SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

19ECT221 – MICROPROCESSORS AND MICROCONTROLLERS

II YEAR - IV SEM

UNIT 3 – Interrupts of 8051

Interrupts- 8051

- Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task.
- It then passes the control to the main program where it had left off.
- 8051 has 5 interrupt signals, i.e. INT0, TFO, INT1, TF1, RI/TI.
- Each interrupt can be enabled or disabled by setting bits of the IE register and the whole interrupt system can be disabled by clearing the EA bit of the same register.

Interrupt Register

IE (Interrupt Enable) Register

This register is responsible for enabling and disabling the interrupt. EA register is set to one for enabling interrupts and set to 0 for disabling the interrupts. Its bit sequence and their meanings are shown in the following figure.

EA	-	-	ES	ET1	EX1	ET0	EX0

EA	IE.7	It disables all interrupts. When EA = 0 no interrupt will be acknowledged and EA = 1 enables the interrupt individually.
-	IE.6	Reserved for future use.
-	IE.5	Reserved for future use.
ES	IE.4	Enables/disables serial port interrupt.
ET1	IE.3	Enables/disables timer1 overflow interrupt.
EX1	IE.2	Enables/disables external interrupt1.
ET0	IE.1	Enables/disables timer0 overflow interrupt.
EX0	IE.0	Enables/disables external interrupt0.

Interrupt Priority Register

IP (Interrupt Priority) Register

We can change the priority levels of the interrupts by changing the corresponding bit in the Interrupt Priority (IP) register as shown in the following figure.

- 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 sequence determines which request is to be serviced.

-	-	PT2	PS	PT1	PX1	PT0	PX0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	

-	IP.6	Reserved for future use.
-	IP.5	Reserved for future use.
PS	IP.4	It defines the serial port interrupt priority level.
PT1	IP.3	It defines the timer interrupt of 1 priority.
PX1	IP.2	It defines the external interrupt priority level.
PT0	IP.1	It defines the timer0 interrupt priority level.
PX0	IP.0	It defines the external interrupt of 0 priority level.

Interrupt programming of 8051

1. **Timer Interrupt Programming**: In microcontroller Timer 1 and Timer 0 interrupts are generated by time register bits TF0 AND TF1. This timer interrupts programming by C code involves:
    - Selecting the configuration of TMOD register and their mode of operation.
    - Enables the IE registers and corresponding timer bits in it.
    - Choose and load the initial values of TLx and THx by using appropriate mode of operation.
    - Set the timer run bit for starting the timer.
    - Write the subroutine for a timer and clears the value of TRx at the end of the subroutine.

Let's see the timer interrupt programming using Timer0 model for blinking LED using interrupt method:

```
#include< reg51 .h>
sbit Blink Led = P2^0;  // LED is connected to port 2 Zeroth pin
void timer0_ISR (void) interrupt 1  //interrupt no. 1 for Timer0
 {
Blink Led=~Blink Led;   // Blink LED on interrupt
TH0=0xFC;   // loading initial values to timer
TL0=0x66;
}
void main()
{
TMOD=0x0l;      // mode 1 of Timer0
TH0 = 0xFC:    // initial value is loaded to timer
TL0 = 0x66:
ET0 =1;         // enable timer 0 interrupt
TR0 = 1;        // start timer
while (1);      // do nothing
 }
```

Interrupt Programming

2. **External Hardware Interrupt Programming**

Microcontroller 8051 is consisting of two external hardware interrupts: INT0 and INT1 as discussed above. These interrupts are enabled at pin 3.2 and pin 3.3. It can be level triggered or edge triggered. In level triggering, low signal at pin 3.2 enables the interrupt, while at pin 3.2 high to low transition enables the edge triggered interrupt.

Let us see the programmable feature of 8051 microcontroller are:

- Enables the equivalent bit of external interrupt in Interrupt Enable (IE) register.
- If it is level triggering, then write subroutine appropriate to this interrupt, or else enable the bit in TCON register corresponding to the edge triggered interrupt.

Consider the edge triggered external hardware interrupt programming is:-

```
void main()
{
IT0 = 1;    // Configure interrupt 0 for falling edge on INT0
EX0 = 1;    // Enabling the EX0 interrupt
EA =1;      // Enabling the global interrupt flag
}
void ISR_ex0(void) interrupt 0
{
<body of interrupt>
}
```

Interrupt Programming

3. **Serial Communication Interrupt Programming** It is used when there is a need to send or receive data. Since one interrupt bit is used for both Transfer Interrupt (TI) and Receiver Interrupt (RI) flags, Interrupt Service Routine (ISR) must examine these flags for knowing the actual interrupt. By the logical OR operation of RI and TI flags causes the interrupt and it is clear by the software alone. Consider the steps involved in serial communication interrupt programming are:-
    - Configure the Interrupt Enable register for enabling serial interrupt.
    - Configure the SCON register for performing transferring and receiving operation.
    - Write a subroutine for given interrupt with appropriate function.

Let's see the program for sending 'E' through serial port with 9600 baud rate using Serial Interrupt:

```
void main()
{
TMOD = 0x20:
TH1= 0xFD;      // baud rate for 9600 bps
SCON = 0x50;
TR1=1;
EA=I;
whlle(l);
}
void ISR_Serial(void) interrupt 4
{
if(TI==l)
{
SBUF= ?E?;
TI=0;
}
else
RI =0;
}
```

References

https://www.javatpoint.com/embedded-system-interrupts-in-8051-microcontroller

https://www.tutorialspoint.com/microprocessor/microprocessor_intel_8255a_programmable_peripheral_interface.htm

https://www.elprocus.com/types-of-interrupts-in-8051-microcontroller-and-interrupt-programming/