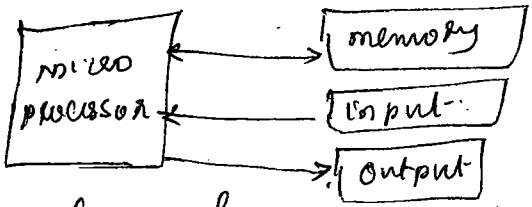
MICROPROCESSORS Module 1

Module I

Definition

is a multipurpose, programmable, clock driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions, and provides results as output.

→ A typical programmable machine can be represented with 4 components: processor, memory, input and output.



The physical components → hardware

set of instructions written for microprocessor to perform a task → program

group of programs – software

→ The microprocessor applications are classified into 2 categories.

* Reprogrammable systems → microprocessor is used for computing & data processing (eg: PC microcomputer)
* embedded systems → microprocessor is a part of final product and is not available for reprogramming to the end user (eg: photostat machine, microwave)

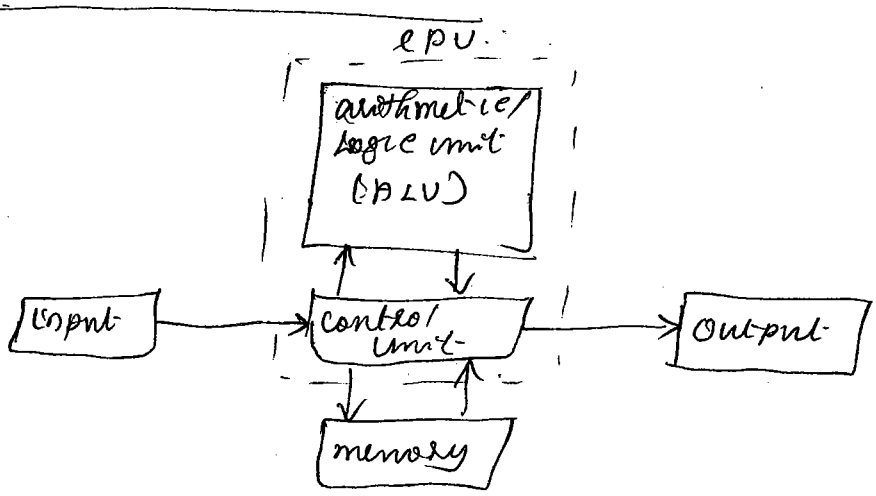
. The microprocessors used in these systems are categorized as:

* micro controllers on one chip
* general-purpose microprocessors with discrete components

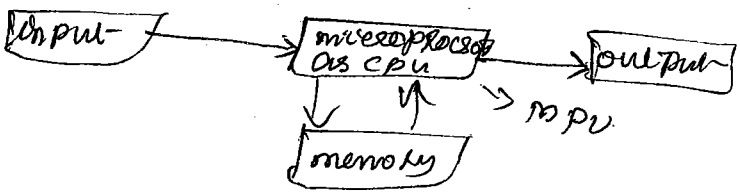
. microprocessor operates on bits (binary digit) represented in terms of voltages in the machine

. groups of bits → words and microprocessors are classified according to the word length

Microprocessor as a CPU (MPU)



Traditional computer



Computer with microprocessor as CPU

- A computer with microprocessor as CPU is known as a microcomputer.
- microprocessor and microprocessor unit (MPU) are used interchangeably.
- MPU implies a complete processing unit with the necessary control signals.

machine language

is a binary language, a set of instructions designed to the machine.

Computer understands machine language.

Eg. 0011 1100
3 C (increments the accumulator by 1)

- is also called 1st generation language)

Instruction

The microprocessor design engineer selects combinations of bit patterns and gives a specific meaning to each.

combination by using electronic logic circuits, is called an instruction

- made up of one word or several words
- often written in hexadecimal code for convenience and error free.

Assembly Language.

- Symbolic code is assigned for each instruction called mnemonic for easy understandability than hexadecimal numbers.
- The complete set of 8085 mnemonics is called the assembly language, and a program written in these mnemonics is called assembly language program

Eg. 0011 1100 ($3C_H$) → INR A.

1000 0000 (80_H) → ADD B

- 'Assembler' is a language translator which converts assembly language to machine language.
- Assembly language is called as 2GL (2nd generation language)
- Low level languages → machine language and assembly language.

High-Level Languages

→ machine independent languages called high level languages.

Eg. BASIC, PASCAL, C, C++, Java.

- 3GL (Third generation language)
- Compiler or interpreter converts high level language to machine language.

Microprocessor Architecture

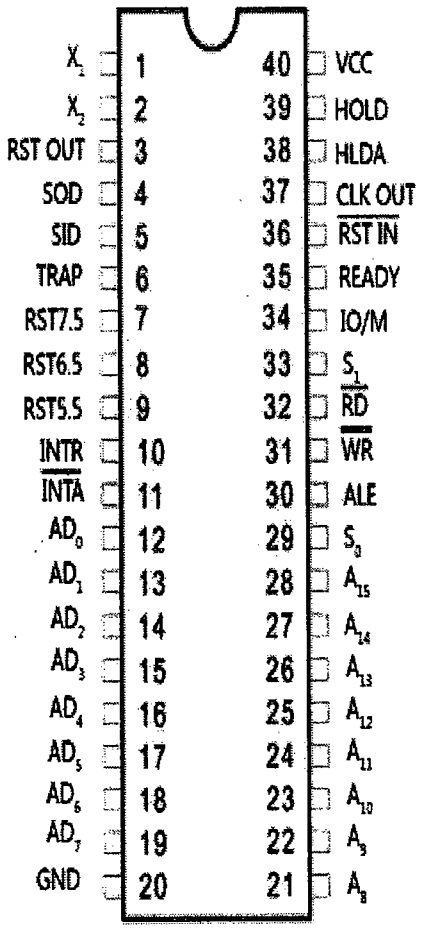
The process of data manipulation and communication via set of instructions is determined by the logic design of the microprocessor, called the architecture

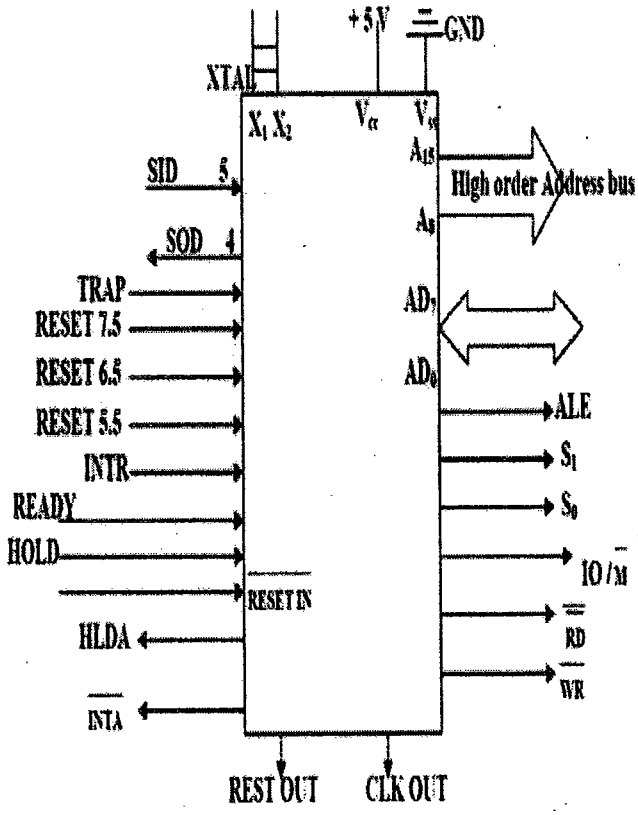
The 8085 microprocessor

The 8085A (commonly as 8085), is an 8bit general purpose microprocessor capable of addressing 64k of memory.

- has 40 pins
- requires +5V single power supply.
- operate with 3MHz single phase clock.
- 8085A-2 version can operate at the maximum frequency of 5MHz.
- predecessor of 8085 → 8080A

The logic pinout of 8085





All the signals can be classified into 6 groups

1) Address bus
2) Data bus
3) Control and status signals
4) Power supply and frequency signals
5) Externally initiated signals
6) Serial I/O ports.

1) Address bus

- 16 signal lines (pins) are used as address bus.
- split into 2 segments : $A_{15}-A_8$ and AD_7-AD_0.
- $A_{15}-A_8$ → unidirectional. from MP to peripheral.
  ↳ used for MSBs (higher order address of 16 bit address)

2) Data bus.

AD_7-A_0 → dual purpose. (bidirectional)
  → called multiplexed address/data bus.
  ↳ used as lower order address bus as well as data bus

3) control and status signals.

- 2 control signals ($\overline{RD}$, $\overline{WR}$)
- 3 status signals (IO/$\overline{M}$, S_1, S_0) } identify the nature of the operation
- ALE ⇒ to indicate the beginning of the operation
  ↳ Address latch enable
  ↳ is a +ve going pulse generated every time the 8085 begins an operation (machine cycle)
  ↳ indicates the bits on AD_7-AD_0 are address bits
  ↳ used to latch the low order address from the multiplexed bus and generate a separate set of 8 address lines

$\overline{RD}$ (Read ⇒ active low)

↳ indicates that the selected I/O or memory device is to be read and data are available on the data bus.

$\overline{WR}$ (write ⇒ active low)

↳ indicates that the data on the data bus are to be written into a selected memory or I/O location.

IO/$\overline{M}$ ⇒ used to differentiate between I/O and memory operations

⇒ high ⇒ I/O operation

⇒ low ⇒ memory operation

⇒ combined with $\overline{RD}$ & $\overline{WR}$ to generate I/O and memory control signals

S_1 and S_0 ⇒ similar to IO/$\overline{M}$ ⇒ can identify various operations

IO/$\overline{M}$	S_1	S_0	operation	Control signals
0	1	1	opcode fetch	$\overline{RD}$ = 0
0	1	0	memory read	$\overline{RD}$ = 0
0	0	1	memory write	$\overline{WR}$ = 0
1	1	0	I/O read	$\overline{RD}$ = 0
1	0	1	I/O write	$\overline{WR}$ = 0
1	1	1	Interrupt acknowledge	$\overline{INTA}$ = 0
Z	0	0	Halt	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA}$ = 1.
Z	X	X	Hold	
Z	X	X	Reset	

Power Si

Z ⇒ high impedance

X ⇒ unspecified (Register

Tri-State

3 states for device (Logic 0, 1, and high impedance)

↳ has 3rd state called enable

↳ when it is enabled, the device functions some way as ordinary logic device.

↳ when disabled, goes into high impedance state (disconnected

power supply and clock frequency

V_{CC} +5 V power supply

V_{SS} : Ground

X_1, X_2 → crystal (RC, LC n/w) is connected to these pins

↳ the frequency is internally divided by 2. ($\frac{6\text{ MHz}}{2} = 3$)

↳ gives the frequency to mp (3 MHz)

↓ Crystal frequency

CLK (OUT) → Clock output → used as the system clock for other devices

Externally Initiated signals including Interrupts.

5 interrupt signals

- INTR (Interrupt Request) → input to mp
- INTA (Interrupt acknowledge → output from mp
- RESET IN (low) → the PC is set to 0, the buses are tristated mpu is reset.
- RESET OUT → indicates that the mpu is being reset → used to reset other devices
- HOLD (input) → indicates that a peripheral such as a DMA controller is requesting the use of address and data buses
- HLDA (output) → hold acknowledge
- RST 7.5, RST 6.5, RST 5.5 } Restart interrupts → that transfer the program control to specific memory locations

READY (input) → used to delay the microprocessor read or write cycles until a slow responding peripheral is ready to send or accept data. When this signal goes ~~high~~ low, the mp waits for an integral no. of clock cycles until it goes high

Serial I/O Ports

2 signals for implementing serial transmission

SID (serial input data)

SOD (serial output data)

Internal block diagram / Architecture / functional block diagram of 8085

definition is same as described under 8085 microprocessor.

(refer page no: 4)