



DEPARTMENT OF COMPUTER & SOFTWARE
ENGINEERING
COLLEGE OF E&ME, NUST, RAWALPINDI



Microprocessor and Microcontroller-Based Design
Lab #12

SUBMITTED TO:

Lab Eng. Hamza Shami

SUBMITTED BY:

Amina Qadeer

Reg # 359607

DE-42 (C&SE)-A

Submission Date: 26-12-2022

1. Write a code that takes input from PORTD and perform the following operations:

- If $\text{PORTD} < 100$, add 50 in it, and show the output on PORTB
- Otherwise clear PORTB

Also show the results on proteus. Attach the code and proteus screenshots of both cases in lab report. Use LEDs for showing output. Input and output should be clear and understandable.

Code:

```
#include <xc.h>

#include <stdint.h>

#include <pic18f452.h>

int main(void)
{
    // Set PORTD as input and PORTB as output
    TRISD = 0xFF; // set all pins of PORTD as input
    TRISB = 0x00; // set all pins of PORTB as output

    // Read value from PORTD
    uint8_t portd_value = PORTD;

    // If PORTD < 100, add 50 in it and show the result on PORTB
    if (portd_value < 100)
    {
        PORTB = portd_value + 50;
    }

    // Otherwise clear PORTB
```

```
else  
  
{  
  
    PORTB = 0x00;  
  
}  
  
while (1) // infinite loop  
  
{  
  
}  
  
return 0;  
  
}
```

Explanation:

I Using the TRISD and TRISB registers, we first set the direction of PORTD as input and the direction of PORTB as output in this code. Then, using the PORTD register, we read the value of PORTD.

The value of PORTD is then checked to see if it is less than 100. If it is, we double it by 50 before storing the outcome in the PORTB register. We clear PORTB by writing 0x00 to it if the value of PORTD is not zero or more than one hundred.

You can use the simulation tools in proteus to generate the results and view the LEDs' output for various PORTD values. In order to clearly and understandably display the input and output, you can also attach the code and proteus screenshots to your lab report.

Proteus:


```
{  
  
    // Set PORTD as input and PORTB as output  
    TRISD = 0xFF; // set all pins of PORTD as input  
    TRISB = 0x00; // set all pins of PORTB as output  
  
    // Read value from PORTD  
    uint8_t portd_value = PORTD;  
  
    // If the input is even, show it on PORTB  
    if (portd_value % 2 == 0)  
    {  
        PORTB = portd_value;  
    }  
  
    // If it is odd, show lower nibble as it is, and higher nibble complemented on PORTB  
    else  
    {  
        PORTB = ((~portd_value & 0xF0) | (portd_value & 0x0F));  
    }  
  
    while (1) // infinite loop  
    {  
  
    }
```

```

return 0;

}

```

Explanation:

We initially use the TRISD and TRISB registers in this code to specify the direction of PORTD as input and the direction of PORTB as output. Then, using the PORTD register, we read the value of PORTD.

Then, using the modulo operator (%), we determine if the value of PORTD is even by determining if it is divisible by 2. If so, we write PORTD to the PORTB register to display the value. By manipulating the value of PORTD using bitwise operations, we can display the lower nibble as is and the upper nibble complemented on PORTB if the value of PORTD is odd.

Proteus:

