MICROCONTROLLER ARCHITECTURE AND PROGRAMMING

Prof. D. A. Patil

Head

Department of B.Sc. Computer Science (Entire)

Smt. Kasturbai Walchand College, (Arts and Science) Sangli

Structure of Course

Code	Paper	Name of the Paper	Marks
Electronics Semester -I			
GEC-303	**Paper -V**	**Computer Organization**	**50**
GEC-304	Paper -VI	Computer Instrumentation	50
Electronics Semester -II			
GEC-403	**Paper- VII**	**Microcontroller Architecture & Programming**	**50**
GEC-404	Paper- VIII	Communication Techniques	50
Practical (Annual)			
LAB-6		Practical Examination Based on Theory Papers V,VI.VII & VIII	100

Course Contents

1. Introduction to Microcontroller
2. 8051 Instruction Set
3. Facilities in 8051
4. Interfacing Methods

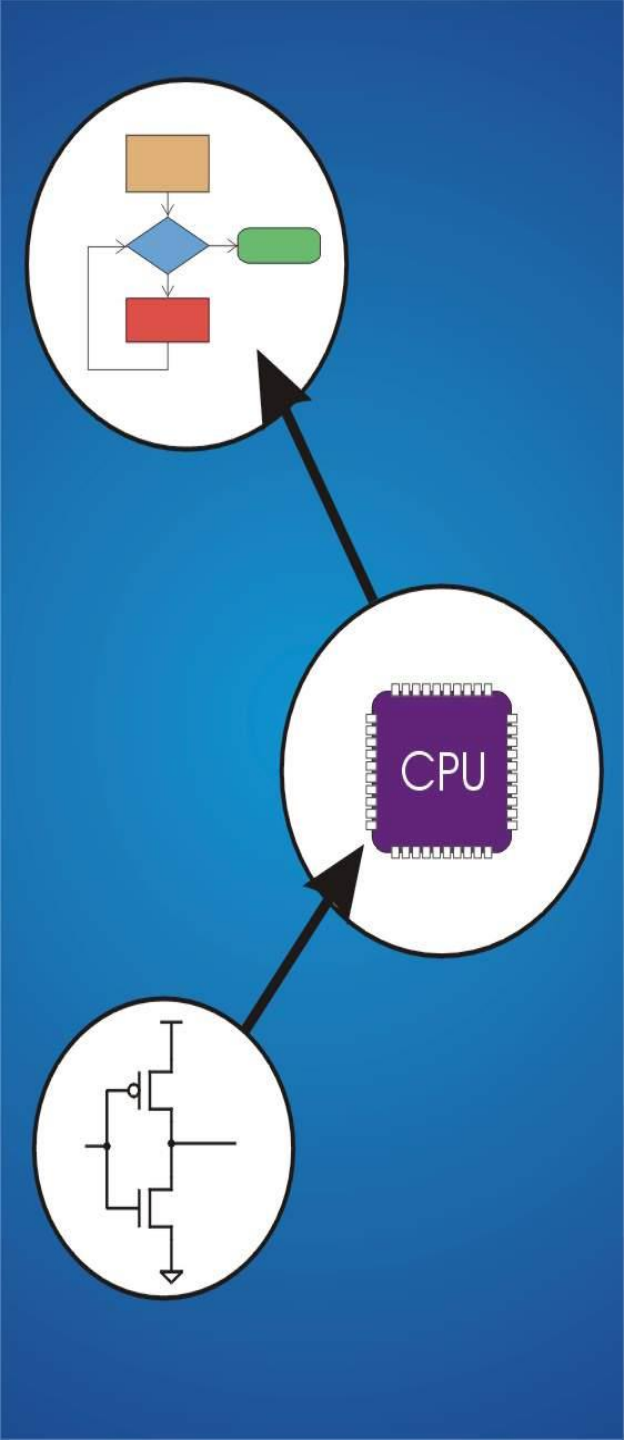
Course Contents

INTRODUCTION TO MICROCONTROLLER

Comparison of Microcontroller & Microprocessor, Survey of 4-Bit, 8-Bit, 16- Bit And32-Bit Microcontrollers and their application areas,

Study of 8051 and its Family(89C51, 8031, 8032, 8052, 8751, Phillips (RD2)89C51VRD2).

Architecture of 8051:Block Diagram of 8051 and Study of Internal Blocks, Reset and Clock, Registers, Flags and Internal Memory, SFR, I/O Ports

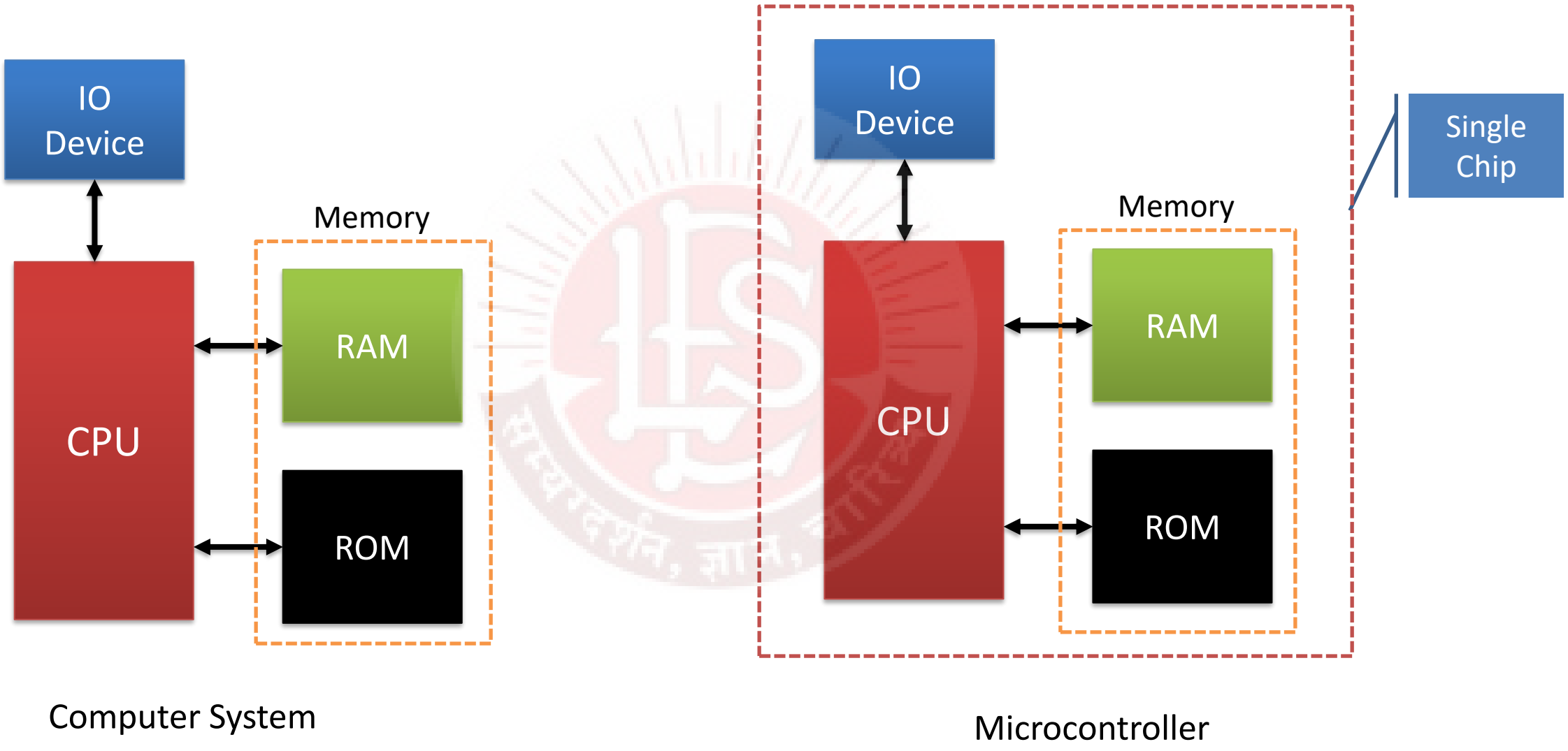


UNIT – I

Introduction to Microcontroller

What is Microcontroller ?

Mini Computer System



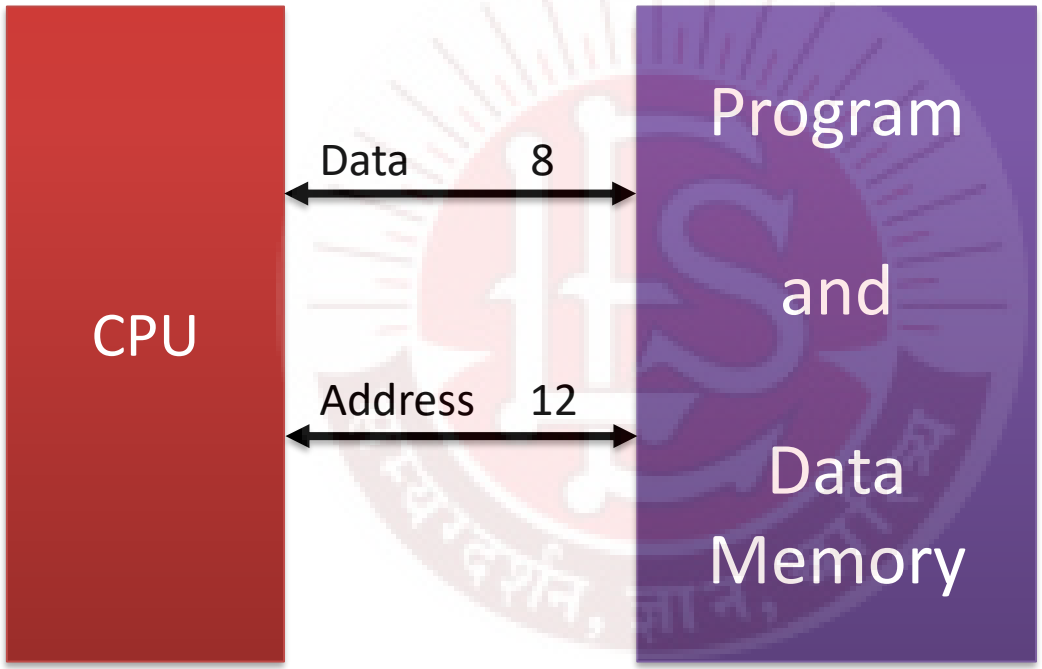
Comparison : Microcontroller - Microprocessor

Microcontroller	Microprocessor
Inbuilt RAM and ROM	Do not have Inbuilt RAM and ROM
Inbuilt Timer	Do not have Inbuilt timers
I/O ports are available	I/O ports are not available and requires extra devices like 8255
Inbuilt Serial Port	Serial port is not available and requires extra device like 8251

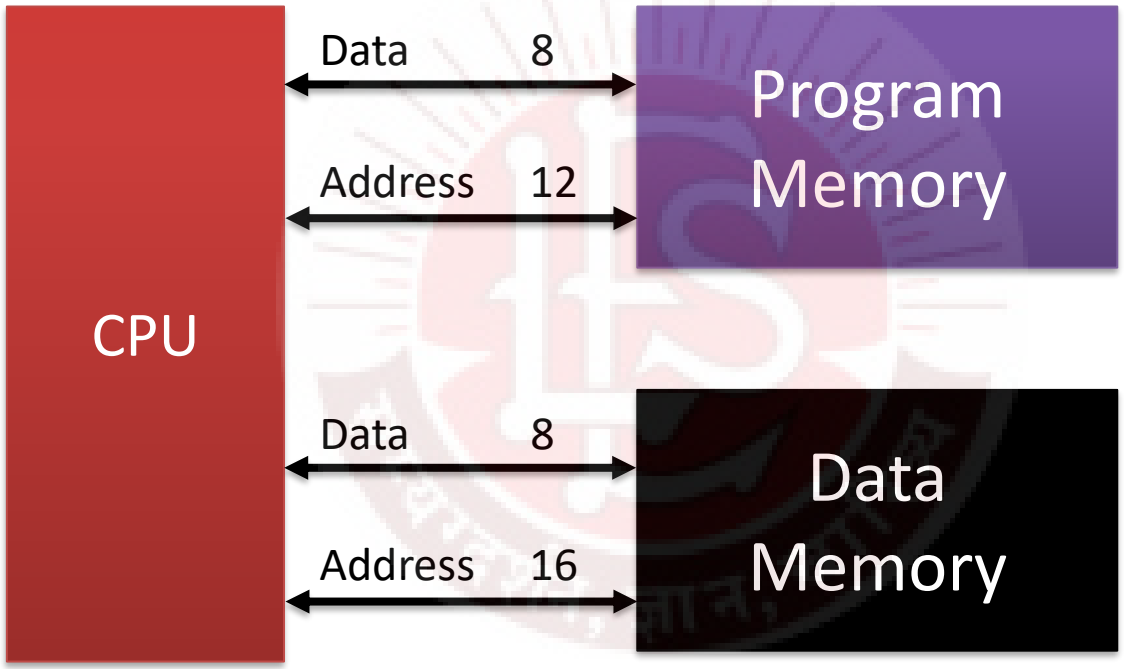
Comparison : Microcontroller - Microprocessor

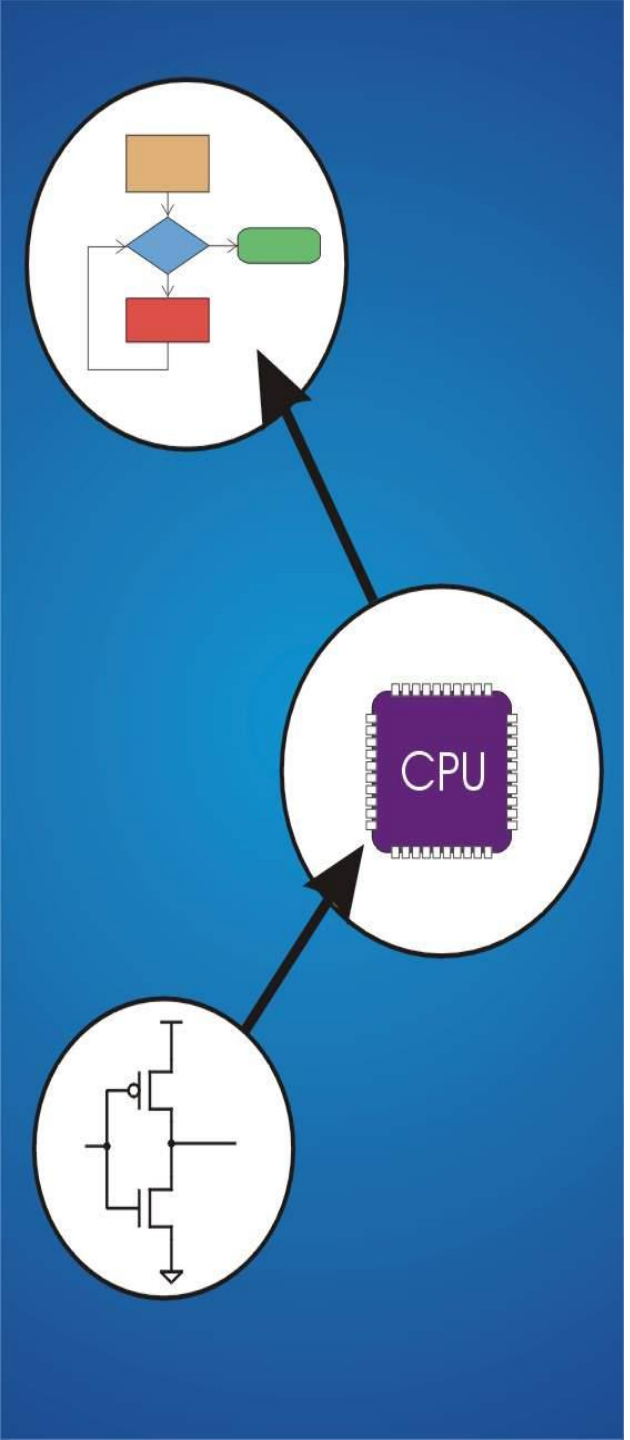
Microcontroller	Microprocessor
Separate memory to store program and data	Program and data are stored in same memory
Many multifunction pins are available	Less multifunction pins are available
Direct Boolean operations on individual bit is possible	Direct Boolean operations is not possible
Few Instructions to read or write data to or from external memory	Many Instructions to read or write data to or from external memory

Fundamental Architecture : Von-Neuman's



Fundamental Architecture : Harvard





UNIT – I

Introduction to Microcontroller

Microcontroller Families

Devices and Families :

On chip ROM Size
(Program Memory)

On Chip RAM Size
(Data Memory)

Number of Timers

I/O Ports

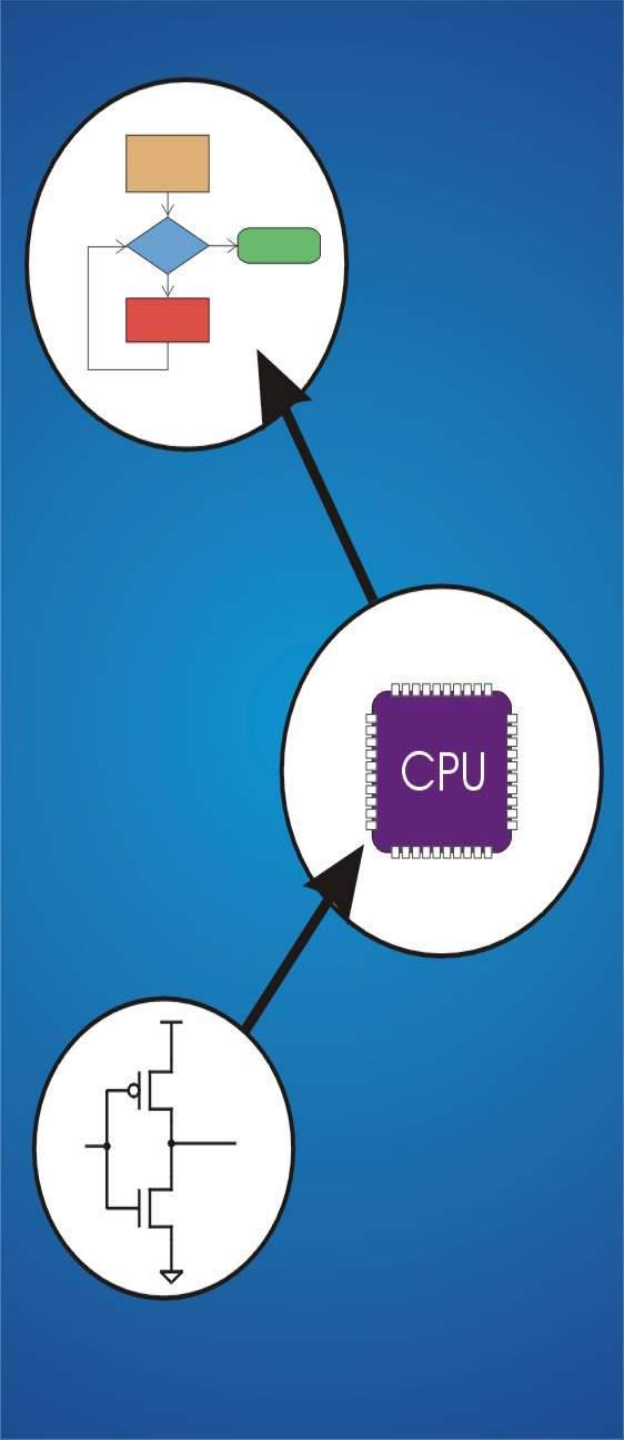
ADC's and DAC's

Many Other features

Family Number	Description
80XXX	ROM Less
82XXX	Predesigned masked ROM 8242 Keyboard controller
83XXX	Masked ROM
86XXX	Ultraviolet EPROM
87XXX	OTP (One Time Programmable ROM)

Study of MCS-51 :

Device	RAM Bytes	ROM	Speed MHz	Timers	Ports
8031	128	-	12	2	4 x 8
8032	256	-	12	2	4 x 8
8051	128	4096	12	2	4 x 8
8052	256	8192	24	2	4 x 8
MCS - 251					
8X251 SA	1 K	8 K	16	3	32
USB Equipped MCS - 251					
80930 AX	512-1024	-	12	-	-
83930 AX	512-2024	8-16 K	12	-	-

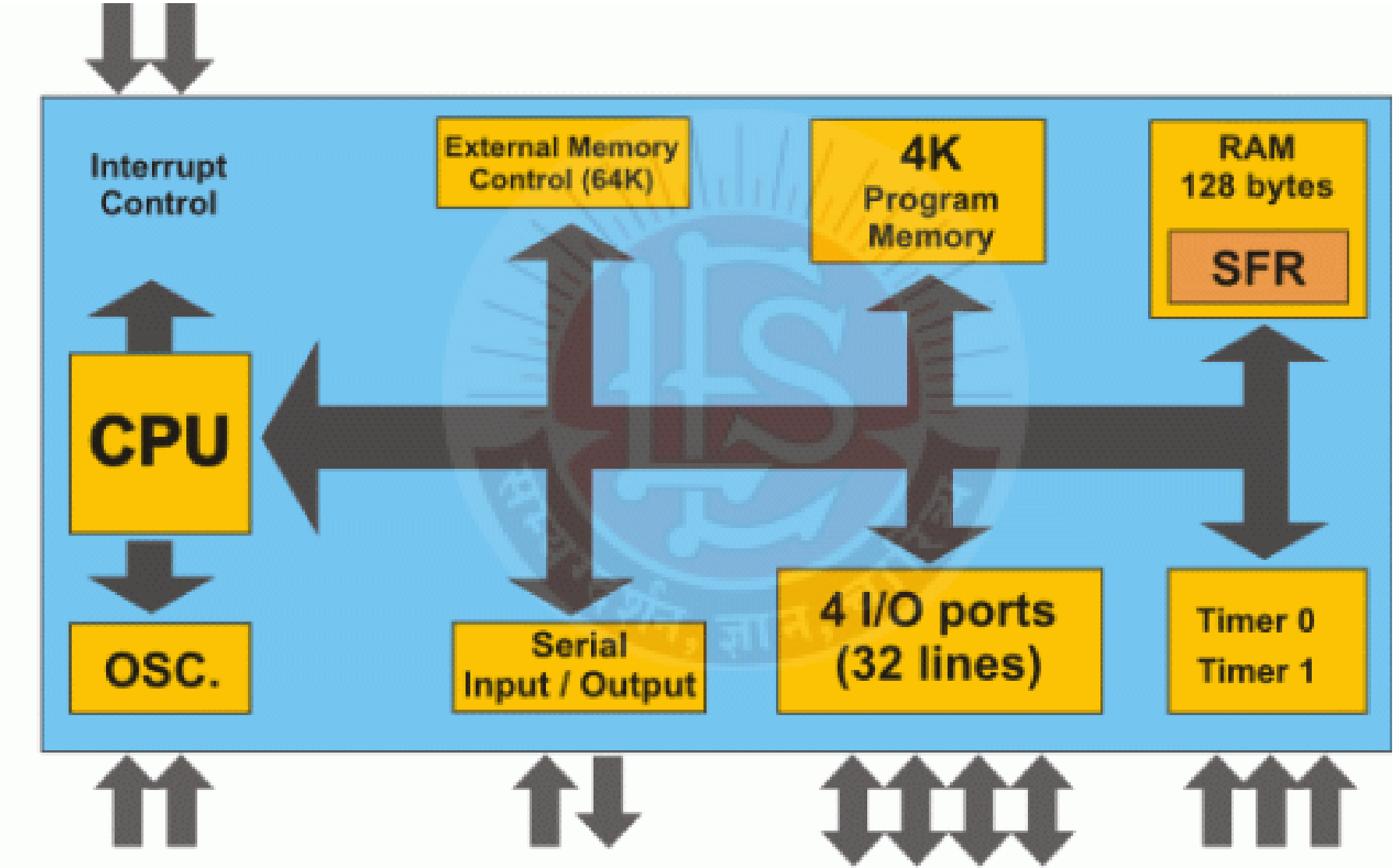


UNIT – I

Introduction to Microcontroller

Architecture of 8051

8051 Microcontroller :



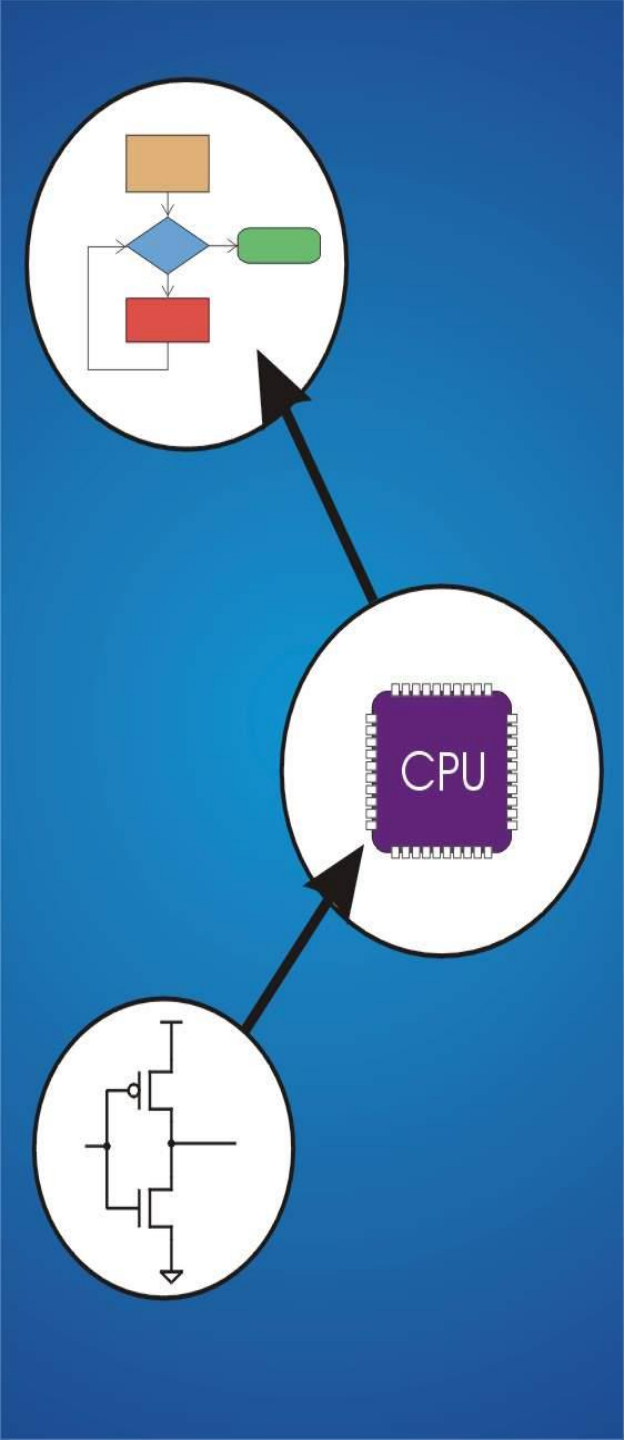
Features of 8051

Pin Diagram

Architecture

8051 Microcontroller : Features

- 8 bit Microcontroller
- 16 bit Address Bus - 2^{16} = 64 K
- Built in RAM – 256 bytes
  - » 128 bytes - Data Memory
  - » 128 bytes - SFR
- Built in ROM – 4 K (4096 bytes)
- Four 8 bit Bidirectional ports – P0, P1, P2 and P3
- Programmable Serial Port
- Two Timers T0 and T1
- Five Interrupt Sources
- Supports Bit level Boolean logic operations directly applied to Port, Registers and SFR
- Four separate register banks – Each bank consists 8 registers

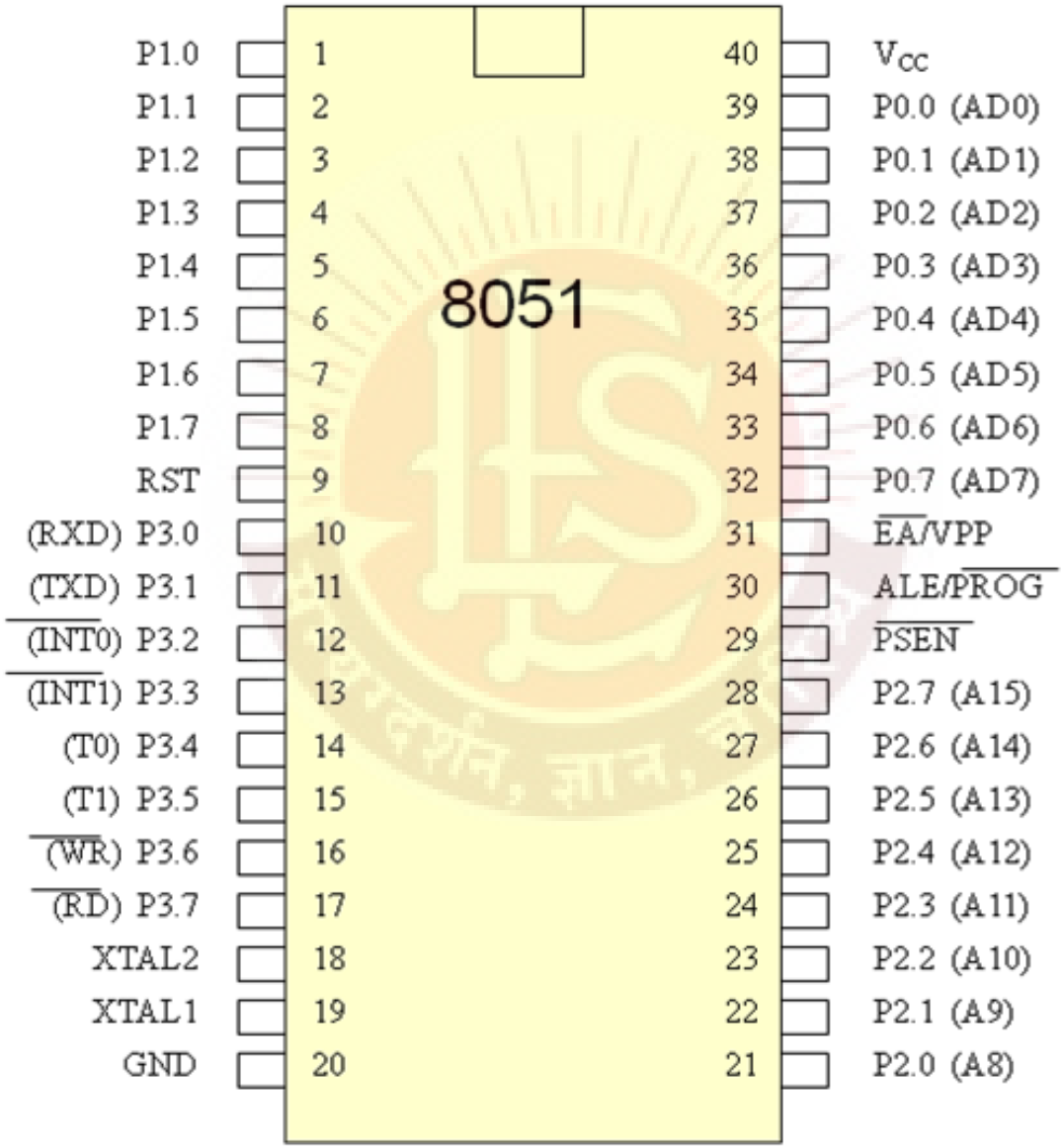


UNIT – I

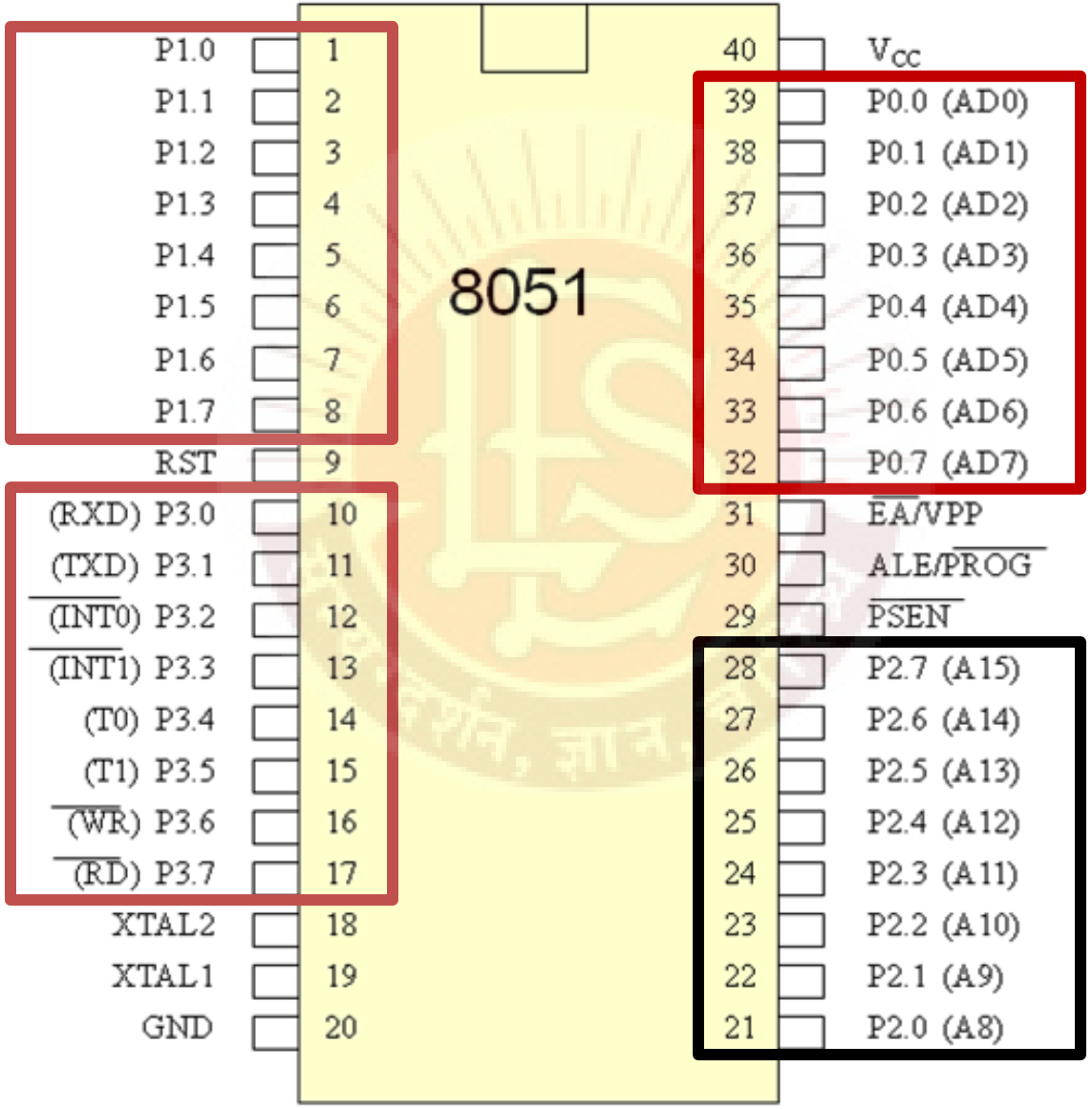
Introduction to Microcontroller

Pin Diagram of 8051

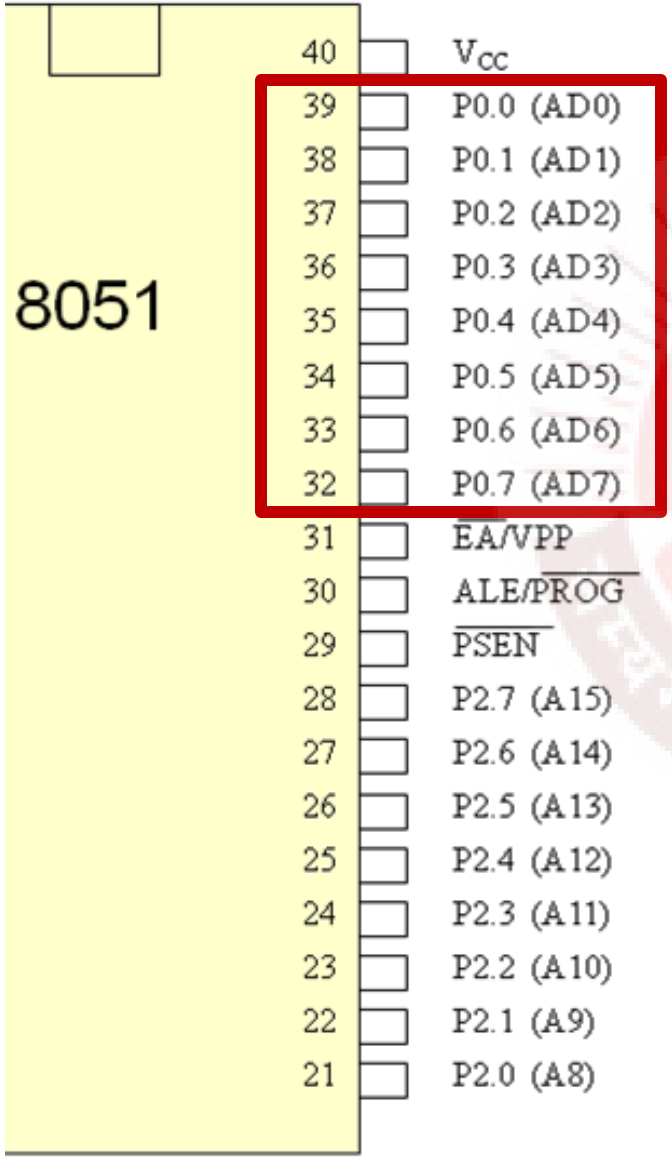
8051 Pin Diagram :



8051 Pin Diagram :

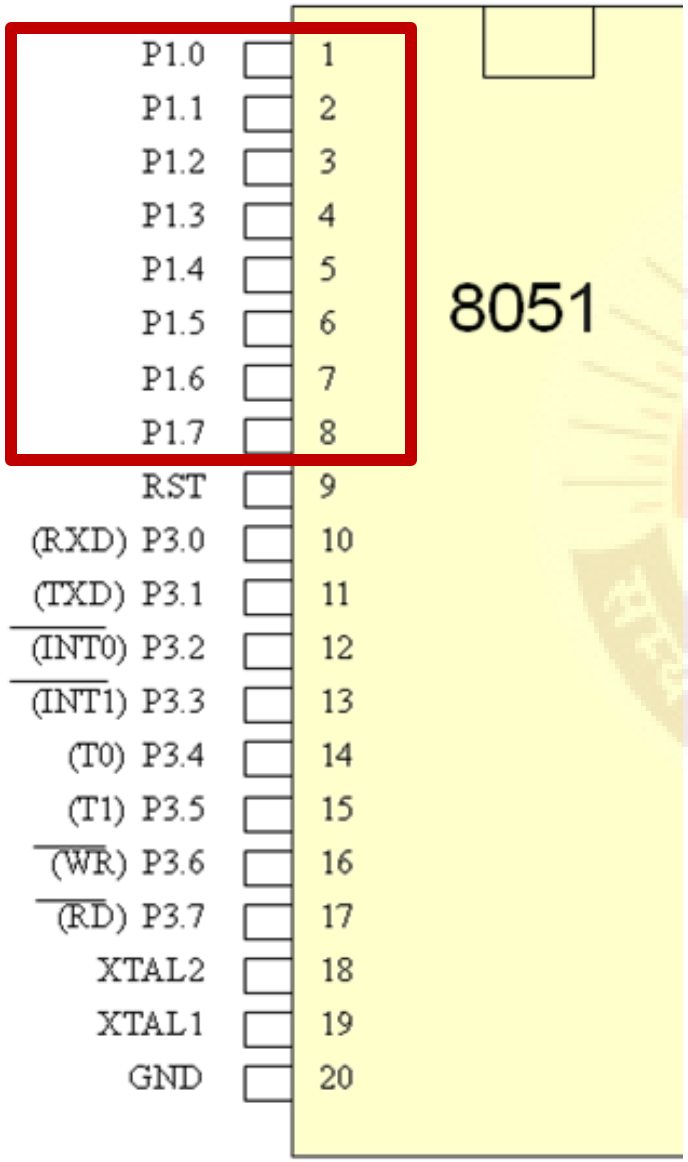


8051 Pin Diagram : **Port 0 (P0.0/AD0 – P0.7/AD7)**



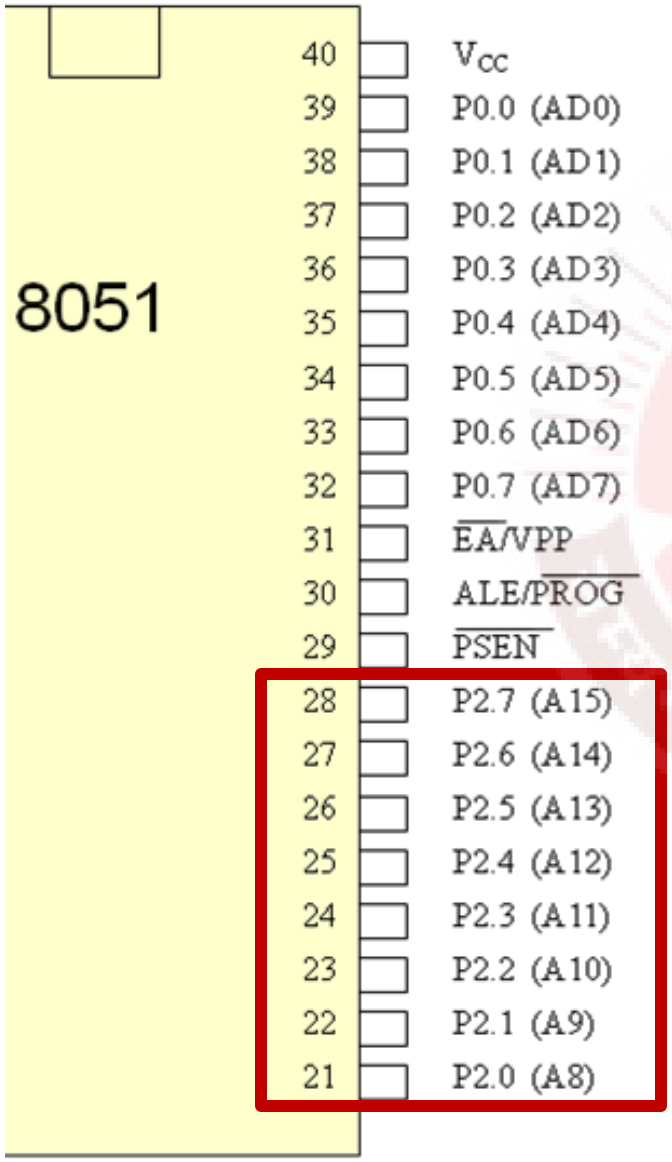
- 8 Bit bidirectional Port
- Open drain port hence external pull-up resistors are required
- Entire port (P0) or one of the pin (P0.0 – P0.7) configured as I/P or O/P
- Port is used to carry low order address (A0 – A7) or Data (D0 – D7)

8051 Pin Diagram : **Port 1 (P1.0 – P1.7)**



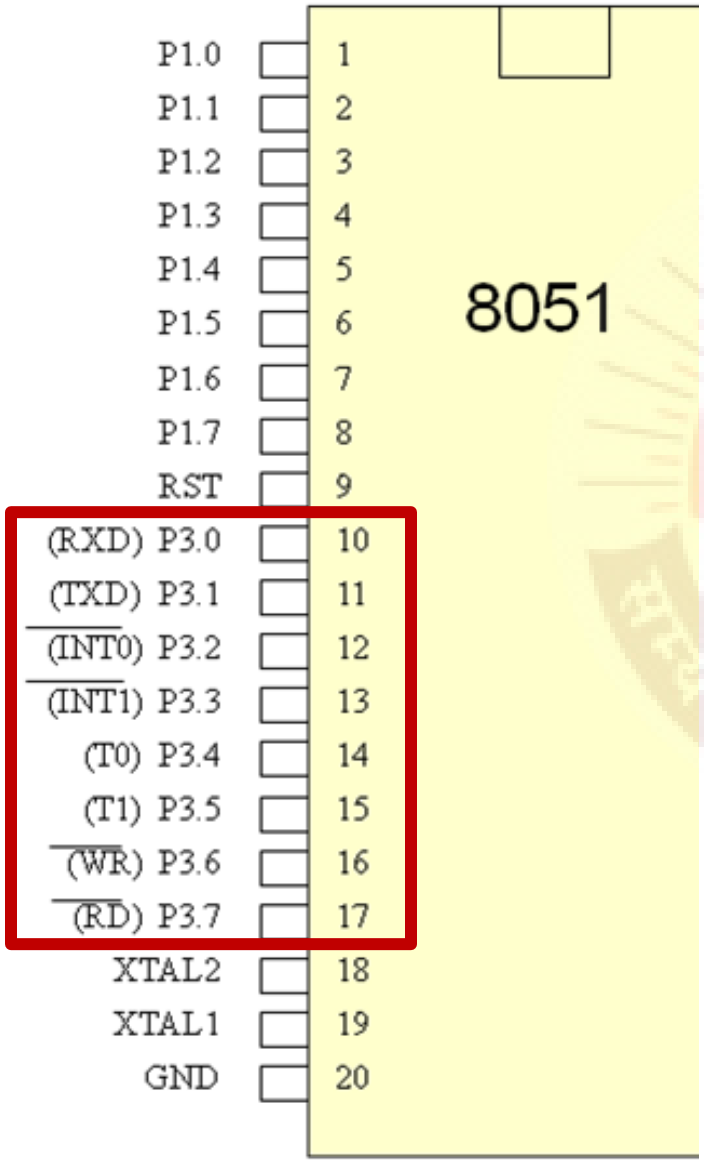
- 8 Bit bidirectional Port
- It has internal pull-up resistors
- Entire port (P1) or one of the pin (P1.0 – P1.7) configured as I/P or O/P

8051 Pin Diagram : **Port 2 (P2.0/A8 – P2.7/A15)**



- 8 Bit bidirectional Port
- It has internal pull-up resistors
- Entire port (P2) or one of the pin (P2.0 – P2.7) configured as I/P or O/P
- Port is used to carry high order address (A8 – A15)

8051 Pin Diagram : **Port 3 (P3.0 – P3.7)**

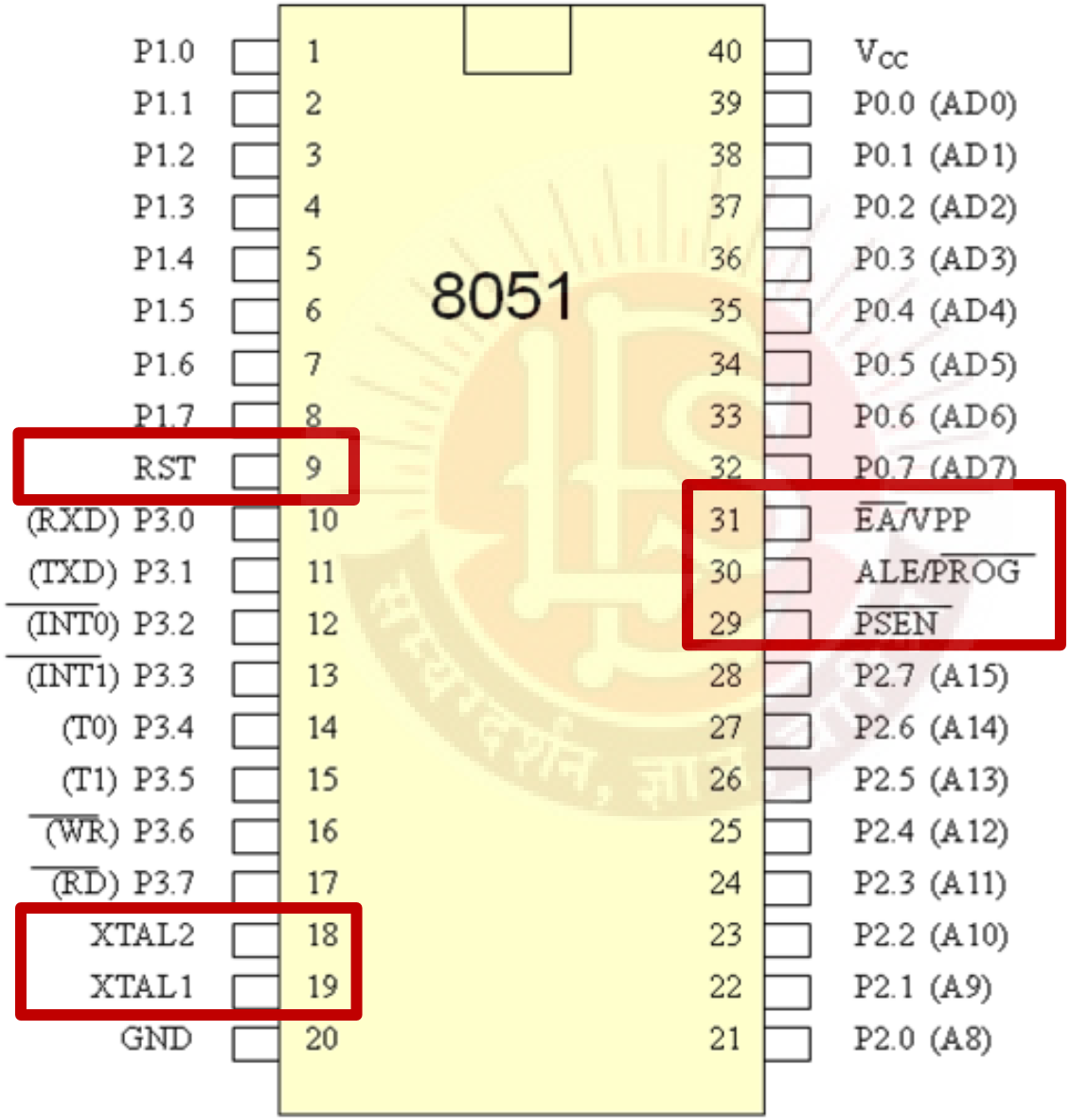


- 8 Bit bidirectional Port
- It has internal pull-up resistors
- Entire port (P3) or one of the pin (P3.0 – P3.7) configured as I/P or O/P
- Port is used to perform special features

8051 Pin Diagram : **Port 3 (Special Features)**

Pin	Name	Alternate Function
P3.0	RxD	Serial Input Line
P3.1	TxD	Serial Output Line
P3.2	INT0	External Interrupt 0
P3.3	INT1	External Interrupt 1
P3.4	T0	Timer 0 External Input
P3.5	T1	Timer 1 External Input
P3.6	WR	Write Control
P3.7	RD	Read Control

8051 Pin Diagram : **Other Pins**



RESET

- If we make RESET = 1 for 2 machine cycles then controller will be reset and it will execute the program at address **0000H**

PSEN (Program Store Enable)

- When 8051 executes program from internal memory it makes **PSEN = 1**
- When 8051 executes program from external memory it makes **PSEN = 0**

EA / Vpp

- **EA = 0**
  - 8051 executes program from external memory
  - 64 K (0000 H – FFFF H)
- **EA = 1 (+5 V)**
  - 8051 executes program from internal memory
  - 4K (0000 H – 03FF H)
- **EA = 1 (+12.75 V)**
  - Internal EEPROM is programmed

ALE / PROG

- **= 1**

  P0 = A0 – A7
  P2 = A8 – A15

- **= 0**

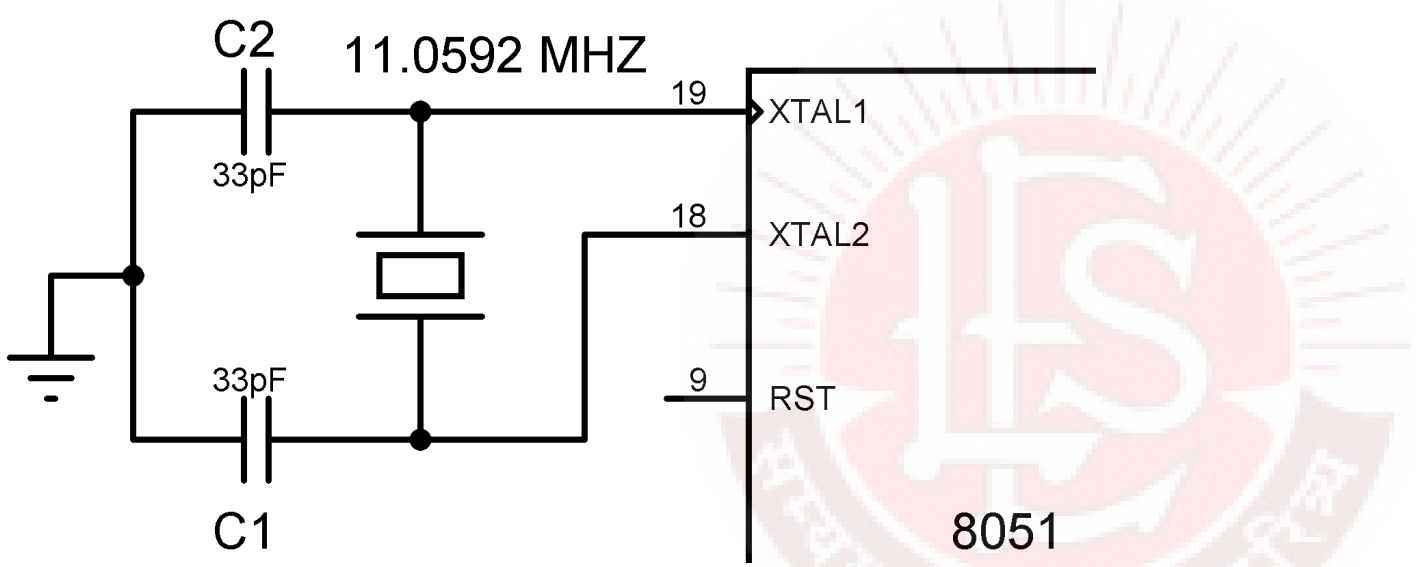
  P0 = D0 – D7

- **= 0**

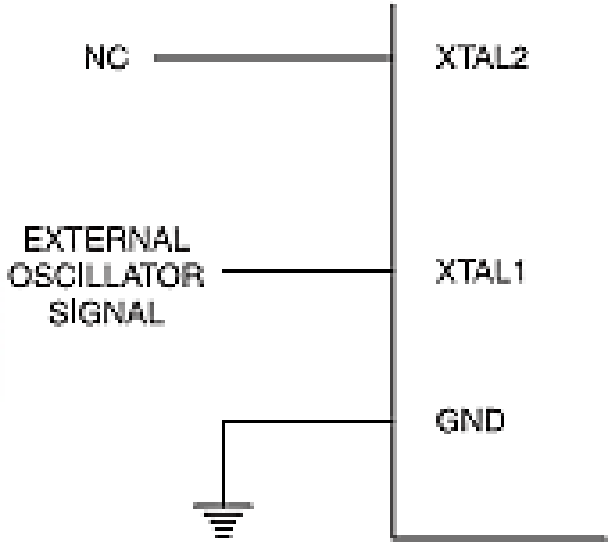
  Indicated Internal EEPROM is being programmed

XTAL1	XTAL2
• Input to the Internal Crystal Oscillator	• Output the Internal Crystal Oscillator

8051 Pin Diagram : **Oscillator**

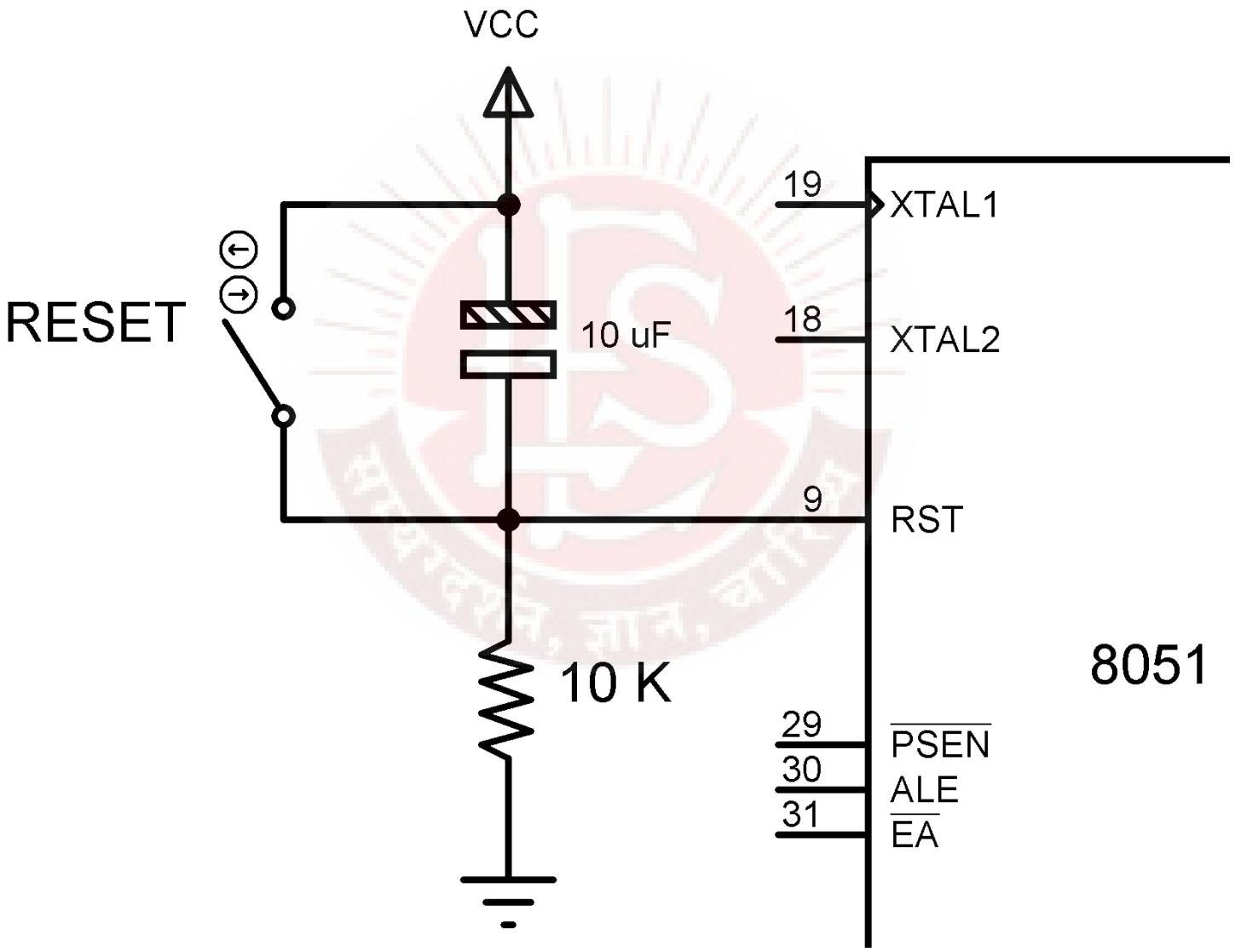


a) Internal Oscillator

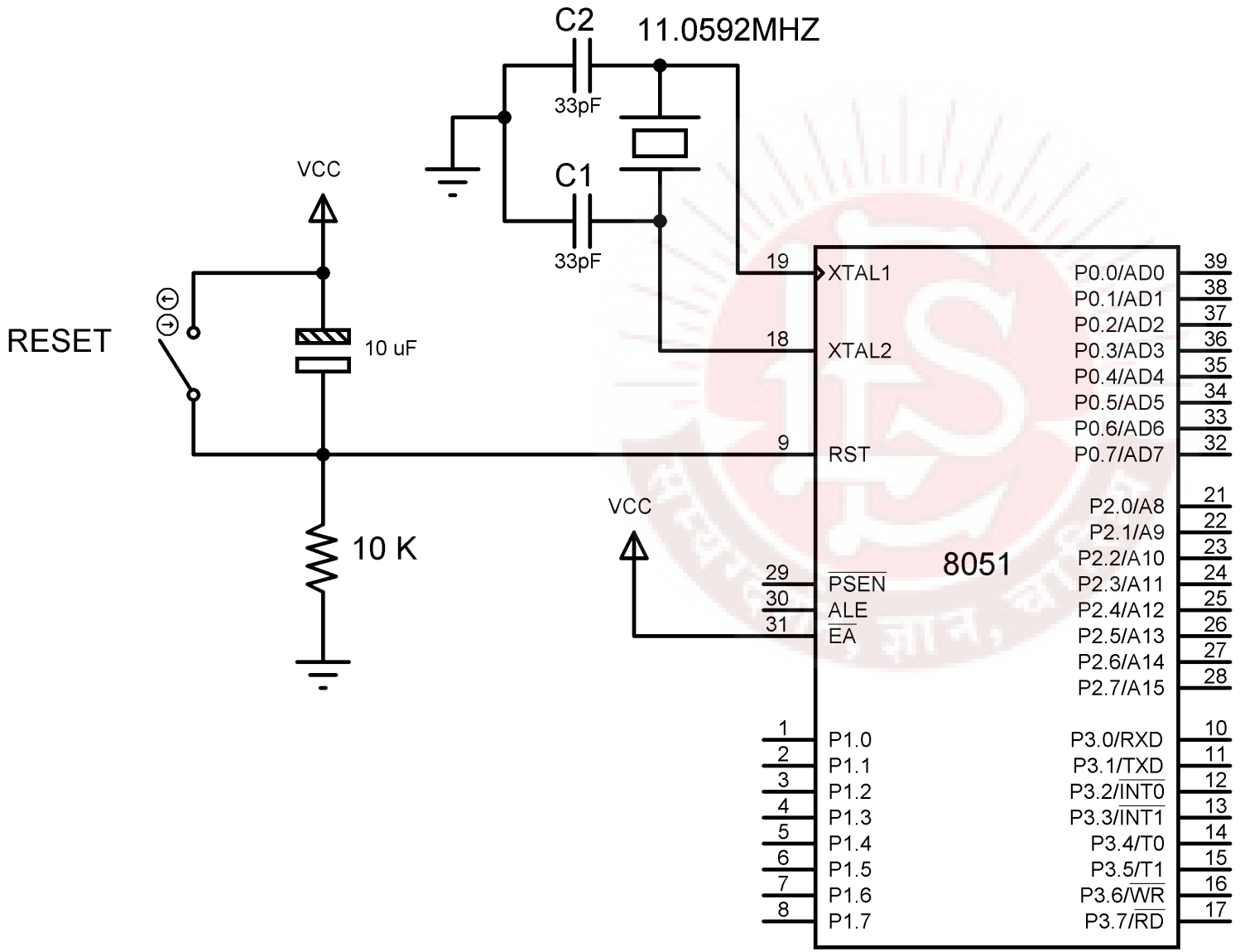


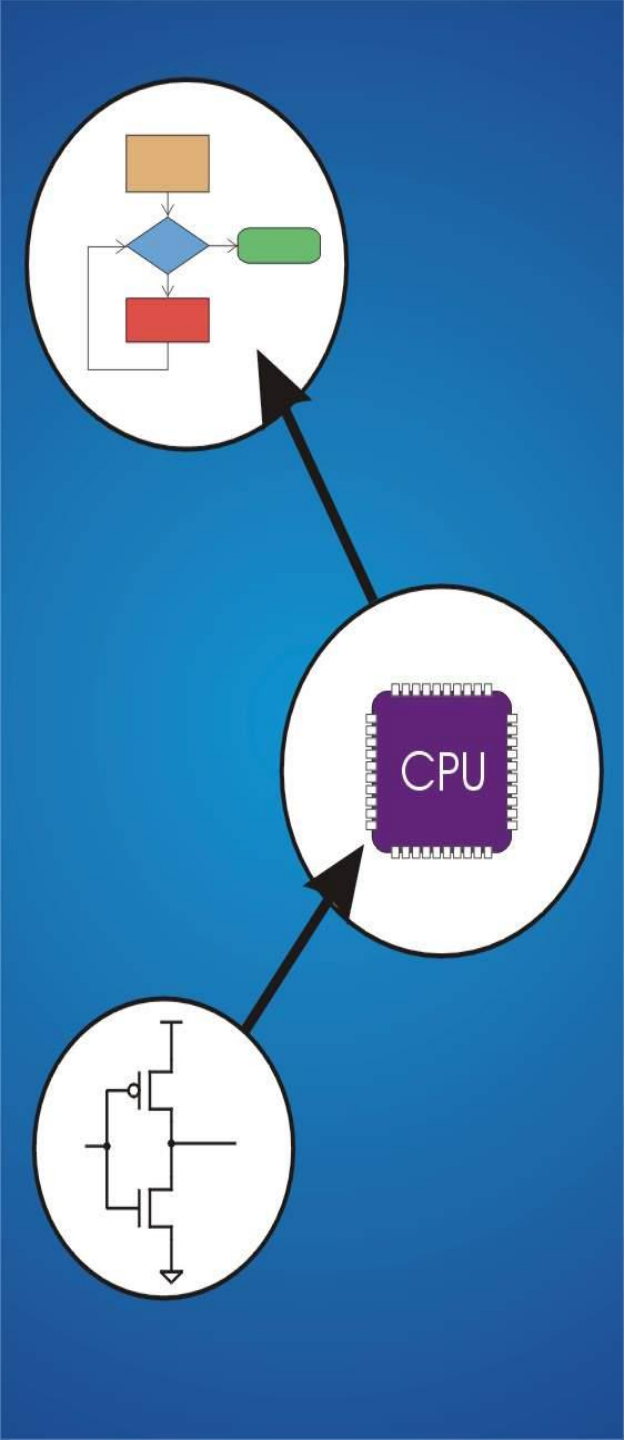
b) External Oscillator

8051 Pin Diagram : **Power on Reset**



Minimum Hardware Required to use 8051



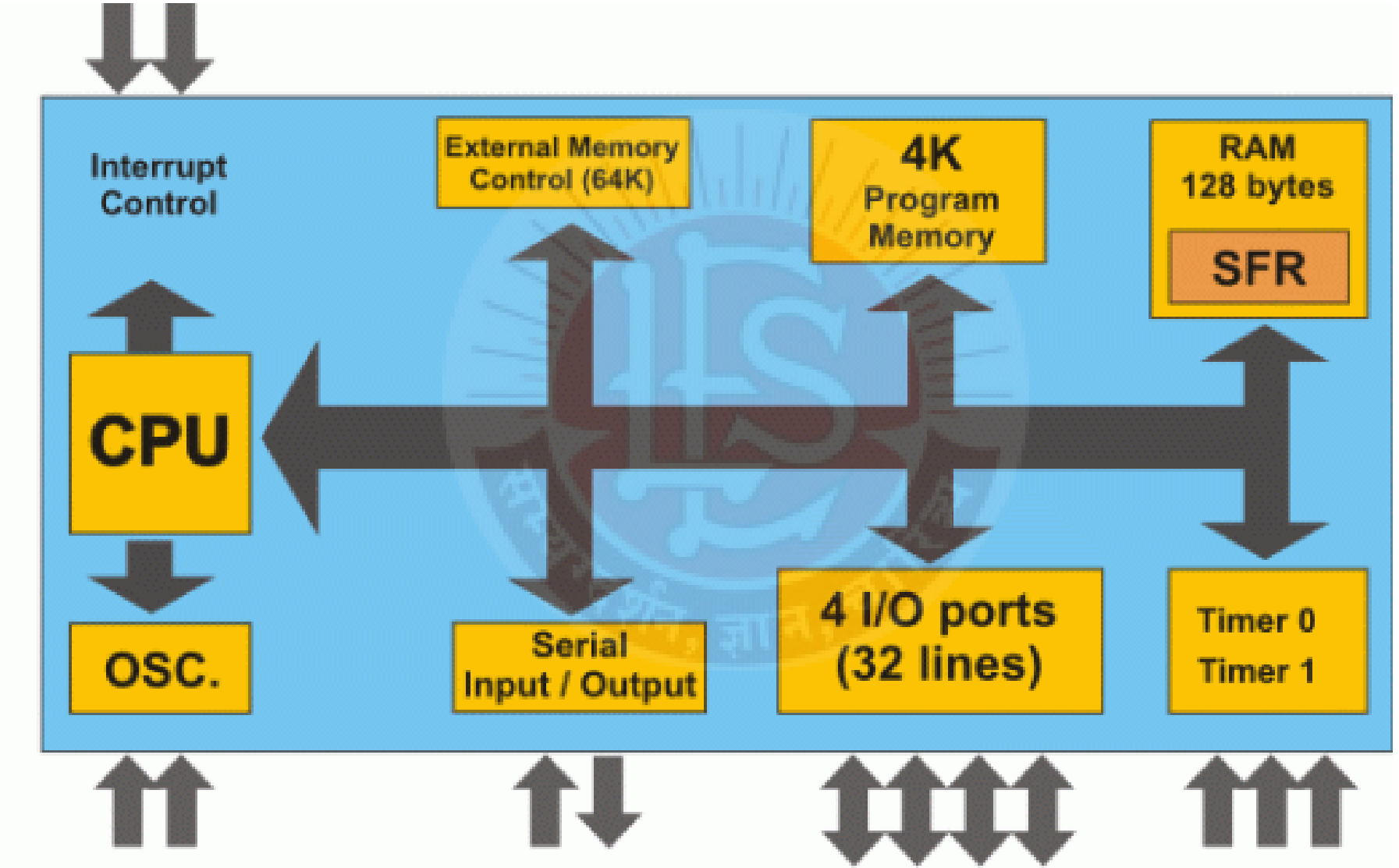


UNIT – I

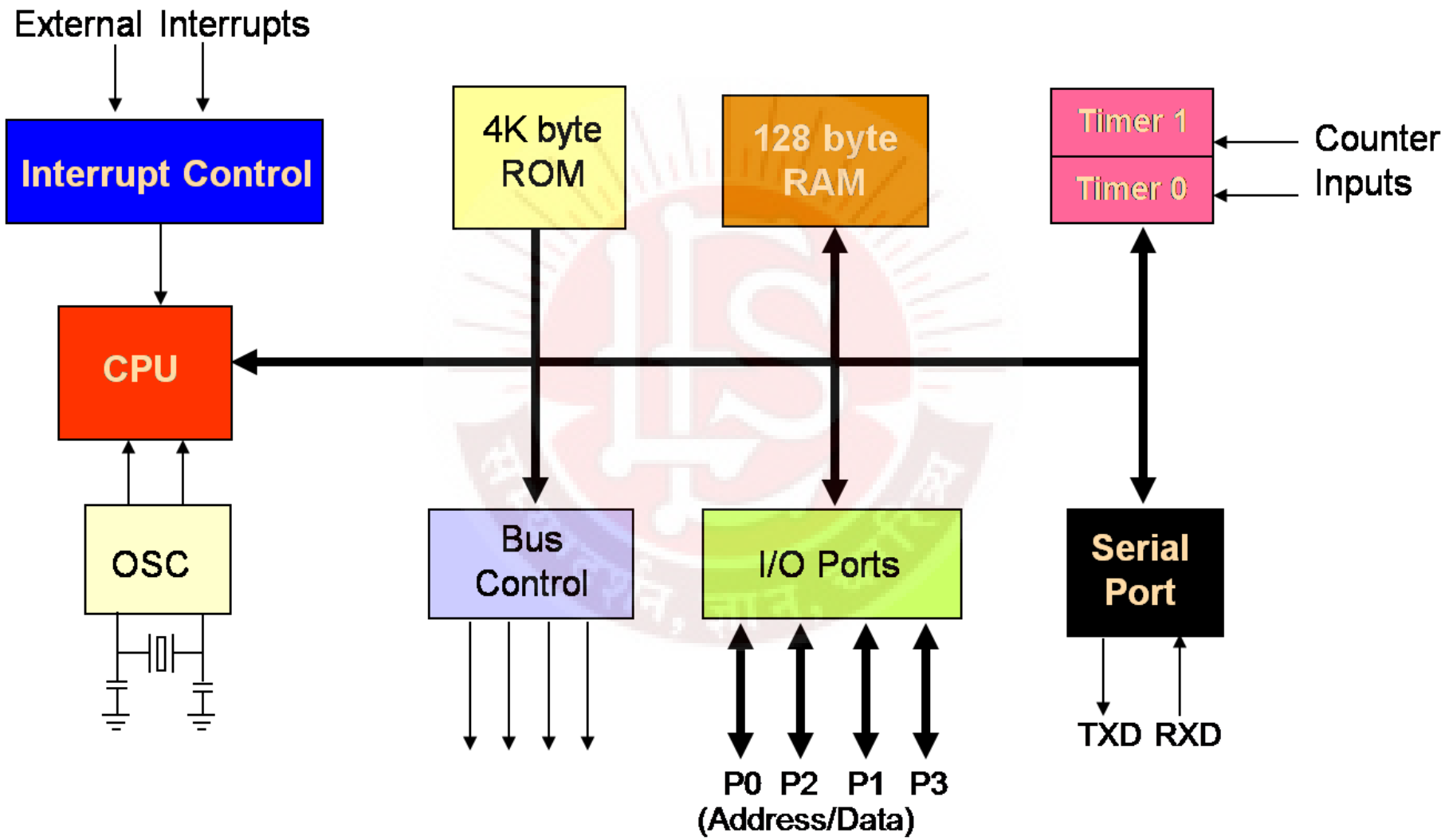
Introduction to Microcontroller

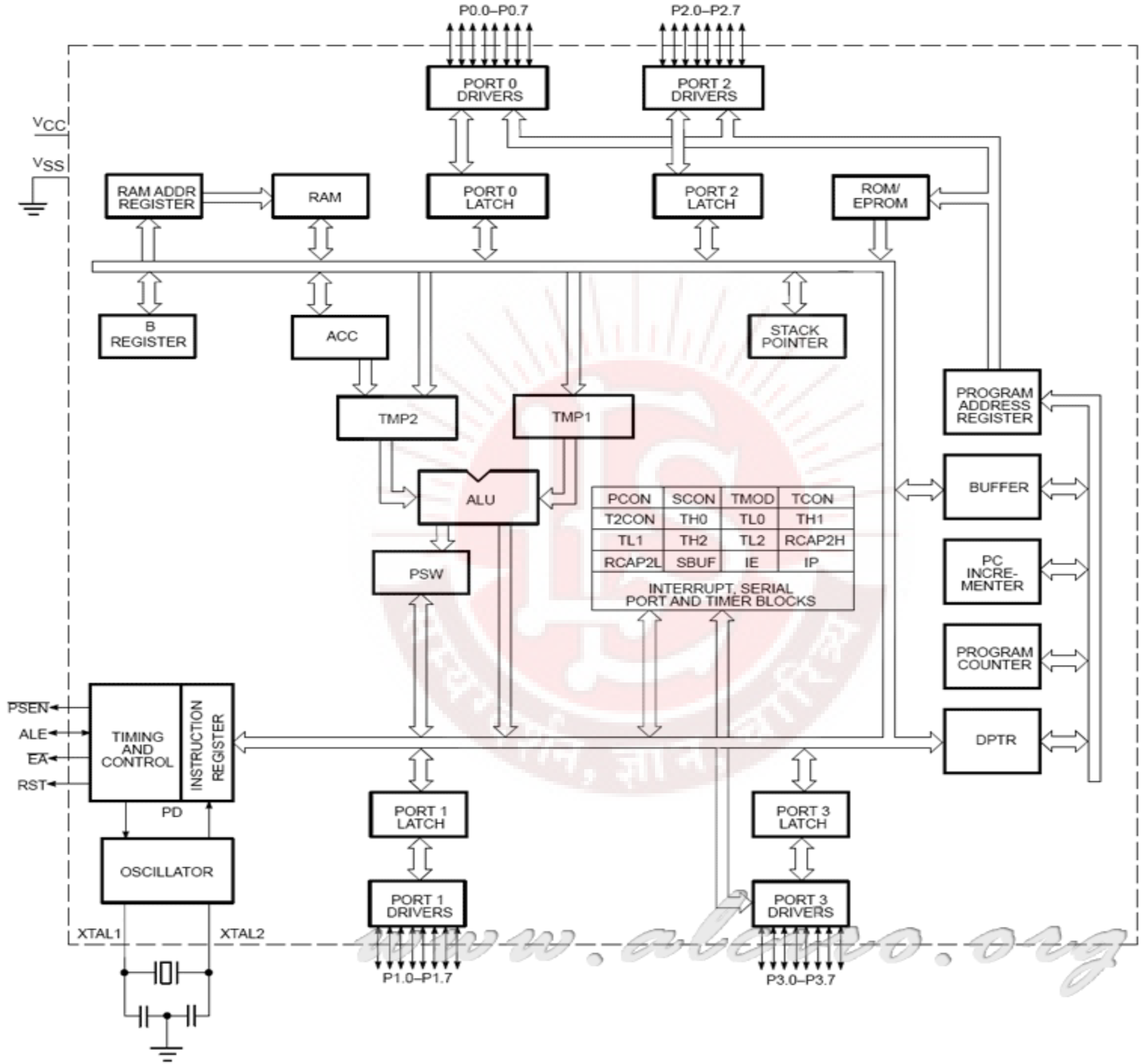
Architecture of 8051

8051 Architecture :

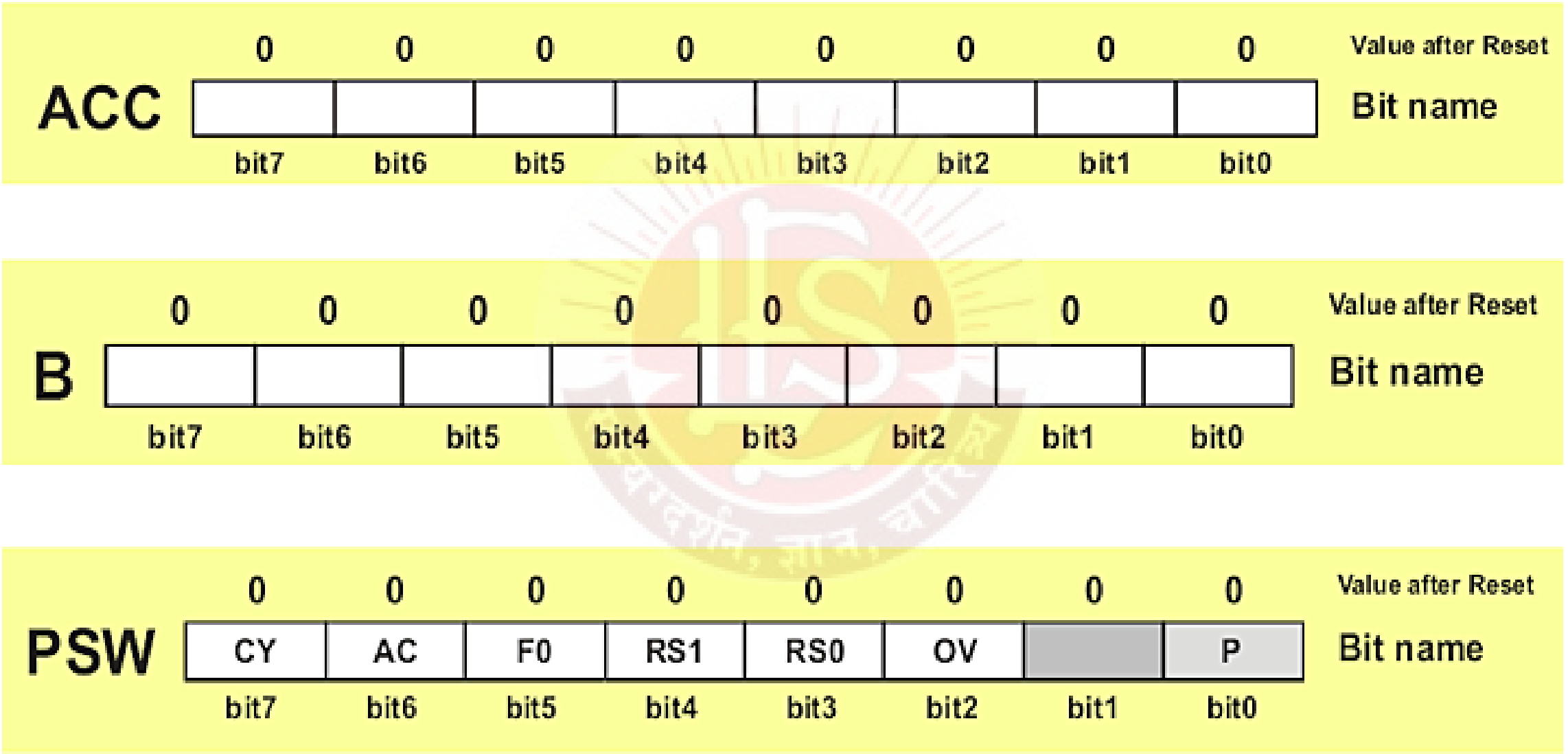


8051 Architecture :





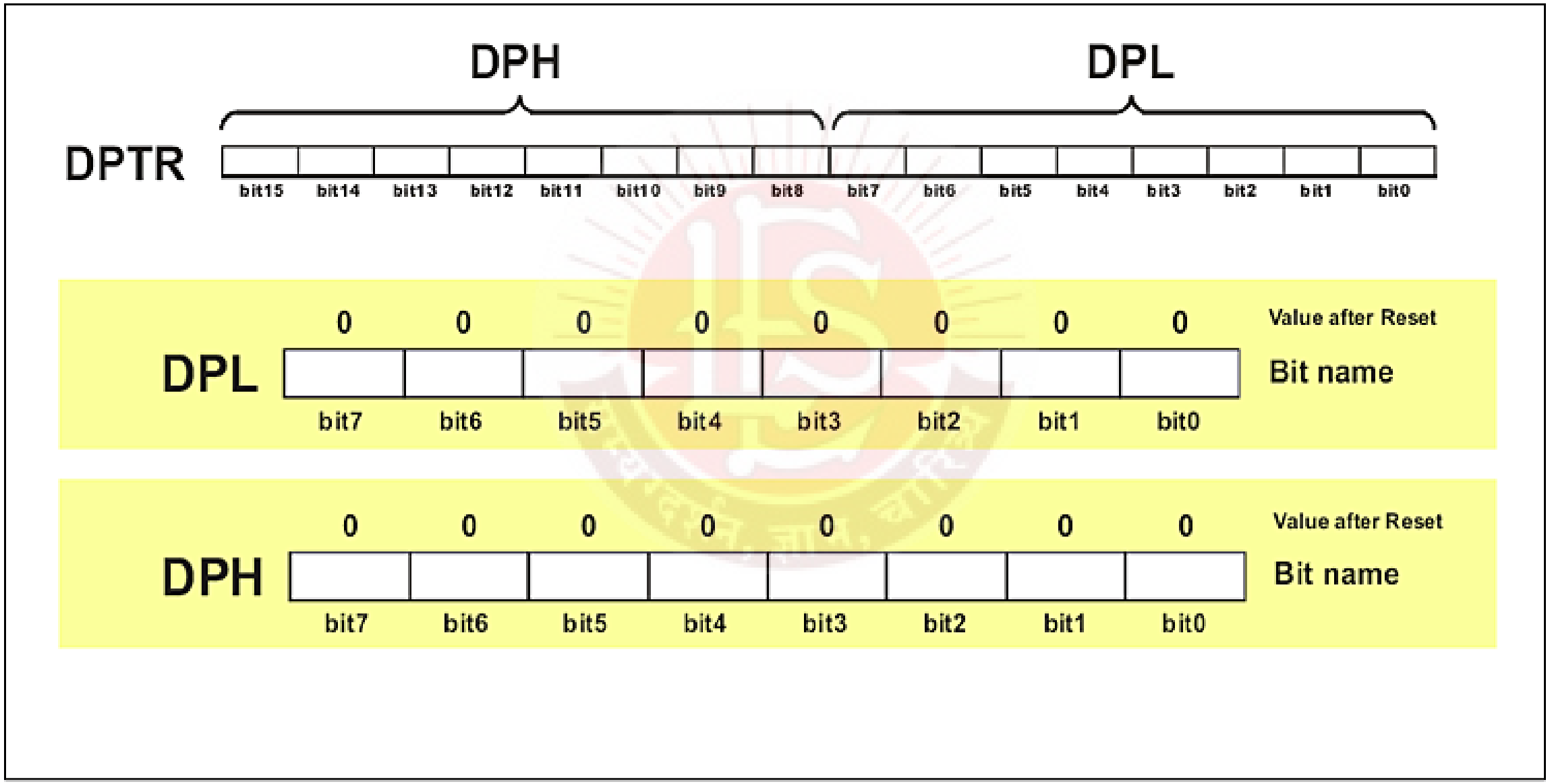
8051 Architecture : ALU



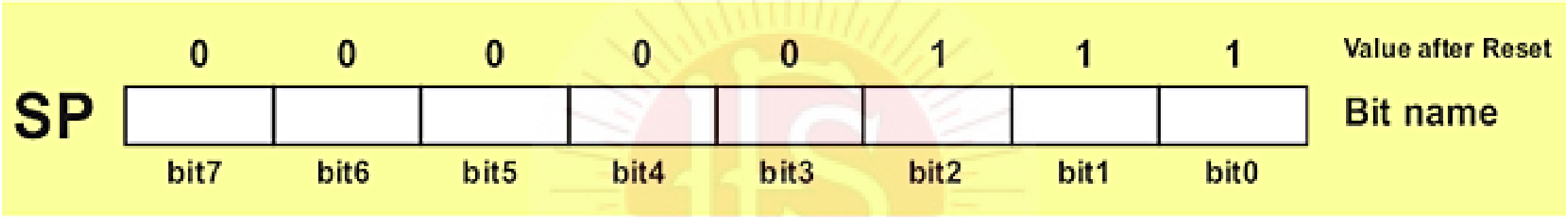
8051 Architecture : Register Bank Select Bits

RS1	RS0	Bank	Address
0	0	Bank 0	00 - 07H
0	1	Bank 1	08 – 0F H
1	0	Bank 2	10 – 17H
1	1	Bank 3	18 – 1FH

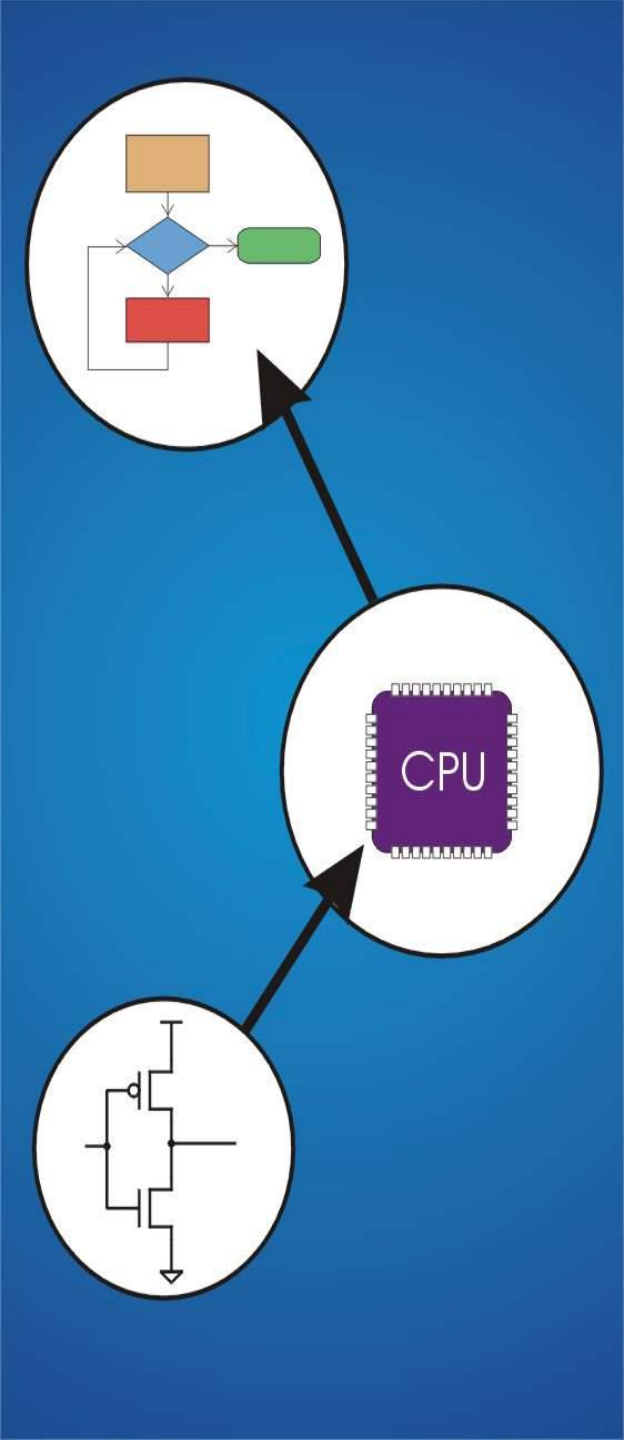
8051 Architecture : Data Pointer



8051 Architecture : ALU : Stack Pointer



8051 Architecture : Register Banks

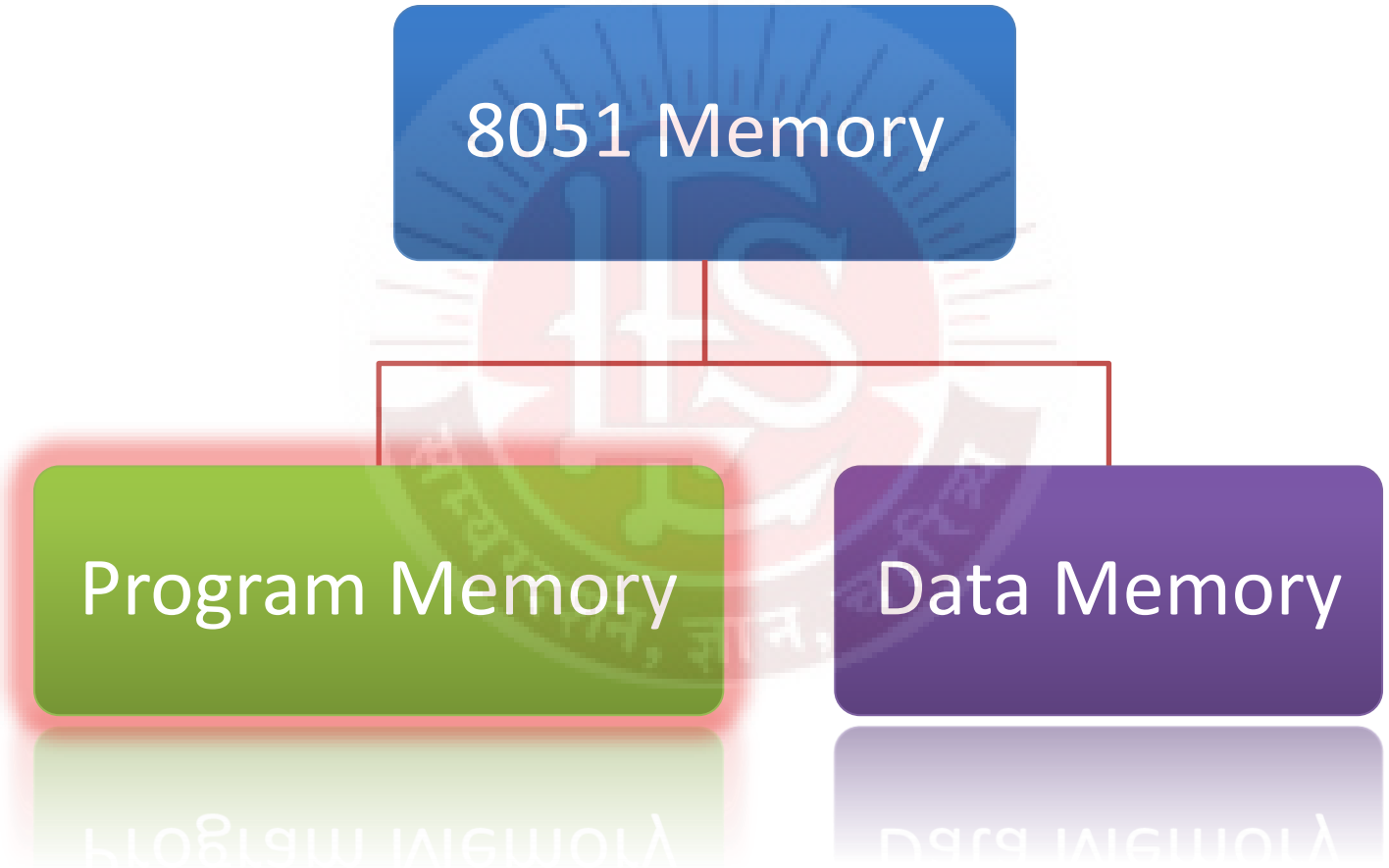


UNIT – I

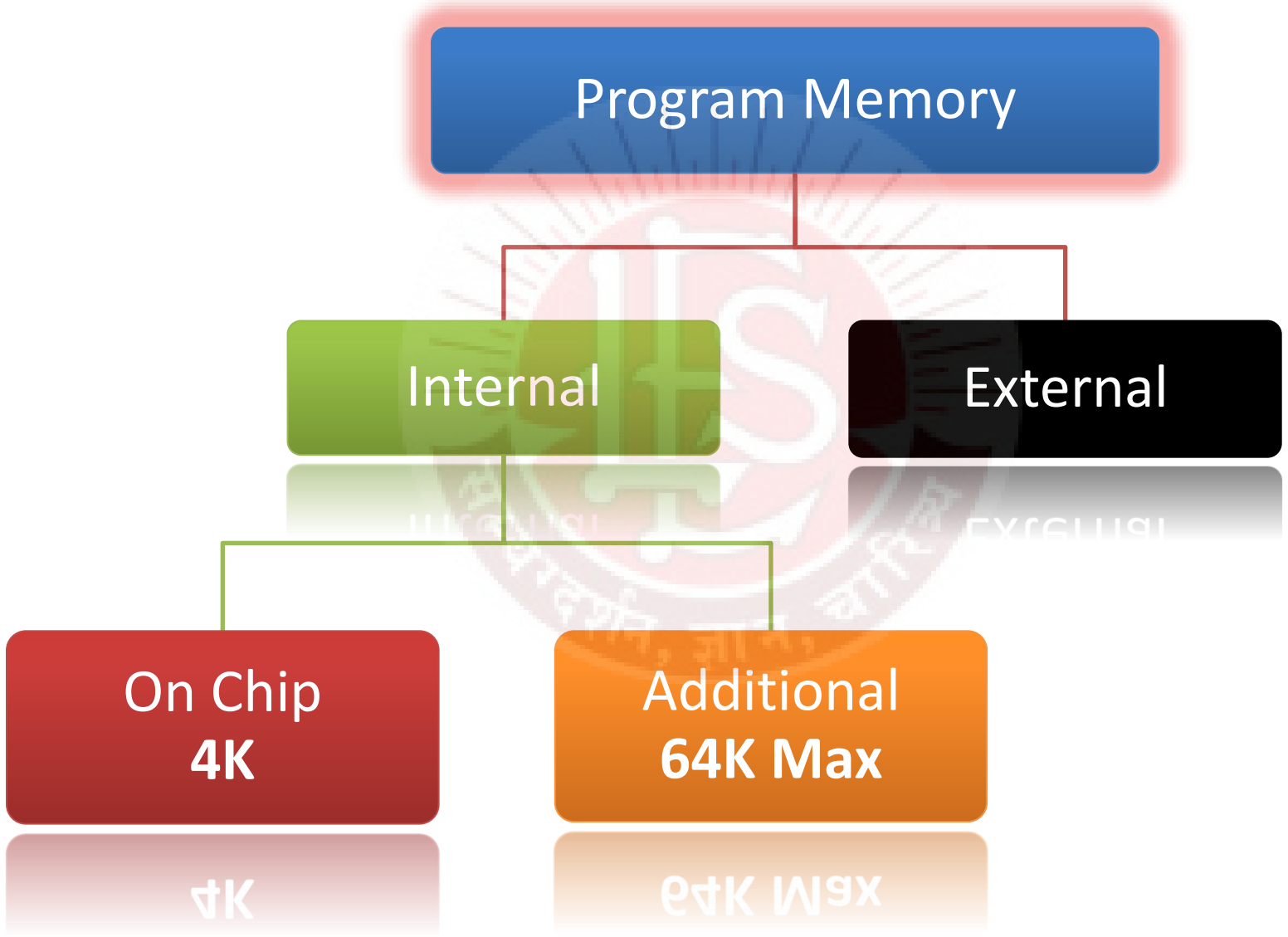
Introduction to Microcontroller

Memory Organization

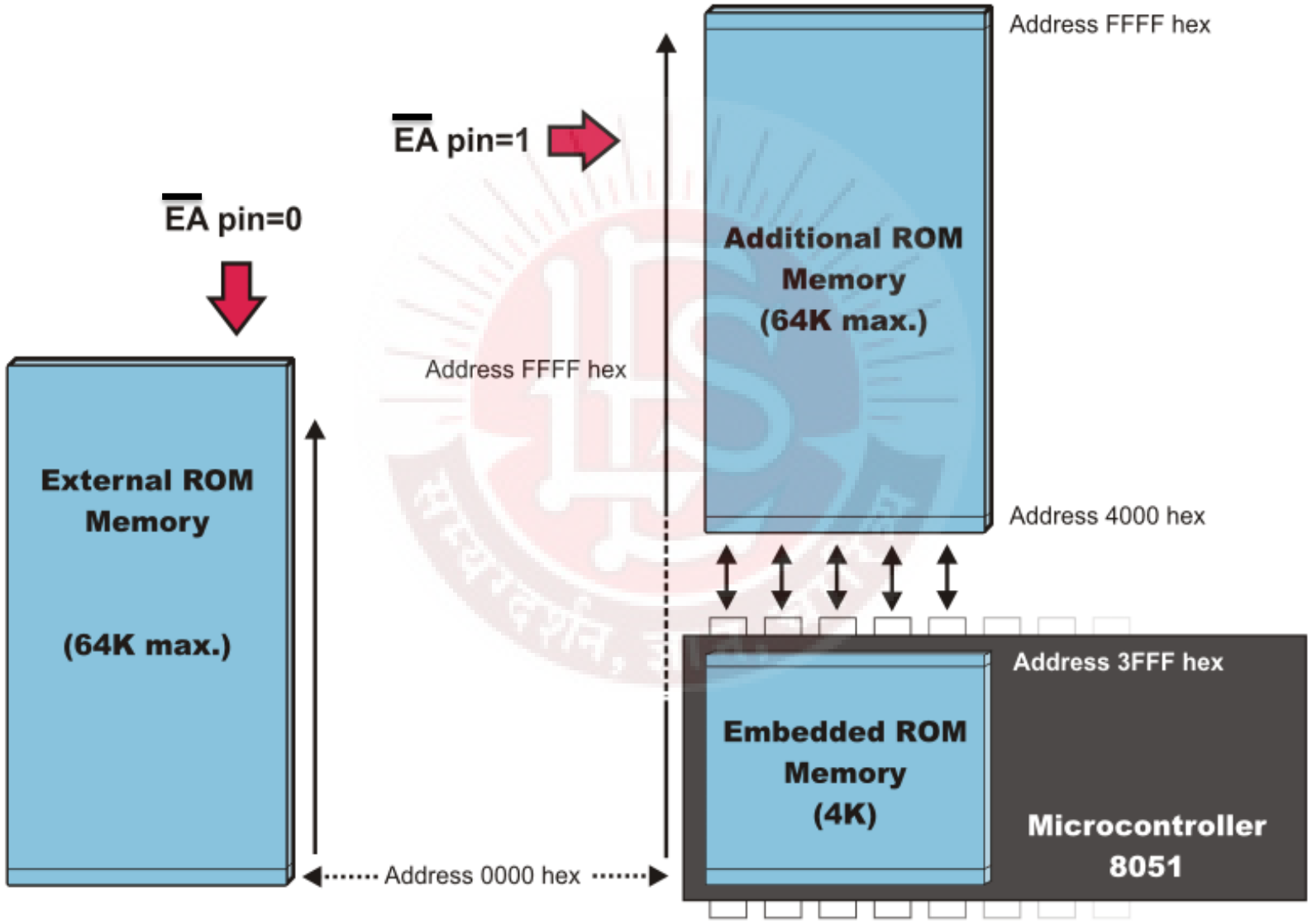
8051 Memory Organization :



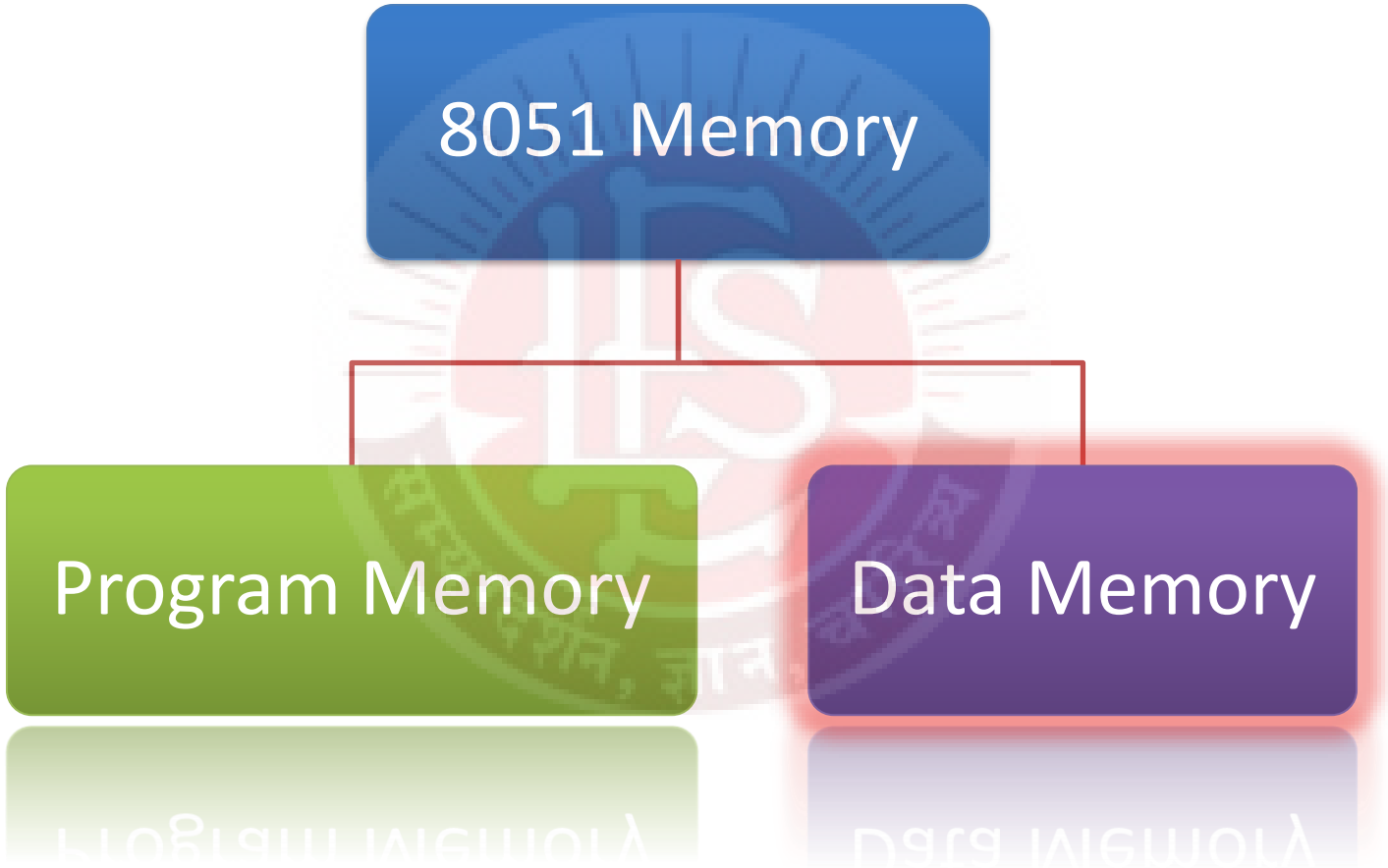
8051 Program Memory :



8051 Program Memory :

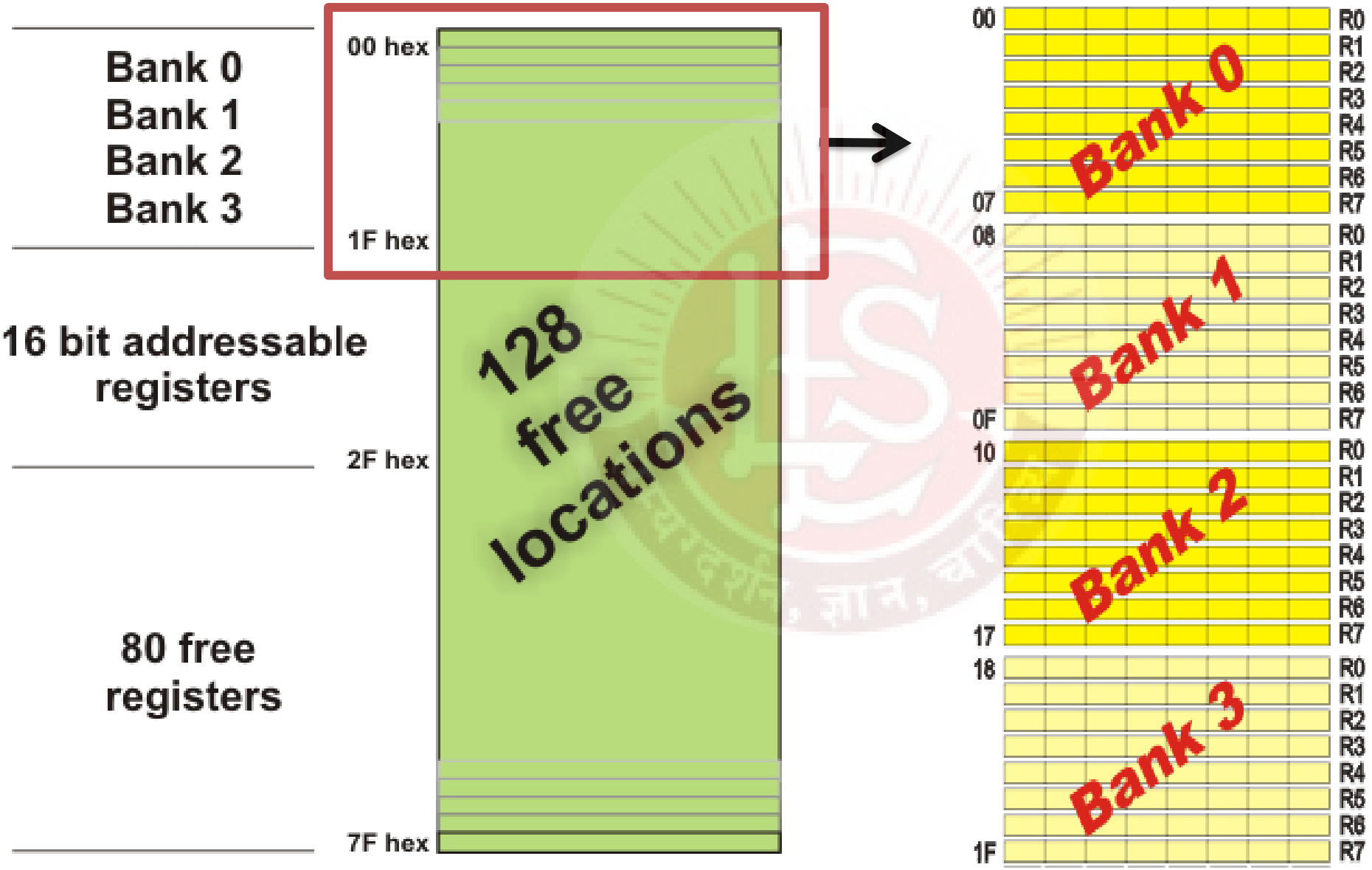


8051 Data Memory :

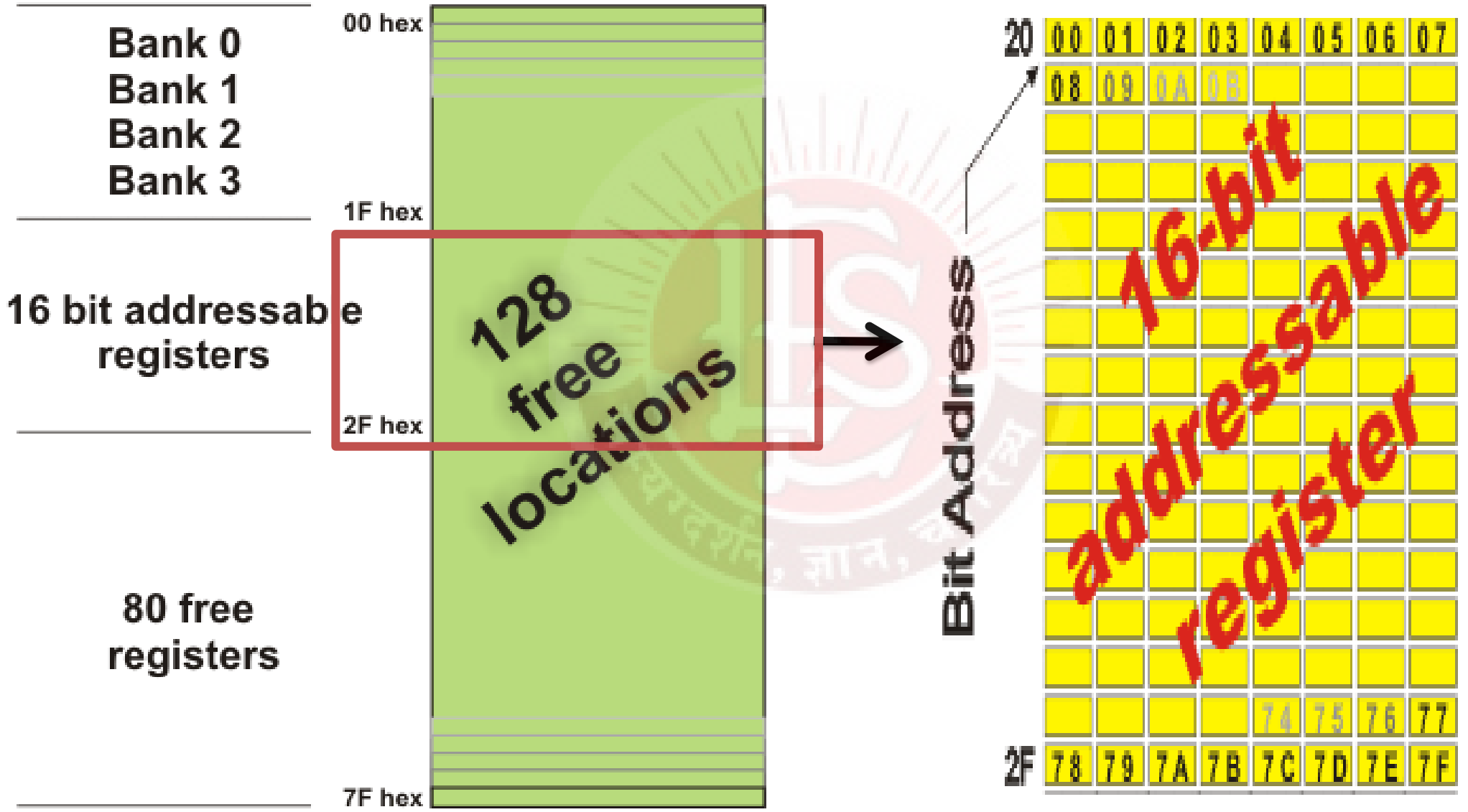


8051 Data Memory : 256 Bytes RAM and SFR

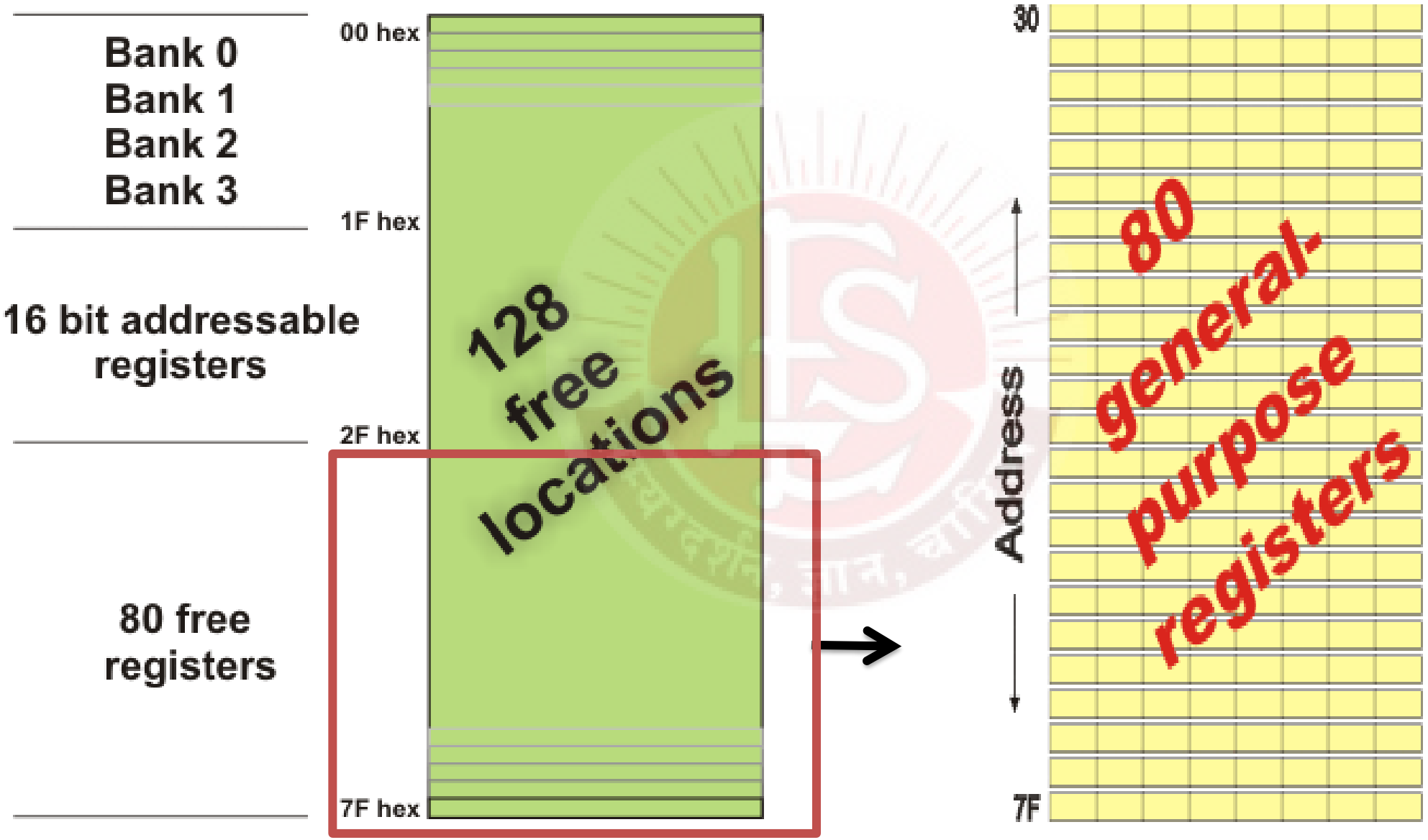
8051 Data Memory : Bank Section



8051 Memory Organization : 16 Bit Addressable Registers



8051 Memory Organization : 80 General Purpose Registers



8051 Data Memory : Total Registers

- **Byte Addressable**
  - ✓ Bank 4 x 8 32
  - ✓ General Purpose 80
- **Bit Addressable** 16 x 8 128

8051 Data Memory : SFR's

8051 Memory Organization : Special Function Registers (SFR)

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8								CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH			PCON	87

Bit Addressable Registers

Course Contents

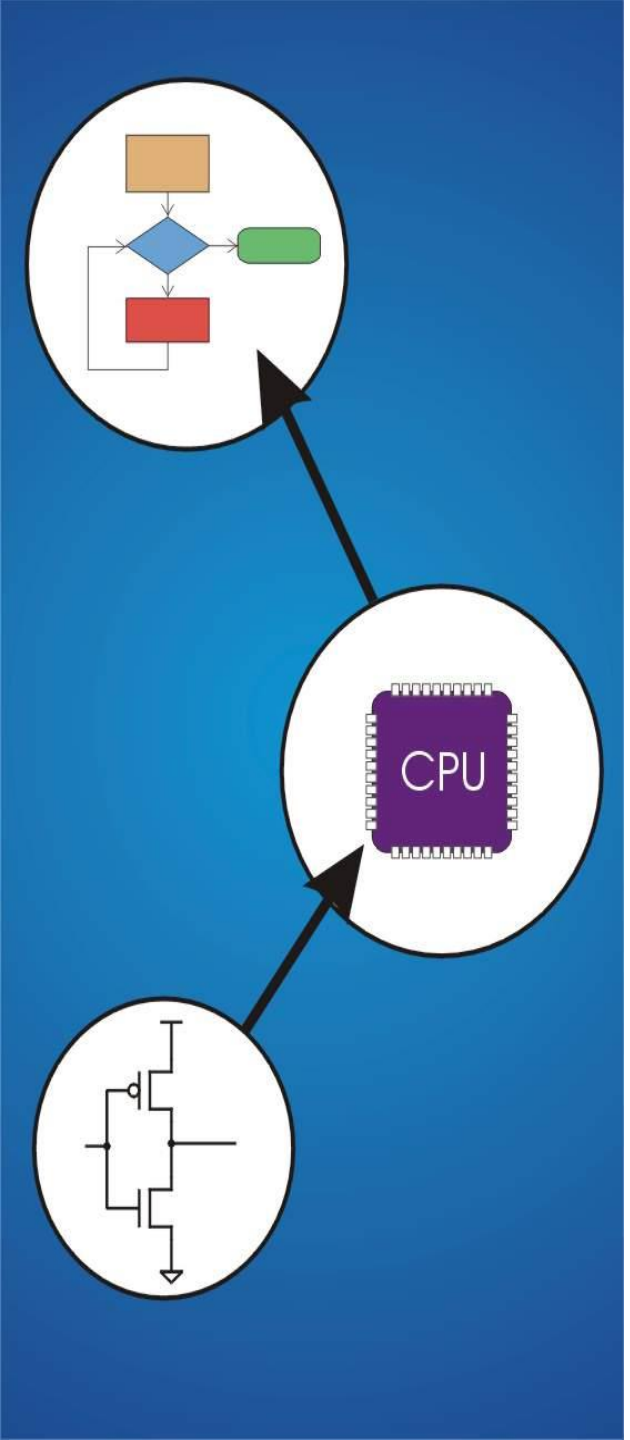
1. Introduction to Microcontroller
2. 8051 Instruction Set
3. Facilities in 8051
4. Interfacing Methods

Course Contents

8051 INSTRUCTION SET

Study of 8051 Instruction Set and Addressing Modes,

Data transfer,
Arithmetic,
Logical,
JUMP,
Loops & CALL instructions,
Bit manipulation Instructions.



UNIT – II

8051 Instruction Set

Prerequisite

Notations used in Instruction

Notations used in Instruction :

Rn **n = 0 to 7**

R0, R1, R2, R3, R4, R5, R6, R7

@Ri **i = 0, 1**

@R0

@R1

Notations used in Instruction :

#8bit Data

```
MOV   R0,  #00H

MOV   R0,  #0F0H
```

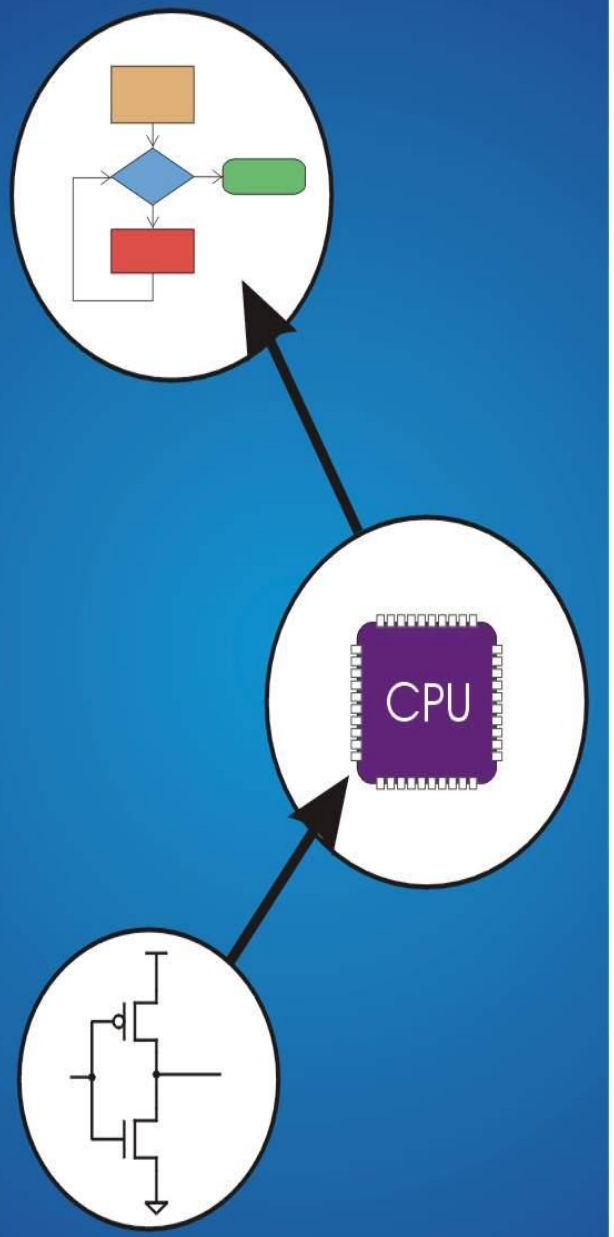
#16 bit Data

```
MOV   DPTR,  #1234H

MOV   DPTR,  #0F000H
```

Notations used in Instruction :

Direct

MOV A, 80H



UNIT – II

8051 Instruction Set

Addressing Modes

Addressing Modes :

Opcode Operands

MOV A, R0

MOV 0E0H, 00H

Various ways to specify the address of the operands

Addressing Modes : Immediate

```
MOV     A,     #02H

MOV     DPTR,  #0F000H
```

Addressing Modes : Direct

MOV A, 80 H

MOV 80 H, A

Addressing Modes : Indirect

MOV A, @R0

MOV A, @DPTR

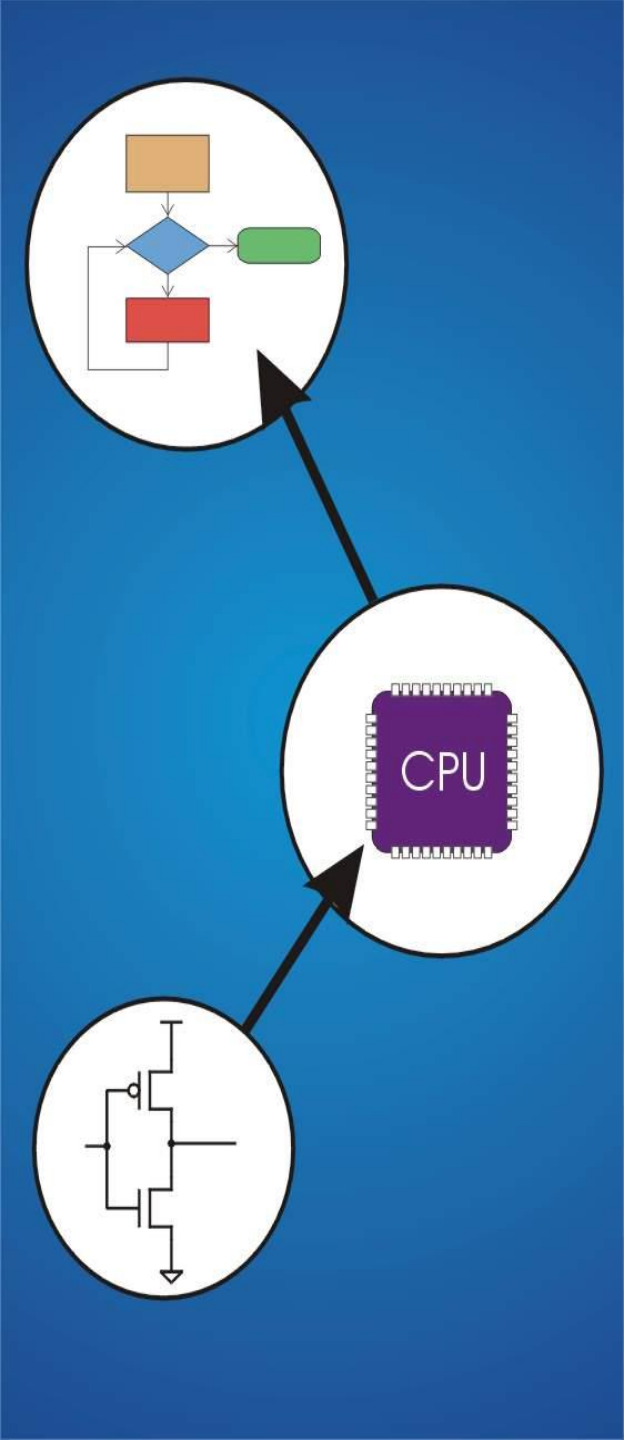
Addressing Modes : Register

MOV A, R0

Addressing Modes : External Indirect

MOVX A, @DPTR

MOVX @DPTR, A



UNIT – II

8051 Instruction Set

Instruction Set

Instruction :

Opcode Operands

Opcode : Operation to be performed

Operands : Data on which operation is carried out

MOV A, R0

A = E0H

R0 = 00H

Instruction Set :

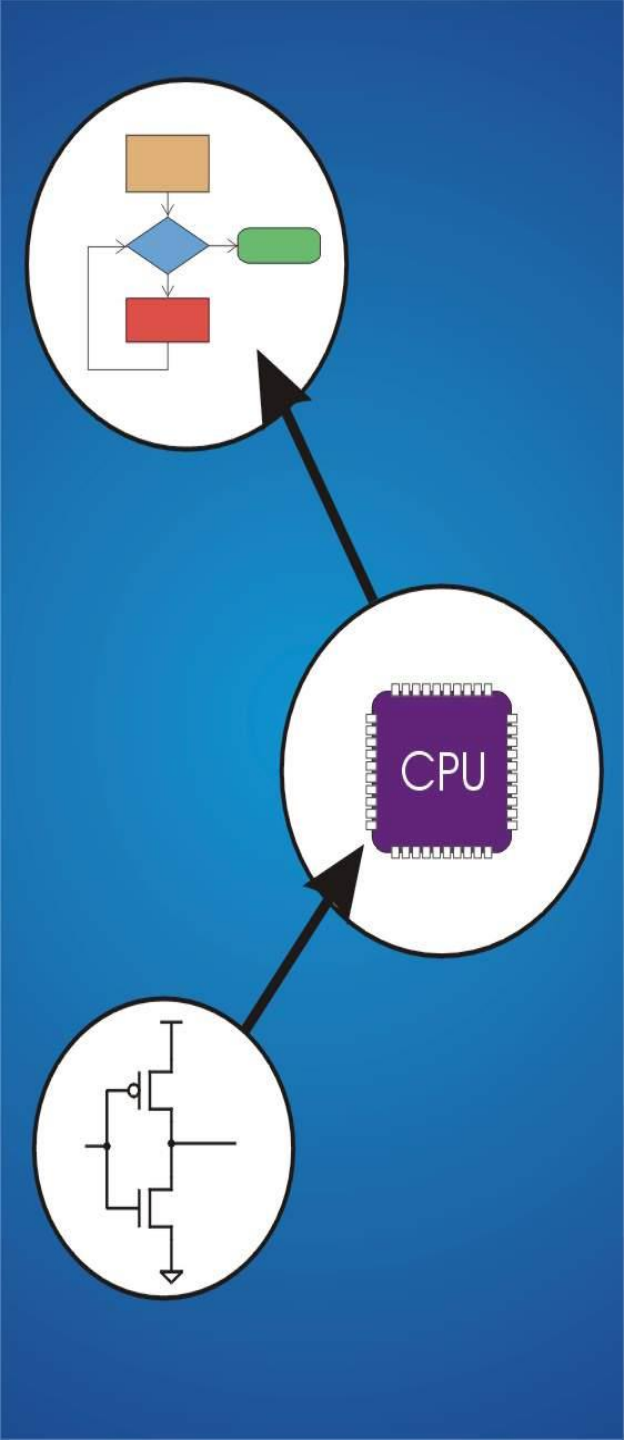
Arithmetic Instructions

Data Transfer

Logical Instructions

Logical Instructions with bits

Branch Instructions



UNIT – II

8051 Instruction Set

Instruction Set

Arithmetic Instructions

Instruction Set : Arithmetic Instructions

ADD A, <Source Byte>

ADD A, #03 H

ADD A, R0

ADD A, @R0

ADD A, 80 H

ADDC A, <Source Byte>

ADDC A, #03 H

ADDC A, R0

ADDC A, @R0

ADDC A, 80 H

Instruction Set : Arithmetic Instructions

SUBB A, <Source Byte>

SUBB A, #03 H

SUBB A, R0

SUBB A, @R0

SUBB A, 80 H

Instruction Set : Arithmetic Instructions

MUL AB

B **A**

MSB LSB

DIV AB

B **A**

Remainder Quotient

Instruction Set : Arithmetic Instructions

INC <Byte>

~~INC #03H~~

INC R0

INC @R0

INC 80 H

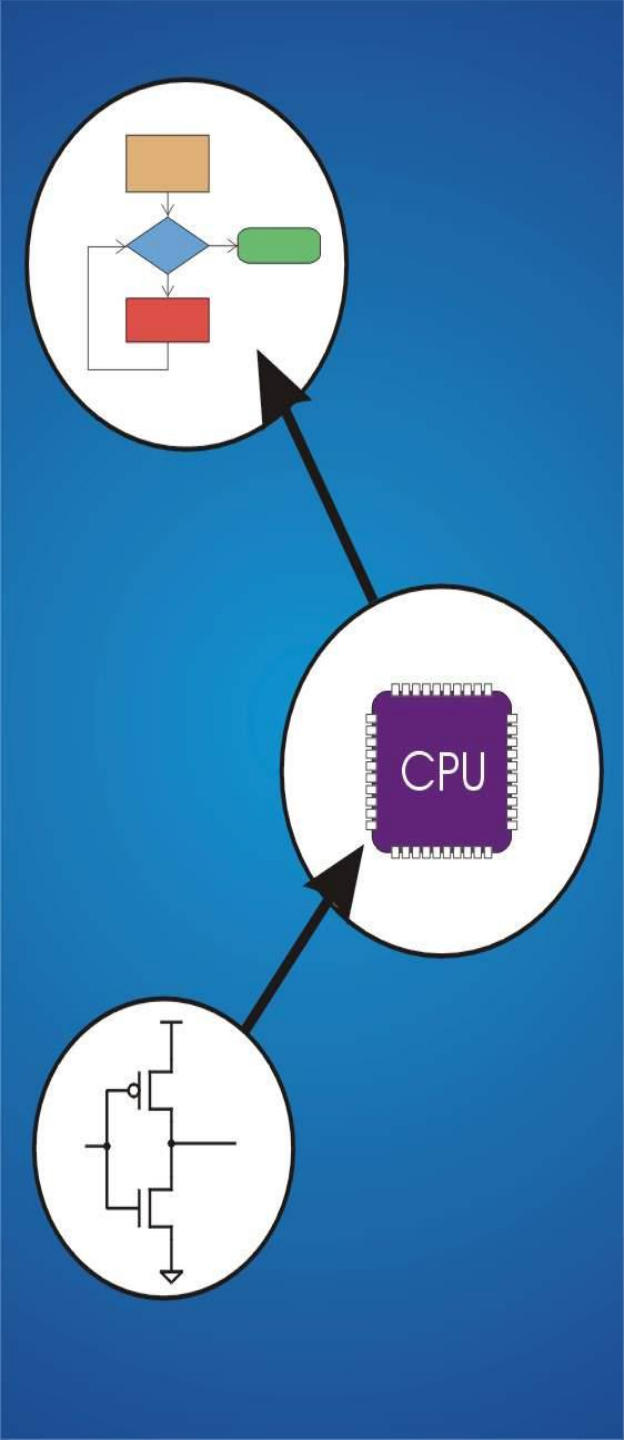
INC DPTR

DEC <Byte>

DEC R0

DEC @R0

DEC 80 H



UNIT – II

8051 Instruction Set

Instruction Set

Data Transfer Instructions

MOV <destination byte>, <Source Byte>

	#05 H
R0	R0
@R0	@R0
80H	80H

Instruction Set : Data Transfer Instructions

MOV <destination byte>, <Source Byte>

MOV R0, #05 H

MOV @R0, #05 H

MOV R0, @R1

MOV DPTR, #16 bit data

MOV DPTR, #1234 H

MOVC A, @A + <base register>

MOVC A, @A+PC

MOVC A, @A+DPTR

MOVC A, @DPTR

Instruction Set : Data Transfer Instructions

MOVX <destination byte>, <Source Byte>

MOVX A, @DPTR

Instruction Set : Data Transfer Instructions

PUSH <Address>

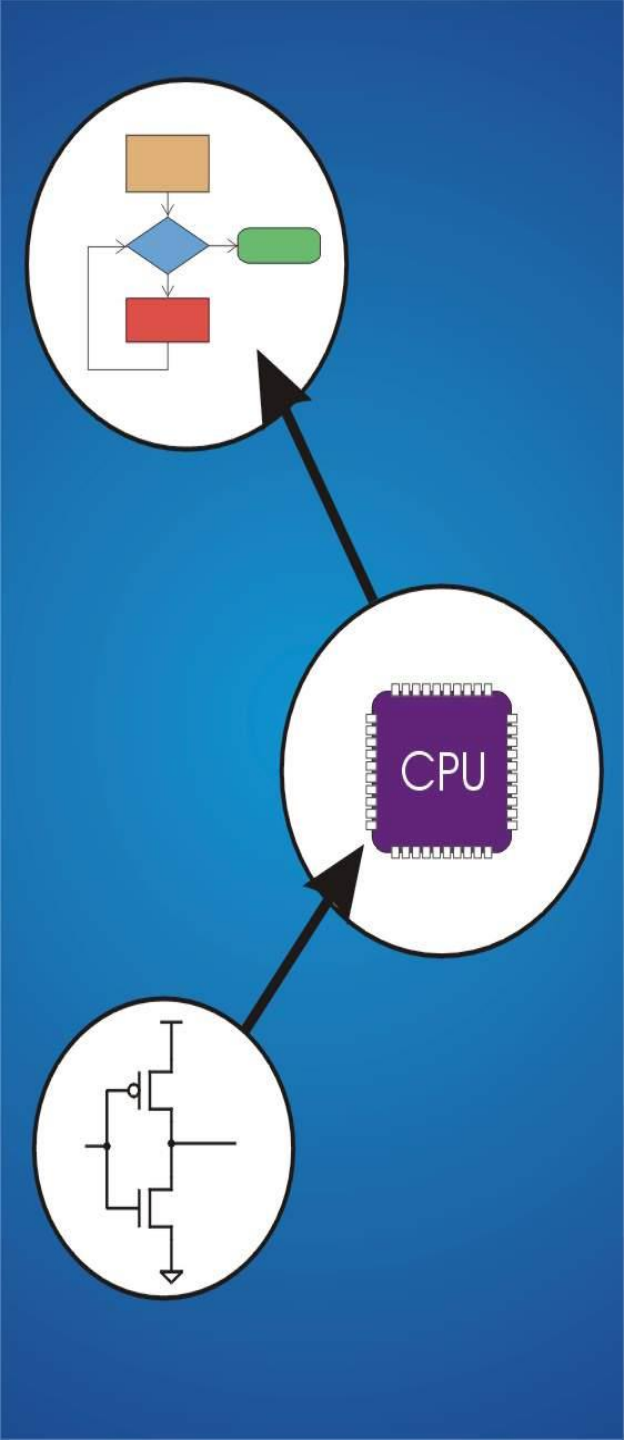
PUSH 80H

PUSH R0

POP <Address>

POP 80H

POP R0



UNIT – II

8051 Instruction Set

Instruction Set

Logical Instructions

Instruction Set : Logical Instructions

ANL <Destination Byte>, <Source Byte>

ANL A, #03 H

ANL A, R0

ANL A, @R0

ANL A, 80 H

A (2F)	0 0 1 0 1 1 1 1
03H	0 0 1 1 0 0 1 1

Instruction Set : Logical Instructions

ORL <Destination Byte>, <Source Byte>

ORL A, #03 H

ORL A, R0

ORL A, @R0

ORL A, 80 H

A (2F)	0 0 1 0 1 1 1 1
03H	0 0 1 1 0 0 1 1

Instruction Set : Logical Instructions

XRL <Destination Byte>, <Source Byte>

XRL A, #03 H

XRL A, R0

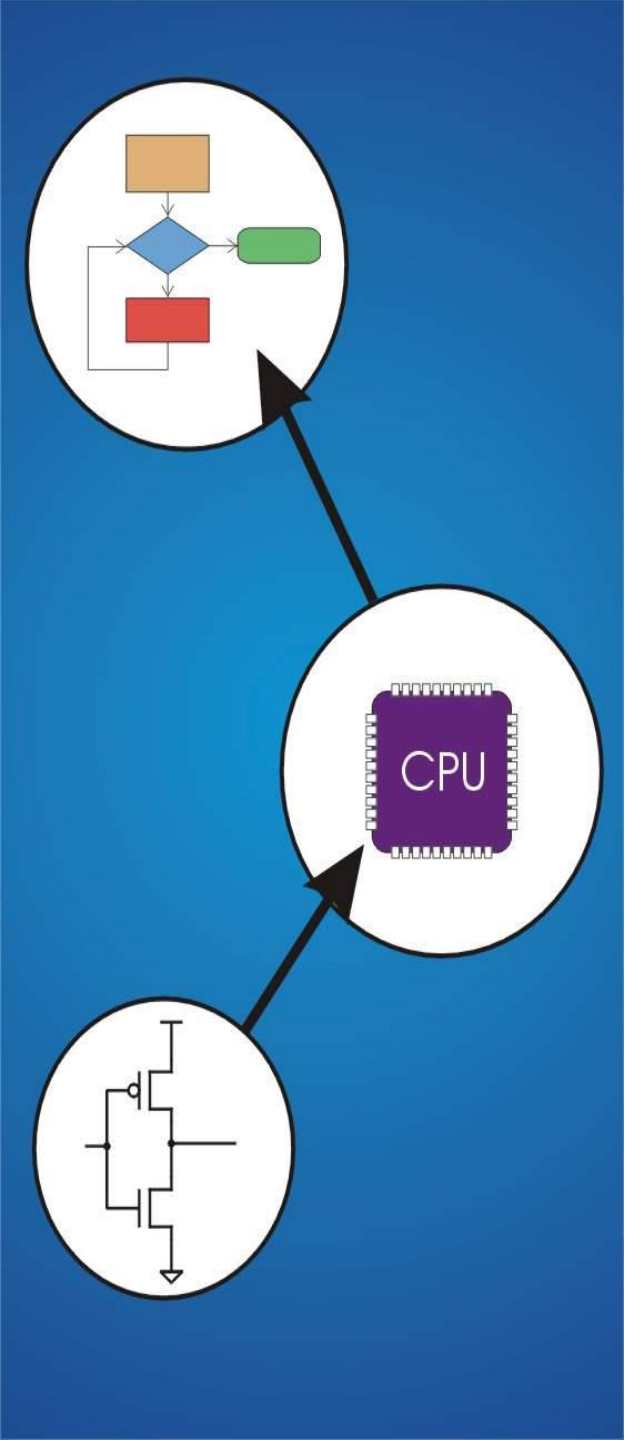
XRL A, @R0

XRL A, 80 H

A (2F)	0 0 1 0 1 1 1 1
03H	0 0 1 1 0 0 1 1

Instruction Set : Logical Instructions

```
CLR   A        ; Clear the Accumulator
CPL   A        ; Complement the Accumulator
RL    A        ; Rotate Accumulator to Left
RLC   A        ; Rotate Accumulator to Left through Carry Flag
RR    A        ; Rotate Accumulator to Right
RRC   A        ; Rotate Accumulator to Right through Carry Flag
```



UNIT – II

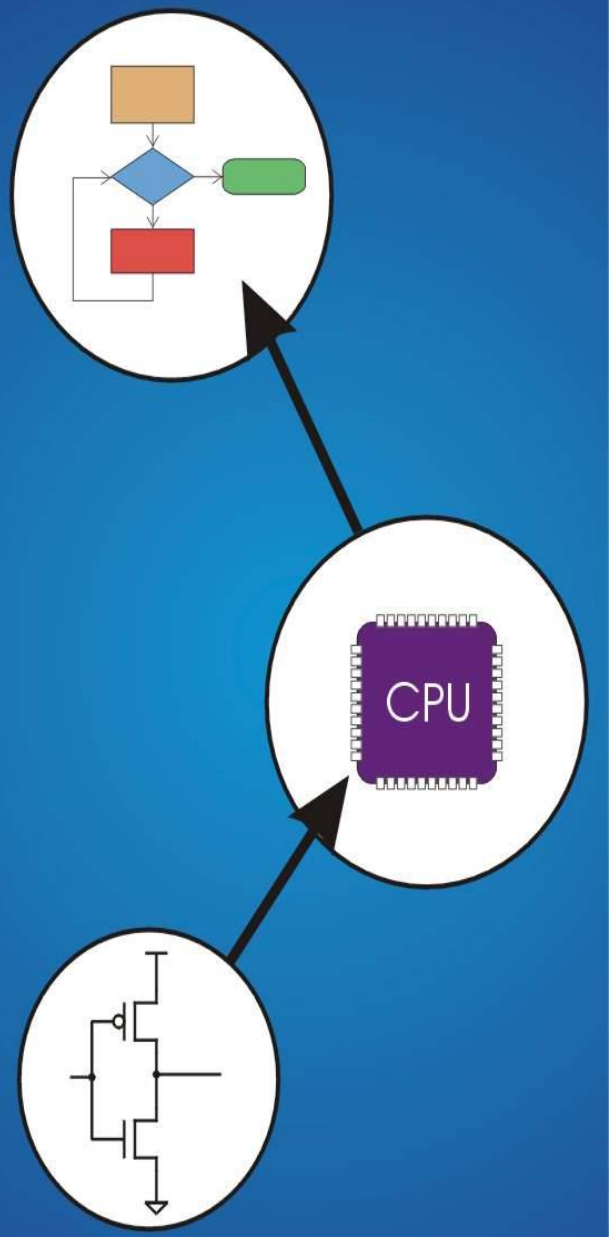
8051 Instruction Set

Instruction Set

Logical Instructions with Bits

Instruction Set : Logical Instructions with bits

CLR <bit>	**SETB <bit>**	**CPL <bit>**
CLR C	SETB C	CPL C
CLR PSW.7	SETB PSW.7	CPL PSW.7
CLR P0.1	SETB P0.1	CPL P0.1



UNIT – II

8051 Instruction Set

Instruction Set

Branch Instructions

Instruction Set : Branch Instructions

Unconditional Branch Instructions

Conditional Branch Instructions

Instruction Set : Unconditional Branch Instructions

ACALL **<Address>** ; ACALL 3000H

LCALL **<Address>** ; LCALL F000H

SJMP **<Relative Address>** ; SJMP 80H

Instruction Set : Unconditional Branch Instructions

RET <Address>

RET 3000H

RETI <Address>

RETI F000H

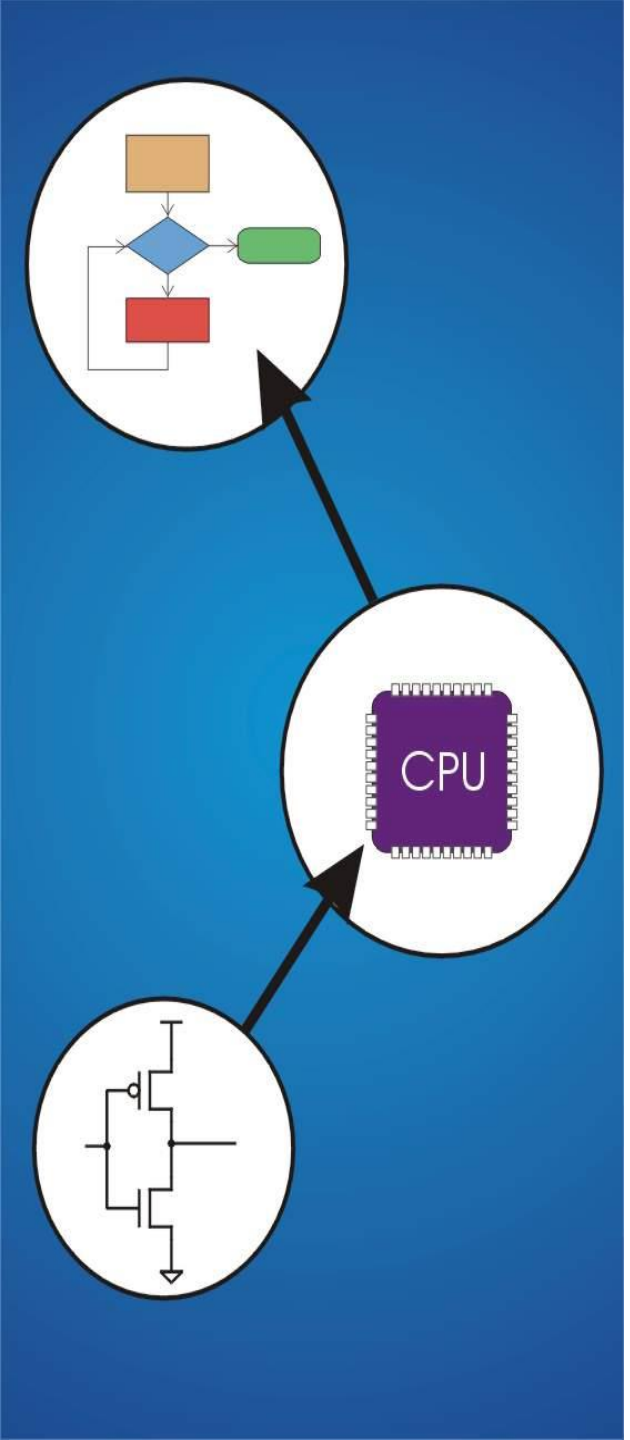
Instruction Set : Conditional Branch Instructions

JZ <Relative Address>

JNZ <Relative Address>

JC <Relative Address>

JNC <Relative Address>



UNIT – II

8051 Instruction Set

Instruction Set

Assembly Language Program

Instruction Set : Assembly Language Program Structure

```
$INCLUDE  (REG51.INC)

ORG 0000 H


LOOP:  SJMP   LOOP

       END
```

Instruction Set : Assembly Language Program

Write an assembly language program to perform addition of two numbers stored in **R0** and **R1**, store result in **R2**

```
$INCLUDE (REG51.INC)

ORG 0000 H

MOV R0, #01H
MOV R1, #04H

MOV A, R0
ADD A, R1
MOV R2, A

LOOP: SJMP LOOP

END
```

Instruction Set : Assembly Language Program

Write an assembly language program to perform addition of two 8 bit numbers stored in internal RAM **30H** and **31H**, store result in **32H**

```
$INCLUDE (REG51.INC)

ORG 0000 H

MOV A, 30H
ADD A, 31H
MOV 32H, A

LOOP: SJMP LOOP

END
```

Course Contents

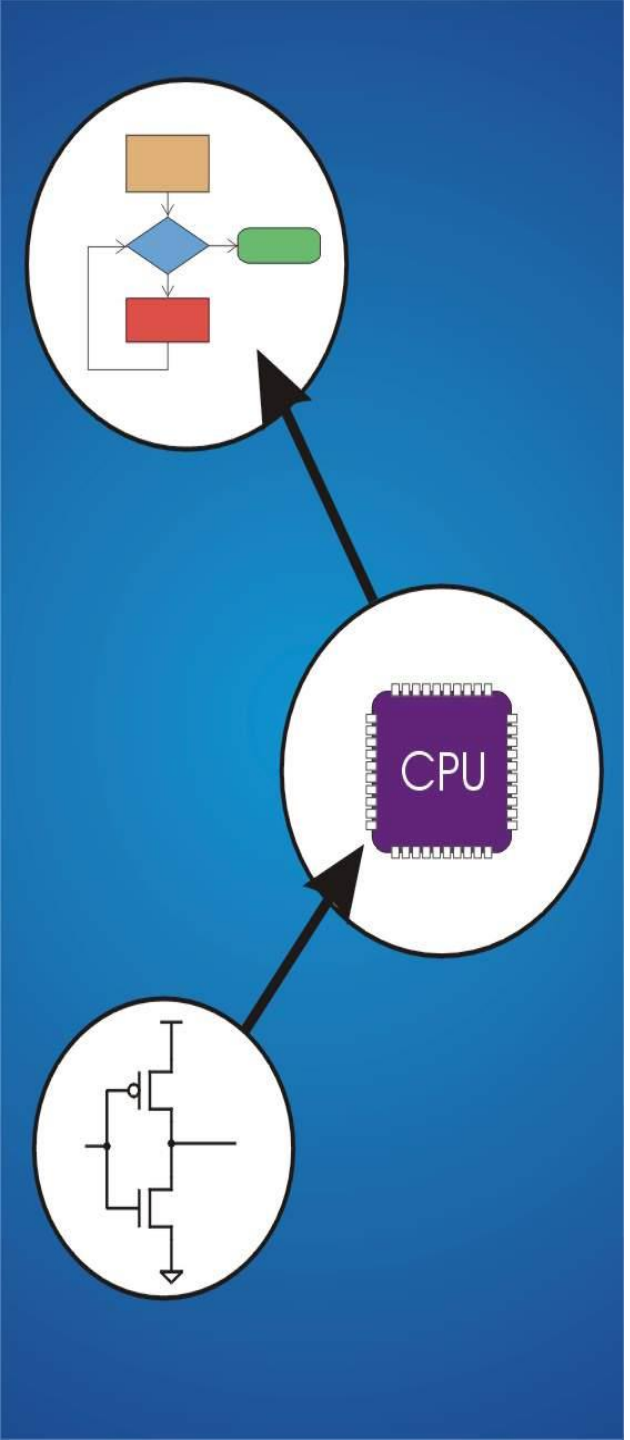
1. Introduction to Microcontroller
2. 8051 Instruction Set
3. Facilities in 8051
4. Interfacing Methods

Course Contents

FACILITIES IN 8051

Timer and Counter:
Timer and Counters, Timer modes, Programming the timers in Mode 1, Mode 2 using assembly and C. Time delay generation.

Serial Port :
Serial port of 8051, RS-232 standard and IC MAX–232, Baud rate in 8051, programming for transmitting character through serial port using assembly and C.



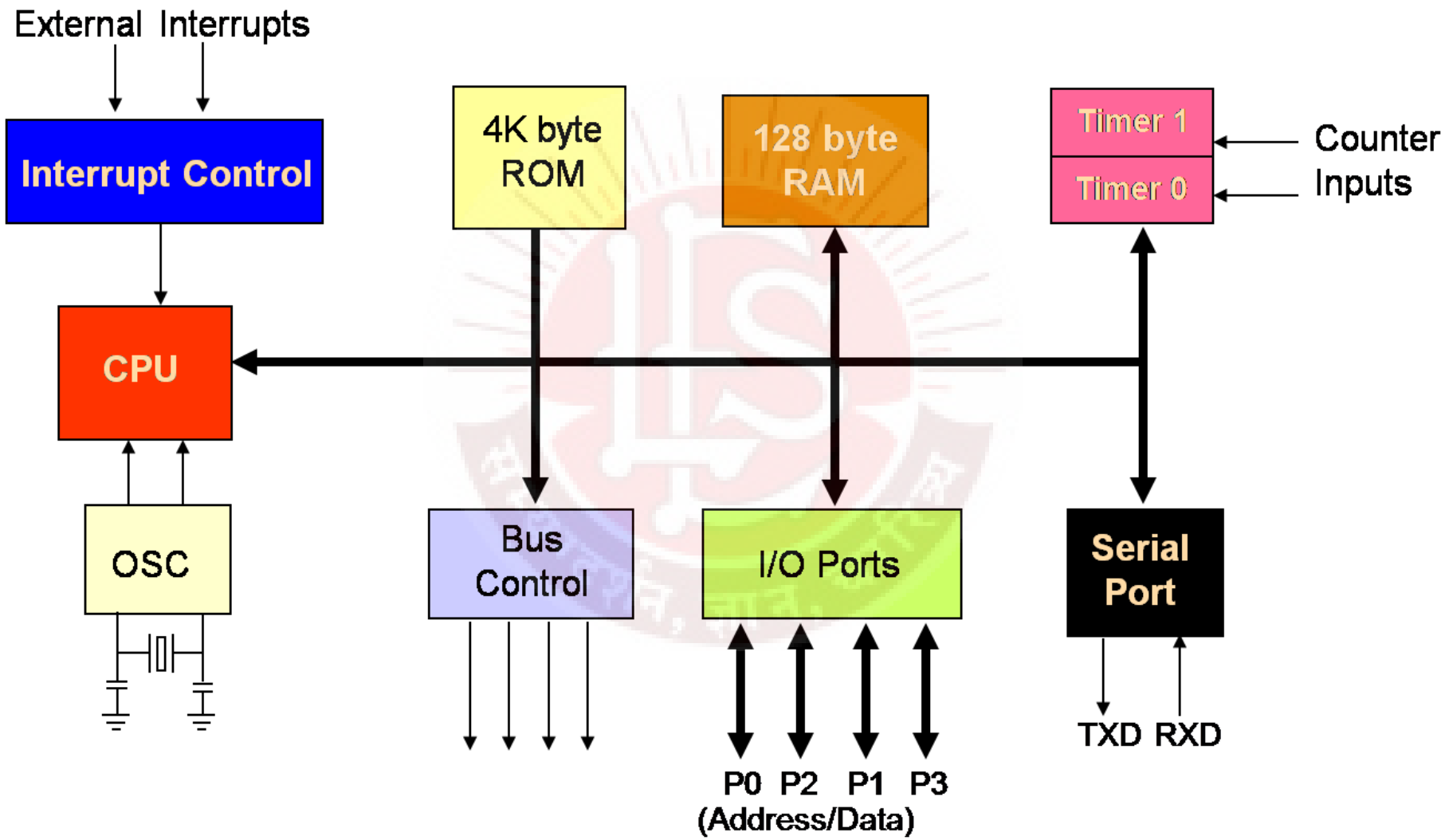
UNIT – III

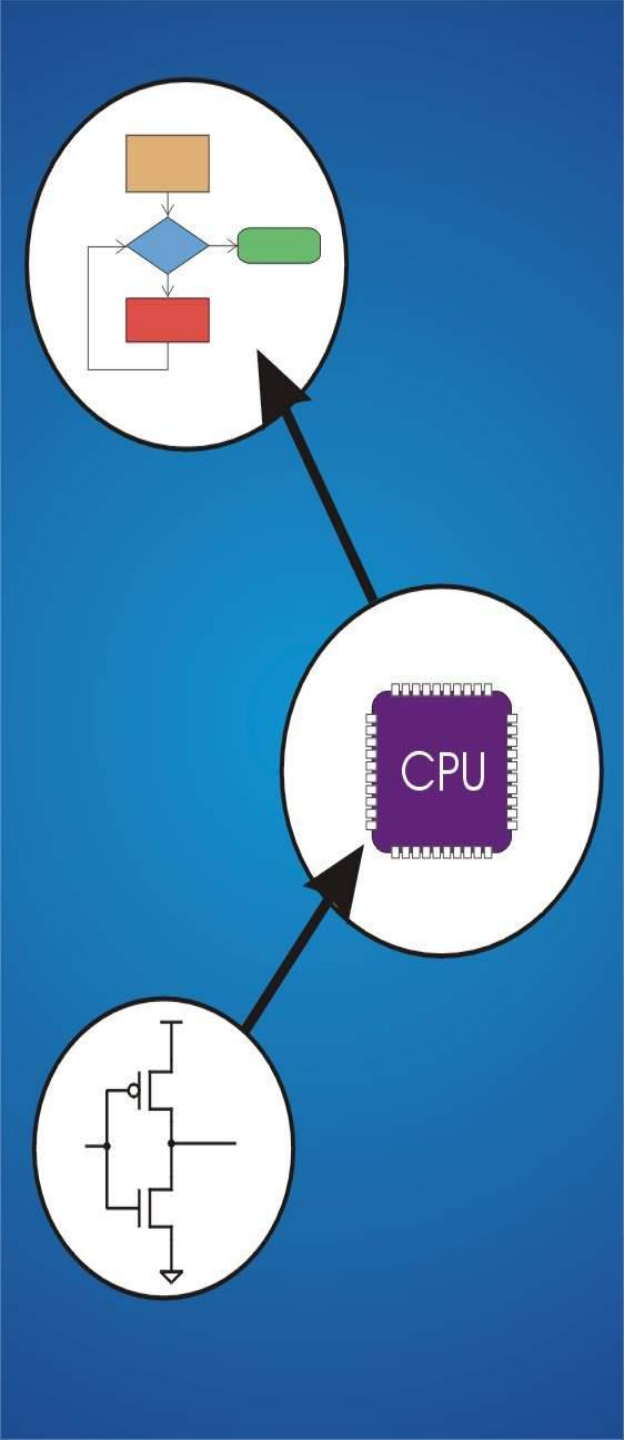
Facilities in 8051

Prerequisite

8051 Architecture

8051 Architecture :



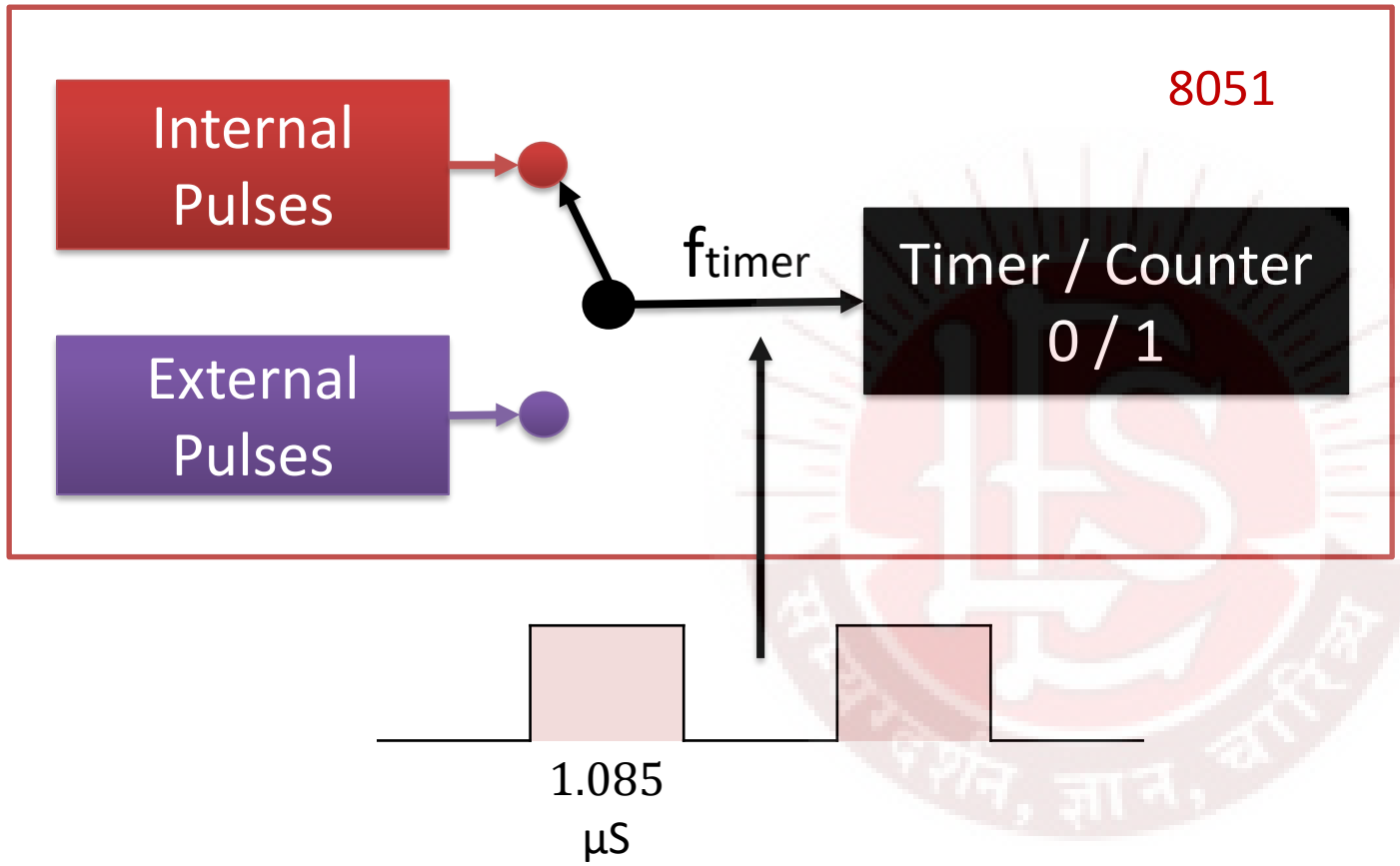


UNIT – III

Facilities in 8051

Timer

Timer / Counter Concept :



Pulse Count Register

65,535

.

.

0000

Pulse Count Register = 100 ← **Counter**

Time = 100 x 2 (1.085 µS) = 217 µS ← **Timer**

Timers :

Timer 0

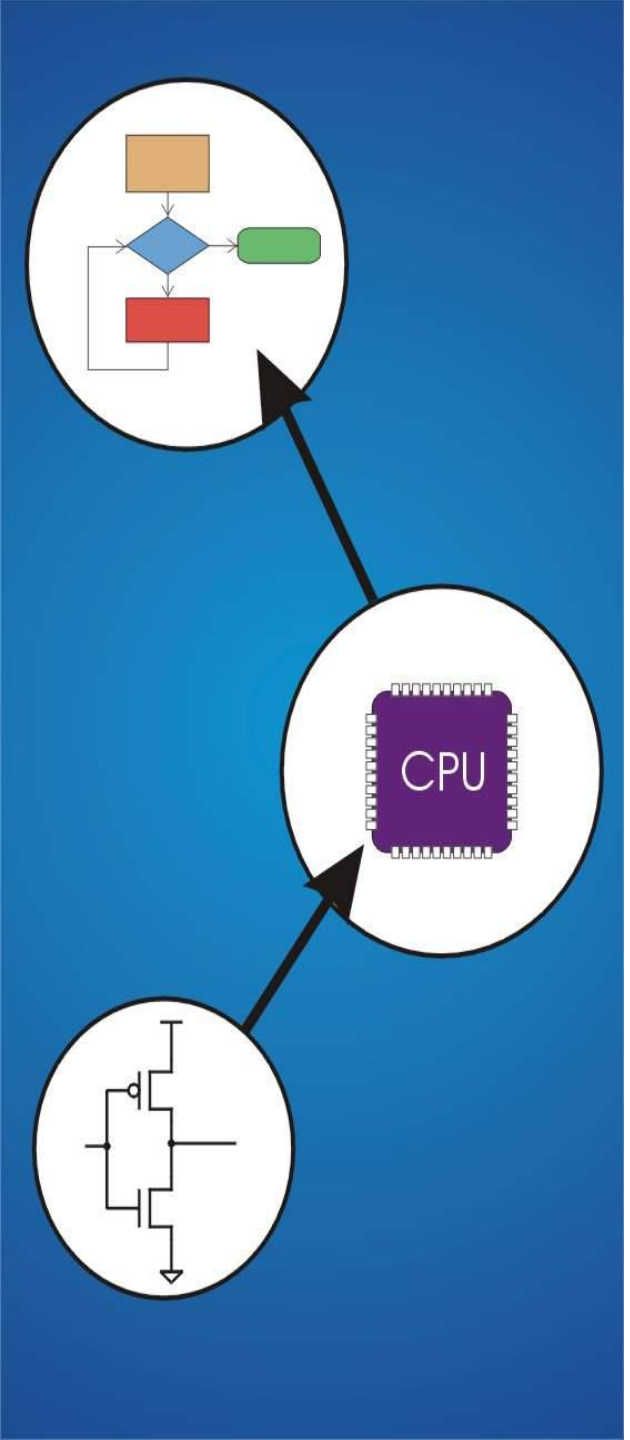
Timer 1

Can be used as Timer or Counter

Both timers are 16 bit Wide

Uses Microcontrollers internal clock

Timer Data Register,
Timer Mode Register and
Timer Control Register



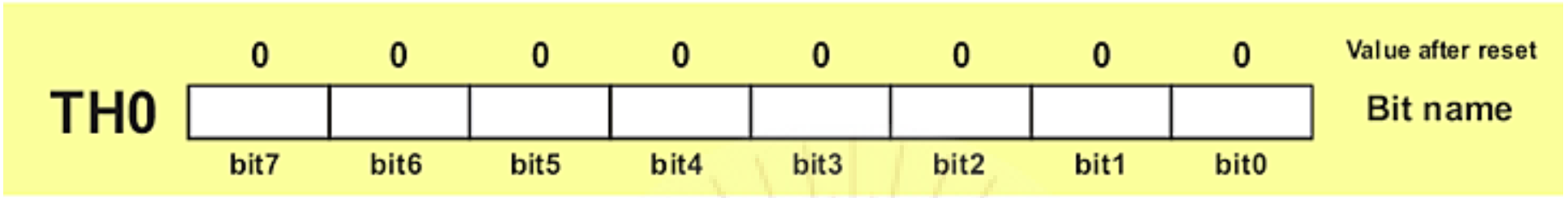
UNIT – III

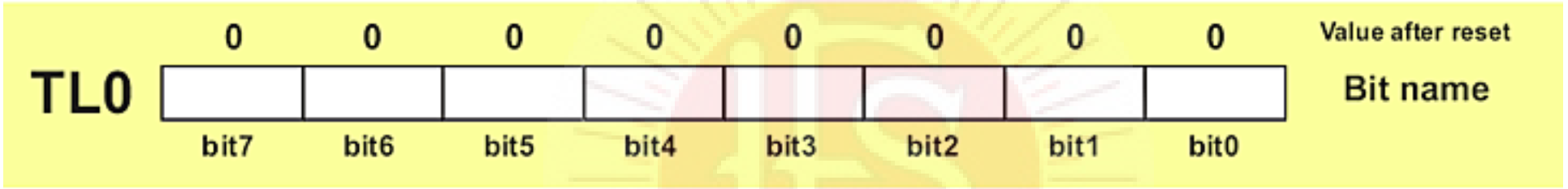
Facilities in 8051

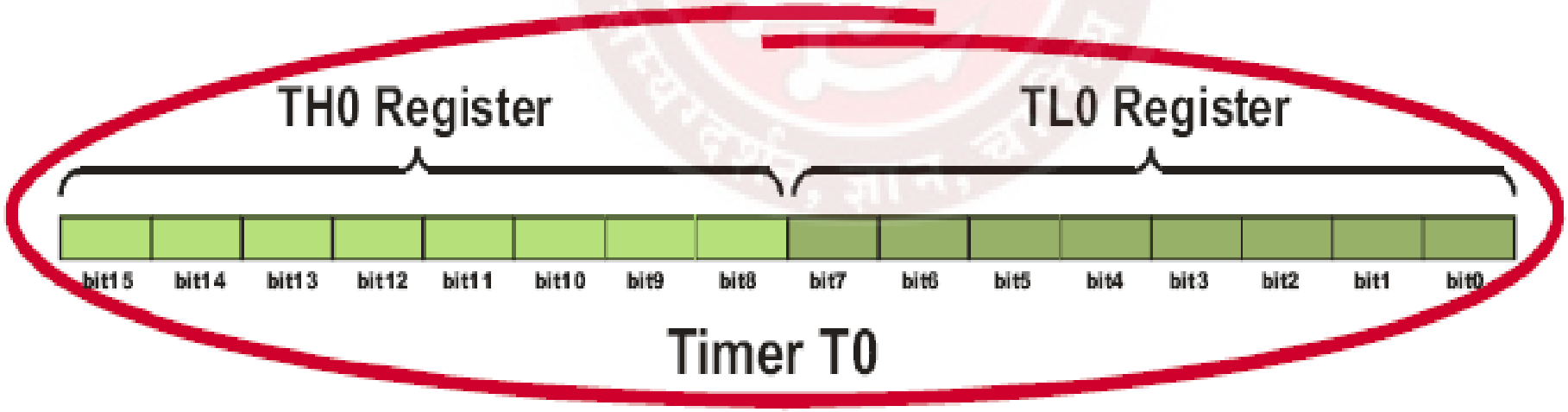
Timer Data Register

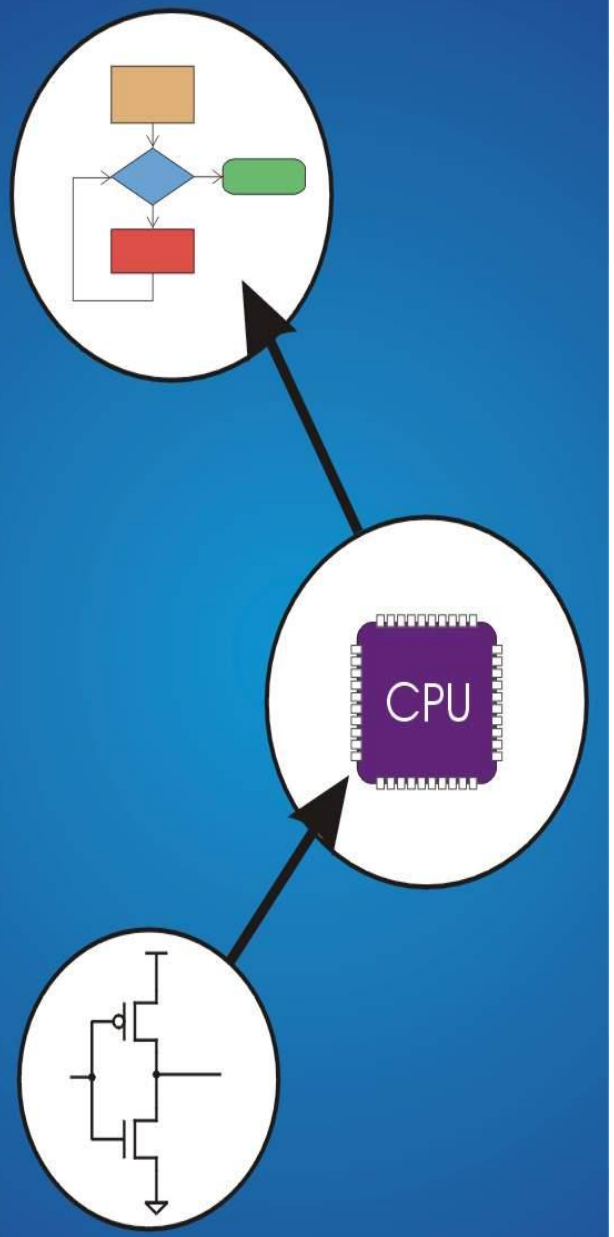
TH and TL

Timer Data Register :









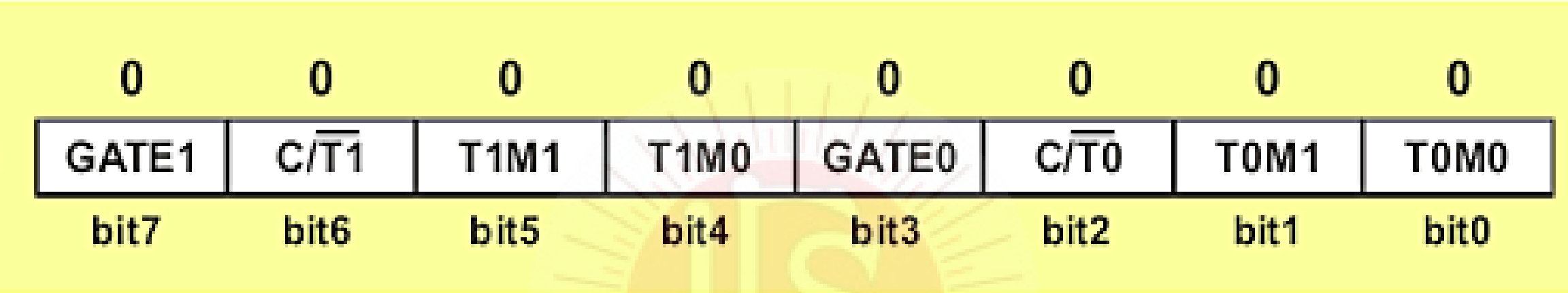
UNIT – III

Facilities in 8051

Timer Mode Register

TMOD

Timer Control Register : TMOD



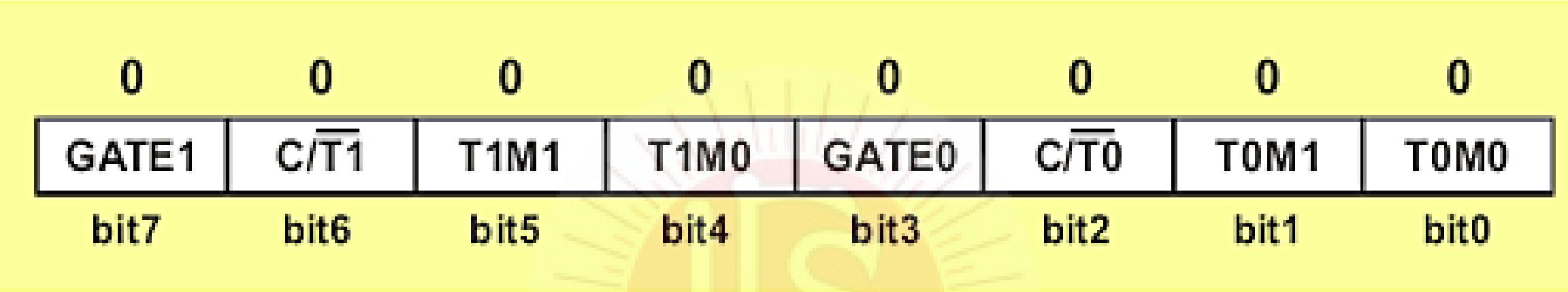
GATE : GATE1 / GATE0

Starts the Timer (1/0) internally or externally.

0 – Starts timer internally.

1 – Starts timers externally by applying pulse to INT1 / INT0 pins

Timer Control Register : TMOD



$C/\overline{T}$: $C/\overline{T1}$, $C/\overline{T0}$

Selects source of clock pulses to be counted up

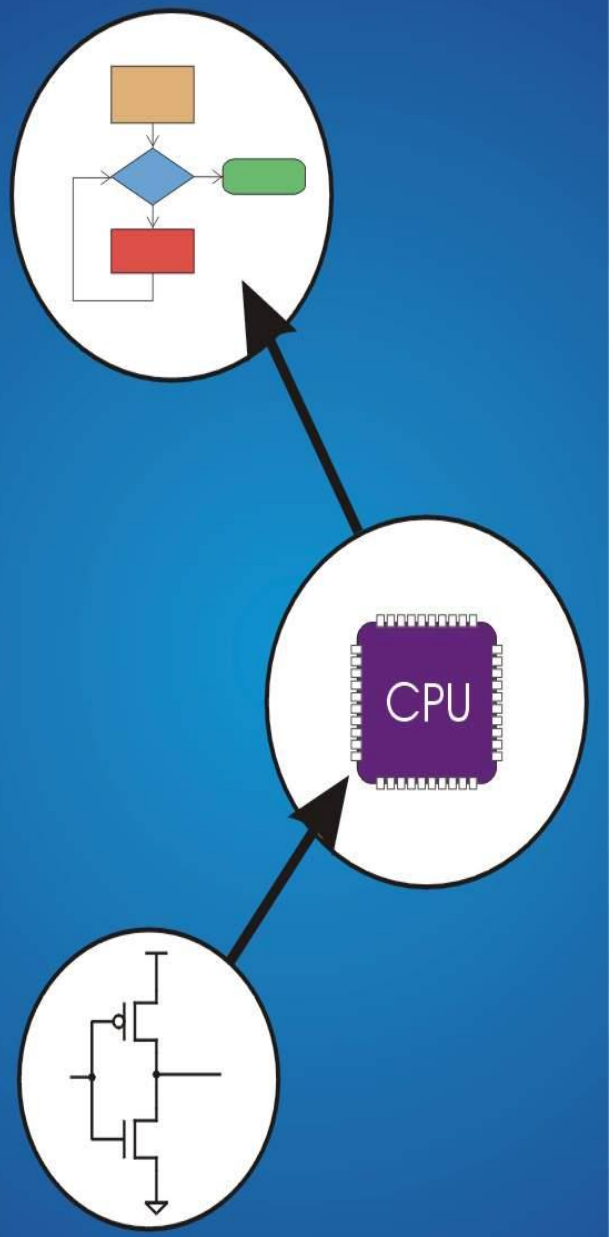
0 – Counts pulses from internal oscillator.

1 – Counts pulses provided to INT1 / INT0 pins.

Timer Control Register : TMOD

0	0	0	0	0	0	0	0
GATE1	C/$\overline{T1}$	T1M1	T1M0	GATE0	C/$\overline{T0}$	T0M1	T0M0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

M1	M0	Mode	Description
0	0	0	13 bit Timer
0	**1**	**1**	**16 bit Timer**
1	**0**	**2**	**8 bit Auto Reload Timer**
1	1	3	Split Timer



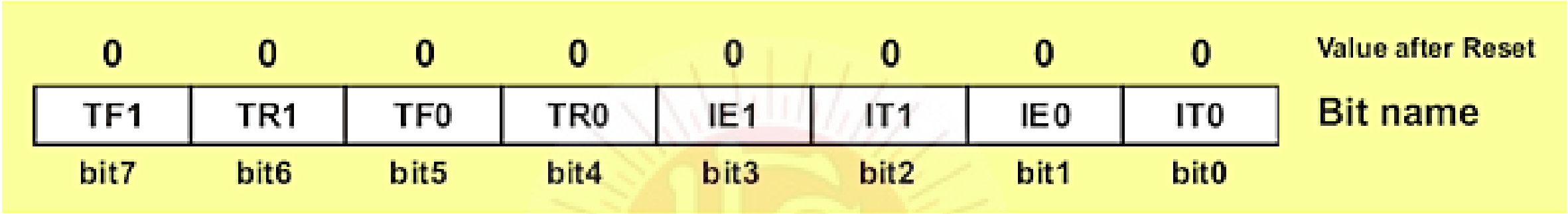
UNIT – III

Facilities in 8051

Timer Control Register

TCON

Timer Control Register : TCON



TF : TF1 / TF0

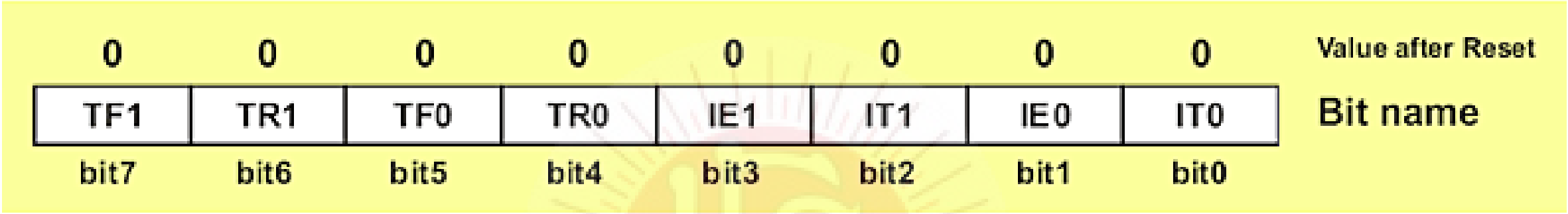
Timer Overflow Flag (1/0)

Whenever timer (T0 / T1) overflows this flag is set (TF1=1 / TF0=1).

16 bit mode – Overflows from FFFF – 0000 H

8 bit mode – Overflows from FF – 00 H

Timer Control Register : TCON



TR : TR1 / TR0

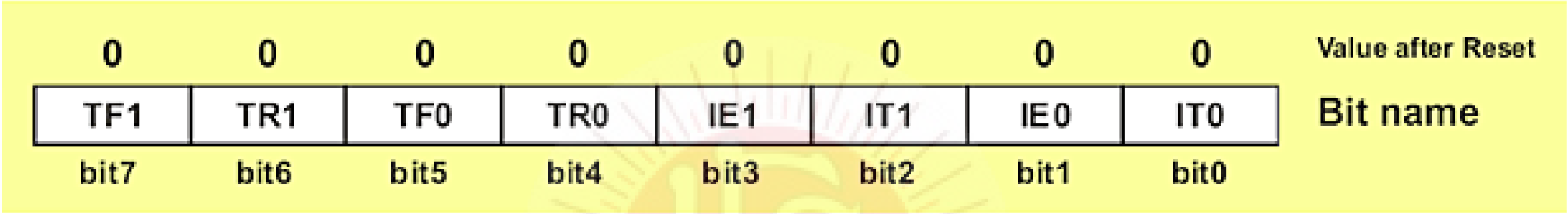
Timer Run Flag (1/0)

Stats or Stops the timer (T0 / T1).

1 – Starts the timer (TR1 = 1, TR0 = 1)

0 – Stops the timer (TR1 = 0, TR0 = 0)

Timer Control Register : TCON



IE : IE1 / IE0

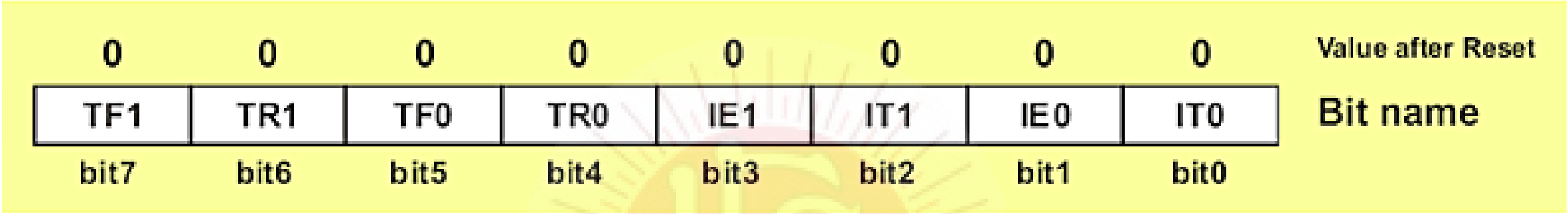
External Interrupt Enable (1/0)

Enables or Disables the Interrupts for timer (T0 / T1).

1 – Enables the Interrupt (IE1 = 1, IE0 = 1)

0 – Disable the Interrupt (IE1 = 0, IE0 = 0)

Timer Control Register : TCON

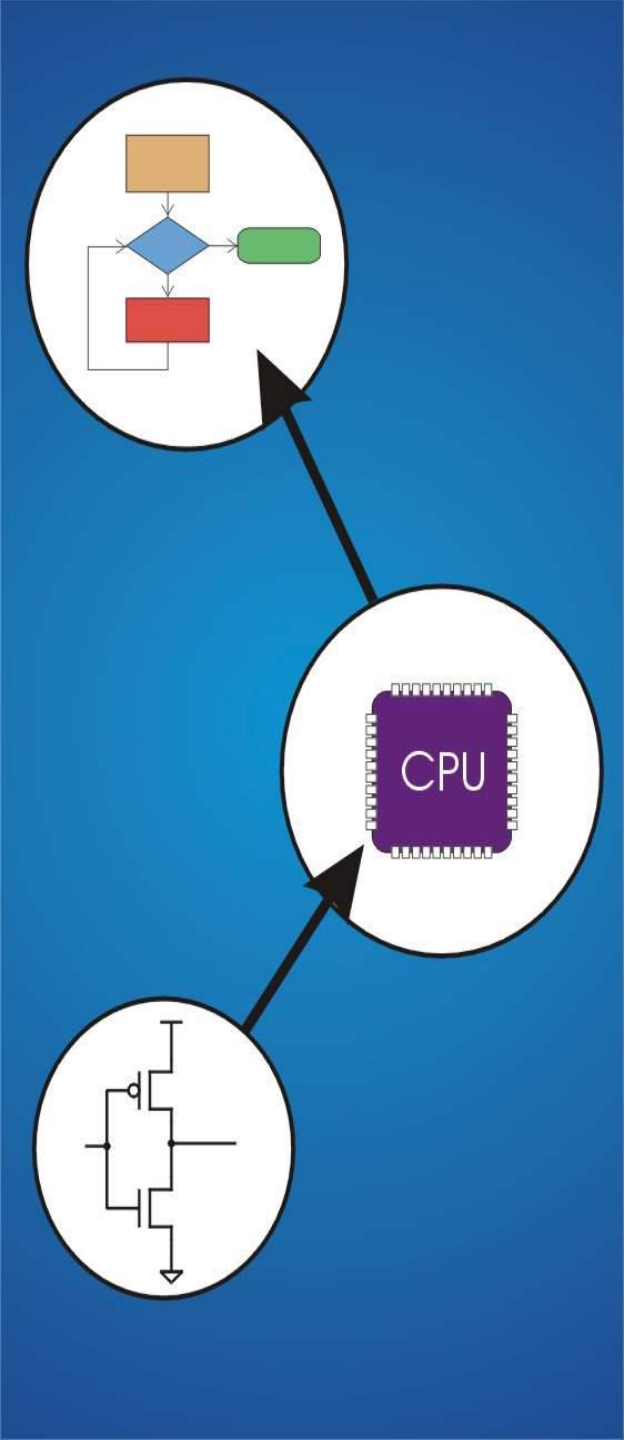


IT : IT1 / IT0

Interrupt Type (1/0)

1 – Interrupts at Negative Edge of clock pulse applied at INT1 / INT0 (IT1 = 1, IT0 = 1)

0 – Interrupts at Logic Zero of clock pulse applied at INT1 / INT0 (IT1 = 1, IT0 = 1)

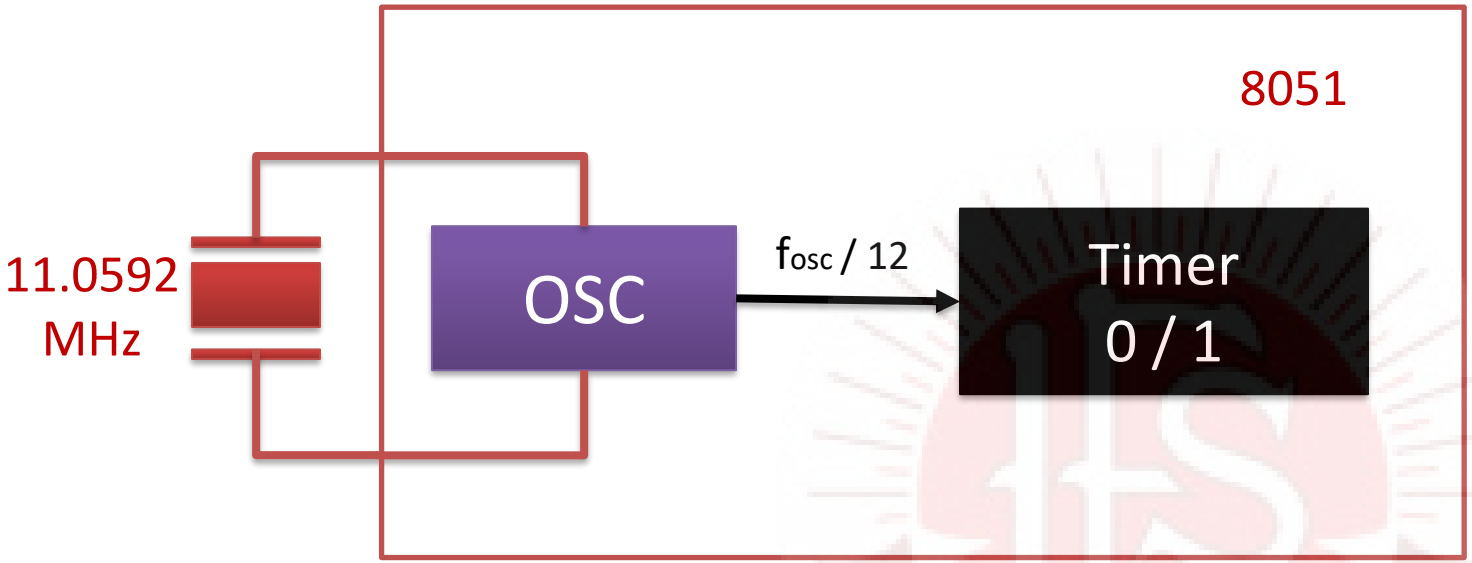


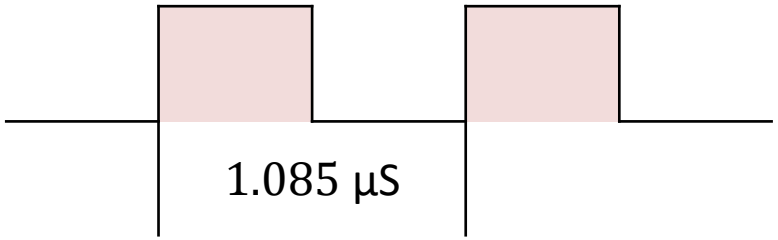
UNIT – III

Facilities in 8051

Timer Clock

Timer Clock :





$$fosc = fxtal = 11.0592\ x\ 10^6$$

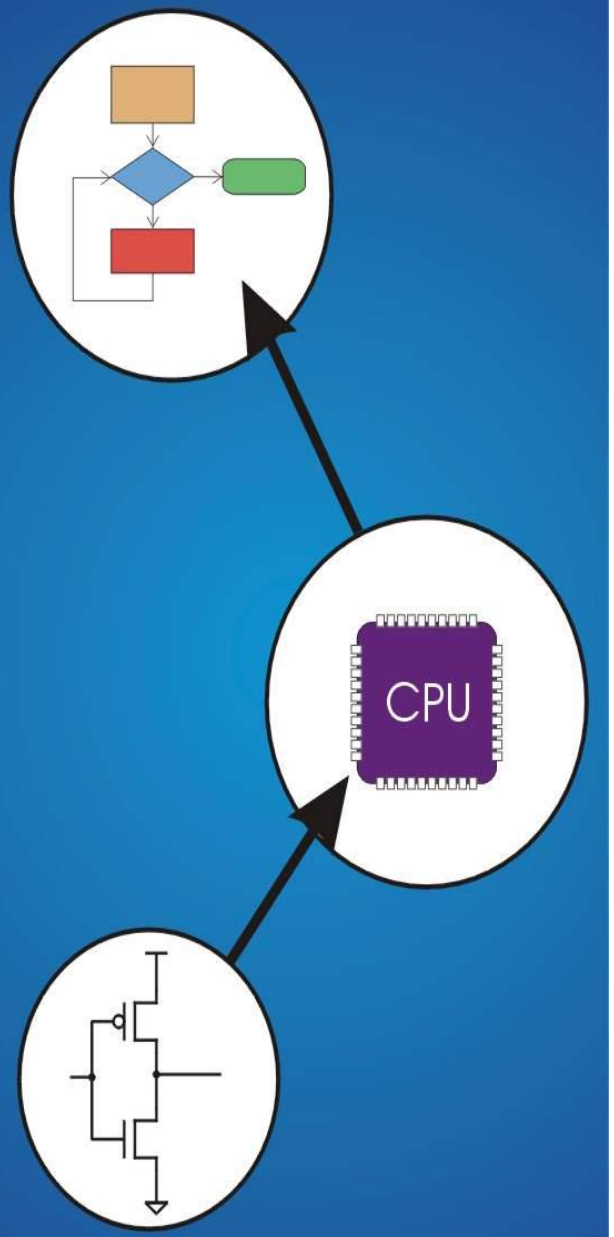
$$ftimer = \frac{fosc}{12} = \frac{11.0592\ x\ 10^6}{12}$$

$$ftimer = 0.9216\ x\ 10^6$$

$$Ttimer = \frac{1}{ftimer} = \frac{1}{0.9216\ x\ 10^6}$$

$$Ttimer = 1.085\ x\ 10^{-6}$$

$$Ttimer = 1.085\ \mu S$$



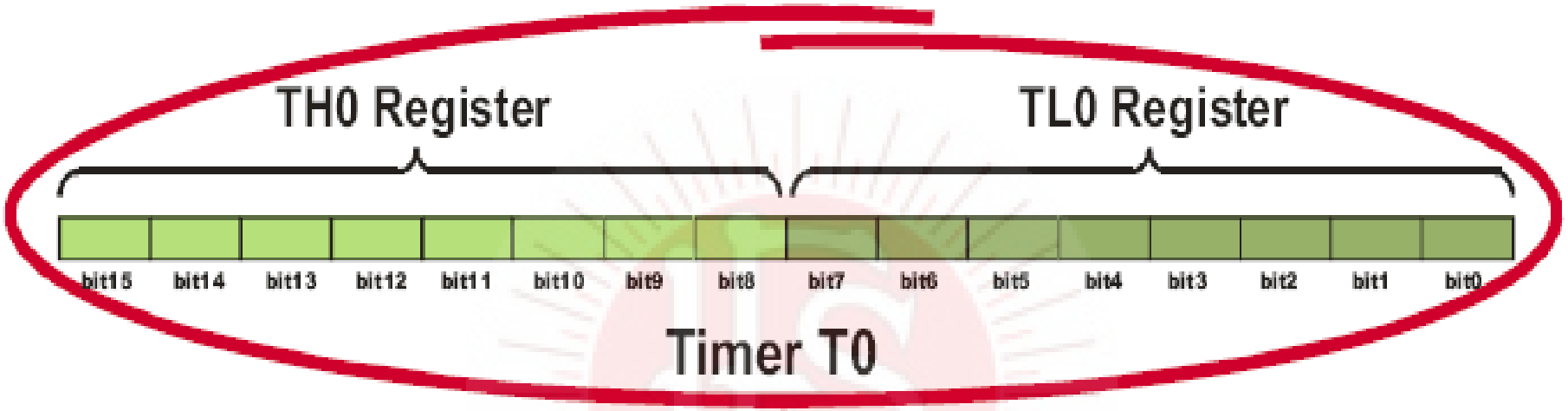
UNIT – III

Facilities in 8051

How to find pulses to be counted by

Timer

Timer Data Register :



$$T = TH0 \times 256 + TL0$$

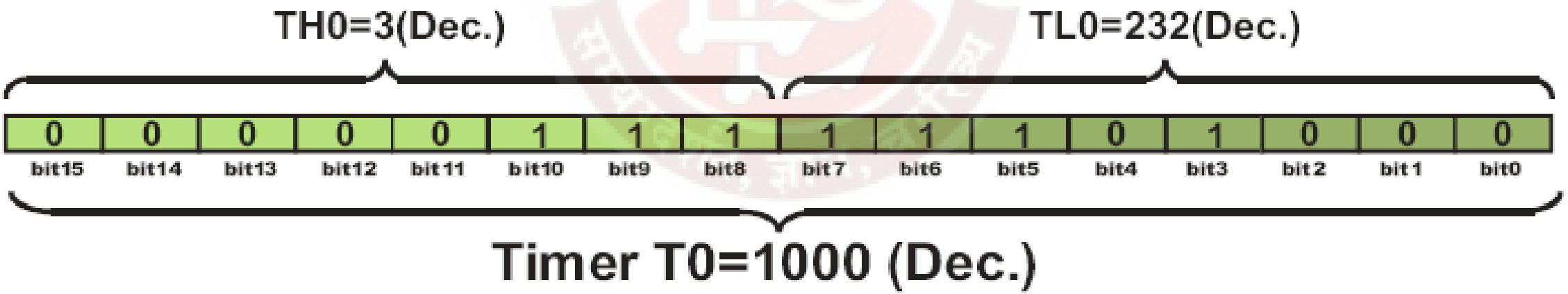
Timer Data Register : TH and TL

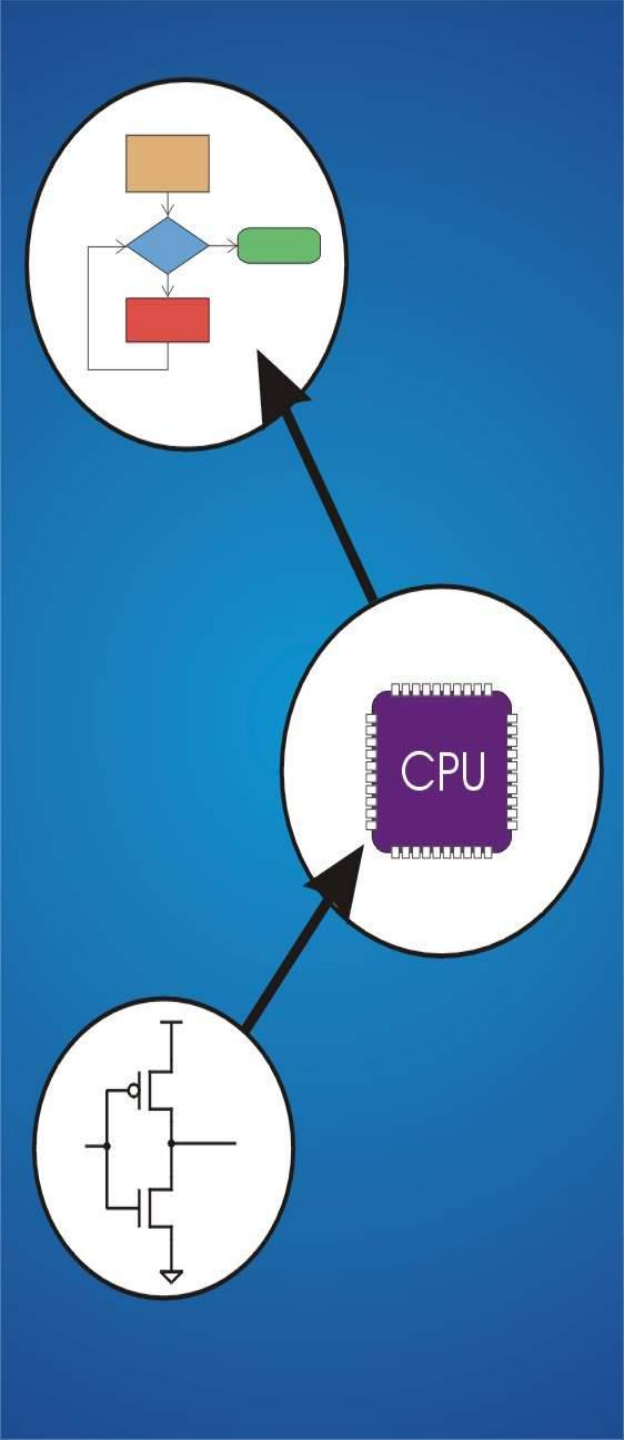
Find the data register content so that T0 Data register contains 1000

T = TH0 × 256 + TL0

1000 = 3 × 256 + 232

TH0 = 3 **TL0 = 232**





UNIT – III

Facilities in 8051

How to Start a

Timer

How to Start Timer :

Configure Timer using **TMOD**

Whether timer started internally or externally ($GATE$ = 0)

Clock Source ($c/\overline{T}$ = 0) ← Internal pulses from oscillator

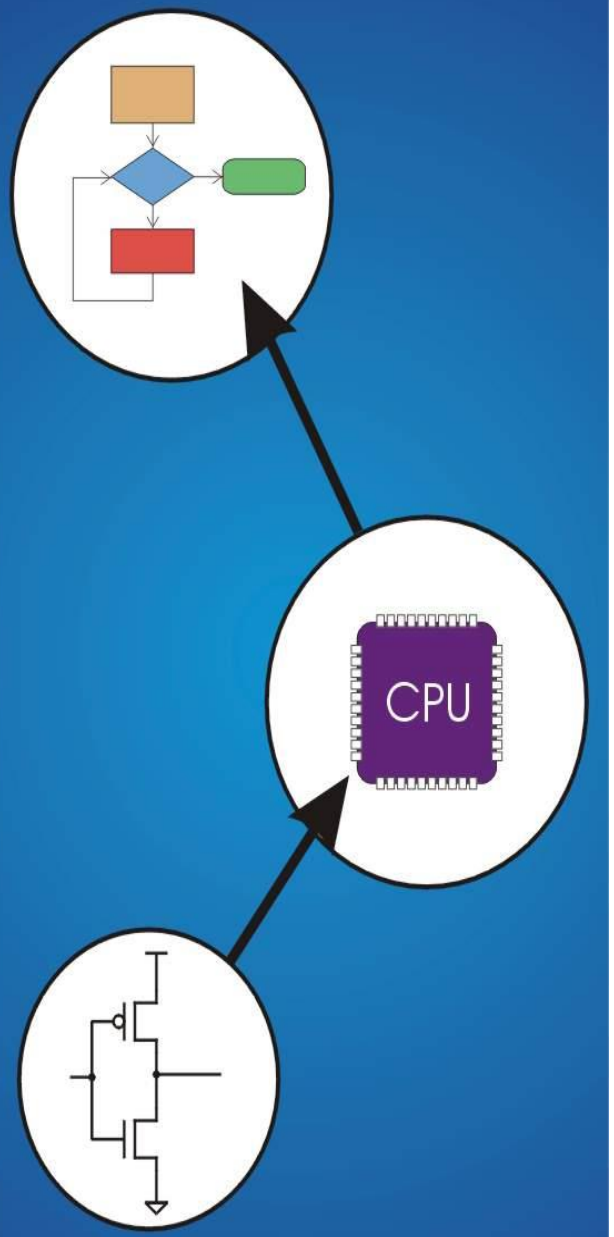
Mode of the timer (16/8 bit)

Load pulsed to be counted in

TH0 and TL0 registers / TH1 and TL1 registers

Start the Timer using **TCON**

TR0 = 1 or TR1 = 1



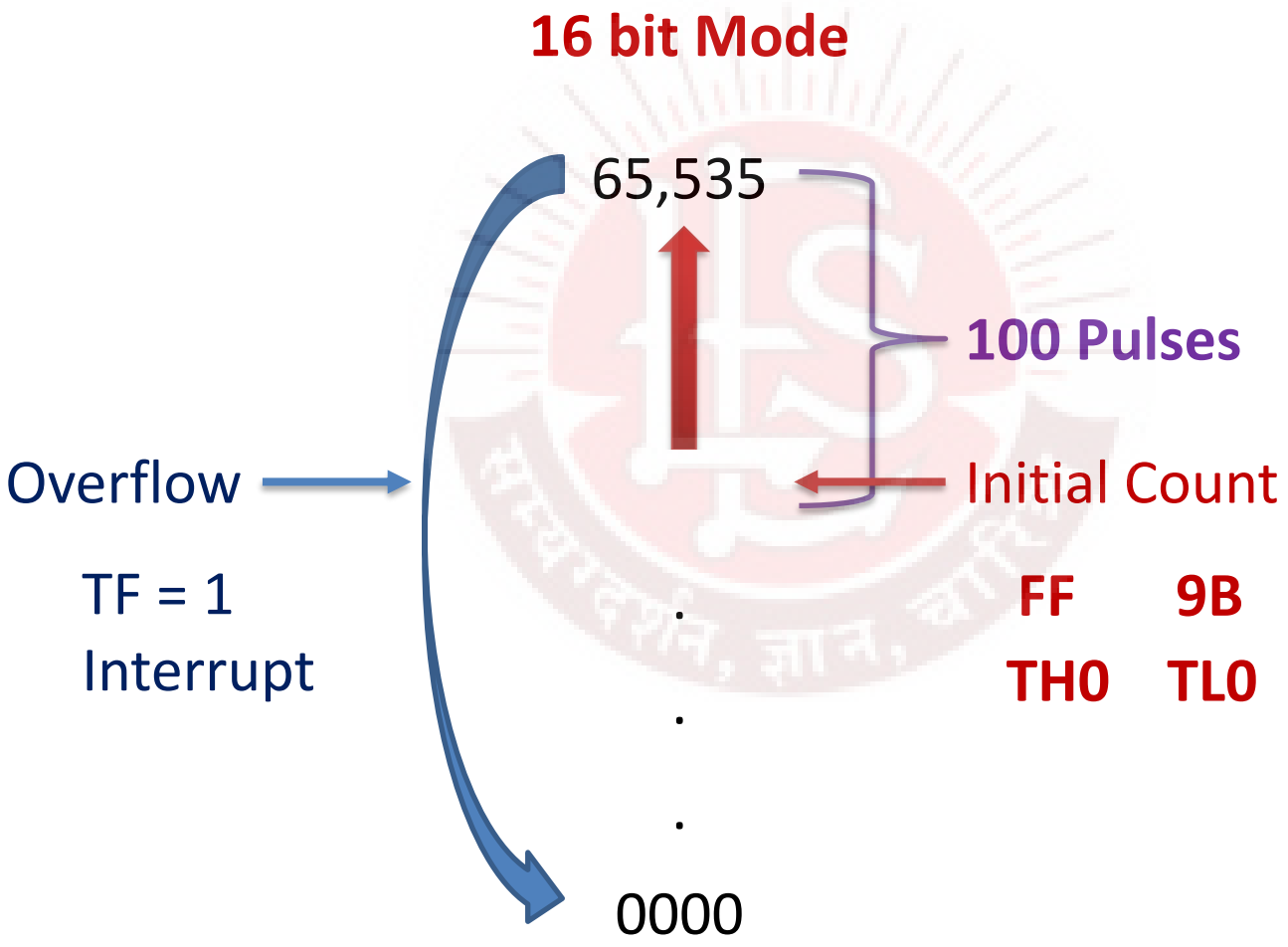
UNIT – III

Facilities in 8051

How to detect Timer Overflow

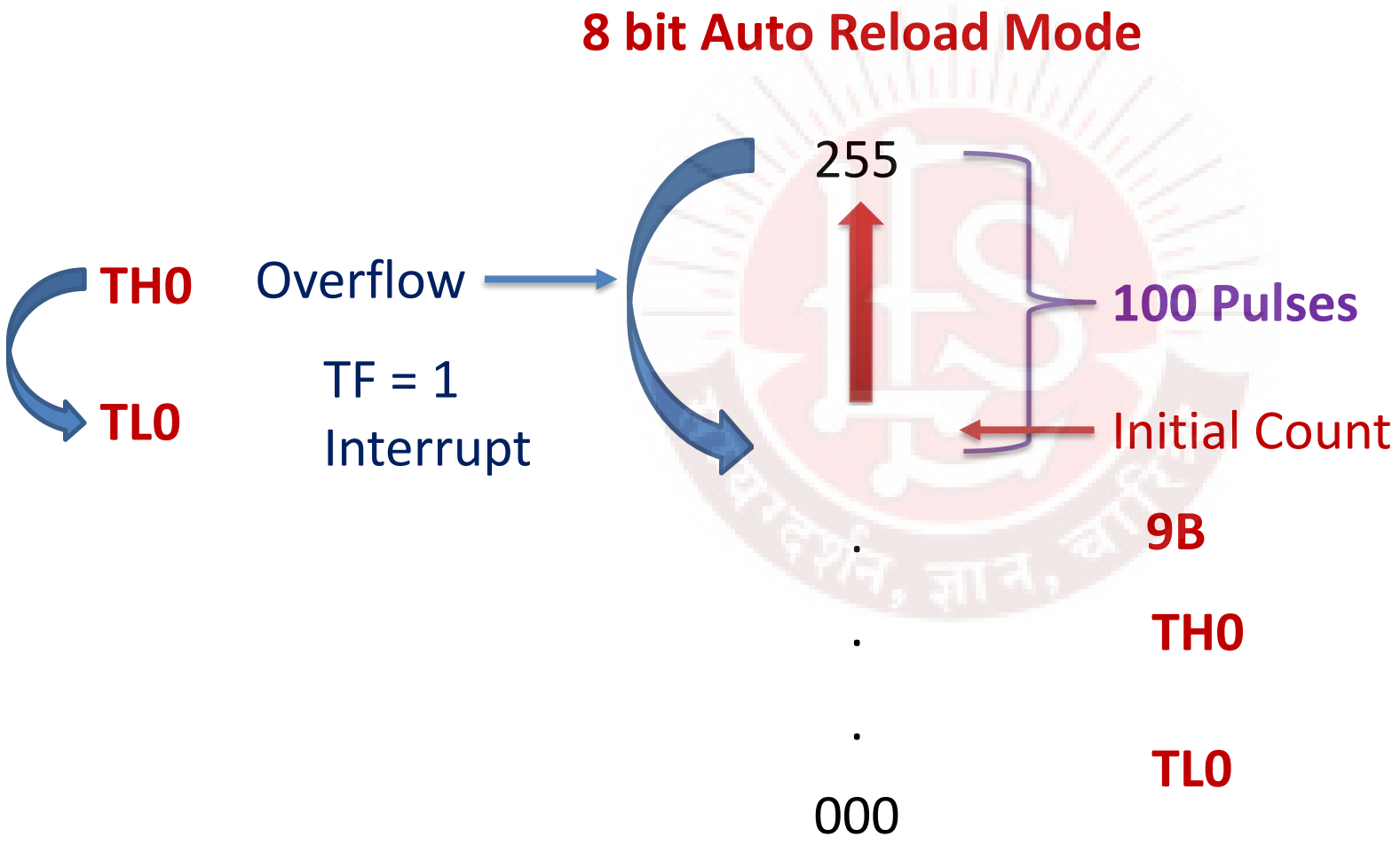
Timer Overflow : What it means ?

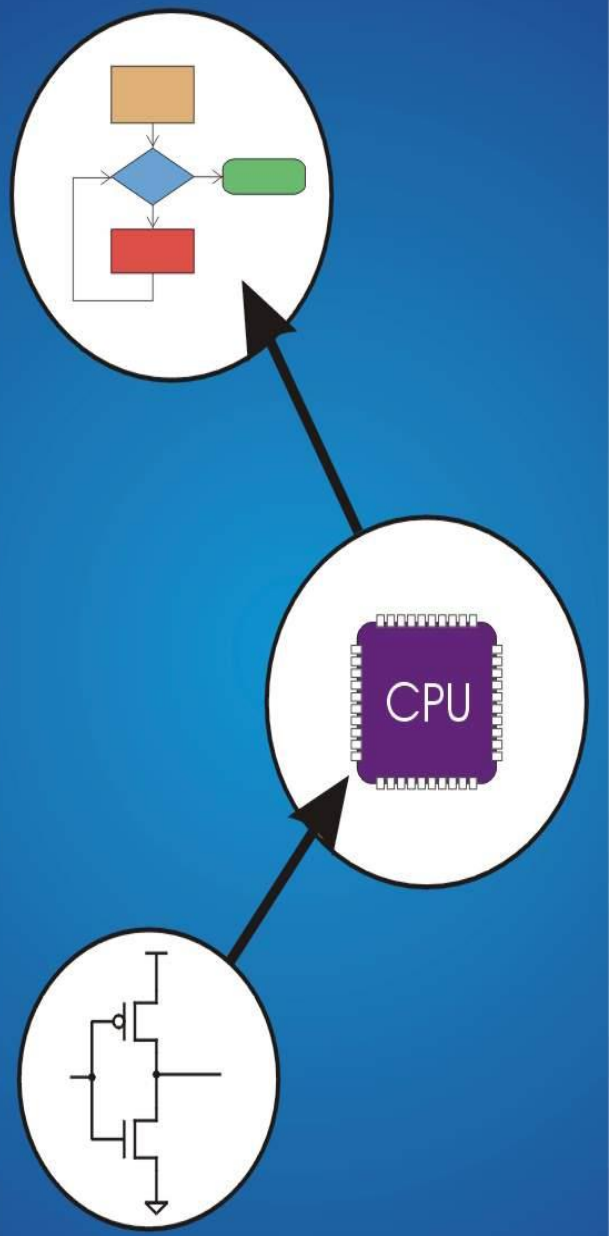
Find the data register content so that T0 counts 100 pulses



Timer Data Register : What it means ?

Find the data register content so that T0 counts 100 pulses





UNIT – III

Facilities in 8051

Time Delay Generation

Timer Delay Generation : Design

Write an assembly language program to generate time delay of **1 mS**.
Use **timer 0** and crystal frequency of **11.0592 MHz**

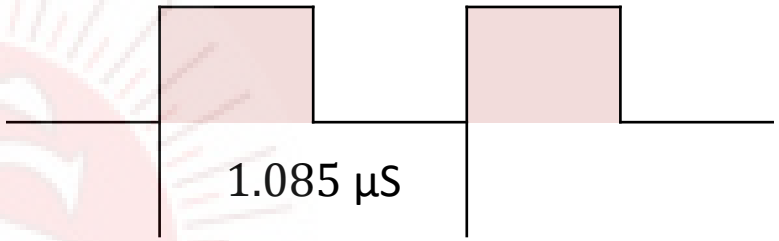
$$fosc = fxtal = 11.0592\ x\ 10^6$$

$$ftimer = \frac{fosc}{12} = \frac{11.0592\ x\ 10^6}{12}$$

$$ftimer = 0.9216\ x\ 10^6$$

$$Ttimer = \frac{1}{ftimer} = \frac{1}{0.9216\ x\ 10^6}$$

$$Ttimer = 1.085\ x\ 10^{-6} = 1.085\ \mu S$$



$$No\ of\ pulses = \frac{Required\ time\ Delay}{Ttimer}$$

$$= \frac{1\ x\ 10^{-3}}{1.085\ x\ 10^{-6}}$$

$$= 921.65$$

$$= 922$$

Timer Delay Generation : Design

Write an assembly language program to generate time delay of **1 mS**.
Use **timer 0** and crystal frequency of **11.0592 MHz**

$$Pulses = 922$$

$$Count = 65{,}536 - 922$$

$$= 64{,}614$$

$$= FC66\ H$$

$$TH0 = FC\ H$$

$$TL0 = 66\ H$$

Design of TMOD Content

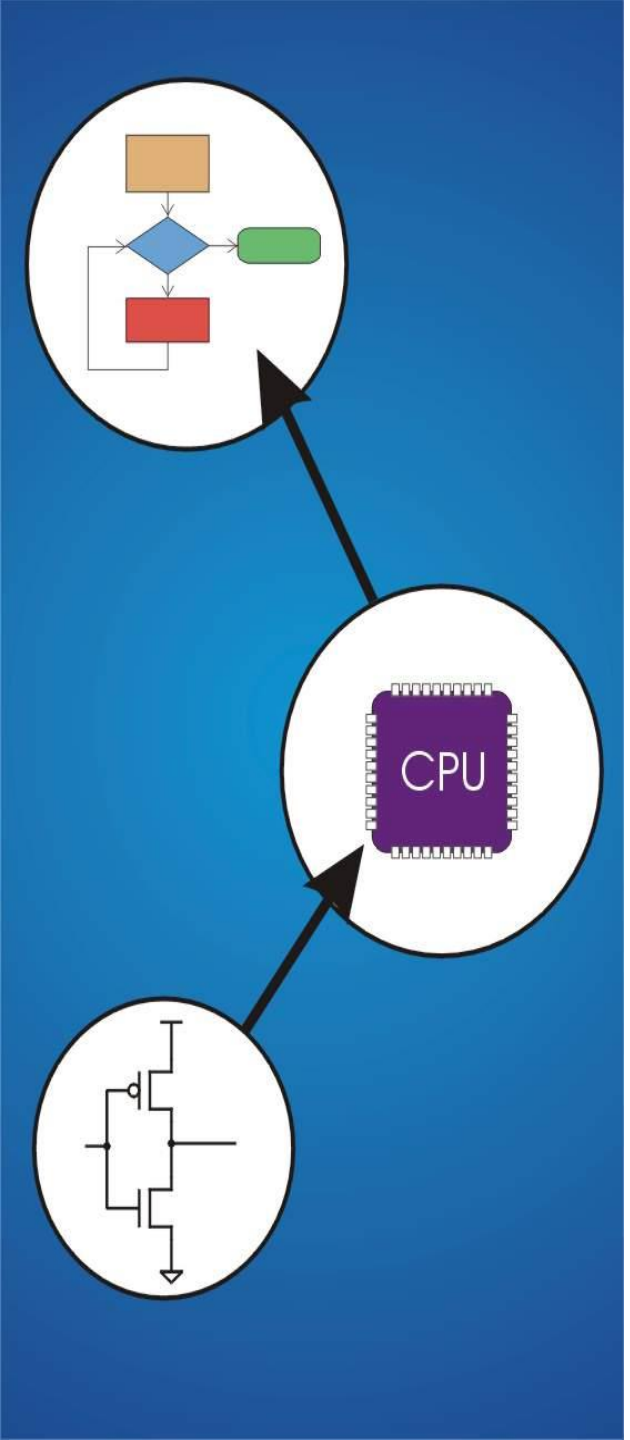
0	0	0	0	0	0	0	1
GATE1	C/$\overline{T1}$	T1M1	T1M0	GATE0	C/$\overline{T0}$	T0M1	T0M0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

$$TMOD = 01\ H$$

Timer Delay Generation : ALP $TH0 = FC\ H$ $TL0 = 66\ H$ $TMOD = 01\ H$

```
        $INCLUDE  (REG51.INC)

        ORG     0000 H

        MOV     P0, #00 H
        MOV     TMOD, #01 H

AGAIN:  MOV     TH0, #0FC H
        MOV     TL0, #66 H
        CPL     P0.0
        SETB    TR0

WAIT:   JNB     TF0, WAIT
        CLR     TR0
        CLR     TF0

LOOP:   SJMP    AGAIN

        END
```

```
#include <reg51.h>

sbit LED = P0 ^ 0;

void delay1ms() {
        TMOD = 0x01;
        TH0 = 0xFC;
        TL0 = 0x66;
        TR0 = 1;
        while(TF0 == 0);
        TR0 = 0;
        TF0 = 0;
}

void main() {
        P0 = 0x00;
        while(1)   {
            LED = ~LED;
            delay1ms();
        }
}
```



UNIT – III

Facilities in 8051

Time Delay Generation

50 µS

Timer Delay Generation : Design

Write an assembly language program to generate time delay of **50 µS**.
Use **timer 0** and crystal frequency of **11.0592 MHz**

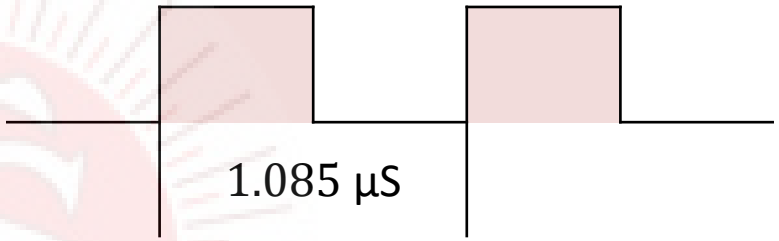
$$fosc = fxtal = 11.0592\ x\ 10^6$$

$$ftimer = \frac{fosc}{12} = \frac{11.0592\ x\ 10^6}{12}$$

$$ftimer = 0.9216\ x\ 10^6$$

$$Ttimer = \frac{1}{ftimer} = \frac{1}{0.9216\ x\ 10^6}$$

$$Ttimer = 1.085\ x\ 10^{-6} = 1.085\ \mu S$$



$$No\ of\ Pulses = \frac{Required\ time\ Delay}{Ttimer}$$

$$= \frac{50\ x\ 10^{-6}}{1.085\ x\ 10^{-6}}$$

$$= 46.08$$

$$= 46$$

Timer Delay Generation : Design

Write an assembly language program to generate time delay of **10 µS**.
Use **timer 0** and crystal frequency of **11.0592 MHz**

$$Pulses = 46$$

$$Count = 255 - 46$$

$$= 209$$

$$= D1\ H$$

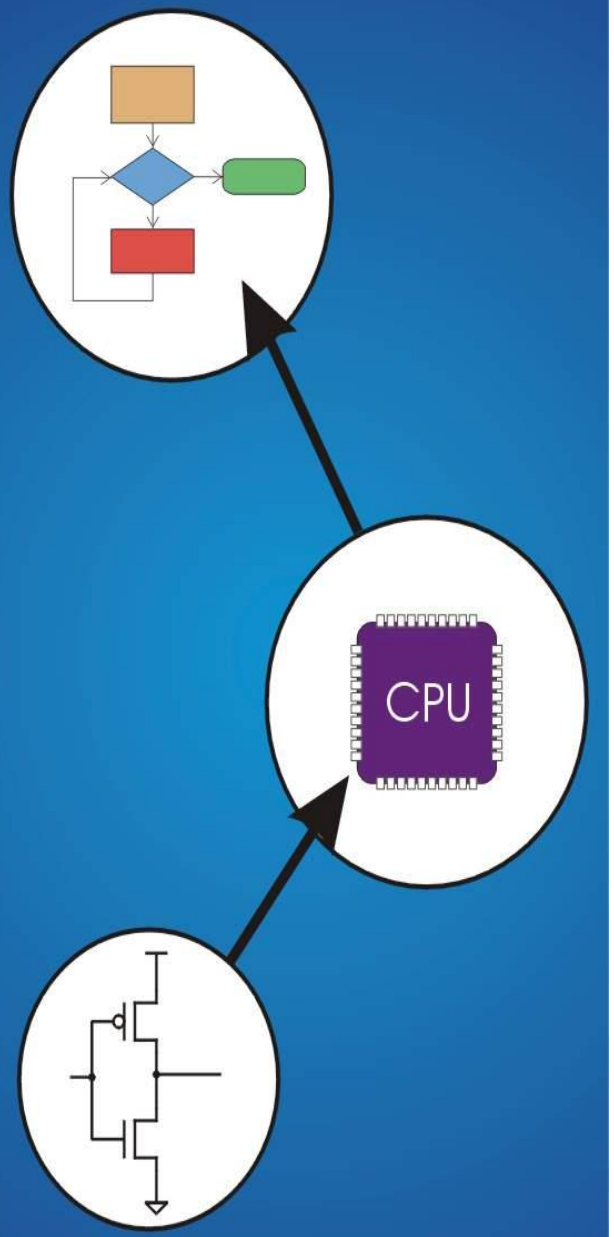
$$TH0 = D1\ H$$

$$TL0 = D1\ H$$

Design of TMOD Content

0	0	0	0	0	0	1	0
GATE1	C/$\overline{T1}$	T1M1	T1M0	GATE0	C/$\overline{T0}$	T0M1	T0M0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

$$TMOD = 02\ H$$



UNIT – III

Facilities in 8051

Time Delay Generation

1 S

Timer Delay Generation - 1 S : Design

Write an assembly language program to generate time delay of **1 S**.
Use **timer 0** and crystal frequency of **11.0592 MHz**

$$Ttimer = 1.085\ x\ 10^{-6} = 1.085\ \mu S$$

$$No.of\ Pulses = \frac{Required\ time\ Delay}{Ttimer}$$

$$= \frac{1000\ x\ 10^{-3}}{1.085\ x\ 10^{-6}}$$

$$= 921.6589\ x\ 10^{3}$$

$$= 9{,}21{,}658.9$$

$$= 9{,}21{,}659$$

$$Max\ Count = 65{,}536$$

$$Iterations = \frac{No.of\ Pulses}{Max\ Count}$$

$$= \frac{9{,}21{,}659}{65{,}536}$$

$$= 14.06$$

$$= 14$$

Timer Delay Generation – 1 S : Design

$Pulses = 65{,}536$

$Count = FFFF$ H

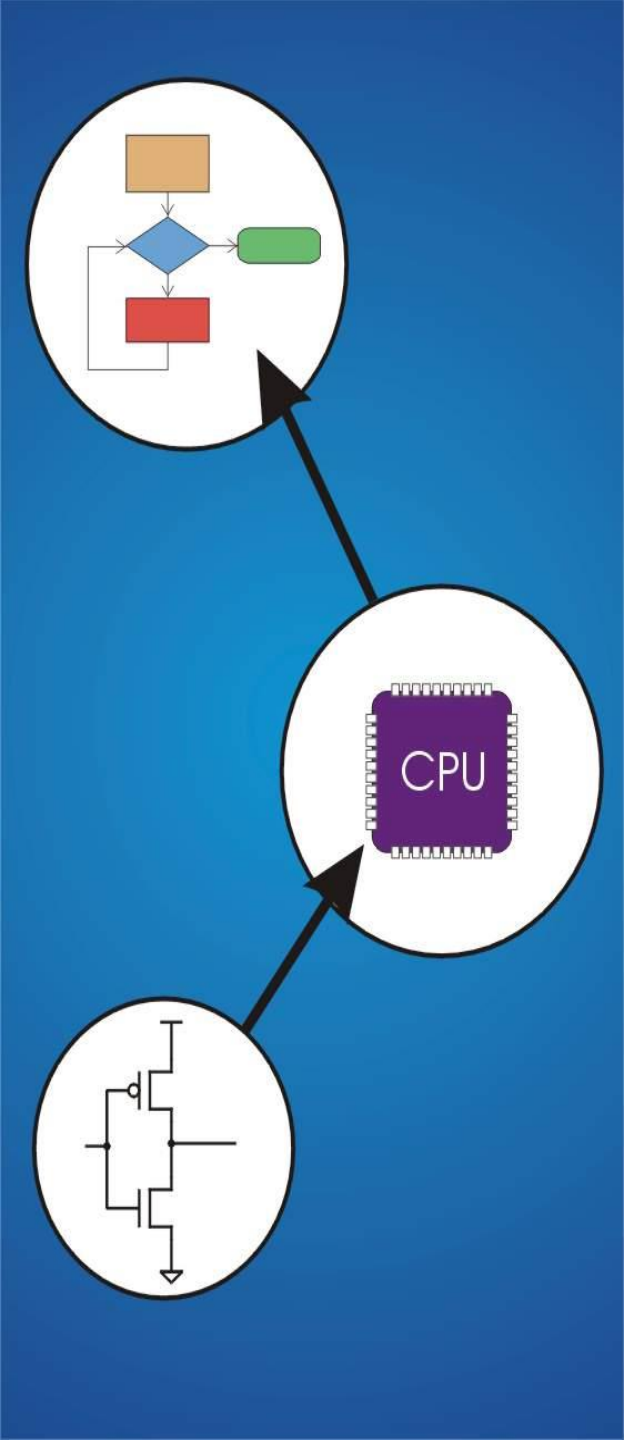
$TH0 = FF\ H$

$TL0 = FF\ H$

Design of TMOD Content

0	0	0	0	0	0	0	1
GATE1	C/$\overline{T1}$	T1M1	T1M0	GATE0	C/$\overline{T0}$	T0M1	T0M0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

$TMOD = 01\ H$



UNIT – III

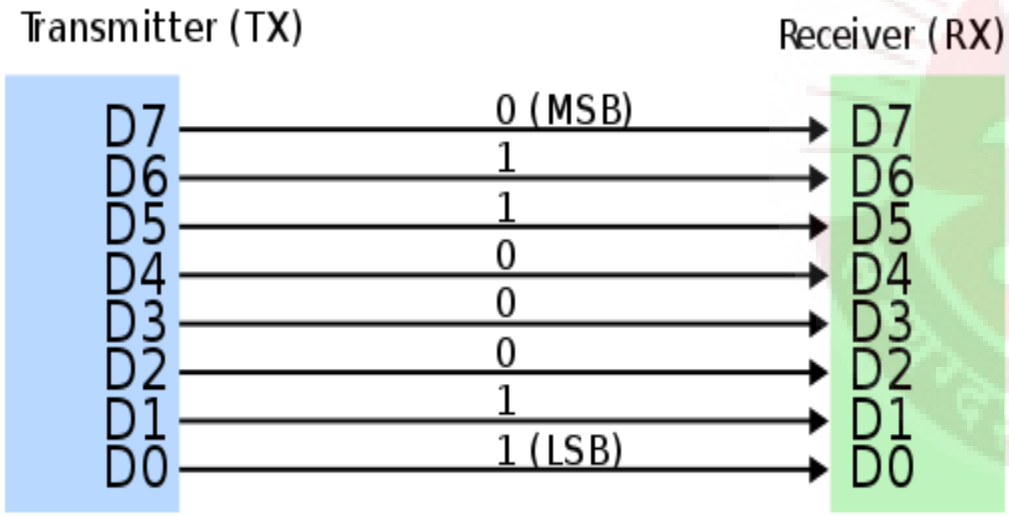
Facilities in 8051

Serial Communication using UART

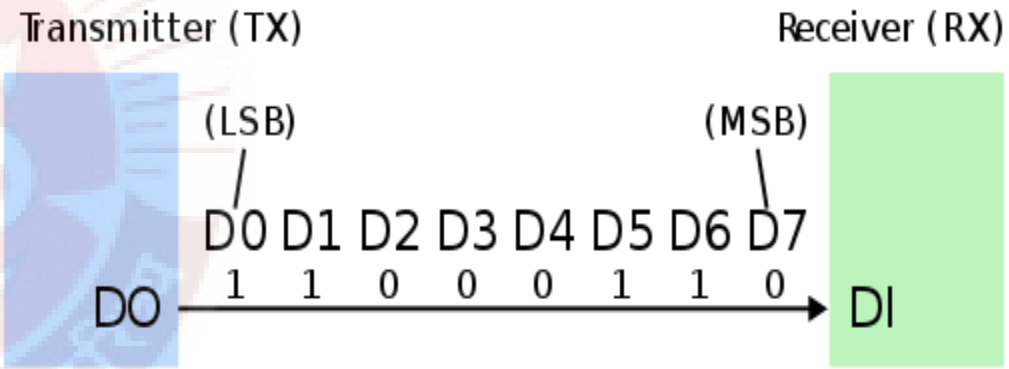
Universal Asynchronous Receiver and Transmitter

UART : Transmission Techniques

Parallel Transmission



Serial Transmission

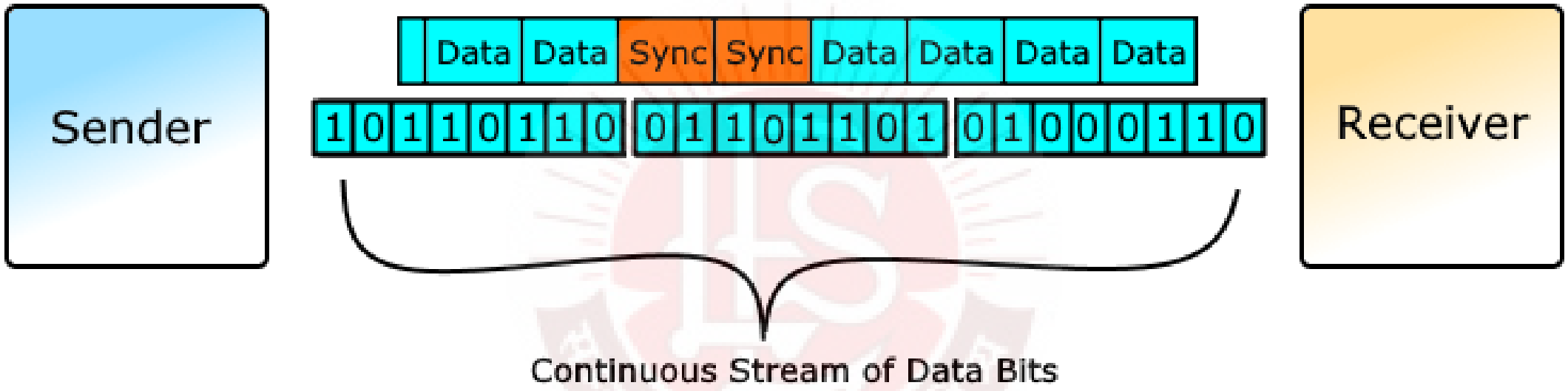


Tx

Rx

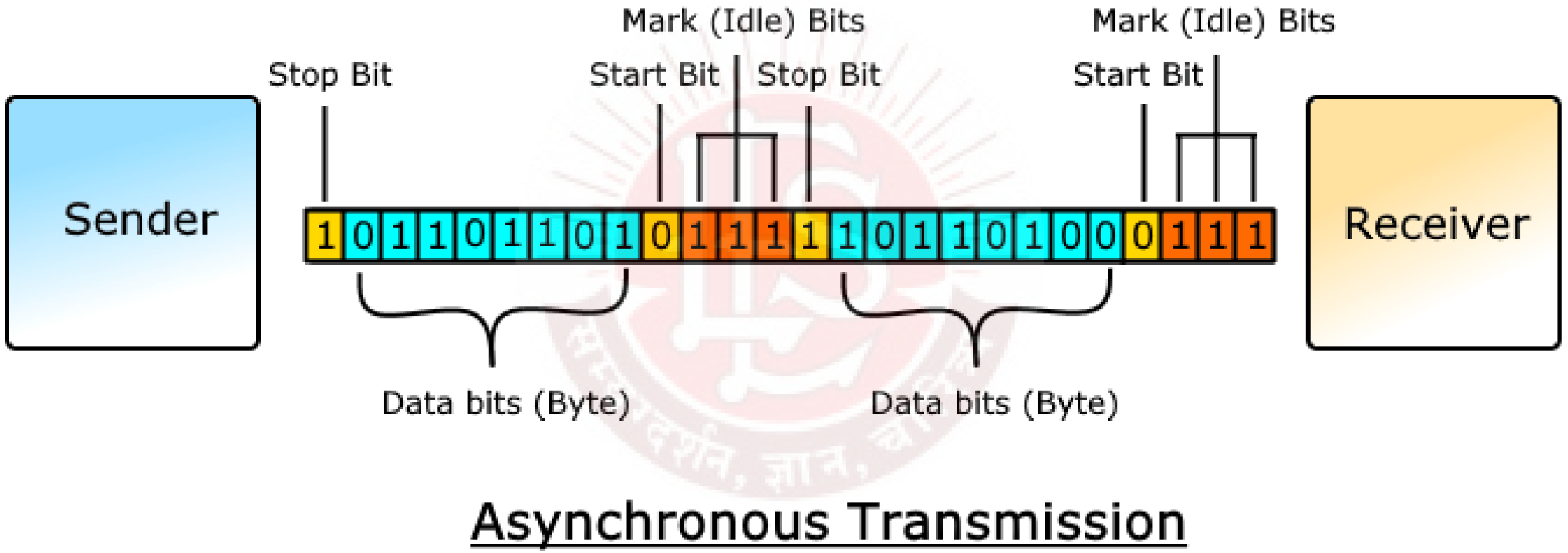
GND

Serial Transmission : **Synchronous** Data Transfer Techniques



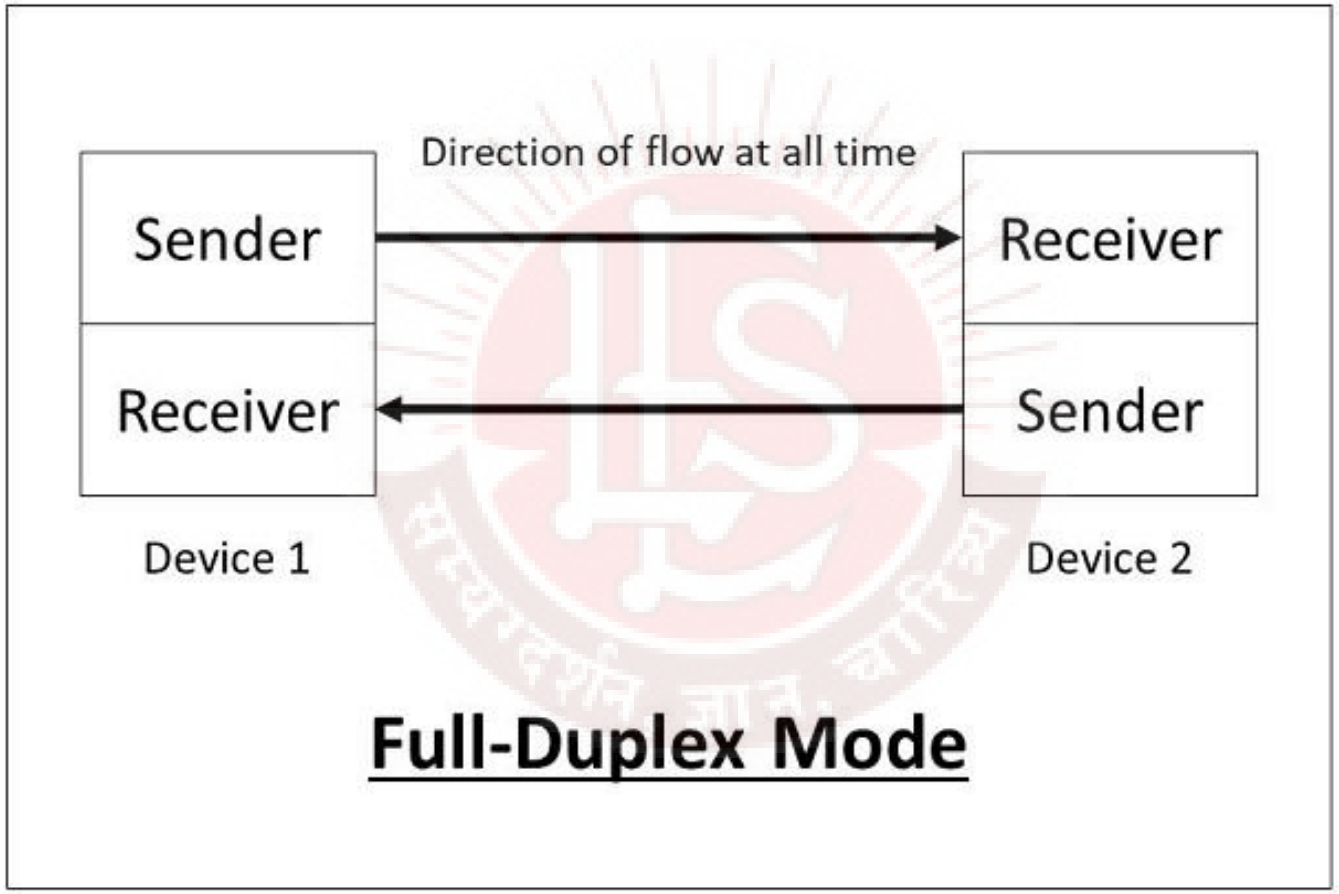
Synchronous Transmission

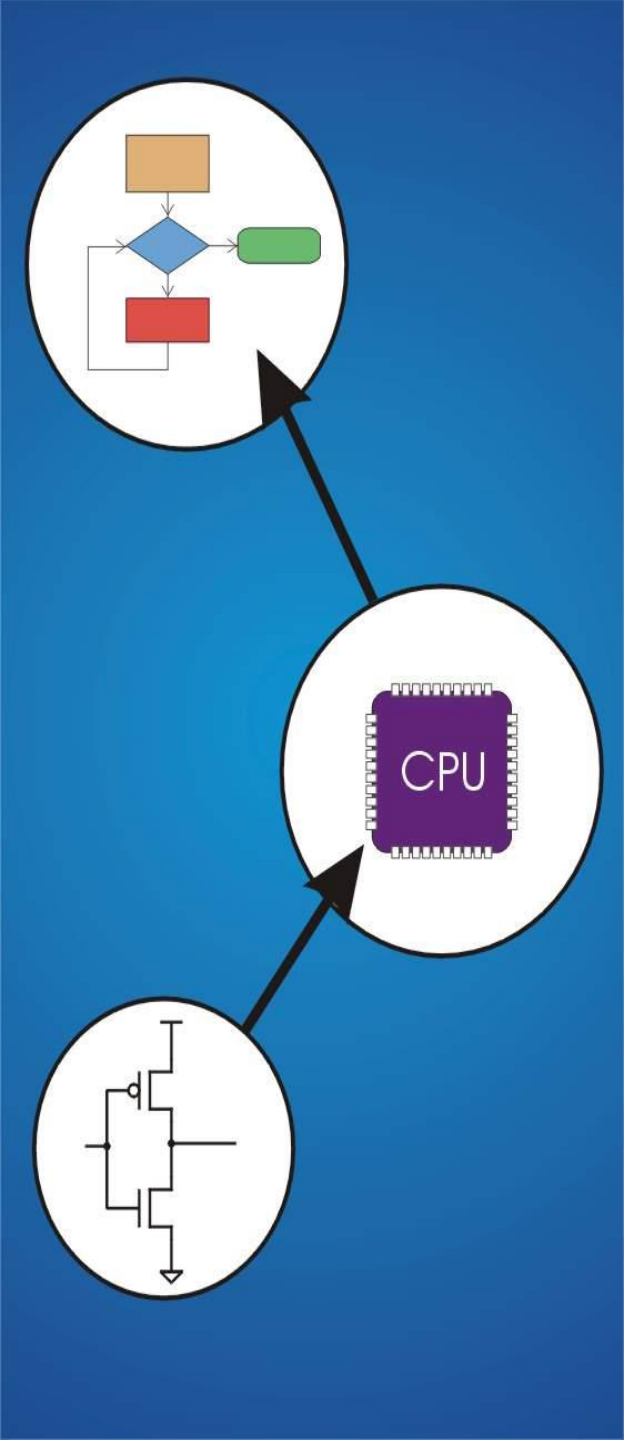
Serial Transmission : Asynchronous Data Transfer Techniques



Asynchronous Transmission

Asynchronous Transmission : Modes



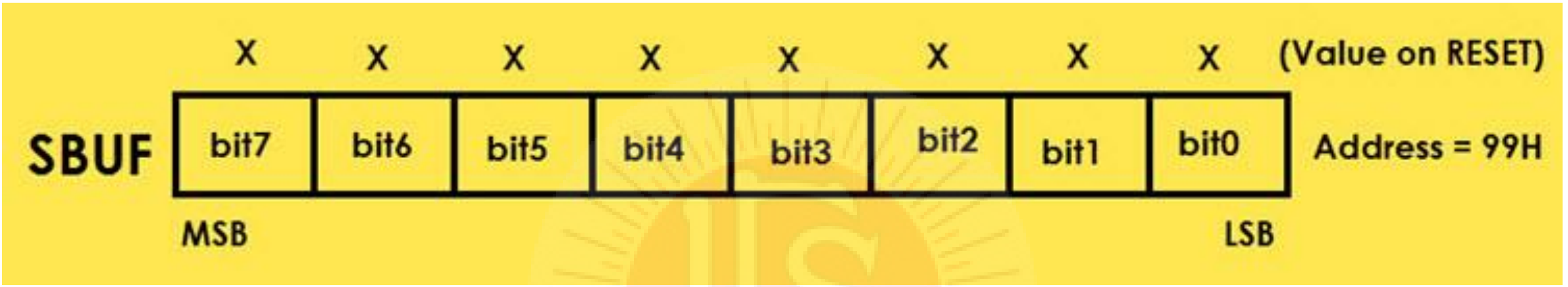


UNIT – III

Facilities in 8051

Microcontroller Registers used for Serial Communication

Serial Communication : Data Register

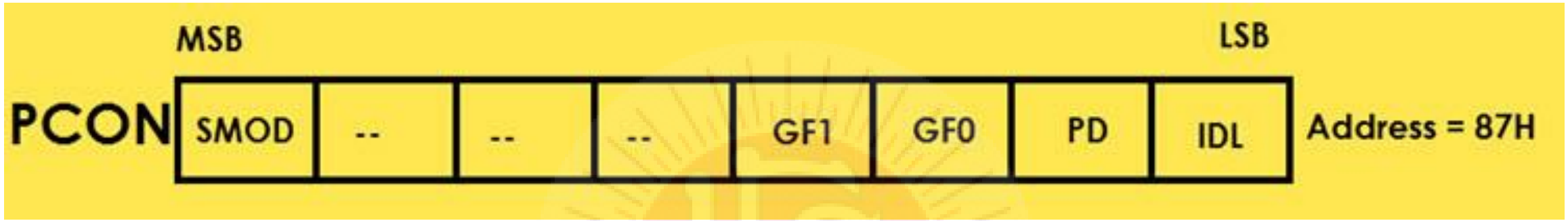


Serial Communication : Control Register

Baud Rate	SM0	SM1	Mode	Description
$f_{osc}/12$	0	0	Mode 0	8 bit shift Register
Timer 1	0	1	Mode 1	8 bit UART
$f_{osc}/32$ or $f_{osc}/64$	1	0	Mode 2	9 bit Shift Register
Timer 1	1	1	Mode 3	9 bit UART

Serial Communication : Control Register

Serial Communication : Control Register



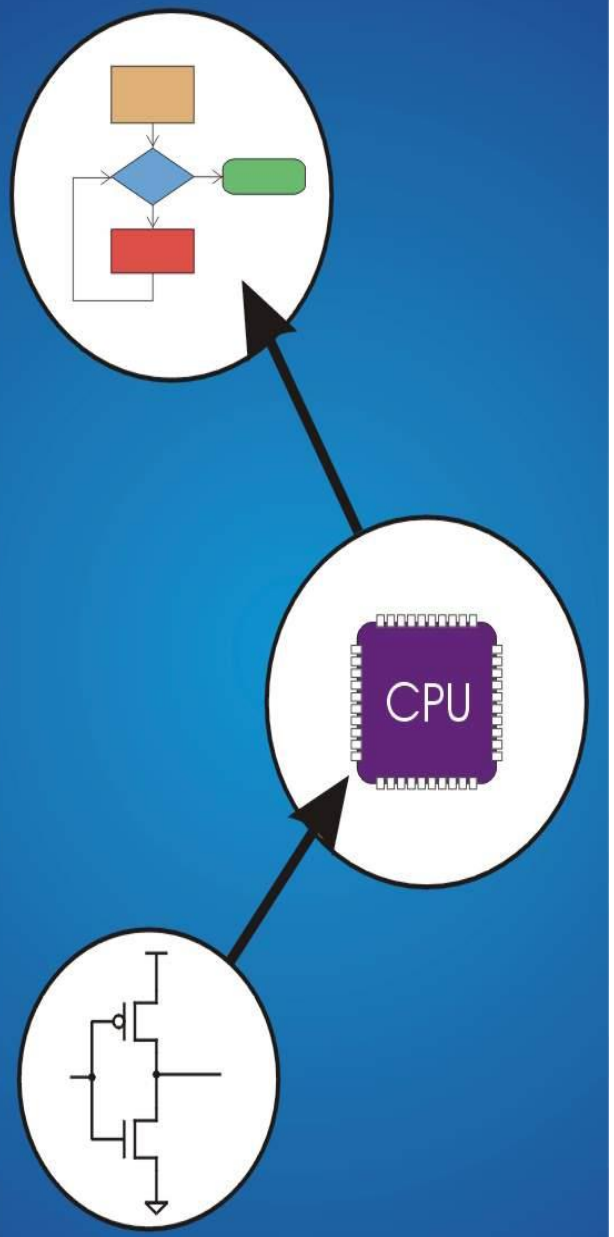
Serial Communication : SCON Register Content

0 1 0 1 0 0 0 0

SCON = 50 H

8 bit Data

Transmit and Receive

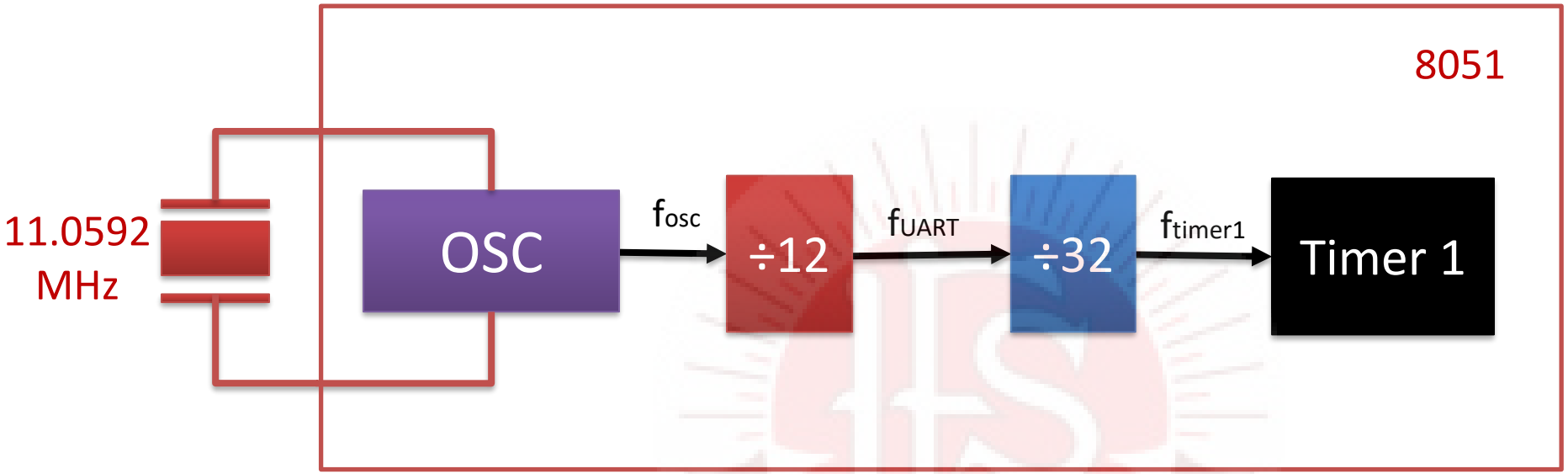


UNIT – III

Facilities in 8051

Baud Rate Generation using Timer 1

Baud Rate Generation :



$$fosc = fxtal = 11.0592\ x\ 10^6$$

$$fUART = \frac{fosc}{12} = \frac{11.0592\ x\ 10^6}{12}$$

$$fUART = 921.6\ x\ 10^3$$

$$ftimer1 = \frac{fUART}{32} = \frac{921.6\ x\ 10^3}{32}$$

$$\mathrm{f}timer1 = 28.8\ x\ 10^3$$

$$ftimer1 = 28{,}800\ Hz$$

Baud Rate Generation :

For Baud rate of 2400

$$2400 = \frac{28{,}800}{12}$$

TH1 = TL1 = -12 = F4 H

For Baud rate of 9600

$$9600 = \frac{28{,}800}{3}$$

TH1 = TL1 = -3 = FD H

Serial Communication : Program

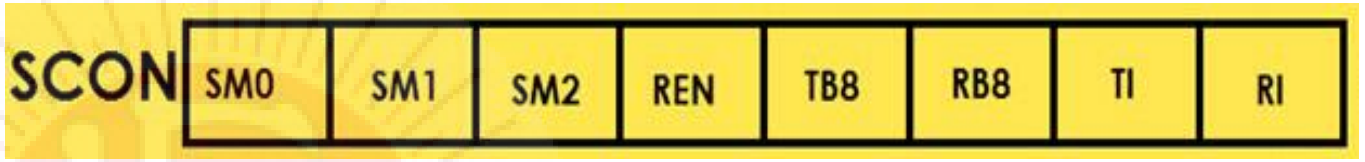
Write a program to transmit 'A' continuously. Use crystal frequency of **11.0592 MHz**

$$ftimer1 = 28{,}800\ Hz$$

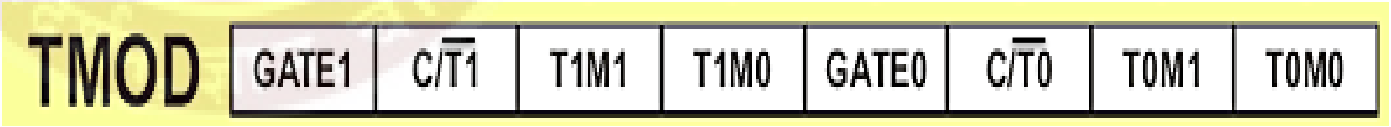
For Baud rate of 9600

$$9600 = \frac{28{,}800}{3}$$

TH1 = **TL1** = **-3** = **FD H**



SCON = 50 H

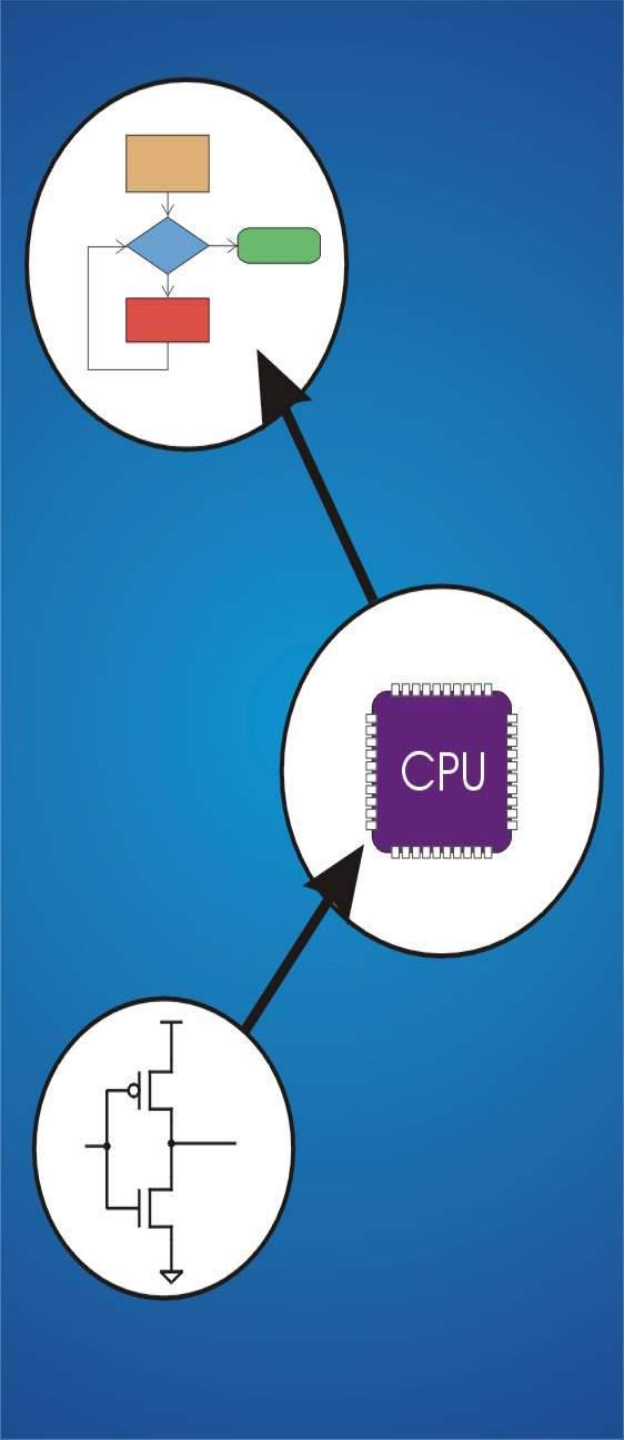


TMOD = 20 H

Serial Communication Program : TH1 = TL1 = FD H SCON = 50 H TMOD = 20 H

```
        $INCLUDE  (REG51.INC)

        ORG       0000 H

        MOV       SCON, #50H
        MOV       TMOD, #20H
        MOV       TH1, #-3
        MOV       TL1, #-3
        SETB      TR1

AGAIN:  MOV       SBUF,  #'A'

WAIT:   JNB       TI,  WAIT
        CLR       TI
        SJMP      AGAIN

        END
```

```
#include <reg51.h>

void main() {
        SCON = 0x50;
        TMOD = 0x20;
        TH1 = -3;
        TL1 = -3;
        TR1 = 1;

        while(1)        {
            SBUF = 'A';
            while(TI == 0);
            TI = 0;
        }
}
```

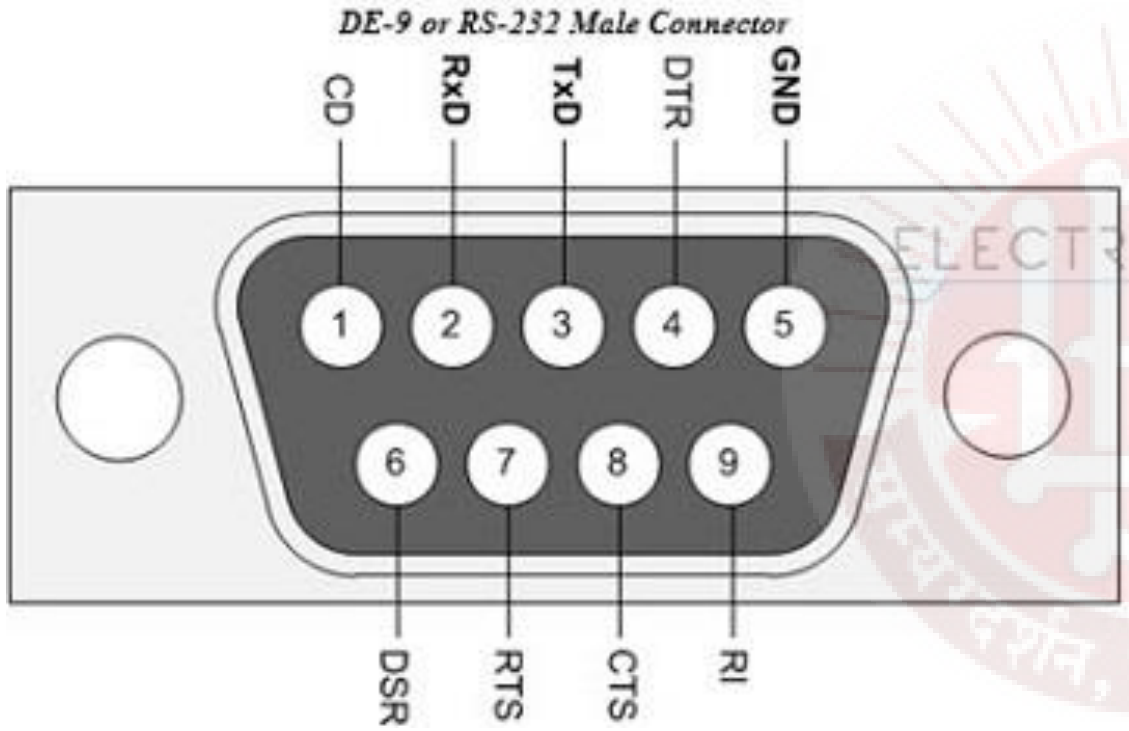


UNIT – III

Facilities in 8051

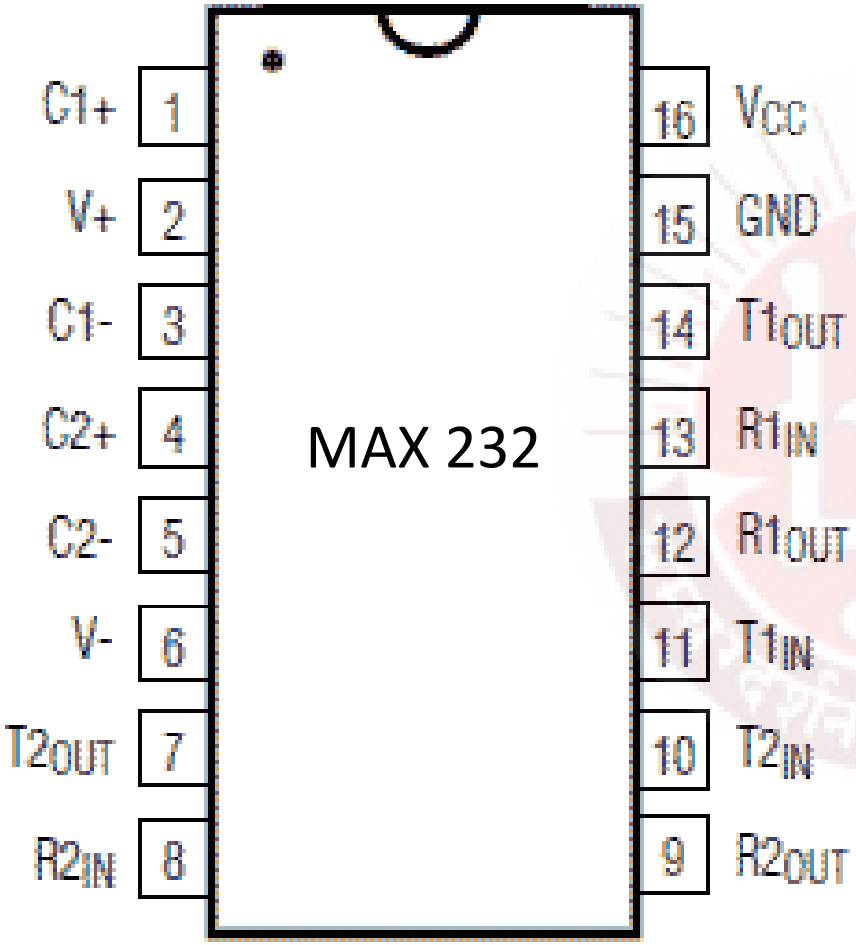
RS232

RS-232 :

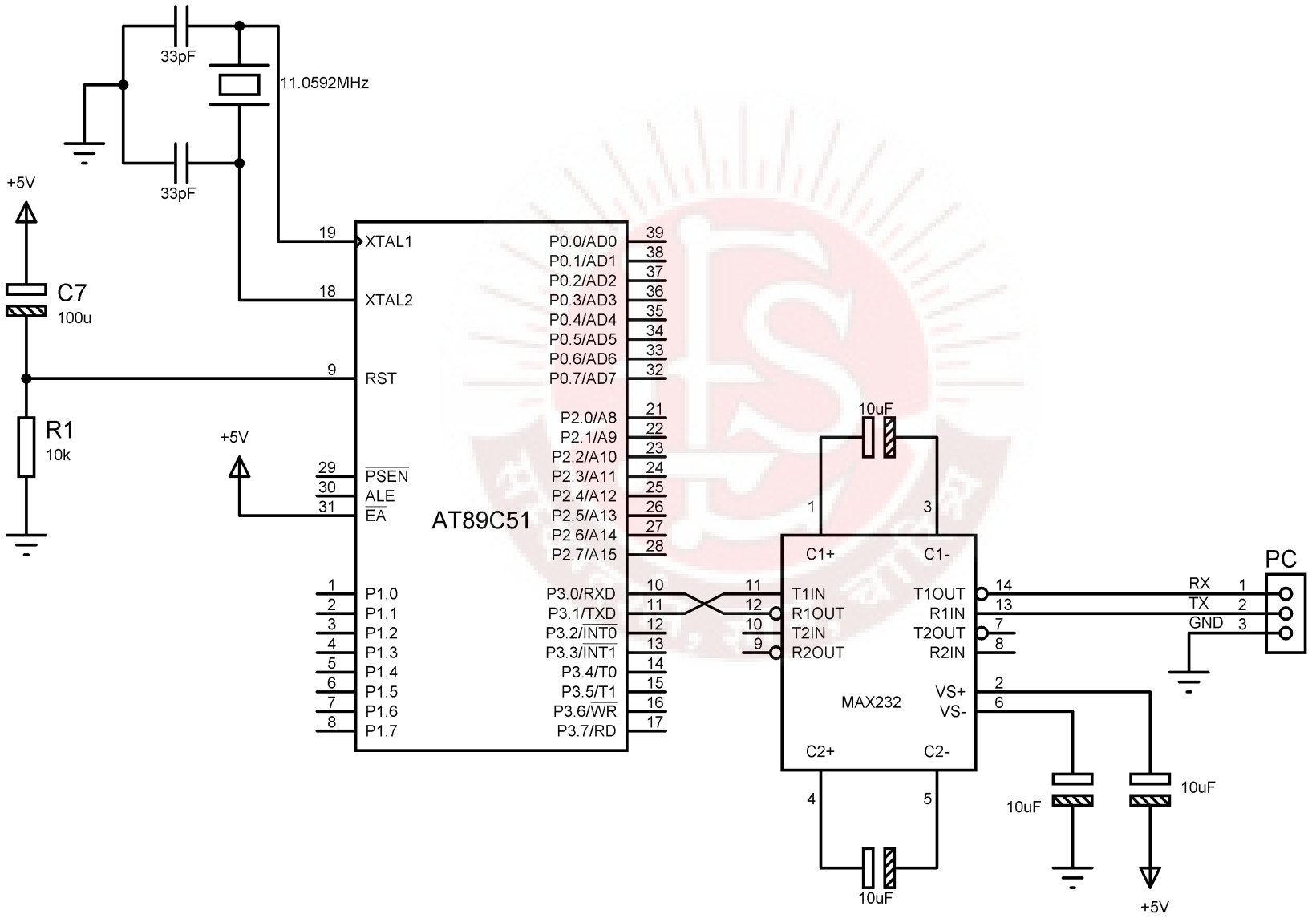


Pin	Description
1	Data Carrier Detect (DCD)
2	Received Data (RxD)
3	Transmitted Data (TxD)
4	Data Terminal Ready (DTR)
5	Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

MAX-232 :


MAX 232
C1+ 1
V+ 2
C1- 3
C2+ 4
C2- 5
V- 6
T2OUT 7
R2IN 8
16 VCC
15 GND
14 T1OUT
13 R1IN
12 R1OUT
11 T1IN
10 T2IN
9 R2OUT

Interfacing Circuit :



Course Contents

1. Introduction to Microcontroller
2. 8051 Instruction Set
3. Facilities in 8051
4. Interfacing Methods

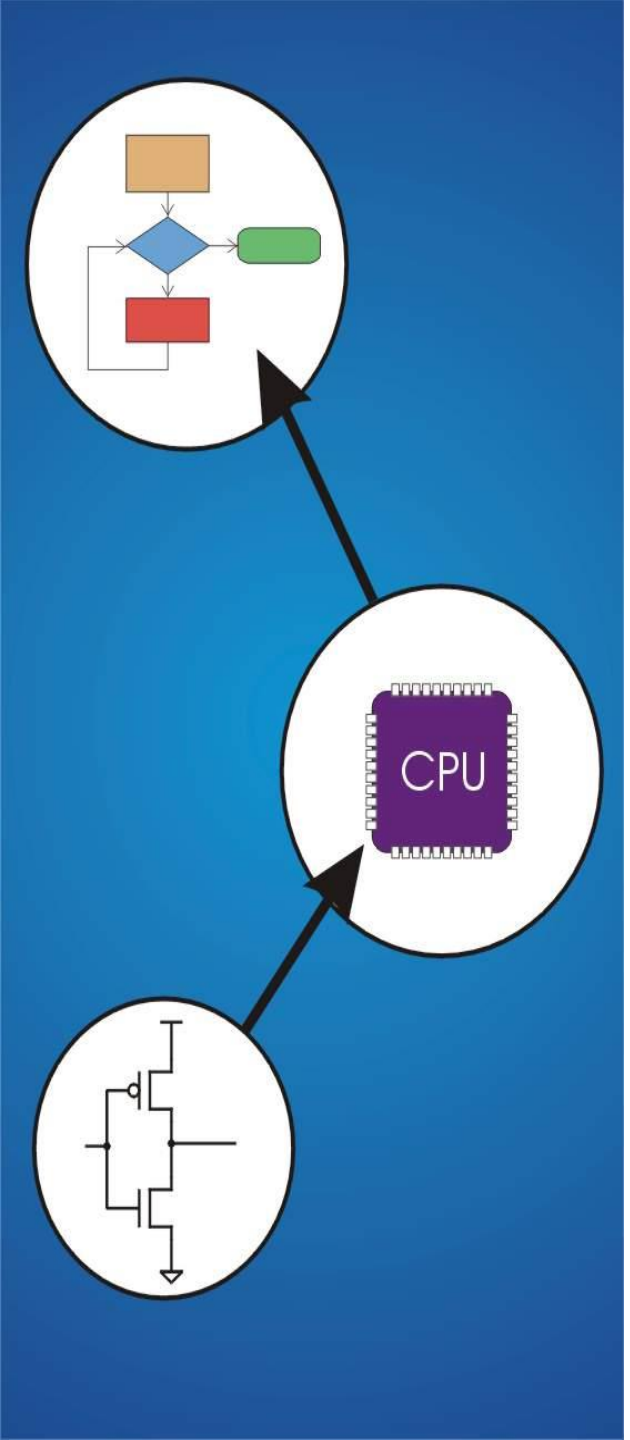
Course Contents

INTERFACING METHODES

Interfacing with 8051:
LED, Switch, Relay, Opto-coupler,
Thumb wheel switch and Seven segment display.

Stepper Motor, DC motor (PWM), LCD (16 X 2),

with respective programming in assembly language OR embedded C for all.



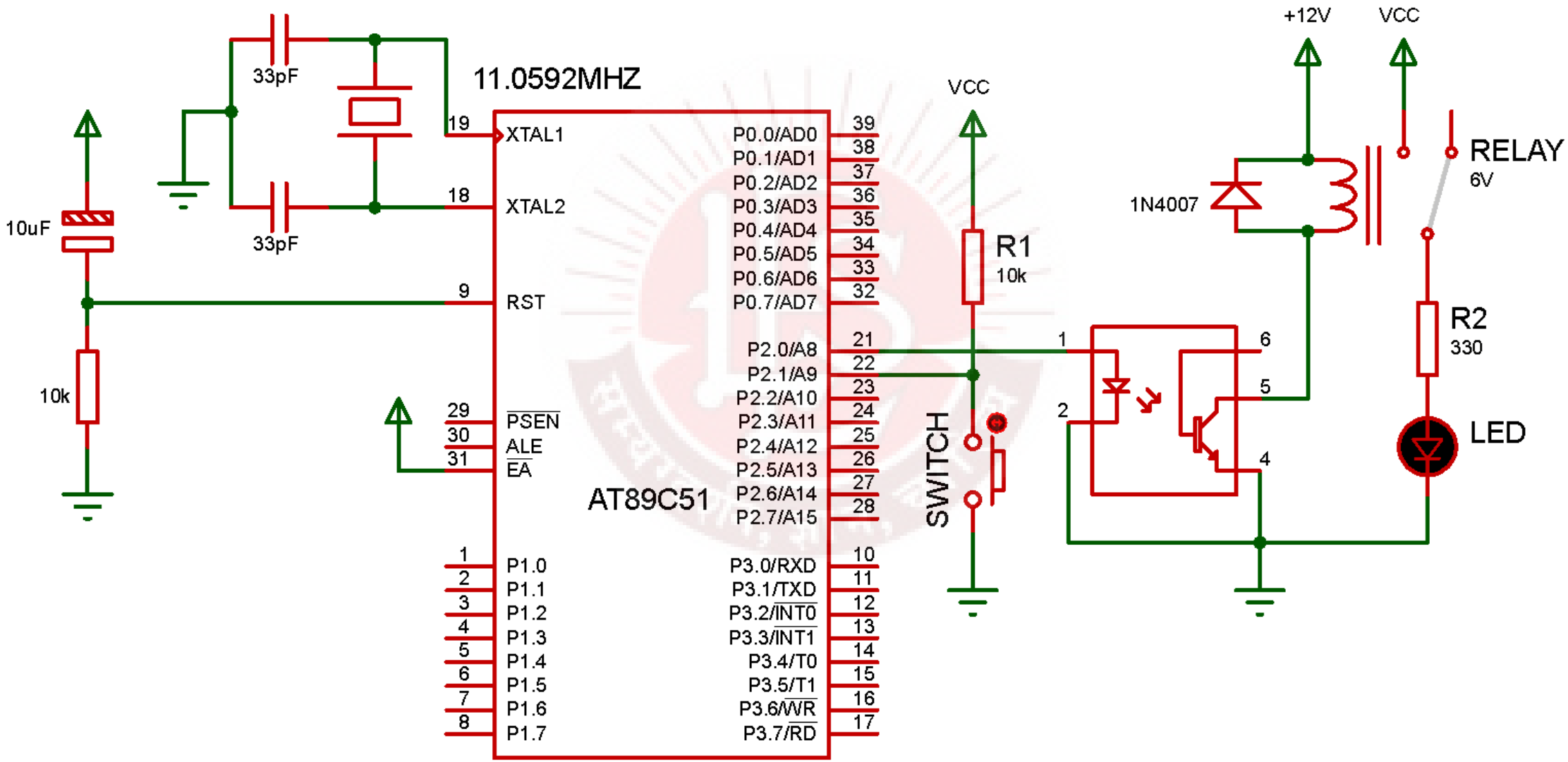
UNIT – IV

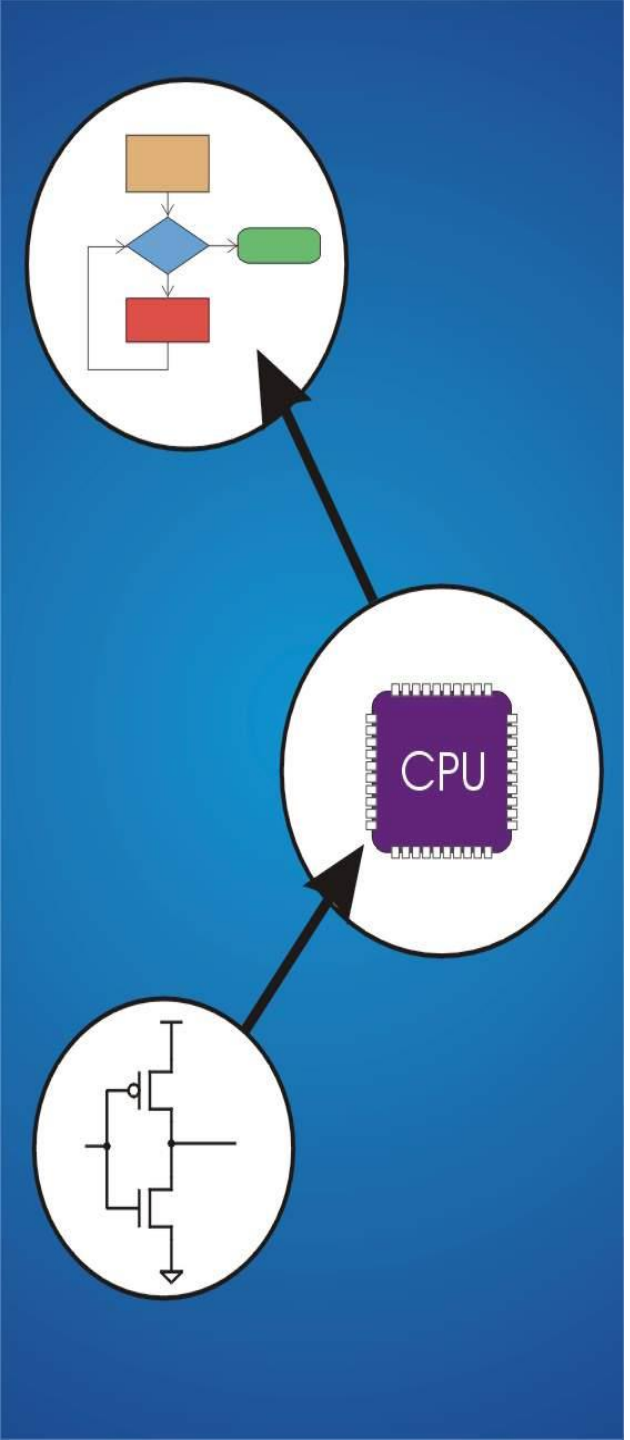
Interfacing Methods

Interfacing

LED, Relay, Optocoupler, Switch

Interfacing : LED, Relay, Optocoupler, Switch





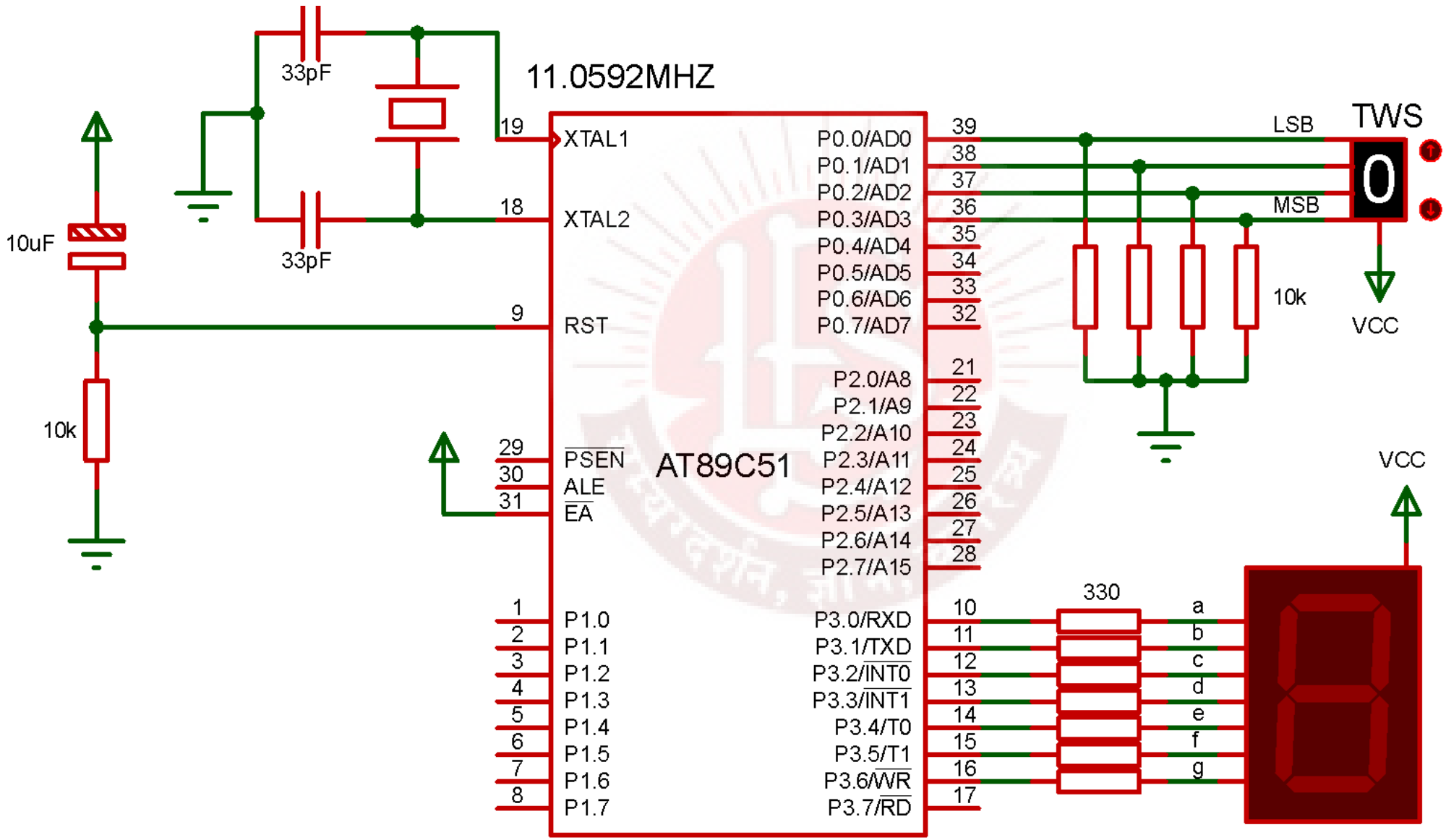
UNIT – IV

Interfacing Methods

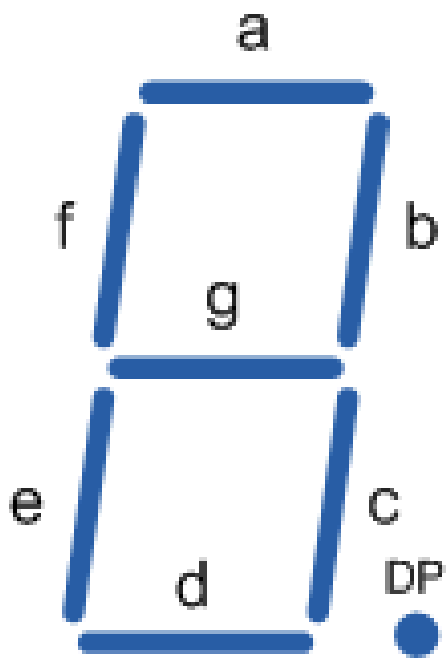
Interfacing

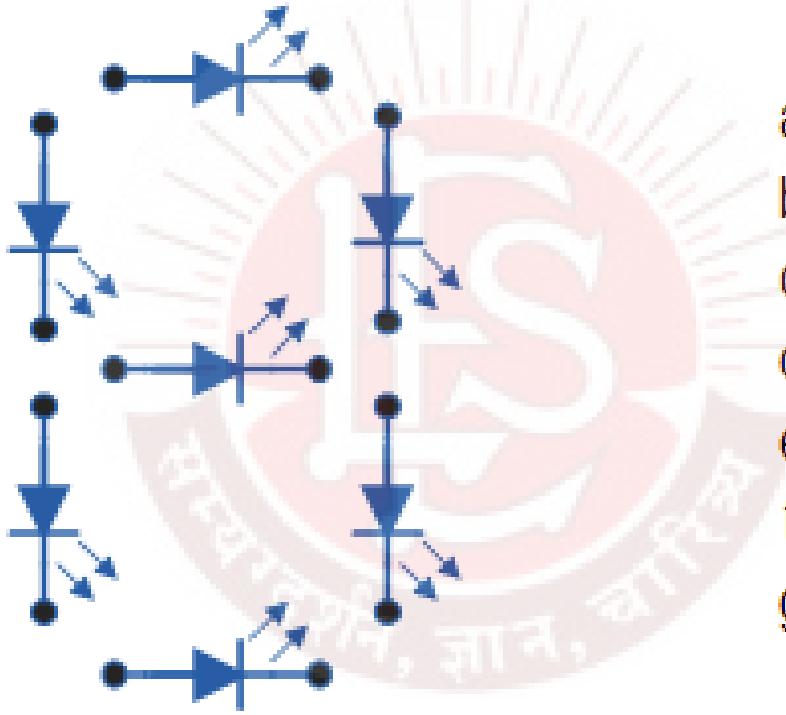
Thumb Wheel Switch and Seven Segment Display

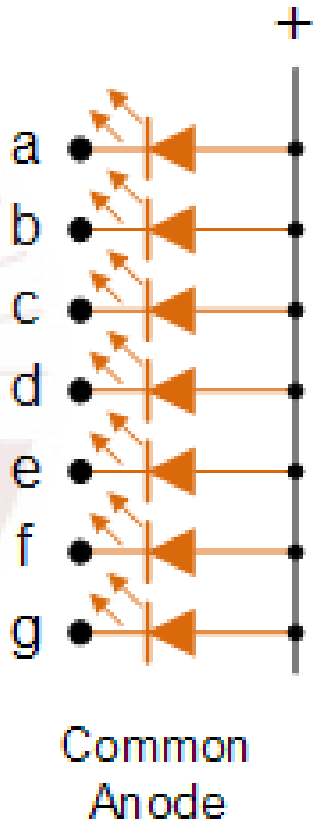
7 Segment : Interfacing



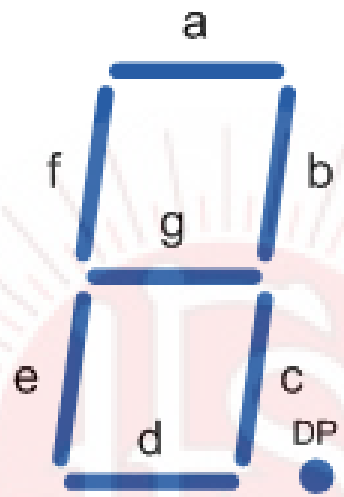
7 Segment : Theory

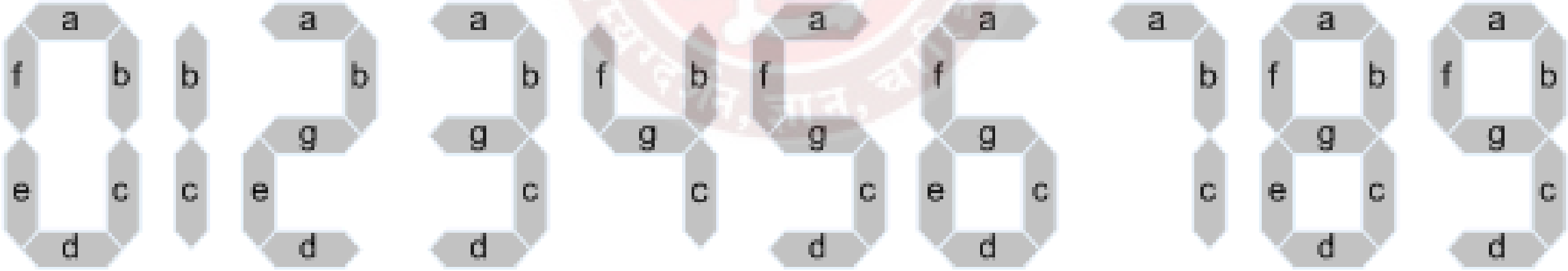






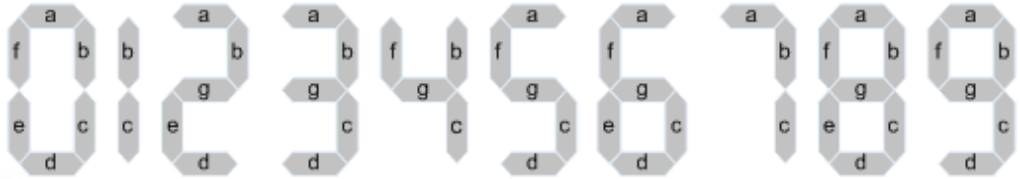
7 Segment : Theory

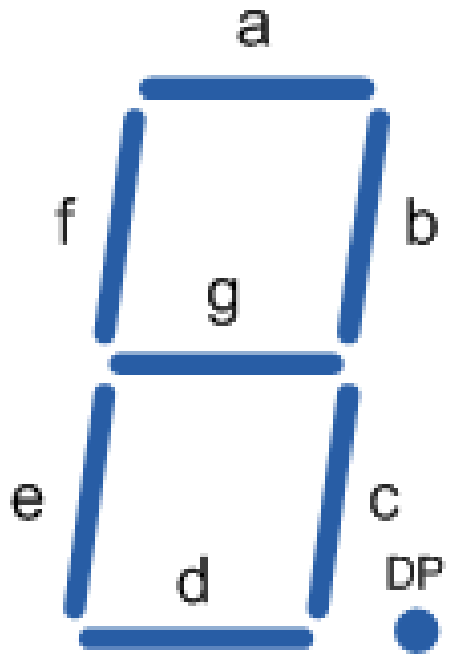




7 Segment : Design - CA

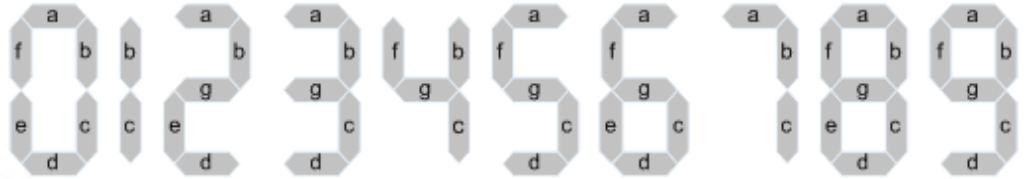
Digit	Outputs								CODE
	dp	g	f	e	d	c	b	a	
0		1							
1		1	1	1	1			1	
2			1			1			
3			1	1					
4				1	1			1	
5				1			1		
6							1		
7		1	1	1	1				
8									
9				1					

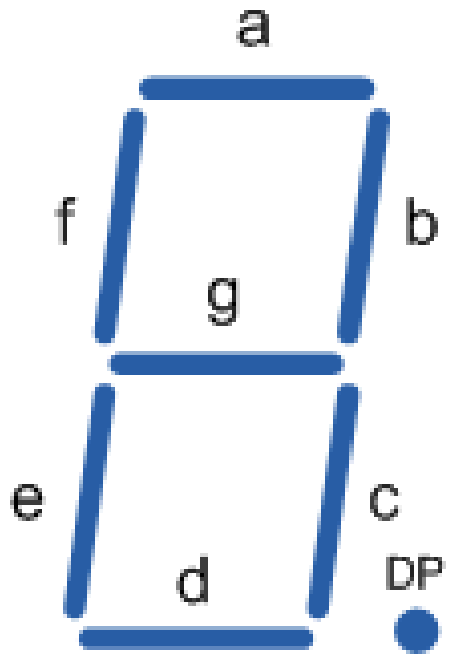




7 Segment : Design - CA

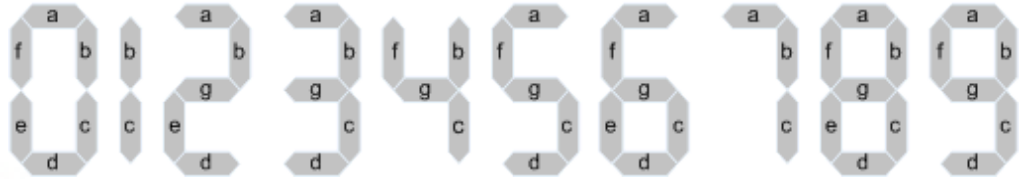
Digit	Outputs								CODE
	dp	g	f	e	d	c	b	a	
0	0	1	0	0	0	0	0	0	
1	0	1	1	1	1	0	0	1	
2	0	0	1	0	0	1	0	0	
3	0	0	1	1	0	0	0	0	
4	0	0	0	1	1	0	0	1	
5	0	0	0	1	0	0	1	0	
6	0	0	0	0	0	0	1	0	
7	0	1	1	1	1	0	0	0	
8	0	0	0	0	0	0	0	0	
9	0	0	0	1	0	0	0	0	

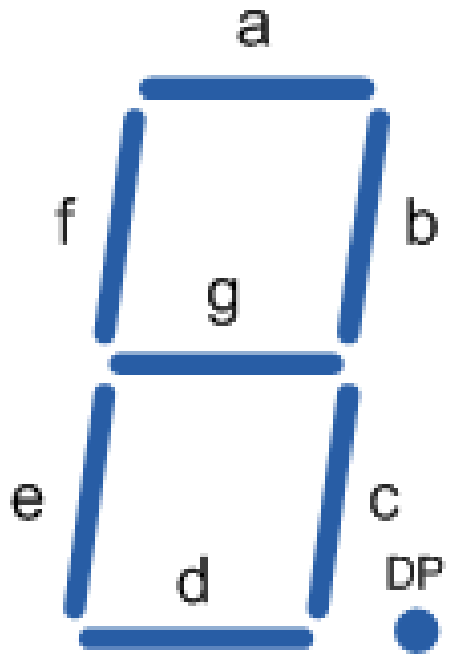


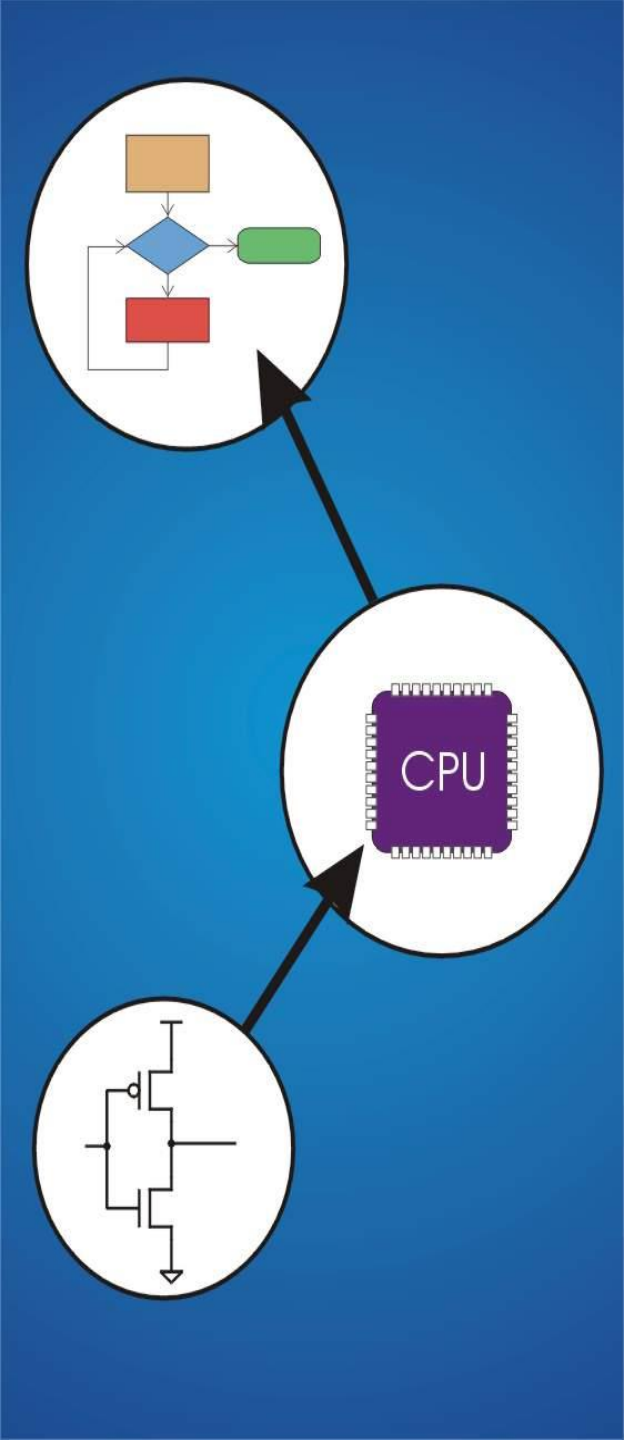


7 Segment : Design - CA

Digit	Outputs								CODE
	dp	g	f	e	d	c	b	a	
0	0	1	0	0	0	0	0	0	40
1	0	1	1	1	1	0	0	1	79
2	0	0	1	0	0	1	0	0	24
3	0	0	1	1	0	0	0	0	30
4	0	0	0	1	1	0	0	1	19
5	0	0	0	1	0	0	1	0	12
6	0	0	0	0	0	0	1	0	02
7	0	1	1	1	1	0	0	0	78
8	0	0	0	0	0	0	0	0	00
9	0	0	0	1	0	0	0	0	10







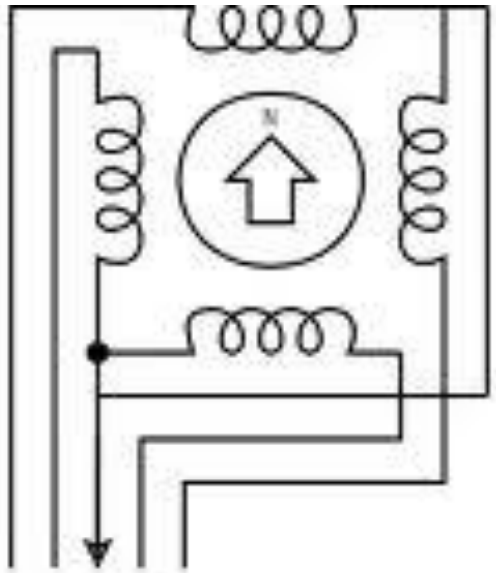
UNIT – IV

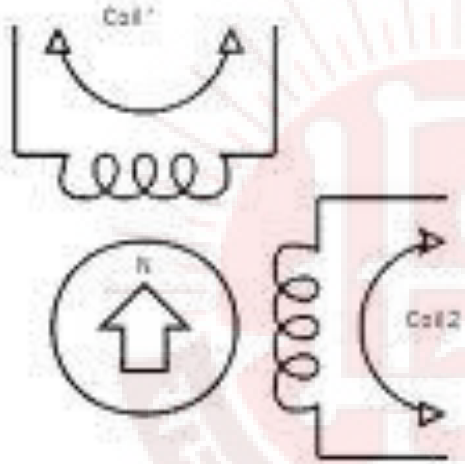
Interfacing Methods

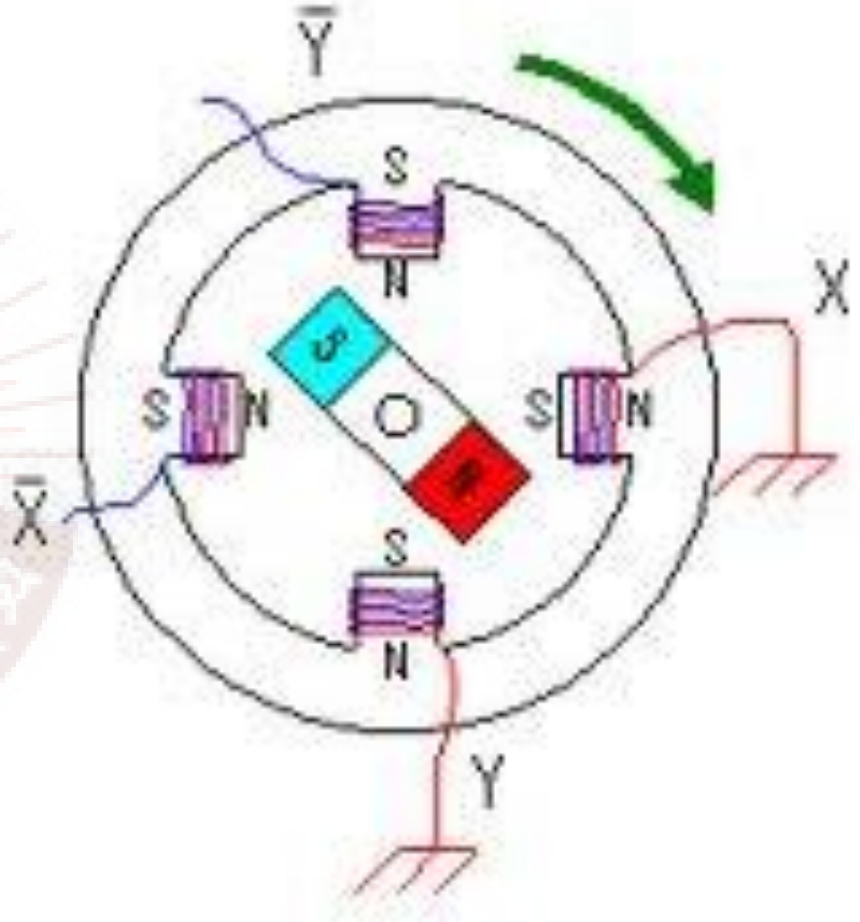
Interfacing

Stepper Motor

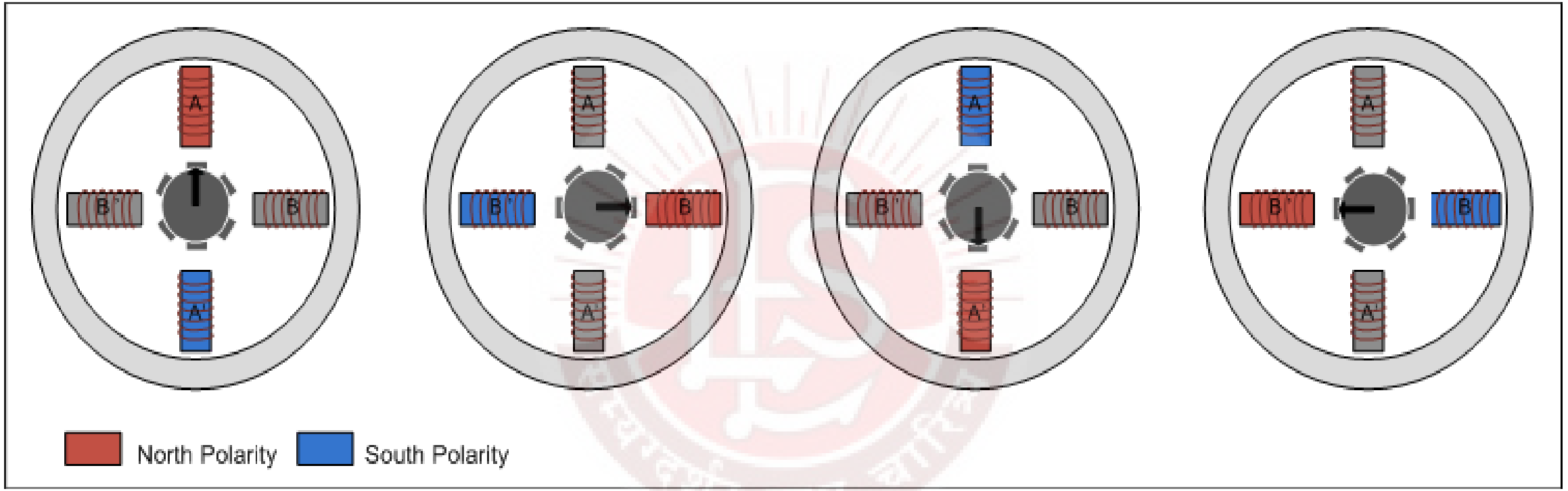
Stepper Motor : Theory







Stepper Motor : Working



Stepper Motor : How to rotate : Full Step Sequence

Clock wise	A	B	C	D	Anti Clock wise	Code
	1	0	0	1		9
	1	1	0	0		C
	0	1	1	0		6
	0	0	1	1		3

Stepper Motor : How to rotate : Half Step Sequence

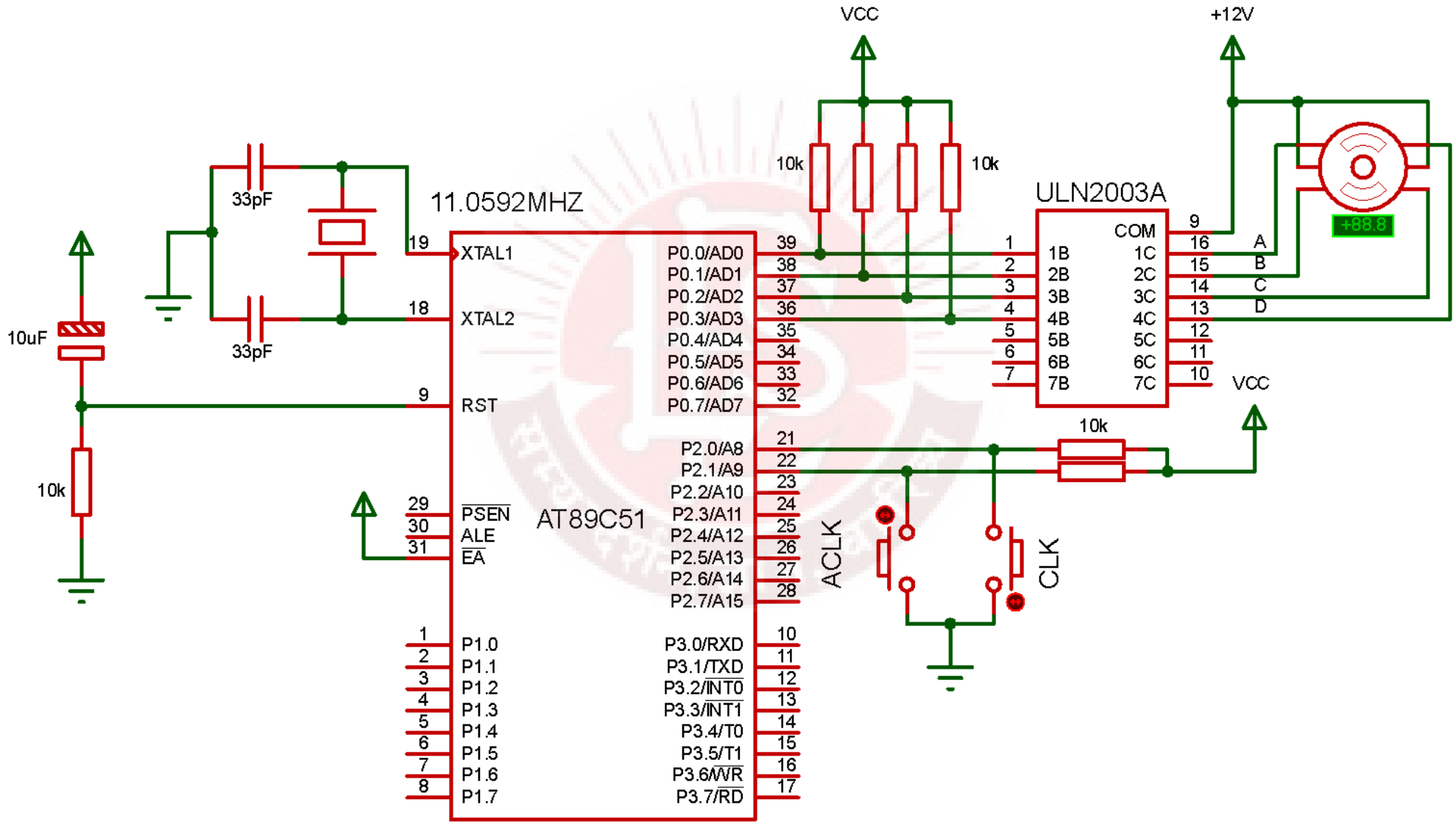
Clock wise	A	B	C	D	Anti Clock wise	Code
	1	0	0	1		9
	1	0	0	0		8
	1	1	0	0		C
	0	1	0	0		4
	0	1	1	0		6
	0	0	1	0		2
	0	0	1	1		3
	0	0	0	1		1

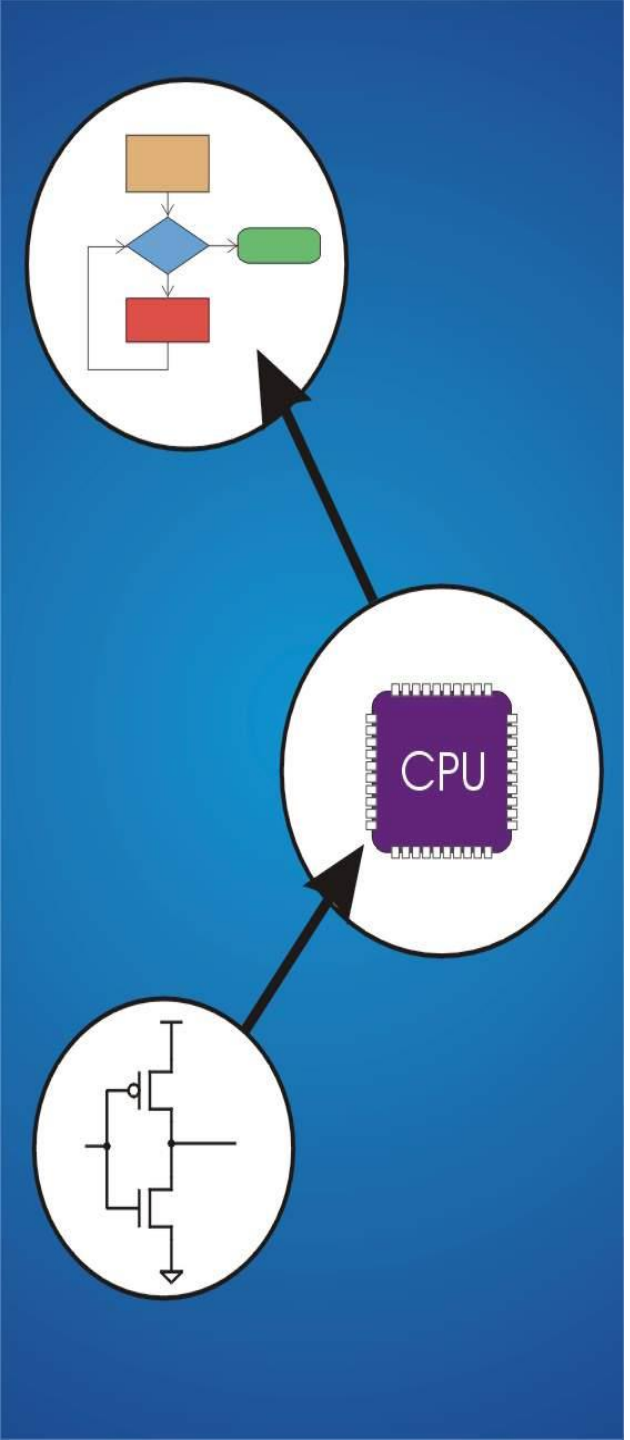
Stepper Motor : Specification

Step Angle	Steps per Revolution
0.72	500
1.8	200
2.0	180
2.5	144
90	4

Clock wise	A	B	C	D	Anti Clock wise
	1	0	0	1	
	1	1	0	0	
	0	1	1	0	
	0	0	1	1	

Stepper Motor : Interfacing





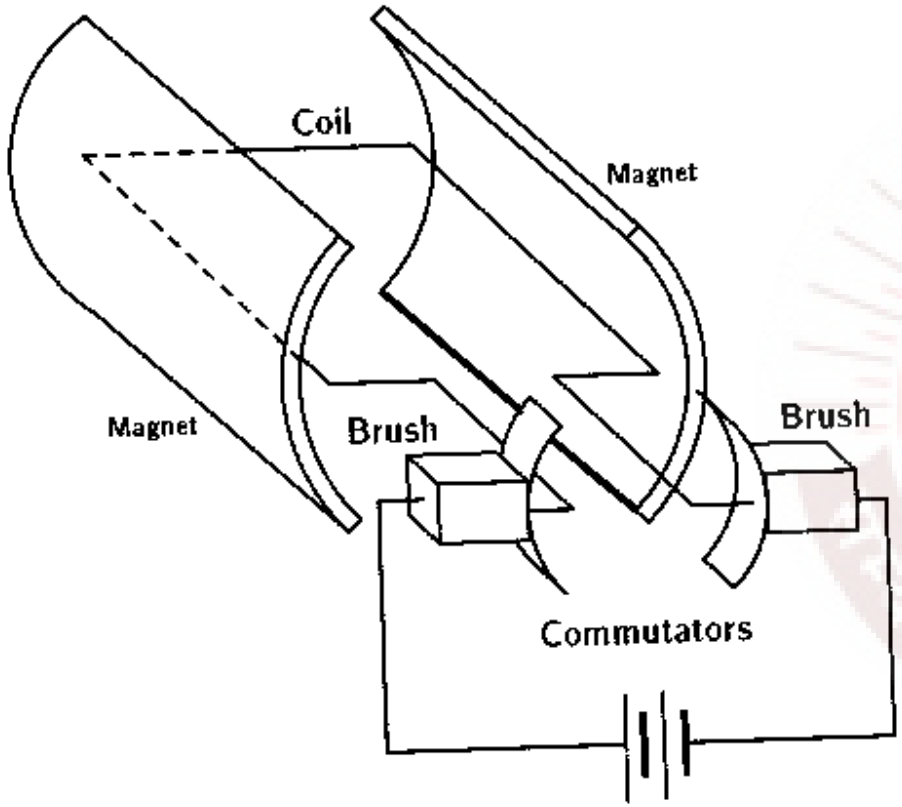
UNIT – IV

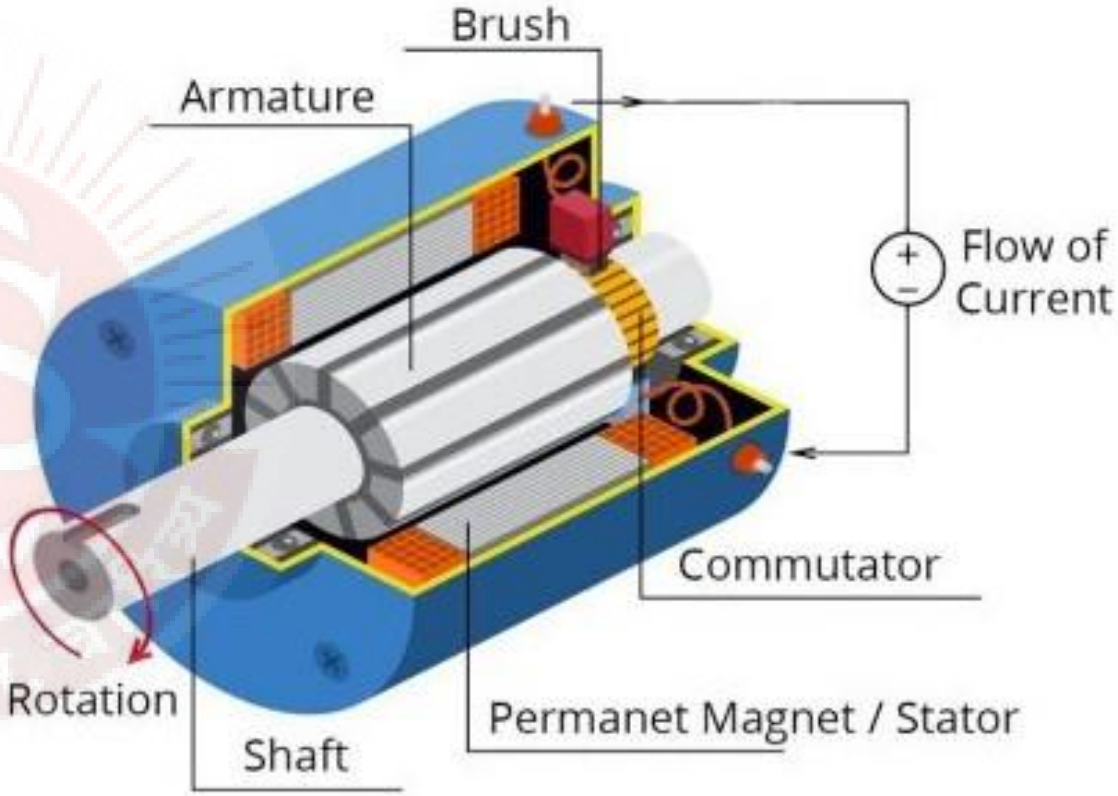
Interfacing Methods

Interfacing

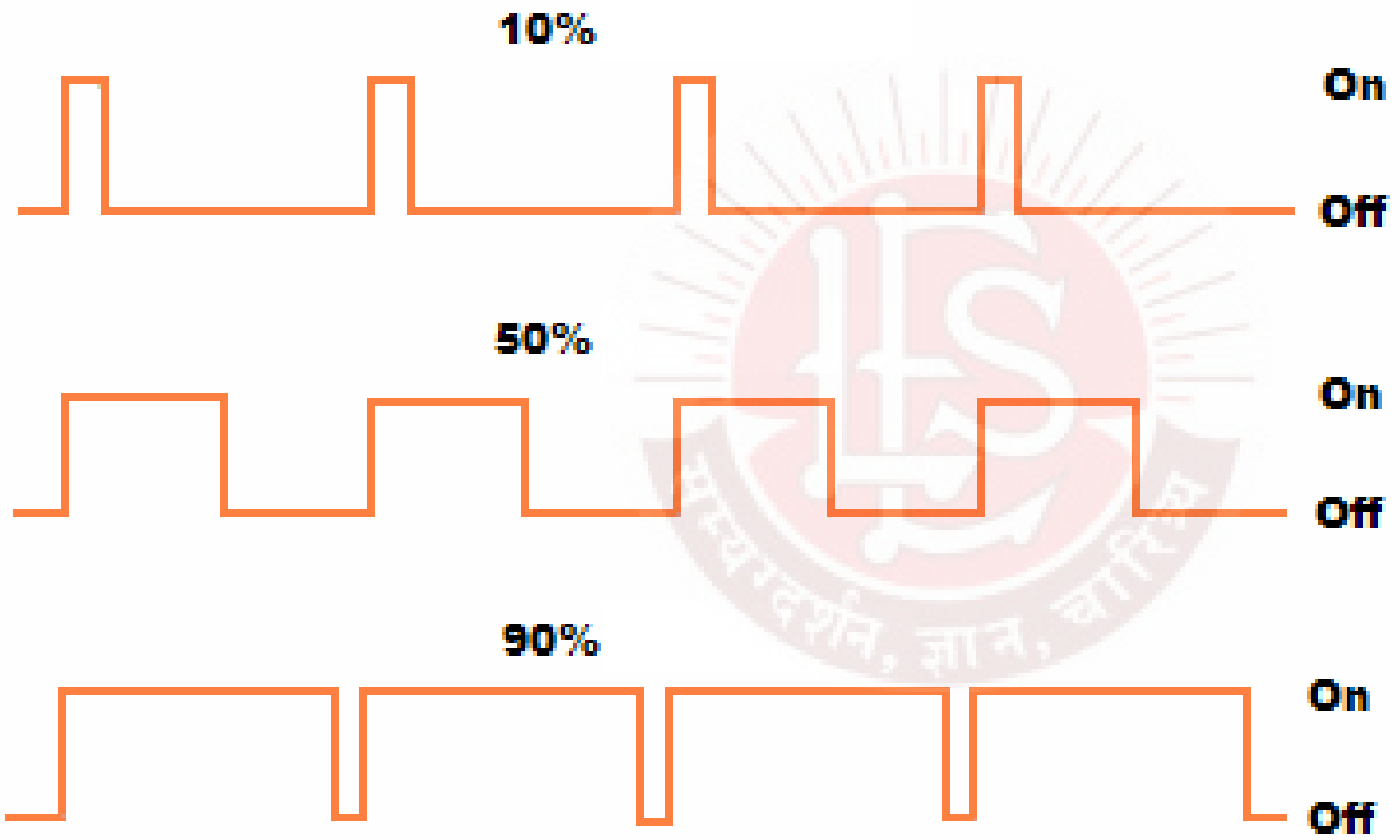
DC Motor

DC Motor : Working



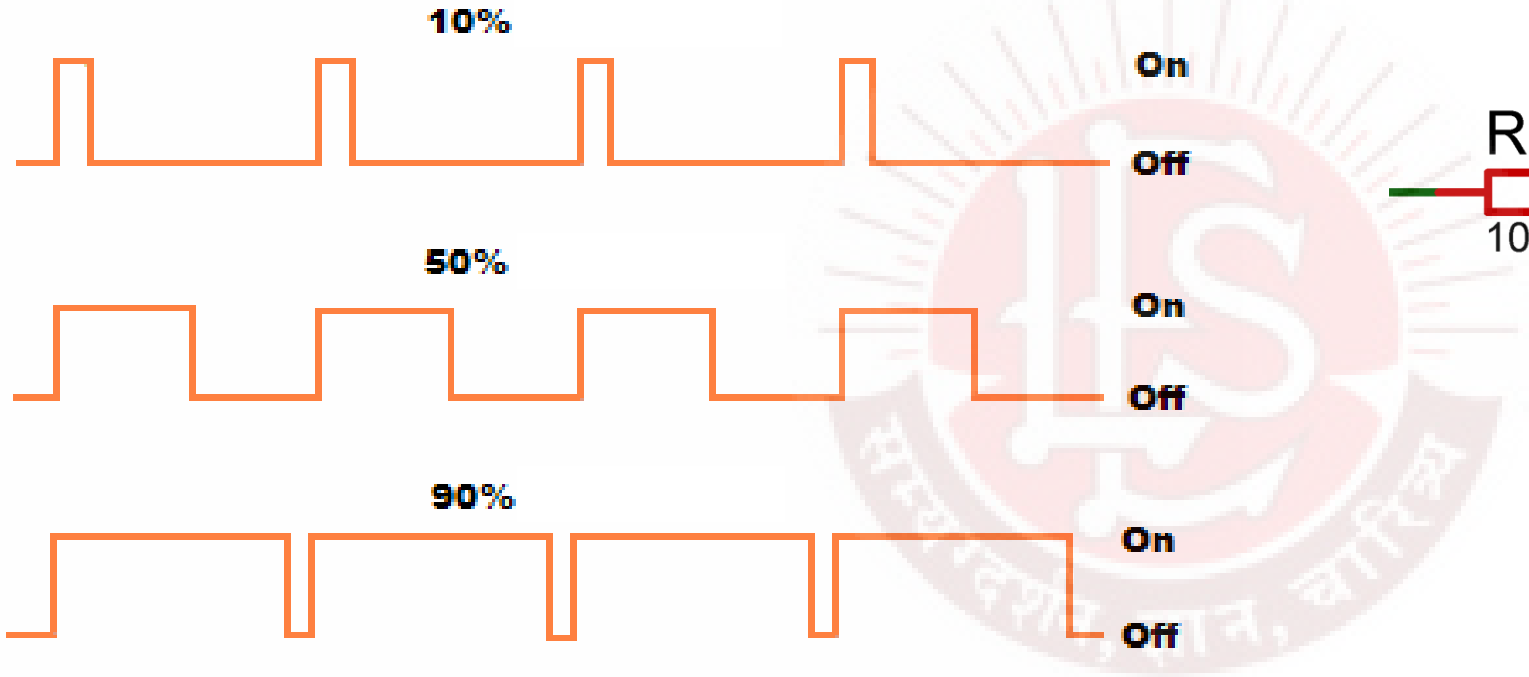


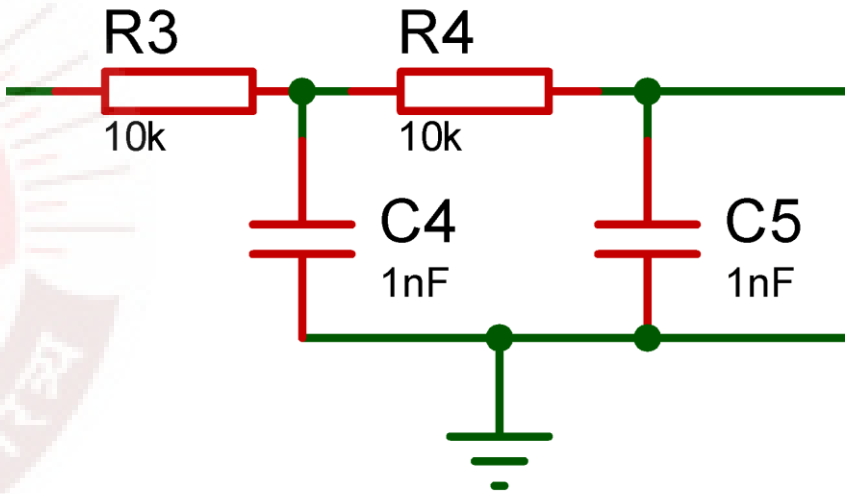
DC Motor : PWM



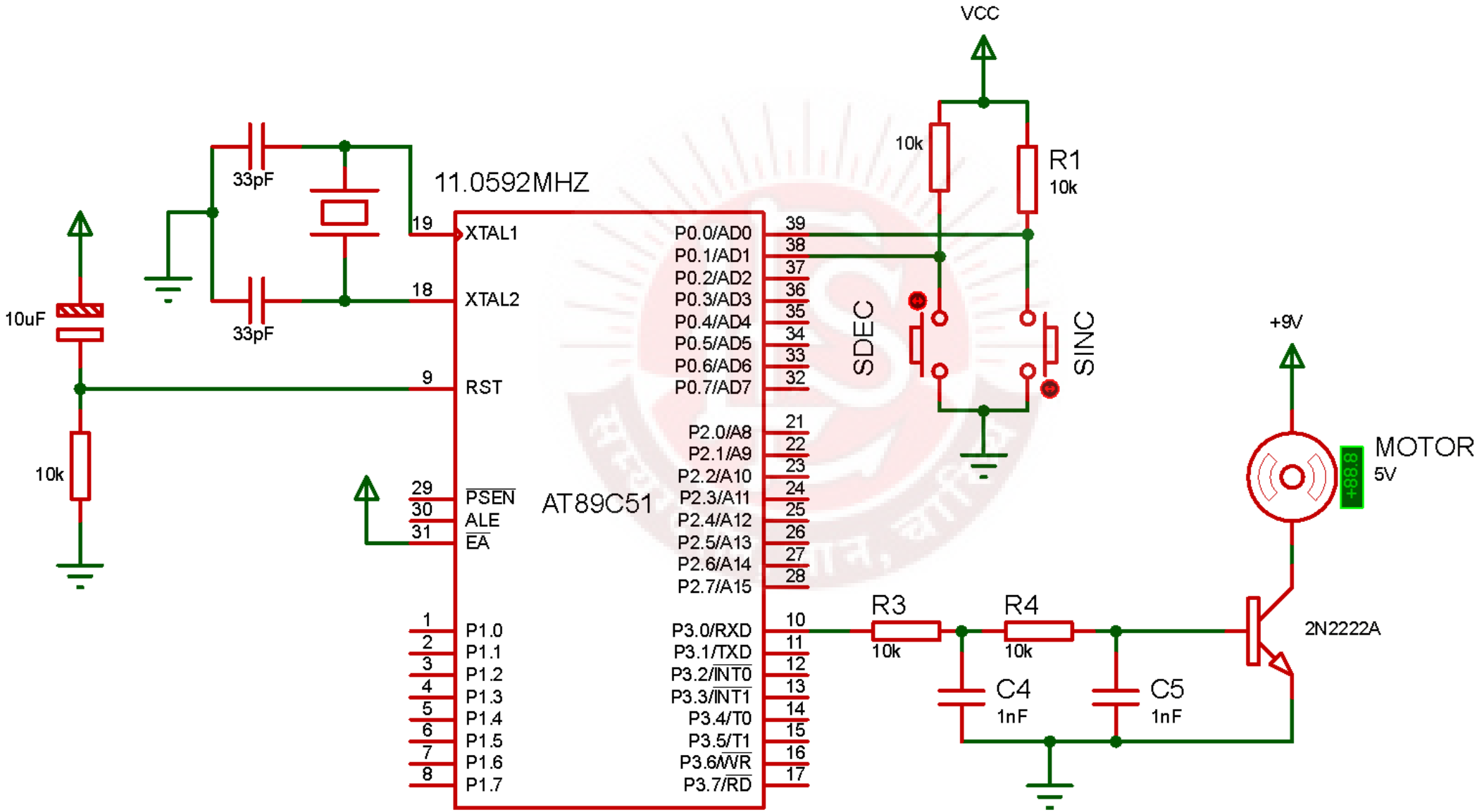
Duty Cycle	Voltage
10 %	0.5 V
50 %	2.5 V
90 %	4.5 V

DC Motor : PWM Frequency Filter





DC Motor : Interfacing



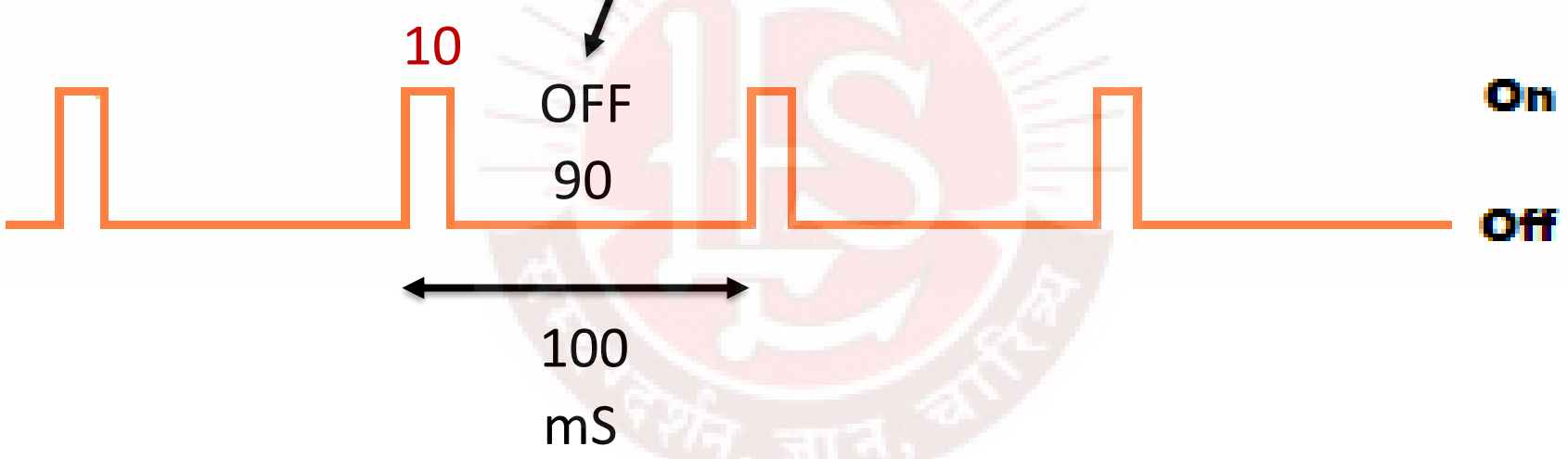
DC Motor : PWM Generation

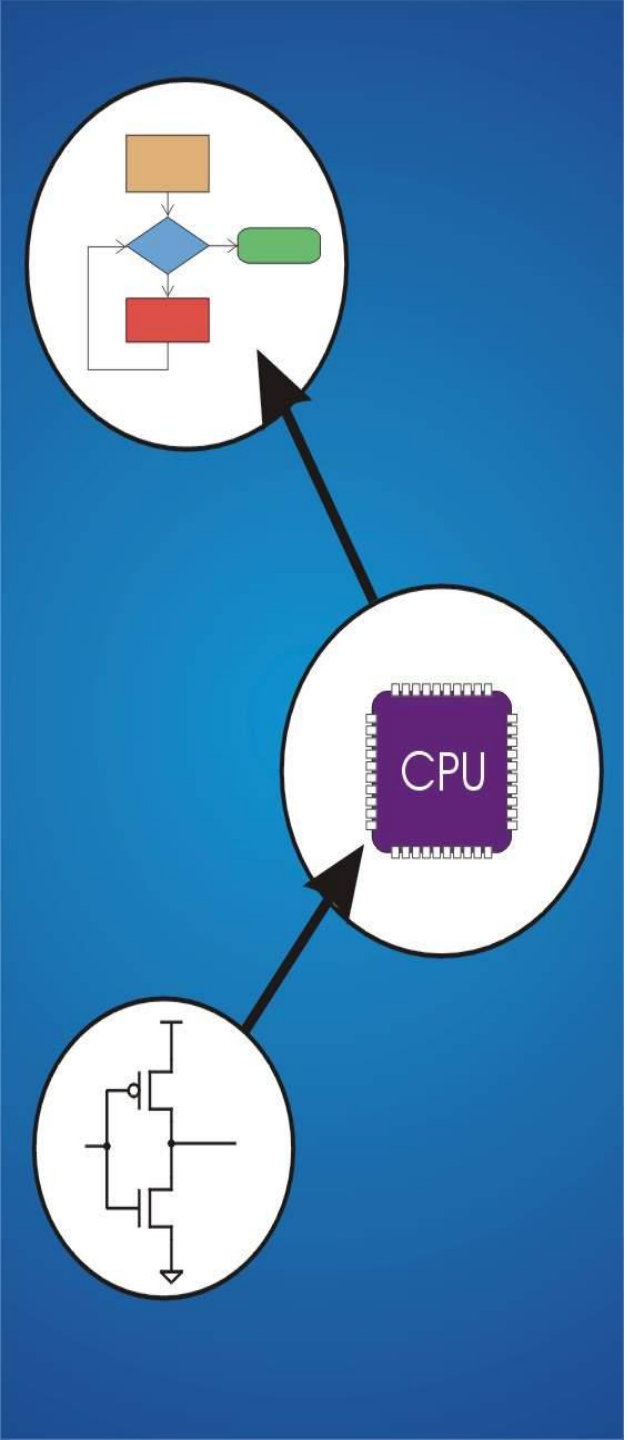
Speed = 10

PWM Pin
P3.0

ON = 10 = Speed

OFF = 90 = 100 - Speed





UNIT – IV

Interfacing Methods

Interfacing

LCD

LCD : Pin Diagram

VSS = 5V

VDD = GND

VEE = Contrast Voltage

RS = 0 Command Register

RS = 1 Data Register

RW = 0 Write

RW = 1 Read

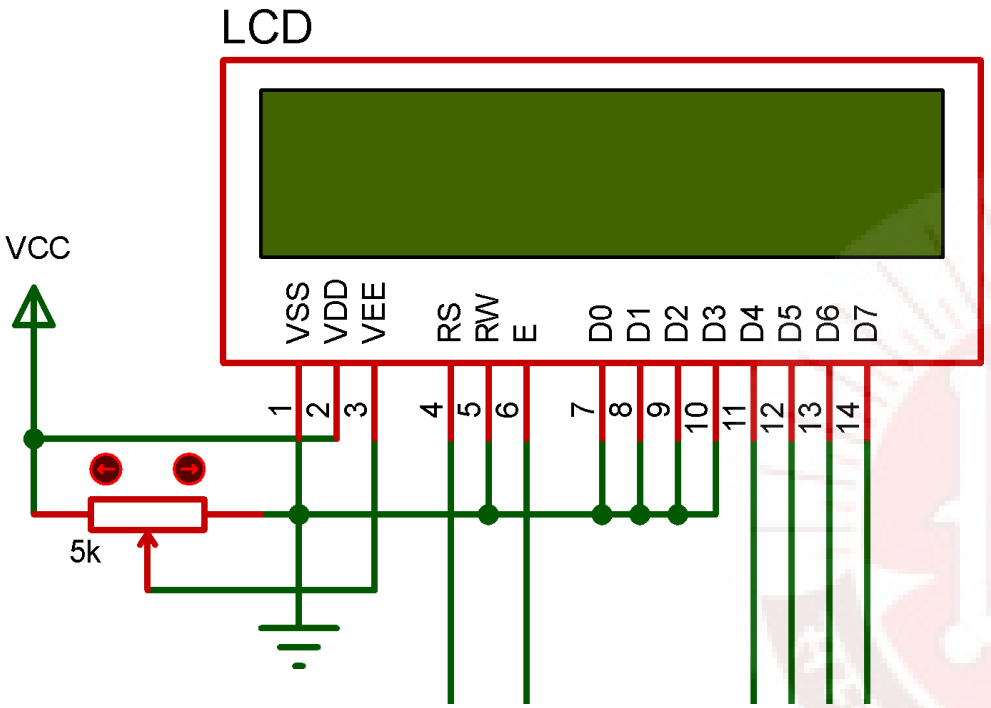
E = 1 -> 0 Enable

Data

8 bit mode = D0-D7

4 bit mode = D4-D7

LCD : Our Connection



VSS = 5V

VDD = GND

VEE = Contrast Votlage

RS = 0 Command Register

RS = 1 Data Register

RW = 0 Write

RW = 1 Read

E = 1 -> 0 Enable

Data

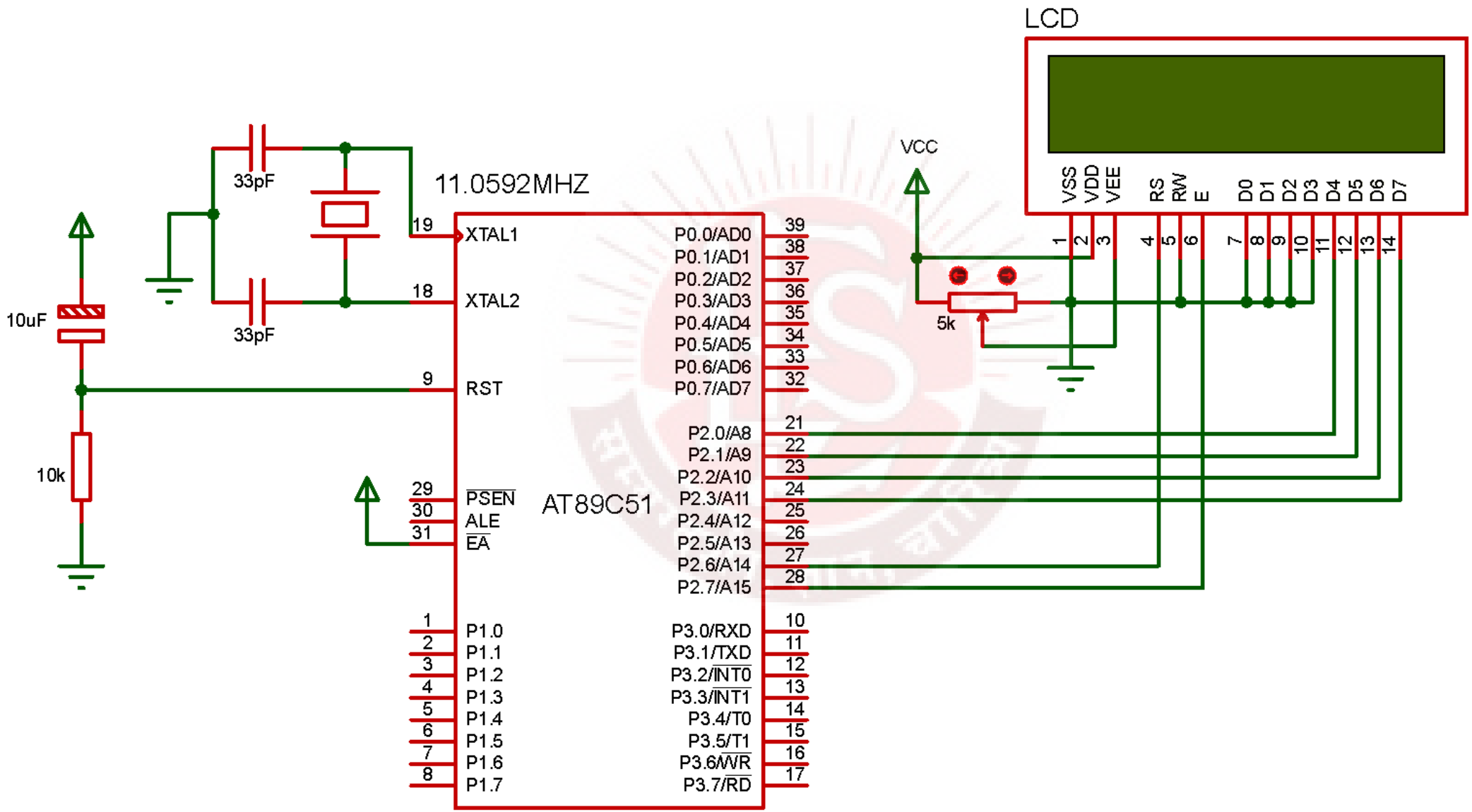
8 bit mode = D0-D7

4 bit mode = D4-D7

LCD : Commands

Code (Hex)	Command to LCD
01	Clear display screen
02	Return home
06	Shift Cursor to Right
0C	Display on cursor on
0F	Display on cursor blinking
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
32	2 lines and 5x7 matrix font
28	4 bit mode

LCD : Interfacing



THANK YOU ?

Prof. D. A. Patil

Department of B.Sc. Computer Science (Entire)

Smt. Kasturbai Walchand College, Sangli