BILKENT UNIVERSITY
COMPUTER ENGINEERING

CS224 – COMPUTER ORGANIZATION

LAB 7 PRELIMINERY REPORT

BORAN YILDIRIM 21400196 – SECTION 3

TRIS (tri-state) Registers
TRIS registers configure the data direction flow through port I/O pin(s). The TRIS register bits determine whether a PORT I/O pin is an input or an output.

PORT Registers
PORT registers allow I/O pins to be accessed (read).

LAT Registers
LAT registers (PORT data latch) hold data written to port I/O pin(s).

2a)

```
int pattern = 0xBC;
int temp = 0xBC;
int disabled = 0;


void main() {
  AD1PCFG = 0xFFFF;

  DDPCON.JTAGEN = 0; // disable JTAG

  TRISA = 0x00;  //portA is output to turn on LEDs.
  TRISE = 0XFF;  //portE is inputs to read push-buttons.

  LATA = pattern;       // turn OFF the PORTA leds
  LATE = 0X00;       // turn OFF the PORTE led ( single
green LED on PIC32 Beti)

  while(1) {
      if (portE == 0x03) {
          if (disabled == 1) {
```

```
            LATA = temp;
            disabled = 0;
        }
        if (LATA % 2 == 1) {
            LATA = LATA >> 1;
            LATA = LATA + 0x80;
        }
        else {
            LATA = LATA >> 1;
        }
     Delay_ms(1000);
    }

    if (portE == 0x01) {
        if (disabled == 1) {
            LATA = temp;
            disabled = 0;
        }
        if (LATA / 0x80 == 1){
            LATA = LATA << 1;
            LATA = LATA + 1;
        }
        else{
            LATA = LATA << 1;
        }
        Delay_ms(1000);
    }

    if (portE % 2 != 1) {
        if (disabled == 0){
            temp = LATA;
            LATA = 0xFF;
            disabled = 1;
        }
    }
  }
}
```

2b)

```
#define SW1 PORTCbits.RC0

unsigned char  const SEGMENT_MAP[10] = {0x3F,0x06,0x5B,
0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

void main(void) {
    TRISB = 0x00;                    //Set PortB to all
outputs
    TRISCbits.RC0=0x01;              //Set PortC.0 asinput
    OSCCON = 0X76;                   //SET fOR INTERNAL
OSCCILATOR
    char digit=0;

    while(1){
        //Check if switch SW1 is closed
        if (!SW1) {
          for(int i = 0; i < 20; i++)
              Delay_ms(1000);   //wait for 100ms

          //Check if switch SW1 is still closed
          if (!SW1) {
              PORTB = (SEGMENT_MAP[digit]);

              for (int i = 0; i < 20; i++)
                    Delay_ms(1000);   //wait for 100ms
              digit++;
              if (digit > 9)
                  digit = 0;
            }
      }

        else {
            int n = 10, first = 0, second = 1, next;

            for (int i = 0; i < n; i++) {
                if ( i <= 1 )
                    next = i;
                else {
                    next = first + second;
                    first = second;
                    second = next;
```

```
                }
                PORTB = next;
            }
        }
    }
}
```