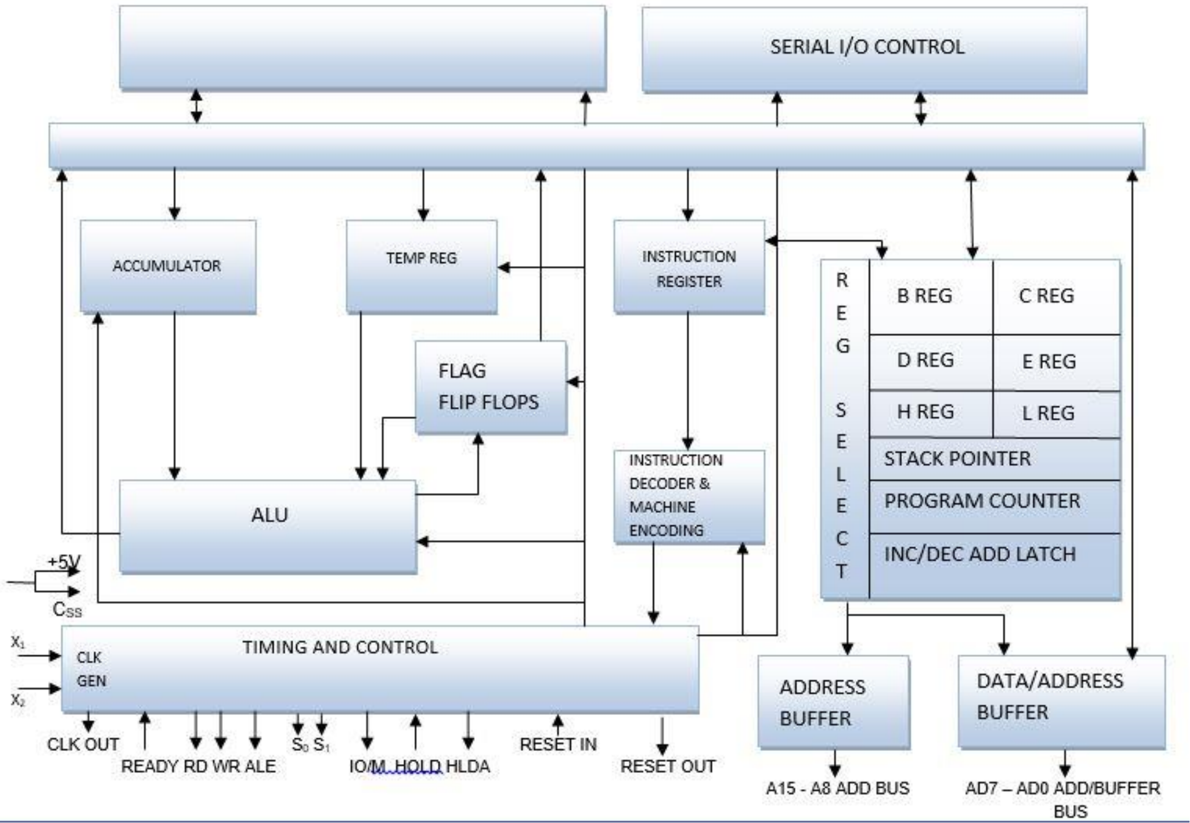
Microprocessor (All points are important)

Main features of 8085 microprocessor:

1. It is an 8 bit microprocessor
2. It operates on a single +5V power supply
3. It has 16 address buses. Hence it can access 2^{16}=64Kbytes of memory.
4. It provides 8 bit I/O address to access (2^8) 256 I/O ports.
5. It operates on clock cycle with 50% duty cycle.
6. It can operate with 3.2 MHz clock frequency.

Block diagram of 8085 (Important)



Registers in 8085 microprocessor (Important)

It has eight addressable 8-bit registers: **A, B, C, D, E, H, L, F,** and two 16-bit registers PC and SP. These registers can be classified as −

- General Purpose Registers
- Temporary Registers: a) **Temporary data register b) W and Z registers**
- Special Purpose Registers: a) **Accumulator b) Flag registers c) Instruction register**
- Sixteen-bit Registers: a) **ProgramCounter (PC) b) Stack Pointer (SP)**

System Buses: (Important)

The system bus in the microprocessor is the pathway that consists of cables or wires to transfer data between the microprocessor, internal sub-components, and Input/output devices. It helps to connect different sub-components of the computer with each other.

Types of System Bus

To do an efficient communication its consists of three different buses such as

1. Data bus,
2. Address bus,
3. Control bus.

Address Bus

- The address bus is used to specify the memory locations or I/O device address.
- The address bus is unidirectional because data travel from microprocessor to memory or I/O device.
- It consists of 16-bit range from 0000H to FFFFH.

Data Bus

- The data bus is used to transfer data between microprocessor to I/O devices/memory or vice versa.
- The data bus is bidirectional because data can travel from microprocessor to memory or I/O device or vice versa.
- It consists of an 8-bit range from 00H to FFH.

Control Bus

The control bus is used to initiate timing or coordination signals to complete various instructions across the system. Some of the examples of control signals are given below:

- Memory Read Signal
- Memory Write Signal
- Opcode Fetch
- I/O Read
- I/O Write
- Read
- Write
- Address Latch Enable

Arithmetic Logic Unit (ALU): (Important)

Inside a computer, there is an Arithmetic Logic Unit (ALU), which is capable of performing logical operations (e.g. AND, OR, Ex-OR, Invert etc.) in addition to the arithmetic operations (e.g. Addition, Subtraction etc.). The control unit supplies the data required by the ALU from memory, or from input devices, and directs the ALU to perform a specific operation based on the instruction fetched from the memory. ALU is the "calculator" portion of the computer.

An arithmetic logic unit(ALU) is a major component of the central processing unit of the a computer system. It does all processes related to arithmetic and logic operations that need to be done on instruction words. In some microprocessor architectures, the ALU is divided into the arithmetic unit (AU) and the logic unit (LU).

Stack memory (Important):

The stack is a LIFO (last in, first out) data structure implemented in the RAM area and is used to store addresses and data when the microprocessor branches to a subroutine. Then the return address used to get pushed on this stack. Also to swap values of two registers and register pairs we use the stack as well.

Instrcution cycle (Important):

The Program and data which are stored in the memory, are used externally to the microprocessor for executing the complete instruction cycle. Thus to execute a complete instruction of the program, the following steps should be performed by the 8085 microprocessor.

- Fetching the opcode from the memory;
- Decoding the opcode to identify the specific set of instructions;
- Fetching the remaining Bytes left for the instruction, if the instruction length is of 2 Bytes or 3 Bytes;
- Executing the complete instruction procedure.

Opcode and Operand

An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: one is task to be performed, called the **operation code (opcode),** and the second is the data to be operated on, called the **operand**. The operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit) data, an internal register, a memory location, or 8-bit (or 16-bit) address. In some instructions, the operand is implicit.

1 One-Byte Instructions

A 1-byte instruction includes the opcode and operand in the same byte. Operand(s) are internal register and are coded into the instruction

Table 2.1 Example for 1 byte Instruction

Task	Op code	Operand	Binary Code	Hex Code
Copy the contents of the accumulator in the register C.	MOV	C,A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator.	ADD	B	1000 0000	80H
Invert (compliment) each bit in the accumulator.	CMA		0010 1111	2FH

2 Two-Byte Instructions

In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. Source operand is a data byte immediately following the opcode. For example:

Table 2.2 Example for 2 byte Instruction

Table 2.2 Example for 2 byte Instruction

Task	Opcode	Operand	Binary Code	Hex Code	
Load an 8-bit data byte in the accumulator.	MVI	A, Data	0011 1110 DATA	3E Data	First Byte Second Byte