

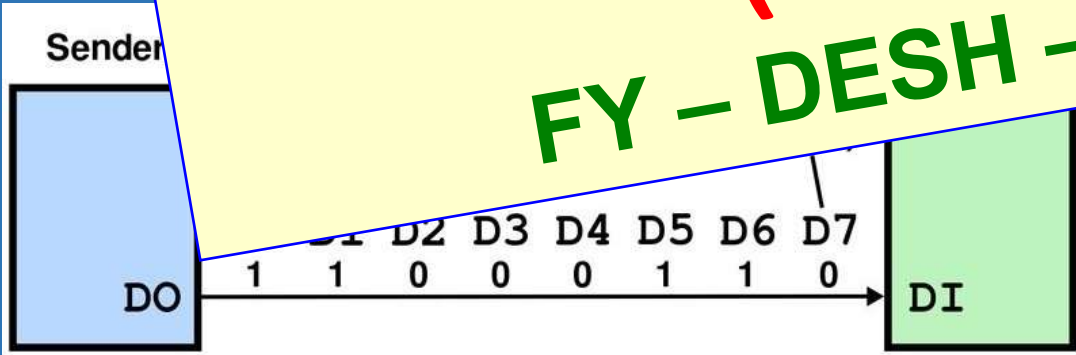
Address	Contents of memory
0x0005	1011 1110 1000 1111 00
0x0004	1011 1111 0111 0101 01
0x0003	1011 1111 0100 0001 10
0x0002	0011 1110 0001 0000 10
0x0001	0011 1111 0110 10
0x0000	0000 0000 0000 0000

– Addresses of the 21 SFRs :-

- 8A – TL0
- 8B – TL1
- 8C – TH0
- 8D – TH1

- 80 – P0\* (Port 0)
- 90 – P1\* (Port 1)

Micro Controllers – 2.3  
8051 continued ...  
Special Function Registers  
(SFRs)  
FY – DESH – VIT



1E	1D	1C	1B	1A	19	18
16	15	14	13	12	11	10
0E	0D	0C	0B	0A	09	08
06	05	04	03	02	01	00

# Addressability – Bit and Byte

- **Bit – Nibble – Byte – Word**
- **Word** can be of 8 bits or 16 bits or 32 bits or 64 bits etc....
- **Bit addressable** : Those memory locations that can be addressed / accessible by a single bit are called as bit addressable. There is a specific memory zone / area in 8051, where this is possible. Each bit has its unique address.
- **Byte addressable** : A complete byte can be tackled together. The complete byte has a unique address and the internal bits do not have their own address. Thus, to change a single bit in a byte, you have to change all bits (keeping others intact). This has to be done carefully !

# Addressability – Bit and Byte

- **Word addressable** : Each memory location is either 8 bit or 16 bit or 32 bit or 64 bit size.
- This complete word of 8, 16, 32 or 64 bits is handled together.
- Suppose, we want to store a number say 12 (decimal)

12 (decimal) = CH ..... What is binary of this ??

12 = 1100H .....but this can be .....

12 = 1100 (4 bit system)

12 = 0000 1100 (8 bit system)

12 = 0000 0000 0000 1100 (16 bit system)

12 = 0000 0000 0000 0000 0000 0000 0000 1100 (32 bit system)

and so on .....

# Addressability – Bit and Byte

**Look at the address of Memory location.**

**Here individual byte has its own address.**

**Same is true for bit addressable memory.**

	Contents of memory
Addr6	1011 1110 1000 1111 0000 1111 1000 1100
Addr5	1011 1111 0111 0101 0111 1100 0001 0000
Addr4	1011 1111 0100 0001 1011 1101 1100 1111
Addr3	0011 1110 0001 0000 1000 0001 1100 0011
Addr2	0011 1111 0110 1000 1100 0111 1011 0111
Addr 1	0011 1111 0101 0111 0110 1010 1010 0100

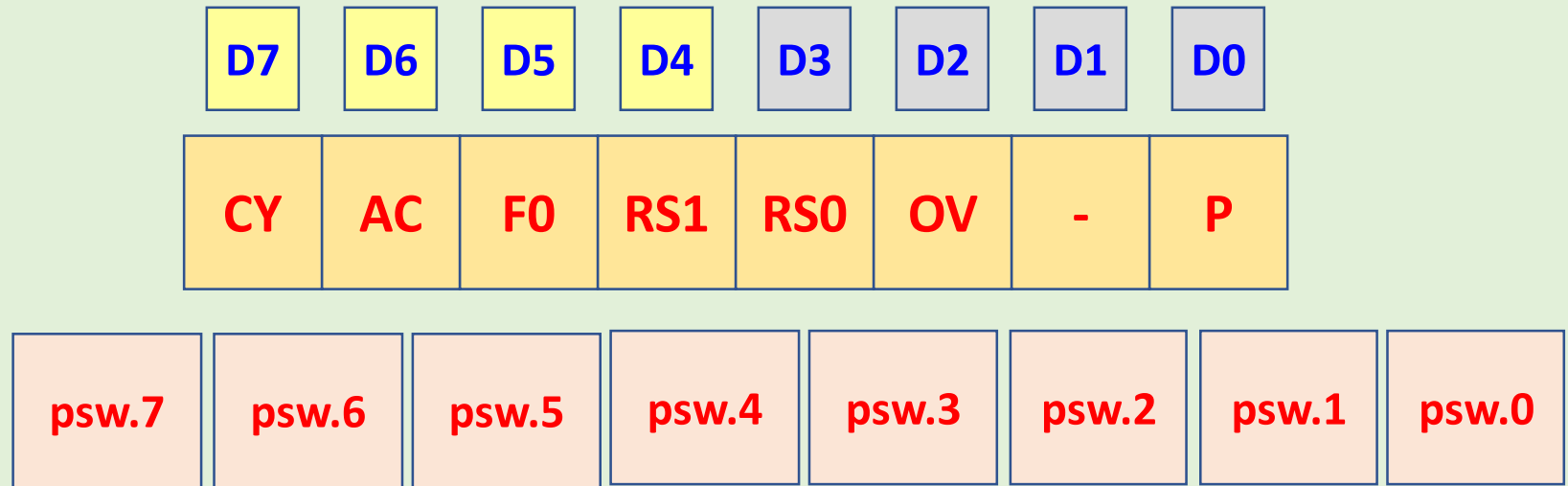
- 1) How many bytes are in the memory ? 1) 24 Bytes
- 2) What is the size of the address ? 2) 16 bits
- 3) How many bits are in the word ? 3) 32 bits

	Contents of memory			
Addr21	1011 1110	1000 1111	0000 1111	1000 1100
Addr17	1011 1111	0111 0101	0111 1100	0001 0000
Addr13	1011 1111	0100 0001	1011 1101	1100 1111
Addr9	0011 1110	0001 0000	1000 0001	1100 0011
Addr5	0011 1111	0110 1000	1100 0111	1011 0111
Addr 1	0011 1111	0101 0111	0110 1010	1010 0100

24 bytes of memory using 16-bit addresses  
and byte addressing

## Example of bit addressability

### Program Status Word – PSW – 8 bit Register



**PSW is a bit addressable register**

# Addressability – Bit and Byte

- Bit addressable instructions – **assume that this a bit addressable memory**

- For e.g.

SETB 00H

SETB 07H

CLR 04H

SETB 2EH

0x05	2F	2E	2D	2C	2B	2A	29	28
0x04	27	26	25	24	23	22	21	20
0x03	1F	1E	1D	1C	1B	1A	19	18
0x02	17	16	15	14	13	12	11	10
0x01	0F	0E	0D	0C	0B	0A	09	08
0x00	07	06	05	04	03	02	01	00

All above instructions

will change the bits

directly at the given memory location.

**Different bit addresses**

# Addressability – Bit and Byte

Bit addressable locations in 8051

20H to 2FH = How many Bytes ?

= 16 Bytes

Total bits =  $16 \times 8 = 128$  bits

All bits are directly addressable.

e.g. what will happen below ?

SETB 20H ..... ?

CLR 2DH ..... ?

Registers like ACC, B, P0, P1, P2, P3, PSW, IP, IE, PCON, SCON are bit addressable.

Other registers are Byte addressable.

0x2F	7F	7E	7D	7C	7B	7A	79	78
0x2E	77	76	75	74	73	72	71	70
0x2D	6F	6E	6D	6C	6B	6A	69	68

0x25	2F	2E	2D	2C	2B	2A	29	28
0x24	27	26	25	24	23	22	21	20
0x23	1F	1E	1D	1C	1B	1A	19	18
0x22	17	16	15	14	13	12	11	10
0x21	0F	0E	0D	0C	0B	0A	09	08
0x20	07	06	05	04	03	02	01	00

# Special Function Registers – Features

- In 8051 microcontroller there are certain registers which use the RAM addresses from 80H to FFH and they are meant for certain specific operations. (Upper 128 bytes).
- These registers are called Special function registers (SFRs).
- Some of these registers are bit and byte addressable.
- Some of SFRs are related to I/O ports (P0,P1,P2 and P3) and some are meant for control operations (TCON,SCON, PCON..) and remaining are the auxiliary SFRs.



# Special Function Registers – Features

- **SFRs Memory addresses are only direct addressable. Even though some of the addresses between 80H and FFH are not assigned to any SFR, they cannot be used as additional RAM area.**
- **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.**

## –: Addresses of the 21 SFRs :-

**8A – TL0**

**8B – TL1**

**8C – TH0**

**8D – TH1**

**88 – TCON\***

**89 – TMOD**

**98 – SCON\***

**99 – SBUF**

**82 – DPL**

**83 – DPH**

**87 – PCON**

**80 – P0\* (Port 0)**

**90 – P1\* (Port 1)**

**A0 – P2\* (Port 2)**

**B0 – P3\* (Port 3)**

**E0 – A\* (Accumulator)**

**F0 – B\* (Extension of A)**

**81 – SP**

**D0 – PSW\***

**A8 – IE\* (Interrupt )**

**B8 – IP\* (Interrupt )**

**Note – There are gaps in the addresses of the SFRs.**

**Put a leading 0 if the address starts with a alphabet. e.g. 0A0, 0E0**

**Registers with a \* are bit and byte addressable. Other only byte addressable.**

# 1) SFR Register – Accumulator or A – E0H

ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	0E0H
E7	E6	E5	E4	E3	E2	E1	E0	

The most important of all special function register is accumulator which is also known as ACC or A.

- The Accumulator holds the result of most of arithmetic and logic operations.
- To access the first bit (bit 0) or to access accumulator as a single byte (all 8 bits at once), A is accessed by direct addressing and its physical address is E0H.
- Accumulator is *both bit and byte addressable*.
- If you want to access the 2<sup>nd</sup> bit (bit 1), you may use E1H directly.
- e.g. --- SETB ACC.5    OR    CLR ACC.2

## 2) SFR--Register B: F0H

B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	0F0H
F7	F6	F5	F4	F3	F2	F1	F0	

**Register B is Bit and Byte addressable.**

**The major purpose of this register is in executing multiplication and division.**

The 8051 micro controller has a single instruction for multiplication (**MUL**) and division (**DIV**).

**Ex: MUL A,B** – When this instruction is executed, data inside **A** and data inside **B** is multiplied and answer is stored back in A.

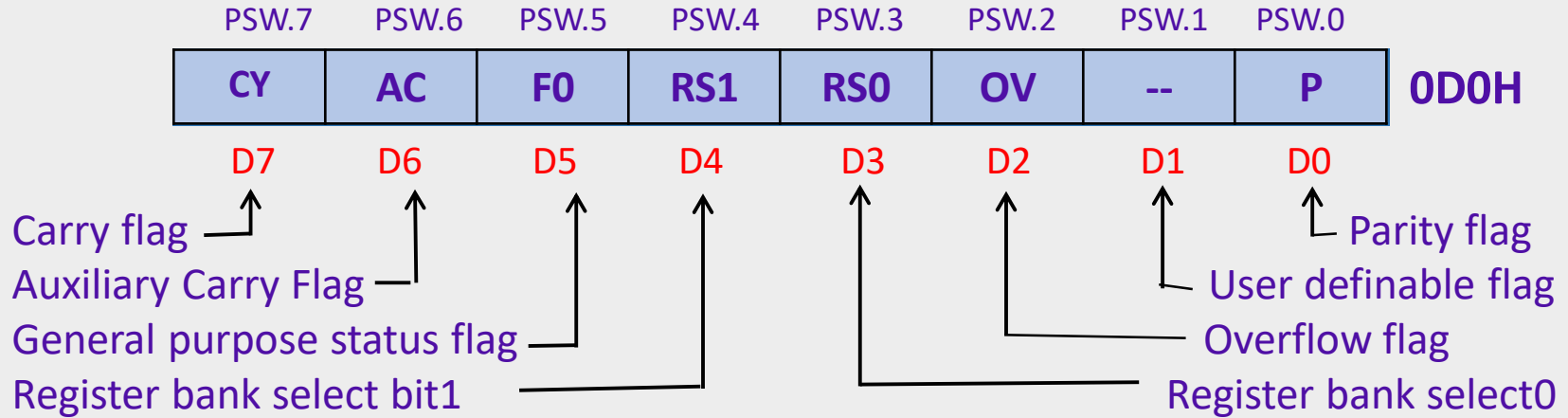
### 3) Stack Pointer SP: 81H

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
------	------	------	------	------	------	------	------

**81H**

- Stack pointer is used to hold the address of internal RAM.
- Stack pointer is an 8 bit register, the direct address of **SP is 81H** and it is only **byte addressable**, which means you cannot access individual bits of stack pointer.
- The content of the stack pointer points to the **“last location”** of system stack. To store something new in system stack, the SP must be incremented by 1 first and then execute the “store” command.
- **Usually after a system reset, SP is initialized as 07H** and data can be stored to stack from 08H onwards. **(Why not 07H onwards ?)** This is usually a default case and programmer can alter values of SP to suit his/her needs.

## 4) Program Status Word (PSW): D0H



- This is a vital SFR in the functioning of micro controller.
- This register reflects the status of the operation that is being carried out in the processor.
- **PSW Register is both bit and byte addressable.** The physical address of PSW starts from D0H. The individual bits are then accessed using D1, D2 ... D7.

## 4) Program Status Word (PSW)

Bit No	Bit Symbol	Direct Address	Name	Function
0	P	D0	Parity	This bit will be set if ACC has odd number of 1's after an operation. If not, bit will remain cleared.
1	–	D1		User definable bit
2	OV	D2	Overflow	OV flag is set if there is a carry from bit 6 for signed number operation, but not from bit 7 of an Arithmetic operation.
3	RS0	D3	Register Bank select	LSB of the register bank select bit. Bit 0
4	RS1	D4	Register Bank select	MSB of the register bank select bits. Bit 1
5	F0	D5	Flag 0	User defined flag
6	AC	D6	Auxiliary carry	This bit is set if data is coming out from bit 3 to bit 4 of Acc during an Arithmetic operation.
7	CY	D7	Carry	Is set if data is coming out of bit 7 of Acc during an Arithmetic operation.

## 4) Program Status Word (PSW)

The selection of the register Banks in PSW and their addresses

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

### **CY -- Carry flag**

This flag is set whenever there is a carry out from the D7 bit. This flag bit is affected after an 8-bit addition or subtraction. It can also be set to 1 or 0 directly by an instruction such as “SETB C” and “CLR C” where “SETB C” stands for “set bit carry” and “CLR C” for “clear carry”.

### **AC -- Auxiliary carry flag**

If there is a carry from D3 to D4 during an ADD or SUB operation, this bit is set; otherwise, it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.



## 4) Program Status Word (PSW)

### *P -- Parity flag*

The parity flag reflects the number of 1 s in the A (accumulator) register only. If the A register contains an odd number of 1s, then  $P = 1$ . Therefore,  $P = 0$  if A has an even number of 1s.

### *OV -- Overflow flag*

This flag is set whenever the result of a signed number operation is large, causing the high-order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations. The overflow flag is only used to detect errors in signed arithmetic operations

## 4) Program Status Word (PSW)

Impact of the ADD instruction on the flag bits CY, AC, and P of the PSW register.

```

1.  MOV A, #38H      0011 1000
    ADD A, # 2FH      0010 1111
                        -----
                        0110 0111
  
```

CY = 0, AC = 1, P = 1 as ACC has five 1's

```

2.  MOV A, # 9CH      1001 1100
    ADD A, # 64H      0110 0100
                        -----
                        1 0000 0000
  
```

CY = 1, AC = 1, P = 0 as ACC has zero 1's

```

3.  MOV A, #88H      1000 1000
    ADD A, #93H      1001 0011
                        -----
                        1 0001 1011
  
```

CY = 1, AC = 0, P = 0 as ACC has four 1's

# Port -- 8051 internal i/o ports

P0, P1, P2 and P3 are byte addresses used to access and perform read or write or other operations.

Direct 8-bit addresses of each are specified in the instructions.

Addresses of bytes at	P0– 080	Floating pins	← Truly Bidirectional
	P1– 090	Internal Pull-ups	} Quasi Bidirectional
	P2– 0A0	Internal Pull-ups	
	P3– 0B0	Internal Pull-ups	

Let us see the details .....

# Port -- 8051 internal i/o ports

- Each port of 8051 has bidirectional capability.
- Port 0 is called 'true bidirectional port' **as it floats (tristate) when configured as input.**
- A quasi-bidirectional I/O is **a type of input-output port on an integrated circuit such as a PIA.** It can be used as an input or output without the use of a control signal for data direction. At power-on the I/Os are HIGH. In this mode, only a current source to VDD is active.

**Let us see the details .....**

## 5) Port 0 and Register P0 – 80H

P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	<b>80H</b>
87	86	85	84	83	82	81	80	

Register P0 is used to control Port 0 pins (32-39).

- When the external memory is used with microcontroller, then the lower address byte (addresses A0-A7) is applied on P0.

- If external memory is not used, all bits of P0 are configured for I/O purposes.

➤ pin no. 30 = 0 will make the port 0 for I/O and pin 30 = 1 for address port.

e.g. pin 30 = 0 and P0.2 = 1 means ....

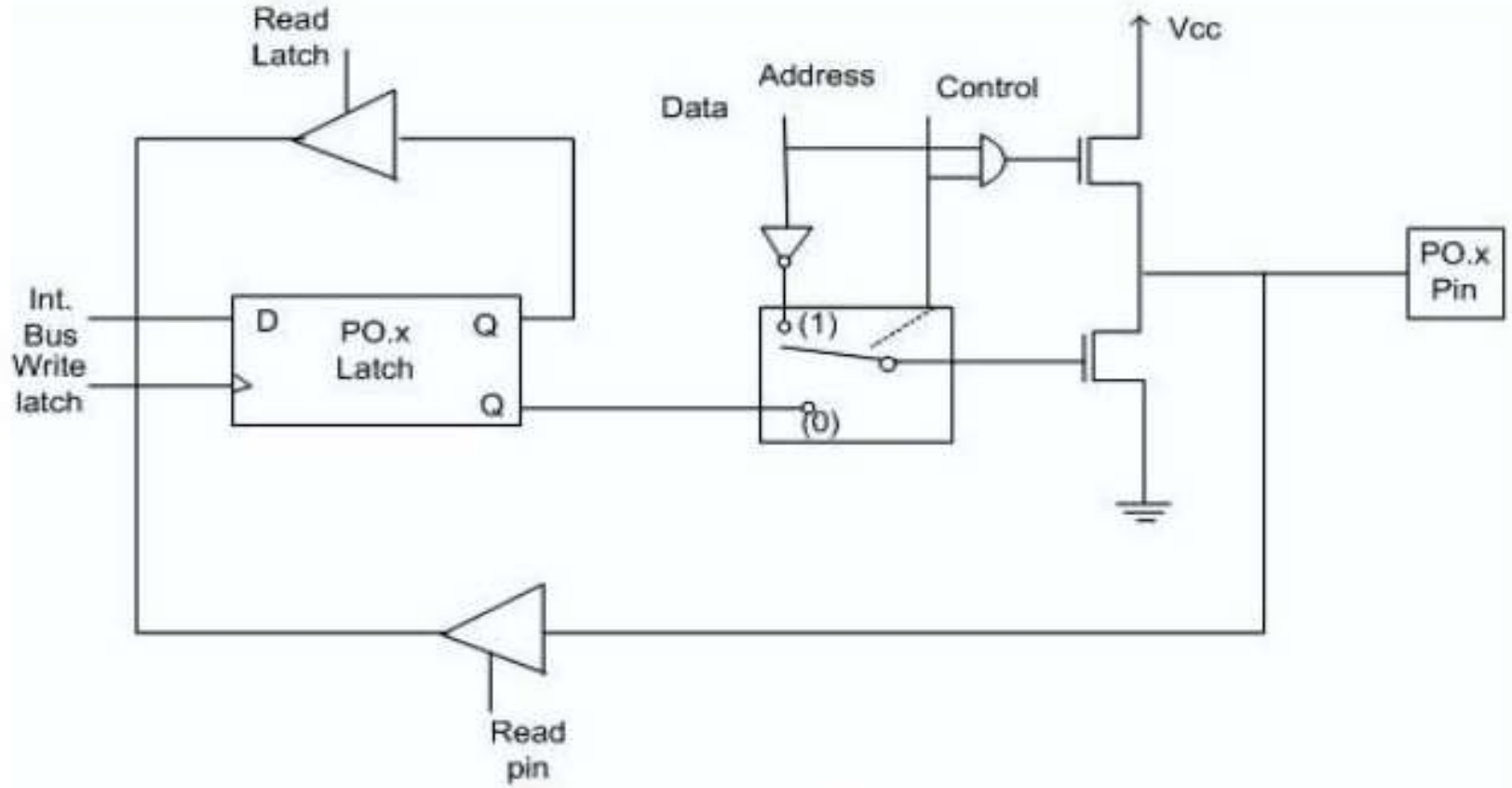
e.g. pin 30 = 0 and P0.6 = 0 means ....

P0 does not contain built-in pull-up resistors. It is a true bidirectional port.

Each pin must be connected to external pull up resistors of 10K, if port is to be used as I/O.

**P0 Register is both bit and byte addressable.**

# Port -- 8051 internal i/o ports



Port0 Structure

## 6) Port 1 and Register P1 : 90H

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	90H
97	96	95	94	93	92	91	90	

- ✓ Port 1 is a dedicated I/O port from pin 1 to pin 8.
- ✓ It is generally used for interfacing to external device like LED, sensors, switches, motors etc.
- ✓ Port pins are internally pulled up. As an I/O port: Standard quasi-bidirectional.
- ✓ P1 is a true I/O port as it doesn't have any alternative functions as in P0. It has a built-in pull-up resistors.

e.g. Writing 1 to 93 will make pin no. \_\_\_\_ as input pin.

Writing 0 to 90 will make pin no. \_\_\_\_ as output pin.

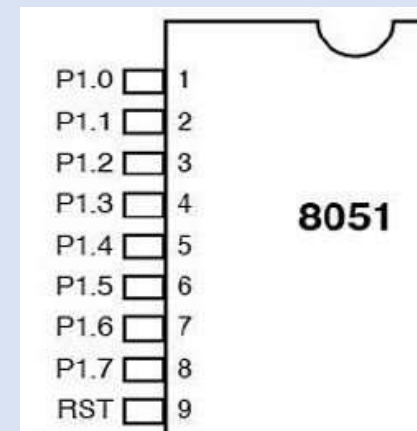
By default, port is configured as input on power-on.

**P1 Register is both bit and byte addressable.**

e.g. **MOV A, #0E5h;**

**MOV P1, A;**

**(find status of all 8 pins)**



## 7) Port 2 and Register P2 -- A0 H

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	<b>A0H</b>
A7	A6	A5	A4	A3	A2	A1	A0	

Like port 0, port 2 is a dual-purpose port.(Pins 21 through 28)

Like P1 ,Port 2 also doesn't require any pull-up resistors. As an I/O port: Standard quasi-bidirectional.

Alternate functions: High byte of address bus for external program and data memory accesses.

Writing 1 to any bit of a port configures that pin as input pin.

while writing 0 configures that as output.

By default, port is configured as input on power-on.

Register is both bit and byte addressable.



## 7) Port 2 and Register P2 -- A0 H

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	<b>A0H</b>
A7	A6	A5	A4	A3	A2	A1	A0	

MOV A, #0E5h

MOV P2, A

- Find out the status of port 2
- What is a precondition for this ?
- Pin 31 = 1 (if pin 31 = 0, it will allow external memory)
- [ **1 1 1 0 0 1 0 1** ]

## 8) Port 3 and Register P3— B0 H

P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
RD	WR	T1	T0	INT1	INT0	TXD	RXD	B0H
B7	B6	B5	B4	B3	B2	B1	B0	

- **Port 3** is also dual purpose but designers generally avoid using this port for I/O because the pins have alternate functions which are related to special features of the 8051. Use of these pins may interfere with the normal operation of the 8051.
- **As an I/O port:** Standard quasi-bidirectional.
- **Alternate functions:**
  - Serial I/O - TXD, RXD
  - Timer clocks- T0, T1
  - Interrupts - INT0, INT1
  - Data memory- RD, WR

**Register is both bit and byte addressable.**

# Timer and Counter : comparison

Timer	Counter
1) The register incremented for every machine cycle.	1) The register is incremented considering 1 to 0 transition at its corresponding external input pin (T0, T1).
2) Maximum count rate is $1/12$ of the oscillator frequency.	2) Maximum count rate is $1/24$ of the oscillator frequency.
3) A timer uses the frequency of the internal clock and generates delay.	3) A counter uses an external signal to count pulses.



**12 pulses of Osc.  
= 1 Machine cycle**

Counter has to perform 2 duties --- 1) Count 2) Store

## 9-10) TCON and TMOD

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0	
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	<b>088H</b>
8F	8E	8D	8C	8B	8A	89	88	

- Two 16-bit up counters are named as T0 and T1.
- Each Timer - programmed to count internal clock pulses.
- Each Counter - programmed to count external pulses.
- *Divided into two 8-bit registers - TL0 (8AH) and TH0 (8CH) -- Location in SFR area*  
*- TL1 (8BH) and TH1 (8DH) -- Location in SFR area*
- T/C are controlled by
  - Timer **MOD**e control register (**TMOD**)
  - Timer/counter **CON**trol register (**TCON**)

## 9) TCON (Timer Control Register): 88H

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0	
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	<b>088H</b>
8F	8E	8D	8C	8B	8A	89	88	

TF1, TF0 .7 .5	TR1, TR0 .6 .4	Ext. interrupt flags IE1, IE0 .3 .1	External Interrupt Type IT1, IT0 .2 .0
Overflow flags for Timer 1 and Timer 0.	Run control bits for Timer1 and Timer0.  Set to Run Reset to Hold i.e. Halt (not stop)	Edge flag for external interrupts 1 and 0.  Set by interrupt edge, Cleared when interrupt is processed.	<b>Type of Interrupts.</b> Reset for 0 level interrupts. Set for falling edge interrupts by software. <b>When IT0 = 1 or IT1 = 1</b> It specifies interrupt on falling edge and <b>when IT0 = 0 or IT1 = 0</b> it specifies low level interrupt .

## 10) TMOD (Timer Mode) Register: 089H

Gate	C/T	M1	M0	Gate	C/T	M1	M0	089H
------	-----	----	----	------	-----	----	----	------

Timer 1

Timer 0

**Gate** – Decides whether the timer starts with external interrupt or with some instructions.

**C/T** – Counter/Timer select bit.

**M1** – Mode bit 1.

**M0** – Mode bit 0.

# 10) TMOD (Timer Mode) Register: 089H

GATE	C/T	M1	M0	GATE	C/T	M1	M0	089H
------	-----	----	----	------	-----	----	----	------

## Timer 1

## Timer 0

**Gate** - Decides whether the timer starts with external interrupt or with instructions.

### **GATE = 0**

The start and stop of a timer is controlled by software using the instruction ....

**SETB TR1** and **CLR TR1** for timer 1 and

**SETB TR0** and **CLR TR0** for timer 0.

The SETB instruction is used to start timer and CLR instruction is used to stop timer.

### **GATE= 1**

Timers can be started and stopped by an external source like an interrupt.

# 10) TMOD (Timer Mode) Register: 089H

GATE	C/T	M1	M0	GATE	C/T	M1	M0	089H
------	-----	----	----	------	-----	----	----	------

## Timer 1

## Timer 0

### C/T (CLOCK / TIMER)

This bit in the TMOD register is used to decide whether a timer is used as a **delay generator (timer)** or an **event manager (counter)**.

If **C/T = 0**, it is a **Timer** for timer delay generation.

If **C/T =1**, it is a **Counter** for counting pulses.

The clock source to create the time delay is the crystal frequency of the 8051 (12MHz)

Timer frequency is always 1/12th of the frequency of the crystal attached to the 8051.



# 10) TMOD (Timer Mode) Register: 089H

GATE	C/T	M1	M0	GATE	C/T	M1	M0
------	-----	----	----	------	-----	----	----

089H

Timer 1

Timer 0

M1	M0	Mode (for both timers)
0	0	13-bit timer mode
0	1	16- bit timer mode
1	0	8-bit auto reload mode
1	1	Split mode

# 10) TMOD (Timer Mode) Register: 089H

## Mode 0 (13-Bit Timer Mode)

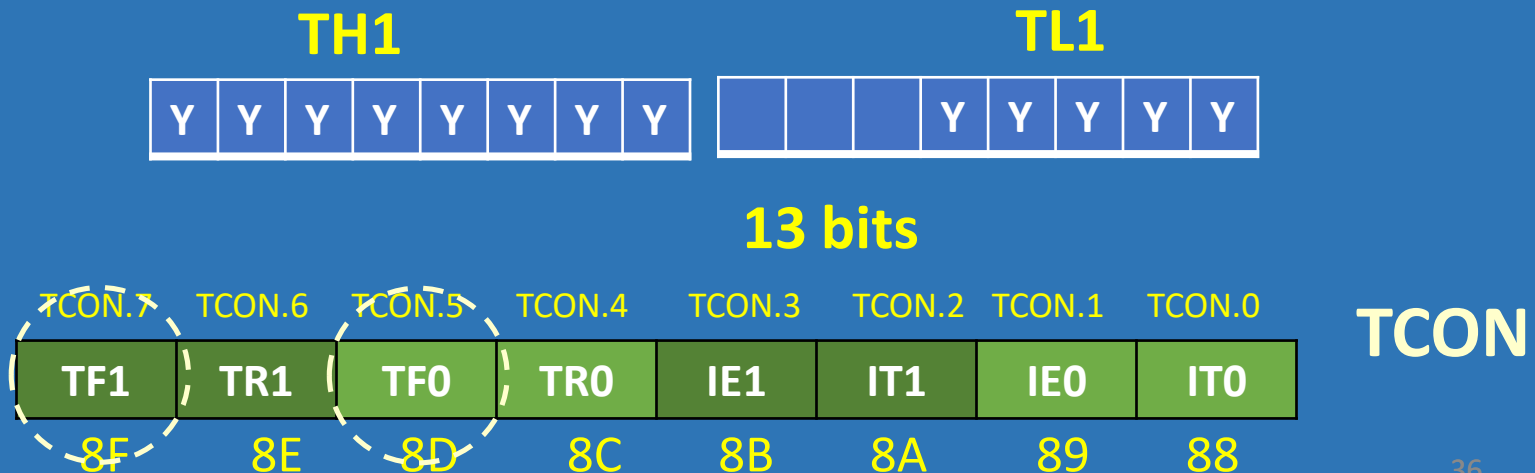
Both Timer 1 and Timer 0 in Mode 0 operate as 13-bit counters.  $2^{13} = 8192$

Timer register has 13-bits 1) 8 bits of TH1 and 2) lower 5 bits of TL1.

The upper 3 bits of TL1 are ignored.

For every 32 counts on TL1, TH1 is incremented by 1.

When TH1 overflows, TF1 in **TCON** is set i.e. = .7 bit



# 10) TMOD—Different modes of timers

Gate	C/T	M1	M0	Gate	C/T	M1	M0	089H
------	-----	----	----	------	-----	----	----	------

## Timer 1

## Timer 0

### Mode 1 (16-Bit Timer Mode)

Timer mode "1" is a 16-bit timer (commonly used mode).  
All 16 bits are used.  $2^{16} = 65536$

As being a full 16-bit timer, the timer has up to 65536 values in it.  
It will overflow back to 0 after 65,536 machine cycles.

## TH1

Y	Y	Y	Y	Y	Y	Y	Y
---	---	---	---	---	---	---	---

## TL1

Y	Y	Y	Y	Y	Y	Y	Y
---	---	---	---	---	---	---	---

## 10) TMOD—Different modes of timers

Gate	C/T	M1	M0	Gate	C/T	M1	M0	089H
------	-----	----	----	------	-----	----	----	------

### Timer 1

### Timer 0

#### Mode 2 (8 Bit Auto Reload)

1. Timer registers are configured as 8-bit timers with automatic reload. It allows only 00 to FF to be loaded into timer register TH1.
2. After TH1 is loaded with 8 bit value, controller gives a copy of it to TL1. Timing is then started by instruction SETB TR1.
3. Timer starts to count up by incrementing TL1 until it reaches FF and rolls over to 00. When it rolls over, it sets TF1 (timer flag).
4. TL1 is reloaded with initial count stored in TH1. Clears TF1 and process will start again.

## 10) TMOD—Different modes of timers

### Mode 3 (Split Timer Mode)

Timer mode "3" is known as **split-timer mode**. When Timer 0 is placed in mode 3, it becomes two separate 8-bit timers. Timer 0 is TL0 and Timer 1 is TH0.

Both the timers count from 0 to 255 and in case of overflow, reset to 0. All the bits that are of Timer 1 will now be tied to TH0.

TH0 = Timer 1



TL0 = Timer 0



TMOD serves only above 2 Split Timers  
and not real T1.

Thus control on real T1 is lost.

# 11, 12, 13 and 14) Timer counter registers

TH 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	08CH
TL1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	08AH
TH0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	08DH
TL0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	08BH

# Timers / Counters : Application

## 1) Interval timing: Timer mode

The timer is programmed to overflow at a regular interval and set the timer overflow flag. Overflow means reaching maximum count, say for e.g. FFFFh. (The maximum count depends on mode of operation, i.e. Mode 1, Mode 2 etc.)

## 2) Event counting: Counter mode

Determine the number of occurrences of an event. An event is any external signal that provides a 1-to-0 transition (falling edge) on a pin of the microcontroller.

# Interrupt System

Interrupt is the event that temporarily suspends the main program, pass the control to the interrupt service routine (ISR) and executes the task. After the routine is executed, the control goes back to the main program where from it had left.

8051 has 5 interrupts. Some manufacturers also state **Reset** as one of the interrupts.

## **Interrupt Sources (in order of priority):      ROM location (Interrupt Vector table)**

1	Reset	0000 H
2	External hardware Interrupt 0 INTO (IE0)	0003 H
3	Timer 0 interrupt (TF0)	000B H
4	External hardware Interrupt 1 INT1 (IE1)	0013 H
5	Timer 1 interrupt (TF1)	001B H
6	Serial COM Port interrupt (RI / TI)	0023 H

Each interrupt type has a separate vector address in ROM.

So the programmer generally writes the program from 0030 H onwards, not disturbing the Interrupt Vector Table (IVT).



## 15) IE Interrupt Enable register: 0A8H

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0	
EA	--	--	ES	ET1	EX1	ET0	EX0	0A8H
--	--	--	AC	AB	AA	A9	A8	

**EX0** : External interrupt 0.

**ET0** : Timer 0 overflow interrupt.

**EX1** : External interrupt 1.

**ET1** : Timer 1 overflow interrupt.

**ES** : Enable serial port interrupt

**EA** : If EA = 0 disables all interrupts, no interrupt is acknowledged.  
If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

0 = Disabled.

1 = Enabled.

**Reset:** Reset cannot be disabled using IE (non-maskable)

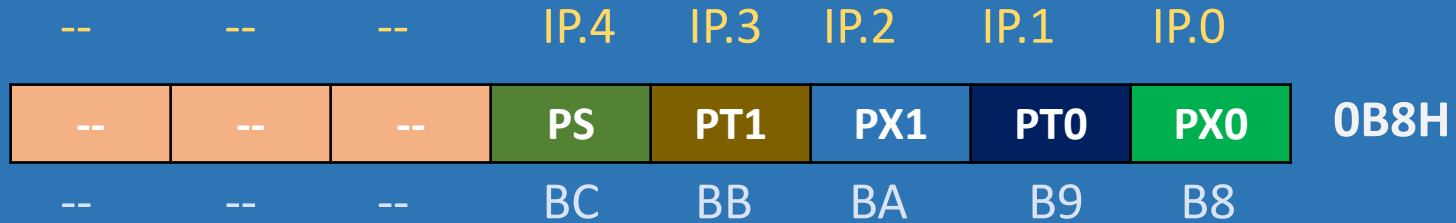
Find status of interrupts if .... 1) MOV IE, #09Ch 2) MOV IE, #0EAh 3) MOV IE, #01Fh

1001 1100

1110 1010

0001 1111

## 16) IP Interrupt Priority Register-- 0B8H



The 8051 starts execution at 0000H after Reset.

PC (program counter) goes to 0000H. Reset has Highest priority – non Maskable.

**PX0 : External interrupt 0.**

**PT0 : Timer 0.**

**PX1 : External interrupt 1.**

**PT1 : Timer 1.**

**PS : Serial interface.**

- A low priority interrupt can only be interrupted by the high priority interrupt, but not interrupted by another low priority interrupt.

- If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served.

- If the requests of the same priority levels are received simultaneously, then the internal polling ('Q') sequence determines which request is to be serviced.

**0 = assigns Low priority**

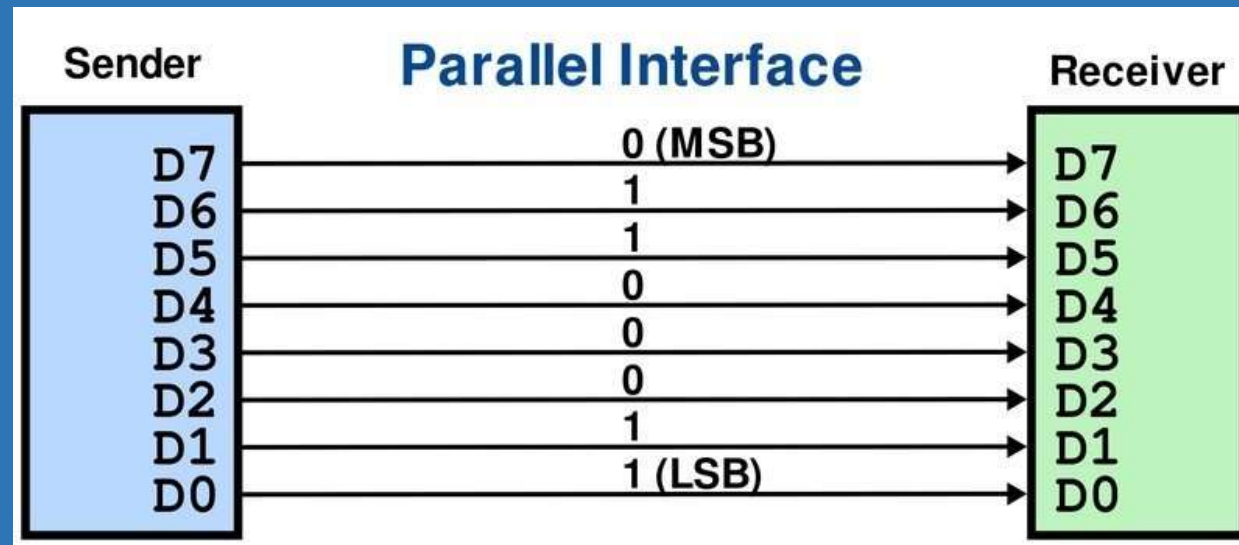
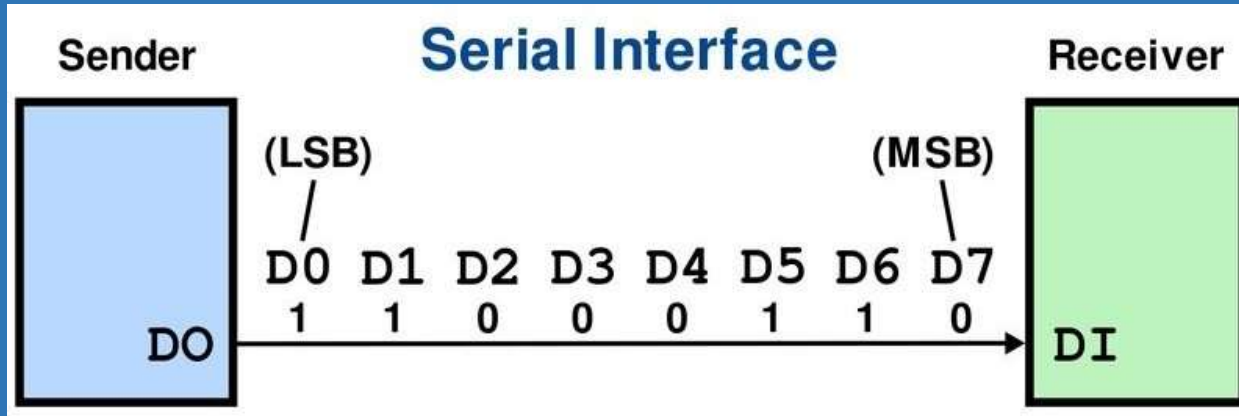
**1 = assigns High priority**



# Serial communication in 8051 : A quick glance.

- There is always a need to connect two or more computers / MuCs etc.
  - The method used to connect them is called as LAN.
  - The communication between two MuCs is done using Serial communication.
  - Using Serial communication a MuC can connect to other devices also.
- 
- The Serial communication, the data is sent bit by bit.
  - The data has to be sent at a particular rate called as Baud rate.
  - Baud rate means bits per second.
  - Baud rates can be 1200, 2400, 4800, 19200, 38400, 57600, and 115200.
  - Most commonly used rate is 9600 bps .
- 
- The Serial communication is done using UART.
  - UART = Universal Asynchronous Receiver Transmitter.
  - 8051 has one serial port which is full duplex. (R and T at the same time.)

# Serial communication in 8051 : A quick glance.





## Serial communication in 8051 : A quick glance.

- Pins 3.0 for and 3.1 from port 3 are used for Serial communication.
  - Pin 3.0 for RXD - Receiving and Pin 3.1 for TXD - Transmitting.
  - Pin no. 10 and 11 resply.
- 
- Two registers to control the Serial communication are used.
  - SCON – to control Mode, Baud rate, Transmit or Receive etc.
  - SBUF – to hold the data during Transmission or Receiving.

## 17) SBUF : Serial buffer Register

bit7	bit6	bit5	bit4	bit3	RB2	bit01	bit0
------	------	------	------	------	-----	-------	------

99 H Byte  
address

**SBUF - 8 bit registers used for serial communication in 8051.**

The byte to be transmitted via TXD line, is first placed in SBUF register.

The byte to be received via RXD line, is actually received in SBUF register.

Along with the byte, it also has a 1) start bit, 2) stop bit and 3) 9<sup>th</sup> bit  
Thus, total 11 bits are transported.

(9<sup>th</sup> bit indicates whether the byte is a data or an address – in mode 2 and 3)

# 18) SCON : Serial Control Register: 98 H

SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0	98 H Byte address
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
9F	9E	9D	9C	9B	9A	99	98	

**2 bits to decide the  
mode of serial  
communication**

SM0	SM1	Serial Mode	
0	0	Mode 0	8 bit Shift register mode (Synchronous type)
0	1	Mode 1	8-bit UART mode with variable baud rate. (Asynchronous type)
1	0	Mode 2	9-bit UART with fixed baud rate. (Asynchronous type)
1	1	Mode 3	9-bit UART with variable baud rate. (Asynchronous type)

# 18) SCON : Serial Control Register: 98 H

SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0	98 H Byte address
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
9F	9E	9D	9C	9B	9A	99	98	

Enables the multiprocessor communication feature in Mode 2 & Mode 3

SM2 = 0 disables multi processor feature  
SM2 = 1 enables it

As 8051 has only one serial port, thus, SM2 = 0  
(8052 has 2 serial ports)



# 18) SCON : Serial Control Register

SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0	98 H Byte address
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
9F	9E	9D	9C	9B	9A	99	98	

## REN: (Receive Enable) SCON.4

**REN = 1 = allows to receive data on the RxD pin.**

**When REN=0, the receiver is disabled.**

(Use SETB or CLR for this)

# 18) SCON : Serial Control Register

SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0	98 H Byte address
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
9F	9E	9D	9C	9B	9A	99	98	

## TB8 (useful in mode 2 and 3 only)

For multiprocessor mode only  
9<sup>th</sup> bit transmitted

**TB8 = 1 it is Data Byte**

**TB8 = 0 it is Address Byte**

**Not useful for mode 0 and mode 1**

## RB8 (useful in mode 2 and 3 only)

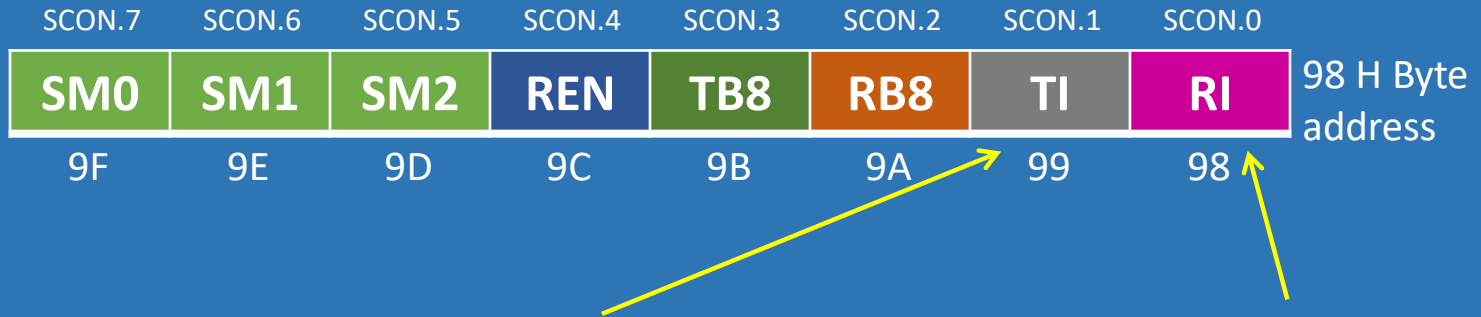
For multiprocessor mode only  
9<sup>th</sup> bit received

**TB8 = 1 it is Data Byte**

**TB8 = 0 it is Address Byte**

**Not useful for mode 0 and mode 1**

# 18) SCON : Serial Control Register



## TI = Transmit Interrupt Flag

### Transmit interrupt flag

When 8051 finishes the 8 bit data transmission , TI flag = 1 and now is ready to transfer next byte.

(Pin no. 11 Pin 3.1 TXD)

## RI = Receive Interrupt Flag

### Receive interrupt flag.

When 8051 finishes the 8 bit data reception , RI flag = 1 and now is ready to receive next byte.

(Pin no. 10 Pin 3.0 RXD)

# 8051 Power down and Idle mode

To **save power** - Power down and Idle mode features are used.

- 8051 has two power-saving modes.
  - Power Down Mode – More saving
  - Idle Mode
- **Power saving modes are set using PCON SFR**

8051 Family Controllers	Operating Oscillator Frequency Fosc	Current required in Normal mode	Current required in Idle mode	Current required in Power Down mode
AT89S51	12 MHz	25 mA	6.5 mA	50 uA
P89V51RD2	12 MHz	11.5 mA	8.5 mA	80-90 uA

## 19) PCON : Power control register : 87H



### Bit 0 – IDL

**1 = Enable Idle mode.**

CPU clock is turned off whereas internal peripheral module such as a timer, serial port, interrupts works normally.

This mode can be cancelled by Interrupt or H/W reset.

**0 = Disable Idle mode.**

## 19) PCON : Power control register : 87H



### Bit 1 – PD: Power down

**1 = Enable Power-Down mode.**

In this mode, the Oscillator clock turned OFF and both CPU and peripherals clock stopped.

Hardware reset cancels this mode.

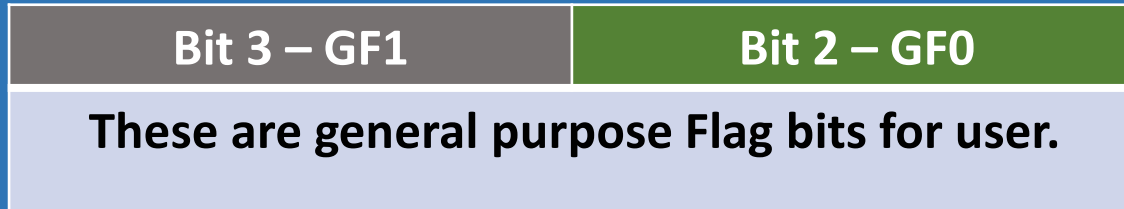
**Voltage reduces from 5 V to 2 V.**

**0 = Disable Power-down mode.**

## 19) PCON : Power control register : 87H



87 H Byte  
address



## 19) PCON : Power control register : 87H



### Bit 7 – SMOD

**Serial Modify bit**

**0** = Baud rate remains fixed

**1** = Baud rate is doubled in UART mode 1 and 3



## 20) and 21) DPTR Register Data Pointer : 82H and 83H

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	82 H
------	------	------	------	------	------	------	------	------

**DPL** Lower byte of data pointer

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	83 H
------	------	------	------	------	------	------	------	------

**DPH** Higher byte of data pointer

60

### **DPTR is meant for pointing to data.**

The Data Pointer (DPTR) is the 8051's only user-accessible 16-bit (2-byte) register.

The Accumulator, R0–R7 registers and B register are 1-byte value registers.

**It is used by the 8051 to access external memory using the address indicated by DPTR. DPTR is the only 16-bit register available and is often used to store 2-byte values.**

Address	Contents of memory
0x0005	1011 1110 1000 1111 00
0x0004	1011 1111 0111 0101 01
0x0003	1011 1111 0100 0001 10
0x0002	0011 1110 0001 0000 10
0x0001	0011 1111 0110 1000 10
0x0000	0011 1111 0110 1000 10

– Addresses of the 21 SFRs :-

- 8A – TL0
- 8B – TL1
- 8C – TH0
- 8D – TH1

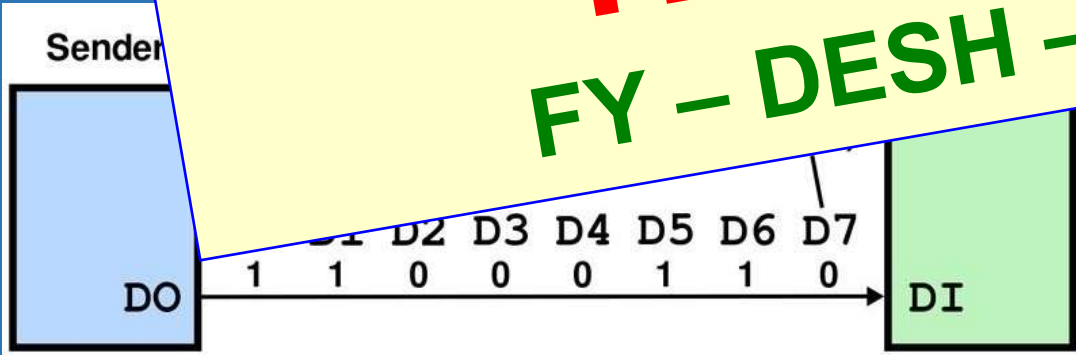
- 80 – P0\* (Port 0)
- 90 – P1\* (Port 1)

# Micro Controllers – 2.3

## 8051 continued ...

### Special Function Registers (SFRs)

**Thanks !**  
**FY – DESH – VIT**



1E	1D	1C	1B	1A	19	18
16	15	14	13	12	11	10
0E	0D	0C	0B	0A	09	08
06	05	04	03	02	01	00