AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
PATEL GUDA, IBRAHIMPATNAM (M), R.R.Dist, 501510, T.S
Ph.No. 08415-201345, www.avniet.ac.in, Email: avn.principal@gmail.com

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL
Accredited with 'B++' Grade by NAAC

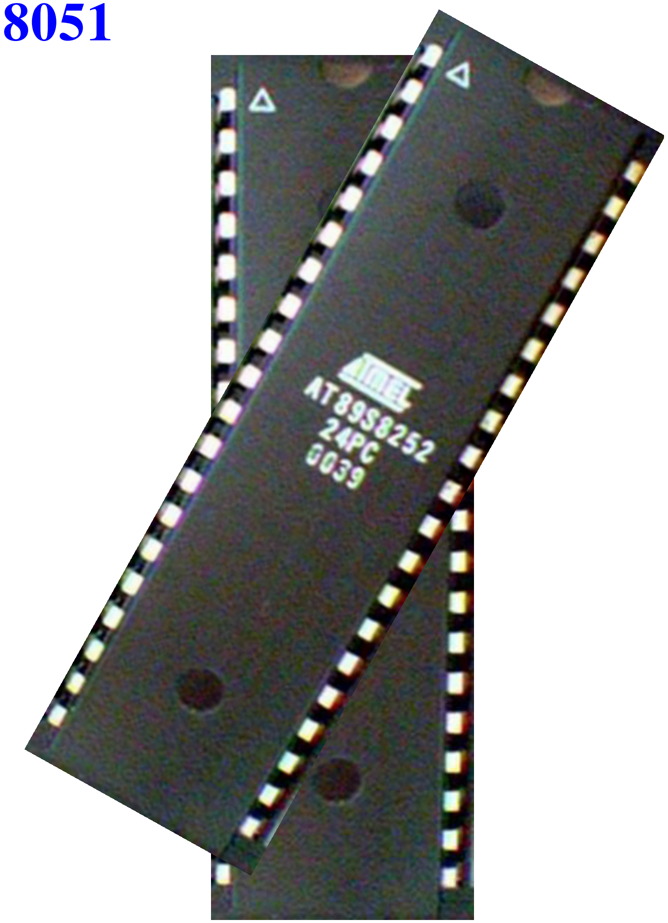
Presentation on "INTRODUCTION TO MICRONTROLLER 8051"

BY

Somashekhar Malipatil
Asst.Professor
Department of ECE

Contents:

- Introduction
- Block Diagram and Pin Description of the 8051
- Registers
- Memory mapping in 8051
- Stack in the 8051
- I/O Port Programming
- Timer
- Interrupt



Why do we need to learn Microprocessors/controllers?

- The microprocessor is the core of computer systems.
- Nowadays many communication, digital entertainment, portable devices, are controlled by them.
- A designer should know what types of components he needs, ways to reduce production costs and product reliable.

Different aspects of a microprocessor/controller

- Hardware :Interface to the real world

- Software :order how to deal with inputs

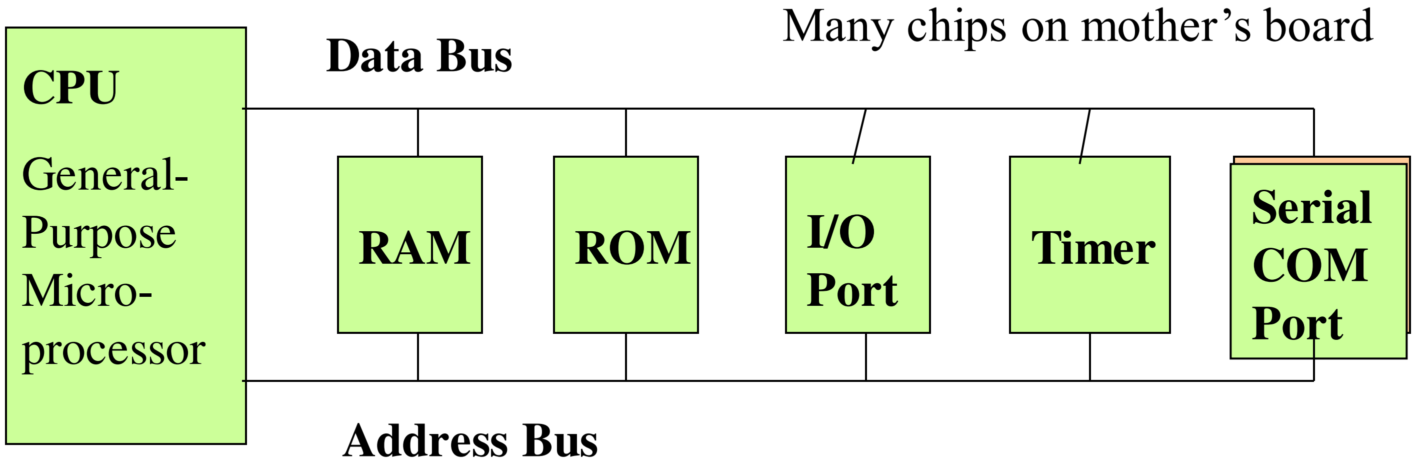
The necessary tools for a microprocessor/controller

- CPU: Central Processing Unit
- I/O: Input /Output
- Bus: Address bus & Data bus
- Memory: RAM & ROM
- Timer
- Interrupt
- Serial Port
- Parallel Port

Microprocessors:

General-purpose microprocessor

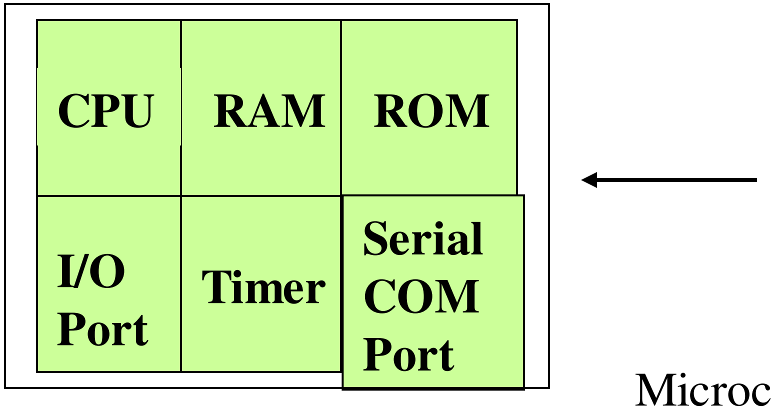
- CPU for Computers
- No RAM, ROM, I/O on CPU chip itself
- Example：Intel's x86, Motorola's 680x0



General-Purpose Microprocessor System

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

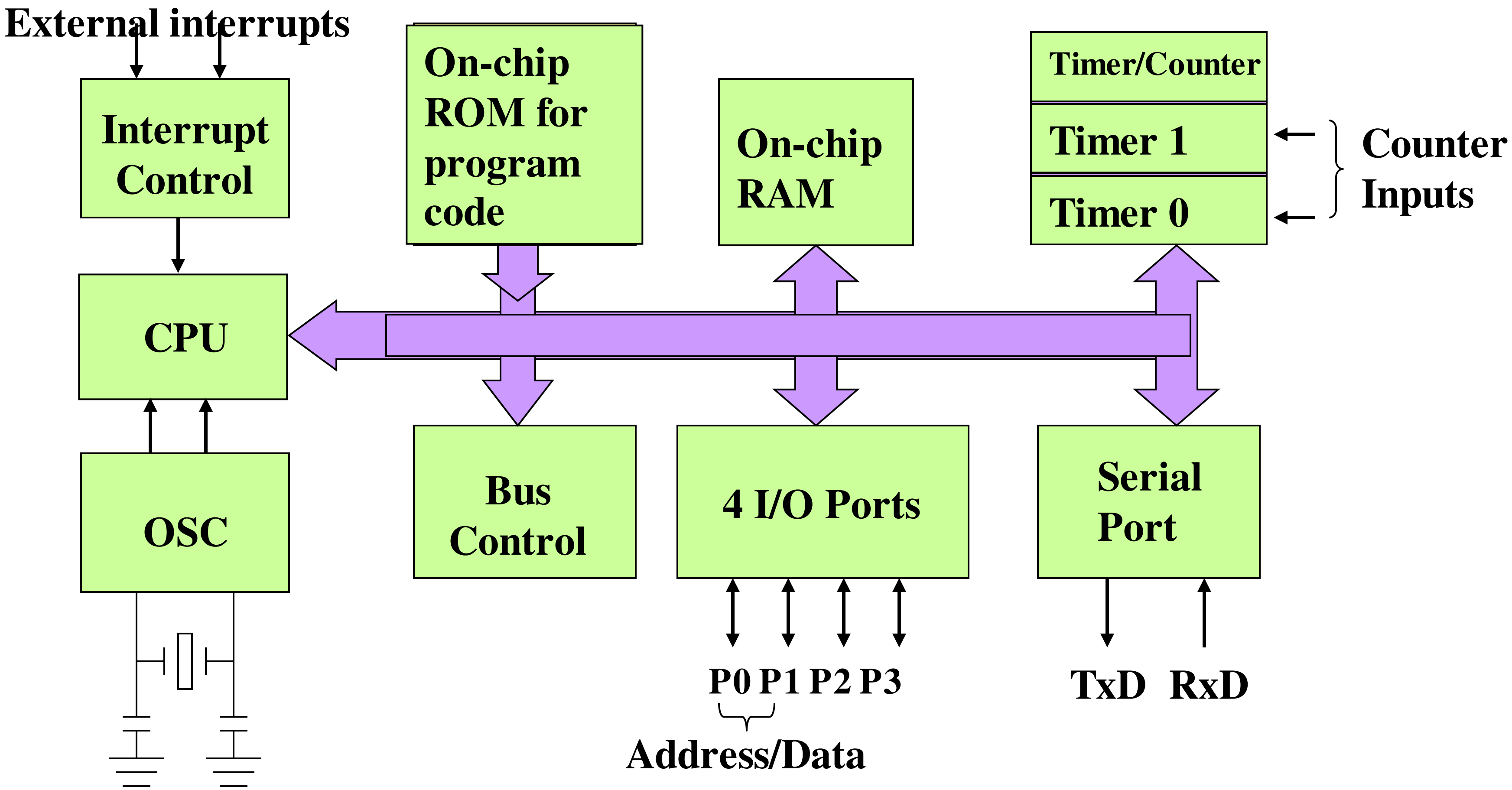
Microprocessor

- 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

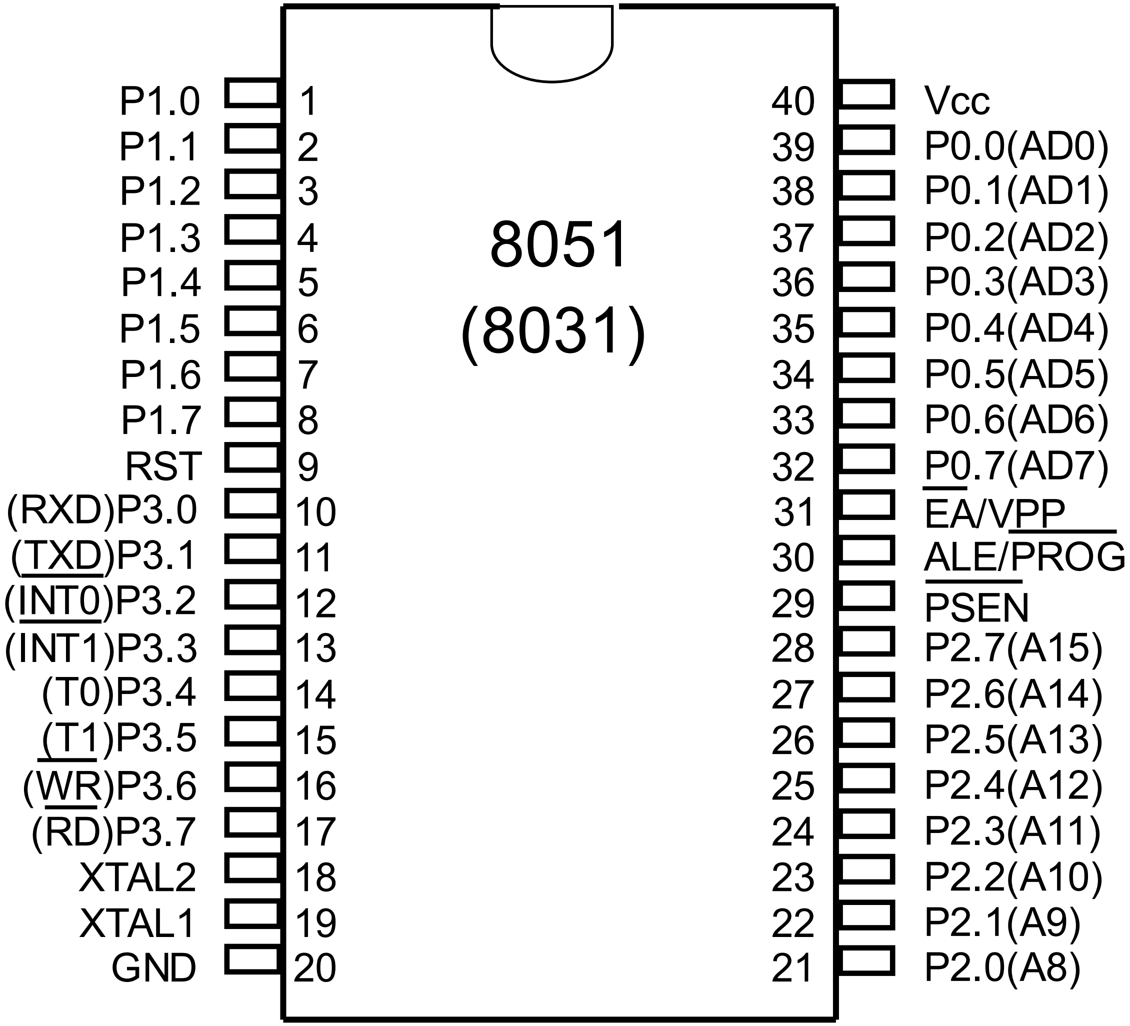
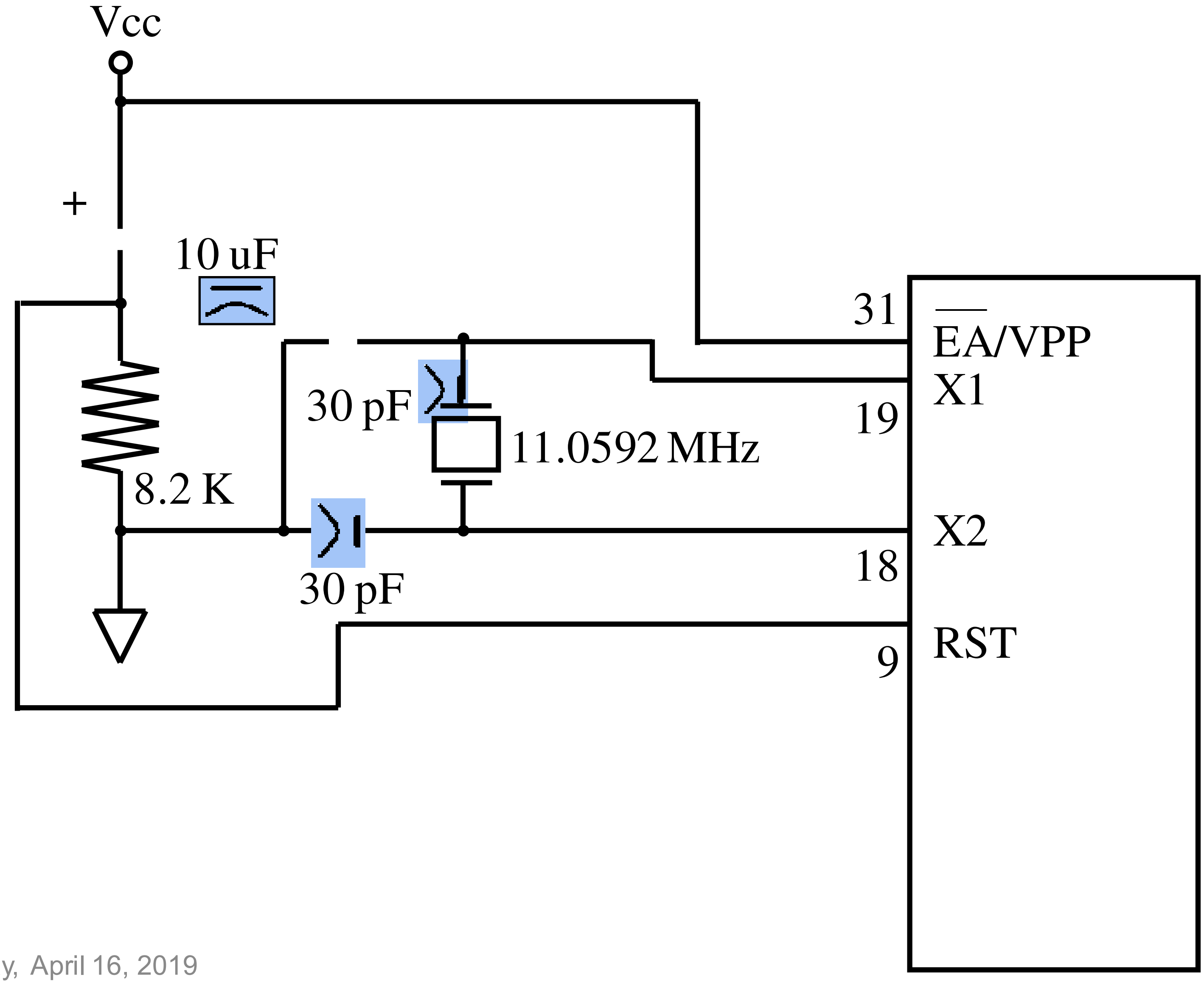
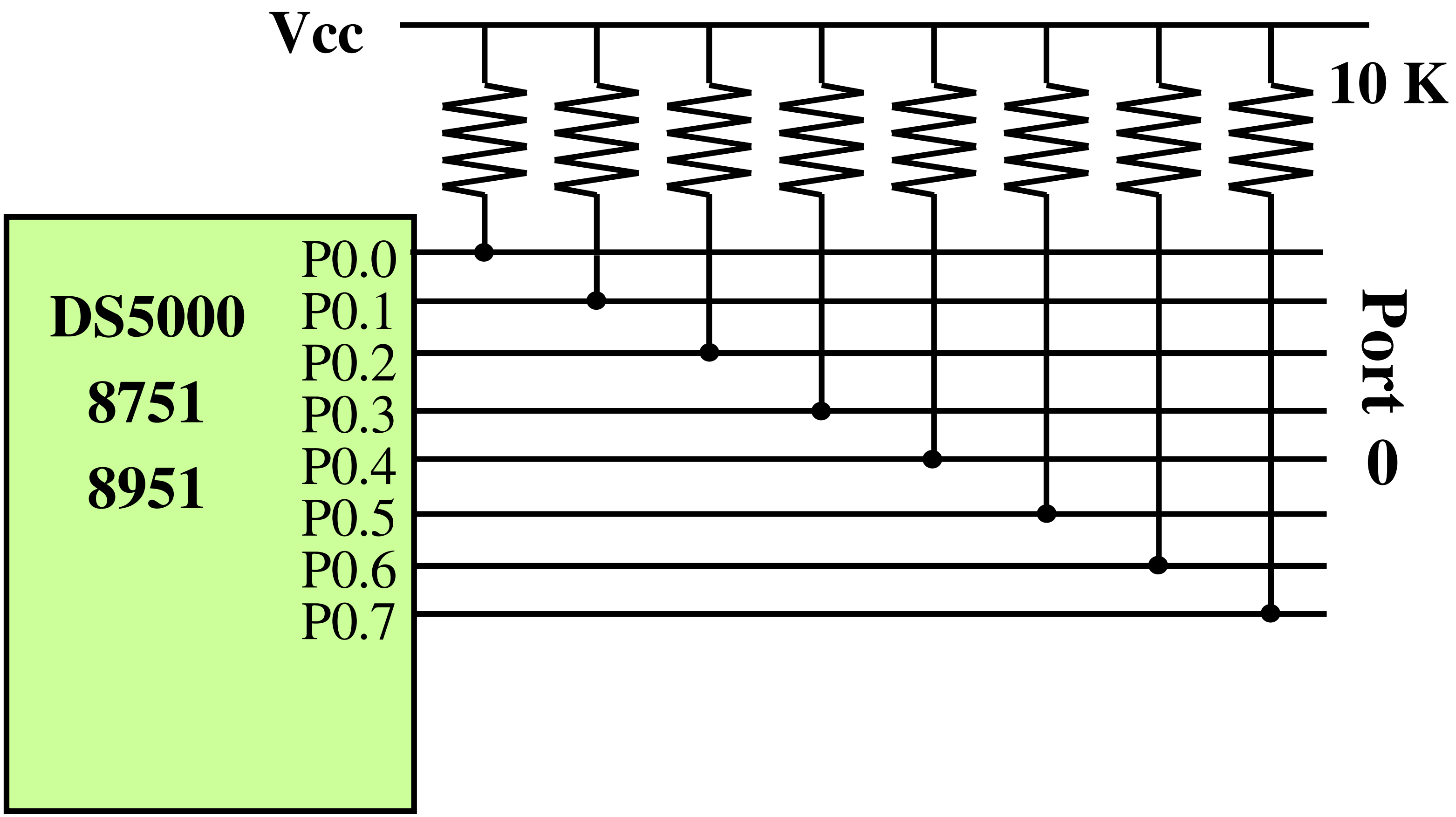


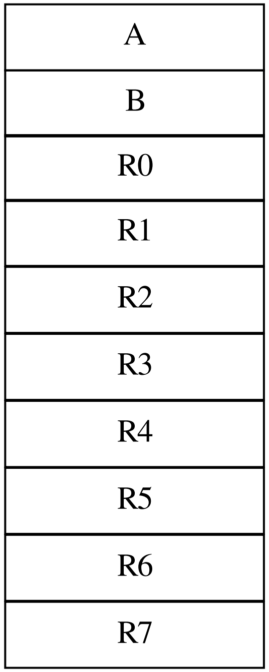
Figure (b). Power-On RESET Circuit



Port 0 with Pull-Up Resistors



Registers



A

B

R0

R1

R2

R3

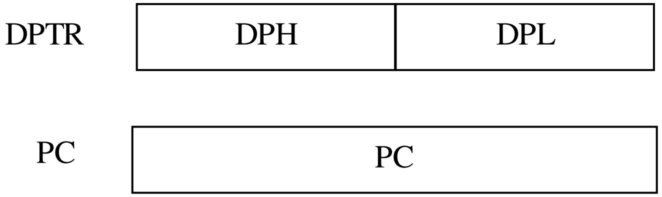
R4

R5

R6

R7

Some 8-bitt Registers of the 8051



DPTR

DPH

DPL

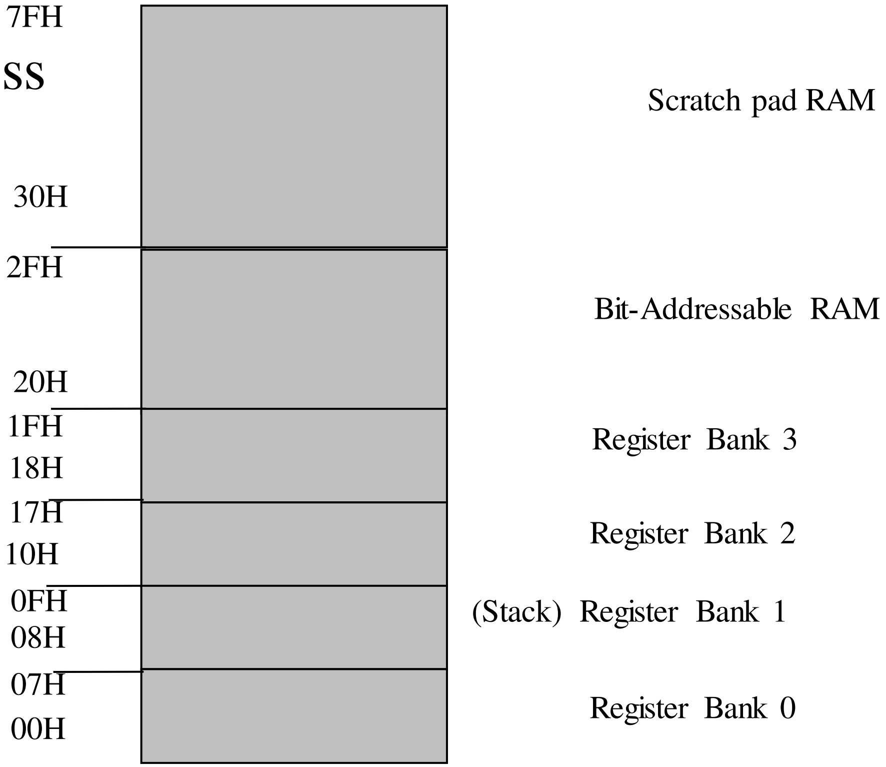
PC

PC

Some 8051 16-bit Register

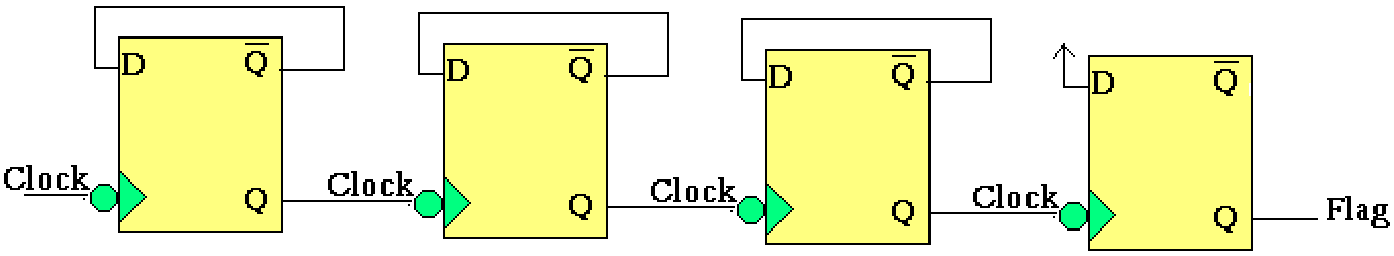
Stack in the 8051

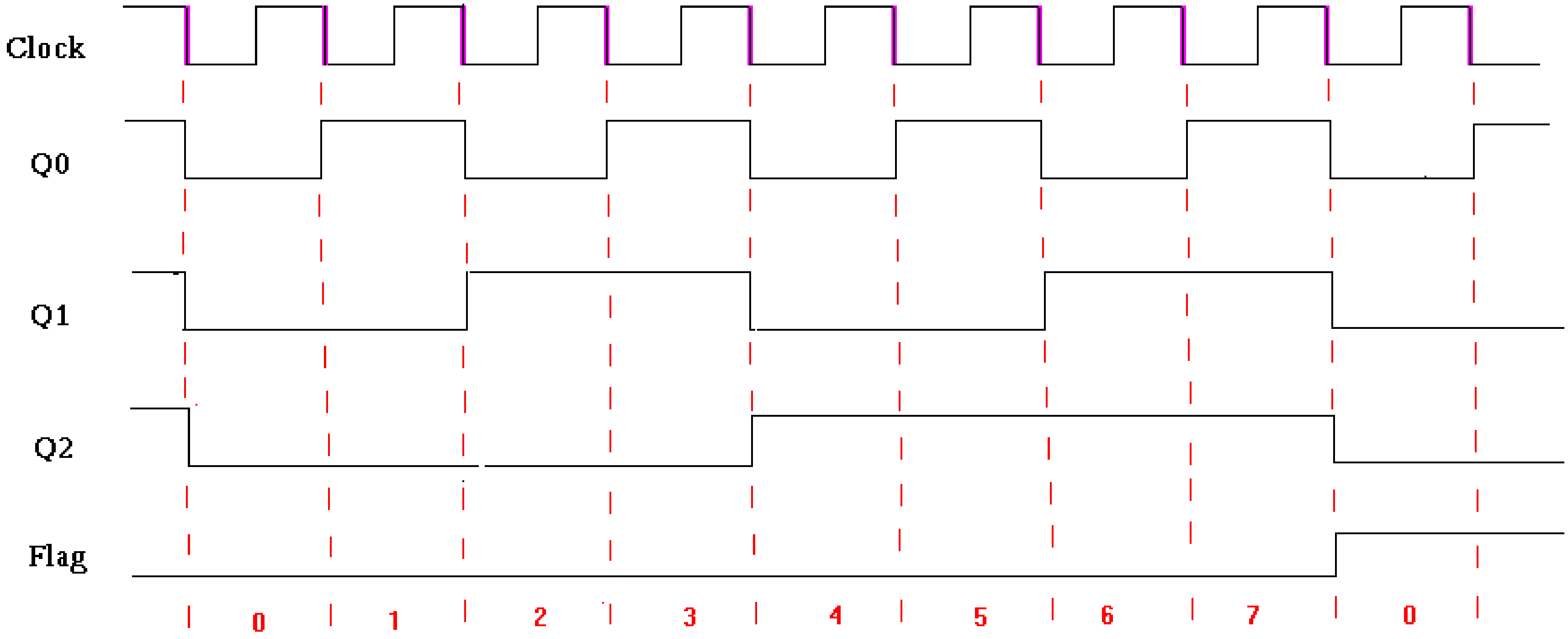
- The register used to access the stack is called **SP** (stack pointer) register.
- The stack pointer in the 8051 is only 8 bits wide, which means that it can take value 00 to FFH. When 8051 powered up, the SP register contains value 07.



Timer:

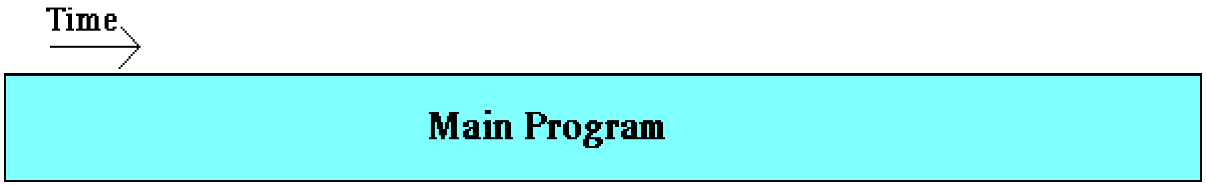
Timer :


D
Q̄
Clock
Q
D
Q̄
Clock
Q
D
Q̄
Clock
Q
D
Q̄
Clock
Q
Flag


Clock
Q0
Q1
Q2
Flag
0
1
2
3
4
5
6
7
0

Interrupt :

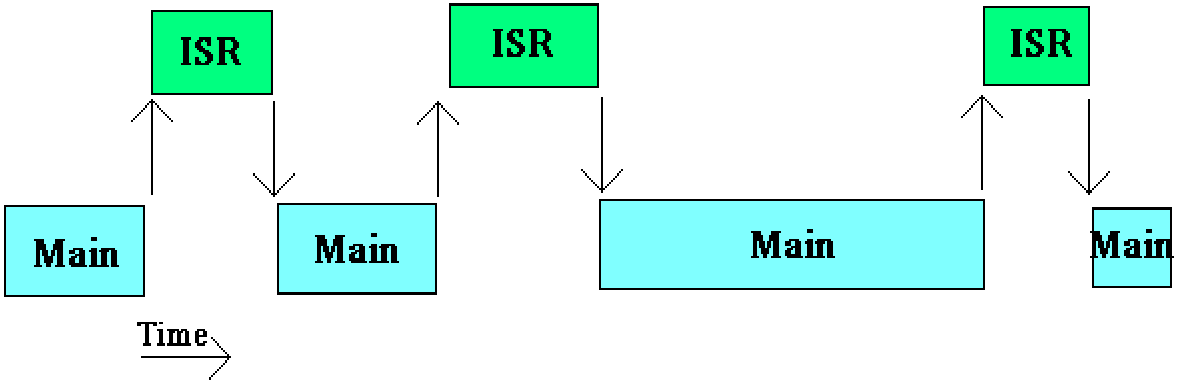
Program execution without intrrupts :



Time

Main Program

Program execution with intrrupts :



ISR

ISR

ISR

Main

Main

Main

Main

Time

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

$$(43)_{10} =$$

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

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

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

Register Addressing Mode

MOV Rn, A ;n=0,..,7

ADD A, Rn

MOV DPL, R6

~~MOV DPTR, A~~

~~MOV Rn, 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
```

Thank You