**Chapter 3: ARCHITECTURE OF 8085**

**Topic – 1: Architectural Information**

**Introduction**

* Microprocessors are even used in **washing machines** & **microwave oven**.
* **8085** came after **8080**, which was also an **8-bit** microprocessor.

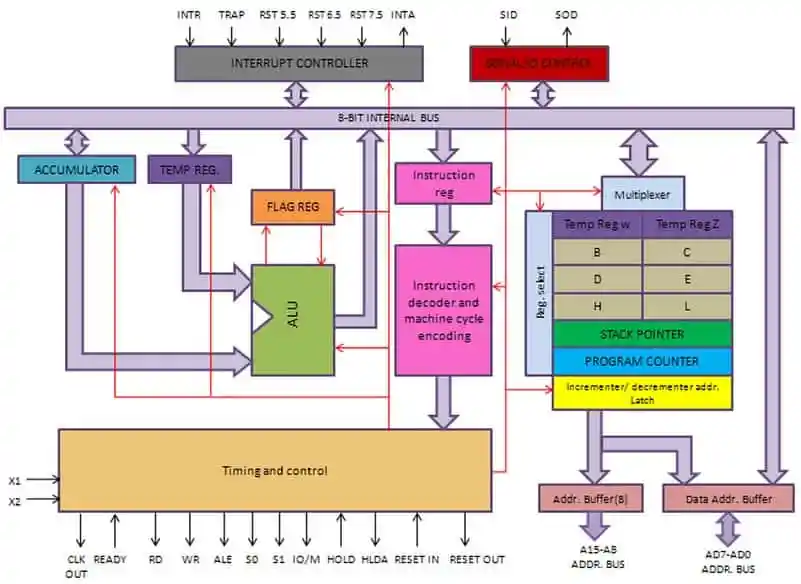
**Processor Architecture**

* **Processor architecture** is the design of internal components of a microprocessor.
* It varies across various **families** of microprocessors.
* One has to learn about the **architecture** of microprocessor in order to learn the **assembly language**.

**8085 Specific Information**

* **8-bit** **data bus** & **16-bit** **address bus**, the **16-bit address bus** adding up to **64 KB**.
* Also has a **16-bit** **program counter** (**PC**) & **16-bit stack pointer** (**SP**).
* There are **six** **8-bit** registers in pair **BC**, **DE**, **HL**.
* Operates at **3.2 MHz** single-phase clock frequency.
* **Single phase clock frequency:** Clock signal pulsates with **one type** of phase.
* Based on "Von-Neuman architecture".
* **Von-Neuman architecture:** Data and instructions are stored in same memory space.

**Topic – 2: Architecture Diagram**



**Topic – 3: Components Description**

**Accumulator**

* Shortly referred to as **'Acc'**.
* Connected to **internal bus** & **ALU**.

**ALU**

* Also does **left shift** & **right shift**.
* Picks data from **memory** & stores the result in **accumulator**.

**Data Bus & Address Bus**

* **Data bus** carries data & is **bidirectional**.
* **Address bus** carries those data to the required location & is **unidirectional**.

**Instuction Register (IR) & Decoder**

* **8-bit** register linked to **instruction cycle**.
* When an instruction is **fetched**, it is loaded to the **IR**.
* **Decoder** decodes that instruction.
* **IR** is **non-programmable** and thus, inaccessible through code.

**General-Purpose Register**

* **B, C, D, E, H, L**
* Can also be used in pairs.

**Program Counter (PC)**

* **16-bit** register.
* Stores address of **next instruction** to be executed.

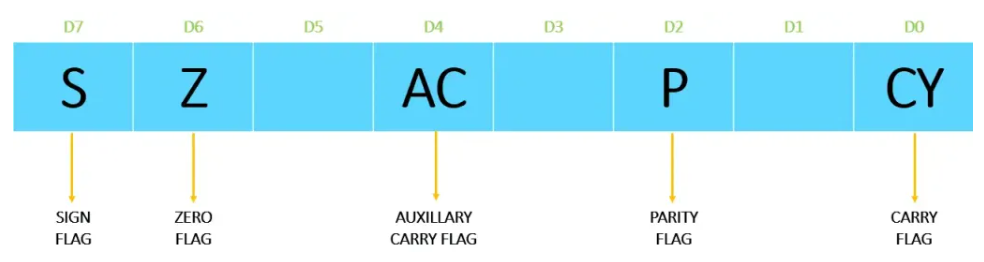
**Stack Pointer (SP)**

* **16-bit** register & similar to **stack**.
* **Incremented/ decremented** by **two** during a **push/ pop** operation.

**Internal Registers**

* **W, Z**
* Also known as **temporary internal register**.
* **8-bit** registers & holds **data** for some **special** arithmetic & logic operations.
* Also holds data for **CALL** and **XCHG** instructions.

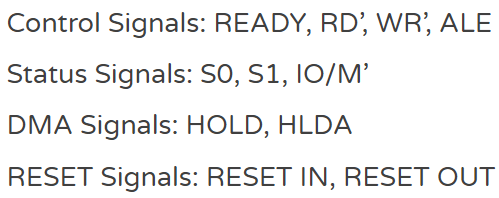
**Flag Register**



* **D1, D3, D5** are empty slots.
* These slots are **reserved** for future updates in microprocessor.
* These **flags** are **flip-flops** that tell status of the operation performed.
* **Zero flag (Z):** Tells if the operation performed by CPU results in **0** or **not**.
* The flags’ values **revert** after the result has been **stored** in the **accumulator**.
* **Auxiliary carry flag (AC):** Tells the carry on the **4th bit** of resulting value.
* **Parity flag (P):** Tells the **parity** of a number, where **even parity** means that it contains even number of **1s**. **P** is set to **0** for **odd parity** & **1** for **even**.
* **Carry Flag (CY):** Used to indicate if there is any **carry out** generated from **MSB**.
* Means it indicates if any **carry** would be generated after the **leftmost digit**.

**Timing & Control Unit**

* Produces the **timing** & **control signals** all across the microprocessor.
* Synchronizes **internal clock** with all microprocessor operations.
* Also establishes connection between microprocessor & peripherals.



**Interrupt Controls**

* This part controls the **interrupt** popping up during a process.
* When an **interrupt** pops up, the control is handed from **main program** to the **interrupt request**.
* This interrupt can be a hardware or software.
* Interrupts can be **maskable** or **non-maskable**.

**Increment/ Decrement Register**

* Memory position can be **increased** or **decreased** by one.
* Used for **manipulating** what **PC** and **SP** point to.

**Address Buffer & Address Data Buffer**

* The contents stored in **SP** are loaded to **address buffer**.
* And contents stored in **PC** are loaded to **address data buffer**.
* This is done to establish a communication between **SP** and **PC** & the **microprocessor**.
* **Memory** & **I/O chips** are connected to these buffers.
* So, CPU can exchange data with **memory** & **I/O chips** if required.