

EE2004 Microcomputer System
Sem B 2018-19
Assignment
Due Date: March 26

1. Write a program get a value x from PORT B and send the square of the value to PORT C. Assume that the lower nibble of PORTB, RB3-RB0, has the value of 0-9. Use a look-up table for the task, i.e. write a subroutine to return the square of the value x , using look up table approach.

```
ORG    0
SETF   TRISB      ;TRISB = FFh (Port B as input)
CLRF   TRISC      ;TRISC = 00 (Port C as output)
B1     MOVF  PORTB,W    ;read x from Port B into WREG
      ANDLW 0x0F      ;mask upper bits
      CALL  XSQR_TABLE ;get x2 from the look-up table
      MOVWF PORTC      ;copy it to Port C
      BRA   B1         ;continue

;look-up table for square of numbers 0-9
XSQR_TABLE
      MULLW 0x2        ;align it for even address
      MOVFF PRODL, WREG ;put it into WREG for indexing
      ADDWF PCL         ;PCL = PCL + WREG
      RETLW D'0'        ;square of 0
      RETLW D'1'        ;square of 1
      RETLW D'4'        ;square of 2
      RETLW D'9'        ;square of 3
      RETLW D'16'       ;square of 4 (10 hex)
      RETLW D'25'       ;square of 5 (19 hex)
      RETLW D'36'       ;square of 6 (24 hex)
      RETLW D'49'       ;square of 7 (31 hex)
      RETLW D'64'       ;square of 8 (40 hex)
      RETLW D'81'       ;square of 9 (51 hex)
      END
```

2. Write a program to send the corresponding ASCII character to PORT D based on the status of the switches defined by the lower three bits of PORT C connected to three switches as defined as:

000	'0'
001	'1'
010	'2'
011	'3'
100	'4'
101	'5'
110	'6'
111	'7'

Assume that the corresponding ASCII is stored in a look-up table in data RAM location starting at 20H as shown as:

20 = ('0'), 21 = ('1'), 22 = ('2'), 23 = ('3'), 24 = ('4'), 25 = ('5'), 26 = ('6'), 27 = ('7')

In your program, use File Select Register FSR as the pointer to access the look-up table elements.

```

        ORG      0
        SETF     TRISC           ;TRISC = FFh (Port C as input)
        CLRF     TRISD          ;TRISD = 00 (Port D as output)
        LFSR     2,0x20          ;load pointer. FSR2 = 20H, RAM address
B1      MOVF     PORTC,W         ;read x from Port C into WREG
        ANDLW    B'00000111'    ;mask upper 5 bits
        MOVFF    PLUSW2,PORTD    ;get data pointed to by FSR2 + WREG
        BRA      B1
        END
```

3. Write a program to check the status of D7 bit of file register location 0x20H continuously and make the following actions:

- (a) If '0', send "NO" to PORTB, one character at a time ("NO" is a string);
- (b) If '1', send "YES" to PORTB, one character at a time ("YES" is a string)

```
MYREG EQU 0x20
        CLRFB  TRISB          ;make Port B an output port
AGAIN   BTFSS  MYREG,7        ;bit test for HIGH
        BRA    OVER          ;it must be LOW
        MOVLW  A'Y'          ;WREG = 'Y' ASCII letter Y
        MOVWF  PORTB         ;issue WREG to Port B
        MOVLW  A'E'          ;WREG = 'E' ASCII letter Y
        MOVWF  PORTB         ;issue WREG to Port B
        MOVLW  A'S'          ;WREG = 'S' ASCII letter Y
        MOVWF  PORTB         ;issue WREG to Port B
        GOTO   AGAIN         ;we can use BRA too
OVER    MOVLW  A'N'          ;WREG = 'N' ASCII letter N
        MOVWF  PORTB         ;issue WREG to Port B
        MOVLW  A'O'          ;WREG = 'O' ASCII letter Y
        MOVWF  PORTB         ;issue WREG to Port B
        GOTO   AGAIN         ;we can use BRA too
```

4. Write a program to put FFH into 16 RAM locations starting at 160H using :

(a). INCF FSRnL , with n = 0

(b). auto – increment with a loop using BNZ

```
(a)
COUNTREG EQU 0x10      ;fileReg loc for counter
CNTVAL EQU D'16'        ;counter value
    MOVLW CNTVAL         ;WREG = 16
    MOVWF COUNTREG       ;load the counter, count = 16
    LFSR 1,0x160          ;load pointer. FSR1 = 60H, RAM address
    MOVLW 0xFF           ;load 0xFF
B2   MOVWF INDF1,0        ;move W to RAM loc FSR1 points to
    INCF FSR1L            ;increment FSR1L, point to next loc
    DECF COUNTREG,F      ;decrement counter
    BNZ  B2              ;loop until counter = zero
```

```
(b)
COUNTREG EQU 0x10      ;fileReg loc for counter
CNTVAL EQU D'16'        ;counter value
    MOVLW CNTVAL         ;WREG = 16
    MOVWF COUNTREG       ;load the counter, count = 16
    LFSR 1,0x160          ;load pointer. FSR1 = 160H, RAM address
    MOVLW 0xFF           ;load 0xFF
B3   MOVWF POSTINC1, 0    ;decrement counter
    DECF COUNTREG,F      ;decrement counter
    BNZ  B3              ;loop until counter = zero
```

5. Write a program to copy a block of data from RAM location 330H – 33FH to locations 360H-36FH using register indirect addressing mode with FSR0 and FSR2.

```
COUNTREG EQU 0x20      ;fileReg loc 20 for counter
CNTVAL EQU 0x0F        ;counter value = 15
    CLRF TRISB          ;make Port B an output (TRISB = FFH)
    MOVLW CNTVAL         ;WREG = 15
    MOVWF COUNTREG,1    ;load the counter, count = 15
    LFSR 1,0x330          ;load pointer. FSR1 = 330H, RAM address
    LFSR 2,0x360          ;load pointer. FSR2 = 360H, RAM address
B3   MOVFF INDF1, INDF2
    INCF FSR1L            ;increment FSR1 to point at next loc
    INCF FSR2L            ;increment FSR2 to point at next loc
    DECF COUNTREG,F      ;decrement counter
    BNZ  B3              ;loop until counter = zero
```