BILKENT UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
CS224- PRELIMINARY DESIGN REPORT

Lab 7
Section 1
Berdan Akyürek 21600904
Ömer Olkun 21100999
25.12.2019

b)

TRISx: These bits control the pins' missions as inputs or outputs. For example if TRISA = 0, this means all pins of Port A are outputs. Or if TRISA = 111111...11, all pins of Port A are inputs.
PORTx: We can flow the data using PORTx. For example we can get PORTA data using X = PORTA or send data to PORTA using PORTA = 0xF3
LATx: This bits is used to write output. We can set input using LATB= 13 and we can set the output to 13.
ODCx: These bits makes each pin behave open-drain or not. If ODC=1, that ports behave open-drain.

c)

```
// This code shows and rotates the pattern (10001000) right or
stops based on the
//input coming from the user. The pattern is to be shown on the
LEDs.
int stop = 0;
int initial = 0b01111111; //Initial pattern. Note that 0 means on,
while 1 means off.
int right = 1;
void main(){
 TRISD = 0x0; // All bits of PORTD are output. 0 means output
// Three bits of PORTA are inputs but only one of them is used in
this example as a
//stop button, others are redundant. 1 means input
TRISA = 0b11111111;
// From PORTD, outputs will be sent to LEDs. Make sure that you
physically connected
//them by looking at the Figure 1, in the directives document.
// Initial pattern is sent to the LEDs through PORTD.
 PORTD = initial;
 while(1)
 {
          int lsb;
          int msb;
          int mask;
          int stopped;
          // Stop button is the push-button which is labeled as 1
on the board.
          if(PORTABits.RA1 == 0)
          { // If stop button clicked
            stop = !stop;
            if(stop)
            {
                    stopped = PORTD;
            }
            if(!stop)
            {

                      PORTD = stopped;
            }

          }

          if (PORTABits.RA2 == 0)
          {
                            right = !right;
          }

          if(!stop)
          {
                   if( right == 1 )
                   {
                            //Rotate right
                            lsb = PORTD & 0x1; // Extract least
significant bit
                            mask = lsb << 7; // Least significant
bit will be the msb of the
                            //shifted pattern
                            PORTD = (PORTD >> 1) | mask; // Paste
lsb to the leftmost bit the
                            //right shifted portd
                   }
                   if( right == 0 )
                   {

                            //Rotate left
                            msb = PORTD & 0b10000000; // Extract
least significant bit
                            mask = msb >> 7; // Least significant
bit will be the msb of the
                            //shifted pattern
                            PORTD = (PORTD << 1) | mask; // Paste
lsb to the leftmost bit the
                            //right shifted portd

                   }
          }
          else
          {
               //Do not shift anything, that is, stop.
               PORTD = 0b11111111;
          }
               delay_ms(1000); // Wait 1 second.
 }
}
// Rotation ends here
```

d)

```
//Berdan Akyurek 21600904
//Omer Olkun 21100999

void main()
{
     int x = 1;
     int result;
     int digit;
     int i;
     int result2;
     int j;

     TRISD = 0x0;
     TRISE = 0x0;

    /*while(1)
     {
             PORTEBits.RE0 = 1;
                         PORTEBits.RE1 = 0;
                         PORTEBits.RE2 = 0;
                         PORTEBits.RE3 = 1;

                                  PORTDBits.RD0 = 1; //a
                                 PORTDBits.RD1 = 1; //b
                                 PORTDBits.RD2 = 0; //c
                                 PORTDBits.RD3 = 1; //d
                                 PORTDBits.RD4 = 1; //e
                                 PORTDBits.RD5 = 0; //f
                                 PORTDBits.RD6 = 1; //g
                                 PORTDBits.RD7 = 0; //dp
     }*/
    while(1)
     {
             result = x*x*x;
             ////////////////////////////////////////
             for( j = 0; j < 360 ; j ++ )
             {
                   result2 = result;
                   for( i = 0; i < 4 ; i ++ )
                   {
                         delay_ms(5);
                         digit = result2 %10;
                         //digit = digit >> 12;

                         PORTEBits.RE0 = 0;
                         PORTEBits.RE1 = 0;
                         PORTEBits.RE2 = 0;
                         PORTEBits.RE3 = 0;
                         //digit = 6;
                         if( i == 0 )
                         {
```

```
    PORTEBits.RE3 = 1;
}
else if (i == 1)
{
    PORTEBits.RE2 = 1;
}
else if (i == 2)
{
   PORTEBits.RE1 = 1;
}
else if (i == 3)
{
   PORTEBits.RE0 = 1;
}

if(digit == 0 )
{
        PORTD = 0x3F; //a

}
else if( digit == 1 )
{
        PORTD = 0x06; //a

}
else if( digit == 2 )
{
        PORTD = 0x5B; //a

}
else if( digit == 3 )
{
        PORTD=0x4F; //a

}
else if( digit == 4 )
{
        PORTD = 0x66; //a

}
else if( digit == 5 )
{
        PORTD = 0x6D; //a

}
else if( digit == 6 )
{
        PORTD = 0x7D; //a

}
else if( digit == 7 )
{
        PORTD = 0x07; //a
```

```
                         }
                         else if( digit == 8 )
                         {
                                 PORTD = 0x7F; //a

                         }
                         else if( digit == 9 )
                         {
                                 PORTD = 0x6F; //a

                         }
                         else
                         {
                                 PORTDBits.RD0 = 0; //a
                                 PORTDBits.RD1 = 0; //b
                                 PORTDBits.RD2 = 0; //c
                                 PORTDBits.RD3 = 0; //d
                                 PORTDBits.RD4 = 0; //e
                                 PORTDBits.RD5 = 0; //f
                                 PORTDBits.RD6 = 1; //g
                                 PORTDBits.RD7 = 0; //dp
                         }
                         ////////////////////////////////////////
                         result2 = result2 / 10;

                   }
             }

             if( x == 21 )
             {
                    x = 0;
             }
             x ++;
             //delay_ms(1000);
     }
}
```