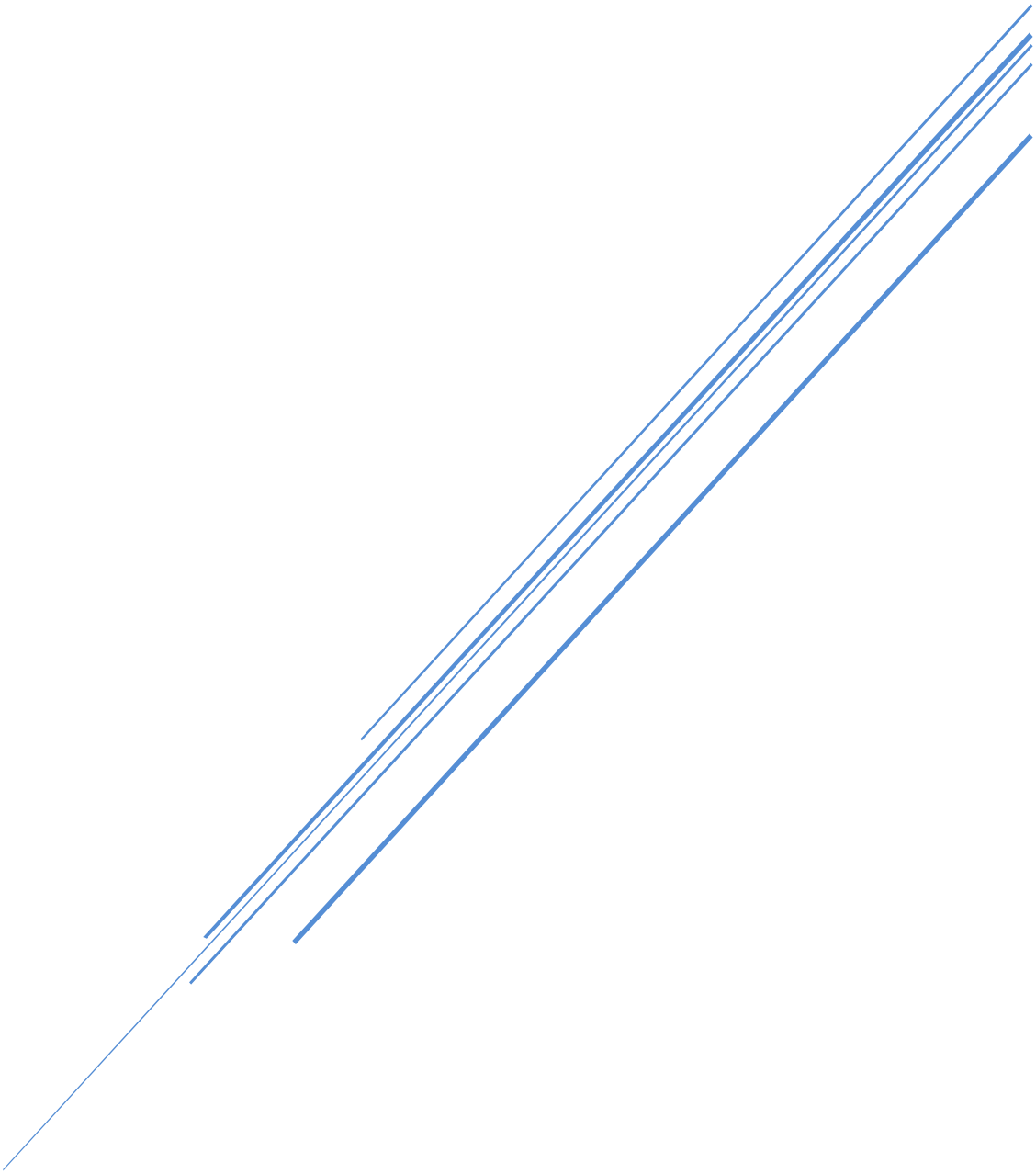
Higher National Diploma in Network Engineering

National Institute of Business Management

Subject

Embedded Application Development

Course Work 1 & 2



COHNDNE20.1F-021
Jadhusan M.S

CONTENTS

Assignment 01

Components:

1. GROUND
2. Animated BI-Colour LED model (Blue/Yellow) with self-flashing
3. Analogue Primitive (RTSWITCH)
4. PULSE Generator

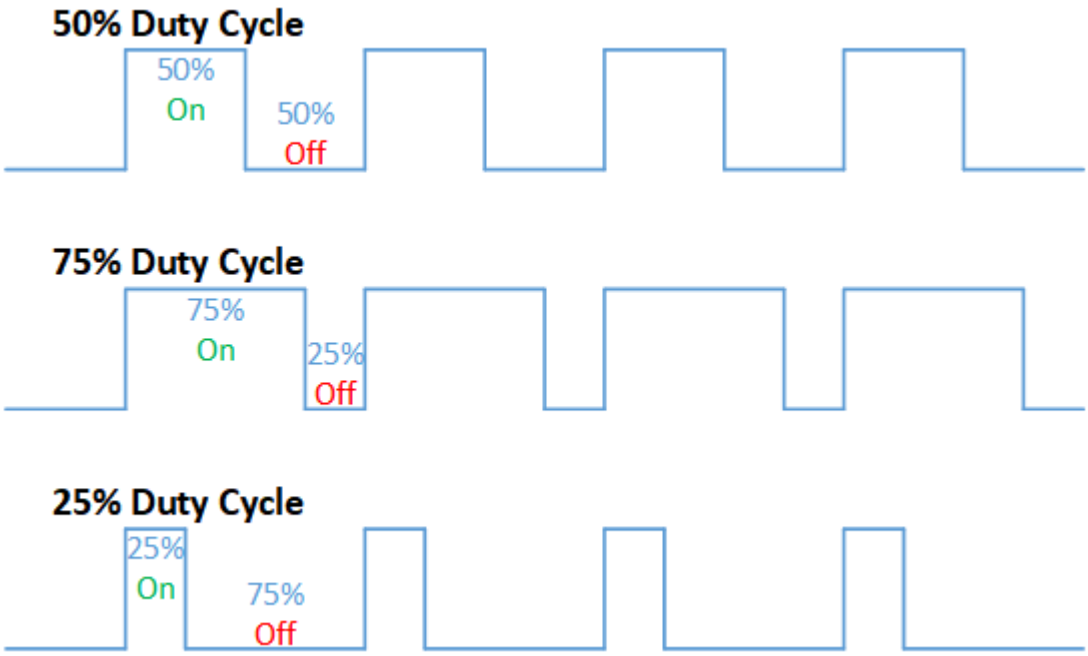
Descriptions:

Glow a LED by using Pulse width Modulation technique and observe how LED blinking time vary with pulse modulation, pulse rising time and pulse frequency.

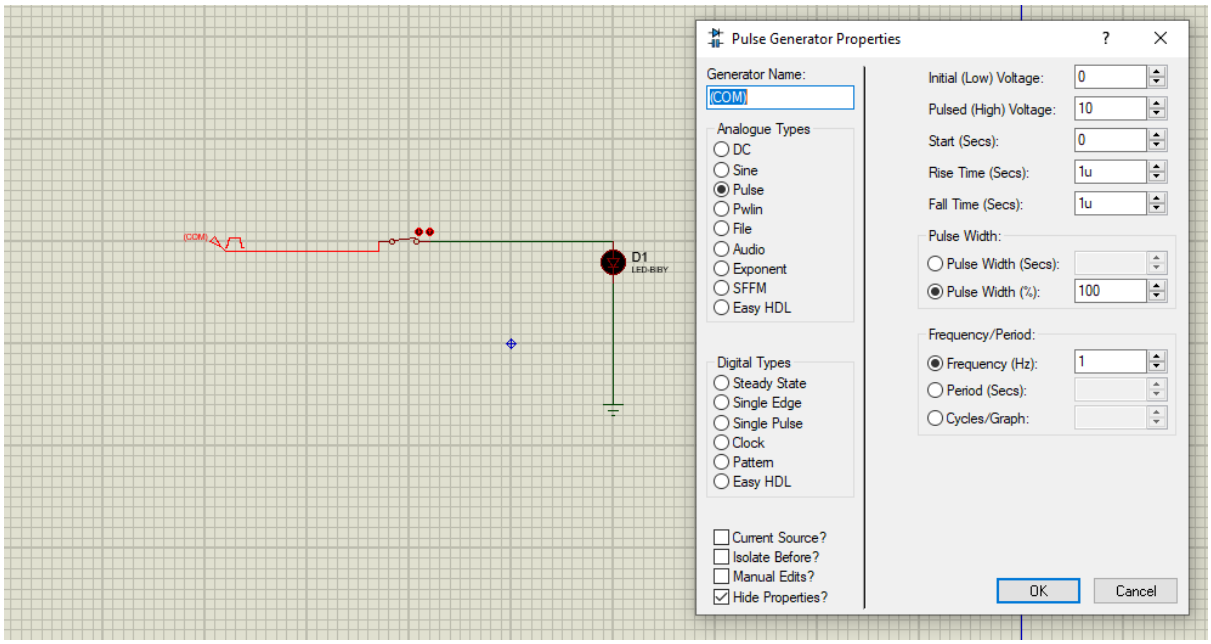
Analysis:

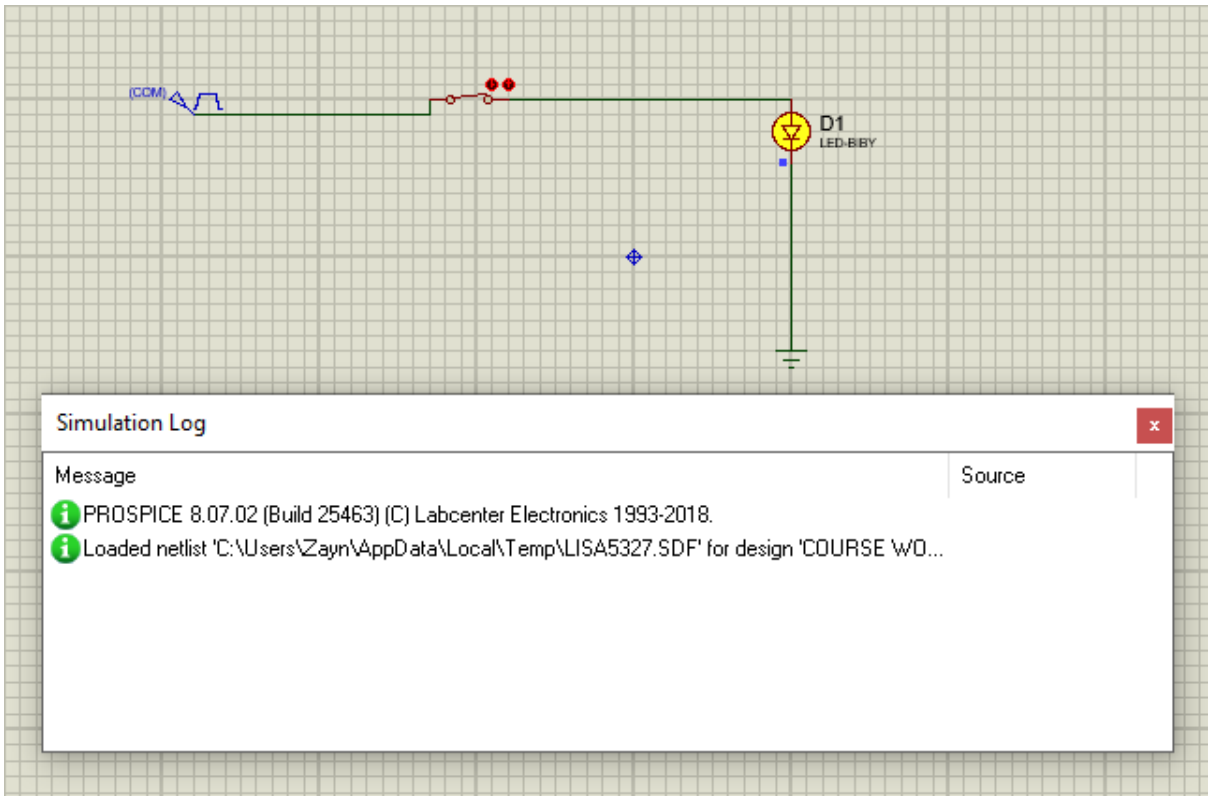
- **Pulse Width Modulation**, the duration that bulb takes to work which is used in pulse width to determine the time span.
- While changing the width, the time span for the bulb to turn on vary!

The **below diagram** explains that the 50% of the width makes the bulb turn on and the balance turn off. And more



My **Practical** has a width range of 100% which makes the bulb **TURN ON** for the duration**.**





Assignment 02

Components:

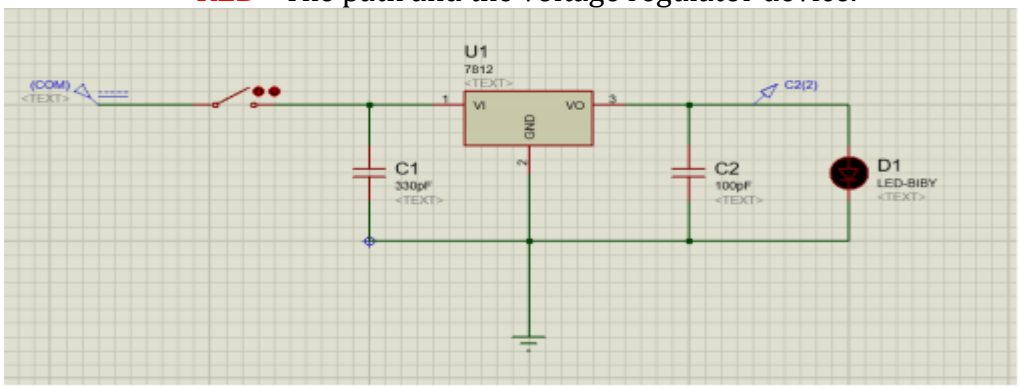
1. GROUND
2. DC Voltmeter
3. Animated BI-Colour LED model (Blue/Yellow) with self-flashing
4. Analogue Primitive (RTSWITCH)
5. PULSE Generator
6. Generator - DC
7. Capacitors – 2 (C1-330p & C2-100p)
8. 12V Fixed 1A Positive Power Supply Regulator – 7812
9. Probe – Voltage

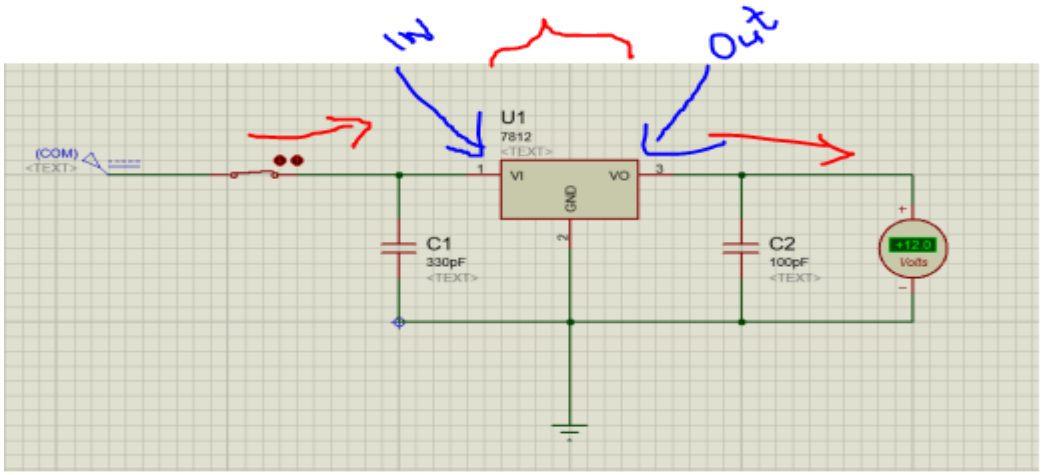
Descriptions:

Design DC Voltage Regulator circuit and observe how voltage regulation functions.

Analysis:

- **Voltage Regulation** gives the summary of the measurements of VOLTAGE throughout the in and out pins between 2 components/nodes.
- **RED**= The path and the voltage regulator device.





Assignment 03

Components:

1. DC motor
2. Logic Gates
3. Pulse/Dc Generator

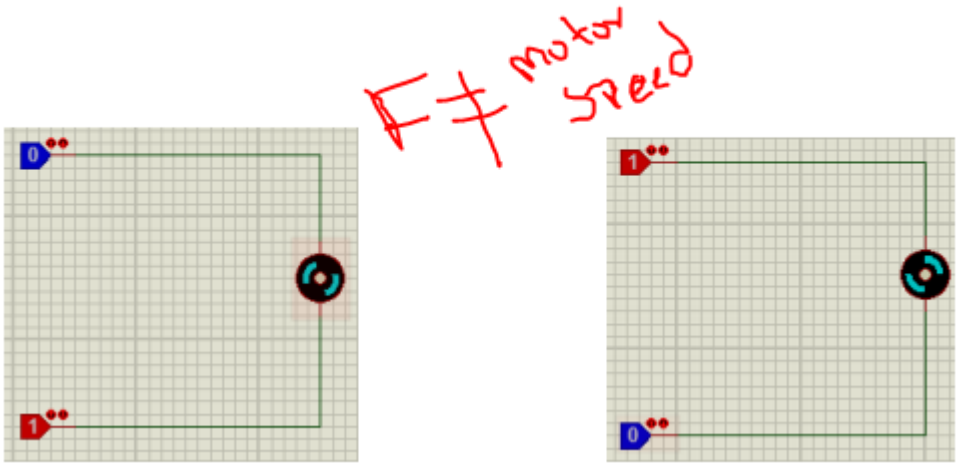
Descriptions:

Running a DC Motor using logic gates & Pulse Generator.

Analysis:

- DC Motor which has 2 modes with clockwise and anti-clockwise, where the motor rotates in the direction of the clock and anti-clockwise is reversed.
- When the logic gates get 1, it turns on (Rotate Clockwise).
- When the logic gates get 0, it turns off (Rotate Anti-Clockwise).
- And the pulse generator defines the pulse frequency.

Therefore, when the frequency gets high the motor start rotating faster.



By Changing Logic gates value shown

Assignment 04

Components:

1. LOGICSTATE
2. 7SEG -BCD
3. 7447 IC

Descriptions:

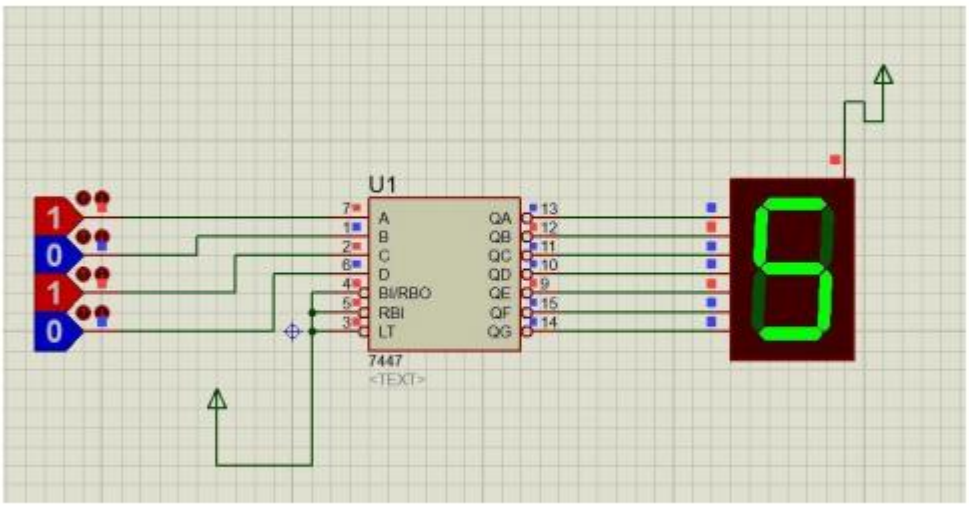
Change the frequency of the clock (5Hz) the 7SEG-BCD lights up and starts to display numbers in randomly.

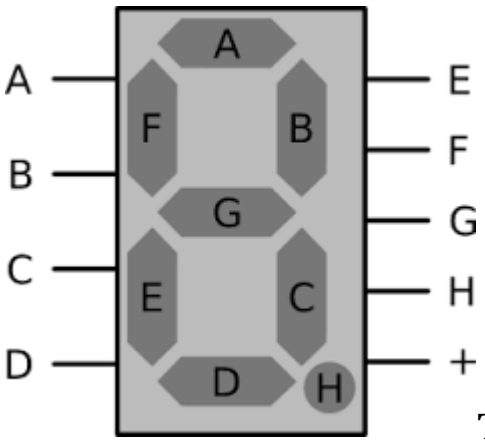
Analysis:

- **7 segment BCD counter, 7** segments is a form of electronic display for displaying decimal number, widely used in clocks.
- The BCD counter is a binary coded decimal, which converts the binary to decimal.
- However, when the voltage use binary the electronic display understands decimal therefore the BCD comes to the place.

The below diagram shows the 7 segment which gives "1" and then the lights turn on and makes the number in clock.
So, the BCD helps to convert the voltage value to the display.

If the time comes to 5, the 7 segment gives "1" to turn the light on accordingly.





The 7 segments using light display.

Assignment 05

Components:

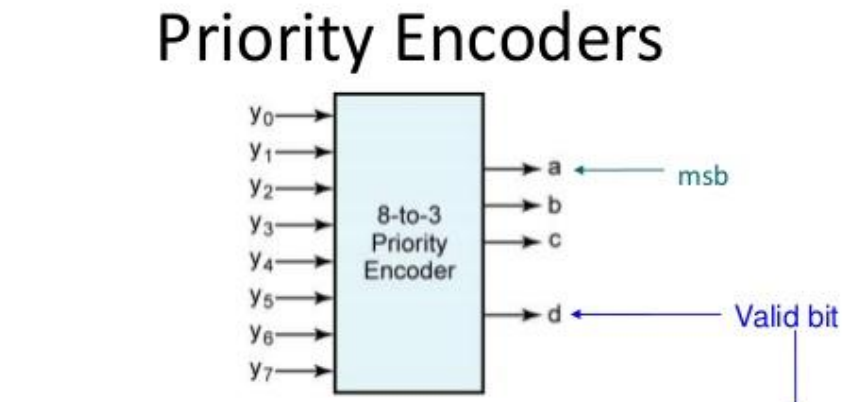
1. 7420 NAND GATEs
2. Logic States

Descriptions:

Analysis:

- **Priority Encoders,** it's a circuit/algorithm compresses binary multiple inputs into smaller number of outputs. Used to reduce the number of wires needed in a particular circuit.
- Just to reduce the wires!

8-to-3-bit priority encoder which has eight active LOW logic '0' inputs
And provides a 3-bit code of the highest ranked input at its output.



y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0	1
X	X	X	1	0	0	0	0	0	1	1	1
X	X	X	X	1	0	0	0	1	0	0	1
X	X	X	X	X	1	0	0	1	0	1	1
X	X	X	X	X	X	1	0	1	1	0	1
X	X	X	X	X	X	X	1	1	1	1	1

Assignment 06

Components:

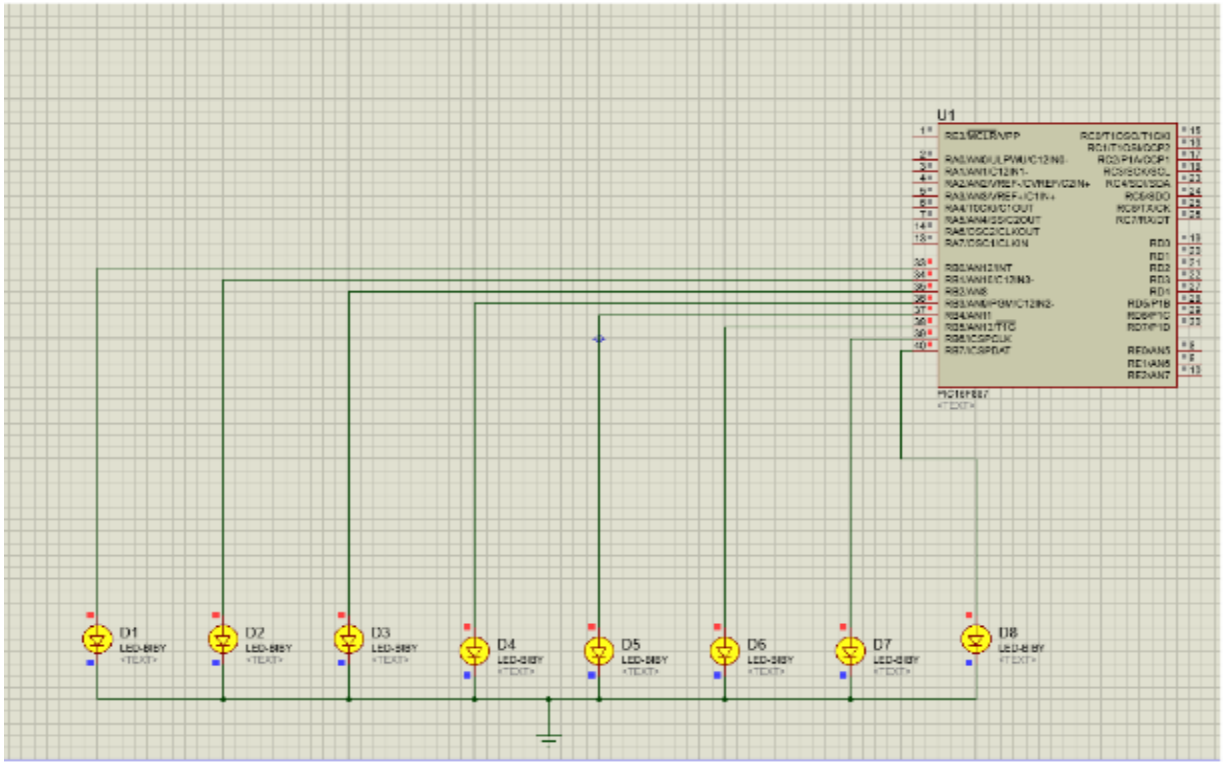
1. LEDs
2. PIC16F887
3. GROUND

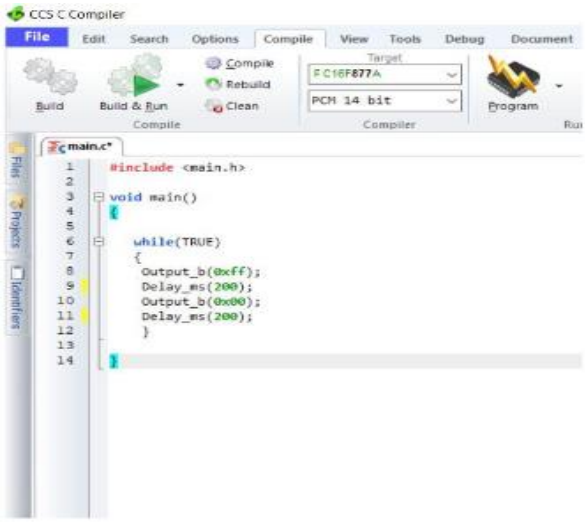
Descriptions:

Blinking 8 LEDS using CCS – Compiler & proteus – 1.

Analysis:

Using CCS compiler, program the IC and make the .hex file and upload it in proteus.





- The pins from 32-40 output B, and the while loop continues running.
- The port B pin root, and use code included an output an (0xff) hex value and the IC gets as binary value. Makes the port b pins UP and gives a delay for 10 sec Makes the pin turn on for 200 milli sec.
- The values are 1 and 1 makes pin to ON state, which means all the port values are 1 all the LEDs turns on.

Assignment 07

Components:

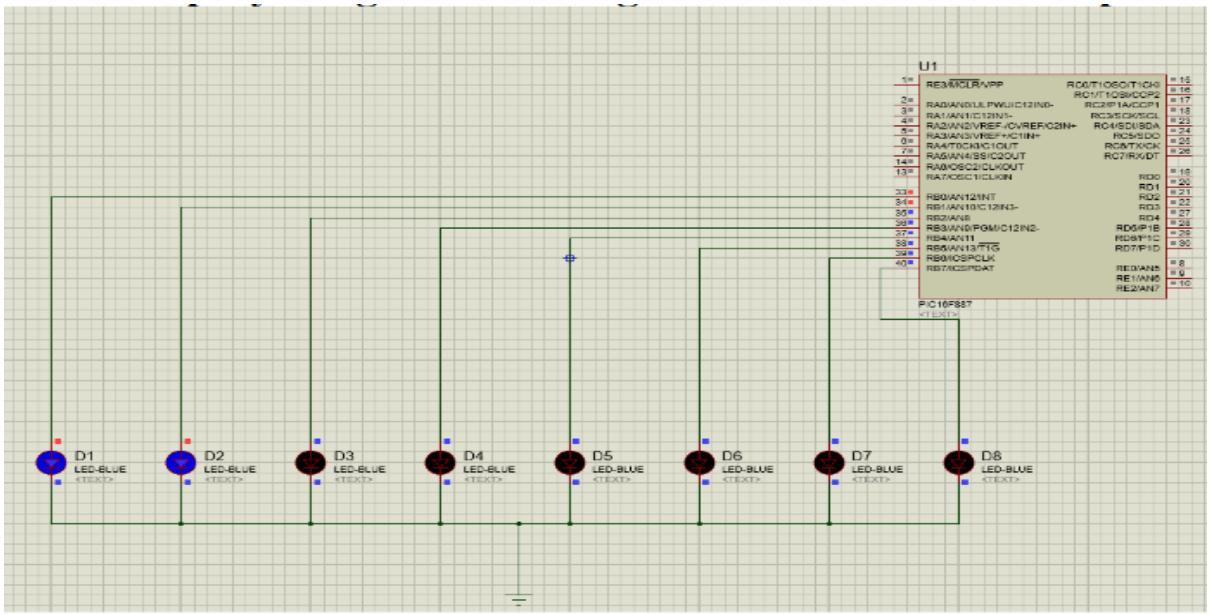
5. LEDs
6. PIC16F887
7. GROUND

Descriptions:

Blinking 8 LEDS using CCS – Compiler & proteus – 2

Analysis:

The Same CCS compiler, circuit like Assignment 06..



main.c

```c
#include <main.h>

void main()
{

   while(TRUE)
   {
    Output_b(0x01);
    Delay_ms(200);
    Output_b(0x03);
    Delay_ms(200);
    Output_b(0x07);
    Delay_ms(200);
    Output_b(0x0f);
    Delay_ms(200);
    Output_b(0x1f);
    Delay_ms(200);
    Output_b(0x3f);
    Delay_ms(200);
    Output_b(0x7f);
    Delay_ms(200);
    Output_b(0xff);
    Delay_ms(200);
    }

}
```

- This while loop contains Output of 8 different HEX value.
- When the HEX converts to Binary, so it shows the flow of how the light get blink.
- We can change the hex value, and make the light get blink in different sequence such as : 1 by 1 light blink or 2 by 2.

Assignment 08

Components:

8. LEDs
9. PIC16F887
10. GROUND

Descriptions:

Blinking 8 LEDS using CCS – Compiler & proteus – 3

Analysis:

This proteus concept is same as the last 2 assignments 6 & 7.
Which makes the code difference to analyse how this functions.

```
#include <main.h>

void main()
{

   while(TRUE)
   {
    Output_b(0xfe);
    Delay_ms(200);
    Output_b(0xfd);
    Delay_ms(200);
    Output_b(0xfb);
    Delay_ms(200);
    Output_b(0xf7);
    Delay_ms(200);
    Output_b(0xef);
    Delay_ms(200);
    Output_b(0xdf);
    Delay_ms(200);
    Output_b(0xbf);
    Delay_ms(200);
    Output_b(0x7f);
    Delay_ms(200);
    }

}
```

- The Hex value with ‘fe,fd,f7,ef,df,bf,7f’. Converting to binary makes like this,
  - 01111111 ,10111111 ,11011111 ,11101111 ,11110111 ,11111011,11111101,11111110.
  - Summarizing this, gives a output of while loop continuous running makes the light state first bulb off and rest turn on for around 200MS constantly

Assignment 09

Components:

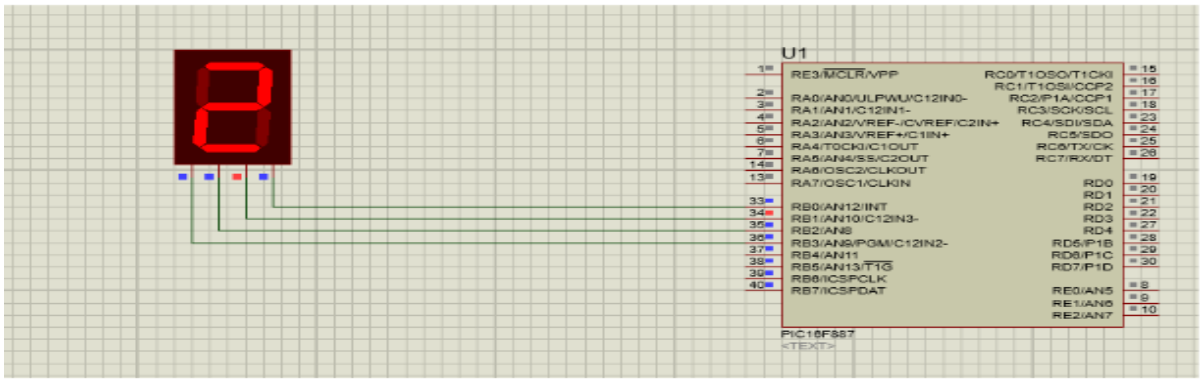
11. 7SEG BCD
12. PIC 16F887

Descriptions:

Controlling 7SEG BCD IC using 16F887 – Compiler & proteus -1

Analysis:

This Proteus anal zing the 7 segment bcd.




```
#include <main.h>

void main()
{

   while(TRUE)
   {
    Output_b(0x01);
    Delay_ms(200);
    Output_b(0x02);
    Delay_ms(200);
    Output_b(0x03);
    Delay_ms(200);
    Output_b(0x04);
    Delay_ms(200);
     Output_b(0x05);
    Delay_ms(200);
     Output_b(0x06);
    Delay_ms(200);
     Output_b(0x07);
    Delay_ms(200);
     Output_b(0x08);
    Delay_ms(200);
     Output_b(0x09);
    Delay_ms(200);
     Output_b(0x0a);
    Delay_ms(200);
     Output_b(0x0b);
    Delay_ms(200);
     Output_b(0x0c);
    Delay_ms(200);
     Output_b(0x0d);
    Delay_ms(200);
     Output_b(0x0e);
    Delay_ms(200);
     Output_b(0x0f);
    Delay_ms(200);
```

- This code while loop, contains about 15 value representing the states. This time it makes a 200ms of delay
- Where the numbers such as 0x01 means 1 which changes and makes the state on, while displaying the time and repeat the same process.

Assignment 10

Components:

13. 7SEG BCD
14. PIC 16F887

Descriptions:

Controlling 7SEG BCD IC using 16F887 – Compiler & proteus – 2

Analysis:

The proteus project is same has assignment 08, which part 2.
In this the while loop code has been changed.

```
#include <main.h>

void main()
{

   while(TRUE)
   {
    Output_b(0x01);
    Delay_ms(50);
    Output_b(0x03);
     Delay_ms(50);
    Output_b(0x05);
     Delay_ms(50);
    Output_b(0x07);
     Delay_ms(50);
     Output_b(0x09);
    Delay_ms(50);
     Output_b(0x0b);
     Delay_ms(50);
     Output_b(0x0d);
     Delay_ms(50);
     Output_b(0x0f);
     Delay_ms(50);

    }

}
```

- The value didn't change and or converted,
- There is a change of delay with 50 MS, then the previous assignments.
- And the while loop constantly runs and makes the state work.

Assignment 11

Components:

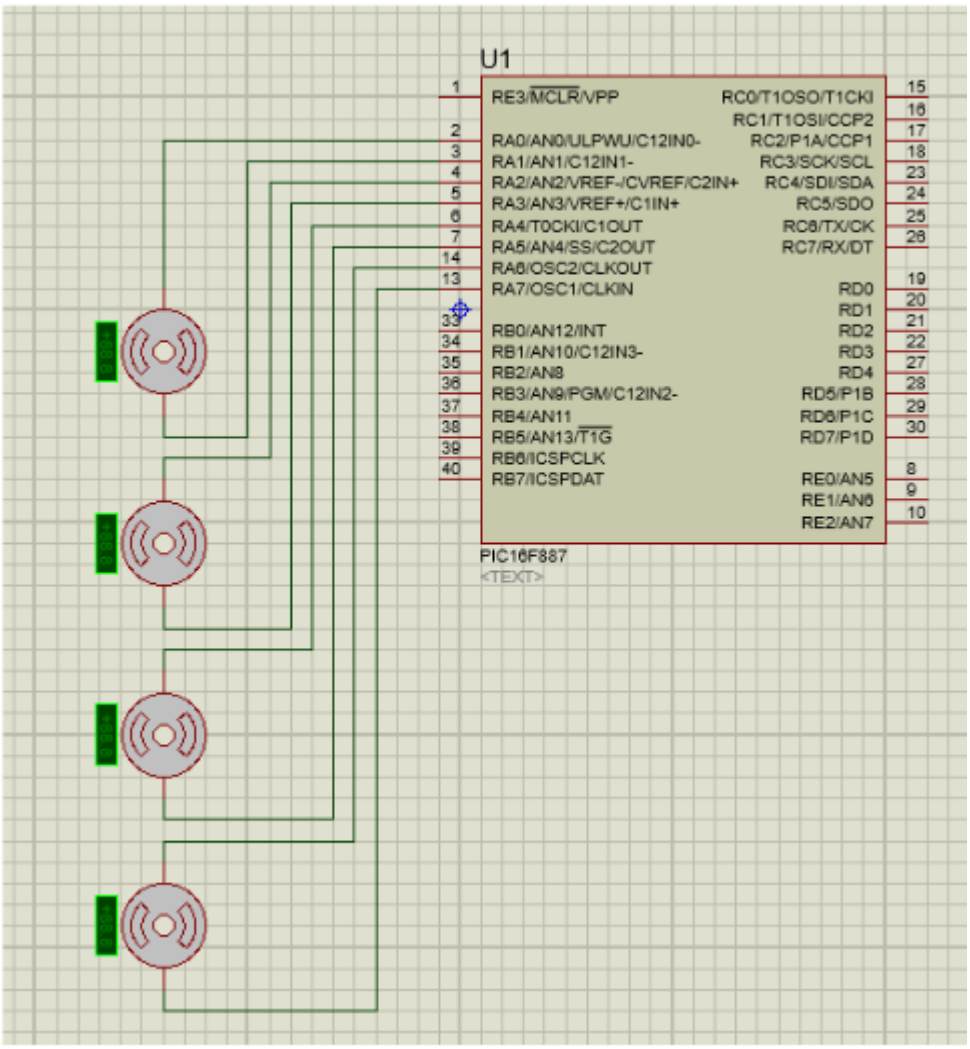
15. DC Motor
16. PIC16F887

Descriptions:

Controlling DC motors using pic 16F887 IC – CCS compiler & Proteus

Analysis:

The below diagram uses the IC used in above assignments making 4 motors connected in different ports.



CODE:

```
#include <main.h>

void main()
{

   while(TRUE)
   {
```

- Having a continuous while loop running.

- While having hex values, the ‘a5’, which makes the binary value 10100101 after the convert.

The binary value (**10100101**)

This explains that 1 rotates the motor to clockwise and 0 to anticlockwise

While anal zing the binary value

The motor rotates depending on the value!

And it works constantly, with the code.

Assignment 12

Components:

1. Arduino

Descriptions:

Now using the Arduino Uno

Analysis:

Arduino boards are able to read inputs. Functions allow structuring the programs in segments of code to perform individual tasks