary memory.
* Therefore, if the storage is full, the devices run a bit slower
* Von Neumann model is the first architectural model designed
* In last Unit we will learn assembly language
* **Williams Stallings book**
* **J. P Hayes book**
* Patterson Book is used to evaluate the efficiency of the systems
* All experiments are based on assembly language coding
* Went through initial parts of the unit. Read the whole syllabus and course outcomes. Little discussion about current trends and technologies.

PROCESSOR ARCHITECTURE

* Graphical user interface, text, application, email

  Description automatically generated
* SIMPLE ADDER
* Text

  Description automatically generated
* Differentiate between Organization and Architecture
* **ISA: Instruction Set Architecture**
* Computer Architecture
  + It deals with functional behavior of computer system
  + It’s the design and implementation of various part of computer system
  + Divided into 2 parts
    - ISA
    - Hardware System Architecture
* Computer Organization
  + It deals with structural relationships.
  + How many buses need to be used, etc. type of questions.
  + Operational Attributes. How the gates and adders will be involved

**CPU Consists of :**

* + **Registers**
  + **ALU**
  + **Timing and control signals**
  + **Interference**
* **Flynn’s Classification (**How data is classified and retrieved by comps)
  + **SISD (Single Instruction Single data) { c = a + b } [ Von Neuman Model]**
  + **SIMD (Single Instruction Multiple Data) { c[x] = A[x] + B[x] }**
  + **MISD (Multiple Instruction Single Data)**
  + **MIMD (Multi Inst and Multi Data) Parallel programming/ Multithreading**



* **X = 2 + 3**
  + **ISA :-**
    - **MOV R1, 02H (Store 2)**
    - **MOV R2, 03H (Store 3)**
    - **ADD R1, R2 (Add R2 to R1)**
    - **STORE X, R1 (Store R1 value in X)**
  + **HSA (Hardware System Architecture):-**

**VON NEUMAN’s MODEL: (Also Princeton Architecture)**

**Diagram

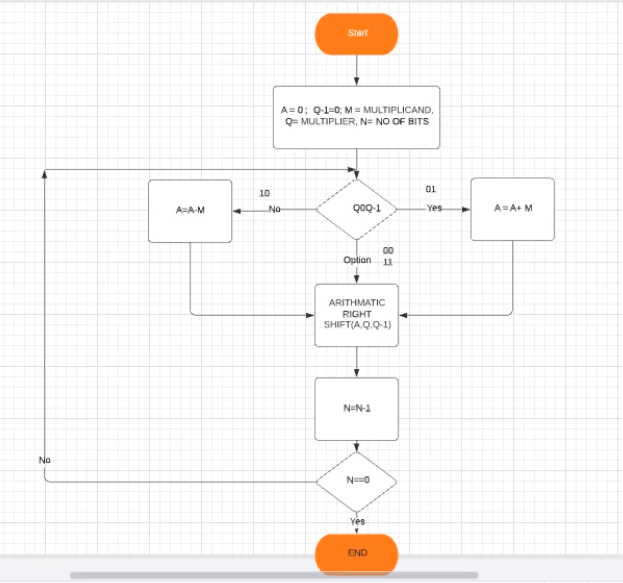
Description automatically generated**

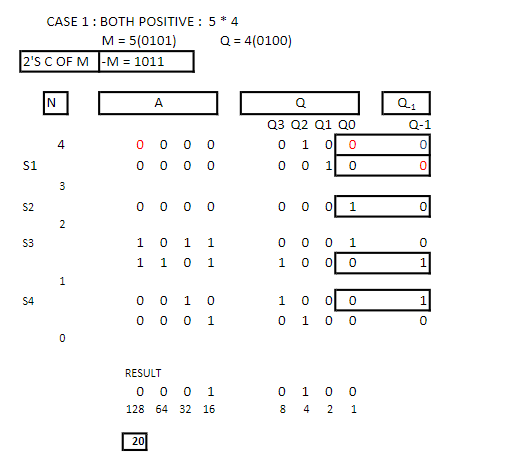
**DRAWBACKS:  
- When we fetch instructions, at a time we can fetch single data only**

**SOln:**

* + **Harwards Architecture (Fetches data in parallel)**
    - **Processor is connected to I/O Peripherals, data Memory and Inst Memory in Parallel**
  + **Modified Harward Architecture (Cache Memory)**
    - **Same as Harwards but the processor has a cache memory where it searches before searching in the primary and secondary memory.**
    - **TRADEOFF’s:-**
      * **Access Time (Cache Mem < Primary Mem < Sec Mem)**
      * **Cost (Cache Mem > Primary Mem > Sec Mem)**
* **A 2 GHz Processor takes 0.5 ns to access data in cache memory**

**BOOTHS ALGORITHM**

****

****