

COURSE: CSE-303 MICROPROCESSOR BASED SYSTEM DESIGN



SEMESTER 6

Name: Zainab Khalid

Reg no: 19 PWCSE 1743

Section: A

Batch: 21 (Spring_2022)

Submitted to: Dr Bilal Habib

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, PESHAWAR
DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING.

TASK 6:

CALCULATOR DESIGN

In this assignment, you are required to design a calculator that should be able to do the following operations,

Addition, Subtraction, Multiplication, and Division.

The result is displayed as soon the user press the “=” button on the keypad. Additionally, pressing the

“C\ON”, clears the contents of the LCD.

Use as many LCD commands to make it user-friendly.

CODE:

```
#include <reg51.h>
#include <stdio.h>
#define ERROR 0 // Any value other than 0 to 9 is good here
//Function declarations
void ports_initialization(void);
void delay(int);
void Lcd_initialization(void);
void write_command(int);
void write_data(char);
void Return(void);
char READ_SWITCHES(void);
char get_key(void);
int get_number(char);
char get_operation(char);
void display_result(int);
sbit RowA = P1^0; //RowA
sbit RowB = P1^1; //RowB
sbit RowC = P1^2; //RowC
sbit RowD = P1^3; //RowD
sbit C1 = P1^4; //Column1
sbit C2 = P1^5; //Column2
sbit C3 = P1^6; //Column3
```

```

sbit C4 = P1^7; //Column4
sbit E = P3^6; //Enable pin for LCD
sbit RS = P3^7; //RS(to check command or data) pin for LCD
// Main program
void main(void)
{
    char key; // key char for keeping record of pressed key
    int number1,number2;
    char operation;
    ports_initialization(); // Make input and output pins as required
    Lcd_initialization(); // Initilize LCD
    write_command(0x80);
    while(1)
    {
        key = get_key(); // Get pressed key
        number1=get_number(key);
        if (number1 != ERROR)
        {
            write_data(key); // Echo the key pressed to LCD
            key=get_key();
            operation=get_operation(key); // Clear scree
            write_data(operation);
            if(operation!=ERROR)
            {
                key=get_key();
                number2=get_number(key);
                if(number2!=ERROR)
                {
                    write_data(key);
                    key=get_key();
                    if(key=='=')
                    {
                        write_data(key);
                        switch(operation){
                            case '+': display_result(number1+number2) ; break;
                            case '-': display_result(number1-number2); break;
                            case 'x': display_result(number1*number2); break;
                            case '/': display_result(number1/number2); break;
                            default: write_command(0x01); //clear screen
                        }
                    }
                }
            }
        }
    }
}

```

```

}
else{
write_command(0x01); //clear screen
}
}
}
void ports_initialization(void)
{
P0 = 0x00; //not used
P1 = 0xf0; //used for generating outputs and taking inputs from Keypad
P2 = 0x00; //used as data port for LCD
P3 = 0x00; //used for RS and E
}
void delay(int d)
{
int i;
for(i=0;i<d;i++); //null statement
}
void write_data(char s)
{
RS = 1; // This is data
P2 = s; //Data transfer
E = 1; // => E = 1
delay(150);
E = 0; // => E = 0
delay(150);
}
void write_command(int a)
{
RS = 0; // This is command
P2 = a; //Data transfer
E = 1; // => E = 1
delay(150);
E = 0; // => E = 0
delay(150);
}
void Lcd_initialization(void)
{
////////// Reset process from datasheet //////////
delay(15000);
write_command(0x30);
delay(4500);
write_command(0x30);

```

```

delay(300);
write_command(0x30);
delay(650);
////////////////////////
write_command(0x38); //function set
write_command(0x0c); //display on,cursor off,blink off
write_command(0x01); //clear display
write_command(0x06); //entry mode, set increment
}
void Return(void) //Return to 0 location on LCD
{
    write_command(0x02);
    delay(1500);
}
char Read_Switches(void)
{
    RowA = 0; RowB = 1; RowC = 1; RowD = 1; //Test Row A
    if (C1 == 0){ delay(10000);while (C1==0); return '7'; }
    if (C2 == 0) { delay(10000); while (C2==0); return '8'; }
    if (C3 == 0) { delay(10000); while (C3==0); return '9'; }
    if (C4 == 0) { delay(10000); while (C4==0); return '/'; }
    RowA = 1; RowB = 0; RowC = 1; RowD = 1; //Test Row B
    if (C1 == 0) { delay(10000); while (C1==0); return '4'; }
    if (C2 == 0) { delay(10000); while (C2==0); return '5'; }
    if (C3 == 0) { delay(10000); while (C3==0); return '6'; }
    if (C4 == 0) { delay(10000); while (C4==0); return 'x'; }
    RowA = 1; RowB = 1; RowC = 0; RowD = 1; //Test Row C
    if (C1 == 0) { delay(10000); while (C1==0); return '1'; }
    if (C2 == 0) { delay(10000); while (C2==0); return '2'; }
    if (C3 == 0) { delay(10000); while (C3==0); return '3'; }
    if (C4 == 0) { delay(10000); while (C4==0); return '-'; }
    RowA = 1; RowB = 1; RowC = 1; RowD = 0; //Test Row D
    if (C1 == 0) { delay(10000); while (C1==0); return '*'; }
    if (C2 == 0) { delay(10000); while (C2==0); return '0'; }
    if (C3 == 0) { delay(10000); while (C3==0); return '='; }
    if (C4 == 0) { delay(10000); while (C4==0); return '+'; }
    return 'n'; // Means no key has been pressed
}
char get_key(void) //get key from user
{
    char key = 'n'; //assume no key pressed
    while(key=='n') //wait untill a key is pressed
        key = READ_SWITCHES(); //scan the keys again and again
}

```

```

return key; //when key pressed then return its value
}
int get_number(char character)
{
    switch(character){
    case '0': return 0; break;
    case '1': return 1; break;
    case '2': return 2; break;
    case '3': return 3; break;
    case '4': return 4; break;
    case '5': return 5; break;
    case '6': return 6; break;
    case '7': return 7; break;
    case '8': return 8; break;
    case '9': return 9; break;
    default: return ERROR; break; //it means wrong input
    }
}
char get_operation(char key)
{
    if(key=='+' || key=='-' || key=='x' || key=='/')
    {
        return key;
    }
    else{
        return ERROR;
    }
}
void display_result(int number)
{
    unsigned char Unit = 0; //It will contain unit digit of numb
    unsigned char Tenth = 0; //It will contain 10th position digit of numb
    if(number<0)
    {
        number = -1*number; // Make number positive
        write_data('-'); // Display a negative sign on LCD
    }
    Tenth = (number/10); // Findout Tenth Digit

    if( Tenth != 0) // If it is zero, then don't display
    write_data(Tenth+0x30); // Make Char of TenthDigit and then display it on LCD
    Unit = number - Tenth*10;

```

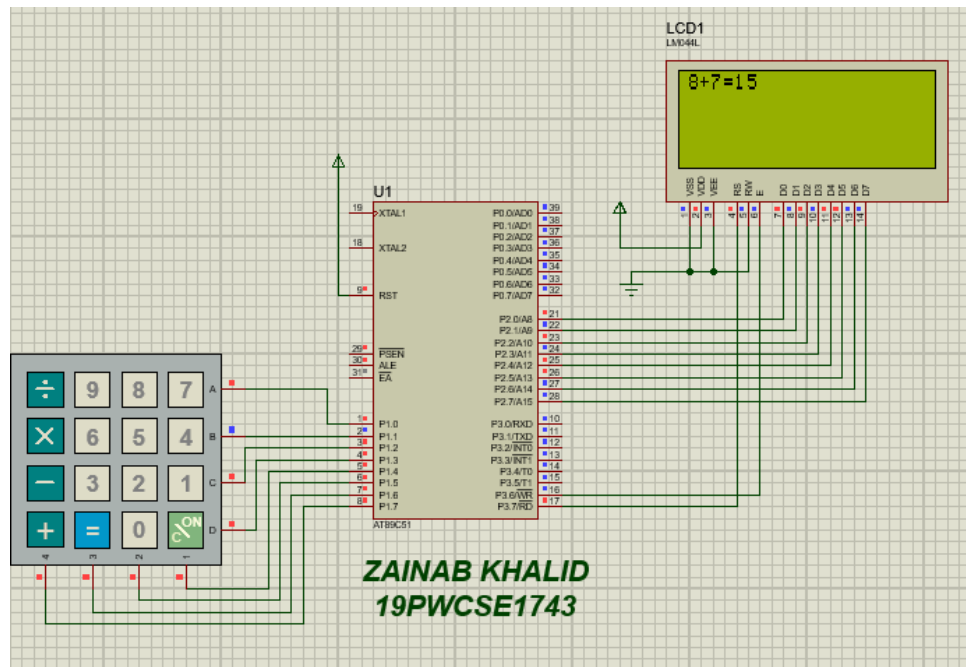
```

write_data(Unit+0x30); // Make Char of UnitDigit and then display it on LCD
}

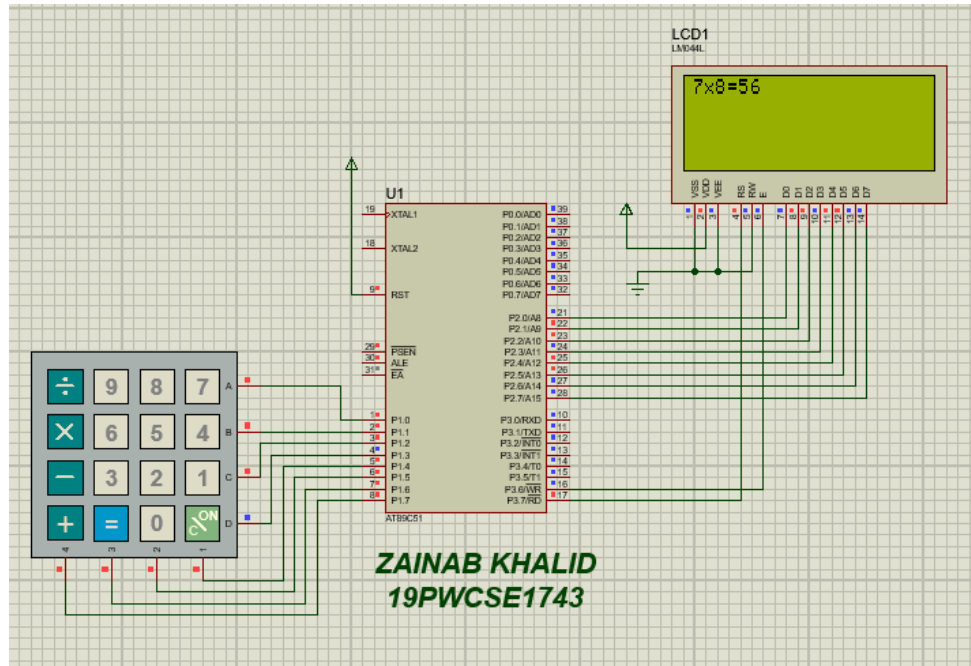
```

OUTPUT:

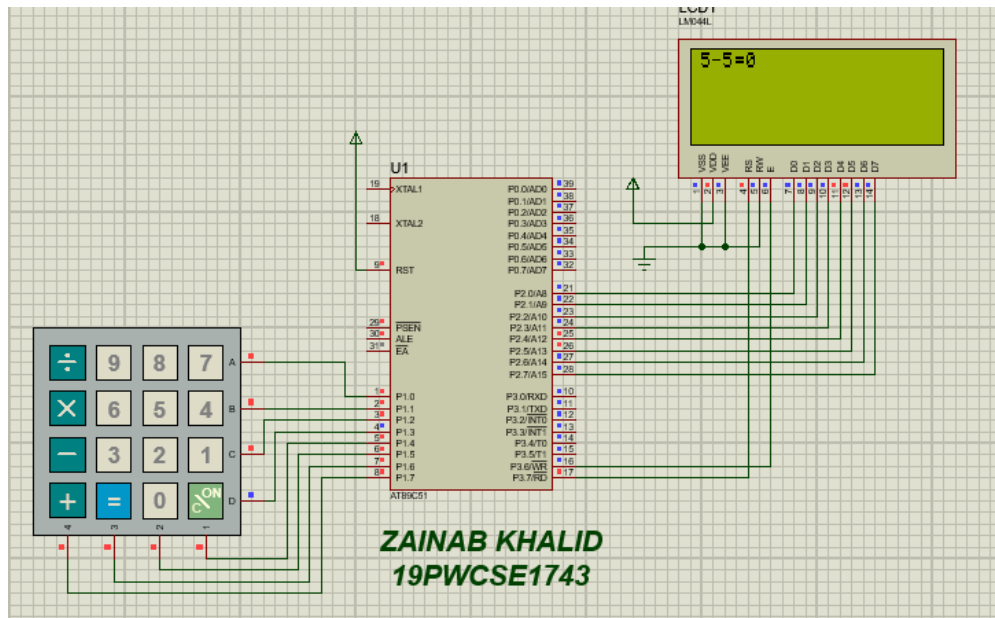
ADDITION:



MULTIPLICATION:



SUBTRACTION:



DIVISION:

