

8051 MICROCONTROLLERS

BY

G HARISH BABU

ASST PROF

DEPT OF ECE

CVR ENGG college



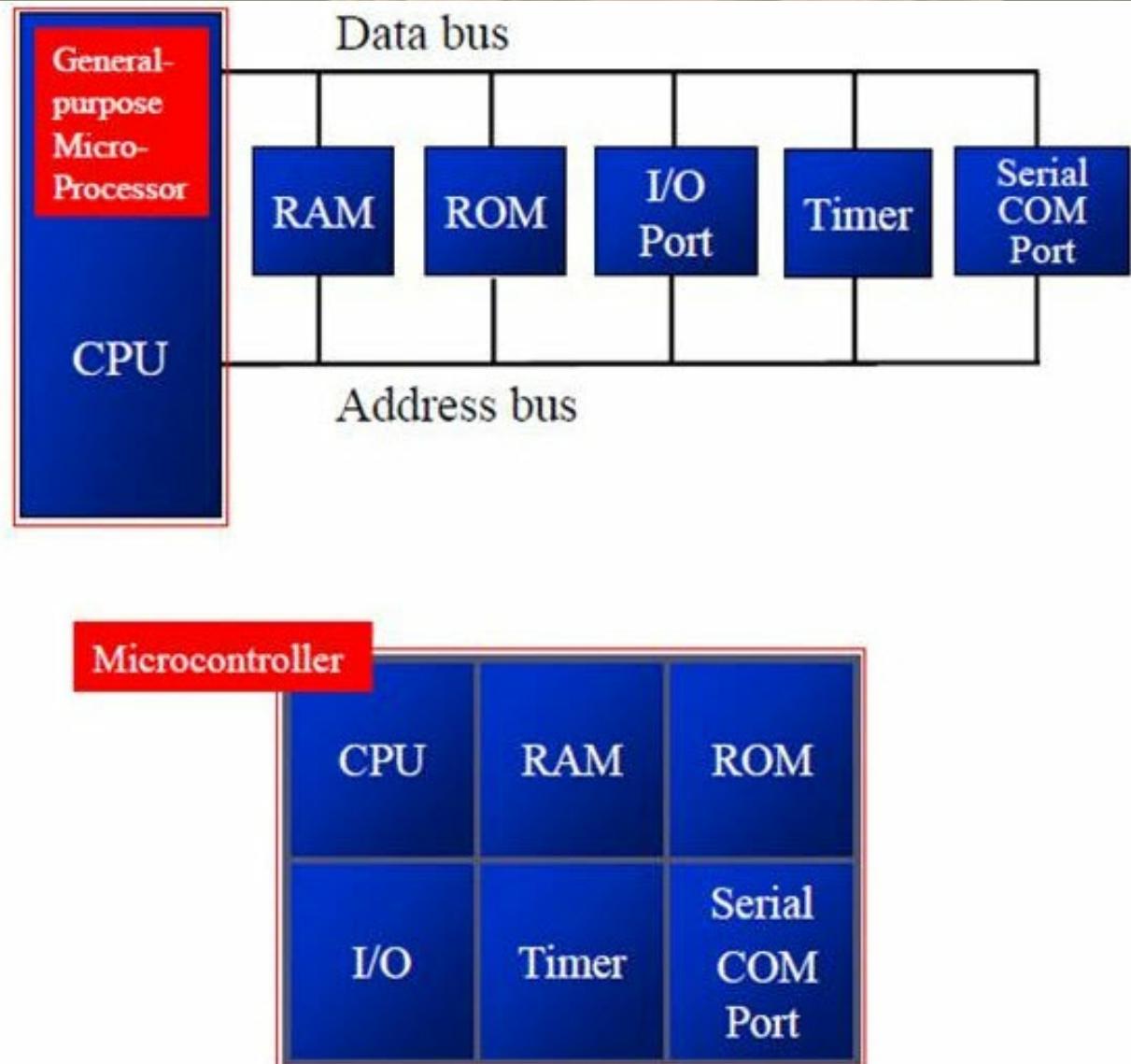
Style

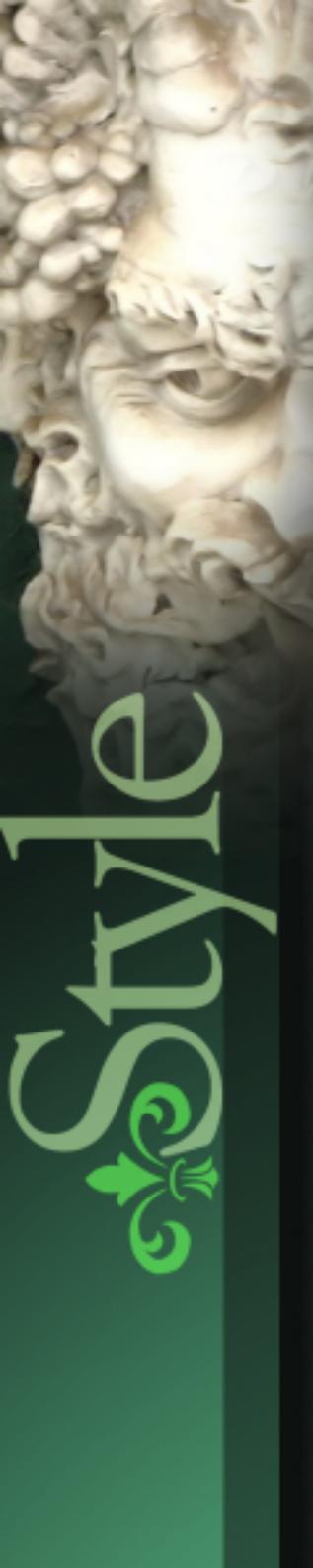


Style

MICRO- CONTROLLERS AND EMBEDDED PROCESSORS

Microcontroller
vs. General-
Purpose
Microprocessor
(cont')



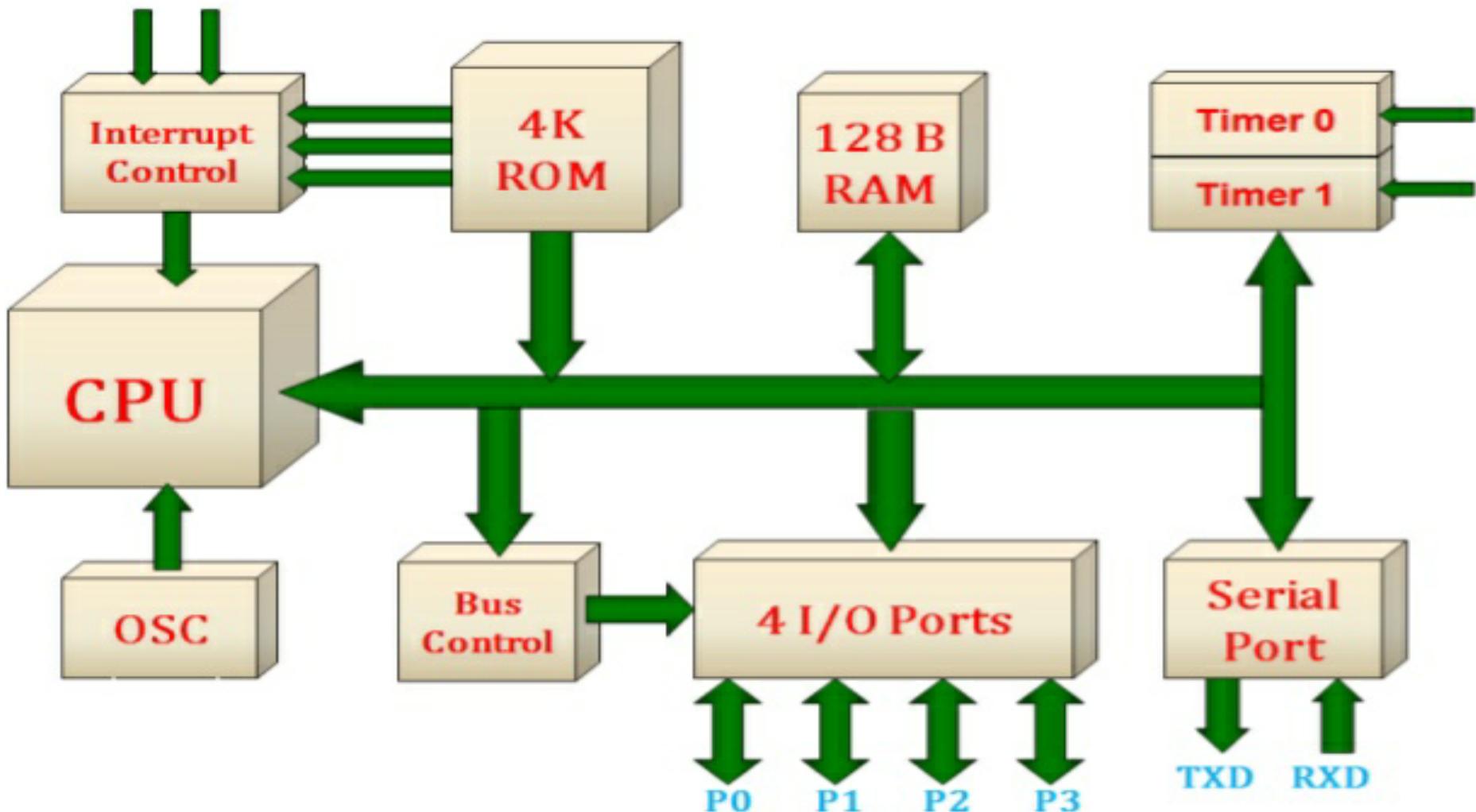


➤Features of the 8051 (MCS 51) family

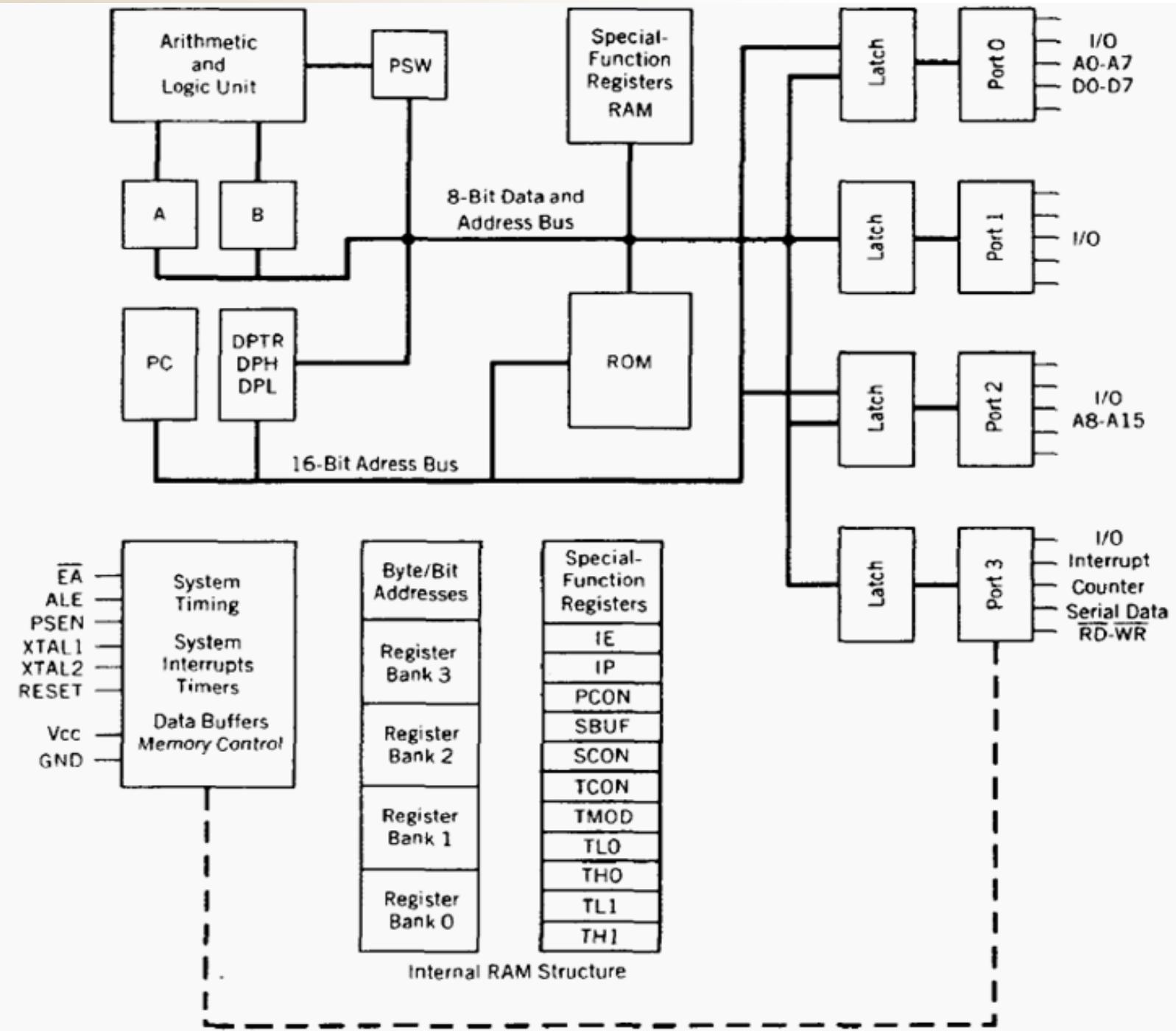
- The key features of 8051 microcontroller are:
- 8 bit CPU with Boolean processing capabilities.
- 4K bytes on-chip *program memory.
- 128 bytes on-chip data memory.
- 64 Kbytes each program and external data address space.
- 32 bidirectional I/O lines organized as four 8-bit I/O ports.
- serial port – Full duplex UART.
- 2 16-bit timers/counters.
- Two-level prioritized interrupt structure.
- Direct byte and bit addressability.
- Four register banks.
- Binary or decimal arithmetic support.
- Hardware multiply and divide operations.
- 12 clock cycles per machine cycle

Feature	8031	8051	8751	8032	8052	8752
Program memory	None ROM less	4K ROM	4K EPROM	None ROM less	8K ROM	8K EPROM
Data memory	128 Bytes	128 Bytes	128 RAM	256 Bytes	256 Bytes	256 Bytes
Timers/ Counters (16-bit)	2	2	2	3	3	3
I/O pins	32	32	32	32	32	32
Serial port	1	1	1	1	1	1
Interrupt sources (Reset not included)	5	5	5	6	6	6

General Block Diagram of 8051



Style





CENTRAL PROCESSING UNIT

- The CPU consists of 8 bit ALU with associated registers like A,B PSW,SP, 16 bit Program Counter (PC) & Data Pointer Register (DPTR).
- The 8051 ALU can perform Arithmetic & Logic functions on 8-bit variables. The arithmetic unit performs additions, subtractions, multiplications & division.
- The Logic unit can perform logical operations such as AND, OR & EX-OR, as well as rotate, clear & complement.

A Register (Accumulator) :

- Accumulator is a 8-bit register & is widely used for many operations like addition, subtraction, multiplication, division & Boolean bit manipulations.
- The A- Register is also used for all data transfers b/w the 8051 and any external memory.

B Register :

- The B-Register is used with the A-Register for multiplication & division operations & has no other function other than as a location where data may be stored.

STACK Pointer (8-bit) :

- The STACK refers to an area of internal RAM used by the CPU to store & retrieve (take back) data quickly.
- The register used to access the stack is called the Stack Pointer (SP) register.
- The Stack pointer register used by the 8051 to hold an internal RAM address that is called the Top of the Stack.
- When 8051 is RESET, the SP is set to 07h.
- The storing of a CPU register in the Stack back into the CPU register is called POP.



Data Pointer (DPTR) :

- D PTR is a 16-bit register, which holds a 16-bit address.
- D PTR can be splitted into two parts:
 1. DPH -> Data Pointer high byte having internal address 83h.
 2. DPL -> Data Pointer Low byte having internal address 82h.

{ The D PTR does not have a single internal address }

Program Counter (PC) :

- PC is a 16-bit register which holds the address of the next instruction to be executed. The PC is automatically incremented after every instruction byte is fetched.
- The 8051 has 16-bit PC hence it can address upto 64K-bytes.

{ PC is the only register that does not have an internal address }

I/P – O/P (I/O Ports) :

- The 8051 has 32 I/O pins configured as four 8-bit parallel ports.
i.e., Port0 , Port1, Port2, Port3.
- All four ports are bi-directional i.e., each pin can be configured as I/P or O/P under software control.

Timers & Counters :

- 8051 use two 16-bit registers namely T0 & T1 either for Timer or Counter.
- The two Timers or Counters are divided into Two 8-bit registers called Timer Low (TL0, TL1) and Timer High (TH0, TH1).

PSW:

Program status word is an 8 bit register. It is also referred as flag register or processor status word.

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0				
CY	AC	F0	RSI	RS0	OV	--	P				
Bit	Symbol	Flag name and description									
7	C (or CY)	Carry; Used in arithmetic, logic and Boolean operations									
6	AC	Auxiliary carry ; useful only for BCD arithmetic									
5	F0	Flag 0; general purpose user flag									
4	RSI	Register bank selection bit 1									
3	RS0	Register bank selection bit 0									
		RSI	RS0								
		0	0	Bank 0							
		0	1	Bank 1							
		1	0	Bank 2							
		1	1	Bank 3							
2	OV	Overflow; used in arithmetic operations									
1	--	Reserved; may be used as a general purpose flag									
0	P	Parity; set to 1 if A has odd number of ones, otherwise reset to 0									

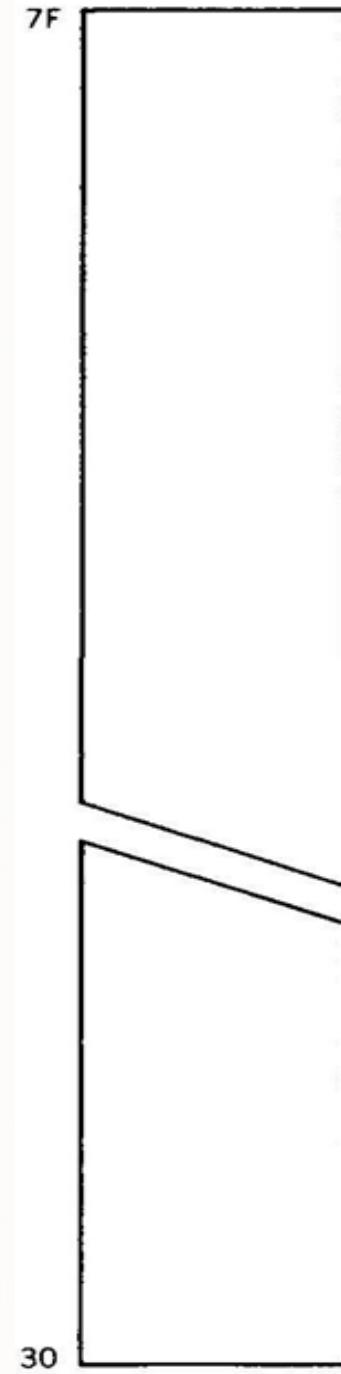
❖ *ON-CHIP MEMORY ORGANIZATION*

- Special function registers (SFRs)
- Internal RAM
- Internal ROM

Style

	Working Registers
00	R0
01	R1
02	R2
03	R3
04	R4
05	R5
06	R6
07	R7
08	R0
09	R1
0A	R2
0B	R3
0C	R4
0D	R5
0E	R6
0F	R7
10	R0
11	R1
12	R2
13	R3
14	R4
15	R5
16	R6
17	R7
18	R0
19	R1
1A	R2
1B	R3
1C	R4
1D	R5
1E	R6
1F	R7

1 microcontrollers		
2F	7F	78
2E	77	70
2D	6F	68
2C	67	60
2B	5F	58
2A	57	50
29	4F	48
28	47	40
27	3F	38
26	37	30
25	2F	28
24	27	20
23	1F	18
22	17	10
21	0F	08
20	07	00



INTERNAL RAM

INTERNAL RAM

- The 8051 has 128 bytes of internal RAM. It can be organized into three distinct areas.
 1. Working registers
 2. Bit addressable registers
 3. General purpose registers.

Working registers

- The first 32 bytes from address from 00 to 1F of internal RAM constitutes of 32 working registers.
- BANK 0 → 8 registers (R0-R7) : 00 to 07
- BANK 1 → 8 registers (R0-R7) : 08 to 0F
- BANK 2 → 8 registers (R0-R7) : 10 to 17
- BANK 3 → 8 registers (R0-R7) : 18 to 1F
- Bits RS0 & RS1 in the PSW determine which bank of register is currently in use.
- When 8051 is RESET , the BANK 0 is selected.



BIT ADDRESSABLE REGISTERS

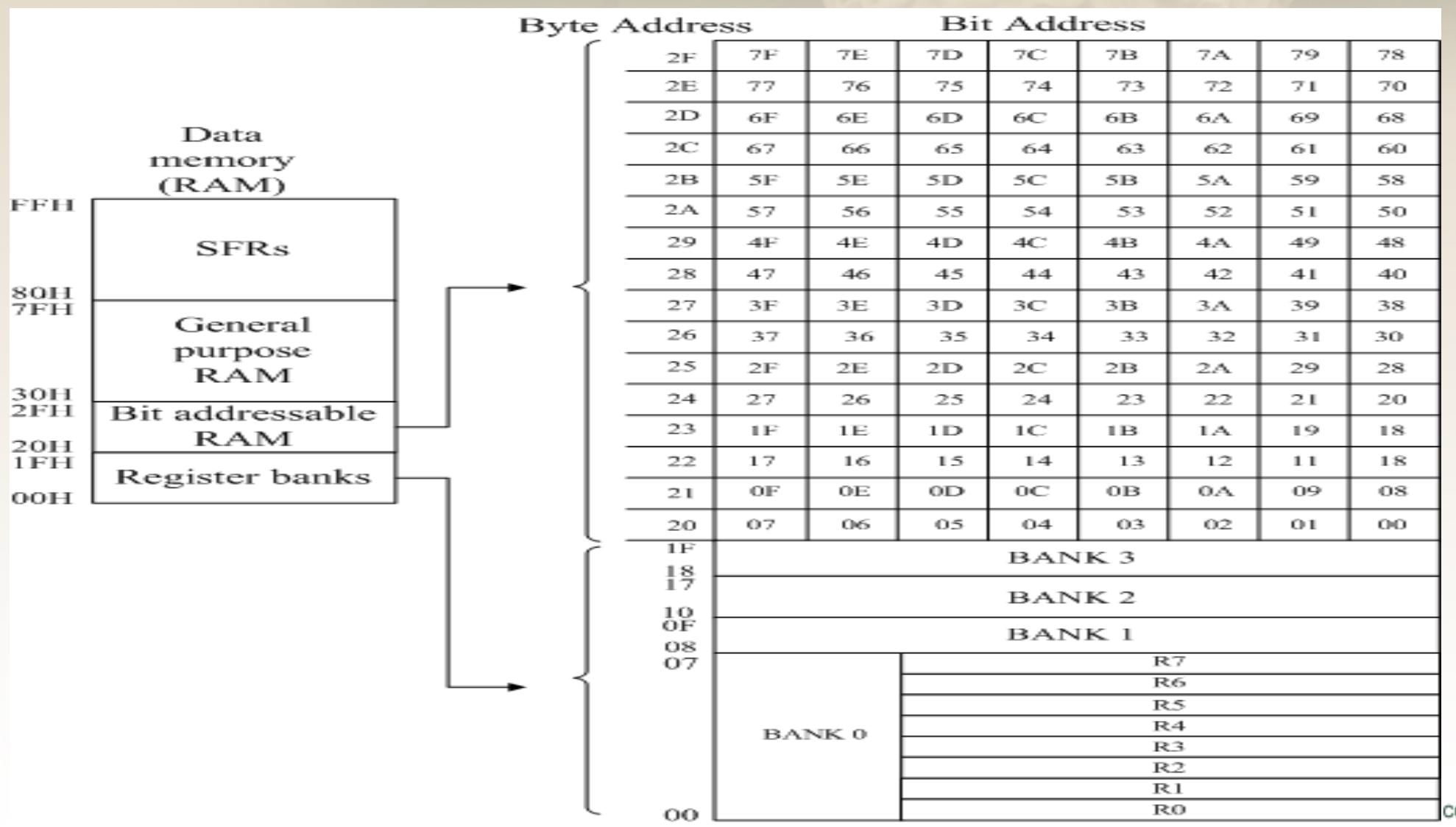
- The 8051 has a bit addressable area of 16 bytes from byte addresses 20H to 2FH in internal RAM, forming a total of 128 (16x8) addressable bits. An addressable bit can be accessed by its bit addresses from 00H to 7FH.

GENERAL PURPOSE REGISTERS

- Bytes from memory locations 30H to 7FH are used for general purpose data storage.



BIT ADDRESSABLE REGISTERS



Example 5-11

Find out to which bit each of the following bits belongs. Give the address of the RAM byte in hex

- (a) SETB 42H, (b) CLR 67H, (c) CLR 0FH
- (d) SETB 28H, (e) CLR 12, (f) SETB 05

Solution:

(a) D2 of RAM location 28H

(b) D7 of RAM location 2CH

(c) D7 of RAM location 21H

(d) D0 of RAM location 25H

(e) D4 of RAM location 21H

(f) D5 of RAM location 20H

	D7	D6	D5	D4	D3	D2	D1	D0
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

Special function registers (SFRs):

- Math registers: A and B.
- Status register: PSW (Program Status Word)
- Program counter: PC
- Pointer registers: DPTR (Data Pointer) and SP (Stack Pointer)
- Input output port latches : P0, P1, P2, and P3.
- Peripheral data registers: TL0, TH0, TL1, TH1, and SBUF.
- Peripheral control registers: IP, IE, TMOD, TCON, SCON, and PCON

Special function registers (SFRs):

- Special Function Register (SFR) are areas of memory that control specific functionality of the 8051 microcontroller.
- The SFRs allow the user to access the ports, serial communication, control and access timers, configure the 8051's interrupt system and power control system.
- Some SFRs can be accessed with bit operations also, i.e they are bit addressable Special Function Registers.

Special function registers (SFRs):

- SCON: (Serial Control) SCON is used to configure the behavior of the 8051's serial port.
- SBUF: (Serial Buffer) SBUF is used to send and received data via serial port.
- IE: (Interrupt Enable) IE register is used to enable and disable specific interrupt.
- IP: (Interrupt Priority) IP register is used to specify the priority of each interrupt.

Special function registers (SFRs):

- PCON: (Power Control) PCON register is used to control the 8051's power control mode.
- TCON: (Timer Control) TCON is used to control the 8051's timer and counter operation.
- TMOD: (Timer Mode) TMOD is used to control the mode of operation of timer and counter.
- TL0/TH0: (Timer 0 Low/High)
- TL1/TH1: (Timer 0 Low/High)



Style

80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF



Blue background are I/O port SFRs
 Yellow background are control SFRs
 Green background are other SFRs



SFR(SPECIAL FUNCTION REGISTER):

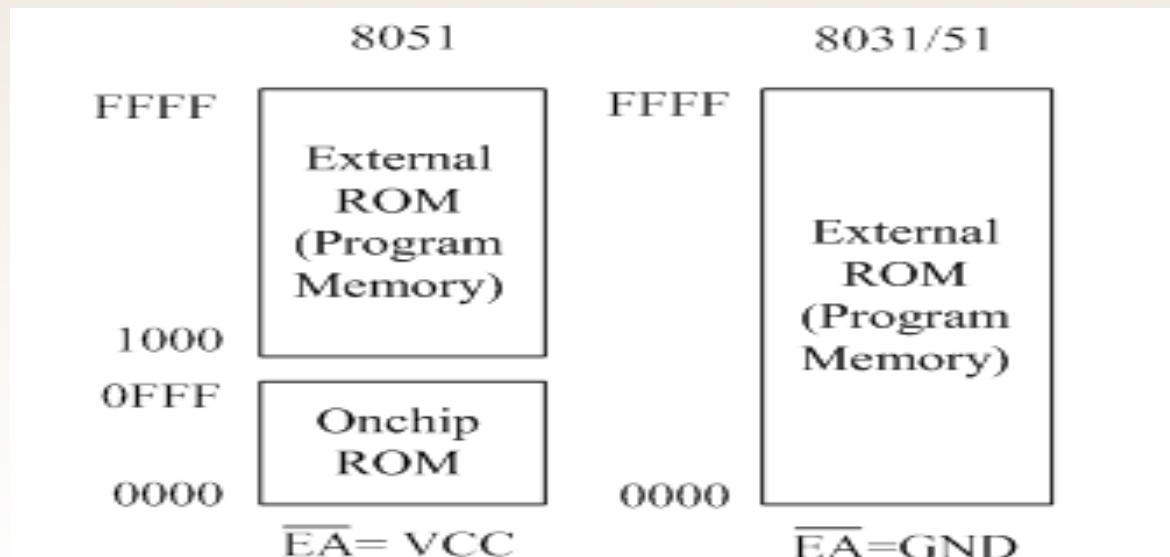
Byte address	Bit address	
FF		
F0	F7 F6 F5 F4 F3 F2 F1 F0	B
E0	E7 E6 E5 E4 E3 E2 E1 E0	ACC
D0	D7 D6 D5 D4 D3 D2 D1 D0	PSW
B8	-- -- BC BB BA B9 B8	IP
B0	B7 B6 B5 B4 B3 B2 B1 B0	P3
A8	AF -- AC AB AA A9 A8	IE
A0	A7 A6 A5 A4 A3 A2 A1 A0	P2
99	not bit-addressable	SBUF
98	9F 9E 9D 9C 9B 9A 99 98	SCON
90	97 96 95 94 93 92 91 90	P1
8D	not bit-addressable	TH1
8C	not bit-addressable	TH0
8B	not bit-addressable	TL1
8A	not bit-addressable	TL0
89	not bit-addressable	TMOD
88	8F 8E 8D 8C 8B 8A 89 88	TCON
87	not bit-addressable	PCON
83	not bit-addressable	DPH
82	not bit-addressable	DPL
81	not bit-addressable	SP
80	87 86 85 84 83 82 81 80	PO

Special Function Registers
www.botskool.com



INTERNAL ROM

- The 8051 contains 4Kbytes of internal ROM (on-chip). It occupies address range from 0000H to 0FFFH. Since it is used to store program instructions (code), it is also called program memory or code memory.

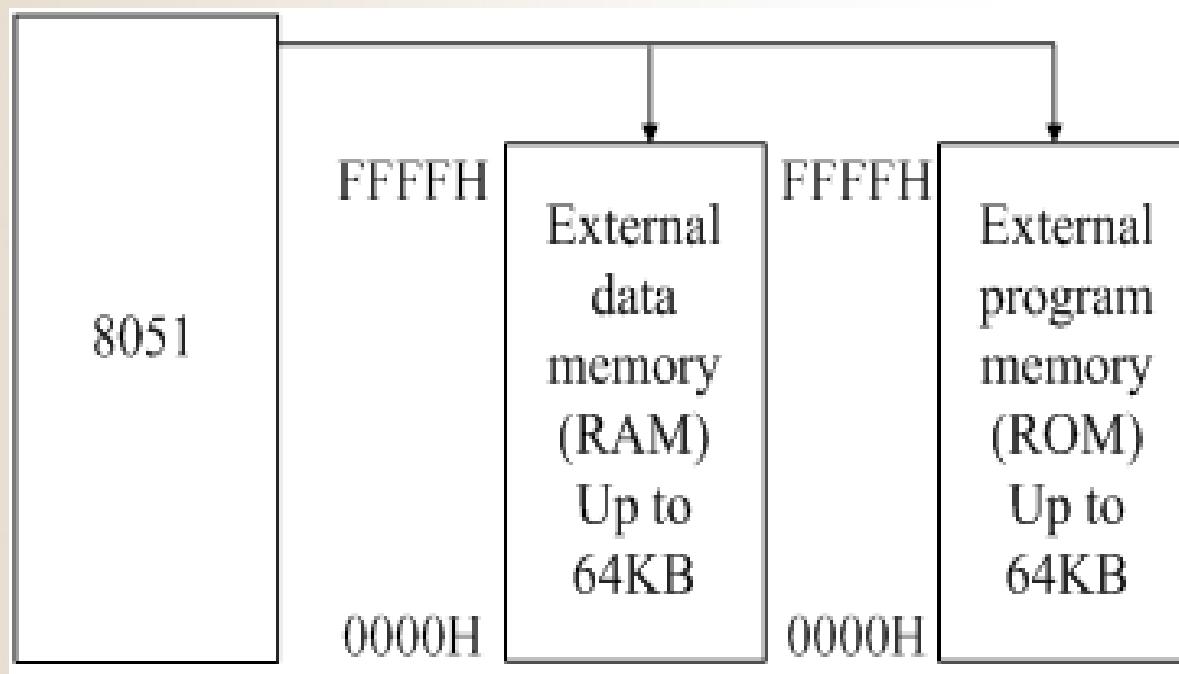


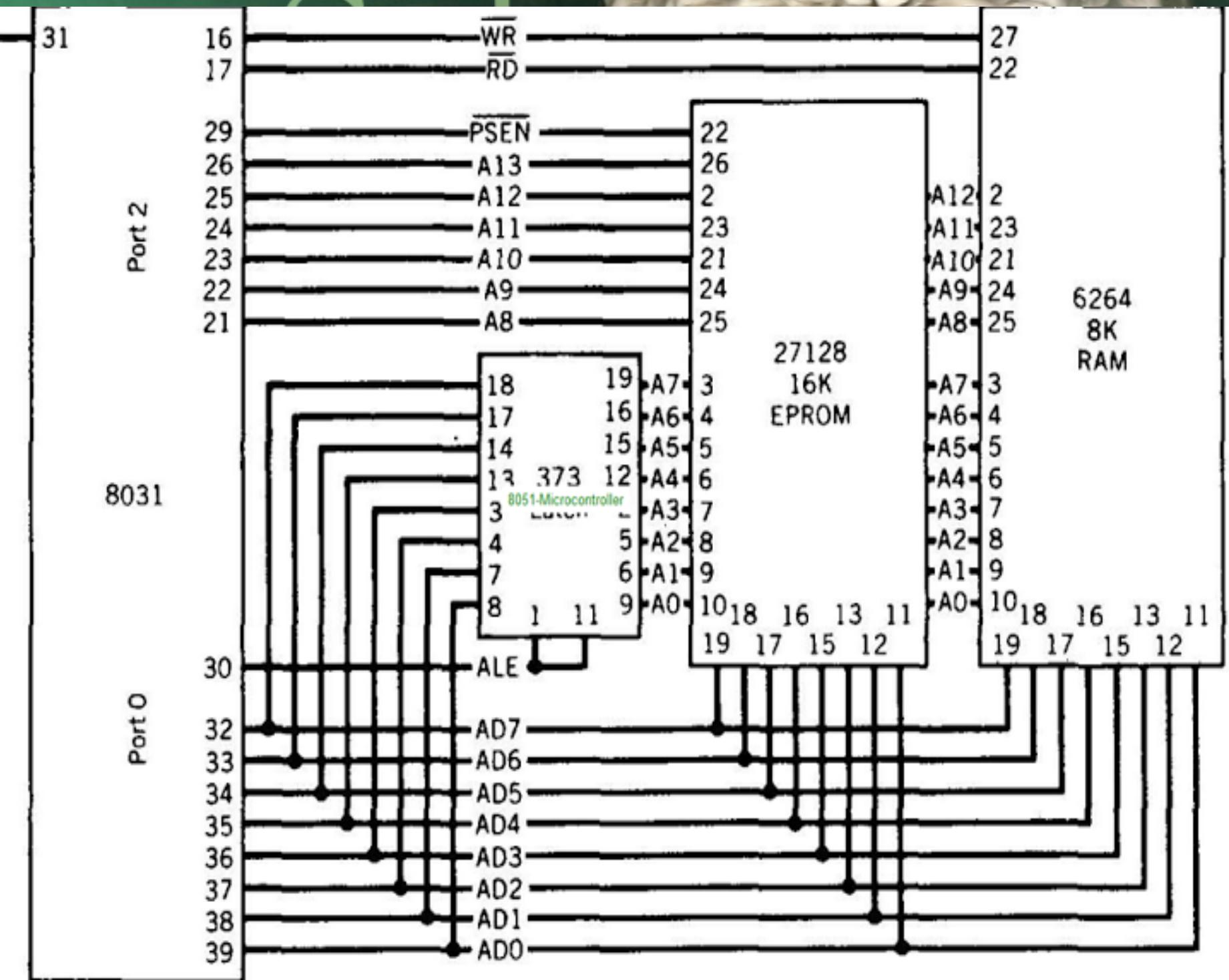
EXTERNAL MEMORY ORGANIZATION

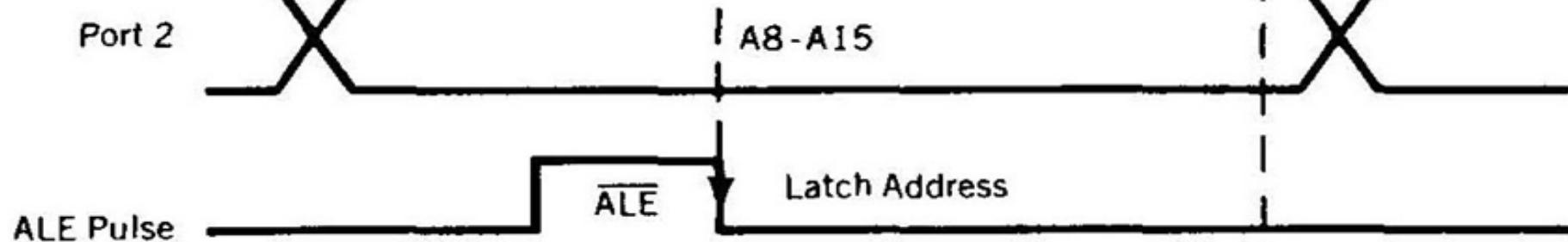
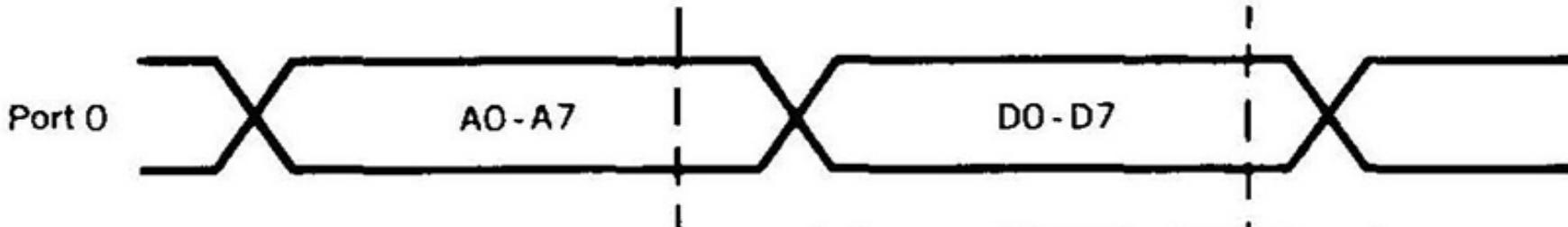
There are two parallel 64 Kilobytes address spaces

- one for the ROM(program memory).
- one for the RAM (data memory).

The data space is accessed using external data movement instructions (MOVX A, *source* or MOVX *destination*, A) and code space is accessed using external code movement instructions (MOVC A, *source*).







External Memory Addressing

8051-Microcontroller



Enable ROM

Reading ROM Using PSEN



Enable Read

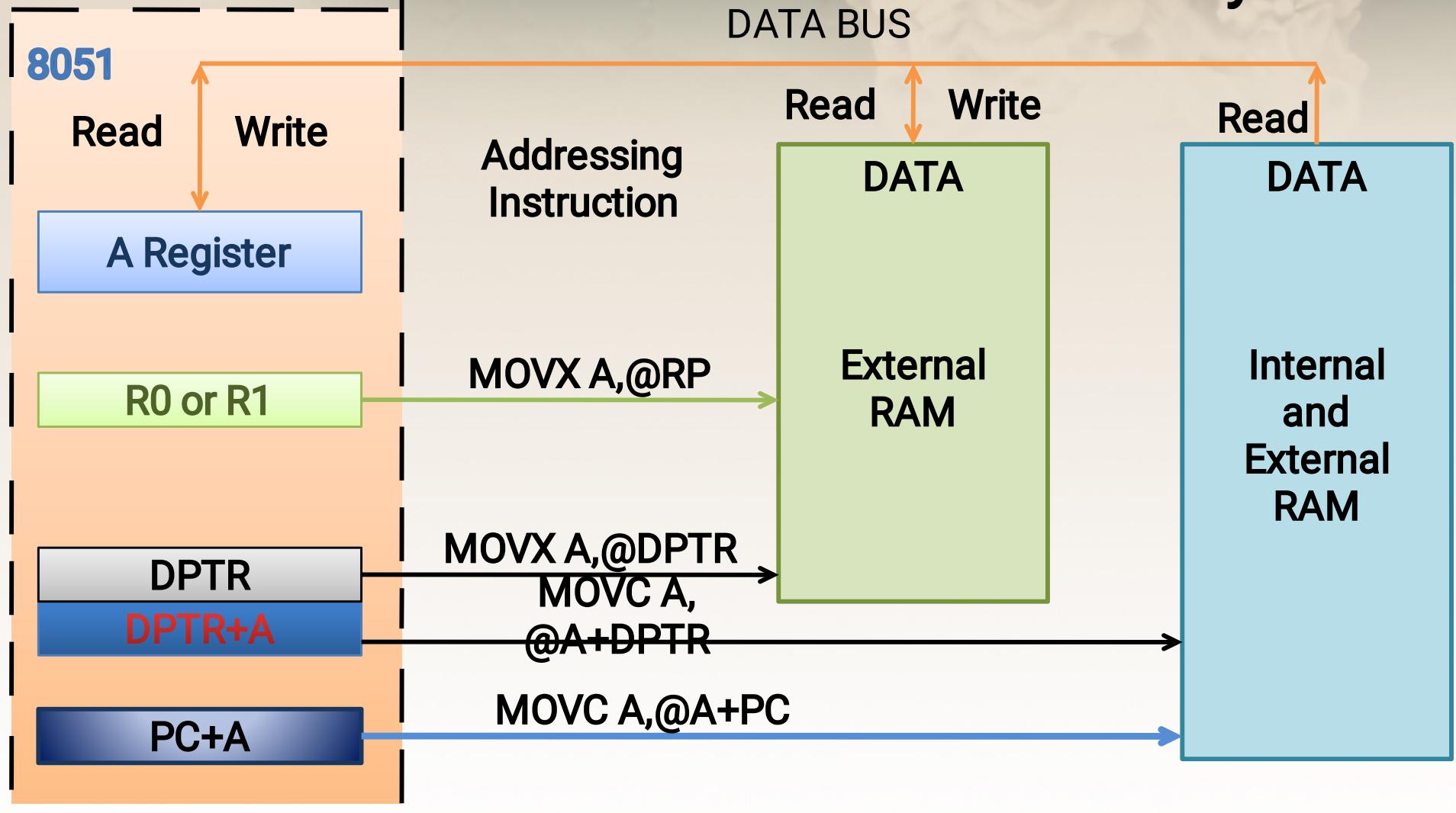


Enable Write

Accessing RAM Using \overline{RD} or \overline{WR}



Instruction to Access External Data Memory





ADDRESSING MODES of 8051

ADDRESSING MODES of 8051

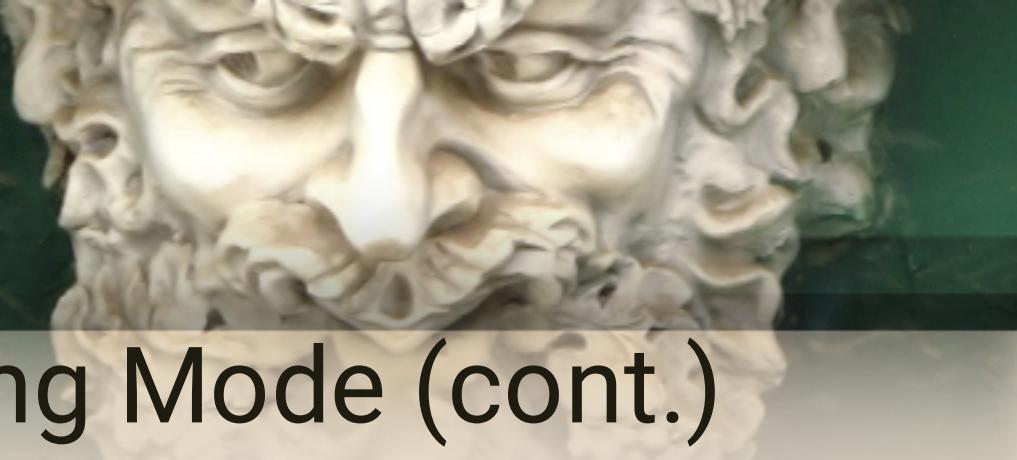
- The CPU can access data in various ways, which are called addressing modes
 - Immediate
 - Register
 - Direct
 - Register indirect
 - Indexed
 - The last three modes actually access memory



Immediate Addressing Mode

- The source operand is a constant
 - The immediate data must be preceded by the pound sign, “#”
 - Can load information into any registers, including 16-bit DPTR register

```
MOV A, #25H           ;load 25H into A
MOV R4, #62            ;load the decimal value 62 into R4
MOV B, #40H            ;load 4 4521H    B
MOV DPTR, #4521H       ;DPTR=4512H
```



Immediate Addressing Mode (cont.)

- DPTR can also be accessed as two 8-bit registers
 - The **high byte DPH** and **low byte DPL**

MOV D PTR, #2550H

is the same as:

MOV DPL, #50H

MOV DPH, #25H

MOV D PTR, #68975 ;illegal!! value > 65535 (FFFFFH)

Immediate Addressing Mode (cont.)

- We can use EQU directive to access immediate data

```
COUNT      EQU 30
...
MOV        R4, #COUNT      ; R4 = 1E (30=1EH)
MOV        DPTR, #MYDATA   ; DPTR=200H

ORG 200H
MYDATA: DB "America"
```

- We can also use immediate addressing mode to send data to 8051 ports

```
MOV P1, #55H
```

Register Addressing Mode

- Use registers to hold the data to be manipulated

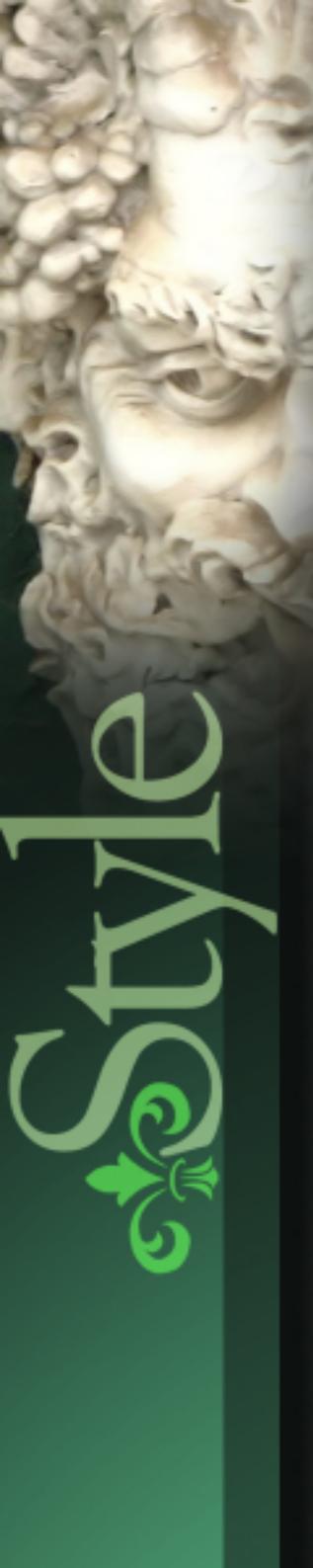
```
MOV A,R0 ; copy the contents of R0 into A  
MOV R2,A ; copy the contents of A into R2  
ADD A,R5 ; add the contents of R5 to contents of A  
ADD A,R7 ; add the contents of R7 to contents of A  
MOV R6,A ; save accumulator in R6
```

- The source and destination registers must match in size
 - MOV DPTR,A will give an error

Register Addressing Mode (cont.)

```
MOV DPTR, #25F5H  
MOV R7, DPL  
MOV R6, DPH
```

- The movement of data between Rn registers is not allowed
 - MOV R4, R7 is invalid

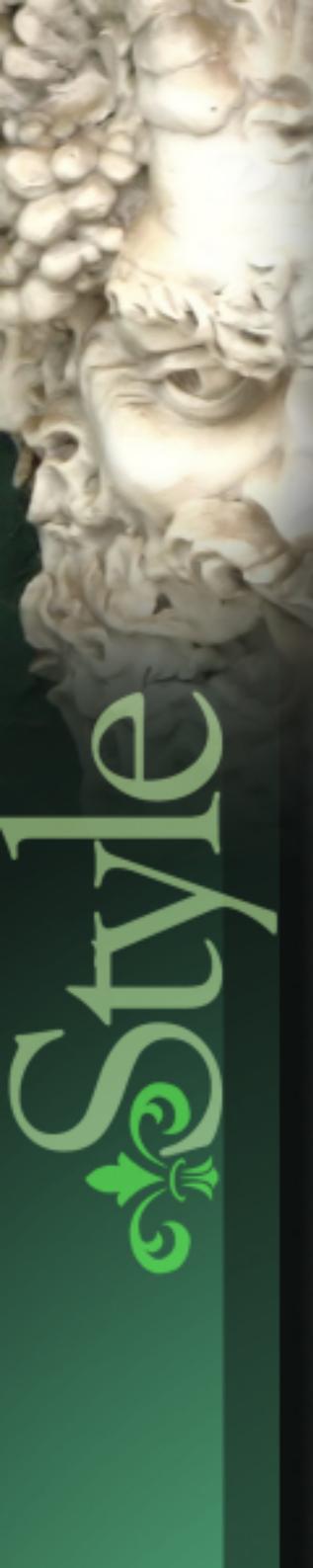


Direct Addressing Mode

- It is most often used the direct addressing mode to access RAM locations 30 – 7FH

```
MOV R0,40H ; save content of RAM location 40H in R0  
MOV 56H,A ; save content of A in RAM location 56H  
MOV R4,7FH ;move contents of RAM location 7FH to R4
```

MOV A,4	; is same as	;sed
MOV A,R4	; which means copy R4 into A	
MOV A,7	; is same as	
MOV A,R7	; which means copy R7 into A	

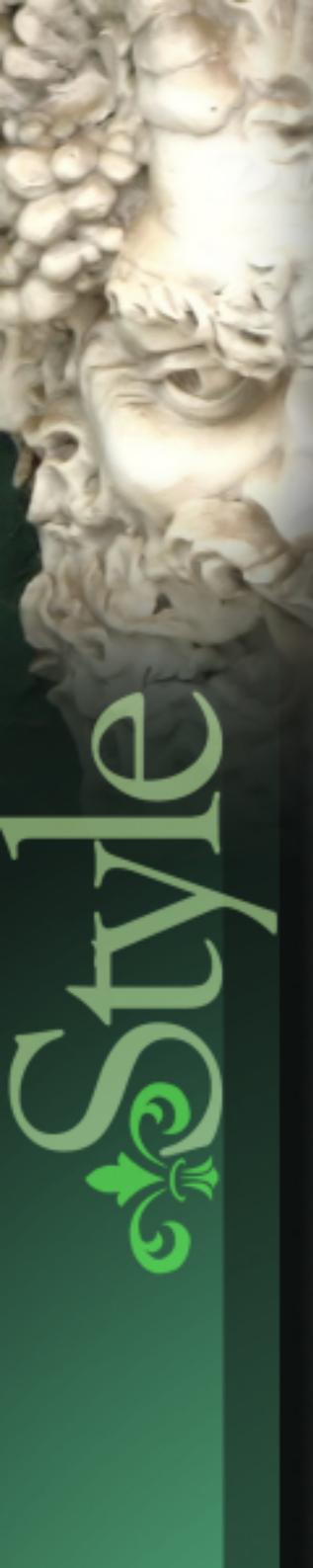


Direct Addressing Mode (cont.)

MOV A,2 ;is the same as
MOV A,R2 ;which means copy R2 into A

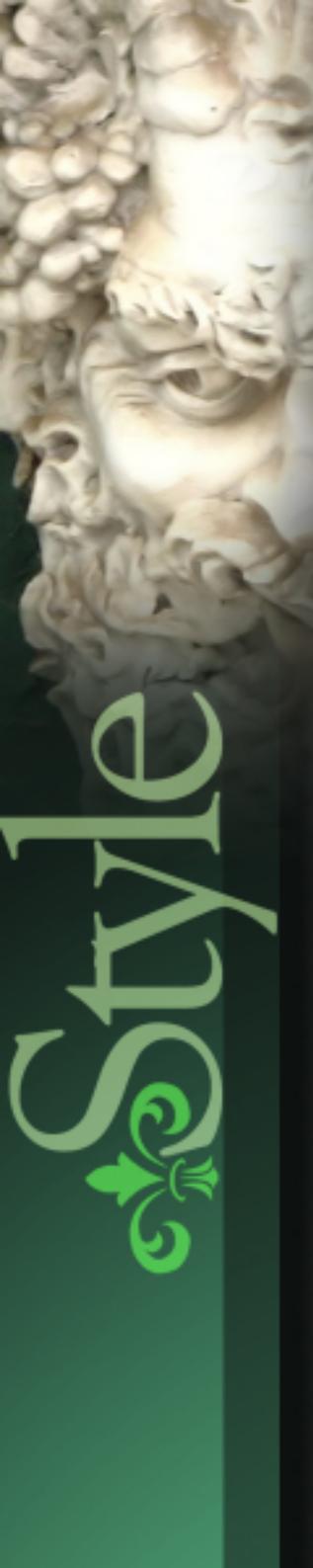
MOV A,0 ;is the same as
MOV A,R0 ;which means copy R0 into A

- Contrast this with immediate addressing mode
 - There is no “#” sign in the operand



SFR Registers and Their Addresses

- The SFR (Special Function Register) can be accessed by their names or by their addresses
 - The SFR registers have addresses between 80H and FFH
 - Not all the address space of 80 to FF is used by SFR
 - The unused locations 80H to FFH are reserved and must not be used by the 8051 programmer



SFR Registers and Their Addresses (cont.)

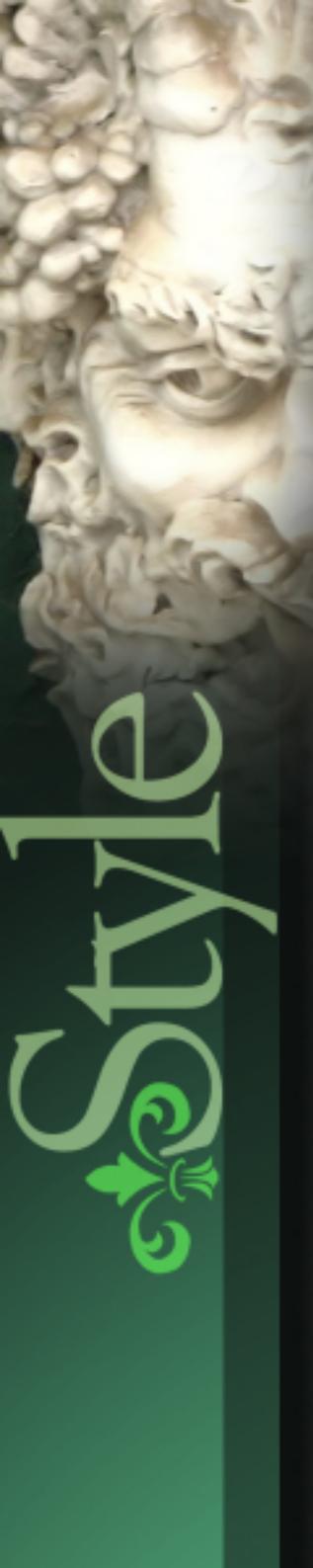
MOV 0EOH,#55H ;is the same as
MOV A,#55H ;which means load 55H into A (A=55H)

MOV 0FOH,#25H ;is the same as
MOV B,#25H ;which means load 25H into B (B=25H)

MOV 0EOH,R2 ;is the same as
MOV A,R2 ;which means copy R2 into A

MOV 0FOH,R0 ;is the same as
MOV B,R0 ;which means copy R0 into B

MOV P1, A ;is the same as
MOV 90H,A ;which means copy reg A to P1



SFR Registers and Their Addresses (cont.)

Table 5-1: 8051 Special Function Register (SFR) Addresses

Symbol	Name	Address
ACC*	Accumulator	0E0H
B*	B register	0F0H
PSW*	Program status word	0D0H
SP	Stack pointer	81H
DPTR	Data pointer 2 bytes	
DPL	Low byte	82H
DPH	High byte	83H
P0*	Port 0	80H
P1*	Port 1	90H
P2*	Port 2	0A0H
P3*	Port 3	0B0H
IP*	Interrupt priority control	0B8H
IE*	Interrupt enable control	0A8H

SFR Registers and Their Addresses (cont.)

TMOD	Timer/counter mode control	89H
TCON*	Timer/counter control	88H
T2CON*	Timer/counter 2 control	0C8H
T2MOD	Timer/counter mode control	0C9H
TH0	Timer/counter 0 high byte	8CH
TL0	Timer/counter 0 low byte	8AH
TH1	Timer/counter 1 high byte	8DH
TL1	Timer/counter 1 low byte	8BH
TH2	Timer/counter 2 high byte	0CDH
TL2	Timer/counter 2 low byte	0CCH
RCAP2H	T/C 2 capture register high byte	0CBH
RCAP2L	T/C 2 capture register low byte	0CAH
SCON*	Serial control	98H
SBUF	Serial data buffer	99H
PCON	Power control	87H

* Bit-addressable

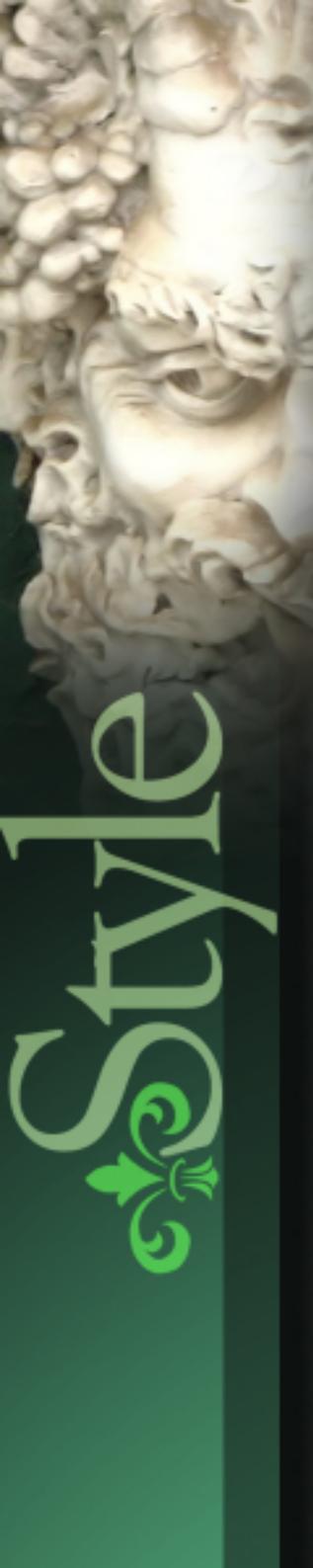
Example 5-1

Write code to send 55H to ports P1 and P2, using (a) their names, (b) their addresses.

Solution:

(a) MOV A, #55H ; A=55H
 MOV P1, A ; P1=55H
 MOV P2, A ; P2=55H

(b) From Table 5-1, P1 address = 90H; P2 address = A0H
 MOV A, #55H ; A=55H
 MOV 90H, A ; P1=55H
 MOV 0A0H, A ; P2=55H



Stack and Direct Addressing Mode

- Only direct addressing mode is allowed for pushing or popping the stack
 - PUSH A is invalid
 - Pushing the accumulator onto the stack must be coded as PUSH 0E0H

Example 5-2

Show the code to push R5 and A onto the stack and then pop them back them into R2 and B, where B = A and R2 = R5

Solution:

```
PUSH 05      ;push R5 onto stack
PUSH 0E0H    ;push register A onto stack
POP 0F0H     ;pop top of stack into B
              ;now register B = register A
POP 02      ;pop top of stack into R2
              ;now R2=R6
```

Example 5-3

Write a program to copy the value 55H into RAM memory locations 40H to 45H using

- (a) direct addressing mode,
- (b) register indirect addressing mode without a loop, and
- (c) with a loop.

Solution:

(a)

```
MOV A, #55H      ; load A with value 55H
MOV 40H, A       ; copy A to RAM location 40H
MOV 41H, A       ; copy A to RAM location 41H
MOV 42H, A       ; copy A to RAM location 42H
MOV 43H, A       ; copy A to RAM location 43H
MOV 44H, A       ; copy A to RAM location 44H
```



Style

(b)

```
MOV A,#55H      ;load A with value 55H
MOV R0,#40H     ;load the pointer. R0=40H
MOV @R0,A       ;copy A to RAM location R0 points to
INC R0          ;increment pointer. Now R0=41H
MOV @R0,A       ;copy A to RAM location R0 points to
INC R0          ;increment pointer. Now R0=42H
MOV @R0,A       ;copy A to RAM location R0 points to
INC R0          ;increment pointer. Now R0=43H
MOV @R0,A       ;copy A to RAM location R0 points to
INC R0          ;increment pointer. Now R0=44H
MOV @R0,A
```

(c)

```
MOV A,#55      ;A=55H
MOV R0,#40H    ;load pointer. R0=40H, RAM address
MOV R2,#05      ;load counter, R2=5
AGAIN: MOV @R0,A ;copy 55H to RAM location R0 points to
        INC R0    ;increment R0 pointer
DJNZ R2, AGAIN ;loop until counter = zero
```



Register Indirect Addressing Mode (cont.)

- The advantage is that it makes accessing data dynamic rather than static as in direct addressing mode
 - Looping is not possible in direct addressing mode

Example 5-4

Write a program to clear 16 RAM locations starting at RAM address 60H.

Solution:

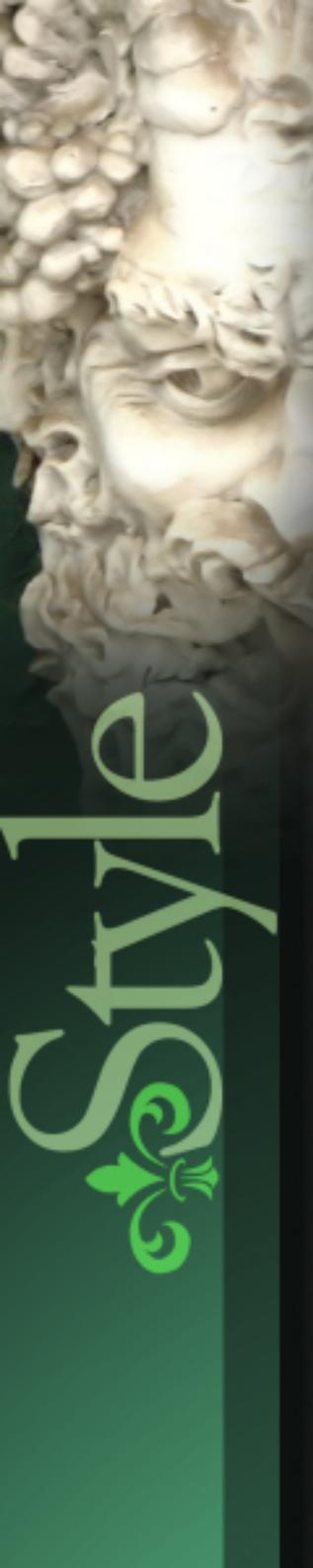
```
CLR      A          ;A=0
MOV      R1, #60H    ;load pointer. R1=60H
MOV      R7, #16     ;load counter, R7=16 (10 in hex)
AGAIN: MOV      @R1, A   ;clear RAM location R1 points to
      INC      R1        ;increment R1 pointer
      DJNZ    R7, AGAIN  ;loop until counter = zero
```

Example 5-5

Write a program to copy a block of 10 bytes of data from RAM locations starting at 35H to RAM locations starting at 60H.

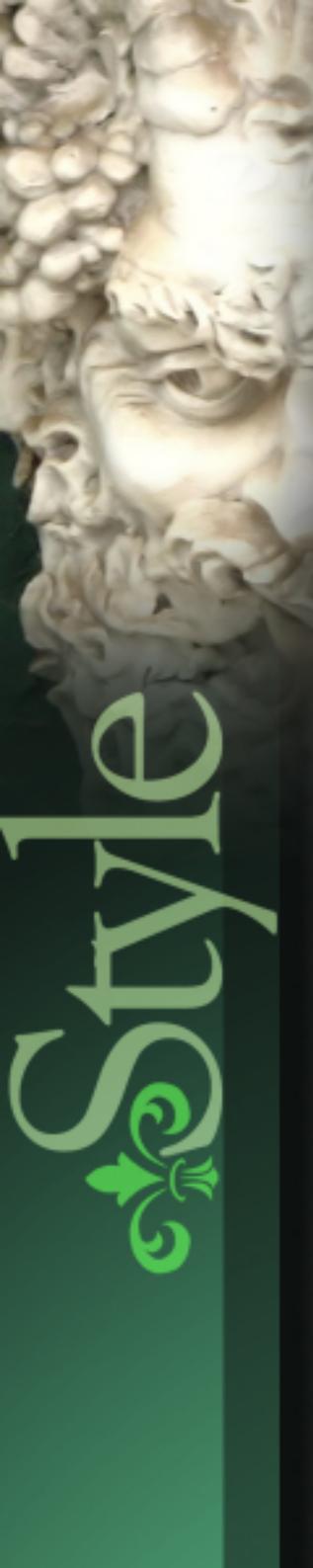
Solution:

```
MOV R0, #35H      ;source pointer
MOV R1, #60H      ;destination pointer
MOV R3, #10       ;counter
BACK:   MOV A, @R0    ;get a byte from source
        MOV @R1, A    ;copy it to destination
        INC R0        ;increment source pointer
        INC R1        ;increment destination pointer
        DJNZ R3, BACK  ;keep doing it for all ten bytes
```



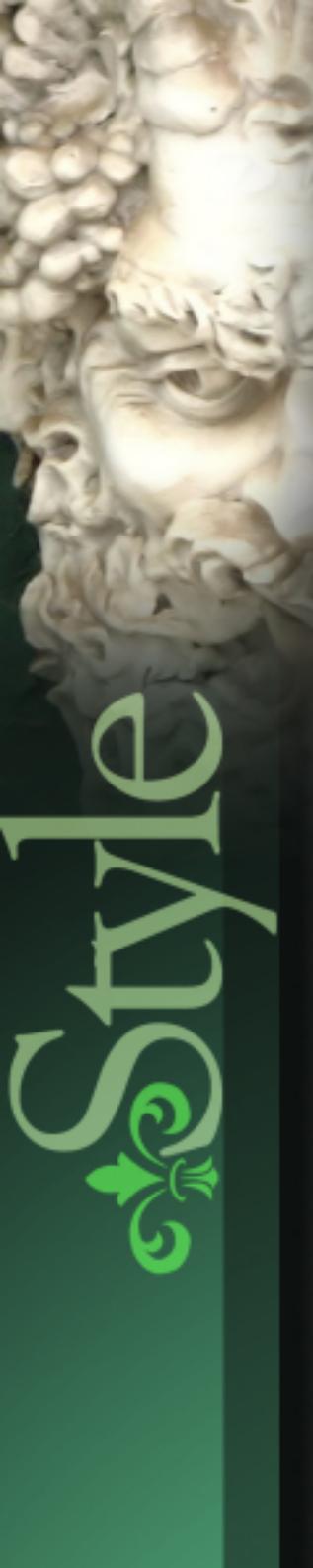
Register Indirect Addressing Mode (cont.)

- R0 and R1 are the only registers that can be used for pointers in register indirect addressing mode
 - Since R0 and R1 are 8 bits wide, their use is limited to access any information in the internal RAM
- Whether accessing externally connected RAM or on-chip ROM, we need 16-bit pointer
 - In such case, the DPTR register is used



Indexed Addressing Mode and On-chip ROM Access

- Indexed addressing mode is widely used in accessing data elements of look-up table entries located in the program ROM
 - The instruction used for this purpose is **MOVC A,@A+DPTR**
 - Use instruction MOVC,
 - “C” means code
 - The contents of A are added to the 16-bit register DPTR to form the 16-bit address of the needed data



Indexed Addressing Mode and MOVX

- In many applications, the size of program code does not leave any room to share the 64K-byte code space with data
 - The 8051 has another 64K bytes of memory space set aside exclusively for data storage
 - This data memory space is referred to as external memory and it is accessed only by the MOVX instruction
 - The 8051 has a total of 128K bytes of memory space
 - 64K bytes of code and 64K bytes of data
 - The data space cannot be shared between code and data

Example 5-10

Write a program to toggle P1 a total of 200 times. Use RAM location 32H to hold your counter value instead of registers R0 – R7

Solution:

```
MOV    P1, #55H ;P1=55H
MOV    32H, #200 ;load counter value
                  ;into RAM loc 32H
LOP1: CPL    P1      ;toggle P1
      ACALL  DELAY
      DJNZ   32H, LOP1 ;repeat 200 times
```