Chapter 10

Programming in C

Lesson 08

C Programming Examples for Ports

Using all 8-bits of Port simultaneously

- 8051 four number 8-bit ports, P0, P1, P2, and P3
- *Declared as unsigned character when all the 8-bits to be used then the data type of a variable used for assigning a byte to the port*
- Ports P0, P1, P2, and P3 the SFRs

Preprocessor directive *#include* reg51.h

- Used to include a header-file for the registers (SFRs) in the source file
- Provides the related SFR declarations of the data type, addresses, pointers and operations during the compilation of the source file

Program to write 0x75 at port P0

- #include <reg51.h> /* Include header file for the registers and SFRs of 8051*/
- void main (void) /* main function */
- {unsigned char portByte; /* Declare variable portByte */
- portByte = 0x75; /* portByte variable is assigned the value= 01110101 */
- P0 = portByte;} /* P0 is assigned (written) the value of portByte */

Program to write 0x75 at port P0

- Compile Source file
- Run using a simulator
- Show that P0 become 0x75
- Show that the port P0 bits b7, b6, b5, b4, b3, b2, b1 and b0 show equal to 01110101

Connect LEDs with the port

- Through an interfacing circuit such that when a bit bn = 0, the corresponding LED is OFF and when = 1 then ON
- The 0th , 2nd, 4th, 5th and 6th LEDs become ON and 1st , 3rd and 7th LEDs will be OFF at P1 when the program is run

Program to complement all bits at port P1 Toggle all bits

- #include <reg51.h> /* Include header file for the registers and SFRs of 8051. */
- void main (void)
- { P1 = ~ P1; /* use P1 = ! P1 for the NOT operations on each bit, then the P1 is assigned the value of complement of P1 in new C99 ANSI compliant compiler */
- }

Results

- When the above program in a source file is compiled and run using a simulator, the simulator shows that P1 = NOT (P1). The port P1 bits b7, b6, b5, b4, b3, b2, b1 and b0 equals to complement of the earlier bits

LEDs Interfacing

- Assume an LED each connected through an interfacing circuit such that when bn = 0, the corresponding LED is OFF and = 1 then ON
- When this statement runs, all LEDs become = OFF if earlier they were ON and all those LEDs become = ON when this statement runs if earlier they were OFF at P1.

Using a Port bit individually

- 8051 four number 8-bit ports, P0, P1, P2, and P3
- Each port bit is also addressable
- Px^n refers to bit bn of Px and thus to pin Px.n state
- x = 0 or 1 or 2 or 3 and n = 0 or 1 or .or 6 or 7

Port bit

- P0^1 refers to bit b1 of P0 and thus to pin P0.1 state
- P0^1 refers to bit b1 of byte at address 0x80
- P0^1 also refers to bit at address 0x81

Program to write 0 at fourth pin of P2

- #include <reg51.h> /* Include header file for the registers and SFRs of 8051. */
- void main (void)
- {sbit portBit = P2^4;/*declare variable portBit address as the fourth bit address in SFR P2. */
- portBit = 0; /* Assign value = 0 at address of portBit */
- }

Results

- When the above program in a source file is compiled and run using a simulator
- Simulator shows that P2 bit 4 resets

Program to complement third bit of P3

```
#include <reg51.h> /* Include header file for the
   registers and SFRs of 8051. */
void main (void)
 {sbit portBit = P3^3;/*declare variable portBit
   address as the third bit address in SFR P2. */
   portBit = ~portBit; /* portBit = !portBit in
   ANSI C99.Assign value = 0 at address of
   portBit if 1 else 1 if 0*/
}
```

Result

- When the above program in a source file is compiled and run using a simulator, the simulator shows that P3 bit 3 complements, 0 of 1 and 1 of 0 initially

Summary

We learnt

- Include header file for the Registers
- Programs for the port byte
- Programs for port bit set, reset, complement

End of Lesson 08 on

C Programming Examples for Ports