Chapter 10

Programming in C

Lesson 11

C Programming Examples for Interrupts

8051 versions 5 or 6 or 7 standard hardware interrupts

- INT0, T0, INT1, T1, serial port and T2
- Can be numbered as 0, 1, 2, 3, 4, 5, …
- The vector addresses (also the service routine start address in case of 8051) start from 0x0003, 0x000B, 0x0013, … .

8051 versions 5 or 6 or 7 standard hardware interrupt vector addresses

- Also the service routine start address in case of 8051) start from 0x0003, 0x000B, 0x0013, … .
- New versions add more interrupts

Cx51 32 interrupt vector addresses

- Cx51 compiler provides for support up to 32 interrupt numbers
- Numbered as 0, 1, 2, 3, 4, 5, … , 30 or 31
- Vector addresses 0x0003 + interrupt number × 0x0008

Two SFRs IE and IP

- IE for enabling the actions on interrupts
- IP for specifying the priorities over the default priorities
- Lower the interrupt number higher is default priority

Interrupt function attribute

- Interrupt number
- Cx51 compiler enables the specification of the attribute after the name of the function and the word interrupt

The interrupts numbers

- 0 or 1 or … INT0, T0, INT1, T1, serial port and T2 are numbered as 0, 1, 2, 3, 4 and 5
- Interrupt number= 1 for the timer T0 function for the interrupt

Cx51 compiler features

- Enables that each Interrupt function can be assigned a bank to use the registers
- Instructions using registers can thus be short

Cx51 compiler

- If required, the saves the contents of DPH, DPL, ACC, B, and PSW SFRs on the stack when the interrupt function called
- The registers required in the interrupt function, which are used by other functions, are also saved on to stack, if they are not in the bank specified as the interrupt attribute

Advantage for association of register-bank with the interrupt function

- When the specific bank are associated with the specific interrupt number— the Context switching instructions reduced
- Context switches fast
- Push of previous function variables to the stack and then pop from the stack later before return most often not be required

Example Interrupt bank number 3

- Bank 3 assigned for the timer T0 function for the interrupt

Interrupt function declarations

- May not or may include a return value
- The return value specified similar to one in a C function

C program configuring the INT1 interrupt at pin 3 at P3 as falling edge interrupt

- Use of IE and TCON registers
- Assume interrupt function returns the number of falling edges at the pulses at INT1 pin
- Assume interrupt disabled on 1000th falling edge

C Program

```
#include <reg51.h> /* Include header file for the
   registers and SFRs of 8051. */
void main (void)
 {
unsigned int numEdges = 0; /* Declare variable
   numEdges as 16-bit positive integer and assign
   it initial value = 0 */
```

Main Program continued

```
IT1 = 1;   /* Configure interrupt 1 for falling
   edge on INT1 pin P3.3 */
EA = 1;    /* Enable Global interrupt */
EX1 = 1;   /* Enable EX0 interrupt */
  .              /* remaining statement in main. */
while (1) {  /* Wait endlessly */
 ; }             /* End of the while loop */
}               /* End of the main */
```

Interrupt Function

```
unsigned int int1ISR (void) interrupt 1 using 2 {
 if (numEdges < 1000){    /* count INT1
  interrupts from 0 up to 3 */
  numEdges ++; } else
{numEdges = 0; /* Reset Count INT1 interrupts
  from 0 on next falling edge,  */
```

Interrupt Function

```
EX1= 0};   /*  and disable INT1 interrupts on
   1000 falling edge. */
  return (numEdges);   /* Function returns an
   unsigned integer numEdges */
  } /* End of interrupt function for INT1 */
```

Summary

We learnt

- Programming for INT pin Interrupts
- Program for INT1 interrupt at pin 3 at P3 as falling edge interrupt

End of Lesson 11 on

C Programming Examples for Interrupts