

ELECTRONICS HUB

PROJECTS | TUTORIALS | COURSES | KITS

[HOME](#)[PROJECTS](#)[MINI PROJECTS](#)[ARDUINO](#)[FREE CIRCUITS](#)[TUTORIALS](#)[SYMBOLS](#)[DIY](#)[COURSES](#)[CONTACT US](#)

YOU ARE HERE: [HOME](#) / [GENERAL](#) / 8051 MICROCONTROLLER SPECIAL FUNCTION REGISTERS (SFRS)

8051 Microcontroller Special Function Registers (SFRs)

NOVEMBER 30, 2017 BY [RAVI](#) — [3 COMMENTS](#)

In the previous 8051 Microcontroller Tutorial, we have seen the Instruction Set and Addressing Modes. In this tutorial, we will take a look at the 8051 Microcontroller Special Function Registers or SFRs.

If you remember from the [8051 Microcontroller Memory Organization](#) Tutorial, the internal RAM or Data Memory of the 8051 Microcontroller is divided in to General Purpose Registers, Bit Addressable Registers, Register Banks and Special Function Registers or SFRs.

The 8051 Microcontroller Special Function Registers are used to program and control different hardware peripherals like Timers, Serial Port, I/O Ports etc. In fact, by manipulating the 8051 Microcontroller Special Function Registers (SFRs), you can assess or change the operating mode of the 8051 Microcontroller.

As a reminder, the following image shows you the basic structure of 8051 Microcontroller's Internal RAM.

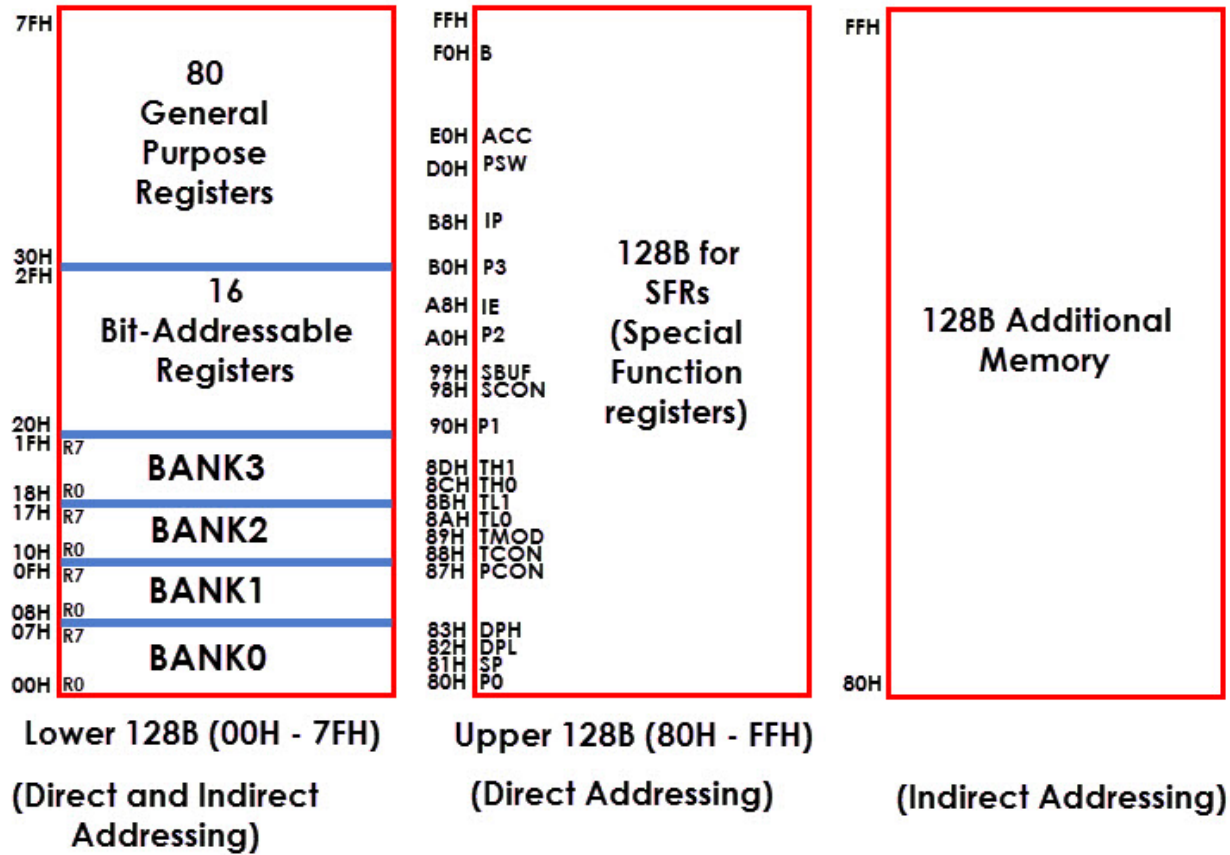


Table of Contents

1. 8051 Microcontroller Special Function Registers (SFRs)
 - 1.1. List of 8051 Microcontroller Special Function Registers
 - 1.2. Categories of 8051 Microcontroller Special Function Registers
2. CPU or Math Registers
 - 2.1. A or Accumulator (ACC)
 - 2.2. B (Register B)
3. Program Status Word (PSW)
4. Pointer Registers
 - 4.1. Data Pointer (DPTR – DPL and DPH)
 - 4.2. Stack Pointer (SP)
5. I/O Port Registers (P0, P1, P2 and P3)
6. Peripheral Control Registers
 - 6.1. PCON (Power Control)
 - 6.2. SCON (Serial Control)
 - 6.3. TCON (Timer Control)
 - 6.4. TMOD (Timer Mode)
 - 6.5. IE (Interrupt Enable)
 - 6.6. IP (Interrupt Priority)
7. Peripheral Data Registers
 - 7.1. SBUF (Serial Data Buffer)
 - 7.2. TL0/TH0 (Timer 0 Low/High)
 - 7.3. TL1/TH1 (Timer 1 Low/High)

8051 Microcontroller Special Function Registers (SFRs)

The 8051 Microcontroller Special Function Registers act as a control table that monitor and control the operation of the 8051 Microcontroller. If you observe in Internal RAM Structure, the Address Space from 80H to FFH is allocated to SFRs.

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.

TIP: As only 21 of the possible 128 SFR memory locations are assigned, it is recommended that the remaining registers or memory locations are not accessed during programming.

“ Before going further, get an idea on [8051 MICROCONTROLLER ARCHITECTURE](https://www.electronicshub.org/8051-microcontroller-special-function-registers/).

List of 8051 Microcontroller Special Function Registers

- A or ACC
- B
- DPL
- DPH
- IE
- IP
- P0
- P1
- P2
- P3
- PCON
- PSW
- SCON
- SBUF
- SP
- TMOD
- TCON
- TLo
- THo

- TL1
- TH1

Categories of 8051 Microcontroller Special Function Registers

All the 21 8051 Microcontroller Special Function Registers (SFRs) along with their functions and Internal RAM Address is given in the following table.

<i>Name of the Register</i>	<i>Function</i>	<i>Internal RAM Address (HEX)</i>
ACC	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

There are many ways to categorize these 21 Special Function Registers but I find the following way as an appropriate one. The 21 Special Function Registers of 8051 Microcontroller are categorized in to seven groups. They are:

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

CPU or Math Registers

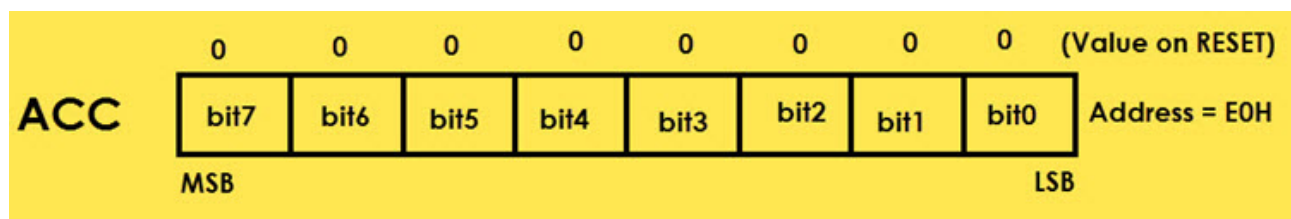
A or Accumulator (ACC)

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)

The name "Accumulator" came from the fact this register is used to accumulate (or store) the result of all Arithmetic and most of the Logical Operations.

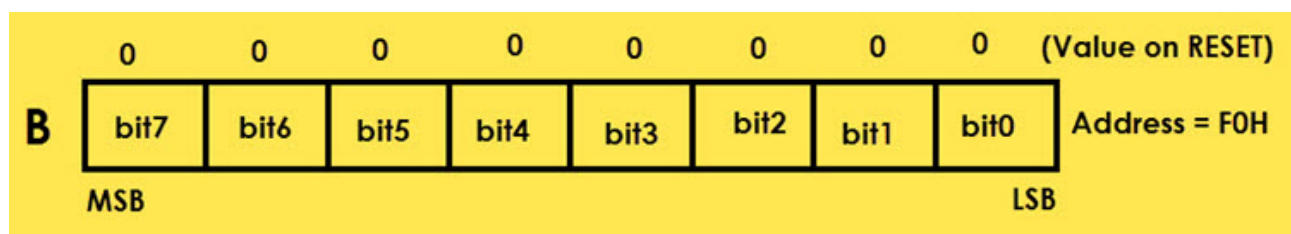


B (Register B)

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

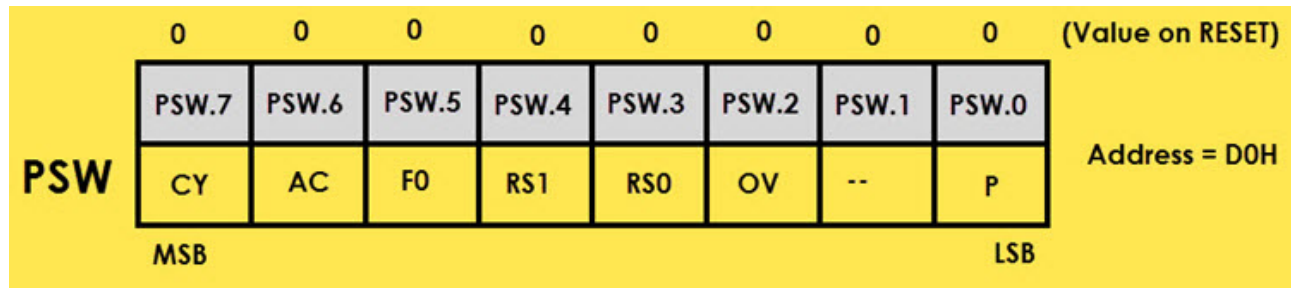
Register B is located at the address F0H of the SFR Address Space.



Program Status Word (PSW)

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. The following image shows the contents of the PSW Register.



The following table describes the function of each flag.

BIT	SYMBOL	FLAG NAME			DESCRIPTION
7	C or CY	Carry			Used in Arithmetic, Logic & Boolean Operations
6	AC	Auxiliary Carry			Used in BCD Arithmetic
5	F0	Flag 0			General Purpose User Flag
4	RS1	Register Bank Selection Bit 1			
3	RS0	Register Bank Selection Bit 0			
		RS1	RS0	Bank	
		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 # of 1's; otherwise Reset

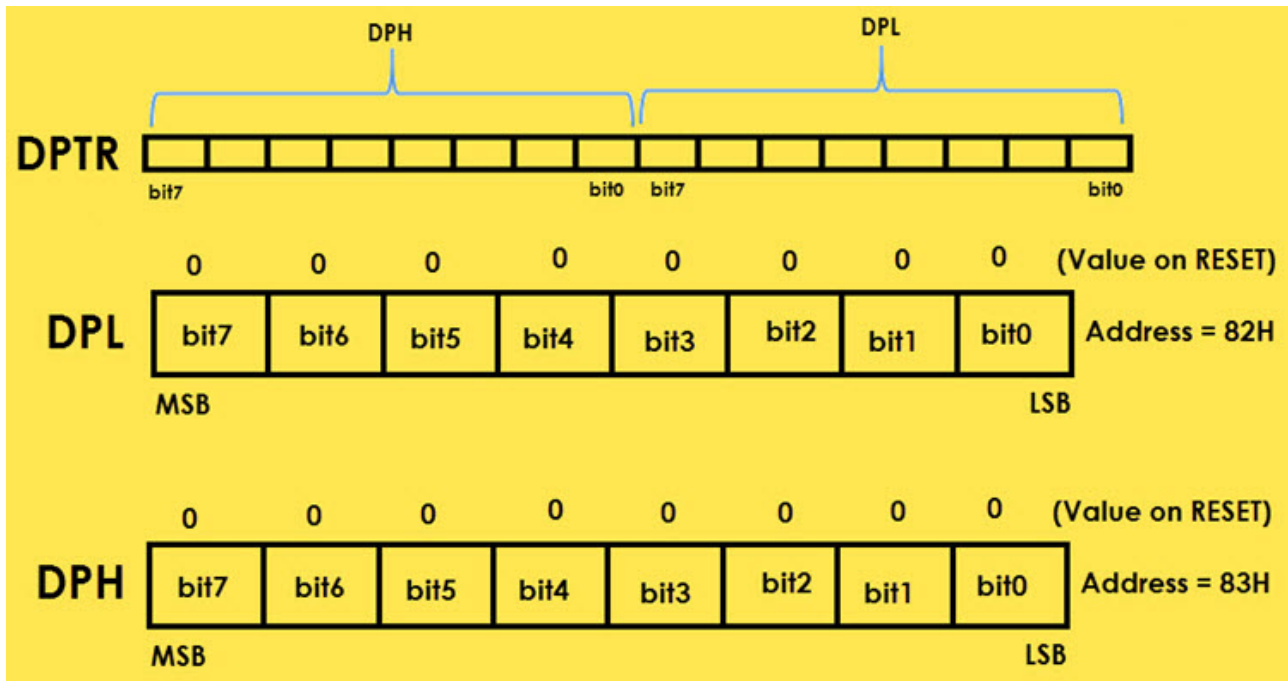
Pointer Registers

Data Pointer (DPTR – DPL and DPH)

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.

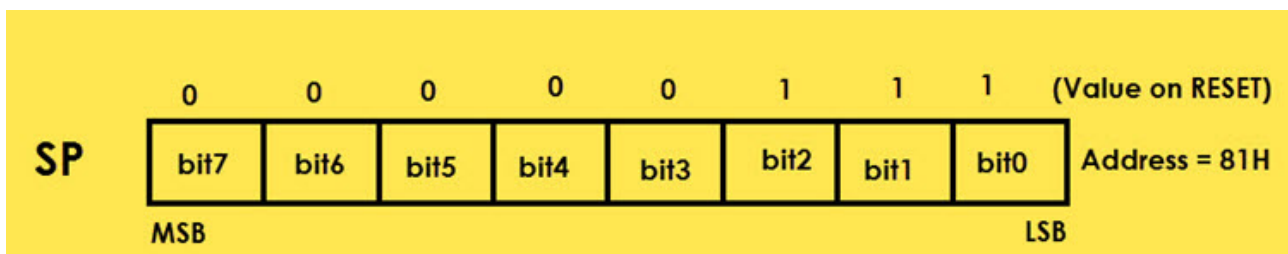
The DPTR Register is used by the programmer addressing external memory (Program – ROM or Data – RAM).



Stack Pointer (SP)

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).



I/O Port Registers (P0, P1, P2 and P3)

The 8051 Microcontroller four Ports which can be used as Input and/or Output. These four ports are P0, P1, P2 and P3. Each Port has a corresponding register with same names (the Port Registers are also P0, P1, P2 and P3). The addresses of the Port Registers are as follows: P0 – 80H, P1 – 90H, P2 – A0H and P3 – B0H.

Each bit in these SFRs corresponds to one physical Pin in the 8051 Microcontroller. All these Port Registers are both Bit Addressable and Byte Addressable. Writing 1 or 0 on a Port Register Bit will reflect as an appropriate voltage (5V and 0V) on the corresponding Pin.

If a Port Bit is SET (declared as 1), the corresponding Port Pin will be configured as Input and similarly if a Port Bit is CLEARED (declared as 0), the corresponding Port Pin is configured as Output. Upon reset, all the Port Bits are SET (1) and hence, all the Port Pins are configured as Inputs.

	1	1	1	1	1	1	1	(Value on RESET)
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
								Address = 80H
	1	1	1	1	1	1	1	(Value on RESET)
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
								Address = 90H
	1	1	1	1	1	1	1	(Value on RESET)
	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
								Address = A0H
	1	1	1	1	1	1	1	(Value on RESET)
	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
								Address = B0H

Peripheral Control Registers

PCON (Power Control)

The PCON or Power Control register, as the name suggests 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.

Apart from these two, the PCON Register can also be used for few additional purposes. The SMOD Bit in the PCON Register is used to control the Baud Rate of the Serial Port.

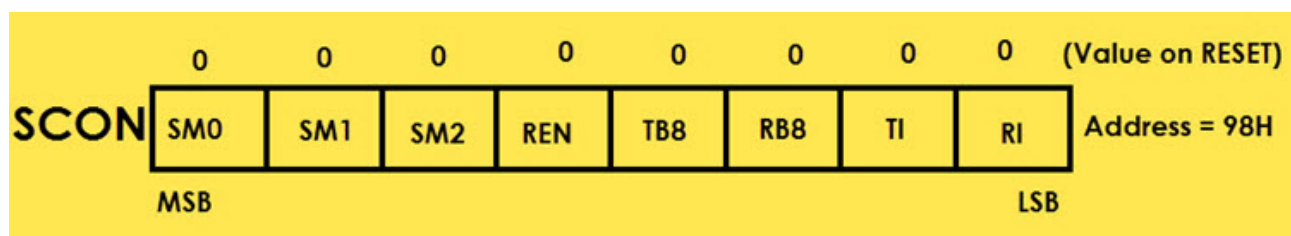
There are two general purpose Flag Bits in the PCON Register, which can be used by the programmer during execution.



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, you can control the Operation Modes of the Serial Port, Baud Rate of the Serial Port and Send or Receive Data using Serial Port.

SCON Register also consists of bits that are automatically SET when a byte of data is transmitted or received.



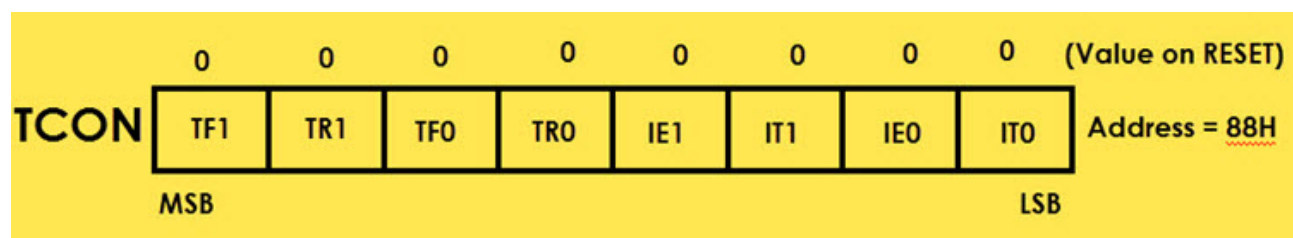
Serial Port Mode Control Bits

SM0	SM1	Mode	Description	Baud Rate

0	0	0	8-Bit Synchronous Shift Register Mode	Fixed Baud Rate (Frequency of oscillator / 12)
0	1	1	8-bit Standard UART mode	Variable Baud Rate (Can be set by Timer 1)
1	0	2	9-bit Multiprocessor Comm. mode	Fixed Baud Rate (Frequency of oscillator / 32 or Frequency of oscillator / 64
1	1	3	9-bit Multiprocessor Comm. mode	Variable Baud Rate (Can be set by Timer 1)

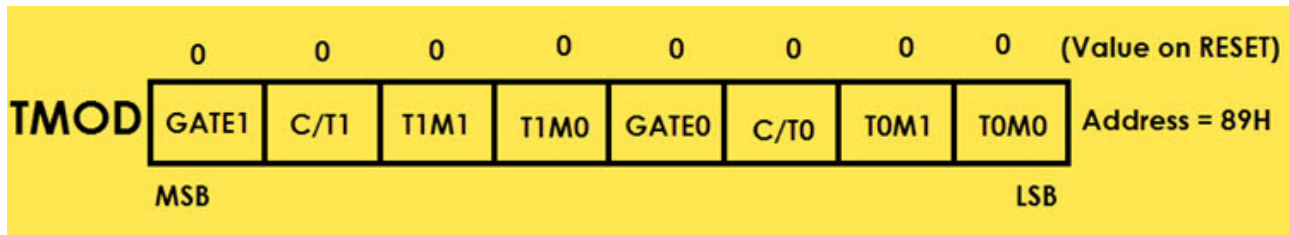
TCON (Timer Control)

Timer Control or TCON Register is used to start or stop the Timers of 8051 Microcontroller. It also contains bits to indicate if the Timers has overflowed. The TCON SFR also consists of Interrupt related bits.



TMOD (Timer Mode)

The TMOD or Timer Mode register or SFR is used to set the Operating Modes of the Timers T0 and T1. The lower four bits are used to configure Timer0 and the higher four bits are used to configure Timer1.



The Gatex bit is used to operate the Timerx with respect to the INTx pin or regardless of the INTx pin.

GATE1 = 1 ==> Timer1 is operated only if INT1 is SET.

GATE1 = 0 ==> Timer1 is operates irrespective of INT1 pin.

GATE0 = 1 ==> Timer0 is operated only if INTO is SET.

GATE0 = 0 ==> Timer0 is operates irrespective of INTO pin.

The C/Tx bit is used selects the source of pulses for the Timer to count.

C/T1 = 1 ==> Timer1 counts pulses from Pin T1 (P3.5) (Counter Mode)

C/T1 = 0 ==> Timer1 counts pulses from internal oscillator (Timer Mode)

C/T0 = 1 ==> Timer0 counts pulses from Pin T0 (P3.4) (Counter Mode)

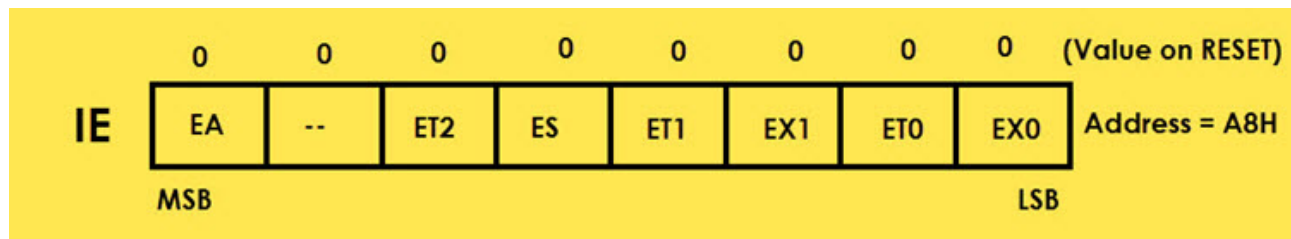
C/T0 = 0 ==> Timer0 counts pulses from internal oscillator (Timer Mode)

TxM0	TxM1	Mode	Description
0	0	0	13-bit Timer Mode (THx – 8-bit and TLx – 5-bit)
0	1	1	16-bit Timer Mode
1	0	2	8-bit Auto Reload Timer Mode
1	1	3	Two 8-bit Timer Mode or Split Timer Mode

NOTE: x = 0 for Timer 0 and x = 1 for Timer 1.

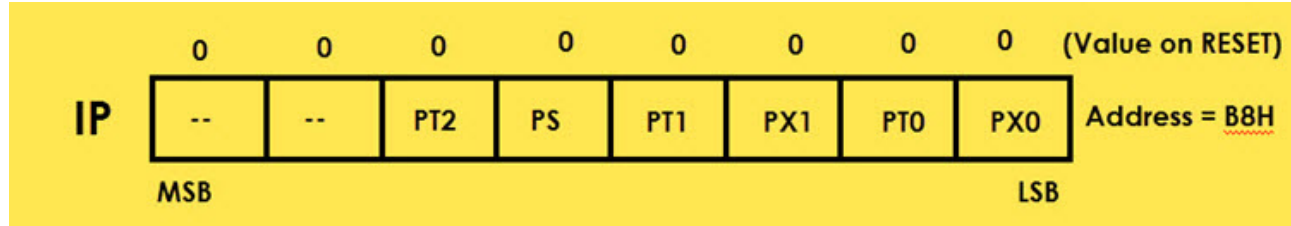
IE (Interrupt Enable)

The IE or Interrupt Enable Register is used to enable or disable individual interrupts. If a bit is SET, the corresponding interrupt is enabled and if the bit is cleared, the interrupt is disabled. The Bit7 of the IE register i.e. EA bit is used to enable or disable all the interrupts.



IP (Interrupt Priority)

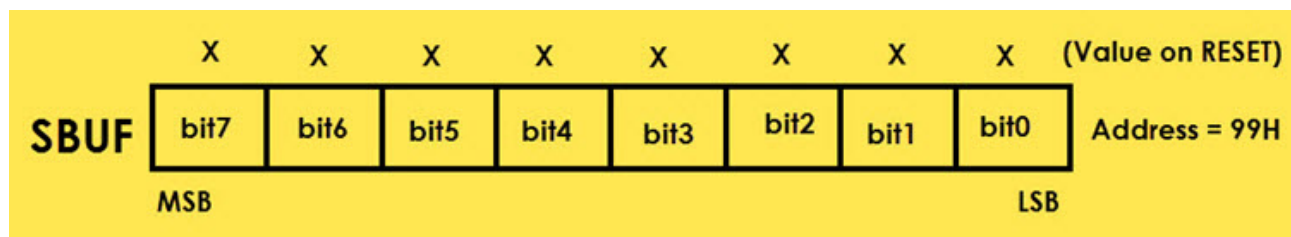
The IP or Interrupt Priority Register is used to set the priority of the interrupt as High or Low. If a bit is CLEARED, the corresponding interrupt is assigned low priority and if the bit is SET, the interrupt is assigned high priority.



Peripheral Data Registers

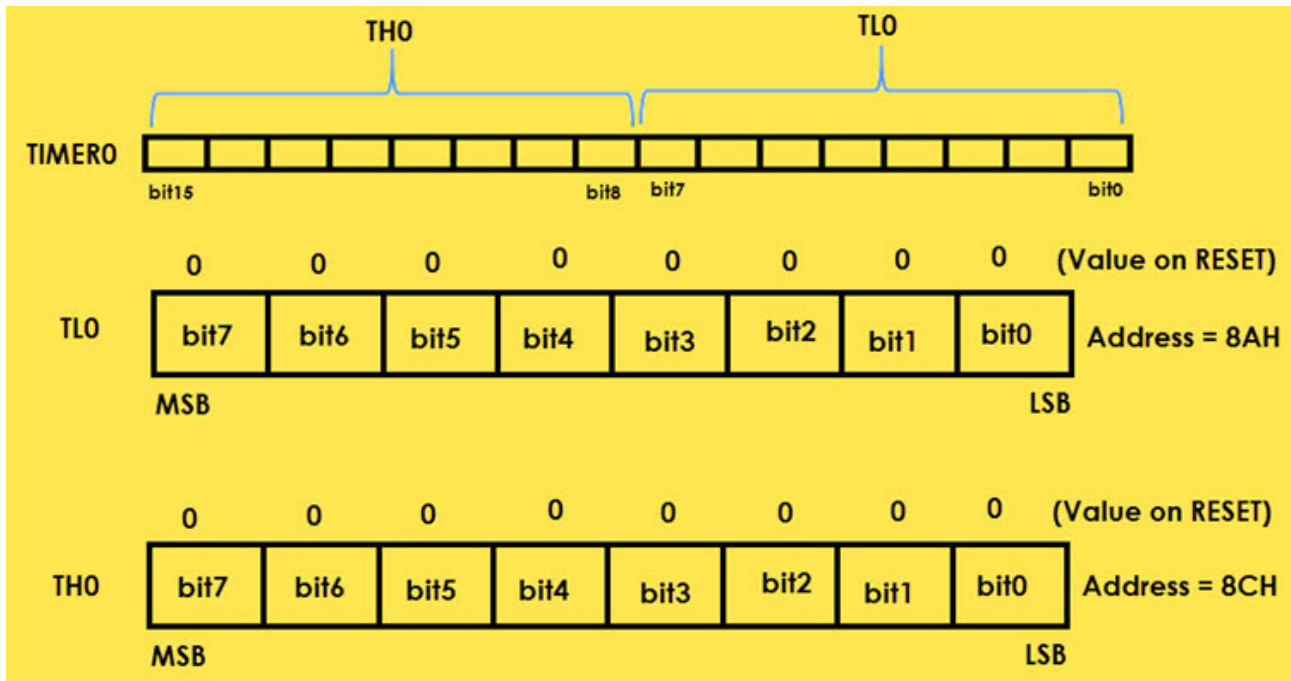
SBUF (Serial Data Buffer)

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



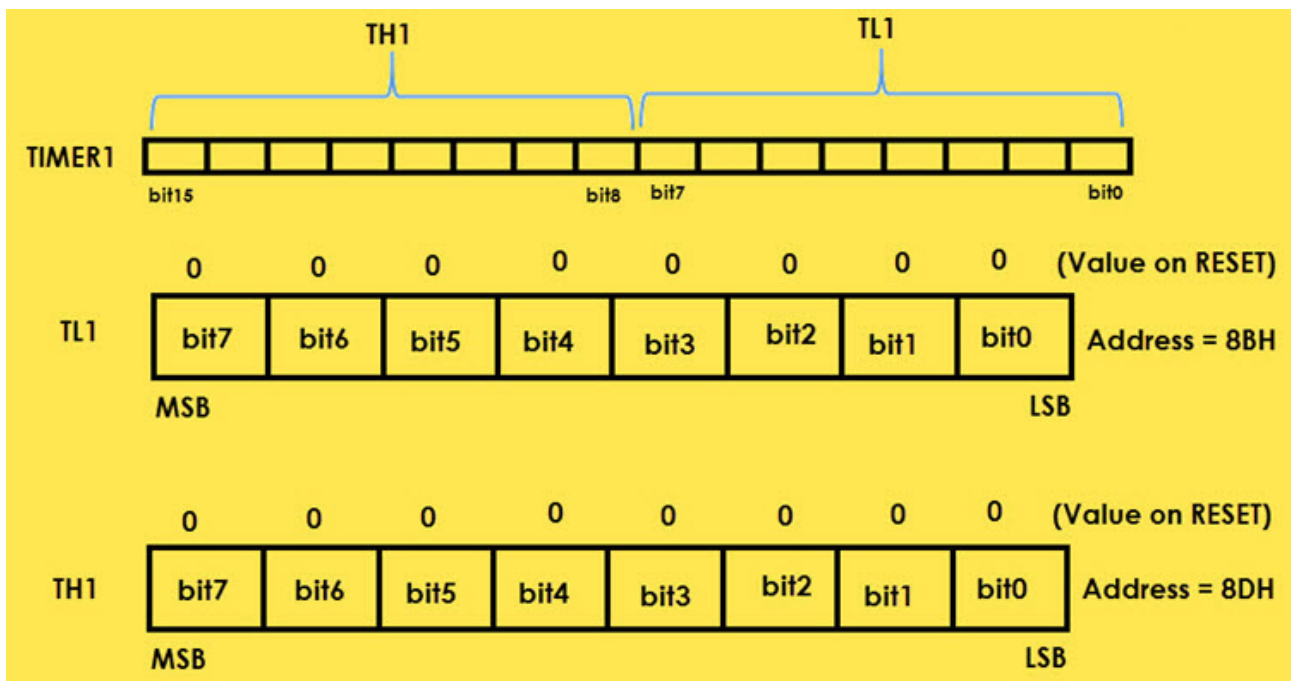
TLo/THo (Timer 0 Low/High)

The Timer 0 consists of two SFRs: TLO and TH0. The TLO is the lower byte and the TH0 is the higher byte and together they form a 16-bit Timer0 Register.



TL1/TH1 (Timer 1 Low/High)

The TL1 and TH1 are the lower and higher bytes of the Timer 1.



In this tutorial, we have seen the 8051 Microcontroller Special Function Registers (SFRs), their addresses, structures, reset values etc.

FILED UNDER: [8051 MICROCONTROLLER, GENERAL](#)

Comments



Madhuri says

APRIL 13, 2018 AT 11:20 PM

Thank you for simple explain i easy to understand

[Reply](#)



surya says

JUNE 16, 2018 AT 1:17 PM

Explanation in usage of each bits can be helpful to understand deeper, please add those

[Reply](#)



Ashish Namdeo says

DECEMBER 11, 2018 AT 5:52 PM

Many thanks to u for making it understandable.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

PROJECTS BY CATEGORY

[Arduino Projects](#) (200+)

[Electronics Projects](#) (250+)

[Mini Project Circuits](#) (160+)

[Mini Project Ideas \(150+\)](#)
[ECE Projects \(150+\)](#)
[EEE Projects \(150+\)](#)
[8051 Projects \(110+\)](#)
[Raspberry Pi Projects \(101+\)](#)
[Electrical Project Ideas \(100+\)](#)
[Embedded Projects \(100+\)](#)
[Latest Electronics Ideas \(100+\)](#)
[Microcontroller Mini Projects \(100+\)](#)
[Robotics Projects \(100+\)](#)
[VLSI Projects \(100+\)](#)
[Solar Projects \(100+\)](#)
[IOT Projects \(100+\)](#)

[Communication Projects \(70+\)](#)
[LED Projects \(70+\)](#)
[Power Electronics Projects \(60+\)](#)
[RFID Projects \(60+\)](#)
[Home Automation Projects \(50+\)](#)
[Matlab Projects \(50+\)](#)
[EIE Projects \(50+\)](#)
[Wireless Projects \(50+\)](#)
[LabView Projects \(45+\)](#)
[Zigbee Projects \(45+\)](#)
[GSM Projects \(40+\)](#)
[555 Timer Circuits \(40+\)](#)
[Sensor Projects \(40+\)](#)
[ARM Projects \(60+\)](#)
[DTMF Projects \(30+\)](#)
[PIC Projects \(30+\)](#)
[Electrical Mini Projects \(25\)](#)
[ESP8266 Projects \(15\)](#)

KITS

[Best Drone Kits \[12\]](#)
[3D Printer Kits \[12\]](#)
[Best Robot Vacuum Clears \[14\]](#)
[Best Waveform Generators \[12\]](#)

[RGB LED Strip Light Kits \[20\]](#)[Best LED Christmas Light Kits \[13\]](#)

SUBSCRIBE FOR UPDATES

Enter your email address:

SUBSCRIBE

GENERAL

[Tutorials](#)
[Symbols](#)
[Courses](#)
[Calculator](#)
[Contact](#)

PROJECTS

[Electrical](#)
[Electronics](#)
[Embedded](#)
[Power](#)
[Robotics](#)
[ARM](#)
[IOT](#)

PROJECTS

[Mini projects](#)
[Microcontroller](#)
[Arduino](#)
[Solar](#)
[Free circuits](#)
[Home Automation](#)
[Seminar Topics](#)
[Electronics Questions](#)

TUTORIALS

[Capacitors](#)
[Resistors](#)
[Filters](#)
[Diodes](#)
[Transistors](#)

TUTORIALS

[Amplifiers](#)
[IO Devices](#)
[Thyristors](#)
[DC Circuits](#)
[Number System](#)

FOLLOW US

[Instagram](#)
[Youtube](#)
[Facebook](#)
[Google Plus](#)
[Twitter](#)

Copyright © 2018 Electronicshub.org