PRESENTATION

On

Subject

PLC & Microcontroller

(Electrical Engg 6th Semester)

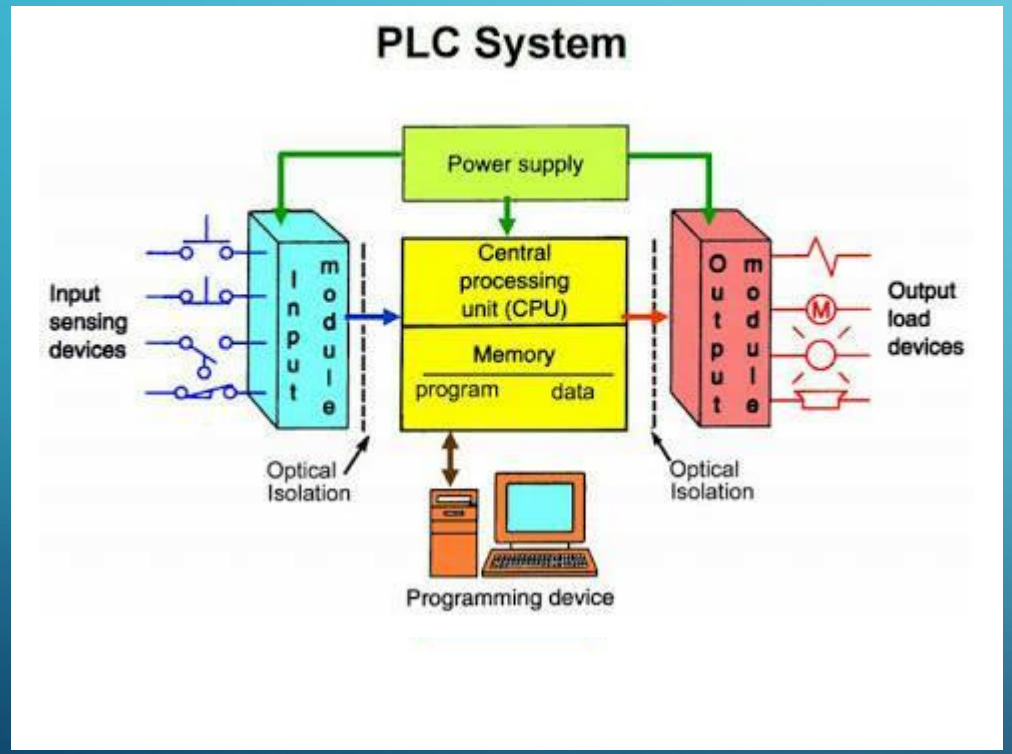
INTRODUCTION OF PLC

- Programmable logic controller
- It is a digital electronic device
- It can be programmable
- It perform all logical function and control the machine
- It increases technology and confort
  It can make a industry fully automated
- It saves time also

ADVANTAGES OF PLC OVER CONVENTIONAL RELAY

- PLCs does not have tear and wear
- It require less space
- It has a software it store a programm
- It has less weight for large machines
- It control more input and output
- It’s maintenance cost is less
- Because it is programmable so one PLC can control other machines
- Noise less operation
- Less power consumption

BLOCK DIAGRAM OF PLC



- PLC consists :
- Microprocessor : CPU ,memory and communication port
- Input&Output module
- Power supply
- Programmable device(PC)

BLOCK DIAGRAM OF PLC LANGUAGES

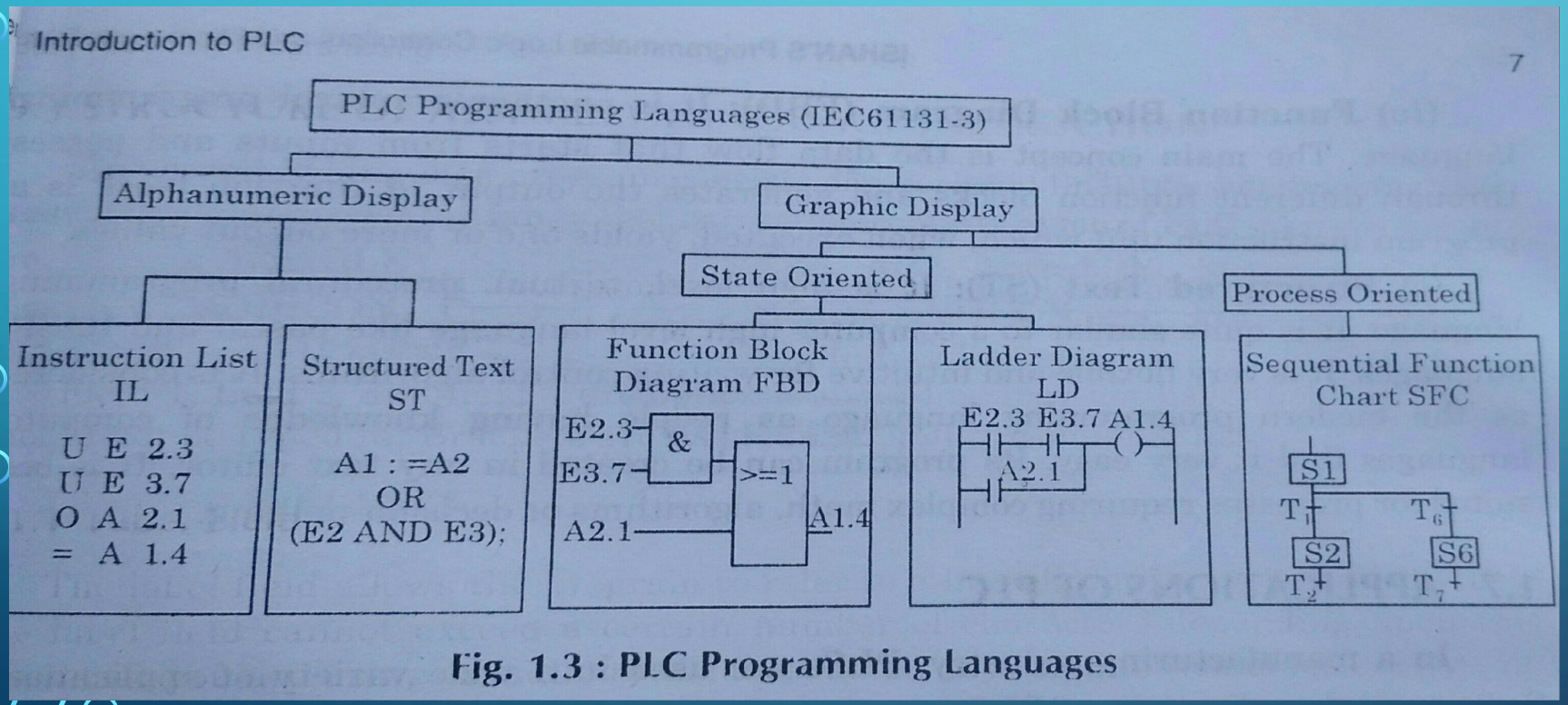
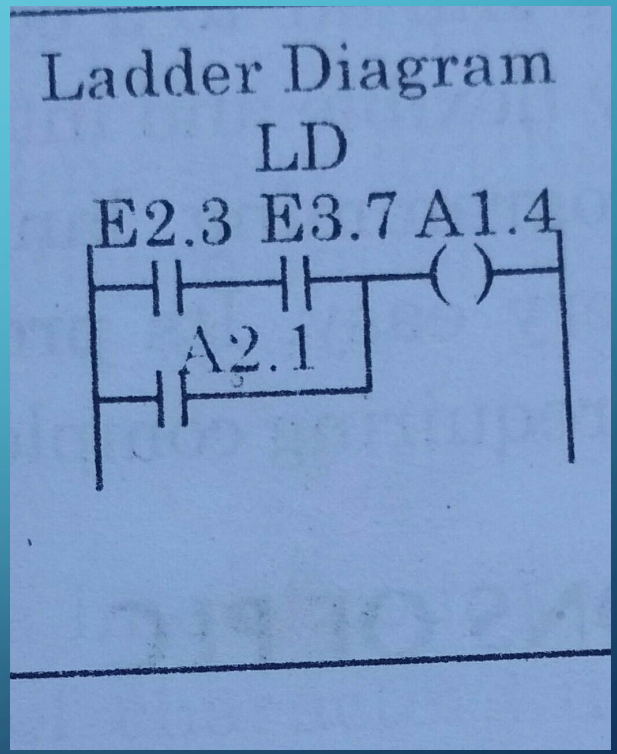


Fig. 1.3 : PLC Programming Languages

- Structured text(ST)

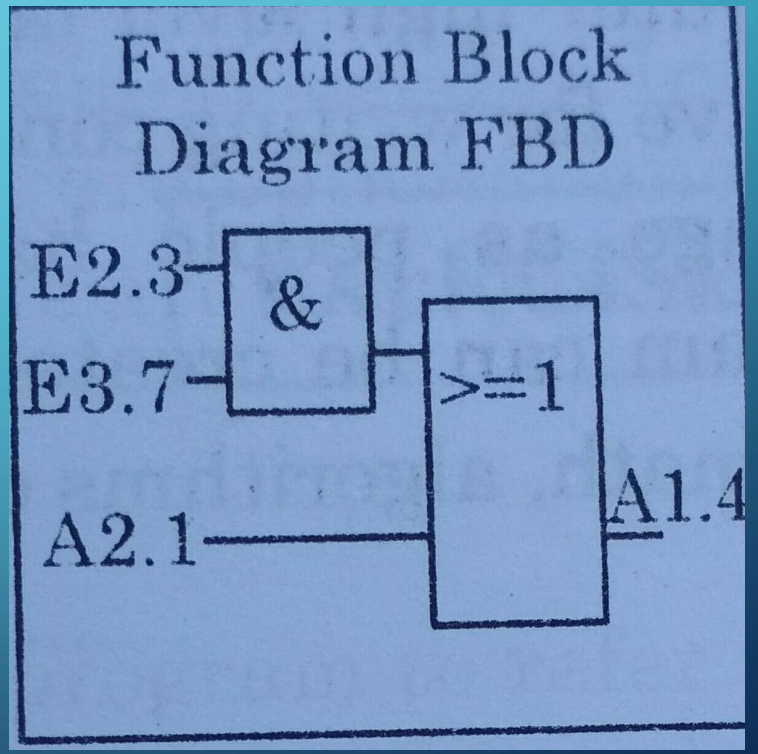
LADDER DIAGRAM

- It is a graphical language
- It is based on relay logic
- It is easy to understand
- It is a low level language
- Generally used in PLCs



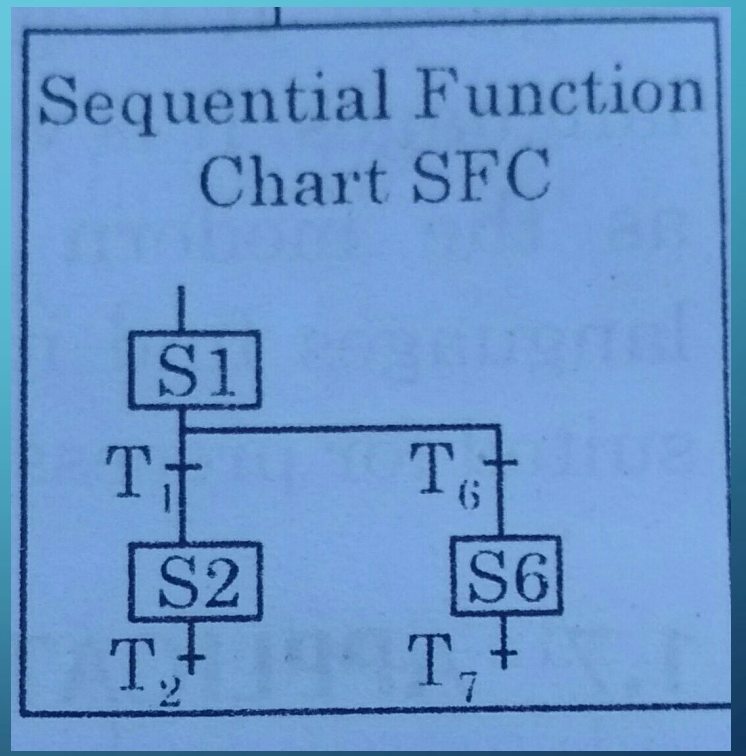
FUNCTION BLOCK DIAGRAM

- It is a graphical language also
- It has a main concept of data flow from input to output through the function block
- It has a function block which is carried out by machines



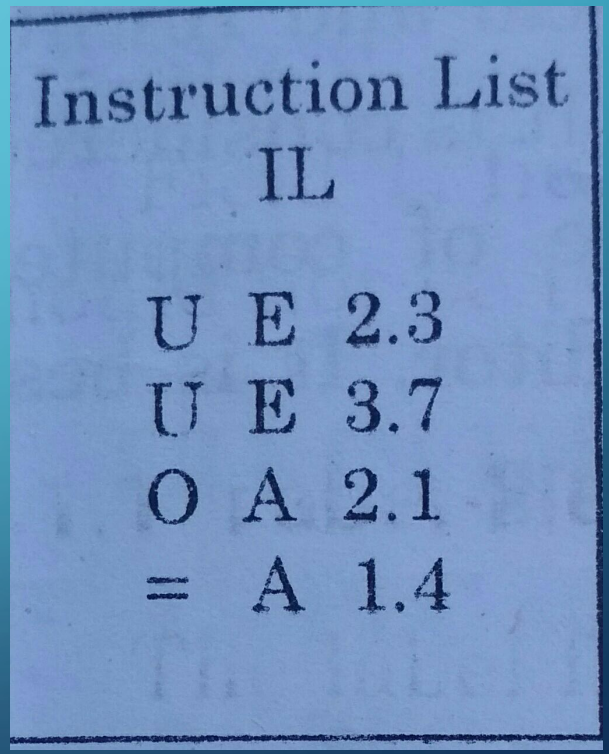
SEQUENTIAL FUNCTION CHART

- It is a high level language
- Less easy to understand
- It is a chart of function in sequence which is perform by machines
- It is as a algorithm
  Use for complex control circuit



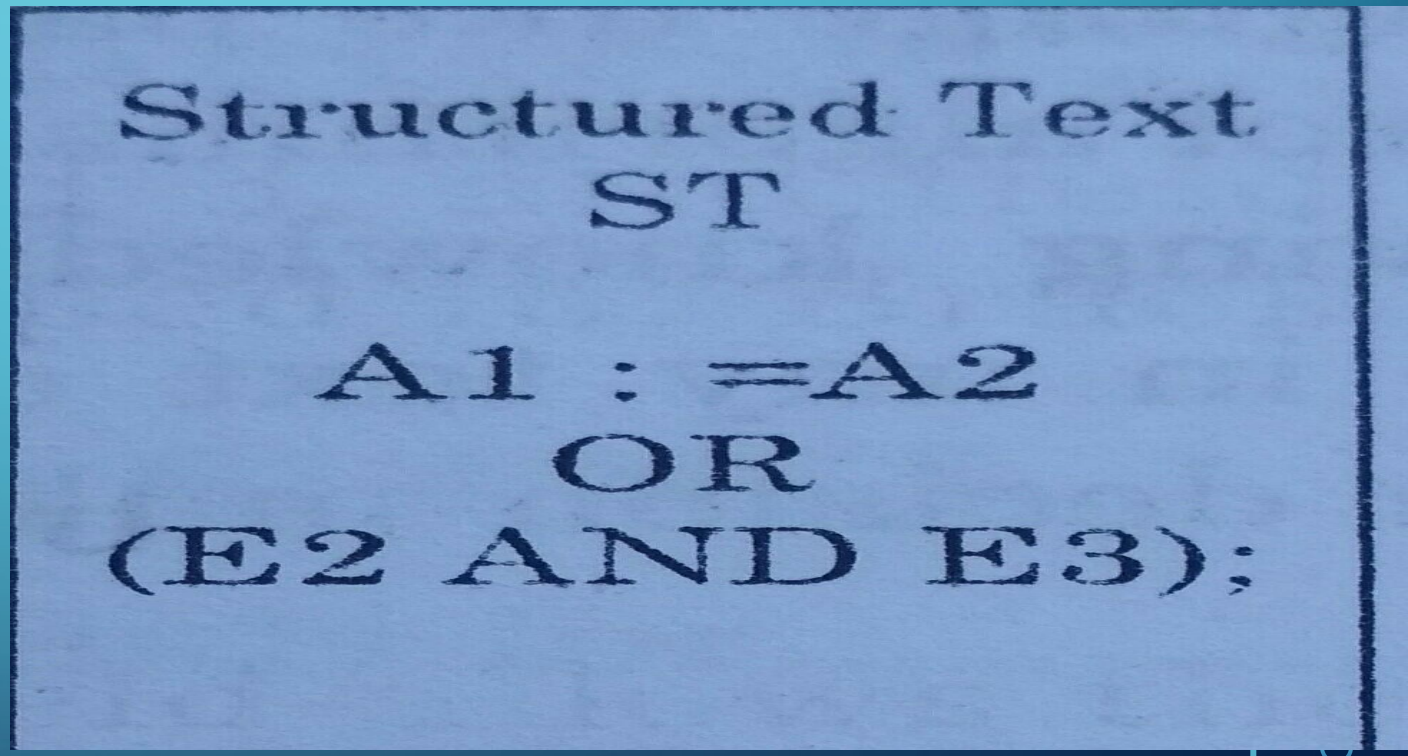
INSTRUCTION LIST

- It is a alphanumeric language
- It is a list of instructions
- It requires less memory area
- Execution rate is very fast
- Difficult to understand



STRUCTURED TEXT

- It is also a alphanumeric language
- Difficult to understand
- Used in complex operation
- High level language
- Used in dicision making

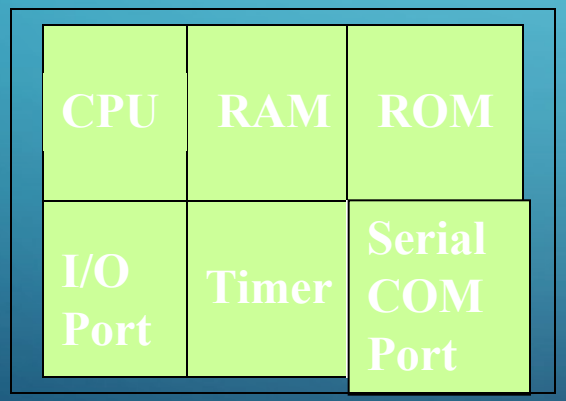


MANUFACTURER OF PLC

- ALLEN BRADLEY
- SIEMENS
- TEXAS
- ABB
- MODICON
- But Allen Bradley & Siemens are mostly used in INDIA

Microcontroller :

- A smaller computer
- On-chip RAM, ROM, I/O ports...
- Example : Motorola's 6811, Intel's 8051, Zilog's Z8 and PIC 16X



A single chip

Microcontroller

Microprocessor vs. Microcontroller

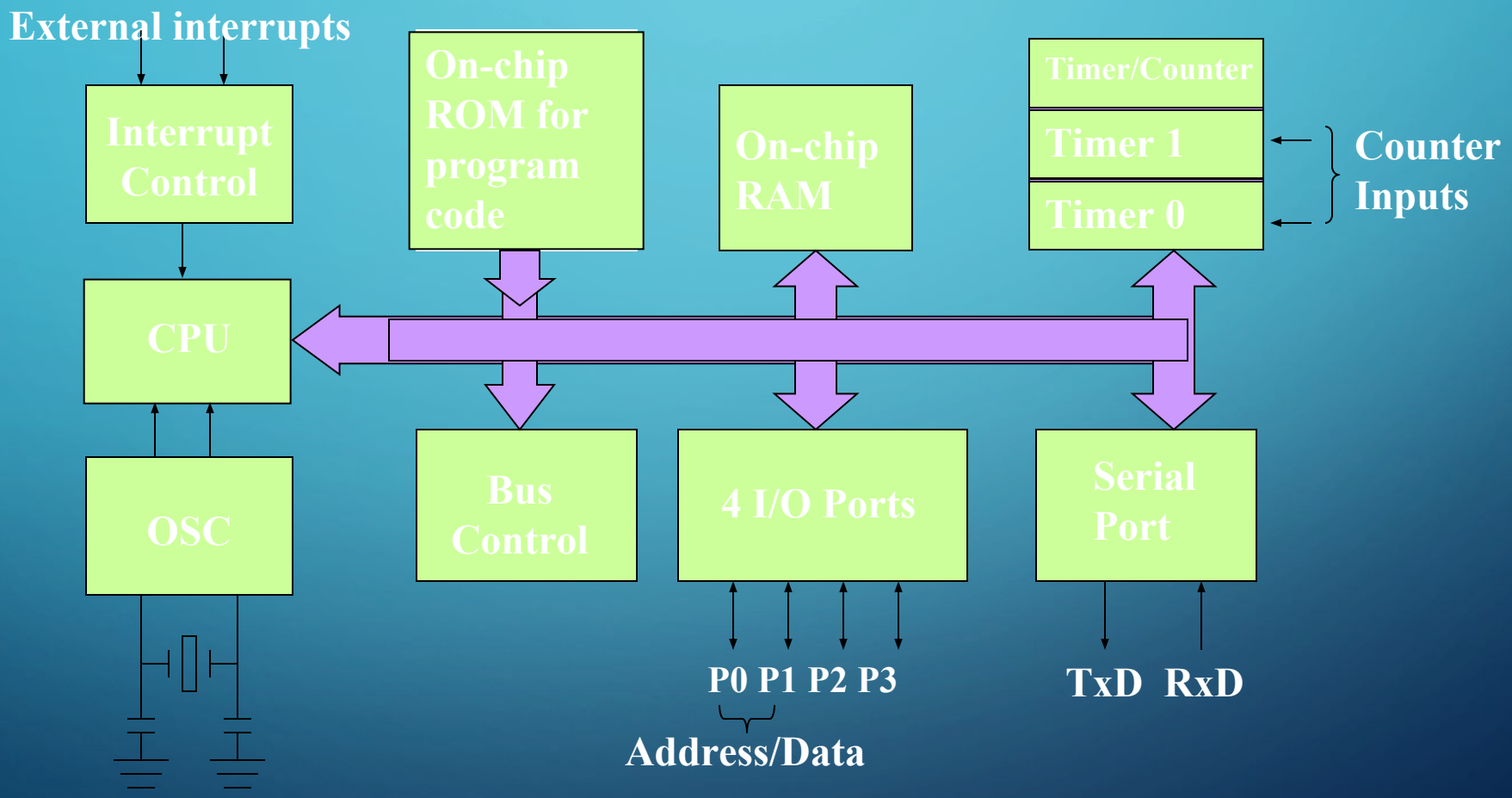
Microprocessor

- CPU is stand-alone, RAM, ROM, I/O, timer are separate
- designer can decide on the amount of ROM, RAM and I/O ports.
- expansive
- versatility
- general-purpose

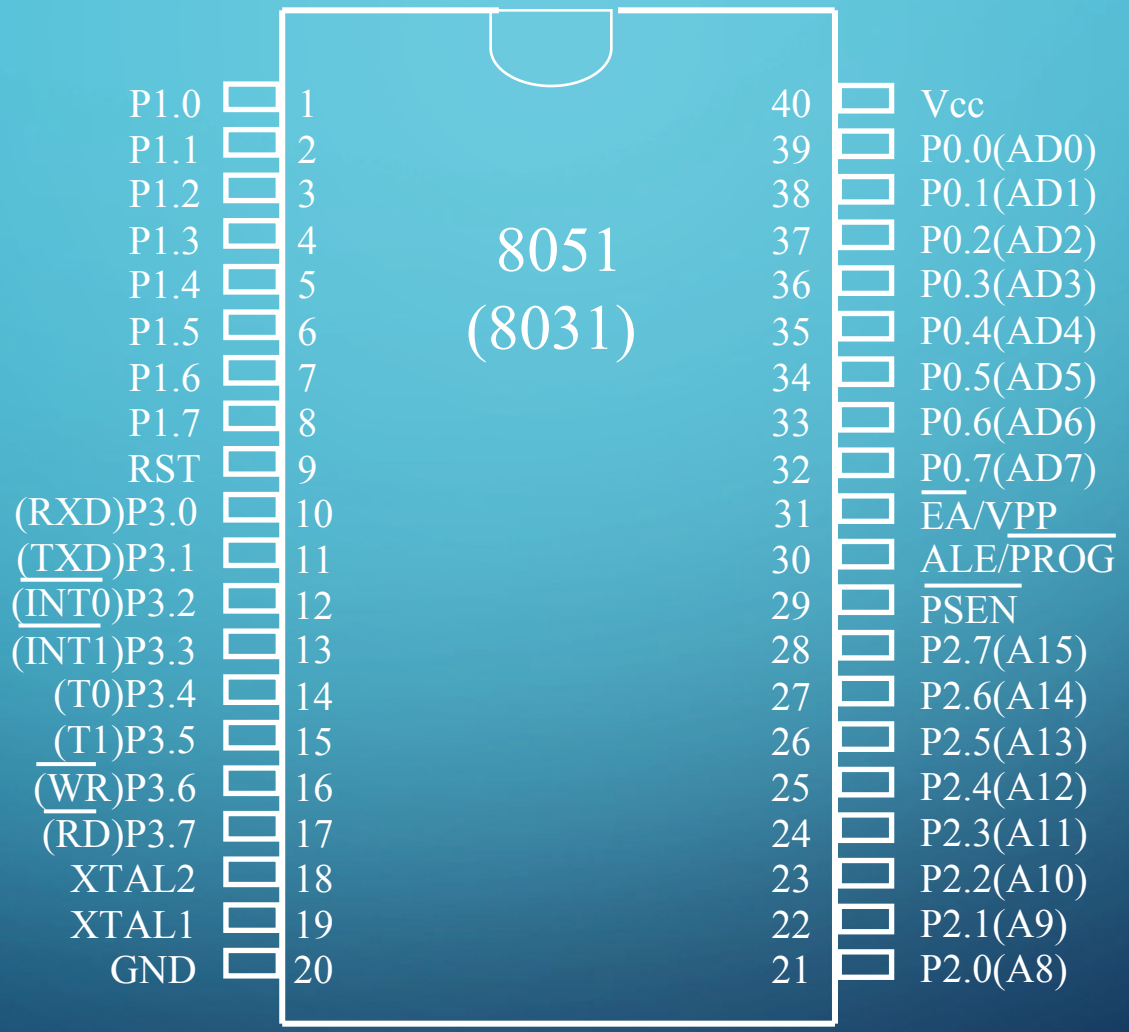
Microcontroller

- CPU, RAM, ROM, I/O and timer are all on a single chip
- fix amount of on-chip ROM, RAM, I/O ports
- for applications in which cost, power and space are critical
- single-purpose

BLOCK DIAGRAM



Pin Description of the 8051



FUNCTION OF PINS

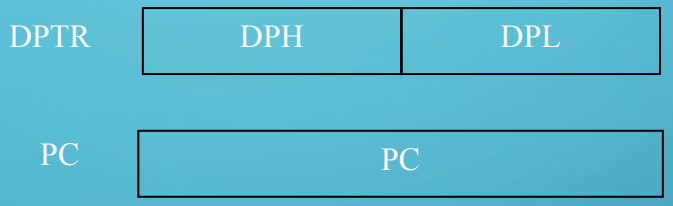
- Vcc (Pin 40) for supply voltage to chip.
- Vss (Pin 20) to ground.
- XTAL 1 and XTAL 2
- RST – Reset (Active high)
- EA – External Access pin (Active low)
- PSEN – Program Store Enable output pin (Active low)
- ALE – Address Latch Enable Pin (Active high)

PORT PINS

- Port 0 (P0) – Pins 32 to 39 , 8-bit open drain bidirectional i/o port.
- Port 1(P1) – Pins 1 to 8 are port 1 pins ,8-bit bidirectional i/o port with internal pull ups
- Port 2 (P2) – Pins 21 to 28 are port 2 pins. Carry high order address signal A8 to A15.
- Port 3 (P3) – Pins 10 to 17 are port 3 pins. It also carry some special signals like RXD, TXD, WR,etc.

Registers in Microcontroller

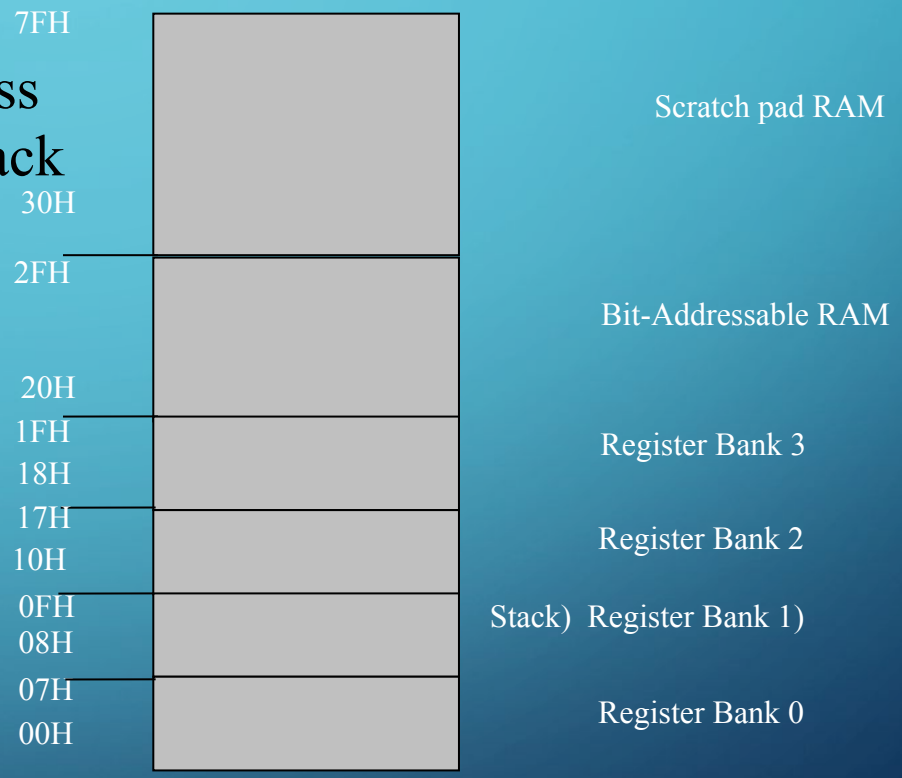
Some 8-bitt Registers of the 8051



Some 8051 16-bit Register

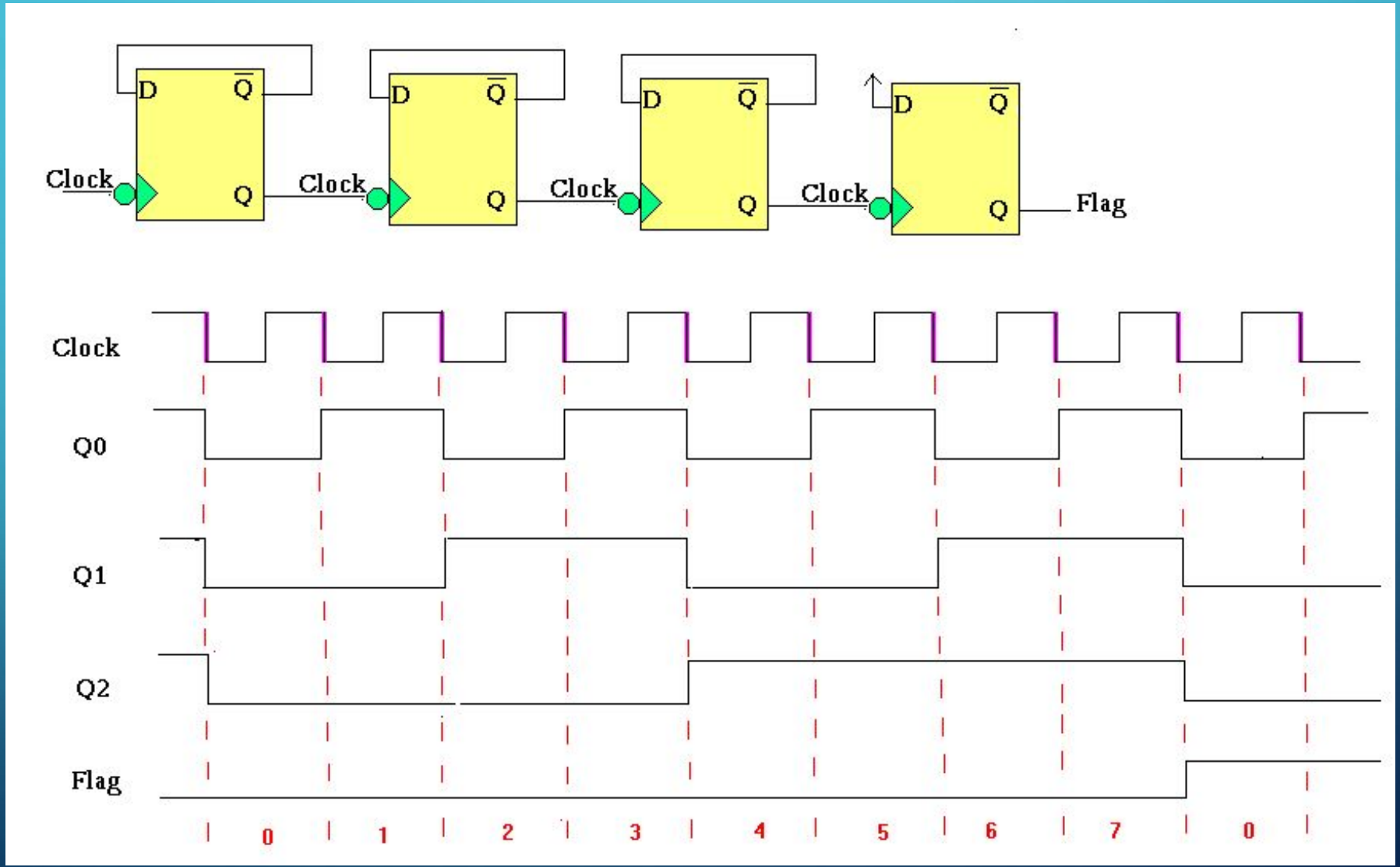
STACK IN THE 8051

- The register used to access the stack is called **SP** (stack pointer) register.
- The stack pointer in the 8051 is only 8 bits wide, which means that it can take value 00 to FFH. <u>When 8051 powered up, the SP register contains value 07.</u>



ER :

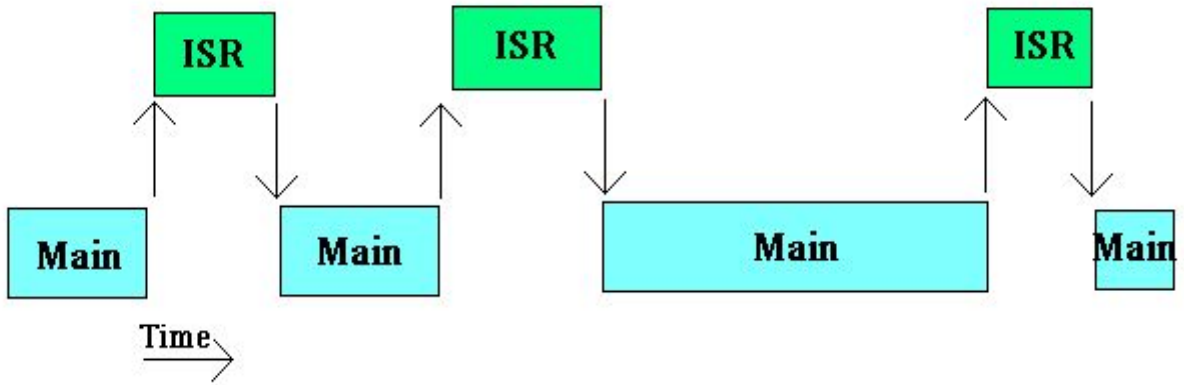
Timer:



INTERRUPT :

Program execution without intrrupts :

Program execution with intrrupts :



ISR : Intrrupt Service Routin

Numerical Bases Used in Programming

- Hexadecimal
- Binary
- BCD

Hexadecimal Basis

- Hexadecimal Digits:

1 2 3 4 5 6 7 8 9 A B C D E F

A=10
B=11
C=12
D=13
E=14
F=15

Decimal, Binary, BCD, & Hexadecimal Numbers

$(43)_{10}=$

$(0100\ 0011)_{BCD}=$

$(\boxed{0010}\boxed{1011})_2=$

$(\ 2\ \ \ B\ )_{16}$

Register Addressing Mode

```
MOV Rn, A      ;n=0,..,7
ADD      A, Rn
MOV DPL, R6
```

~~MOV DPTR, A~~

~~MOV Rm, Rn~~

Direct Addressing Mode

Although the entire of 128 bytes of RAM can be accessed using direct addressing mode, it is most often used to access RAM loc. 30 – 7FH.

```
MOV R0, 40H
MOV 56H, A
MOV A, 4        ; ≡ MOV A, R4
MOV 6, 2        ; copy R2 to R6
                ; MOV  R6,R2 is invalid !
```

Immediate Addressing Mode

MOV A,#65H

MOV R6,#65H

MOV DPTR,#2343H

MOV P1,#65H

SETB bit ; bit=1
CLR bit ; bit=0

```
SETB    C           ; CY=1
SETB    P0.0            ;bit 0 from port 0 =1
SETB    P3.7            ;bit 7 from port 3 =1
SETB    ACC.2           ;bit 2 from ACCUMULATOR =1
SETB    05          ;set high D5 of RAM loc. 20h
```

Note:

CLR instruction is as same as SETB
i.e.:

```
    CLR         C   ;CY=0
```

But following instruction is only for CLR:

```
    CLR         A   ;A=0
```

DEC byte ;byte=byte-1

INC byte ;byte=byte+1

```
INC     R7
DEC     A
DEC     40H     ; [40]=[40]-1
```

LOOP and JUMP Instructions

Conditional Jumps :

JZ	Jump if A=0
JNZ	Jump if A/=0
DJNZ	Decrement and jump if A/=0
CJNE A,byte	Jump if A/=byte
CJNE reg,#data	Jump if byte/=#data
JC	Jump if CY=1
JNC	Jump if CY=0
JB	Jump if bit=1
JNB	Jump if bit=0
JBC	Jump if bit=1 and clear bit

Call instruction

```
        SETB    P0.0
        .
        .
        CALL    UP
        .
        .
        .
UP:     CLR  P0.0
        .
        .
        RET
```