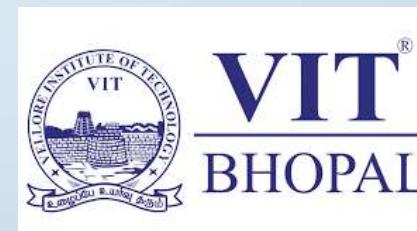




Microprocessors & Microcontrollers (ECE3004)

Module – 03 (Part – 01)
8051 Microcontroller

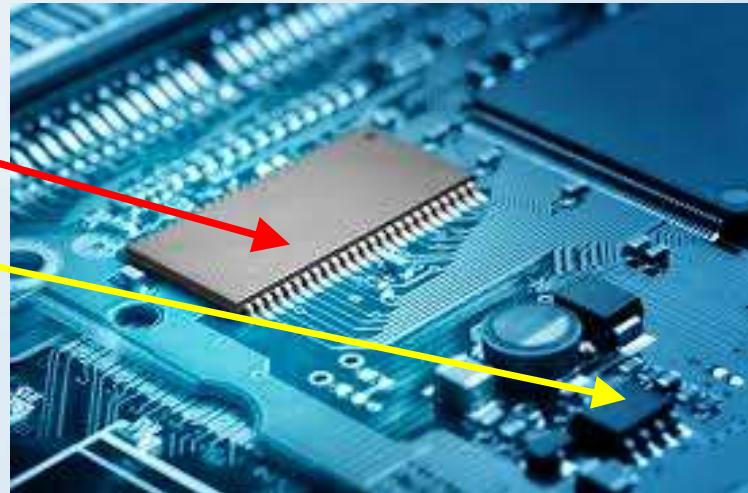
Dr. Susant Kumar Panigrahi
Assistant Professor
School of Electrical & Electronics Engineering



Learning Methodology

❖ Hardware Information

- Internal architecture of the processor (8085, 8086, 8051 etc.)
- Internal architecture of peripheral interface controller (viz. 8255, 8251, 8253, 8237, 8259 etc.)
- Circuit connections of the processor & peripheral devices (like connection among components in a PCB)



❖ Software Instructions

- Complex Instruction Set Computer (CISC)
- Reduced Instruction Set Computer (RISC)

Module-3 Syllabus

8051 Microcontroller:

- Intel MCS-51 family features – 8051 -organization and architecture, addressing modes, Instruction set, conditional instructions, I/O Programming, Arithmetic logic instructions, single bit instructions, interrupt handling, programming counters, timers and Stack.

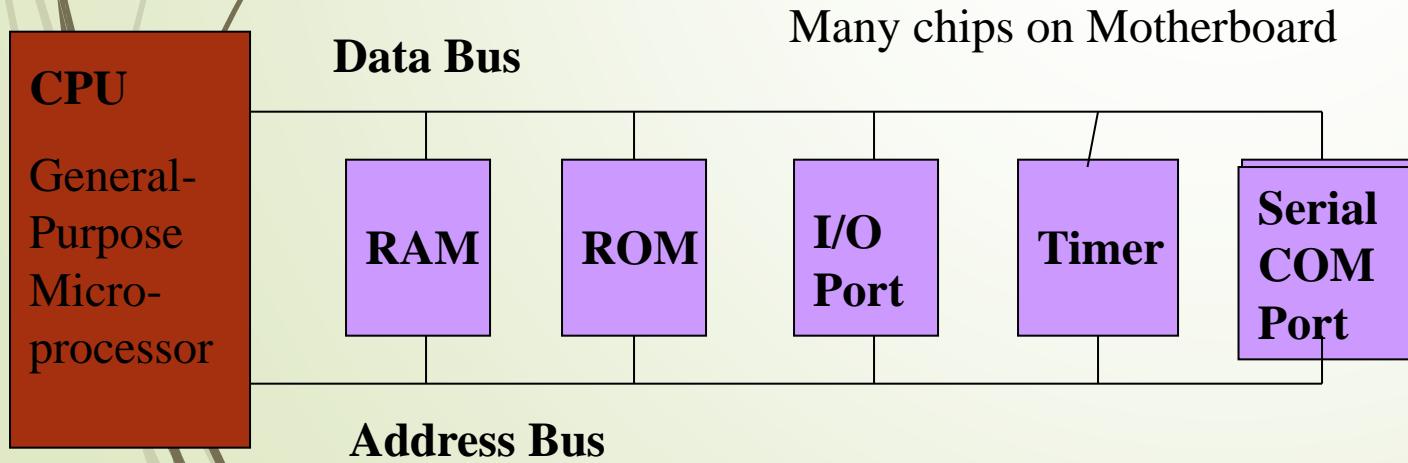
Unit-III : Introduction to Microcontrollers

Why μC Over μP ?

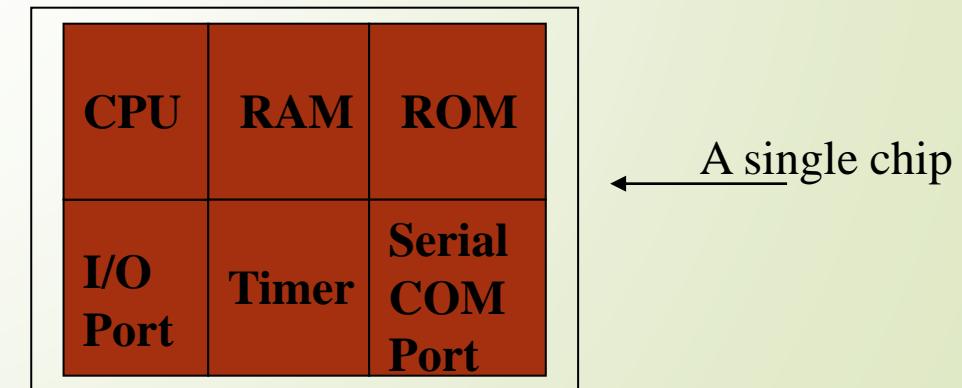
Microprocessor although very efficient in computing, has following limitations:

- ✓ The overall system cost is high.
- ✓ A large sized PCB is required for assembling all the components.
- ✓ Overall product design requires more time.
- ✓ Physical size of the product is big, no RAM, ROM, I/O on CPU chip itself.
- ✓ Discrete components are used, the system is not reliable.

General Purpose Microprocessor System



**A Single Chip Micro-Computer:
Microcontroller**



Embedded System & Popularity of μ C.

- ✓ Embedded system means the processor is embedded into that application.
- ✓ An embedded product uses a microcontroller (or μ P) to do one task only.
- ✓ In an embedded system, there is only one application software that is typically burned into ROM.
- ✓ *Example* : Printer, Keyboard, Video game player

**A Single Chip Micro-Computer:
Microcontroller**

CPU	RAM	ROM
I/O Port	Timer	Serial COM Port

← A single chip

Advantages of Microcontroller based System

7

- ✓ As the peripherals are integrated into a single chip, the overall system cost is very less
- ✓ The product is of small size compared to microprocessor based system
- ✓ The system design now requires very little efforts
- ✓ As the peripherals are integrated with a microprocessor the system is more reliable
- ✓ Though microcontroller may have on chip ROM, RAM and I/O ports, additional ROM, RAM I/O ports may be interfaced externally if required
- ✓ On chip ROM provide a software security

Components of Microcontroller

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

Applications of μ C:

1. Home Appliances control
2. Calculator
3. Toys
4. Automobile engine control systems
5. implantable medical devices
6. power tools
7. Other Embedded Systems.....

Three criteria in Choosing a Microcontroller

8

1. Meeting the computing needs of the task efficiently and cost effectively
 - speed, the amount of ROM and RAM, the number of I/O ports and timers, size, packaging, power consumption
 - easy to upgrade
 - cost per unit
2. Availability of software development tools
 - assemblers, debuggers, C compilers, emulator, simulator, technical support
3. Wide availability and reliable sources of the microcontrollers.

Types of Microcontroller

9

- Atmel
 - ARM
 - Intel
 - 8-bit
 - 8XC42
 - MCS48
 - **MCS51**
 - 8xC251
 - 16-bit
 - MCS96
 - MXS296
 - National Semiconductor
 - COP8
 - Microchip
 - 12-bit instruction PIC
 - 14-bit instruction PIC
 - PIC16F84
 - 16-bit instruction PIC
 - NEC
-
- Motorola
 - 8-bit
 - 68HC05
 - 68HC08
 - 68HC11
 - 16-bit
 - 68HC12
 - 68HC16
 - 32-bit
 - 683xx
 - Texas Instruments
 - TMS370
 - MSP430
 - Zilog
 - Z8
 - Z86E02

MCS-51 “Family” of Microcontrollers

10

Feature	8031	8051	8052	8751
ROM	NO	4kB	8kB	4kB UV Eprom
RAM (Bytes)	128	128	256	128
TIMERS	2	2	3	2
I/O PINS	32	32	32	32
SERIAL PORTS	1	1	1	1
INTERRUPT SOURCES	6	6	8	6

Introduction to 8051 Microcontroller



8051 Features

1. 8-bit CPU with registers A (accumulator) and B
2. 16-bit program counter (PC) and data pointer (DPTR)
3. 8-bit program status word (PSW)
4. 8-bit stack pointer (SP)
5. Internal ROM or EPROM of 4k (8051)
6. Internal RAM of 128 bytes
 - 4 register banks each containing 8 registers of 8-bit
 - 16-bytes bit addressable registers
 - 80 bytes general purpose data memory
7. 32 input-output pins arranged as 4 eight-bit ports (P0 – P3)
8. 3 16 bit timer/counters : T0, T1, & T2
9. Full duplex serial data receiver/transmitter
10. Control registers : TCON, TMOD, SCON, PCON, IP, and IE, T2CON, T2MOD
11. 2 external and 3 timer interrupt and 1 serial interrupt sources
12. Oscillator and clock circuit

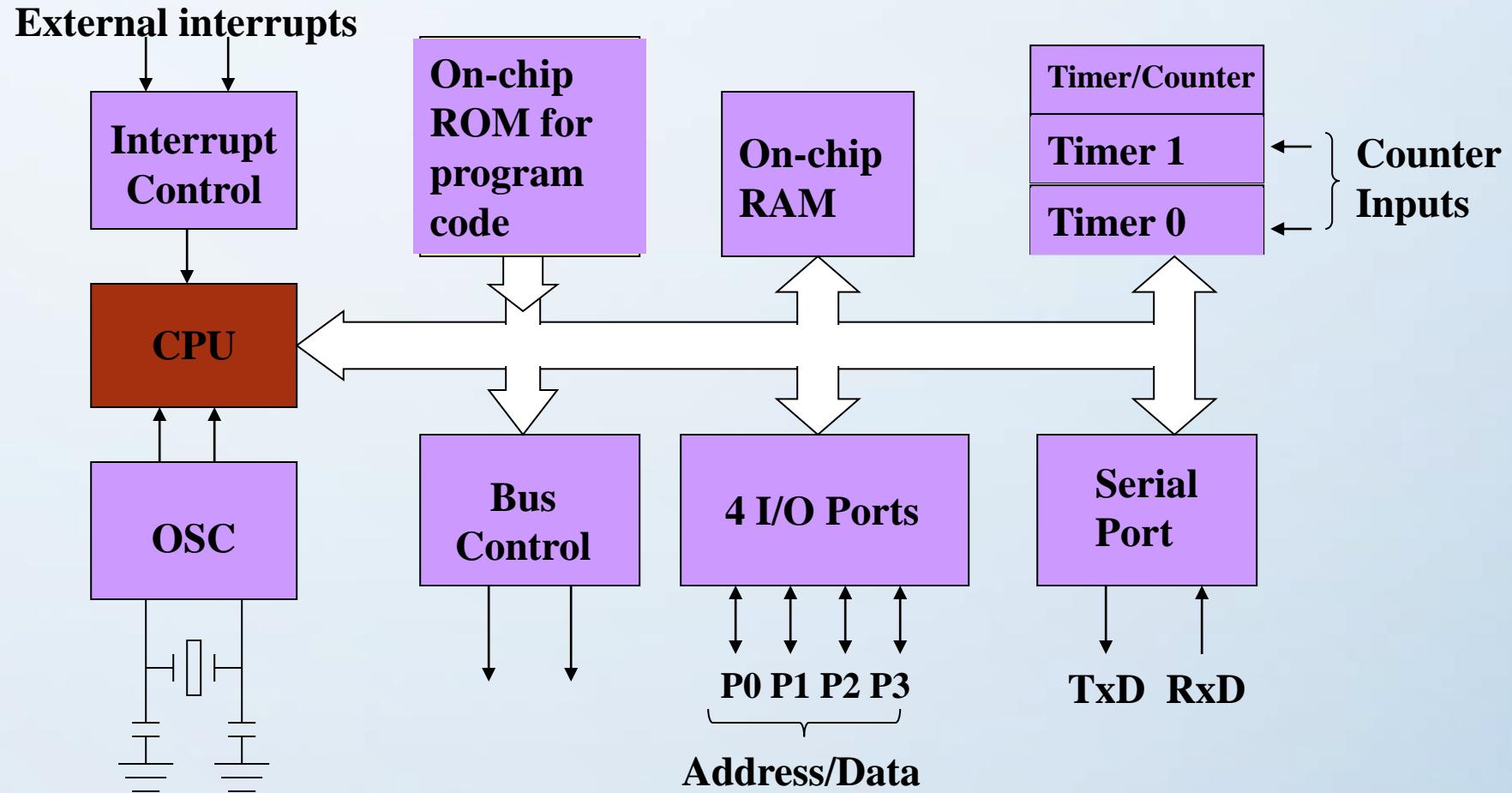
Internal Architecture

- Registers
 - PSW, A
 - B
 - Special Function Registers (like TMOD, TCON, IP, IE etc.)
 - Program Counter,
 - Stack Pointer
- Memory: ROM, RAM
- Peripheral I/O Controller
 - Parallel: Port (P0 – P3)
 - Serial: P3.0, P3.1
 - Interrupt: P3.2, P3.3
 - Timer: T0, T1, T2
 - SPI: P1.4 – P1.7

External Components

- Peripheral Devices
 - ADC, DAC
 - Keyboard
 - Display: 7-Segment, LCD etc.
 - ...Many more

8051 Block Diagram



8051 Pin Diagram

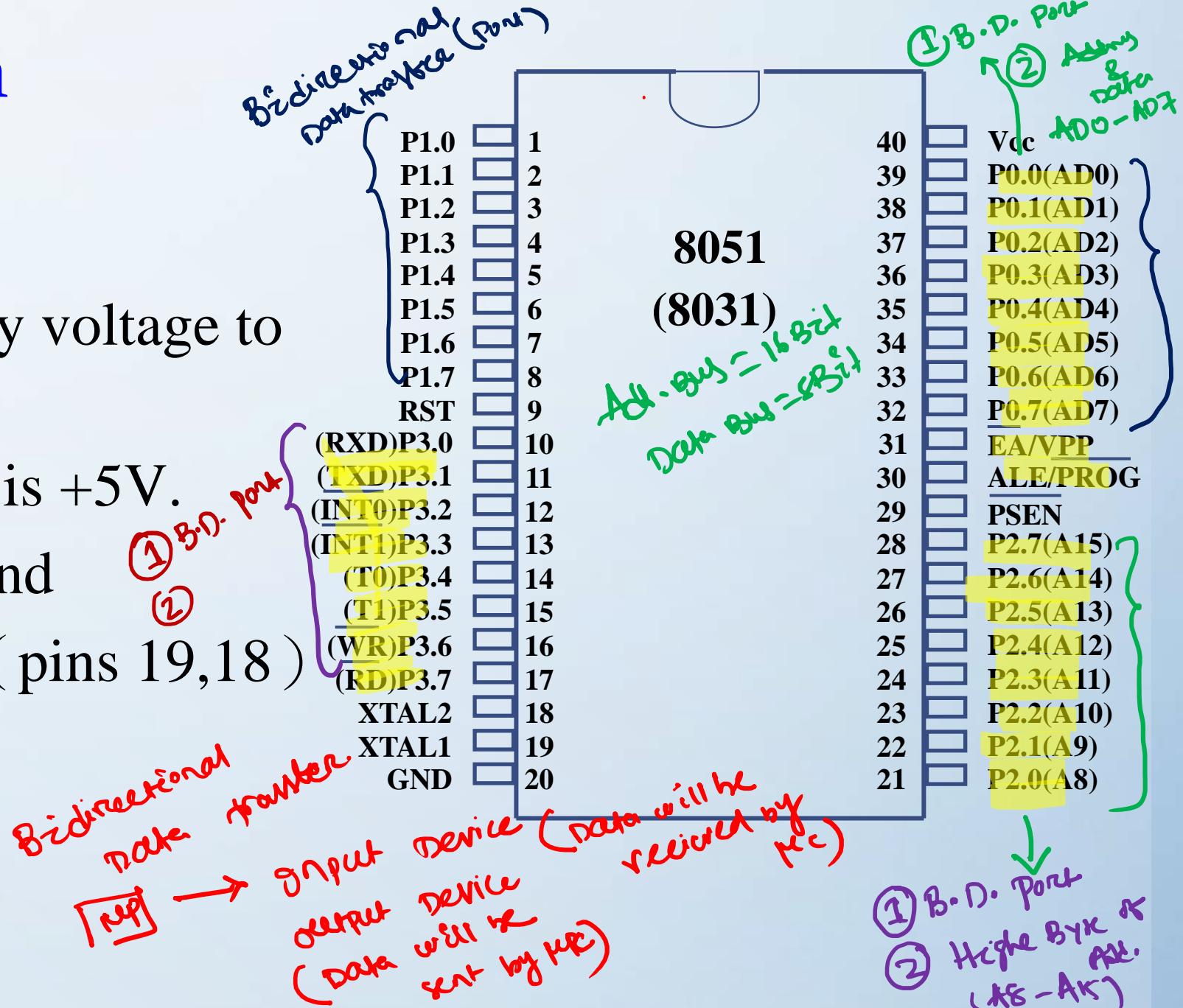
✓ Vcc (pin 40) :

- Vcc provides supply voltage to the chip.
- The voltage source is +5V.

✓ GND (pin 20) : ground

✓ XTAL1 and XTAL2 (pins 19,18)

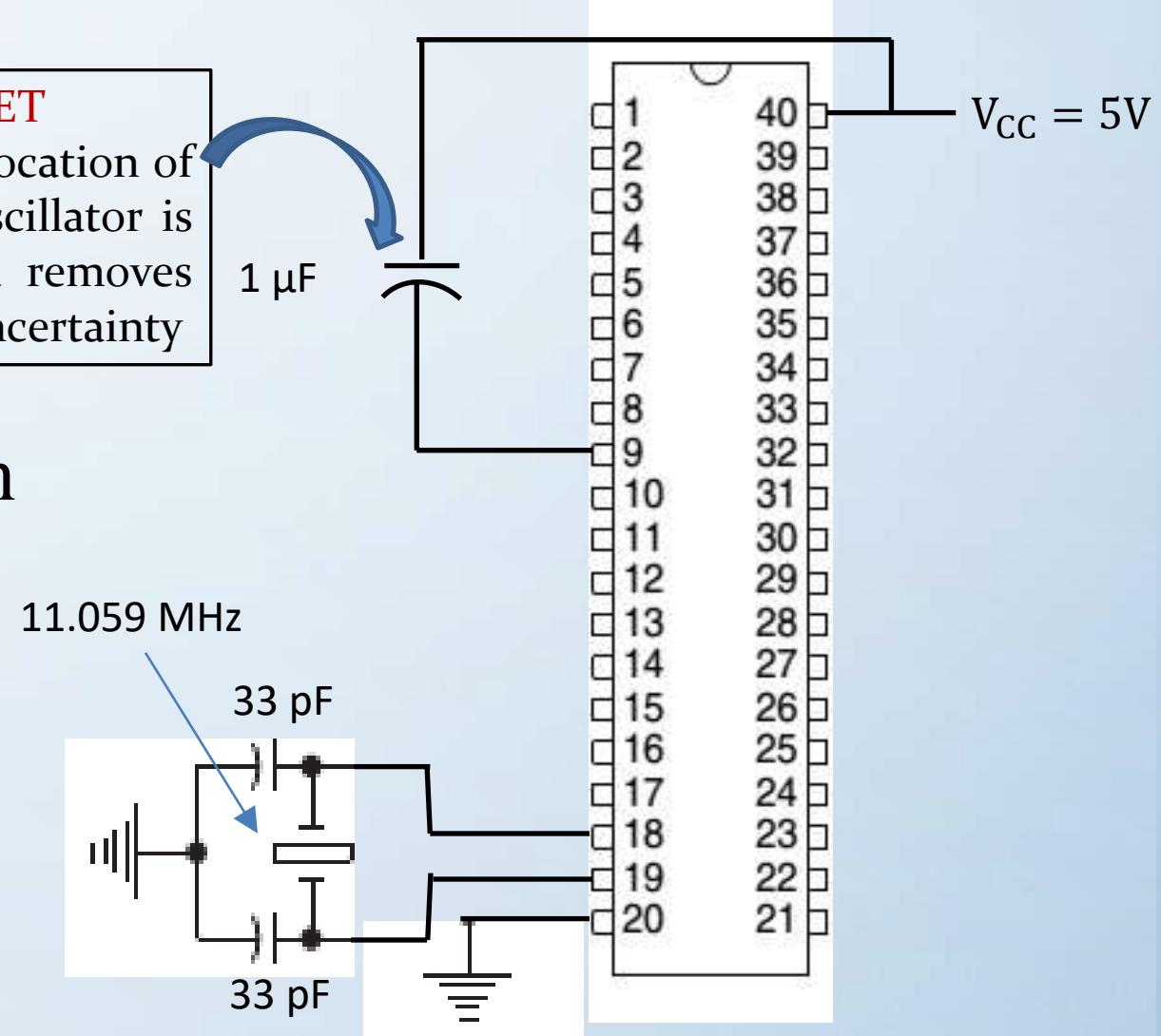
4- Ports
P0 P1 P2 P3
8-pin
(8-Bits)



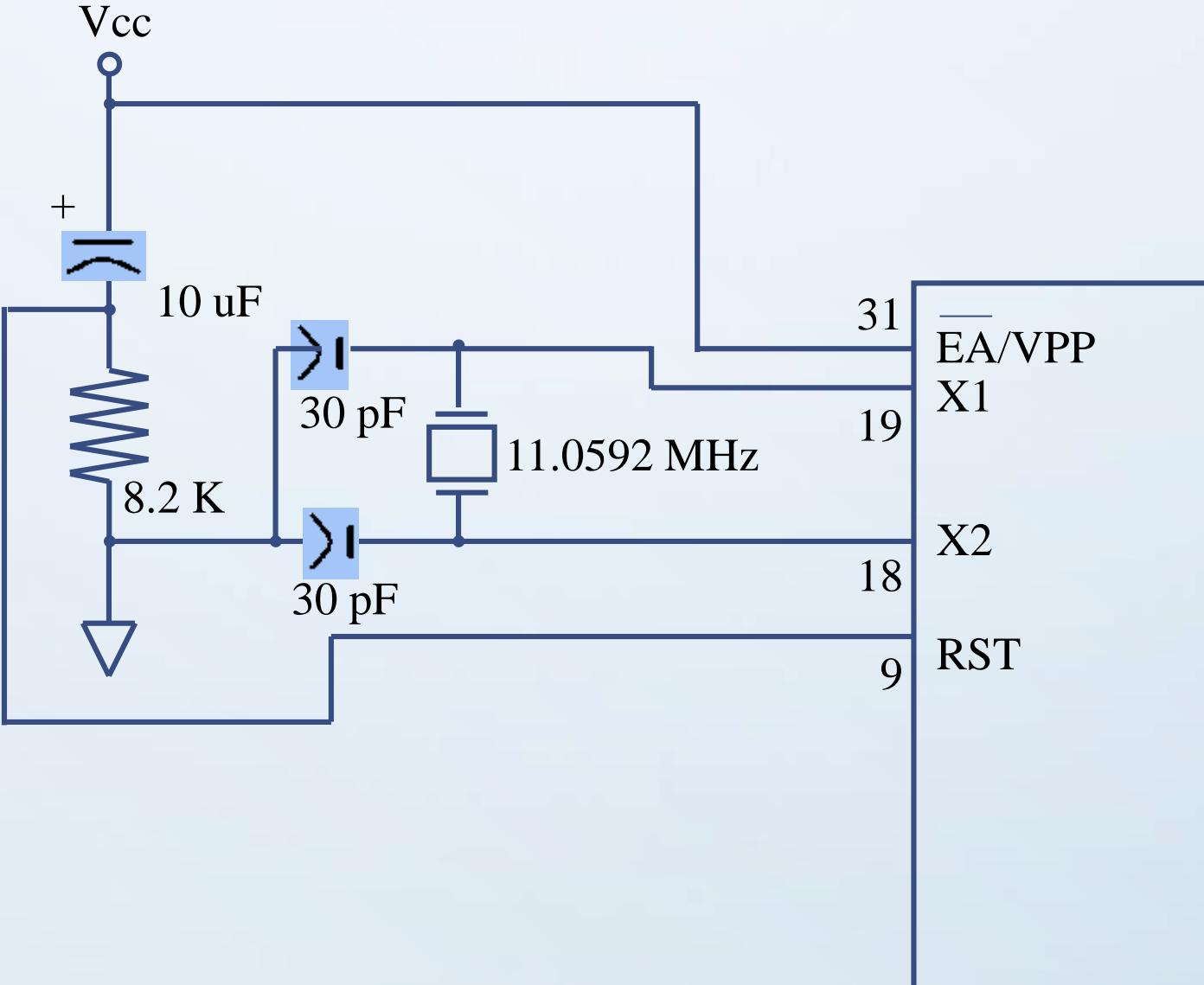
8051 Pin Diagram

- ✓ XTAL 01 (Pin 19), XTAL 02 (Pin 18): External oscillator(quartz).
- ✓ Using a quartz crystal oscillator
- ✓ We can observe the frequency on the XTAL2 pin.

Power on RESET
PC points '0' location of ROM after Oscillator is stabilized and removes the start up uncertainty



8051 Pin Diagram



- RST (Pin 9) : Reset
 - ✓ It is an input pin and is active high (normally low) .
 - ✓ The high pulse must be high at least 2 machine cycles.
 - ✓ It is a power-on reset.
 - ✓ Upon applying a high pulse to RST, the microcontroller will reset and all values in registers will be lost.
 - ✓ Reset values of some 8051 registers

8051 Pin Diagram

- ✓ \overline{EA} (Pin 31) : external access
 - There is no on-chip ROM in 8031 and 8032 .
 - The \overline{EA} pin is connected to GND to indicate the code is stored externally.
 - \overline{PSEN} & ALE are used for external ROM.
 - For 8051, /EA pin is connected to Vcc.
- ✓ \overline{PSEN} (Pin 29) : Program store enable
 - This is an output pin and is connected to the OE pin of the ROM.
- ✓ ALE (Pin 30) : address latch enable
 - It is an output pin and is active high.
 - 8051 port 0 provides both address and data.
 - The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 latch.
- ✓ I/O port pins
 - The four ports P0, P1, P2, and P3.
 - Each port uses 8 pins.
 - All I/O pins are bi-directional.

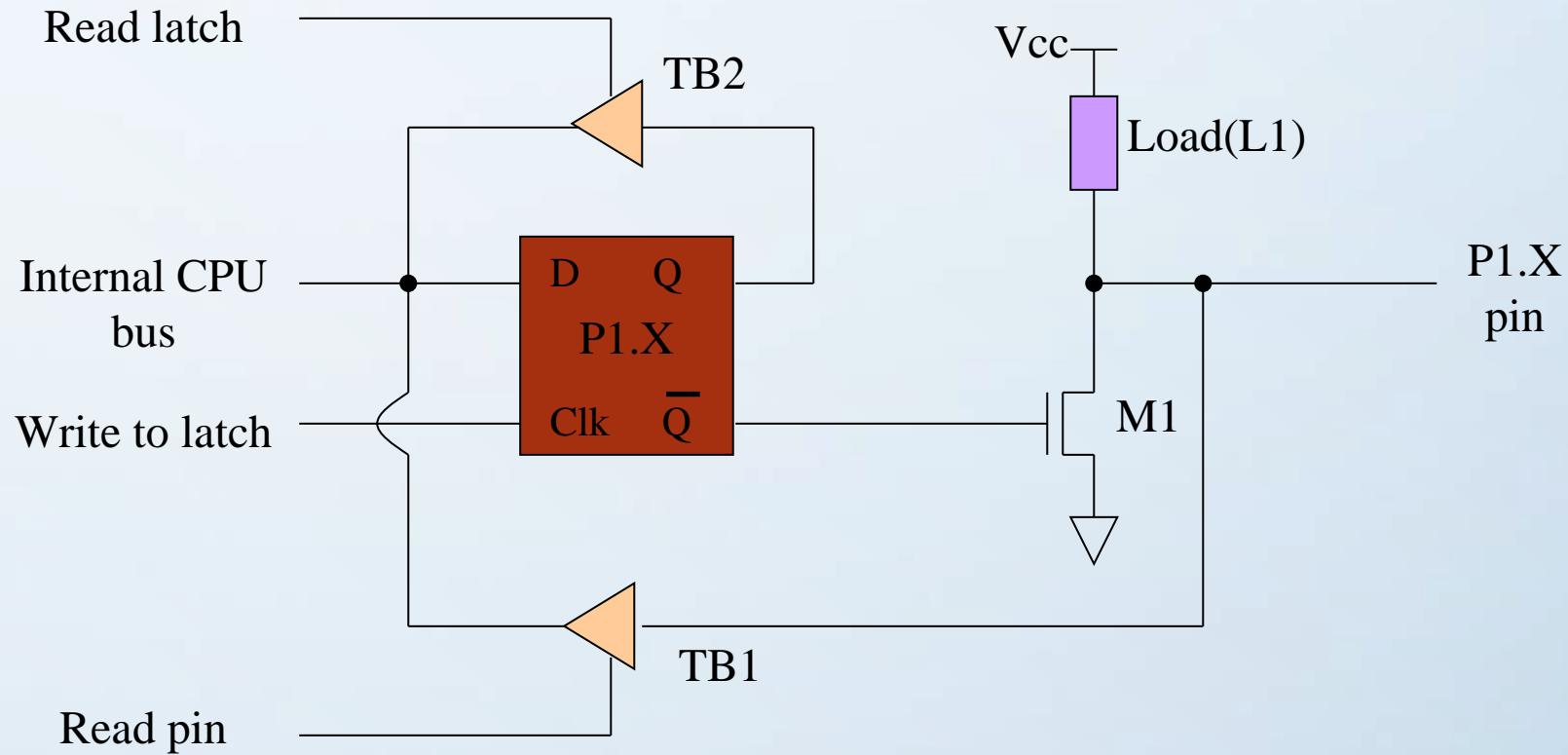
Pins of I/O Port

- ✓ The 8051 has four I/O ports
 - Port 0 (pins 32-39) : P0 (P0.0~P0.7)
 - Port 1 (pins 1-8) : P1 (P1.0~P1.7)
 - Port 2 (pins 21-28) : P2 (P2.0~P2.7)
 - Port 3 (pins 10-17) : P3 (P3.0~P3.7)
 - Each port has 8 pins.
 - Named P0.X (X=0,1,...,7) , P1.X, P2.X, P3.X
 - Ex : P0.0 is the bit 0 (LSB) of P0
 - Ex : P0.7 is the bit 7 (MSB) of P0
 - These 8 bits form a byte.
- ✓ Each port can be used as input or output (bi-direction).

Hardware Structure of I/O Pin

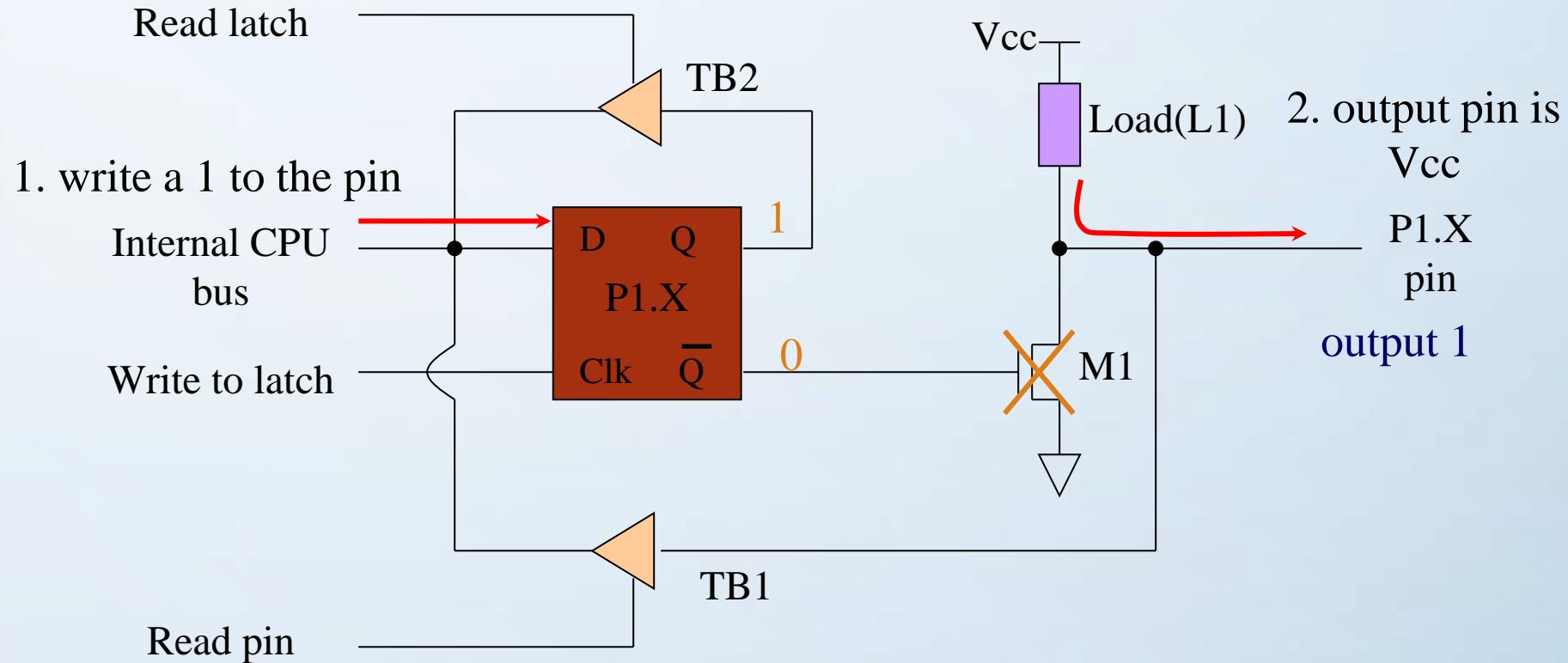
- Each pin of I/O ports
 - Internal CPU bus : communicate with CPU
 - A D latch store the value of this pin
 - D latch is controlled by “Write to latch”
 - $\text{Write to latch} = 1$: write data into the D latch
 - 2 Tri-state buffer :
 - TB1: controlled by “Read pin”
 - $\text{Read pin} = 1$: really read the data present at the pin
 - TB2: controlled by “Read latch”
 - $\text{Read latch} = 1$: read value from internal latch
 - A transistor M1 gate
 - Gate=0: open
 - Gate=1: close

A Pin of Port 1



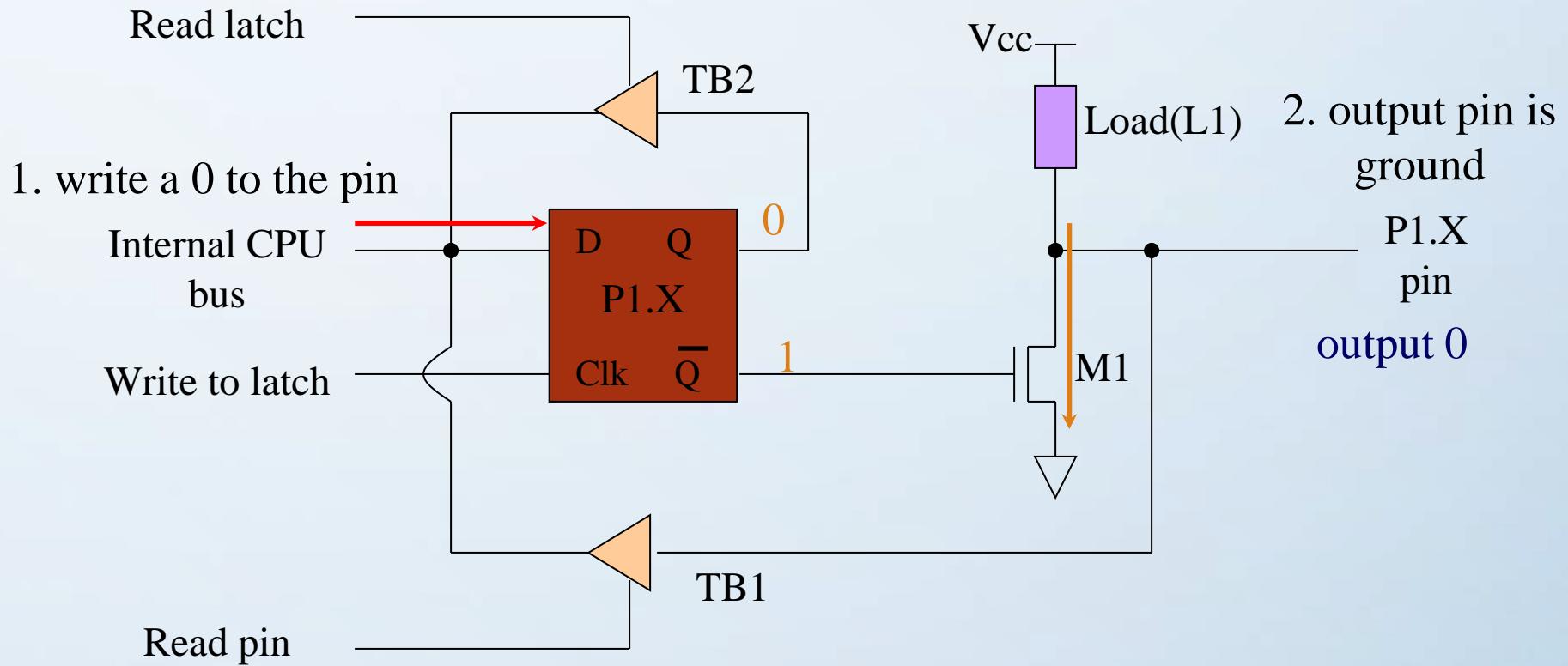
8051 IC

Writing “1” to Output Pin P1.X



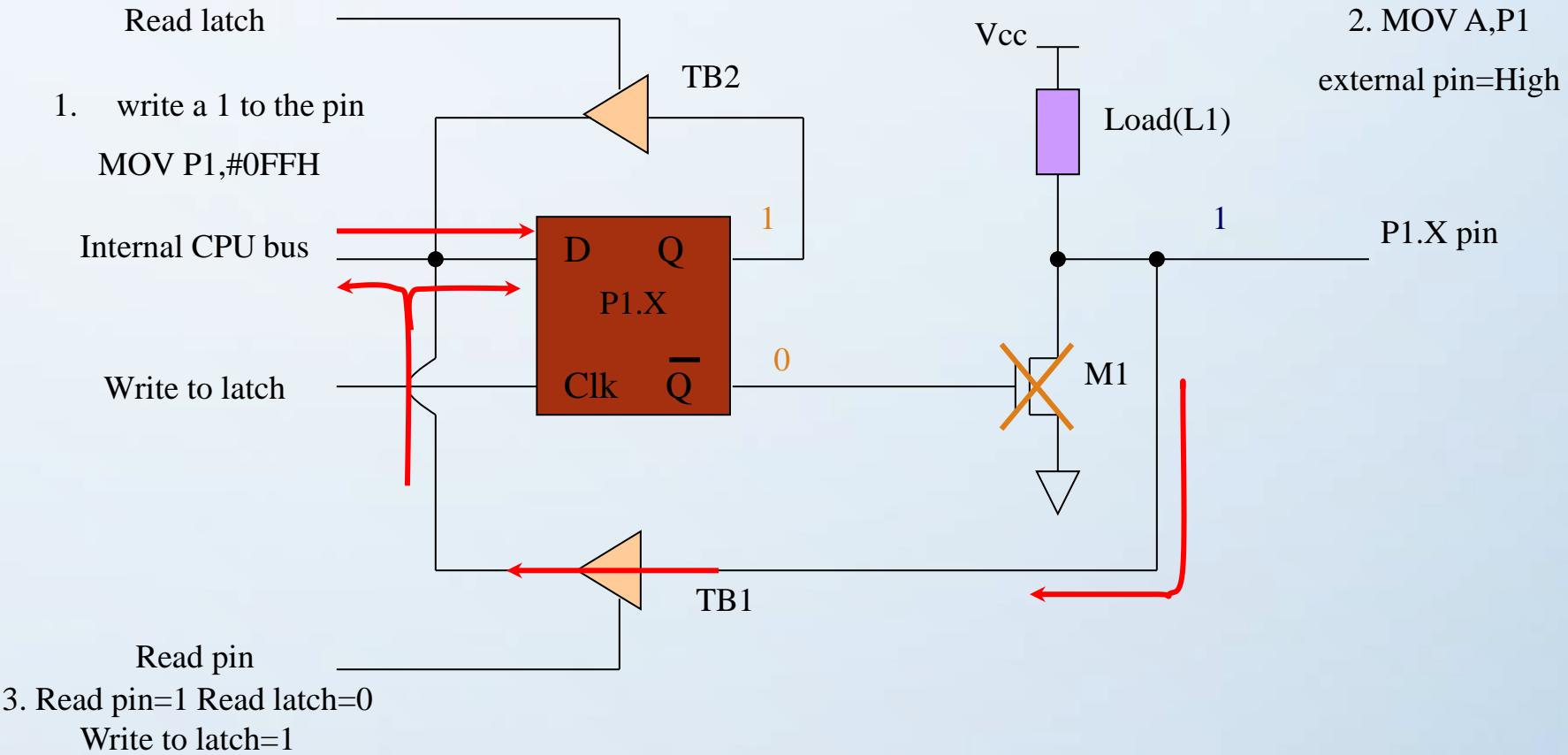
8051 IC

Writing “0” to Output Pin P1.X



8051 IC

Reading “High” at Input Pin

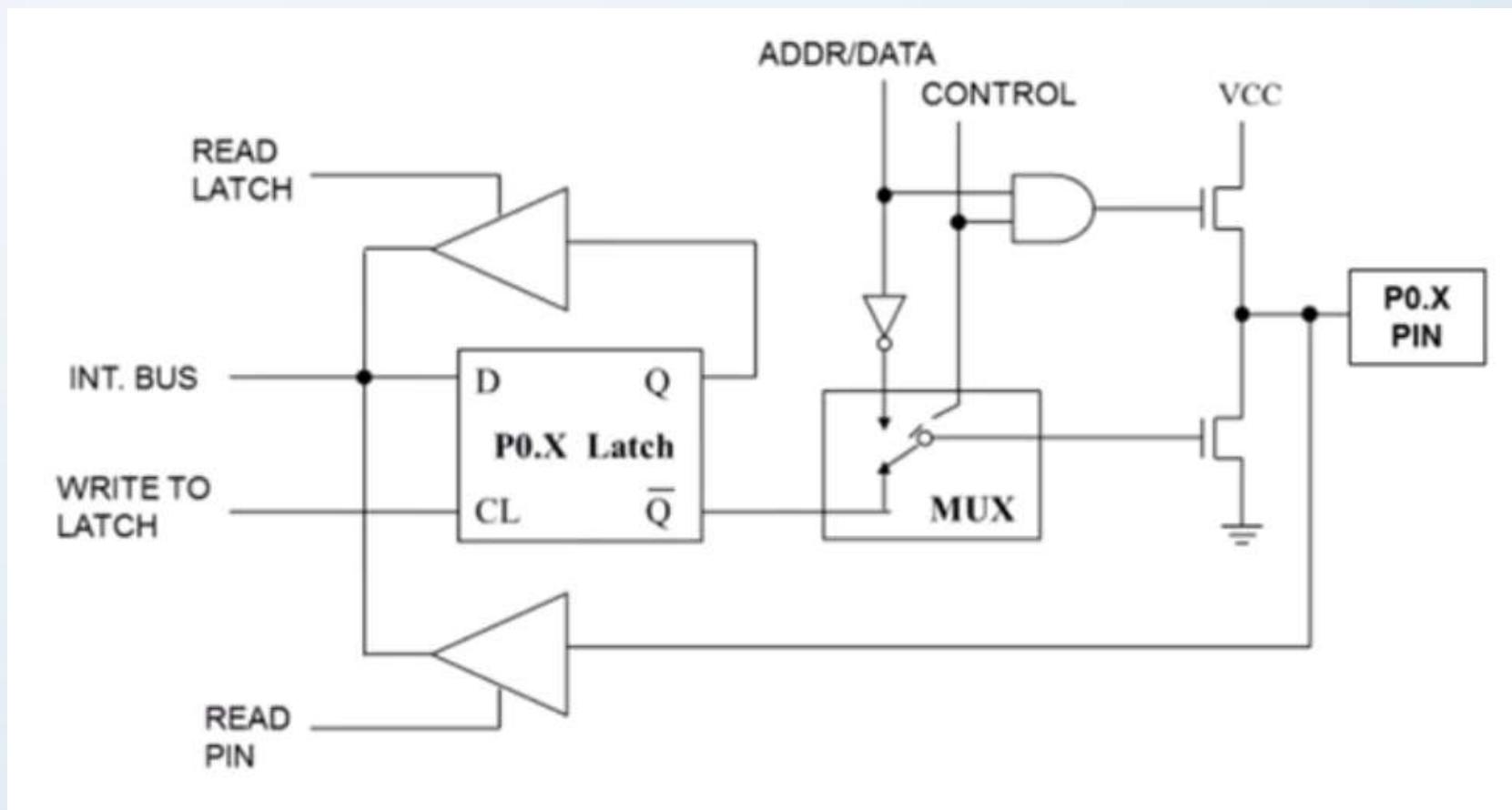


8051 IC

Other Pins

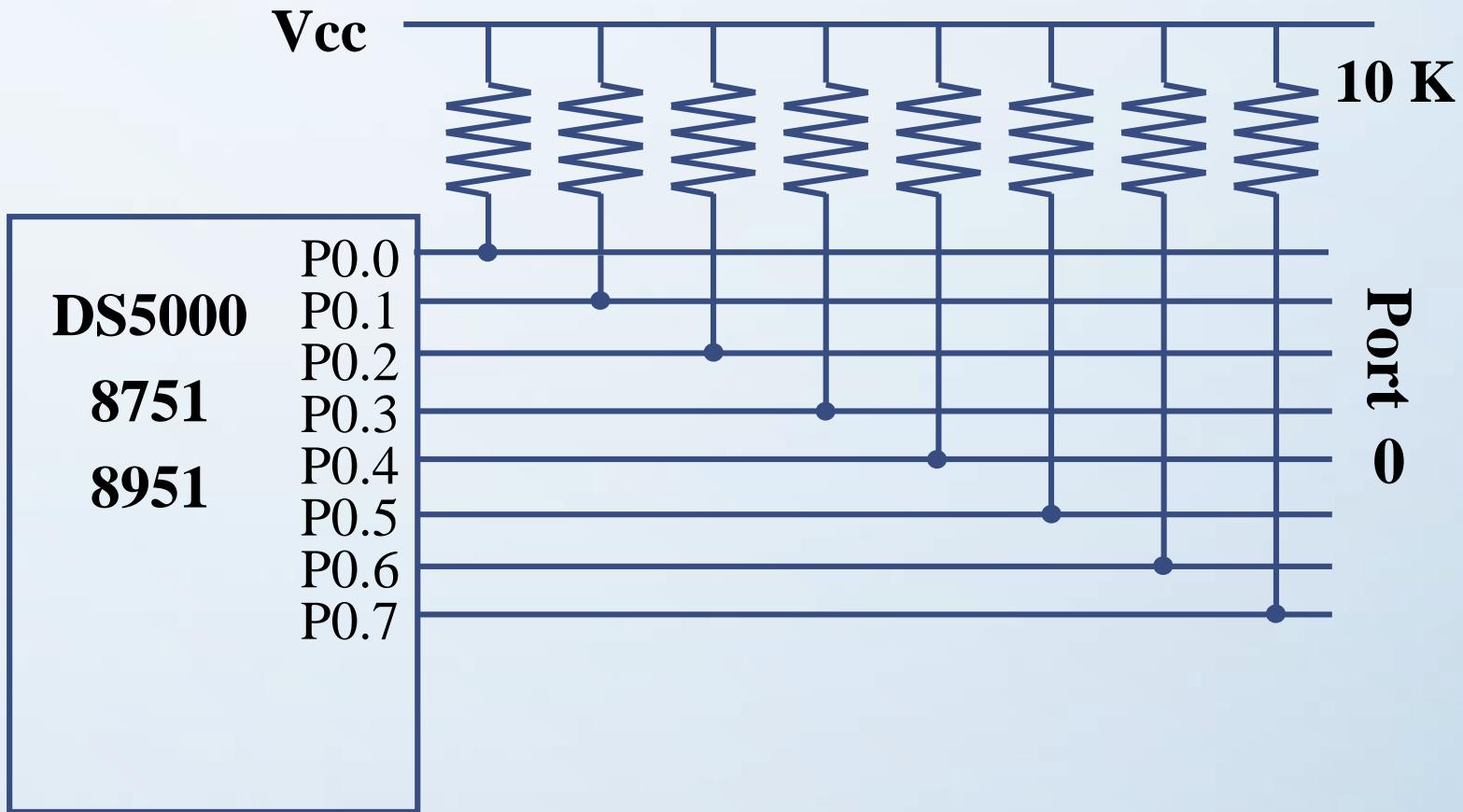
- Port P1, P2, and P3 have internal pull-up resistors.
 - P1, P2, and P3 are not open drain.
- Port P0 has no internal pull-up resistors and does not connects to Vcc inside the 8051.
 - P0 is open drain.
 - Compare the figures of P1.X and P0.X.
- However, for a programmer, it is the same to program P0, P1, P2 and P3.
- All the ports upon RESET are configured as output.

A Pin of Port P0



8051 IC

A Pin of Port P0



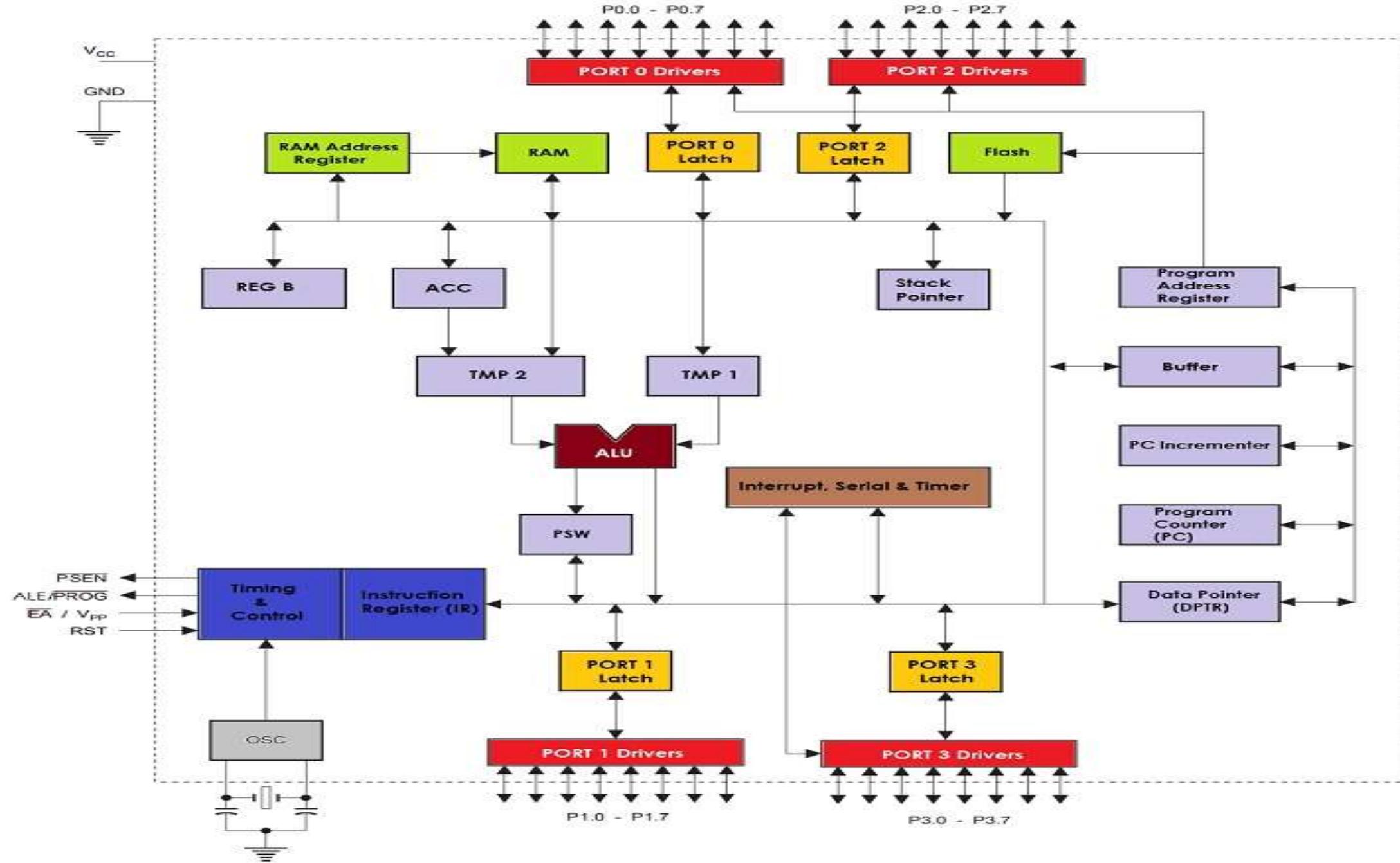
Port P₃, Alternative Function

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

8051 Microcontroller: Internal Architecture

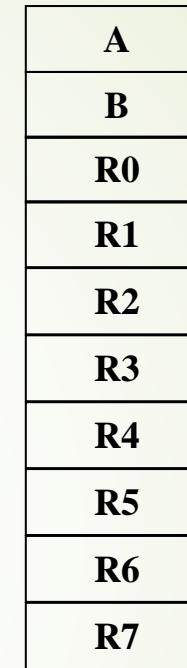


8051 Architecture Diagram

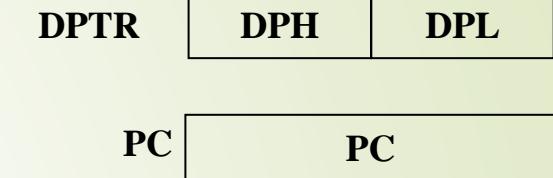


Register Organization: 8051

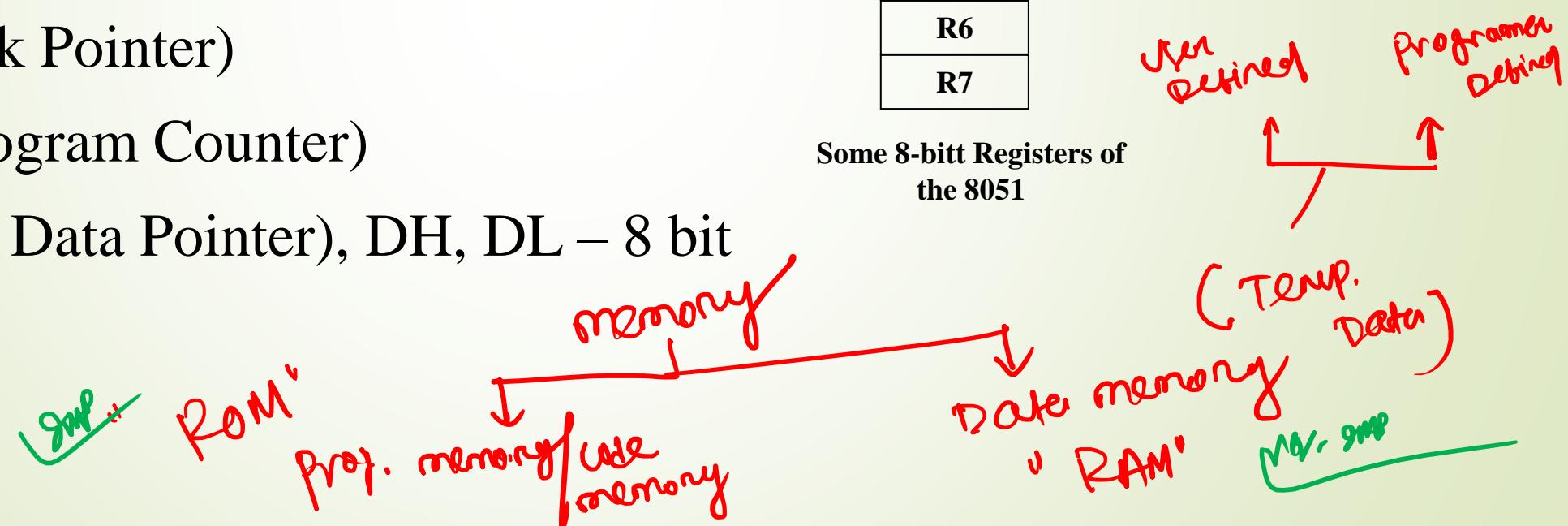
- ✓ A (8-bit Accumulator)
- ✓ B (8-bit register for Multiplication & Division)
- ✓ PSW (8-bit Program Status Word) (*same function as flag in 8086*)
- ✓ SP (8-bit Stack Pointer)
- ✓ PC (16-bit Program Counter)
- ✓ DPTR (16-bit Data Pointer), DH, DL – 8 bit each



Some 8-bit Registers of the 8051



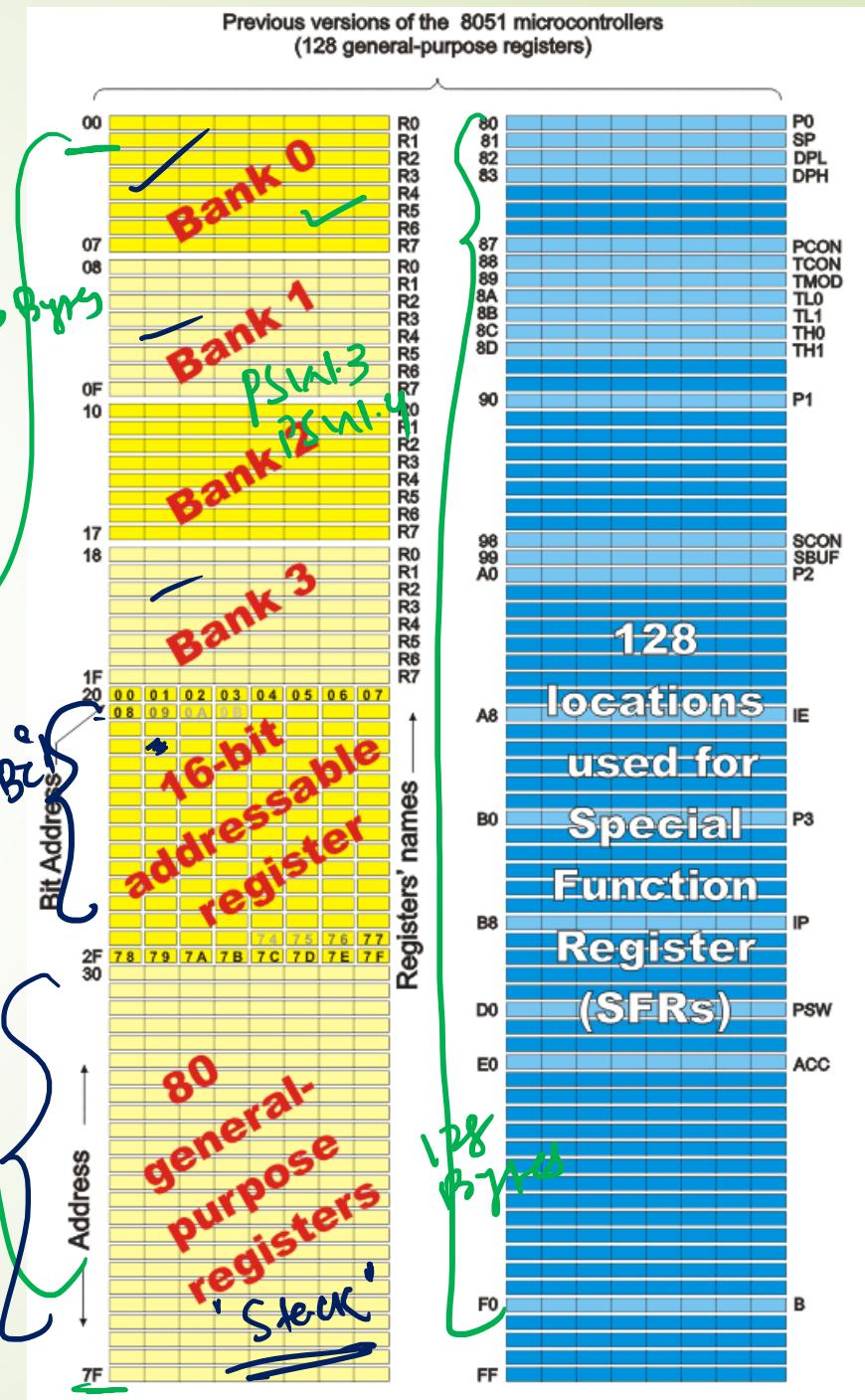
Some 8051 16-bit Register



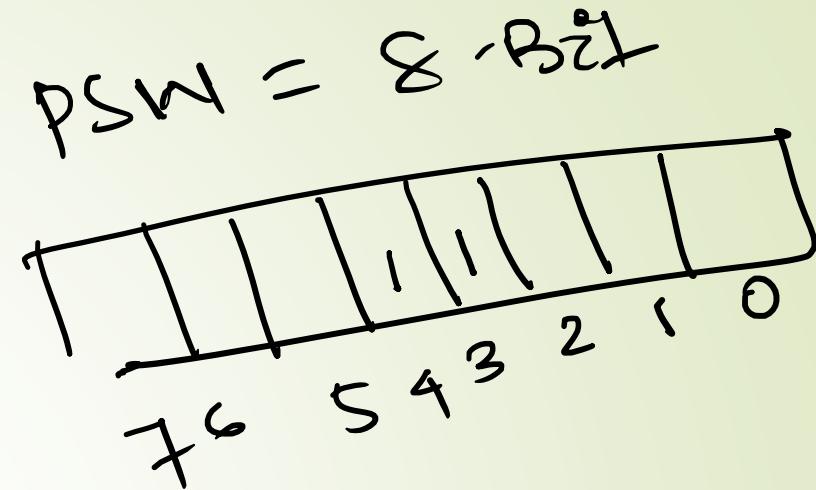
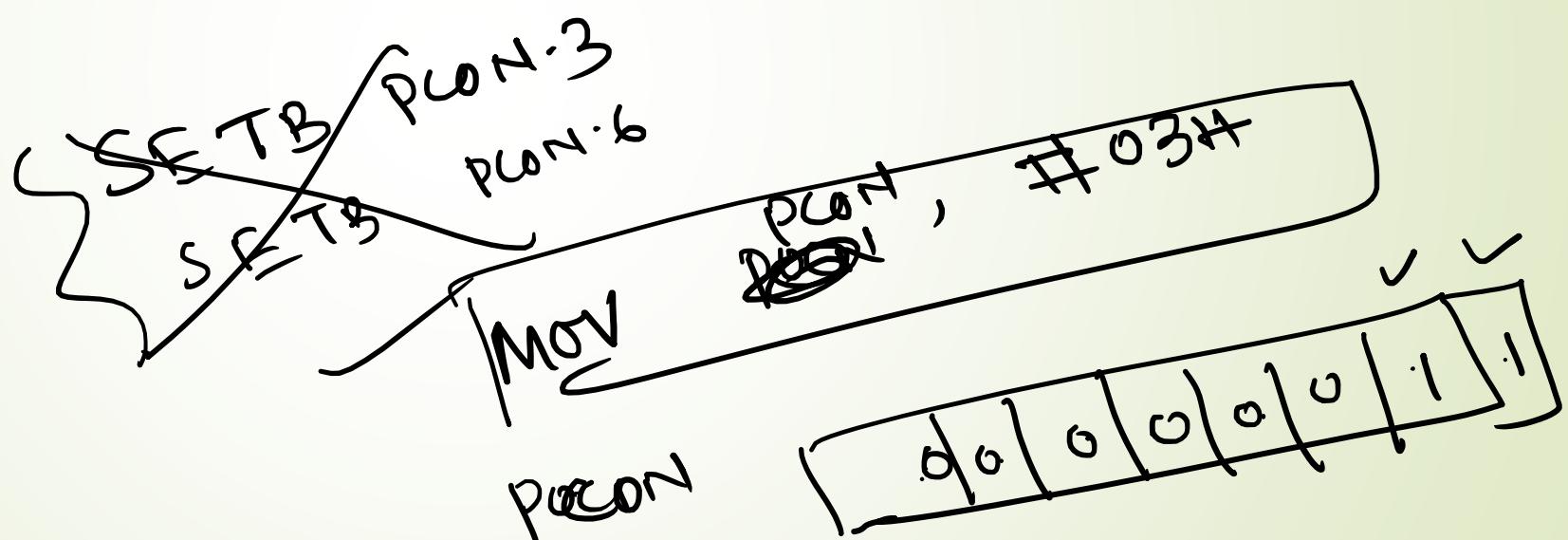
Special Function Registers (SFR)

RAM = 128 Bytes

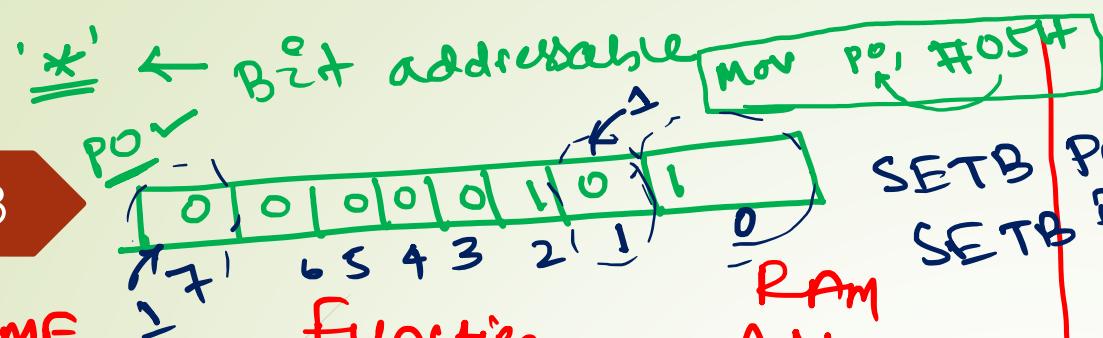
- ✓ Special Function Registers act as a control table that monitor and control the operation of the 8051 Microcontroller.
- ✓ By manipulating the Special Function Registers (SFRs), we can assess or change the operating mode of the 8051 Microcontroller.
- ✓ Address Space from 80H to FFH is allocated to SFRs. 128 Bytes
- ✓ Out of these 128 Memory Locations (80H to FFH), there are only 21 locations that are actually assigned to SFRs.
- ✓ Each SFR has one Byte Address and also a unique name which specifies its purpose.
- ✓ Since the SFRs are a part of the Internal RAM Structure, you can access SFRs as if you access the Internal RAM.
- ✓ The main difference is the address space: first 128 Bytes (00H to 7FH) is for regular Internal RAM and next 128 Bytes (80H to FFH) is for SFRs.



32



33



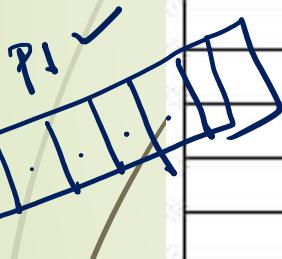
<u>NAME</u>	<u>Function</u>	<u>RAM Address</u>
A*	Accumulator	0E0H
B*	MUL, DIV	0F0H
DPL	External memory	83H
DPL	"	82H
DPTR		
IE*	Interrupt Enable	A8H
IP*	Interrupt Priority	B8H
PO*	I/O Port 0	80H
P1*	I/O Port 1	80H

<u>NAME</u>	<u>Function</u>	<u>RAM Add.</u>
P2*	I/O Port 2	A0
P3*	I/O Port 3	B0
PCON	Power control	87
PSW*	Program Status word	DO
SCON*	Serial Port	98
SBUF	Serial Data Buffer	99
SP	Stack pointer	81
TMOD	Timer mode control	89
TCON*	Timer control register	88
TL0	Timer 0 lower Byte	8A
TH0	Timer 0 Higher Byte	8C
TL1	Timer 1 lower Byte	8B
TH1	Timer 1 Higher Byte	8D

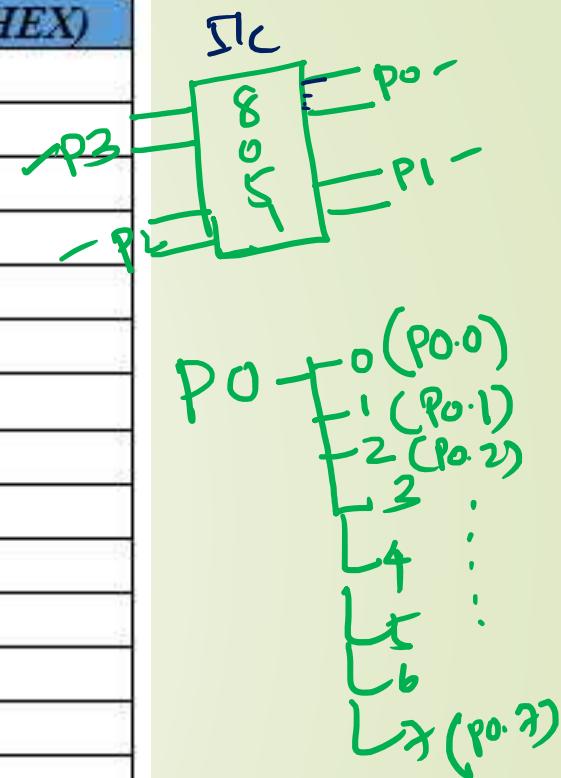
Special Function Registers (SFR) and RAM Locations

34

Timer 2
Counters



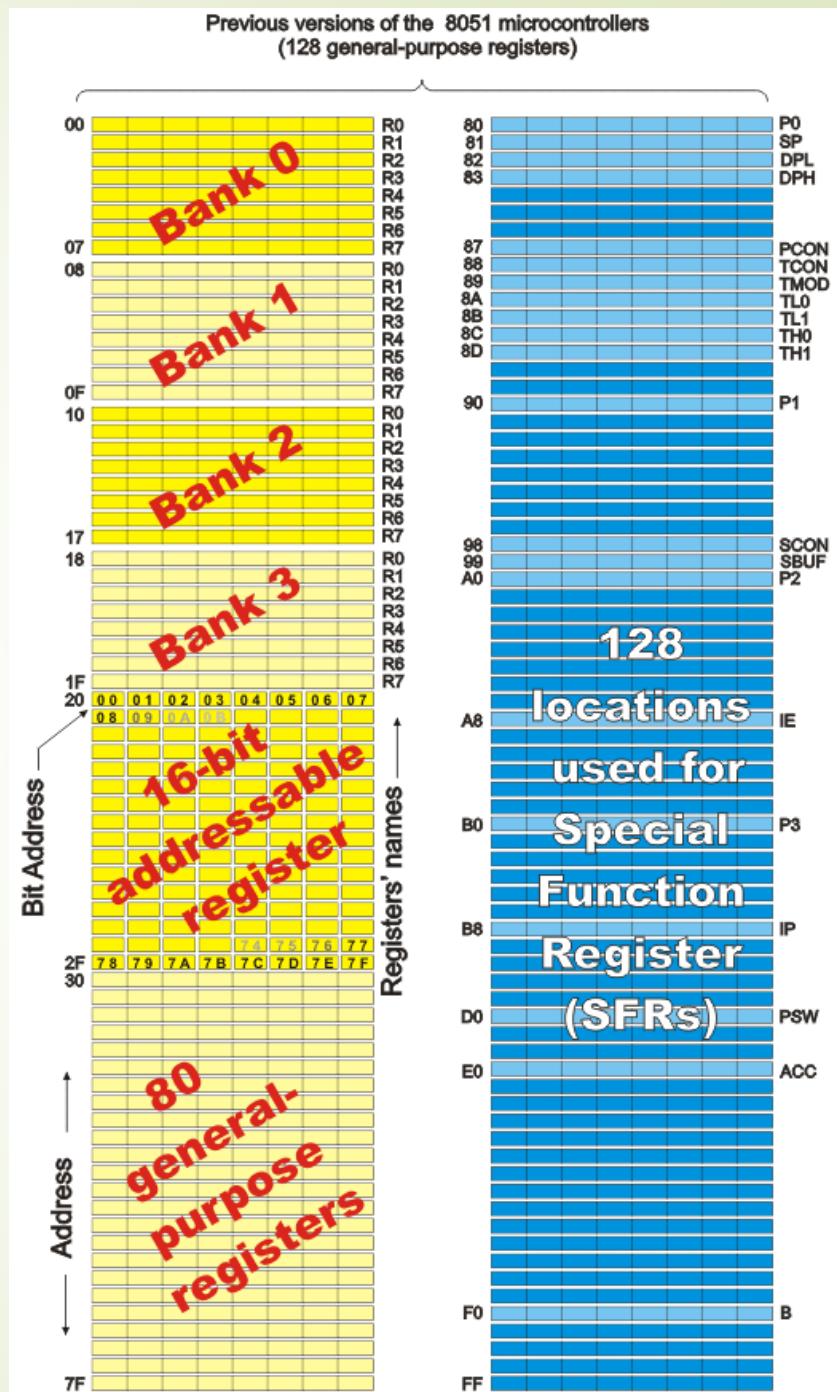
Name of the Register	Function	Internal RAM Address (HEX)
ACC (A)	Accumulator	E0H
B	B Register (for Arithmetic)	F0H
DPH	Addressing External Memory	83H
DPL	Addressing External Memory	82H
IE ✓	Interrupt Enable Control	A8H
IP ✓	Interrupt Priority	B8H
P0 ✓	PORT 0 Latch	80H
P1 ✓	PORT 1 Latch	90H
P2 ✓	PORT 2 Latch	A0H
P3 ✓	PORT 3 Latch	B0H
PCON	Power Control	87H
PSW	Program Status Word	D0H
SCON	Serial Port Control	98H
SBUF	Serial Port Data Buffer	99H
SP	Stack Pointer	81H
TMOD ✓	Timer / Counter Mode Control	89H
TCON ✓	Timer / Counter Control	88H
TL0 ✓	Timer 0 LOW Byte	8AH
TH0 ✓	Timer 0 HIGH Byte	8CH
TL1 ✓	Timer 1 LOW Byte	8BH
TH1 ✓	Timer 1 HIGH Byte	8DH



ELECTRONICS HUB

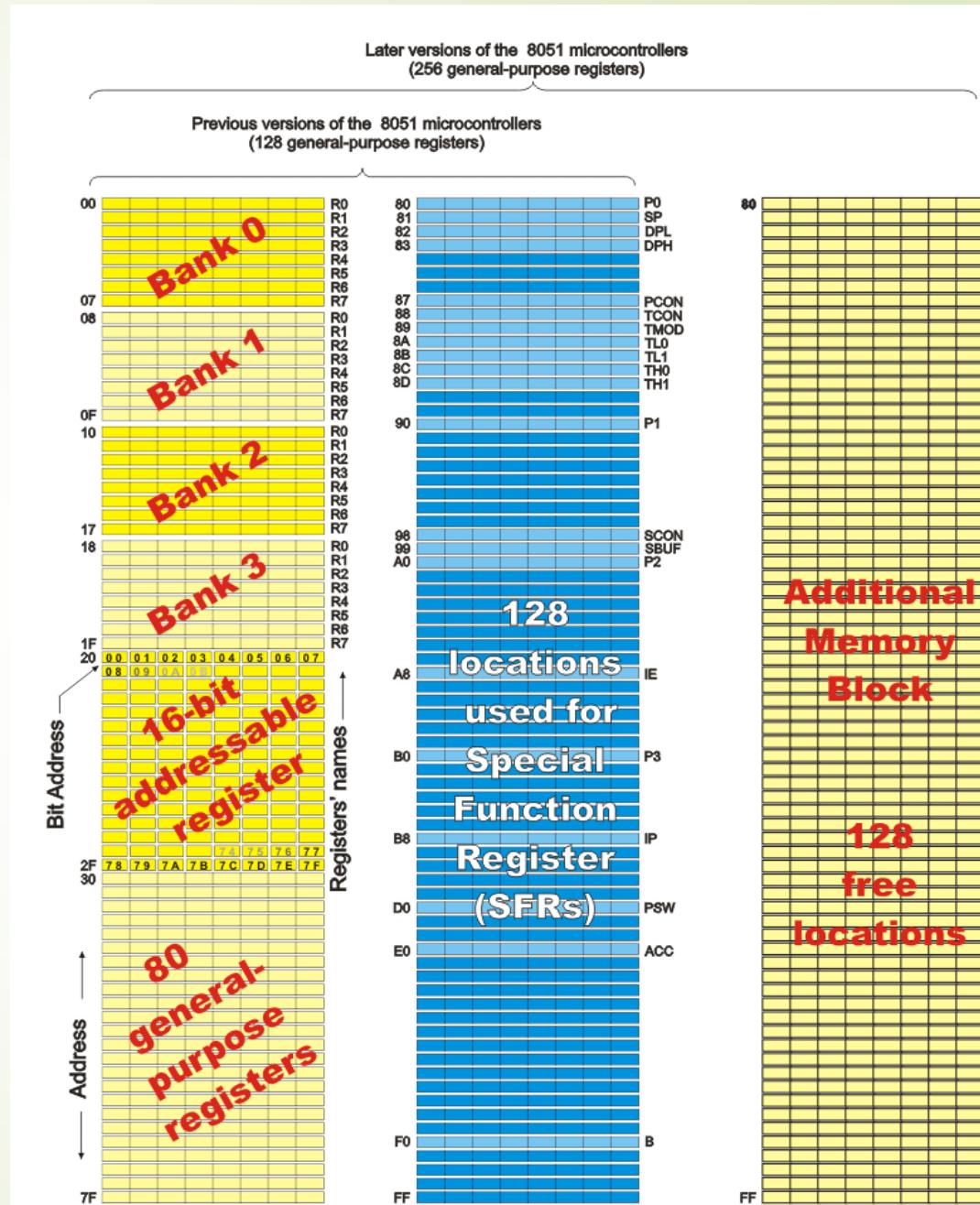
SFR 8051

- ✓ **Math or CPU Registers:** A and B
- ✓ **Status Register:** PSW (Program Status Word)
- ✓ **Pointer Registers:** DPTR (Data Pointer – DPL, DPH) and SP (Stack Pointer)
- ✓ **I/O Port Latches:** P0 (Port 0), P1 (Port 1), P2 (Port 2) and P3 (Port 3)
- ✓ **Peripheral Control Registers:** PCON, SCON, TCON, TMOD, IE and IP
- ✓ **Peripheral Data Registers:** TL0, TH0, TL1, TH1 and SBUF

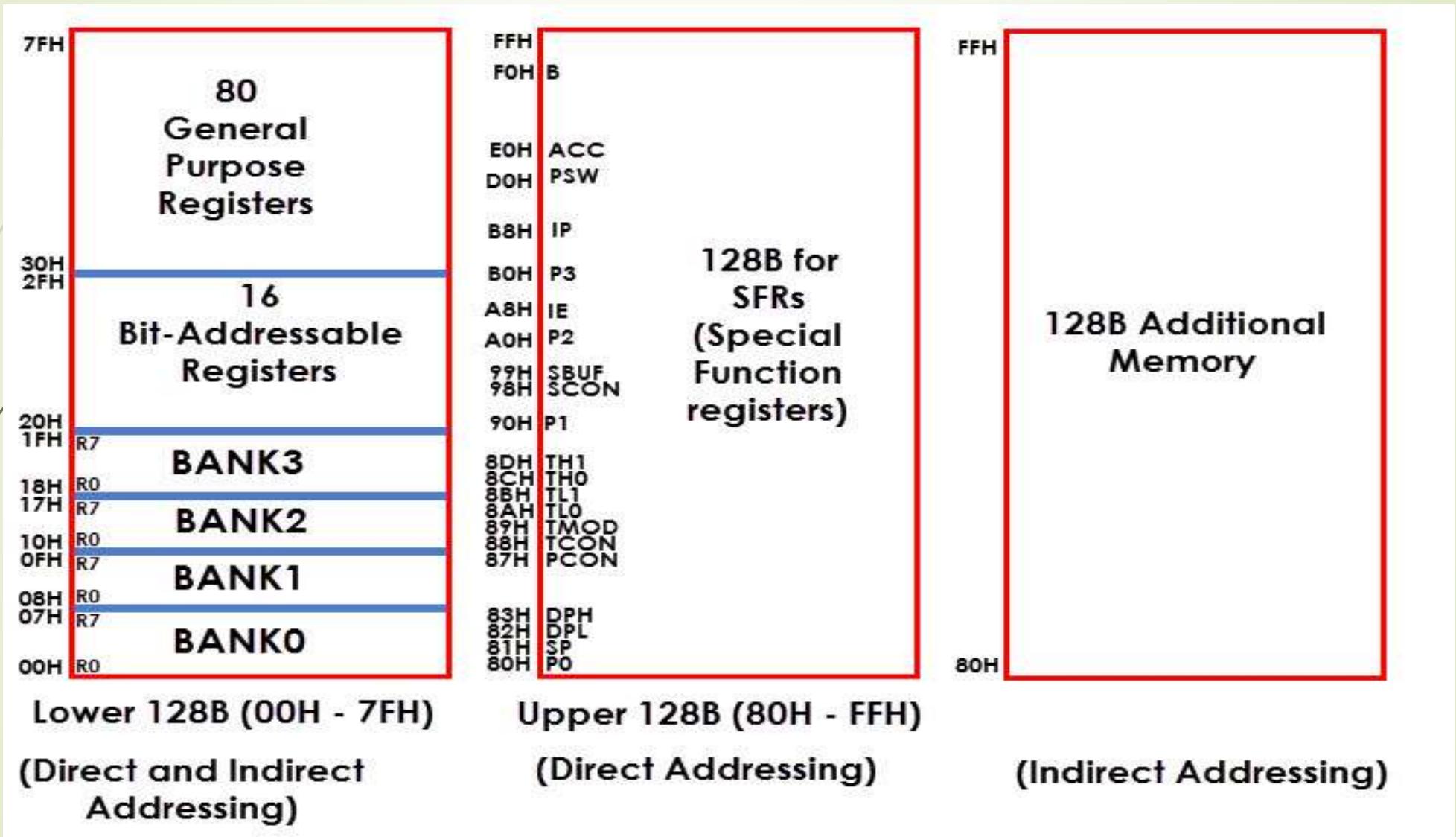


On-chip Memory Internal RAM

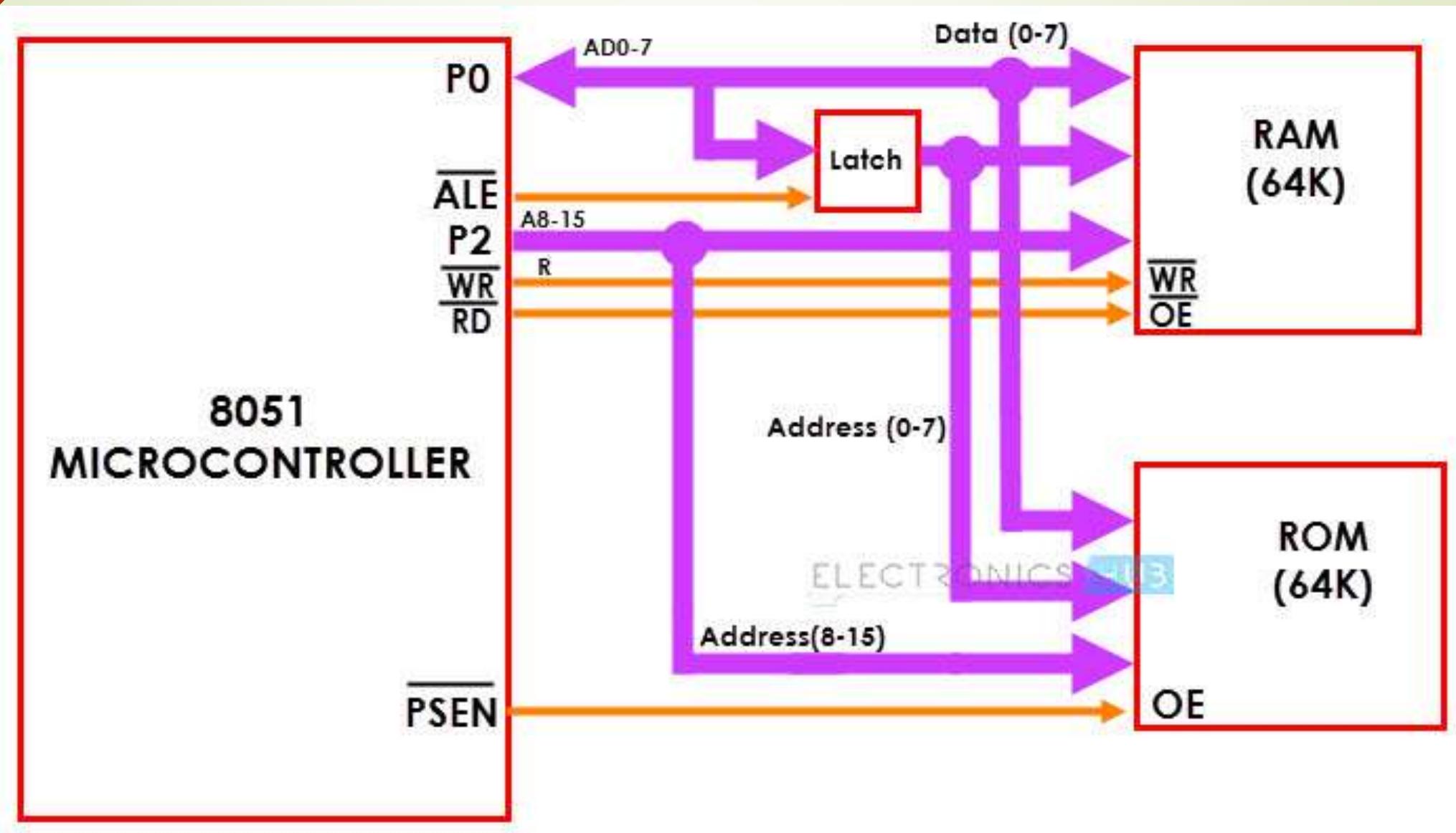
- ✓ The internal RAM or Data Memory of the 8051 Microcontroller is divided into:
 - General Purpose Registers
 - Bit Addressable Registers
 - Register Banks
 - Special Function Registers or SFRs.



Internal RAM Structure

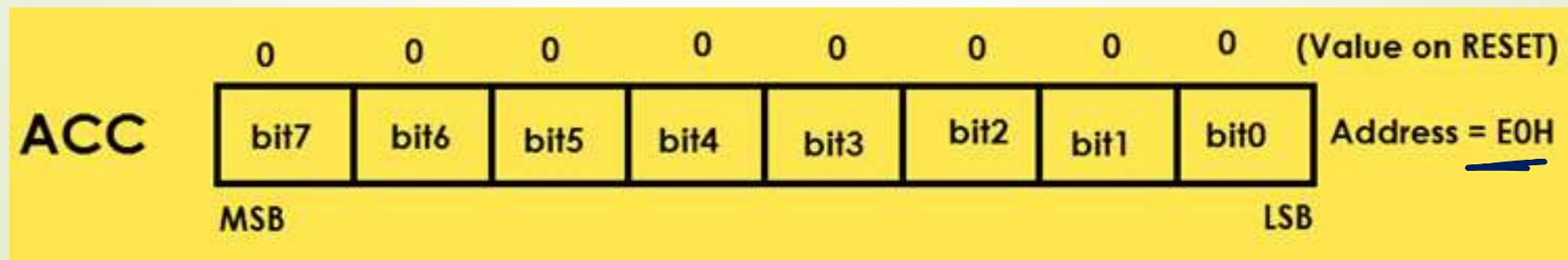


External Memory Access



A or Accumulator (ACC)

- ✓ The Accumulator or Register A is the most important and most used 8051 Microcontroller SFRs.
- ✓ The Register A is located at the address E0H in the SFR memory space.
- ✓ The Accumulator is used to hold the data for almost all the ALU Operations.
- ✓ Some of the operations where the Accumulator is used are:
- ✓ Arithmetic Operations like Addition, Subtraction, Multiplication etc.
- ✓ Logical Operations like AND, OR, NOT etc.
- ✓ Data Transfer Operations (between 8051 and External Memory)



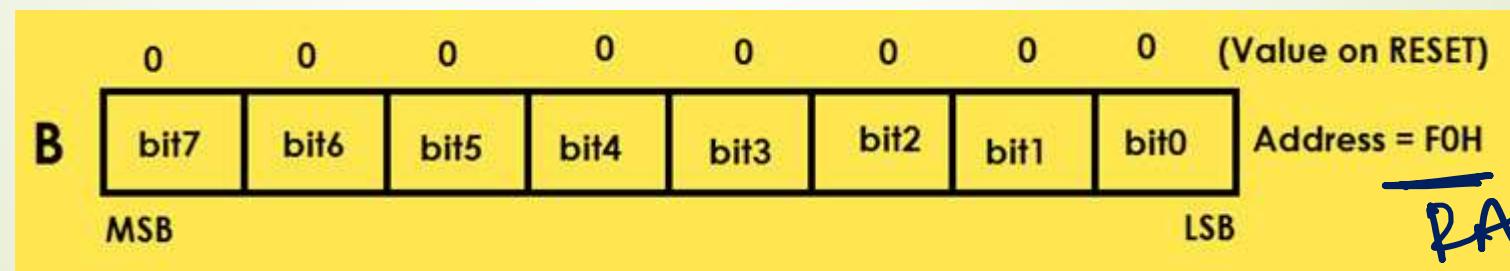
B (Register B)

40

The B Register is used along with the ACC in Multiplication and Division operations.

A
B

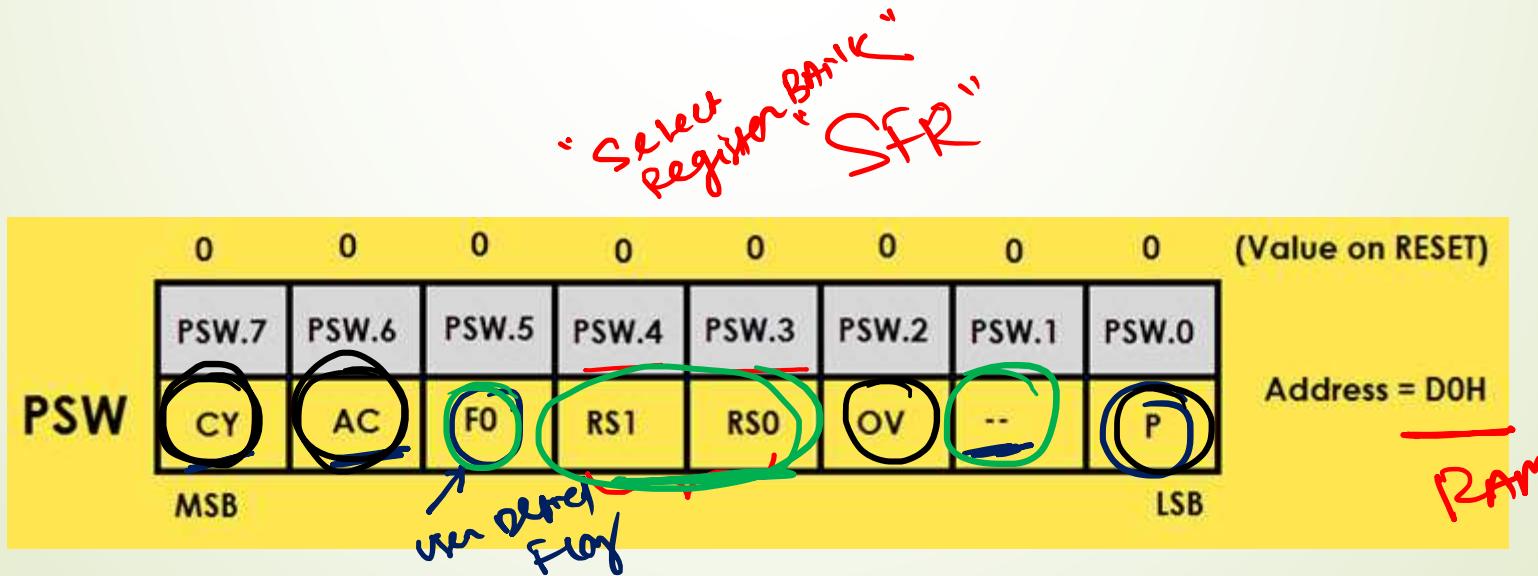
- These two operations are performed on ~~data~~ that are stored only in Registers A and B.
- During Multiplication Operation, one of the operand (multiplier or multiplicand) is stores in B Register and also the higher byte of the result.
- In case of Division Operation, the B Register holds the divisor and also the remainder of the result.
- It can also be used as a General Purpose Register for normal operations and is often used as an Auxiliary Register by Programmers to store temporary results.
- Register B is located at the address F0H of the SFR Address Space.



Program Status Word (PSW)

41

- ✓ The PSW or Program Status Word Register is also called as Flag Register and is one of the important SFRs.
- ✓ The PSW Register consists of Flag Bits, which help the programmer in checking the condition of the result and also make decisions.
- ✓ Flags are 1-bit storage elements that store and indicate the nature of the result that is generated by execution of certain instructions.



Program Status Word (PSW)

No. of 1's
will be All
Even Parity

P - Parity bit. If a number stored in the accumulator is even then this bit will be automatically set (1), otherwise it will be cleared (0). It is mainly used during data transmit and receive via serial communication.

"A"

$P = 1$

$P = 0$

$A \leftarrow 1011\ 0110$

Odd parity

- ✓ **Bit 1.** This bit is intended to be used in the future versions of microcontrollers.
- ✓ **OV Overflow** occurs when the result of an arithmetical operation is larger than 255 and cannot be stored in one register. Overflow condition causes the OV bit to be set (1). Otherwise, it will be cleared (0).
- ✓ **RS0, RS1 - Register bank select bits.** These two bits are used to select one of four register banks of RAM. By setting and clearing these bits, registers R0-R7 are stored in one of four banks of RAM.

8 Bits
1011 0110
Higher Nibble
Lower Nibble

F0 - Flag 0. This is a general-purpose bit available for use.

✓ **AC - Auxiliary Carry Flag** is used for BCD operations only.

$AC = 1$

✓ **CY - Carry Flag** is the (ninth) auxiliary bit used for all arithmetical operations and shift instructions.

$CY = 1$

$PSW.4$	$PSW.3$	SPACE IN RAM
<u>RS1</u>	<u>RS0</u>	
0	0	Bank0 [00h-07h]
0	1	Bank1 [08h-0Fh]
1	0	Bank2 [10h-17h]
1	1	Bank3 [18h-1Fh]

Q. Show the content of PSW after execution of the following instruction:

MOV A, #9CH

ADD A, #64H

AC
CY
OV
P

Soln:-

A \leftarrow 9CH

A \leftarrow A + 64H

$$\begin{array}{r} 10011100 \\ 01100100 \\ \hline 00000000 \end{array}$$

CF = 1
CY = 1
OF = 0

AC = 1

A
P = 0 (default value)

Ex:-

MOV A, #0FH

ADD A, #F1H

CF = 1, AC = 1
OF = 0, P = 0

$$\begin{array}{r} 00001111 \\ 11110001 \\ \hline 1000000000 \end{array}$$

1 \leftarrow even no's or 1's
0 \leftarrow odd no's of 1's

Stack Pointer (SP)

44

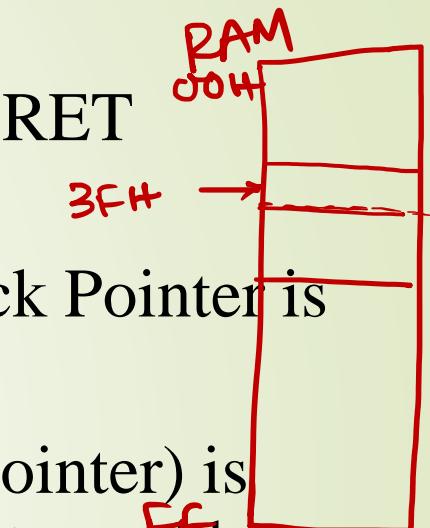
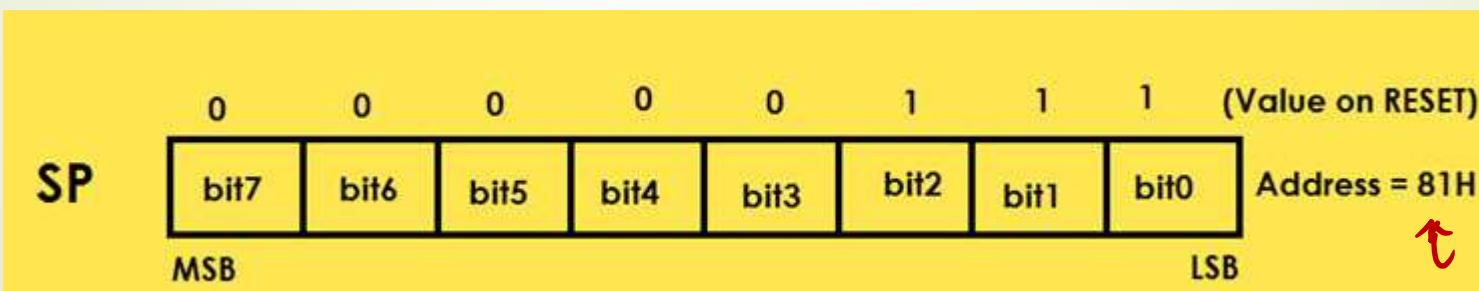
- ✓ SP or Stack Pointer points out to the top of the Stack and it indicates the next data to be accessed.

~~Stack Pointer can be accessed using PUSH, POP, CALL and RET Instructions.~~

- ✓ The Stack Pointer is an 8-bit register and upon reset, the Stack Pointer is initialized with 07H

When writing a new data byte into the stack, the SP (Stack Pointer) is automatically incremented by 1 and the new data is written at an address SP+1.

When reading data from stack, the data is retrieved from the Address in SP and after that the SP is decremented by 1 (SP-1).



~~"8086"~~
PUSH ← 16 bit
"SP" PUSH AX
SP ← SP - 2

45

PUSH operation in 8051

$\text{PUSH} \rightarrow \checkmark 25H, 12H \& 0F3H$

$\checkmark SP \leftarrow 3FH$

Register BANK

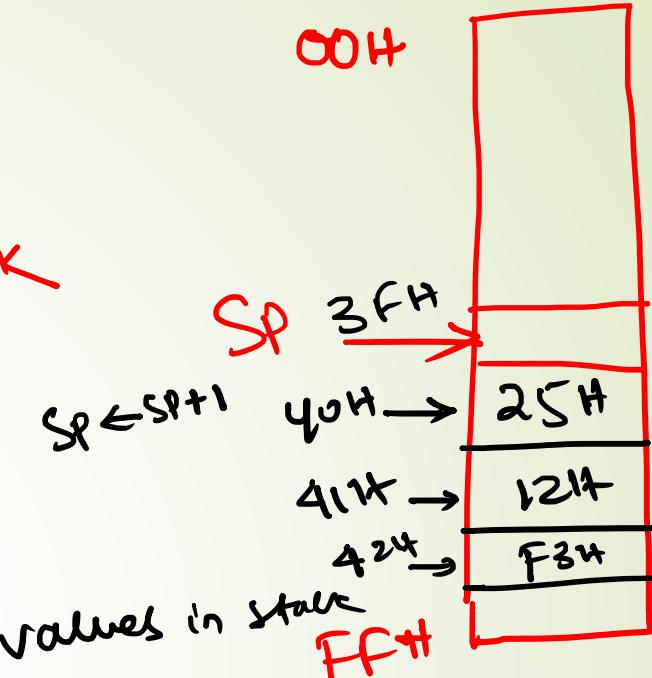
→ Initialize SP

→ Select R. Bank

→ Store the values to be pushed in the register R7

→ PUSH command to push those values in stack

SP_{ini}:



MOV SP, #3FH ; Initialization.

CLR PSW.3 ; BANK 0 is selected
CLR PSW.4

MOV R0, #25H

MOV R2, #12H

MOV R6, #0F3H

SP _{new} ← 42H

SP_{new} ← SP + 3

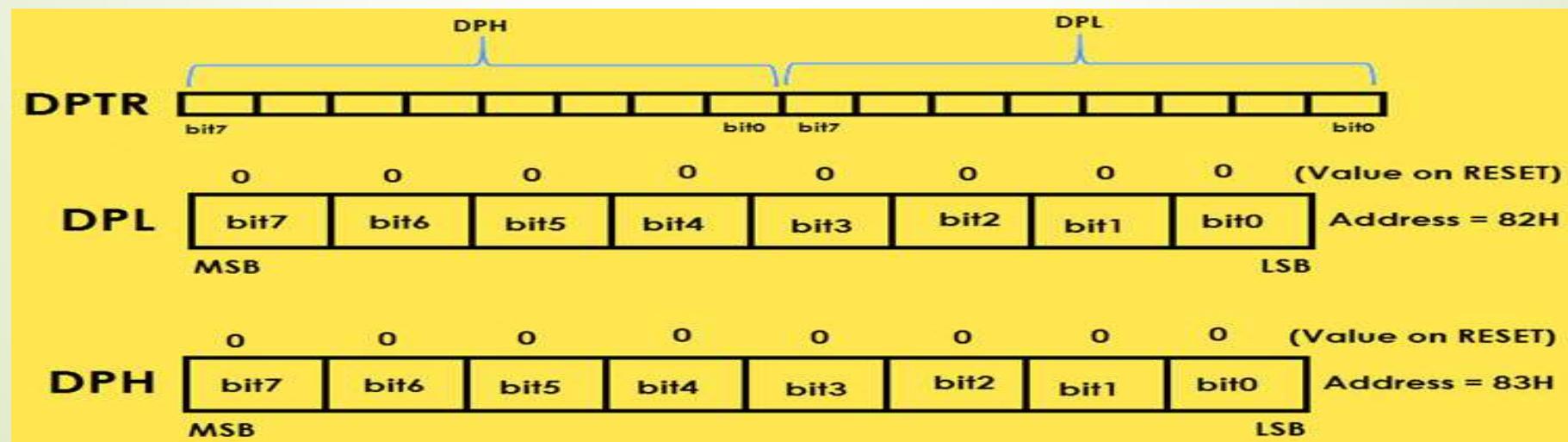
PUSH 0^-
PUSH 2^-
PUSH 6^-

POP 6
POP 2
POP 0

Data Pointer (DPTR – DPL and DPH)

46

- ✓ The Data Pointer is a 16-bit Register and is physically the combination of DPL (Data Pointer Low) and DPH (Data Pointer High) SFRs.
- ✓ The Data Pointer can be used as a single 16-bit register (as DPTR) or two 8-bit registers (as DPL and DPH).
- ✓ DPTR doesn't have a physical Memory Address but the DPL (Lower Byte of DPTR) and DPH (Higher Byte of DPTR) have separate addresses in the SFR Memory Space. DPL = 82H and DPH = 83H.
- ✓ The DPTR Register is used by the programmer addressing external memory (Program – ROM or Data – RAM).

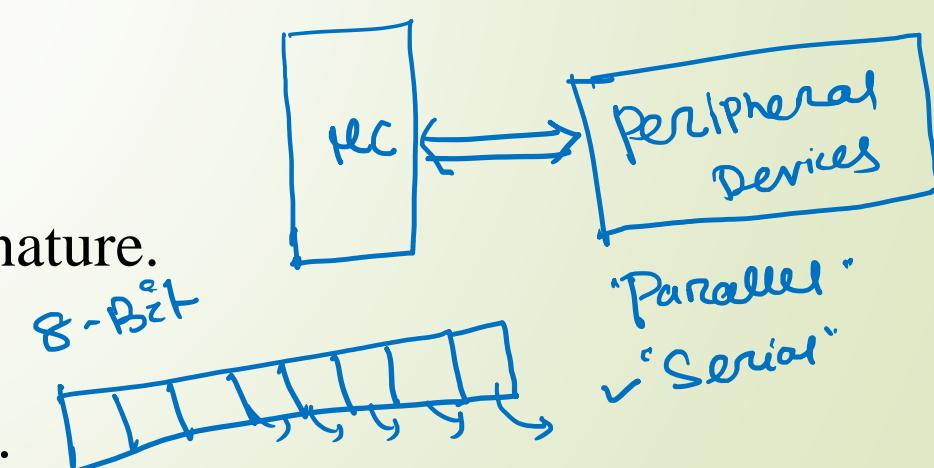
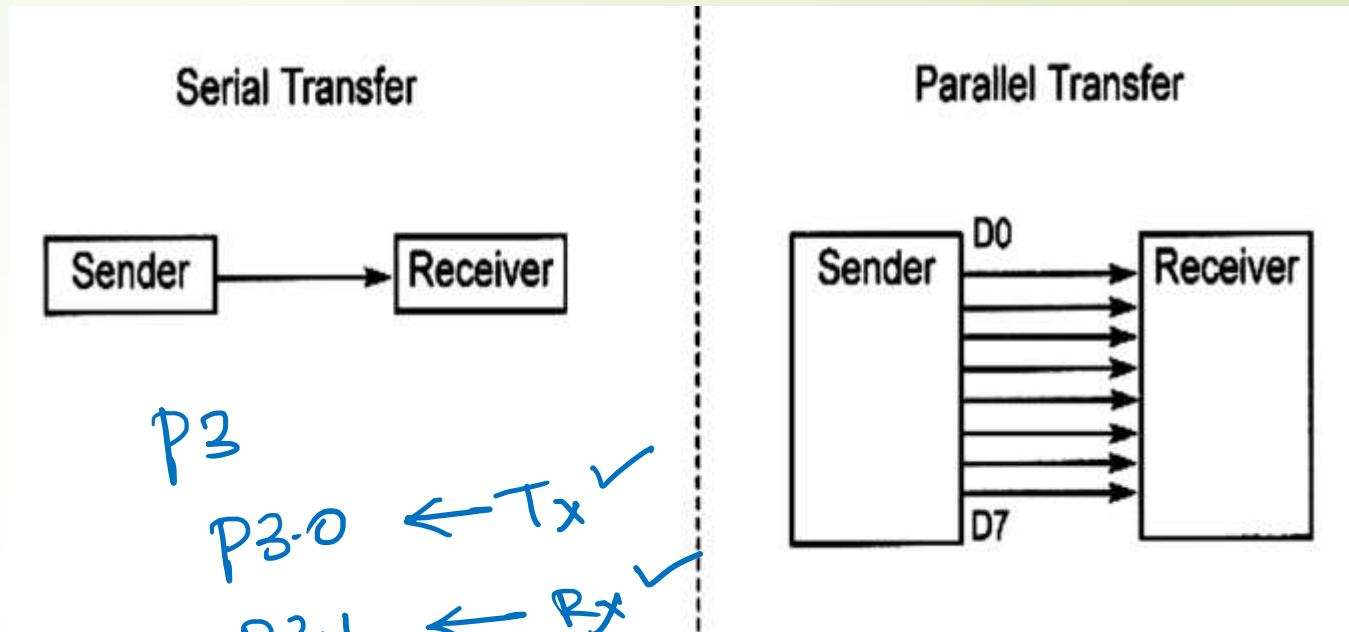


Serial Communications

47

Advantages of Serial Communications

- ✓ To reduce the cost of cable (long distance).
- ✓ Minimize the effect of noise (cross-talk).
- ✓ Availability of suitable communication media.
- ✓ Many devices have inherent serial communication in nature.
- ✓ Small connector.
- ✓ Clock skew between different channels is not an issue.



Types of Serial communications

~~Synchronous Simplex~~

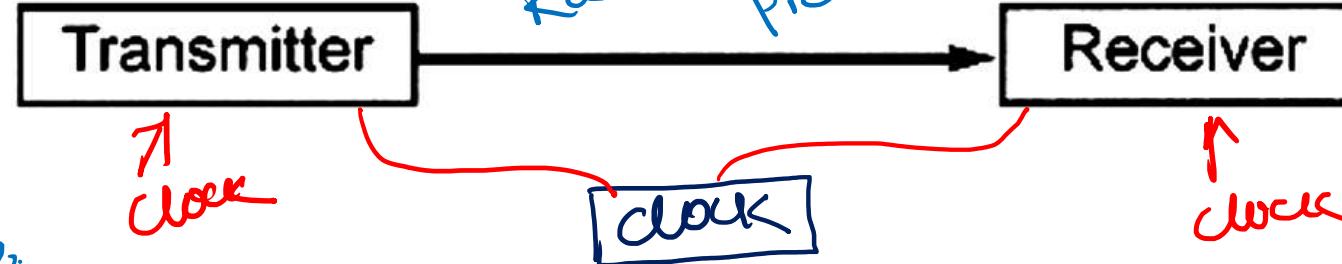
Half Duplex

Full Duplex

one crystal
one JART

8052
2 JART

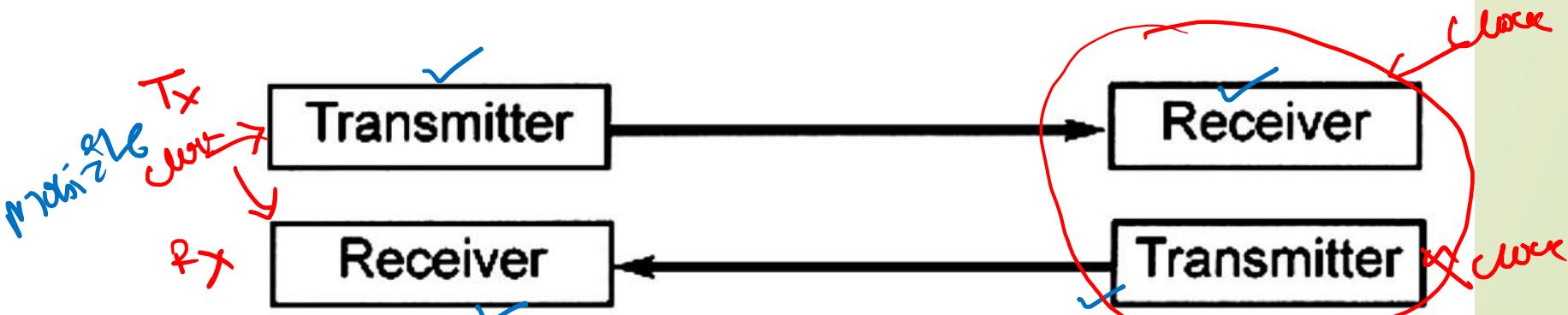
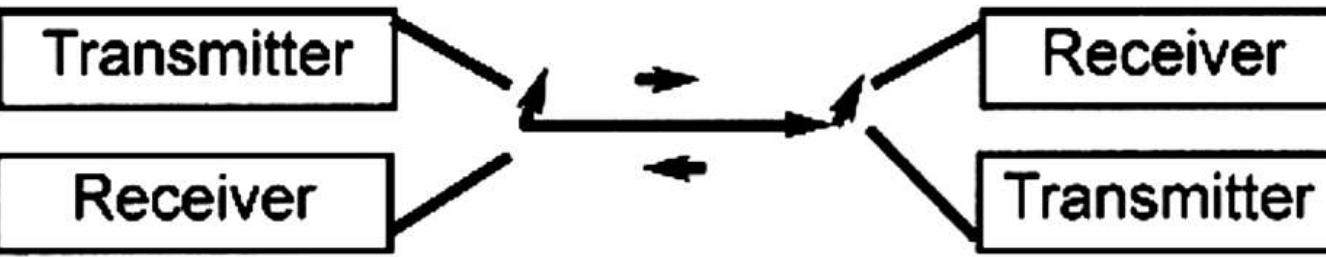
Walkie Talkie



8051

↓
89C51
89S51

89C52
89S52

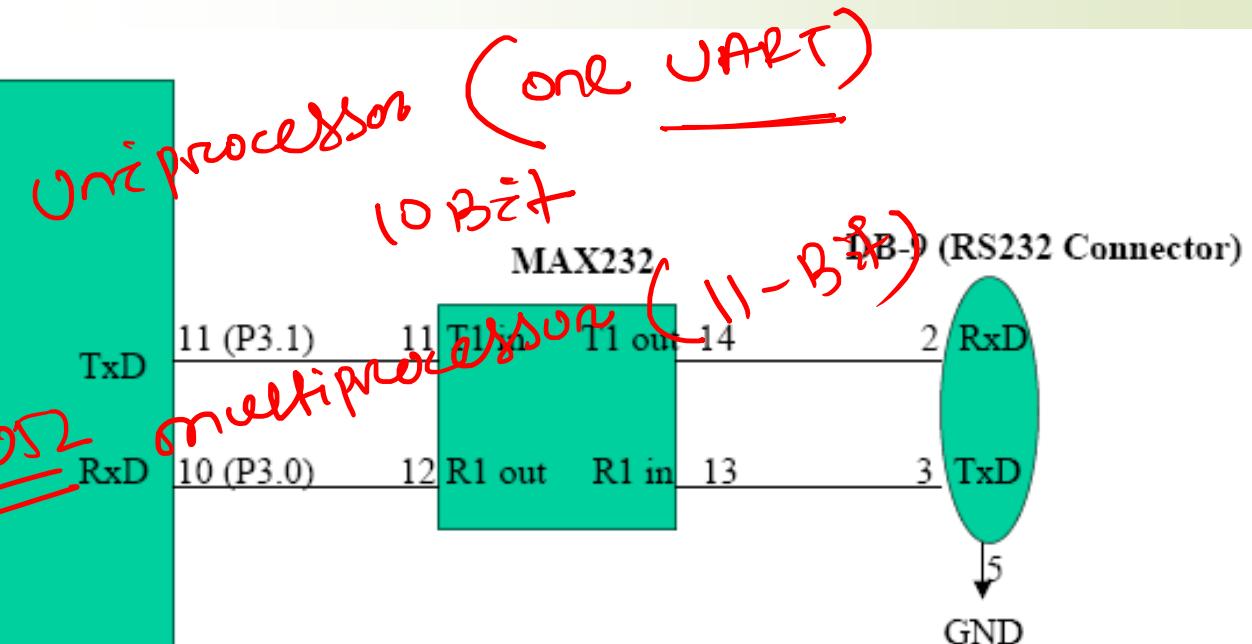
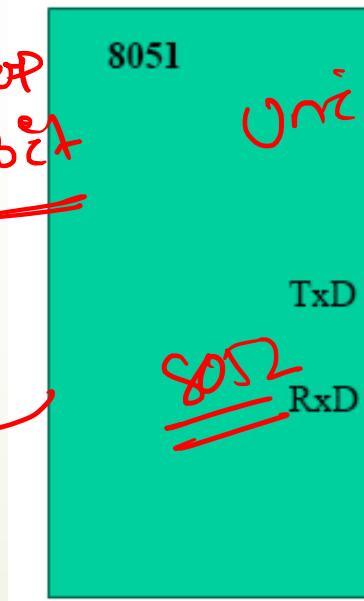
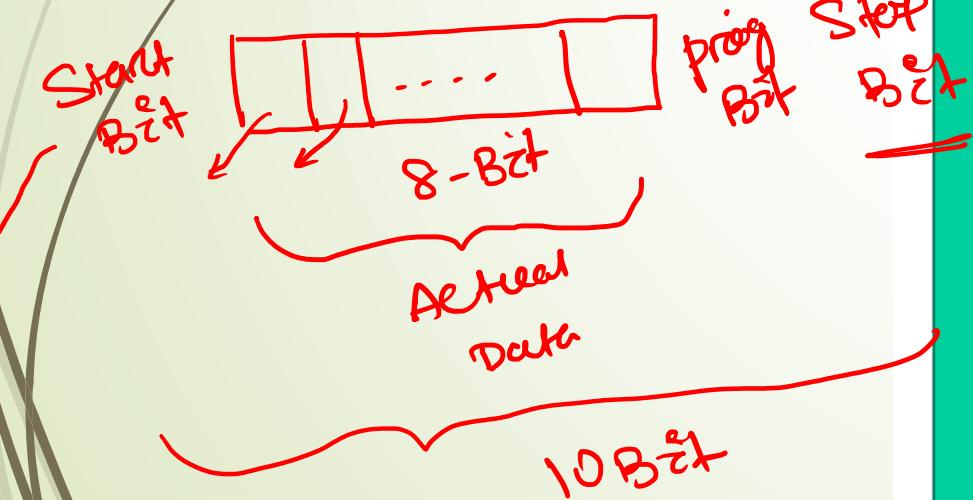


RxD and TxD pins in the 8051

49

- ✓ The 8051 has two pins for transferring and receiving data by serial communication. These two pins are part of the Port3(P3.0 & P3.1)
- ✓ These pins are TTL compatible and hence they require a line driver to make them RS232 compatible
- ✓ Max232 chip is one such line driver in use.
- ✓ Serial communication is controlled by an 8-bit register called SCON register, it is a bit addressable register.

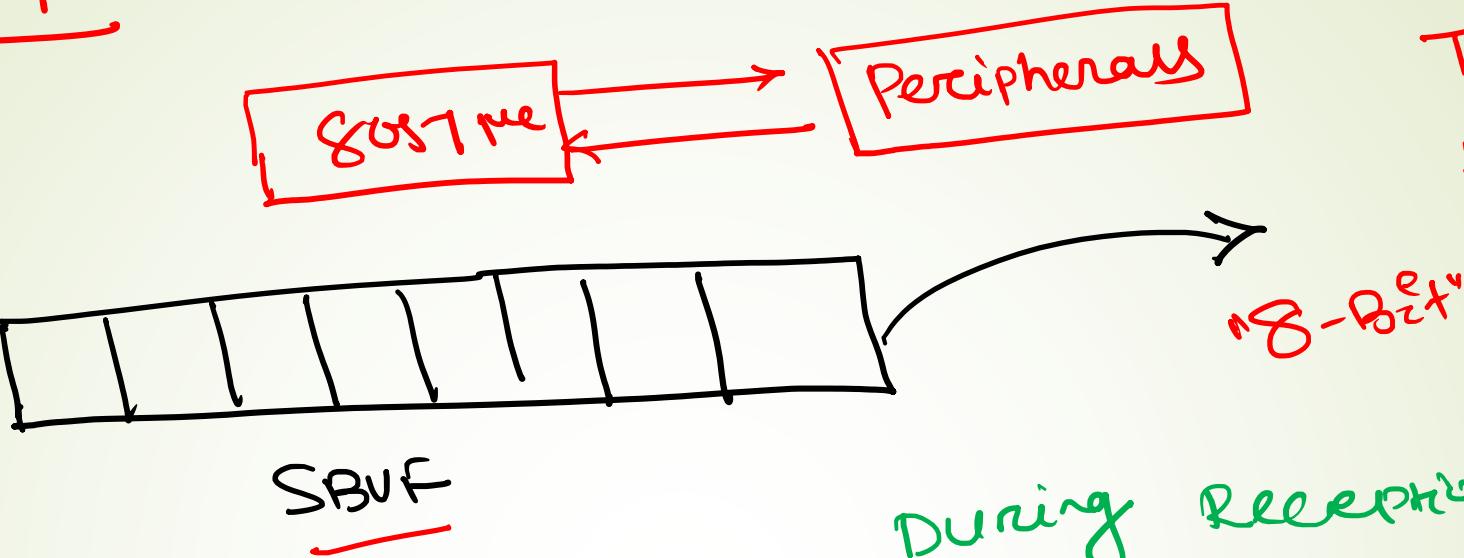
SBUF
SCON
PCON



Number of bits send/received per second
BAUD RATE

SBUF :- Serial Buffer

Data has to be stored in the SBUF Register



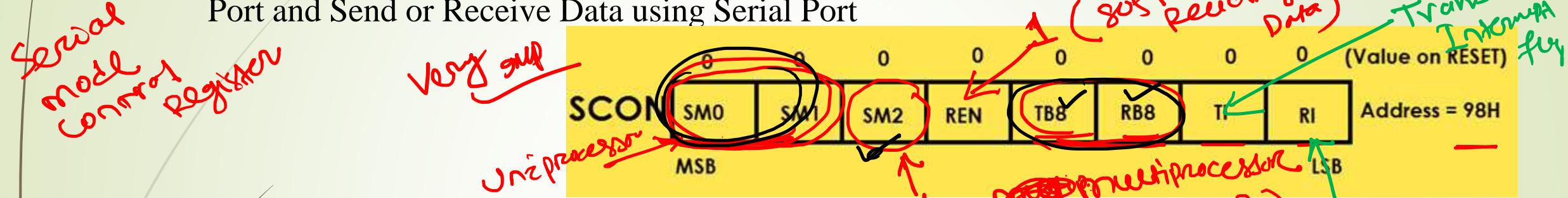
Tx ← Transmit
Rx ← Receive

"SOSTR supports 8-bit operation"
"SBUF = 8-bit"

During Reception of Data
Data will be coming from some peripheral device (serial communicator)
(Data) \rightarrow will be stored/received by SBUF in SOSTR

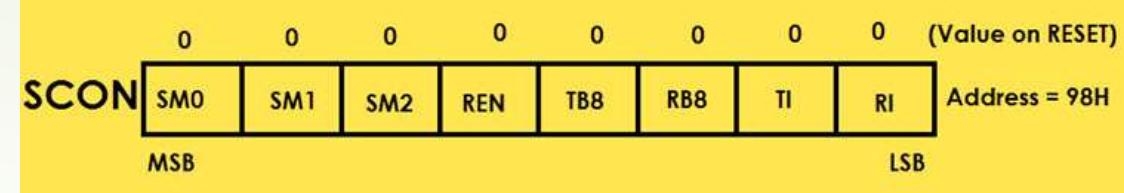
Peripheral Control Registers: SCON (Serial Control)

- ✓ The Serial Control or SCON SFR is used to control the 8051 Microcontroller's Serial Port. It is located as an address of 98H.
- ✓ Using SCON, we can control the Operation Modes of the Serial Port, Baud Rate of the Serial Port and Send or Receive Data using Serial Port

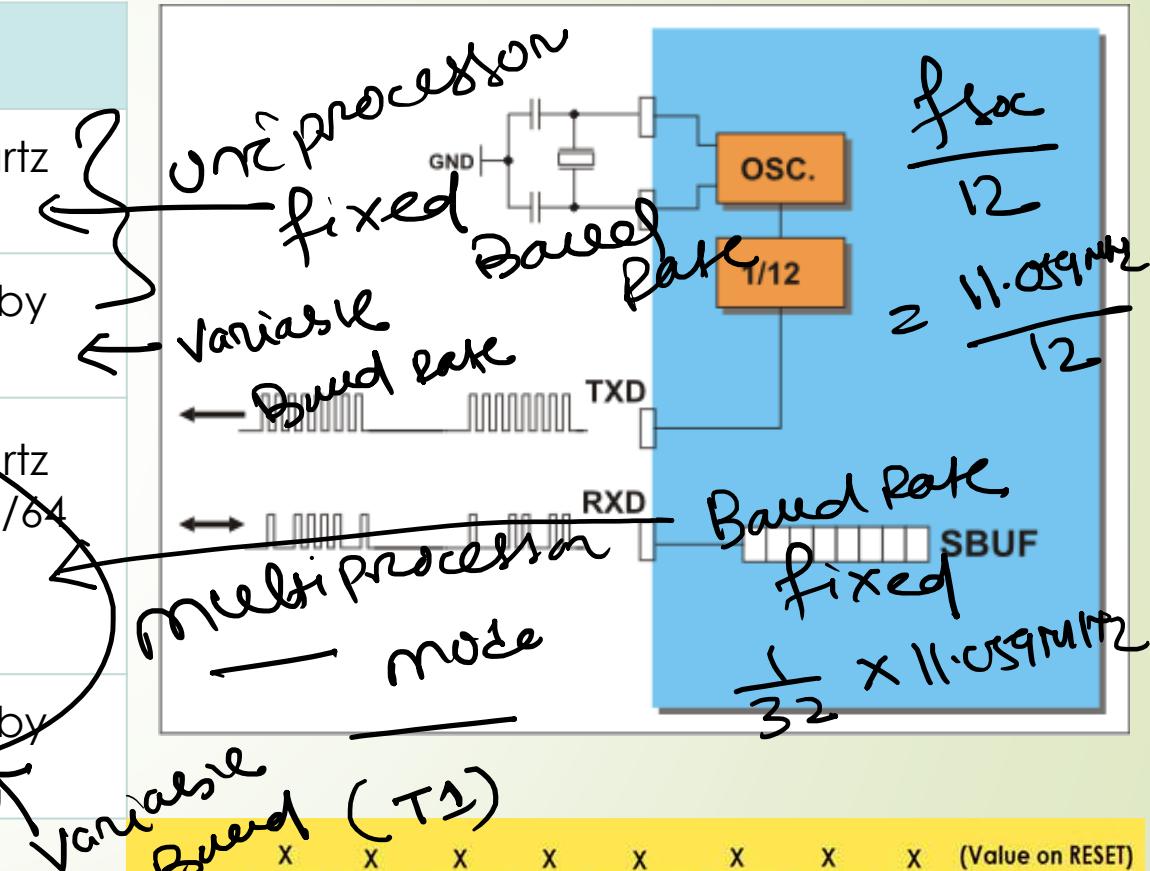


- ✓ **SM0** - Serial port mode bit 0 is used for serial port mode selection.
- ✓ **SM1** - Serial port mode bit 1.
- ✓ **SM2** - Serial port mode 2 bit, also known as multiprocessor communication enable bit. When set, it enables multiprocessor communication in mode 2 and 3, and eventually mode 1. It should be cleared in mode 0.
- ✓ **REN** - Reception Enable bit enables serial reception when set. When cleared, serial reception is disabled.
- ✓ **TB8** - Transmitter bit 8. Since all registers are 8-bit wide, this bit solves the problem of transmitting the 9th bit in modes 2 and 3. It is set to transmit a logic 1 in the 9th bit.
- ✓ **RB8** - Receiver bit 8 or the 9th bit received in modes 2 and 3. Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1.
- ✓ **TI** - Transmit Interrupt flag is automatically set at the moment the last bit of one byte is sent. It's a signal to the processor that the line is available for a new byte transmit. It must be cleared from within the software.
- ✓ **RI** - Receive Interrupt flag is automatically set upon one byte receive. It signals that byte is received and should be read quickly prior to being replaced by a new data. This bit is also cleared from within the software.

Peripheral Control Registers: SCON (Serial Control)

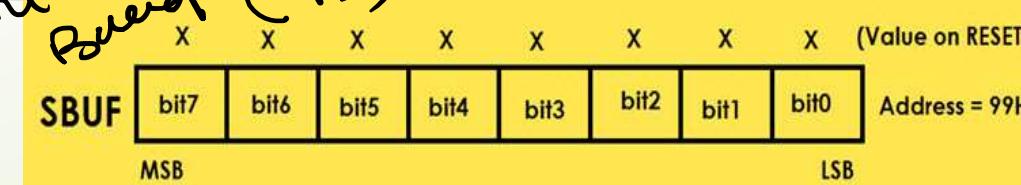


SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	8-bit Shift Register	✓ 1/12 the quartz frequency
0	1	1	8-bit UART	Determined by the <u>timer 1</u>
1	0	2	9-bit UART	1/32 the quartz frequency (1/64 the quartz frequency)
1	1	3	9-bit UART	Determined by the <u>timer 1</u>



Peripheral Data Registers: SBUF (Serial Data Buffer)

The Serial Buffer or SBUF register is used to hold the serial data while transmission or reception.



Peripheral Control Registers: PCON (Power Control)

- ✓ The PCON or Power Control register is used to control the 8051 Microcontroller's Power Modes and is located at 87H of the SFR Memory Space.
- ✓ Using two bits in the PCON Register, the microcontroller can be set to Idle Mode and Power Down Mode.
 - ✓ During Idle Mode, the Microcontroller will stop the Clock Signal to the ALU (CPU) but it is given to other peripherals like Timer, Serial, Interrupts, etc. In order to terminate the Idle Mode, you have to use an Interrupt or Hardware Reset.
 - ✓ In the Power Down Mode, the oscillator will be stopped and the power will be reduced to 2V. To terminate the Power Down Mode, you have to use the Hardware Reset.
- ✓ The PCON Register can also be used control the Baud Rate of the Serial Port using SMOD Bit.
- ✓ There are two general purpose Flag Bits in the PCON Register, which can be used by the programmer during execution.



Peripheral Control Registers: PCON (Power Control)

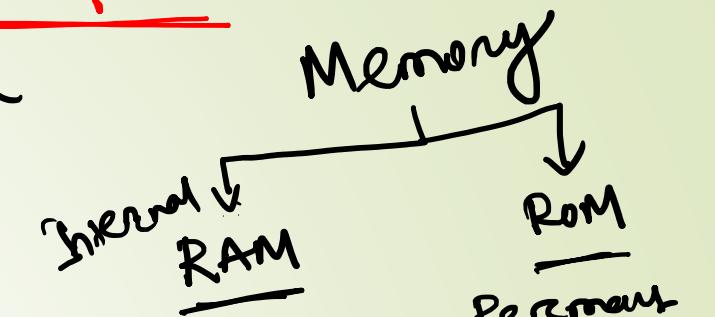
PCON	0	X	X	X	0	0	0	0	Value after Reset
	SMOD				GF1	GF0	PD	IDL	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

The purpose of the Register PCON bits is:

- ✓ SMOD Baud rate is twice as much higher by setting this bit.
- ✓ GF1 General-purpose bit (available for use).
- ✓ GF1 General-purpose bit (available for use).
- ✓ GF0 General-purpose bit (available for use).
- ✓ PD By setting this bit the microcontroller enters the Power Down mode.
- ✓ IDL By setting this bit the microcontroller enters the Idle mode.

ADDRESSing Modes of COST

1. Immediate Addressing Mode
2. Register Addressing mode
3. Direct Addressing mode
4. Register Addressing mode
5. Indirect Addressing mode

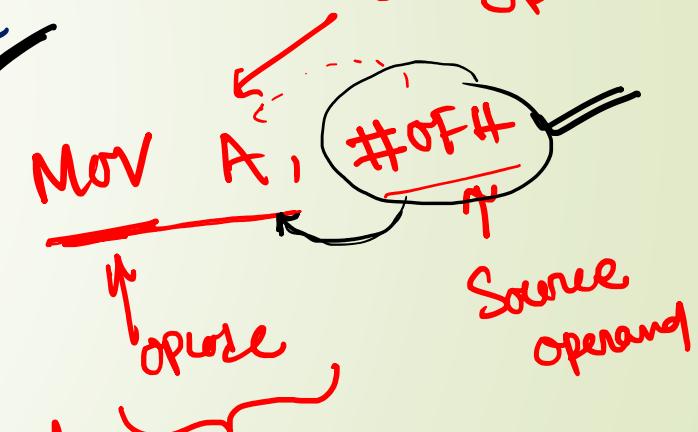
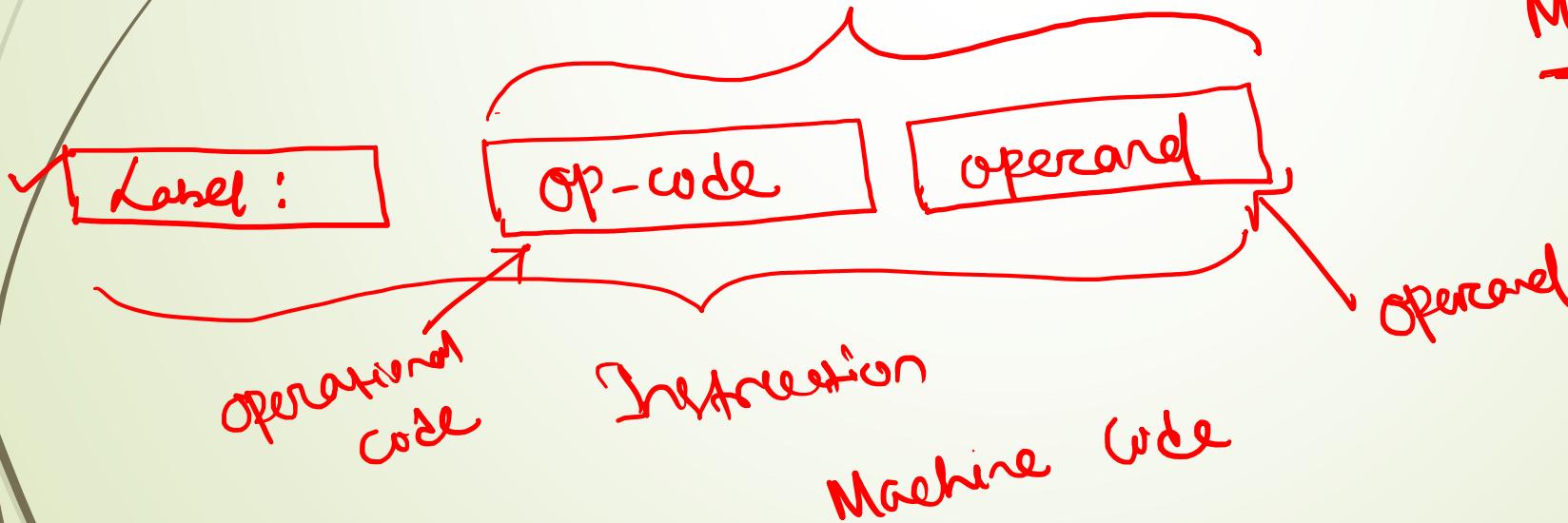


Permanent Data from ROM

External Data from RAM

Destination operand

Source operand



Mov A, R0 ;

Machine code

Programming in 8086

①

MOV AL, 05H ✓

MOV BX, 0F34H

Immediate A.M.

②

MOV AL, [4000H] ✓

Direct A.M.

Memory location
(Add.)

③

MOV AL, 50H [BX] ✓

Base relative
A.M.

PA = 4000H x 10H
+ 50H Data Segment

Programming in 8051

①

MOV A, #05H ✓

value Immediate A.M.

②

MOV A, 40H ✓

Direct
A.M.

00H
to
FFH RAM

Address of
internal RAM

③

MOV A, @RD ✓

register
Indirect
A.M.

Content index
the register Address (RAM)
as an

1. Immediate A.M.

Data/value is immediately given in the instruction.

Immediate
A.M.

~~MOV A, #05H~~ ← Hex

MOV A, #05 ← Decimal

MOV A, #01010101 ← Binary

MOV DPTR, #0512H ✓ ← Data pointer

MOV DPL, #12H ✓

MOV DPH, #05H ✓

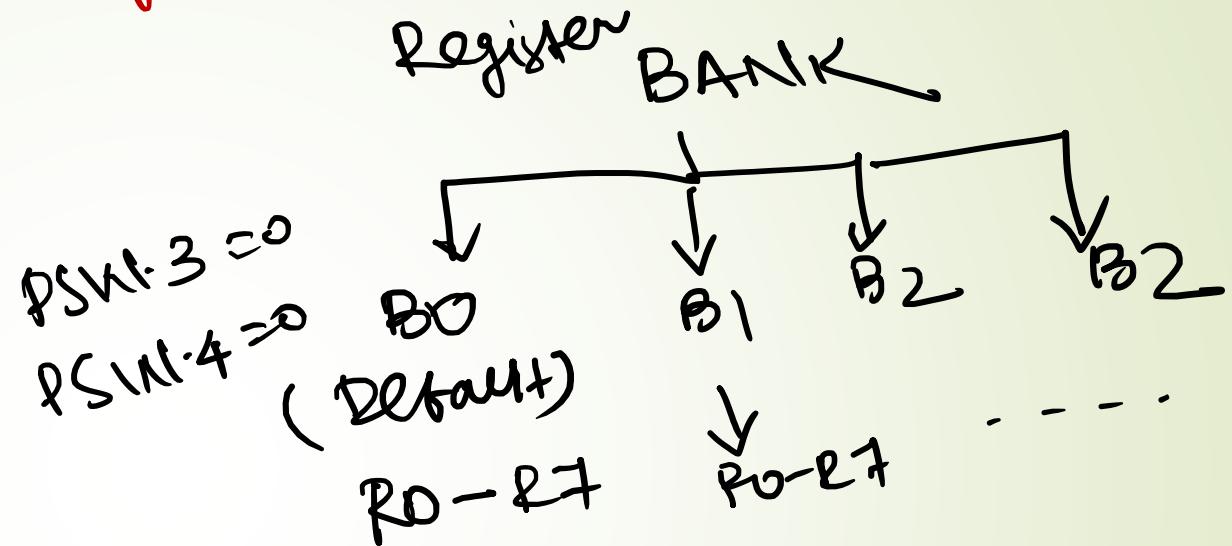
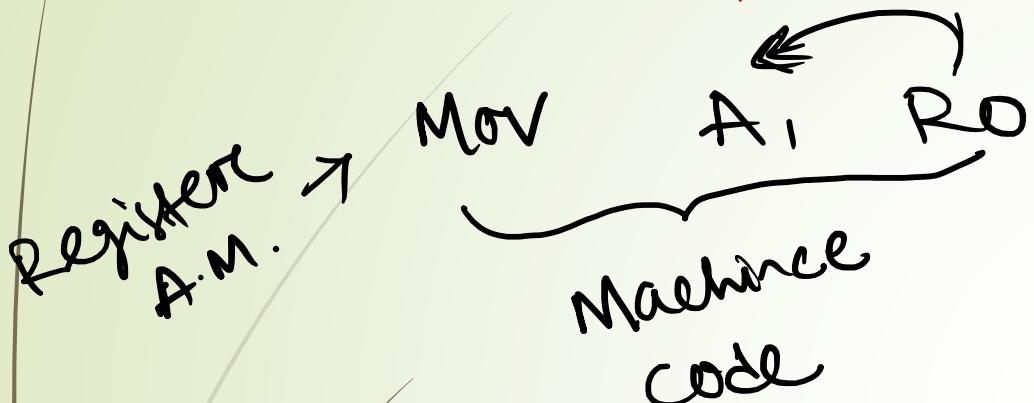
MOV P1, #05H

$$\begin{aligned} & \text{16 Bit} \\ & \text{ROM} \leftarrow 2^{16} \\ & = 64 \text{KB} \end{aligned}$$

~~MOV A, #25H~~ ✓
MOV A, #25H
Immediate A.M.
Same
SIP
25H
25H
25H
25H
25H
25H
25H

2. Register A.M.

Data/ value's given through a register



~~MOV DPTR, A~~ ← Register A.M.

DPTR, A

↑
16-Bit

↑
N

↓
8-Bit

~~MOV R0, R7~~ ← RAM

R0

R7

RAM

MOV B, R0

MOV A, R1

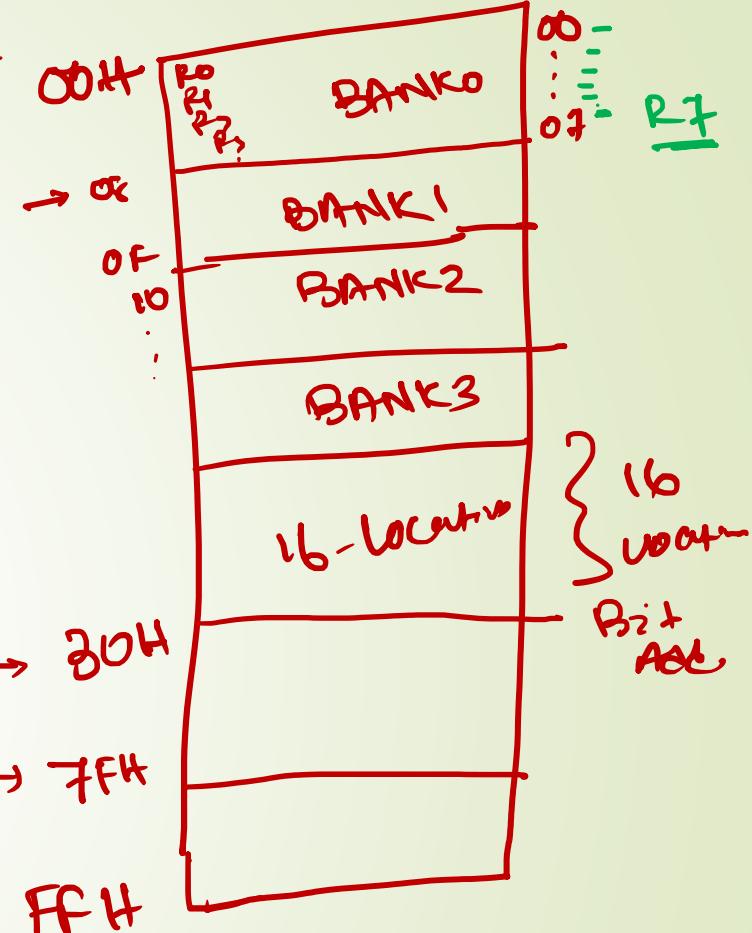
3. Direct Addressing Mode :-

Here the operand is the memory location ; that holds the data.

~~MOV A, 40H~~ ← Address location
~~MOV A, 41H~~
~~MOV 56H, A~~
~~ADD A, 56H~~

~~MOV A, 7~~ ← 7th location in RAM
 Copy the content into A
~~MOV A, F7~~

~~MOV 712~~
~~MOV R7, R2 ; BANK2~~



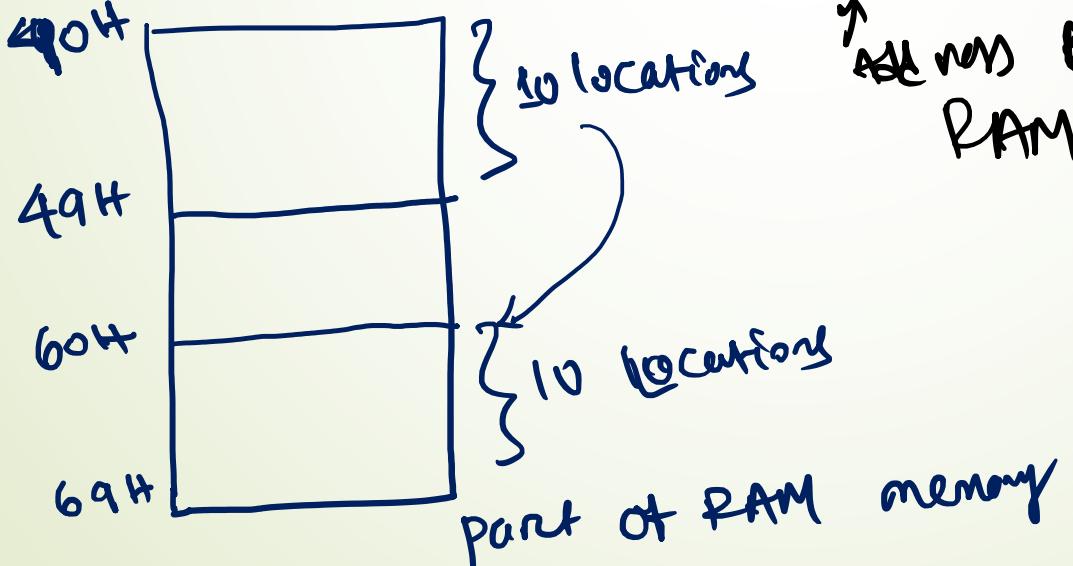
4. Register to direct A.M.

A register in this mode is used to point to the memory location to access the data stored in it.

MOV R0, 40H

MOV A, @R0; copy the content held by the
register as the memory loc
→ to the A

Draw



Copy the content inside
the locations 40H to 49H
→ to the locations
60H to 69H

Prog

```

MOV A, 40H
MOV 60H, A
MOV A, 41H
MOV 61H, A
MOV A, 42H
MOV 62H, A
.
.
```

Repeat this for
10 Hns

~~Segm~~
~~base~~
segm &
segment indirect
A.M.

Direct A.M.
Increase
the
length
of
word

Prog:-

Mov R0, #40H ; Initialization (Source)
 Mov R1, #60H ; Destination word initialization
 Mov R2, 10 ; Counter

Level
 → HERE:
 ↓

Loop
 & Register
 indirect A.M.

Mov A, C_{R0}
 Mov C_{R1}, A
 INC R0 ; 10 times
 INC R1
 DJNZ R2, HERE ; R2 = 10
 ↓
 Decrement it
 Jump if Not zero

10 Bytes of data
 from 40H to 49H
 into 60H to 69H

$$\begin{aligned}
 R2 &= 10 - 1 = 9 \\
 R2 &= 9 - 1 = 8
 \end{aligned}$$

Limitations of Register indirect A.M.

1. only R0 & R1 Registers can be used in this mode

@R0
@R1

2. Both R0 & R1 are of 8-Bit; total ~~256~~ locations can be accessed.

$2^8 = 256$ Locations can be accessed.

External RAM
External ROM

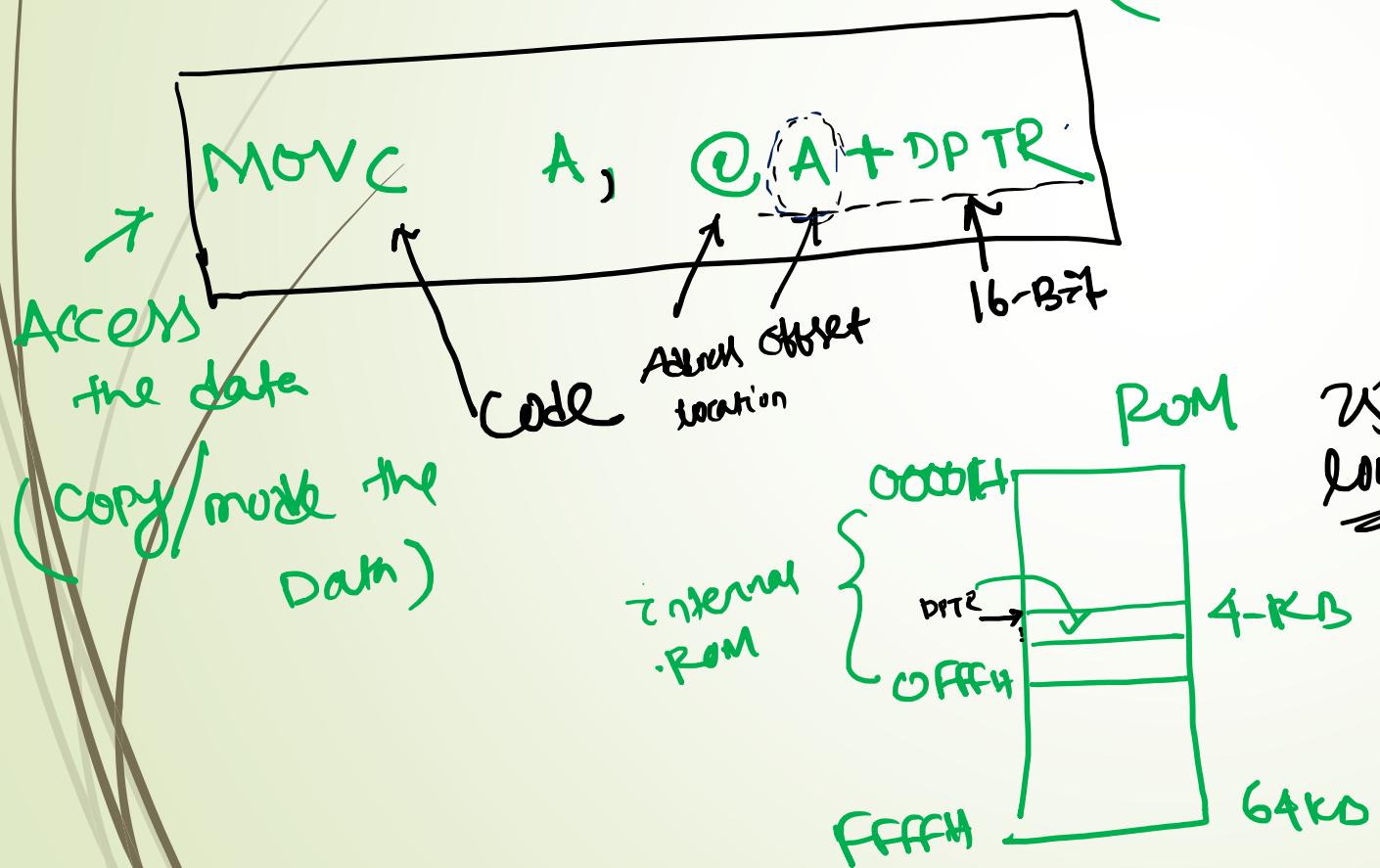
} Internal RAM

$$\uparrow \\ 1\text{-Bit} = 2^1 = 64\text{KB}$$

Internal RAM : 00 H to FF H
256 locations

5. Indexed Addressing Mode :-

It is used to access the data in the program memory space (ROM)



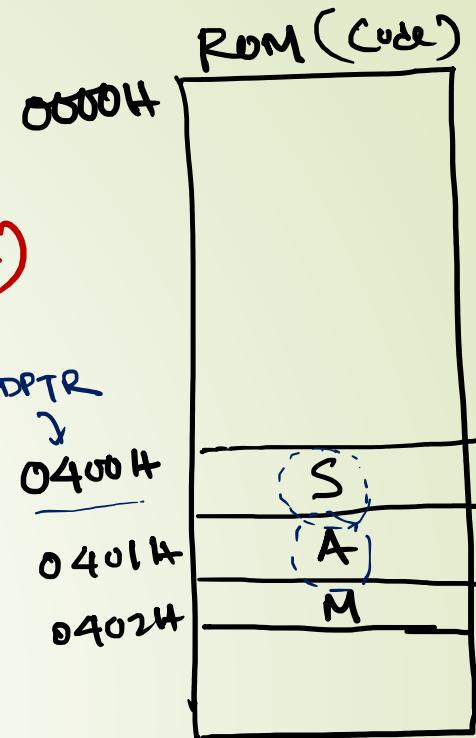
Sof!

Prog : ROM location : 0400H → "SAM" is

Burned in μC 89C51. Copy the Data

→ Locations Starting from 0400H (inside ROM)

→ to the RAM Locations Starting from 60H.



MAIN:

```

    MOV A, #00H
    MOV DPTR, #0400H
    MOVC A, @A+DPTR
    MOV 60H, A
    INC DPTR ; 0401H
    CLR A
  
```

Burn 'SAM' to
ROM

Org 0400H

DB "SAM"

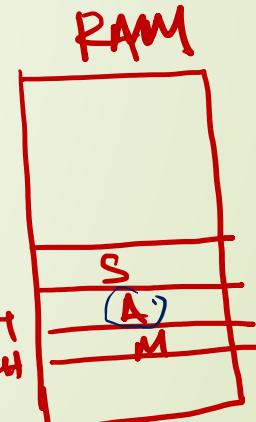
end

```

MOVC A, @A+DPTR
MOV 61H, A
INC DPTR
CLR A
MOVC A, @DPTR
    
```

00H

60H → 61H 62H



66

Prog. :-

Seven segment display (0 to 9). Store the look-up table value for 7-seg display into the ROM location starting from 0400H. Display these values in the Port 1 where the 7-seg display is connected.

0400	0	3F
0401	1	0b
0402	2	5B
0403	3	4F
0404	4	66
0405	5	6D

'ROM'

