The 8051 Microcontrollers

Instructor

Prof. Zhizheng Wu

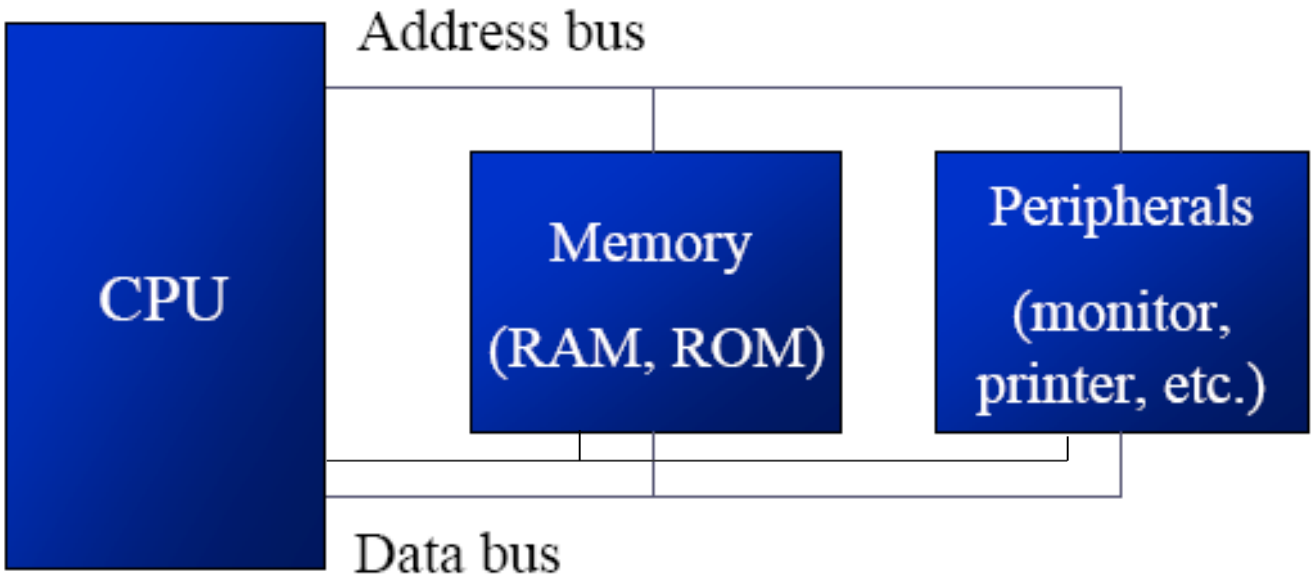
吴智政

School of Mechatronic Engineering and Automation

OUTLINE

- Inside the computer
- Intel microcontroller families
- The 8051 microcontroller

INSIDE THE COMPUTER



The main parts inside the computer

INSIDE THE COMPUTER

- CPU (Central Processing Unit)
  - ➢ Execute information stored in memory
- I/O (Input/output) devices
  - ➢ Provide the means of communicating with CPU
- Memory
  - ➢ RAM (Random Access Memory) – temporary storage of programs that computer is running
    - ✓ The data is lost when computer is off
  - ➢ ROM (Read Only Memory) – contains programs and information essential to operation of the computer
    - ✓ The information cannot be changed by user, and is not lost when power is off
    - * It is called nonvolatile memory

INSIDE THE COMPUTER

- RAM memory is called volatile memory since cutting off the power to the IC will result in the loss of data

- There are three types of RAM
  - Static RAM (SRAM)
  - NV-RAM (nonvolatile RAM)
  - Dynamic RAM (DRAM)

- Static RAM (SRAM)

  - Storage cells in static RAM memory are made of flip-flops and therefore do not require refreshing in order to keep their data

  - The use of 4-transistor cells plus the use of CMOS technology has given birth to a high capacity SRAM, but its capacity is far below DRAM

- NV - RAM

  - NV-RAM combines the best of RAM and ROM
  - It uses extremely power-efficient SRAM cells built out of CMOS
  - It uses an internal lithium battery as a backup energy source
  - It uses an intelligent control circuitry. The main job of this control circuitry is to monitor the power supply pin constantly to detect loss of the external power supply

- Dynamic RAM (DRAM)
  - Dynamic RAM uses a capacitor to store each bit
  - Advantage:
    - It cuts down the number of transistors needed to build the cell
    - High density (capacity), cheaper cost per bit, and lower power consumption per bit
  - Disadvantage:
    - It must be refreshed periodically, due to the fact that the capacitor cell loses its charge
    - While it is being refreshed, the data cannot be accessed

- ROM is a type of nonvolatile memory that does not lose its contents when the power is turned off

- There are different types of read-only memory
  - PROM
  - EPROM
  - EEPROM
  - Flash EPROM
  - Mask ROM

- PROM refers to the kind of ROM that the user can burn information into

  - PROM is a user-programmable memory
  - For every bit of the PROM, there exists a fuse
  - If the information burned into PROM is wrong, that PROM must be discarded since its internal fuses are blown permanently

- EPROM was invented to allow making changes in the contents of PROM after it is burned

- A widely used EPROM is called UV-EPROM
  - UV stands for ultra-violet
  - One can program the memory chip and erase it thousands of times
  - The only problem with UV-EPROM is that erasing its contents can take up to 20 minutes
  - The major disadvantage of UV-EPROM, is that it cannot be programmed while in the system board

- EEPROM stands for **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory

  - Erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage and therefore instant
  - One can select which byte to be erased, in contrast to UV-EPROM, in which the entire contents of ROM are erased
  - One can program and erase its contents while it is still in the system board

- Flash EPROM has become a popular user-programmable memory chip since the early 1990s

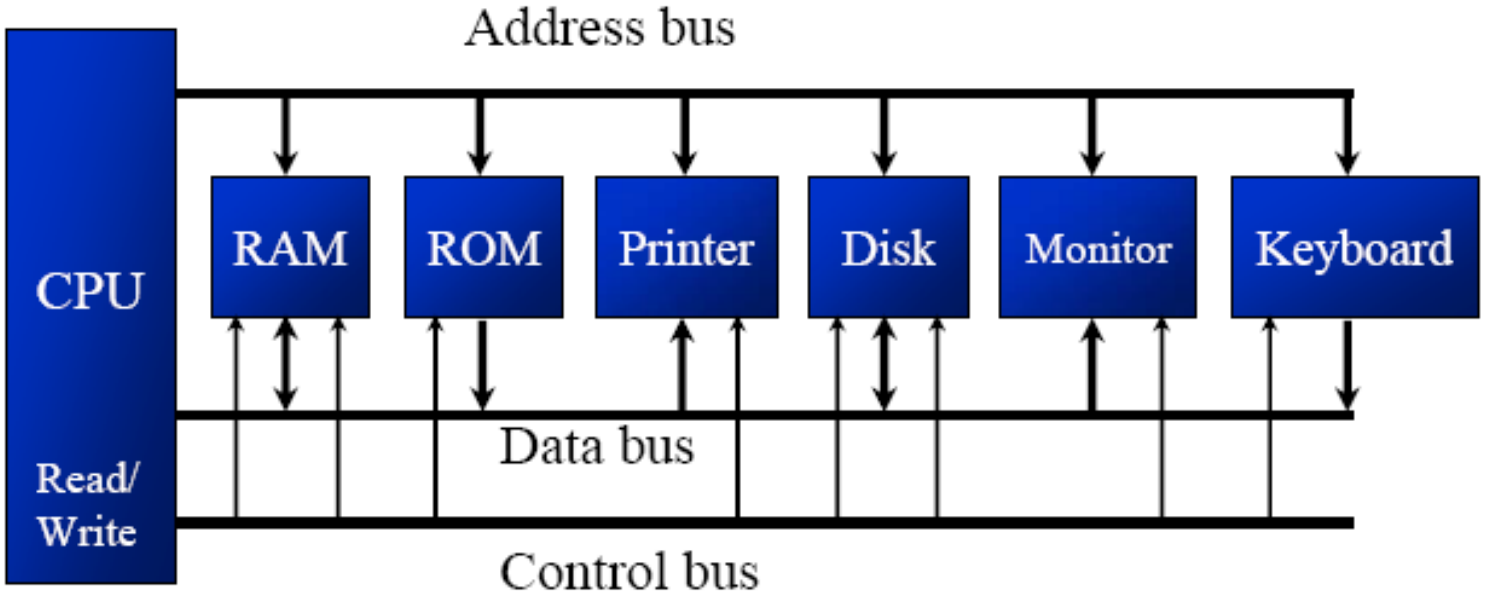
  - The erasure method is electrical
  - The process of erasure of the entire contents takes less than a second, or might say in a flash
  - The flash memory can be programmed while it is in its socket on the system board

- Mask ROM refers to a kind of ROM in which the contents are programmed by the IC manufacturer, not user programmable

  - Mask ROM is used when the needed volume is high and it is absolutely certain that the contents will not change

  - The main advantage of mask ROM is its cost, since it is significantly cheaper than other kinds of ROM, but if an error in the data/code is found, the entire batch must be thrown away

INSIDE THE COMPUTER



- Carries information from place to place through a bus
  - Address bus
  - Data bus
  - Control bus

- Address bus
  - For a device (memory or I/O) to be recognized by the CPU, it must be assigned an address
  - The address assigned to a given device must be unique
  - The CPU puts the address on the address bus, and the decoding circuitry finds the device
  - The number of locations with which a CPU can communicate is always equal to 2^x, where x is the address lines
  - The address bus is unidirectional

- Data bus

  - The CPU either gets data from the device or sends data to it

  - Data buses are bidirectional

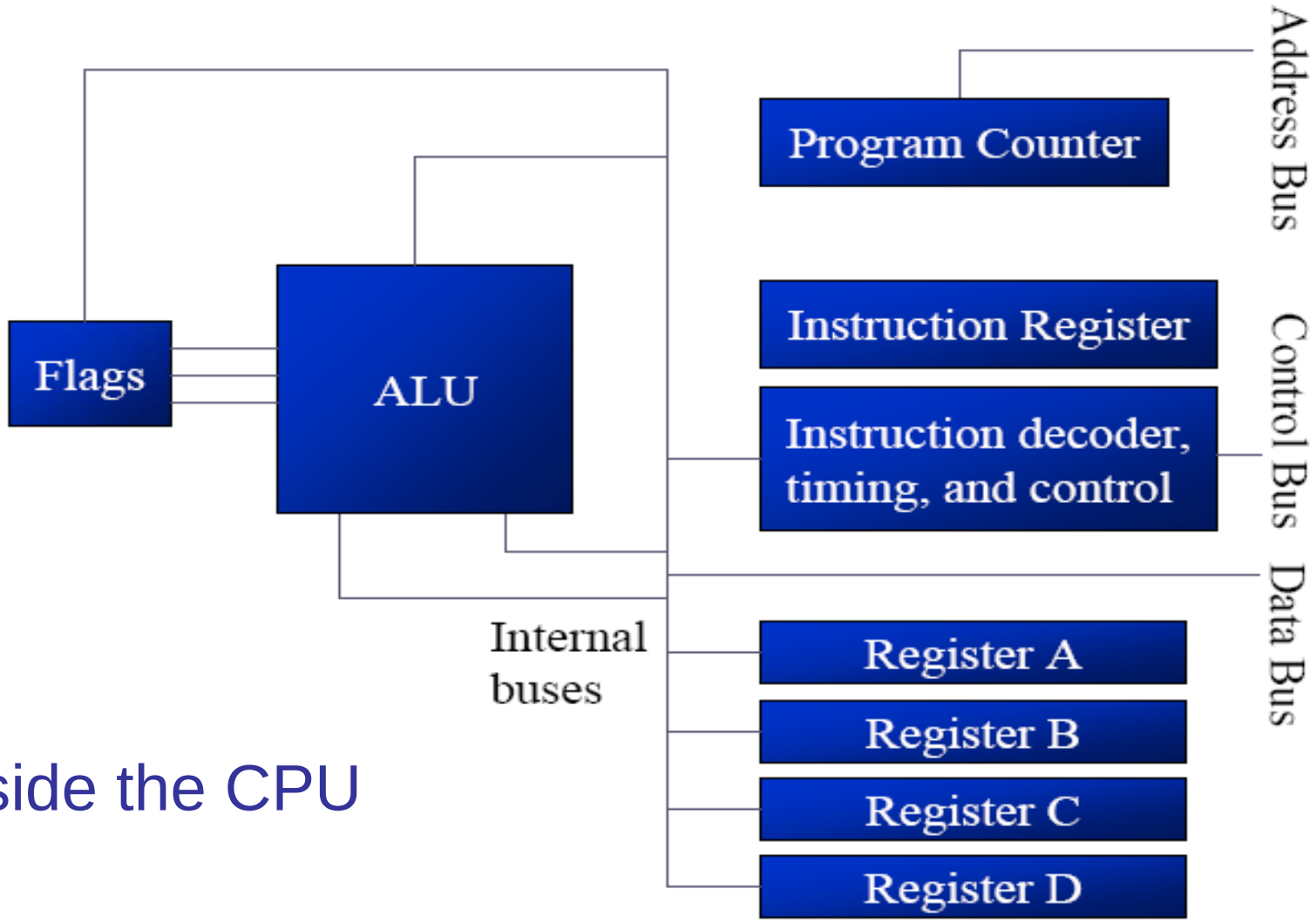
  - The average size of data buses in CPUs varies between 8 and 64

- Control bus

  ➢ Provides read or write signals to the device to indicate if the CPU is asking for information or sending it information

  ➢ Control buses are unidirectional

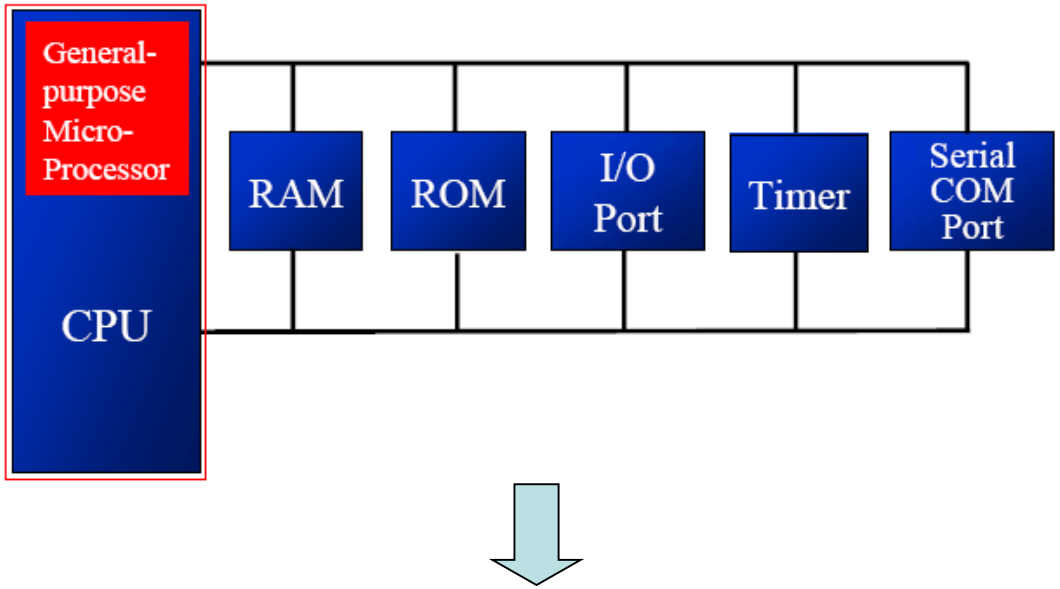
INSIDE THE COMPUTER



Inside the CPU

- Inside the CPU
  - ALU (arithmetic/logic unit)
    - ✧ Performs arithmetic functions such as add, subtract, multiply, and divide, and logic functions such as AND, OR, and NOT
  - Program counter
    - ✧ Points to the address of the next instruction to be executed
    - ✧ As each instruction is executed, the program counter is incremented to point to the address of the next instruction to be executed
  - Instruction decoder
    - ✧ Interprets the instruction fetched into the CPU
    - ✧ A CPU capable of understanding more instructions requires more transistors to design

INSIDE THE COMPUTER



INTEL MICROCONTROLLER FAMILIES

- Intel Microcontroller Families
  - MCS-48
  - MCS-41
  - MCS-51
  - MCS-96

INTEL MICROCONTROLLER FAMILIES

- Intel Microcontroller Families
  - MCS-48
  - MCS-41
  - MCS-51
  - MCS-96

INTEL MICROCONTROLLER FAMILIES

- Intel MCS-48
  - Introduced 1976
  - 8-bit, over 90 instructions with 90% of them being single byte
- Intel MCS-41
  - Introduced 1979
  - 8-bit, over 90 instructions with 70% of them being single byte

INTEL MICROCONTROLLER FAMILIES

- Intel MCS-51
  - Introduced 1980
  - 8-bit, has 111 instructions with 64 of them being single byte
- Intel MCS-96
  - Introduced 1982
  - 16-bit

THE 8051 MICROCONTROLLER

- Intel introduced 8051, referred as MCS-51, in 1981

  - The 8051 is an 8-bit processor
  - The CPU can work on only 8 bits of data at a time
  - The 8051 had 128 bytes of RAM
  - 4K bytes of on-chip ROM
  - Two timers
  - One serial port
  - Four I/O ports, each 8 bits wide
  - 6 interrupt sources

- The 8051 became widely popular after allowing other manufactures to make and market any flavor of the 8051, but remaining code-compatible
  - Intel (original)
  - Atmel
  - Philips/Signetics
  - AMD
  - Infineon (formerly Siemens)
  - Matra
  - Dallas Semiconductor/Maxim

THE 8051 MICROCONTROLLER

Feature	8051	8052	8031
ROM (on-chip program space in bytes)	4K	8K	0K
RAM (bytes)	128	256	128
Timers	2	3	2
I/O pins	32	32	32
Serial port	1	1	1
Interrupt sources	6	8	6

The comparison of 8051, 8052 and 8031

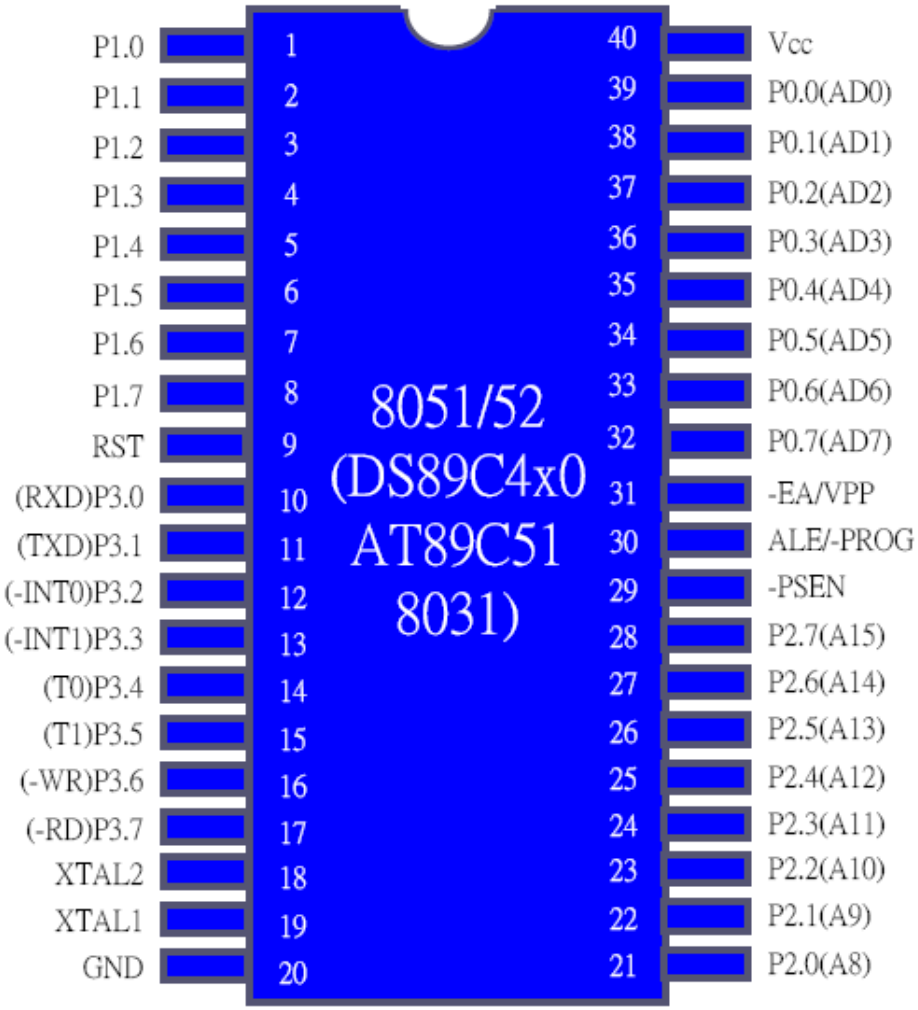
- 8051 family members

  - Have 40 pins dedicated for various functions such as I/O, -RD, -WR, address, data, and interrupts

  - Come in different packages, such as
    - DIP(dual in-line package)
    - QFP(quad flat package)
    - LLC(leadless chip carrier)

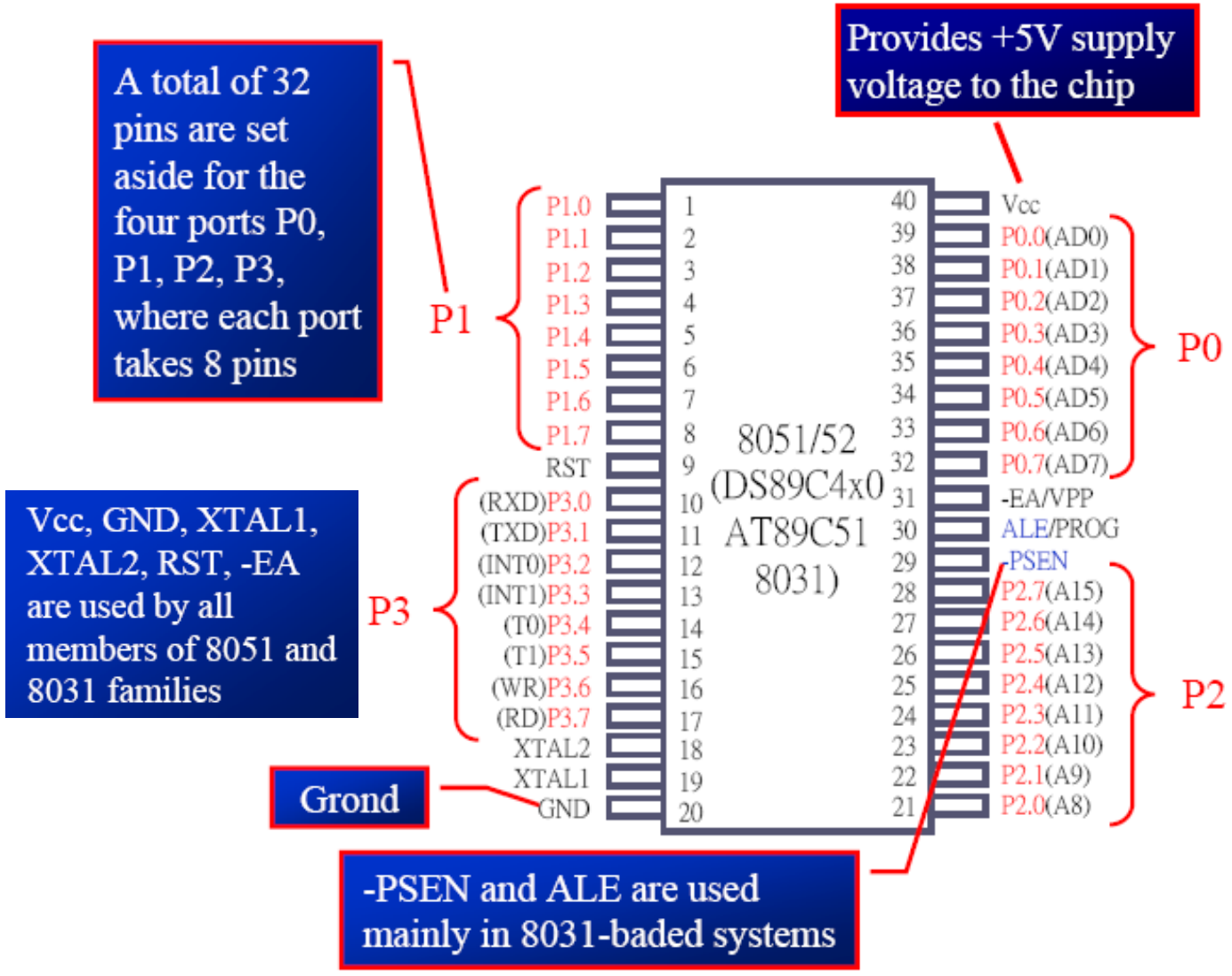
  - Some companies provide a 20-pin version of the 8051 with a reduced number of I/O ports for less demanding applications

THE 8051 MICROCONTROLLER

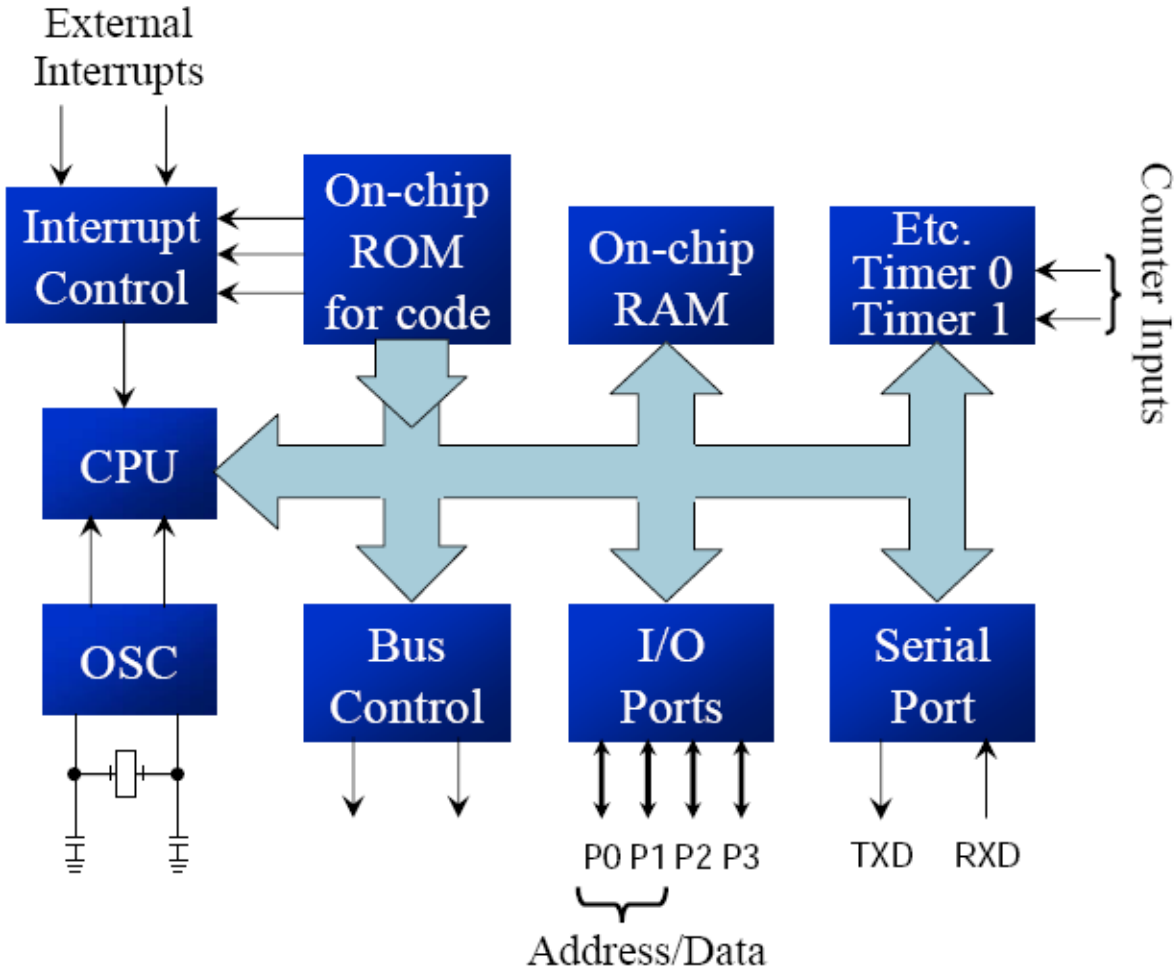


8051 pin diagram

THE 8051 MICROCONTROLLER



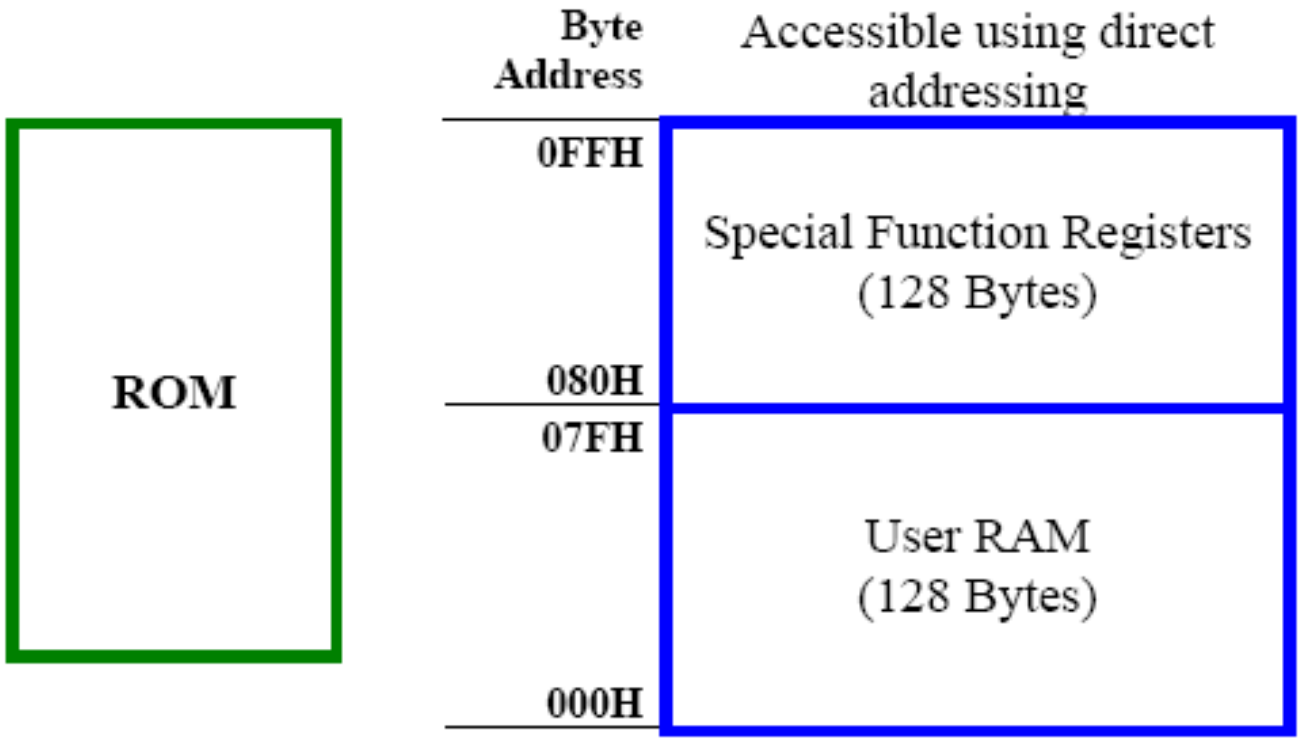
THE 8051 MICROCONTROLLER



The structure of the 8051 microcontroller

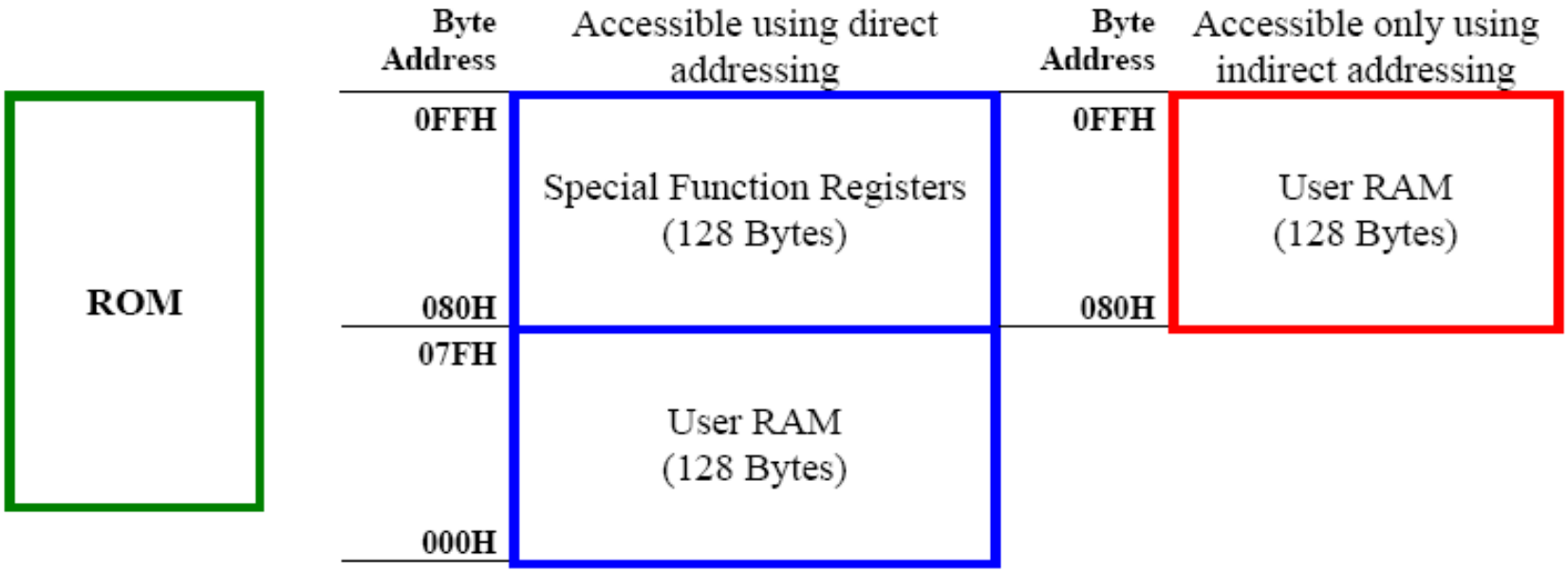
- Organization of the internal memory
  - 8051 implements a separate memory space for programs (code) and data.
  - Both code and data memory may be internal however both expand using external components to a maximum of 64K code memory and 64K data memory.
  - Internal memory consists of on-chip ROM and on-chip data RAM.

- Organization of the internal memory



8051 on-chip memory map

- Organization of the internal memory



8052 on-chip memory map

- Organization of the internal RAM memory

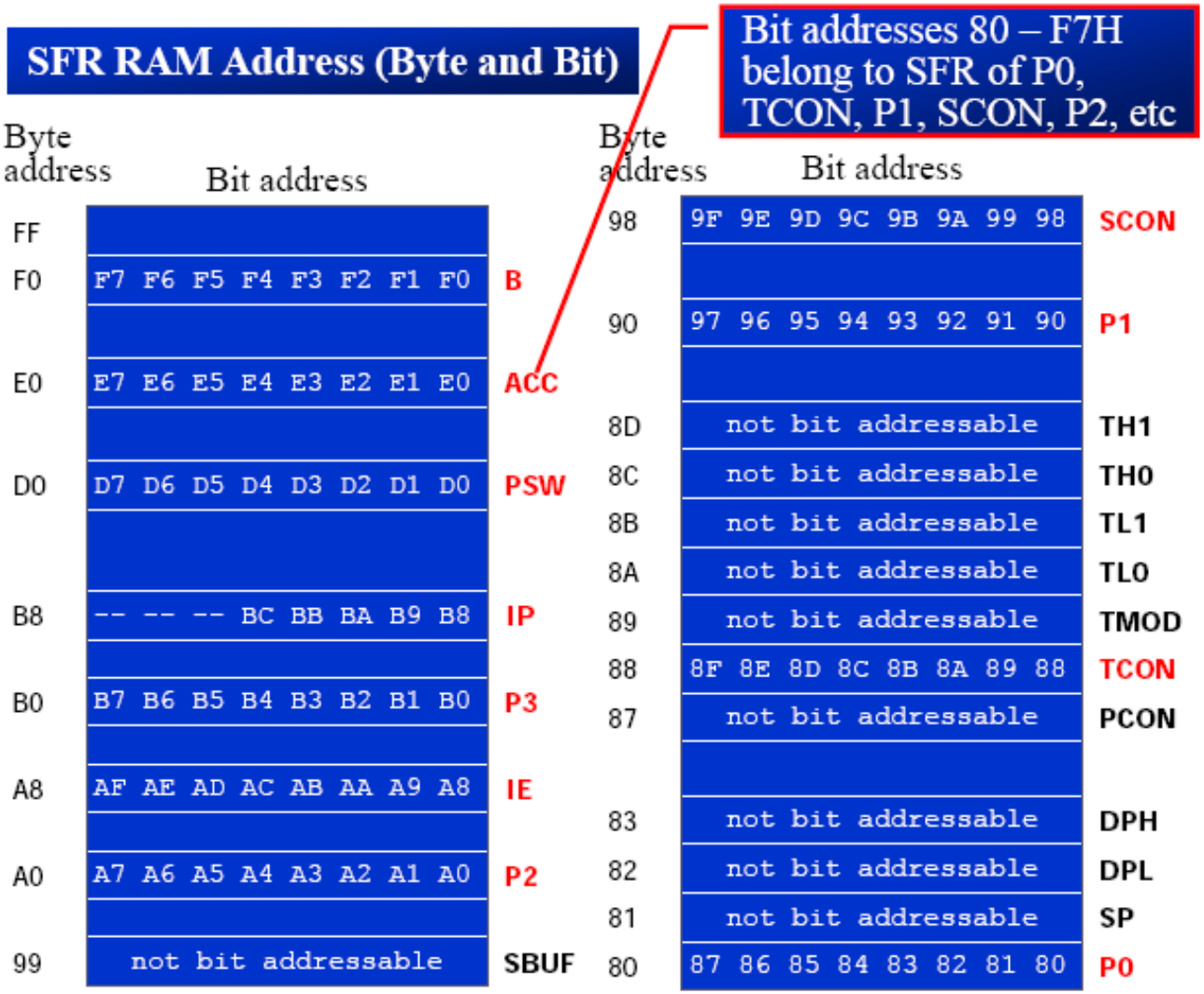
  - On-chip RAM contains register banks, bit addressable storage, general purpose storage, special function registers.

78								7F	User RAM and Stack Space (80 bytes, 30H-7FH)
70								77	
68								6F	
60								67	
58								5F	
50								57	
48								4F	
40								47	
38								3F	
30								37	
28								2F	Bit addressable area
20								27	
18	R0	R1	R2	R3	R4	R5	R6	R7	1F	Reg. Bank 3
10	R0	R1	R2	R3	R4	R5	R6	R7	17	Reg. Bank 2
08	R0	R1	R2	R3	R4	R5	R6	R7	0F	Reg. Bank 1
00	R0	R1	R2	R3	R4	R5	R6	R7	07	Reg. Bank 0

- Organization of the internal RAM memory

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

I/O SFRs
Control SFRs
Other SFRs

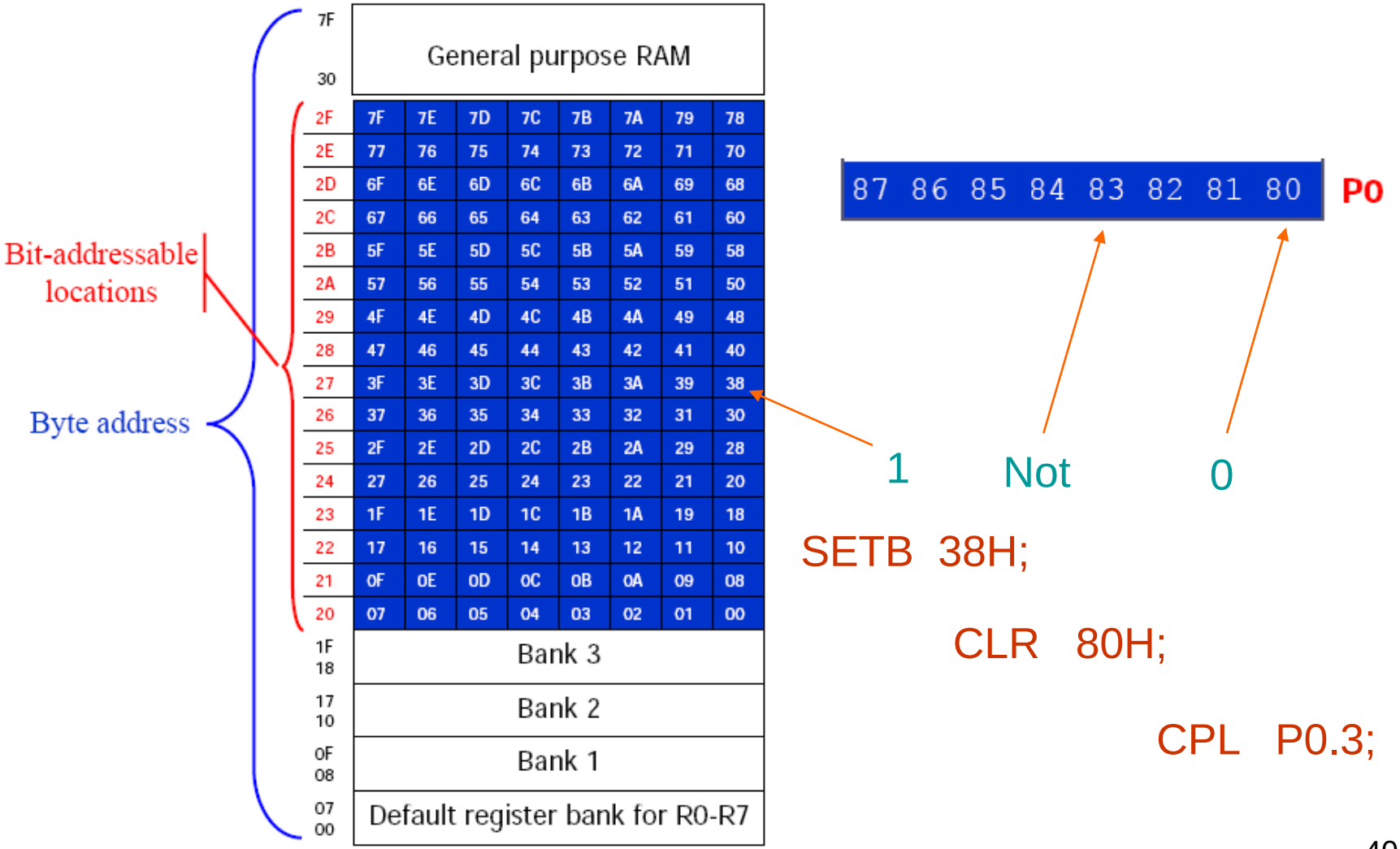


SFR (Special Function Register)

- Containing 210 bit-addressable location
  - 128 are at byte addresses 20H – 2FH
  - The rest is in the special function register B, ACC, PSW, T2CON, IP, IE, P0, P1, P2, P3, SCON, TCON
- Example:

```
SETB  38H;      Set the bit
CLR   80H;      Clear the bit
CPL   P0.3;     Complement the bit
```

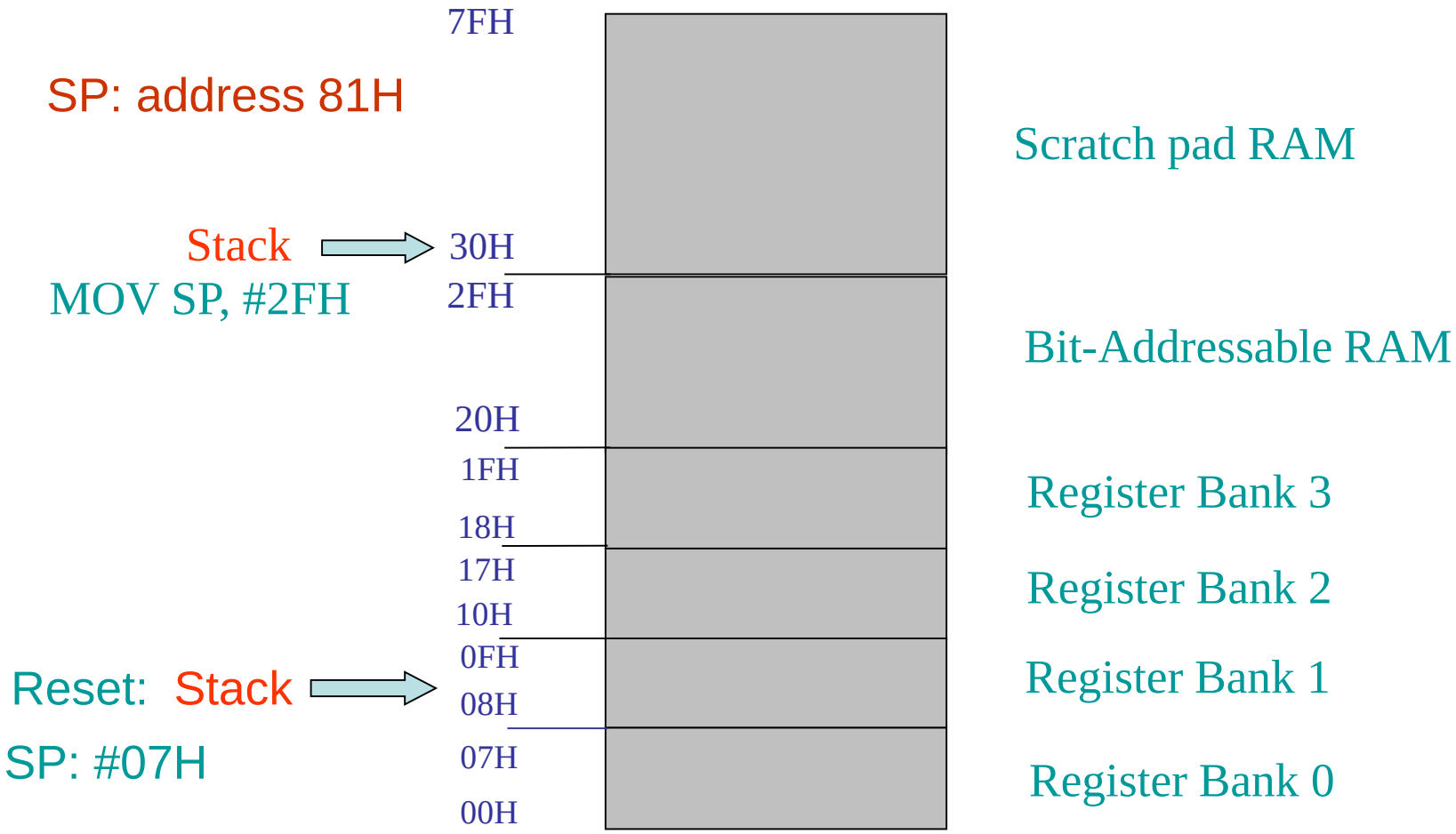
THE 8051 MICROCONTROLLER



- Register are used to store information temporarily, while the information could be a byte of data to be processed, or an address pointing to the data to be fetched

- The most widely used registers
  - A (Accumulator) - For all arithmetic and logic instructions
  - B, R0, R1, R2, R3, R4, R5, R6, R7, SP
  - PSW (program status word)
  - DPTR (data pointer) – 16bits

- RAM memory stack space allocation in the 8051



- PSW (program status word) register
  - PSW register is 8 bits wide, only 6 bits in the register are used

CY	AC	F0	RS1	RS0	OV	--	P

Bit	Address	Description
CY	PSW.7	Carry flag. Set whenever there is a carry out of D7 bit. Can be modified using **SETB C** and **CLR C**.
AC	PSW.6	Auxiliary carry flag. Set whenever there is a carry from bit D3 to bit D4. Otherwise, it is cleared.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1
RS0	PSW.3	Register Bank selector bit 0
OV	PSW.2	Overflow flag.
--	PSW.1	User-defined bit.
P	PSW.0	Parity flag. Set when the number of 1s in the accumulator is odd. Otherwise, it is 0.

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

- MOV destination, source; copy source to destination

"#" signifies that it is a value

```
MOV   A,#55H     ;load value 55H into reg. A
MOV   R0,A       ;copy contents of A into R0
                 ;(now A=R0=55H)
MOV   R1,A       ;copy contents of A into R1
                 ;(now A=R0=R1=55H)
MOV   R2,A       ;copy contents of A into R2
                 ;(now A=R0=R1=R2=55H)
MOV   R3,#95H    ;load value 95H into R3
                 ;(now R3=95H)
MOV   A,R3       ;copy contents of R3 into A
                 ;now A=R3=95H
```

MOV @R0, A ?

- EXAMPLE:

```
SETB    RS0;
CLR     RS1;
MOV     R7,   #66H;
```

Question: which byte address is changed?

```
MOV     A,  #55H
MOV     DPTR,  #1000H
MOVX    @DPTR,  A
```

Question: why is MOVX used?

- 40 pins of 8051

➢PSEN (Program Store Enable): is a control signal that enables external code memory.

- It is usually connected to the output enable (/OE) pin of the external code memory (e.g., EPROM)

- When executing a program from internal ROM, /PSEN remains in the inactive (high) state

- EA (external access) is an input pin and must be connected to Vcc or GND

  - The 8051 family members all come with on-chip ROM to store programs

    - EA pin is connected to Vcc

  - The 8031 and 8032 family members do no have on-chip ROM, so code is stored on an external ROM and is fetched by 8031/32

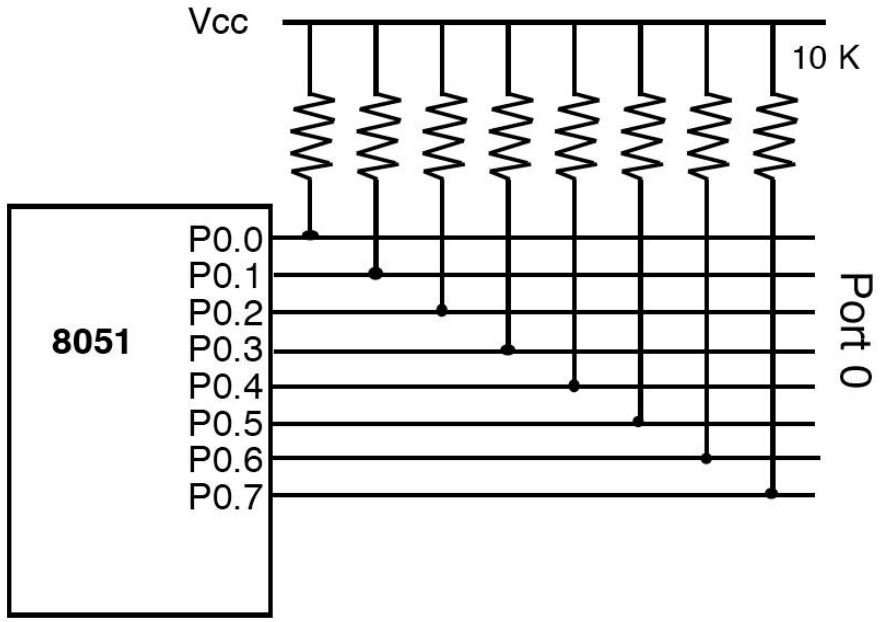
    - EA pin must be connected to GND to indicate that the code is stored externally

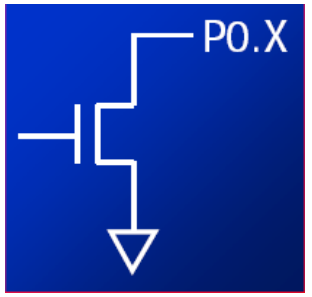
- ALE (Address Latch Enable): 8051 uses ALE for demultiplexing the address and data bus
  - ➢ ALE pulses at 1/6th the on chip oscillator
  - ➢ Port 0 provides both address and data
    - ☞ The 8031 multiplexes address and data through port 0 to save pins
    - ☞ ALE pin is used for demultiplexing the address and data by connecting to the G pin of the 74LS373 chip

- 32 pins P0, P1, P2 and P3 function as I/O port lines
  - **Port 0:** a dual purpose port on pins 32-39
  - **Port 1:** a dedicated I/O port on pins 1-8
  - **Port 2:** dual purpose port on pins 21-28 (could be a general purpose I/O or high byte of the address bus for external memory)
  - **Ports 3:** dual purpose port on pins 10-17
  - All the ports upon RESET are configured as **input**, ready to be used as output ports

- Port 0 is designated as AD0-AD7, allowing it to be used for both address and data
  - ➢ When connecting an 8051/31 to an external memory, port 0 provides both address and data
  - ➢ The 8051 multiplexes address and data through port 0 to save pins
  - ➢ ALE indicates if P0 has address or data
    - ☞ When ALE=0, it provides data D0-D7
    - ☞ When ALE=1, it has address A0-A7

- Port 0 can be used for input or output, each pin must be connected externally to a 10K ohm pull-up resistor
  - ➢ This is due to the fact that P0 is an open drain, unlike P1, P2, and P3
  - ➢ Open drain is a term used for MOS chips in the same way that open collector is used for TTL chip





- In 8051-based systems with no external memory connection
  - ➢ Both P1 and P2 are used as simple I/O without pull-up resistor
- In 8031/51-based systems with external memory connections
  - ➢ Port 2 must be used along with P0 to provide the 16-bit address for the external memory
    - ☞ P0 provides the lower 8 bits via A0 – A7
    - ☞ P2 is used for the upper 8 bits of the 16-bit address, designated as A8 – A15, and it cannot be used for I/O

- Port 3 can be used as input or output
  - Port 3 does not need any pull-up resistors
- Port 3 has the additional function of providing some extremely important signals
  - Alternate pin functions for Port 3

P3 Bit	Function	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2	$\overline{\text{INT0}}$	12
P3.3	$\overline{\text{INT1}}$	13
P3.4	T0	14
P3.5	T1	15
P3.6	$\overline{\text{WR}}$	16
P3.7	$\overline{\text{RD}}$	17

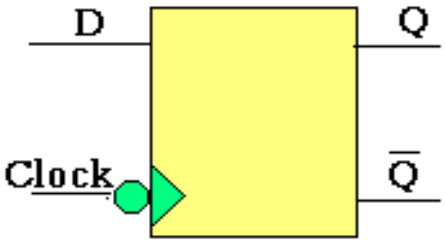
Serial communications

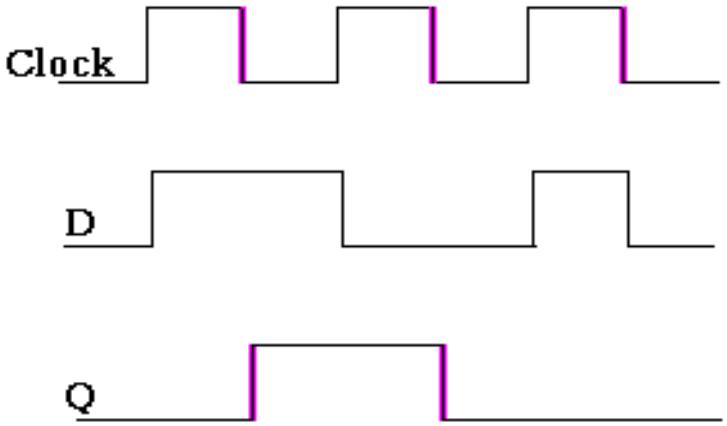
External interrupts

Timers

Read/Write signals of external memories

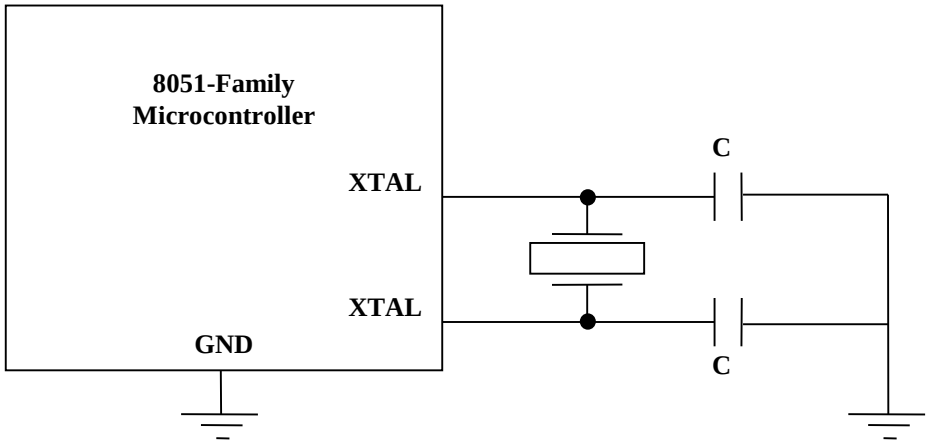
- D Latch: flip-flop





- The 8051 has an on-chip oscillator but requires an external clock to run it

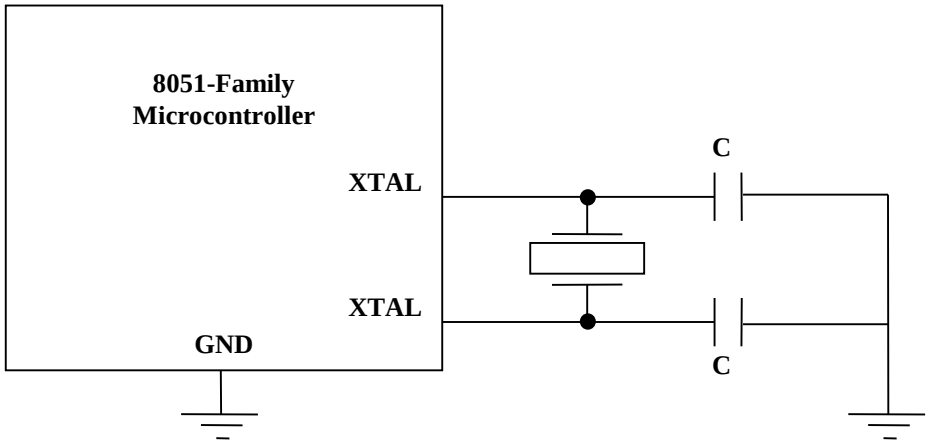
  - A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18)
  - The quartz crystal oscillator also needs two capacitors of 30 pF value



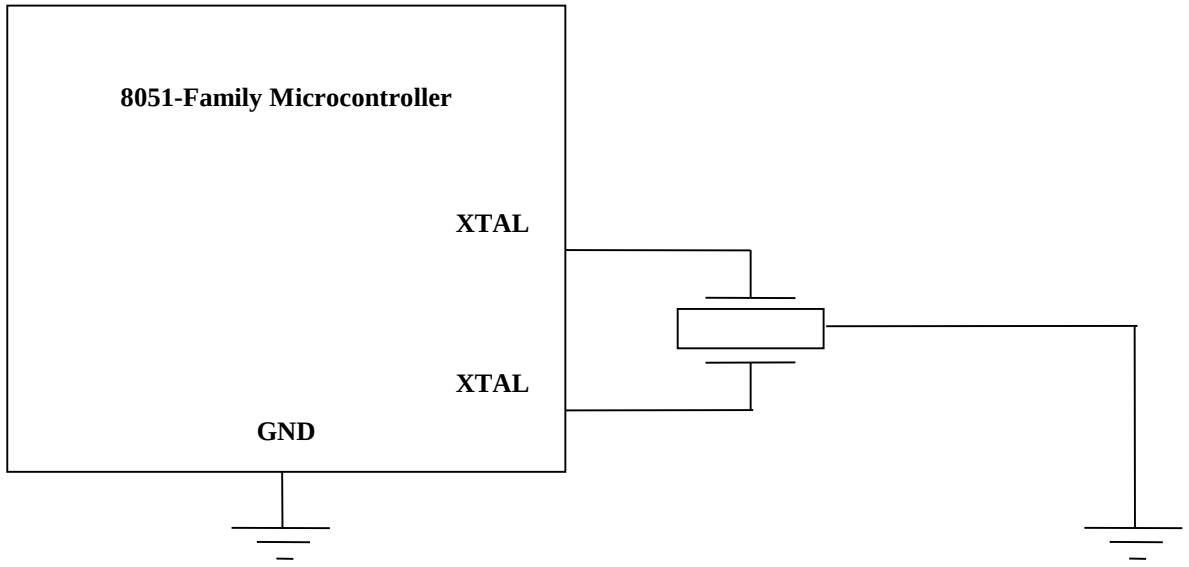
- The 8051 has an on-chip oscillator but requires an external clock to run it

  - A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18)

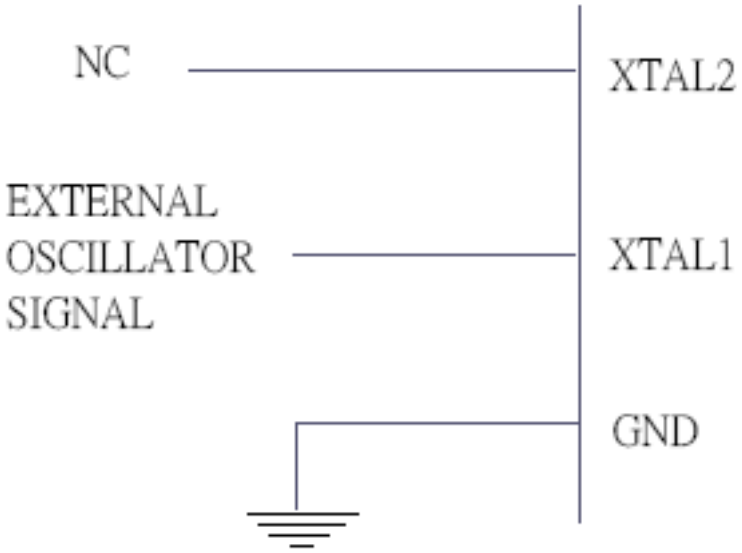
  - The quartz crystal oscillator also needs two capacitors of 30 pF value



- Ceramic oscillators cost half the price of quartz crystal oscillators, but are less stable

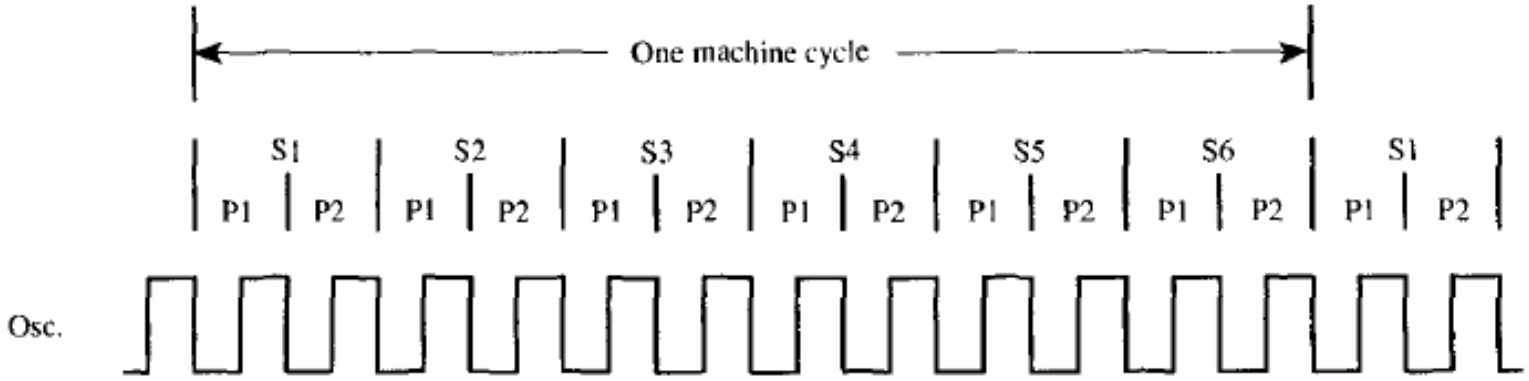


- If you use a frequency source other than a crystal or ceramic oscillator, such as a TTL oscillator
  - It will be connected to XTAL1
  - XTAL2 is left unconnected



- The speed of 8051 refers to the maximum oscillator frequency connected to XTAL

  - e.g. A 12-MHz chip must be connected to a crystal with 12 MHz frequency or less



- RESET pin is an input and is active high (normally low)
  - ➢ Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities
  - ➢ This is often referred to as a power-on reset
  - ➢ Activating a power-on reset will cause all values in the registers to be lost

RESET value of some 8051 registers

we must place the first line of source code in ROM location 0

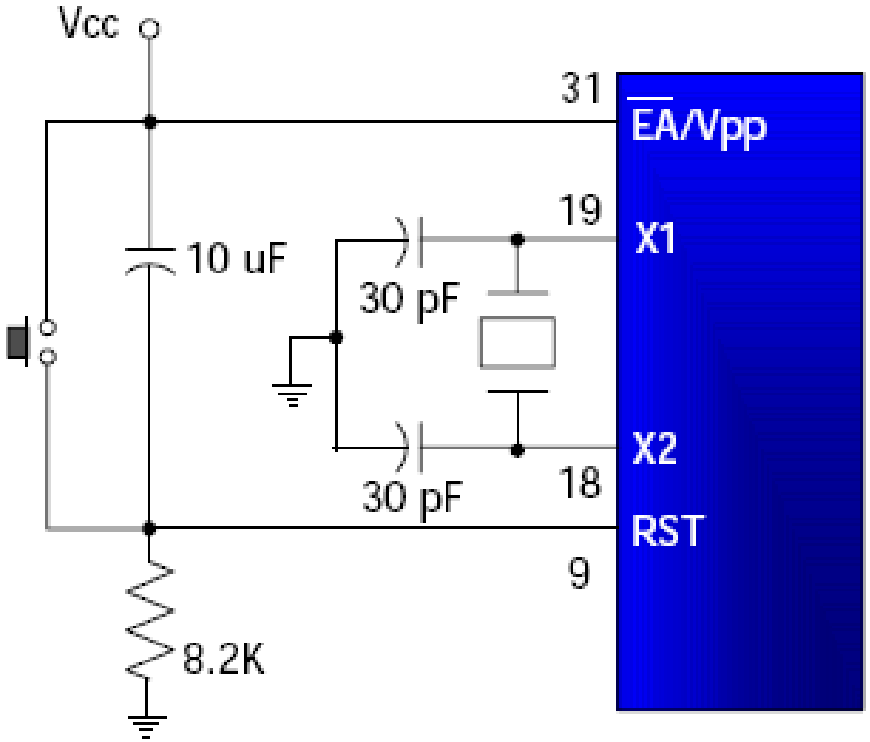
Register	Reset Value
PC	0000
DPTR	0000
ACC	00
PSW	00
SP	07
B	00
P0-P3	FF

- All the ports upon RESET are configured as input, ready to be used as output ports
  - When the first 0 is written to a port, it becomes an output
  - To reconfigure it as an input, a 1 must be sent to the port
  - To use any of these ports as an input port, it must be programmed

- Example

```
        MOV A, #0FFH;     A=FF hex
        MOV P0, A;        make P0 an input port; by writing it with all 1s
        MOV A, #000H;     A=00 hex
        MOV P1, A;        make P1 an output port; by writing it with all 0s
LOOP:   MOV A, P0;        get data from P0
        MOV P1, A;        send it to port 1
        SJMP LOOP;        keep doing it
```

• In order for the RESET input to be effective, the high pulse must be high for a minimum of 2 machine cycles before it is allowed to go low



Power on reset circuit