

**BILKENT UNIVERSITY  
COMPUTER ENGINEERING  
CS 224  
COMPUTER ORGANIZATION**

**PRELIMINARY DESIGN REPORT  
LAB 07**

**BERK YILDIZ**

**21502040  
SECTION 4**

**27.12.2018**

## 1. b)

A Special Function Register is a register within a microprocessor, which controls or monitors various aspects of the microprocessor's function [1].

- The TRISx register control bits determine whether each pin associated with the I/O port is an input or an output [2].
- Data on an I/O pin is accessed via a PORTx register. A read of the PORTx register reads the value of the I/O pin, while a write to the PORTx register writes the value to the port data latch [2].
- The LATx register associated with an I/O pin eliminates the problems that could occur with read-modify-write instructions. A read of the LATx register returns the values held in the port output latches, instead of the values on the I/O pins [2].
- Each I/O pin can be individually configured for either normal digital output or open-drain output. This is controlled by the ODCx, associated with each I/O pin. The open drain feature allows the generation of outputs higher than VDD on any desired digital only pins by using external pull-up resistors [2].

For the Part2.b special function registers are TRISA, LATA, TRISD and LATD.

**d)**

```
unsigned short shifter, portd_index;
unsigned int    digit, number;
unsigned short portd_array[4];

//----- Returns mask for common cathode 7-seg. display
unsigned short mask(unsigned short num) {
    switch (num) {
        case 0 : return 0x3F;
        case 1 : return 0x06;
        case 2 : return 0x5B;
        case 3 : return 0x4F;
        case 4 : return 0x66;
        case 5 : return 0x6D;
        case 6 : return 0x7D;
        case 7 : return 0x07;
        case 8 : return 0x7F;
        case 9 : return 0x6F;
    } //case end
}

void interrupt() {
    LATA = 0;                                // Turn off all 7seg displays
    LATD = portd_array[portd_index];         // bring appropriate value to PORTD
    LATA = shifter;                          // turn on appropriate 7seg. display

    // move shifter to next digit
    shifter <<= 1;
    if (shifter > 8u)
        shifter = 1;
}
```

CS224  
Section 4  
Fall 2018  
Lab 07  
Berk Yıldız / 21502040

```
// increment portd_index
portd_index++;
if (portd_index > 3u)
    portd_index = 0;
TMR0L = 0; // reset TIMER0 value
TMR0IF_bit = 0; // Clear TMR0IF
}

void main() {
    ANSELA = 0; // Configure PORTA pins as digital
    ANSELB = 0; // Configure PORTB pins as digital
    ANSELD = 0; // Configure PORTD pins as digital

    TRISA = 0; // Configure PORTA as output
    LATA = 0; // Clear PORTA
    TRISD = 0; // Configure PORTD as output
    LATD = 0; // Clear PORTD

    TOCON = 0xC4; // Set TMR0 in 8bit mode, assign
                  //prescaler to TMR0
    TMR0L = 0; // clear TMR0L
    digit = 0;
    portd_index = 0;
    shifter = 1;

    number = 1; // Initial number value
    GIE_bit = 1;
    TMR0IE_bit = 1;
do {
    digit = number / 1000u; // extract thousands digit
    portd_array[3] = mask(digit); // and store it to PORTD array
    digit = (number / 100u) % 10u; // extract hundreds digit
    portd_array[2] = mask(digit); // and store it to PORTD array
    digit = (number / 10u) % 10u; // extract tens digit
    portd_array[1] = mask(digit); // and store it to PORTD array
    digit = number % 10u; // extract ones digit
    portd_array[0] = mask(digit); // and store it to PORTD array
```

```
    Delay_ms(1500);                                // 1.5 seconds delay

for(int i =1; i <= 21, i++){                        //Update number
    number = i*i*i;
    if (number > 9999u){
        number = 0;
    }
}
} while(1);                                         // endless loop
}
```

## References

- [1] [https://en.wikipedia.org/wiki/Special\\_function\\_register](https://en.wikipedia.org/wiki/Special_function_register)
- [2] <http://ww1.microchip.com/downloads/en/DeviceDoc/61120D.pdf>