

CPU ARCHITECTURE

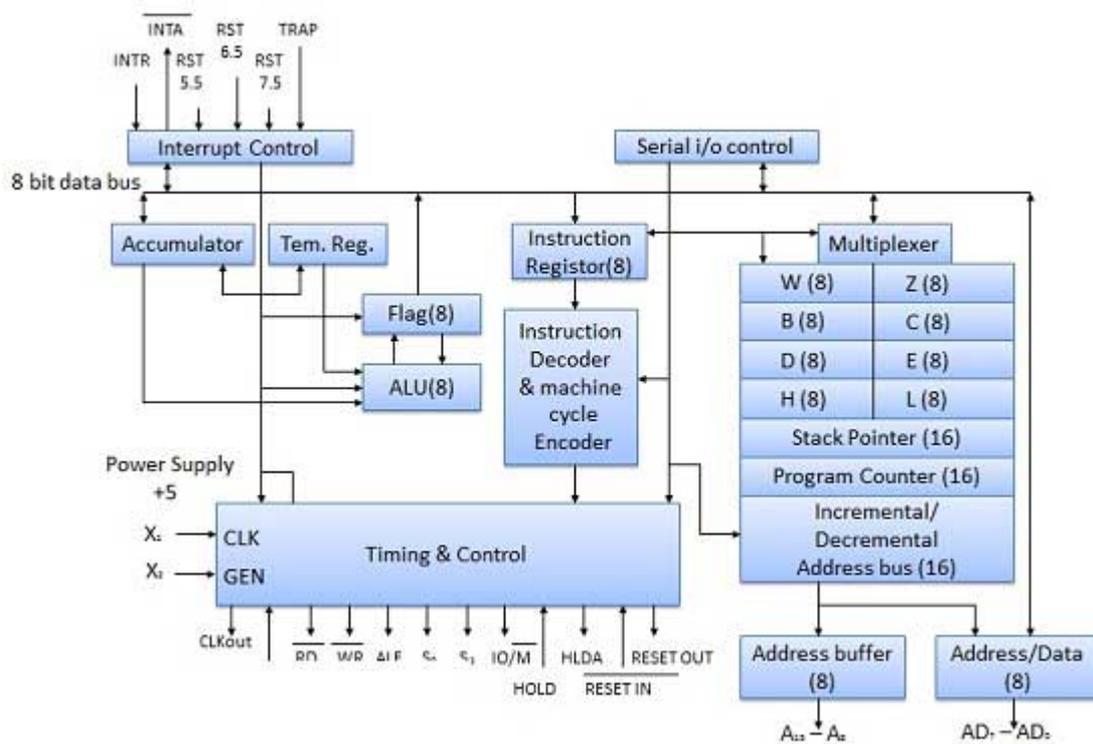
Microprocessing unit is synonymous to central processing unit, CPU used in traditional computer. Microprocessor MPU acts as a device or a group of devices which do the following tasks.

- communicate with peripherals devices
- provide timing signal
- direct data flow
- perform computer tasks as specified by the instructions in memory

8085 Microprocessor

The 8085 microprocessor is an 8-bit general purpose microprocessor which is capable to address 64k of memory. This processor has forty pins, requires +5 V single power supply and a 3-MHz single-phase clock.

Block Diagram



ALU

The ALU perform the computing function of microprocessor. It includes the accumulator, temporary register, arithmetic & logic circuit & and five flags. Result is stored in accumulator & flags.

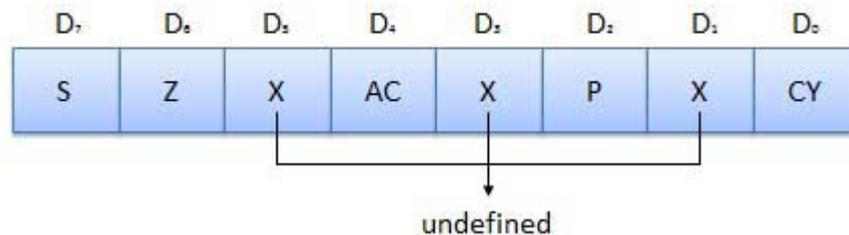
Block Diagram



Accumulator

It is an 8-bit register that is part of ALU. This register is used to store 8-bit data & in performing arithmetic & logic operation. The result of operation is stored in accumulator.

Diagram



Flags

Flags are programmable. They can be used to store and transfer the data from the registers by using instruction. The ALU includes five flip-flops that are set and reset according to data condition in accumulator and other registers.

- **S Sign flag** – After the execution of an arithmetic operation, if bit D₇ of the result is 1, the sign flag is set. It is used to signed number. In a given byte, if D₇ is 1 means negative number. If it is zero means it is a positive number.
- **Z Zero flag** – The zero flag is set if ALU operation result is 0.
- **AC Auxiliary Carry flag** – In arithmetic operation, when carry is generated by digit D₃ and passed on to digit D₄, the AC flag is set. This flag is used only internally BCD operation.
- **P Parity flag** – After arithmetic or logic operation, if result has even number of 1s, the flag is set. If it has odd number of 1s, flag is reset.
- **C Carry flag** – If arithmetic operation result is in a carry, the carry flag is set, otherwise it is reset.

Register section

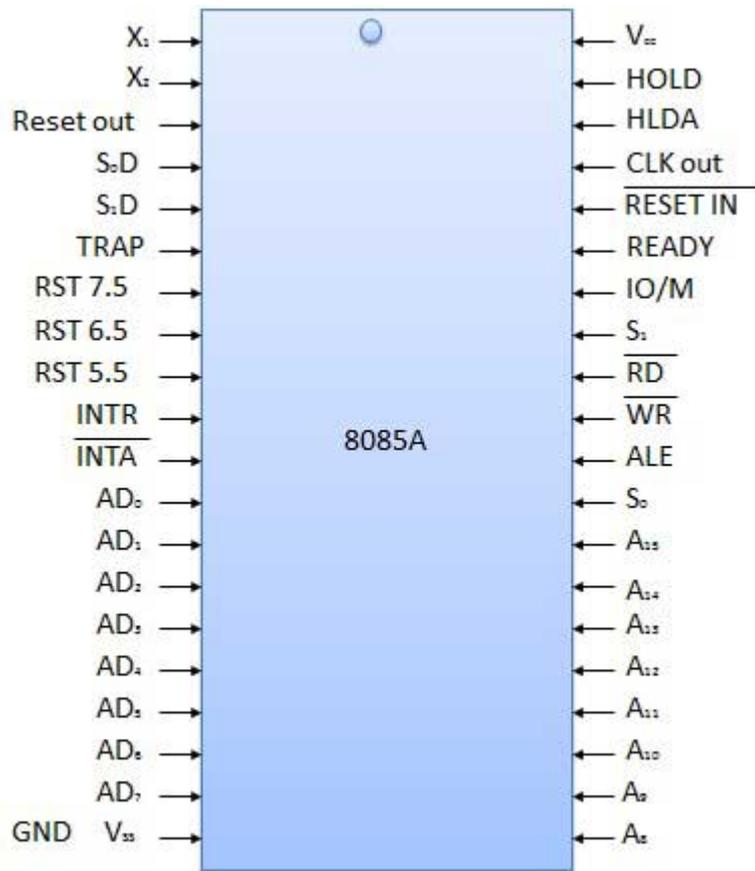
It is basically a storage device and transfers data from registers by using instructions.

- **Stack Pointer SP** – The stack pointer is also a 16-bit register which is used as a memory pointer. It points to a memory location in Read/Write memory known as stack. In between execution of program, sometime data to be stored in stack. The beginning of the stack is defined by loading a 16-bit address in the stack pointer.
- **Program Counter PC** – This 16-bit register deals with fourth operation to sequence the execution of instruction. This register is also a memory pointer. Memory location have 16-bit address. It is used to store the execution address. The function of the program counter is to point to memory address from which next byte is to be fetched.
- **Storage registers** – These registers store 8-bit data during a program execution. These registers are identified as B, C, D, E, H, L. They can be combined as register pair BC, DE and HL to perform some 16 bit operations.

Time and Control Section

This unit is responsible to synchronize Microprocessor operation as per the clock pulse and to generate the control signals which are necessary for smooth communication between Microprocessor and peripherals devices. The RD bar and WR bar signals are synchronous pulses which indicates whether data is available on the data bus or not. The control unit is responsible to control the flow of data between microprocessor, memory and peripheral devices.

PIN diagram



All the signal can be classified into six groups

S.N.	Group	Description
1	Address bus	The 8085 microprocessor has 8 signal line, $A_{15} - A_8$ which are uni directional and used as a high order address bus.
2	Data bus	The signal line $AD_7 - AD_0$ are bi-directional for dual purpose. They are used as low order address bus as well as data bus.
3	Control signal and Status signal	<p>Control Signal</p> <p>RD bar – It is a read control signal <i>activelow</i>. If it is active then memory read the data.</p> <p>WR bar – It is write control signal <i>activelow</i>. It is active when written into selected memory.</p> <p>Status signal</p> <p>ALU AddressLatchEnable – When ALU is high. 8085 microprocessor use address bus. When ALU is low. 8085 microprocessor is use data bus.</p> <p>IO/M bar – This is a status signal used to differentiate between i/o and memory operations. When it is high, it indicate an i/o operation and when it is low, it indicate</p>

memory operation.

S₁ and S₀ – These status signals, similar to i/o and memory bar, can identify various operations, but they are rarely used in small system.

V_{CC} – +5v power supply.

V_{SS} – ground reference.

4 Power supply and frequency signal

X, X – A crystal is connected at these two pins. The frequency is internally divided by two operate system at 3-MHz, the crystal should have a frequency of 6-MHz.

CLK out – This signal can be used as the system clock for other devices.

INTR i/p – Interrupt request.

INTA bar o/p – It is used as acknowledge interrupt.

TRAP i/p – This is non maskable interrupt and has highest priority.

HOLD i/p – It is used to hold the executing program.

HLDA o/p – Hold acknowledge.

5 Externally initiated signal

READY i/p – This signal is used to delay the microprocessor read or write cycle until a slow responding peripheral is ready to accept or send data.

RESET IN bar – When the signal on this pin goes low, the program counter is set to zero, the bus are tri-stated, & MPU is reset.

RESET OUT – This signal indicate that MPU is being reset. The signal can be used to reset other devices.

RST 7.5, RST 6.5, RST 5.5 Request interrupt – It is used to transfer the program control to specific memory location. They have higher priority than INTR interrupt.

6 Serial I/O ports

The 8085 microprocessor has two signals to implement the serial transmission serial input data and serial output data.

Instruction Format

Each instruction is represented by a sequence of bits within the computer. The instruction is divided into group of bits called field. The way instruction is expressed is known as instruction format. It is usually represented in the form of rectangular box. The instruction format may be of

the following types.

Variable Instruction Formats

These are the instruction formats in which the instruction length varies on the basis of opcode & address specifiers. For Example, VAX instruction vary between 1 and 53 bytes while X86 instruction vary between 1 and 17 bytes.

Format



Advantage

These formats have good code density.

Drawback

These instruction formats are very difficult to decode and pipeline.

Fixed Instruction Formats

In this type of instruction format, all instructions are of same size. For Example, MIPS, Power PC, Alpha, ARM.

Format



Advantage

They are easy to decode & pipeline.

Drawback

They don't have good code density.

Hybrid Instruction Formats

In this type of instruction formats, we have multiple format length specified by opcode. For example, IBM 360/70, MIPS 16, Thumb.

Format



Advantage

These compromise between code density & instruction of these type are very easy to decode.

Addressing Modes

Addressing mode provides different ways for accessing an address to given data to a processor. Operated data is stored in the memory location, each instruction required certain data on which it has to operate. There are various techniques to specify address of data. These techniques are called Addressing Modes.

- **Direct addressing mode** – In the direct addressing mode, address of the operand is given in the instruction and data is available in the memory location which is provided in instruction. We will move this data in desired location.
- **Indirect addressing mode** – In the indirect addressing mode, the instruction specifies a register which contain the address of the operand. Both internal RAM and external RAM can be accessed via indirect addressing mode.
- **Immediate addressing mode** – In the immediate addressing mode, direct data is given in the operand which move the data in accumulator. It is very fast.
- **Relative addressing mode** – In the relative address mode, the effective address is determined by the index mode by using the program counter in stead of general purpose processor register. This mode is called relative address mode.
- **Index addressing mode** – In the index address mode, the effective address of the operand is generated by adding a content value to the contents of the register. This mode is called index address mode.

Loading [MathJax]/jax/output/HTML-CSS/jax.js