Laboratory Report #9

Name: **German E Felisarta III** **Group Number: 3**

Laboratory Exercise Title: **Hardware Interrupt Interfacing** **Date Completed:** **11/28/2020**

Exercise 9-1:

Fig 1. Circuit Schematic

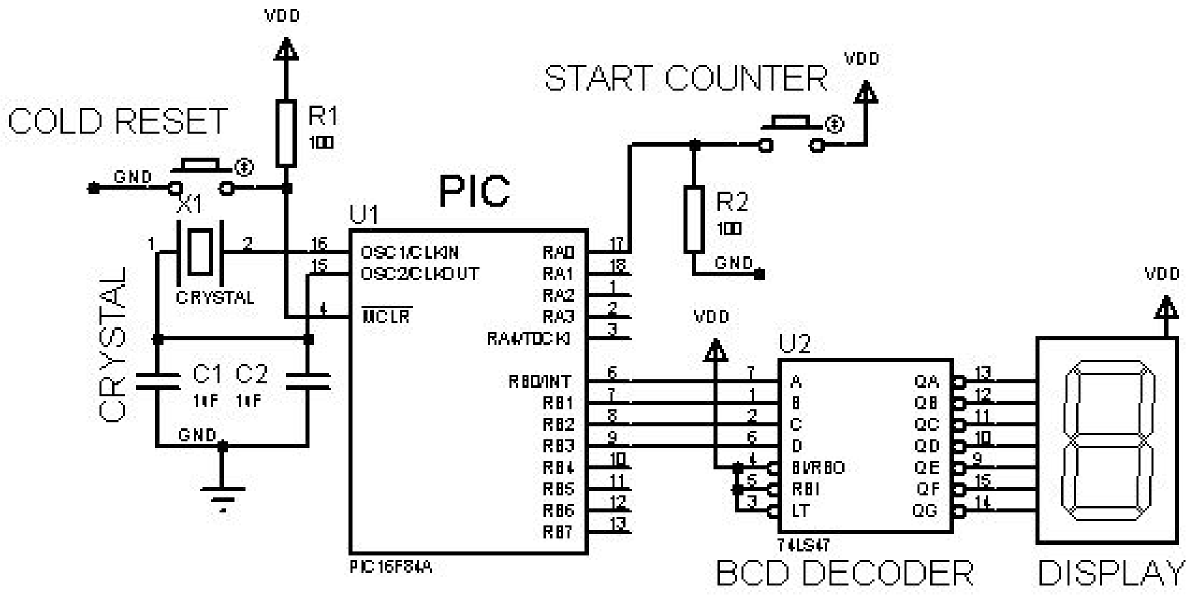


Fig 2. Assembly code for LE9-2

```
;GERMAN E FELISARTA III 16101002   CpE3104 Grp 1

LIST P=16F84A, F=INHX8M                  ; MCU is PIC16F84A, output is Intel Hex
INCLUDE<P16F84A.INC>                     ; include this file to use register names instead of addresses
__CONFIG _CP_OFF & _WDT_ON & _XT_OSC ; code protection off, Watchdog Timer on, XTAL osc used

    ; user-defined register declaration and memory assignment
    COUNT EQU 0CH                        ; assign COUNT to memory address 0CH

    ; initialize PIC
    BSF STATUS, RP0                      ; set register bank to 1
    CLRF TRISB                              ; clear register TRISB (bank 1)
                                            ; sets all bits in PORTB as "output"
    MOVLW 0DH                                       ; load literal value 0DH to Wreg
                                            ; prescaler 1:64 assigned to Watchdog Timer (WDT)
    MOVWF OPTION_REG                                ; move data in Wreg to register OPTION_REG (bank
0/1)
    BCF STATUS, RP0                         ; set register bank to 0

                ; program start
START CLRF PORTB                            ; clear register PORTB (bank 0)
    MOVLW 0AH                                       ; load literal value 0AH to Wreg
    MOVWF COUNT                             ; move data in Wreg to register COUNT

HERE BTFSS PORTA, 0                                 ; check if RB0 (PORTB) if equal to 1, if true skip next
line
   GOTO HERE                                        ; jump to label HERE
                                            ; wait for button press
   MOVLW 09H                                        ; SETS PORTB to 9
   MOVWF PORTB
COUNTDOWN                                   ; Countdown loop from 9-0
   SLEEP                                    ; put CPU to sleep (acts as delay)
                                            ; WDT will wake up CPU at certain time interval
   DECFSZ COUNT, 1                          ; decrement register COUNT, skip next line if result is 0
   GOTO INC                                 ; jump to label INC
   GOTO START                                       ; jump to label START

INC  DECF PORTB                                     ; increment register PORTB
   GOTO COUNTDOWN                                           ; jump to label HERE

END                                 ; end of program
```

Fig 3. System Turns on and displays ‘0’

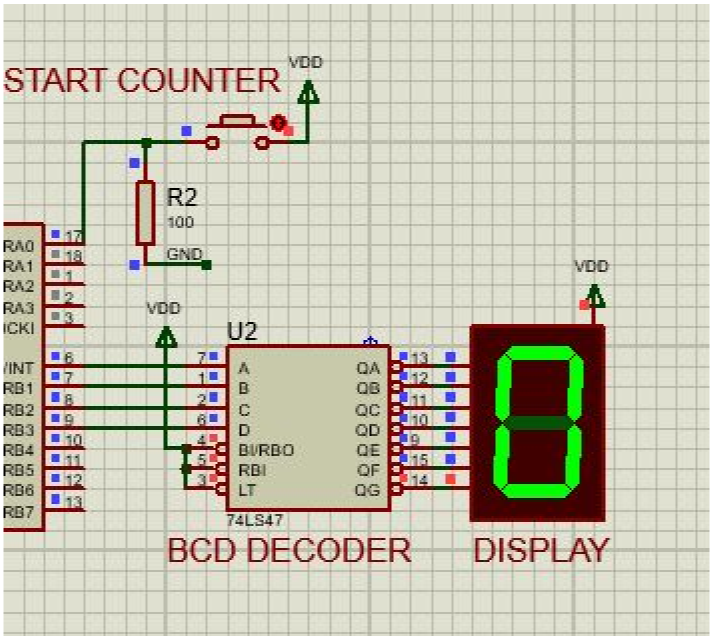


Fig 4. Button Pressed Starts count at 9

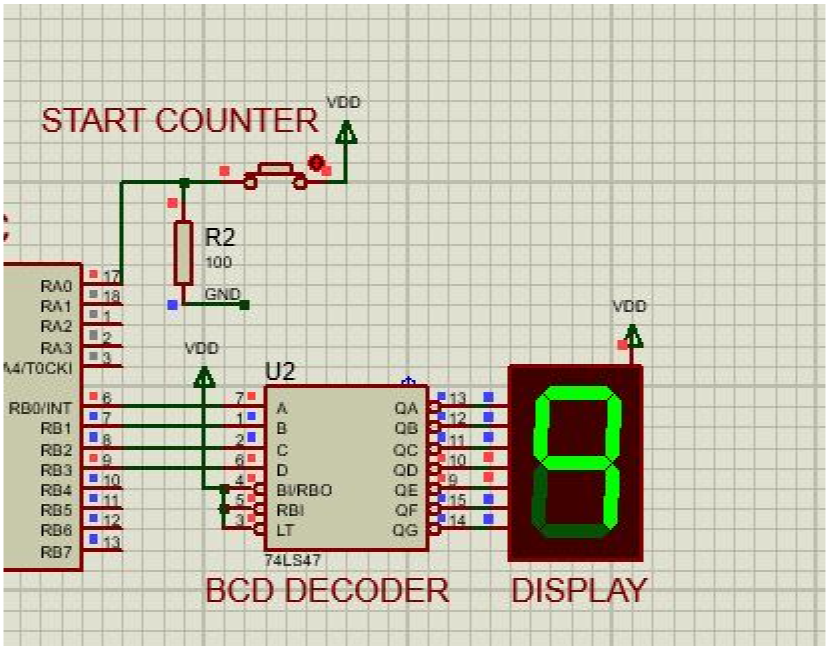
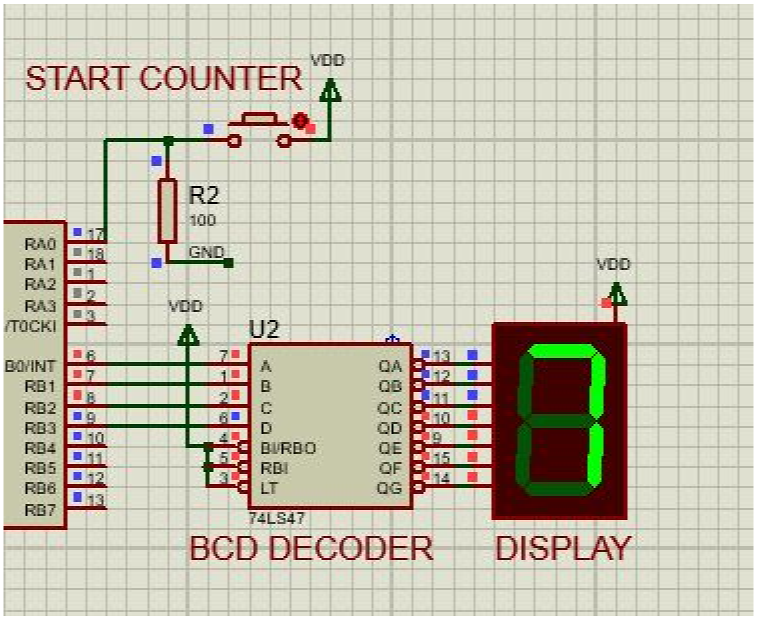


Fig 5. Countdown Continues



Exercise 9-2:

Unlike the 8086 assembly, I have noticed that in order to assign hex values for PORTB, one has to put a 0 on the most significant bist (Leftmost) part of the hex number in order for the PIC to properly display that number into PORTB.

For example: 4FH will not display properly but 04FH will.

For this exercise, I used another Display because I was trying to solve the problem above. But ended up solving it with the zero msb solution. I left the display as is which is a 7SEGMENT-DIGITAL display component in proteus

Fig 1. Circuit Schematic

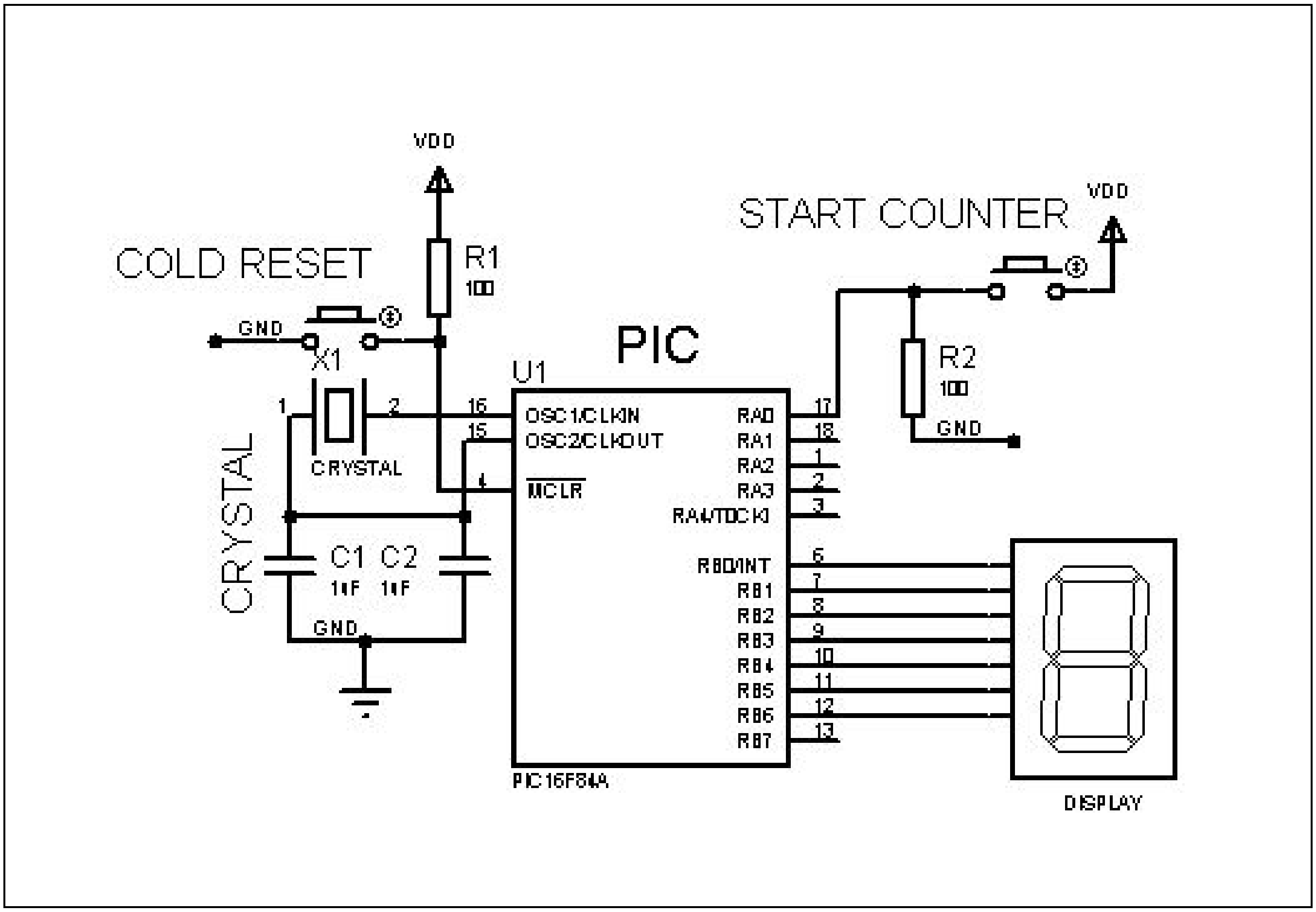


Fig 2. Assembly code for 7-1

```
;GERMAN E FELISARTA III 16101002   CpE3104 Grp 1

    LIST P=16F84A, F=INHX8M                 ; MCU is PIC16F84A, output is Intel Hex
    INCLUDE<P16F84A.INC>                    ; include this file to use register names instead of addresses
    __CONFIG _CP_OFF & _WDT_ON & _XT_OSC ; code protection off, Watchdog Timer on, XTAL osc used

    ; user-defined register declaration and memory assignment

    ; initialize PIC
    BSF STATUS, RP0                         ; set register bank to 1
    CLRF TRISB                              ; clear register TRISB (bank 1)
                                            ; sets all bits in PORTB as "output"
    MOVLW 0DH                                       ; load literal value 0DH to Wreg
                                            ; prescaler 1:64 assigned to Watchdog Timer (WDT)
    MOVWF OPTION_REG                                ; move data in Wreg to register OPTION_REG (bank
0/1)
    BCF STATUS, RP0                         ; set register bank to 0

                ; program start
START CLRF PORTB                            ; clear register PORTB (bank 0)
    MOVLW 03FH                              ; SETS PORTB to 0
    MOVWF PORTB

HERE BTFSS PORTA, 0                                 ; check if RB0 (PORTB) if equal to 1, if true skip next
line
   GOTO HERE                                        ; jump to label HERE
                                            ; wait for button press
   GOTO COUNTDOWN

COUNTDOWN                                   ; Countdown loop from 9-0
   SLEEP                                    ; put CPU to sleep (acts as delay)
                                            ; WDT will wake up CPU at certain time interval

   MOVLW 06FH                                      ; SETS PORTB to 9
   MOVWF PORTB
   SLEEP

   MOVLW 07FH                                      ; SETS PORTB to 8
   MOVWF PORTB
   SLEEP

   MOVLW 007H                                      ; SETS PORTB to 7
   MOVWF PORTB
   SLEEP

   MOVLW 07DH                                      ; SETS PORTB to 6
   MOVWF PORTB
   SLEEP

   MOVLW 06DH                                      ; SETS PORTB to 5
   MOVWF PORTB
   SLEEP
```

```
MOVLW 066H                          ; SETS PORTB to 4
MOVWF PORTB
SLEEP

MOVLW 04FH                          ; SETS PORTB to 3
MOVWF PORTB
SLEEP

MOVLW 05BH                          ; SETS PORTB to 2
MOVWF PORTB
SLEEP

MOVLW 006H                          ; SETS PORTB to 1
MOVWF PORTB
SLEEP

MOVLW 03FH                          ; SETS PORTB to 0
MOVWF PORTB
SLEEP

GOTO HERE                            ; Restart cycle
END                        ; end of program
```

Fig 3. Initial Start Displays 0 and waiting for input

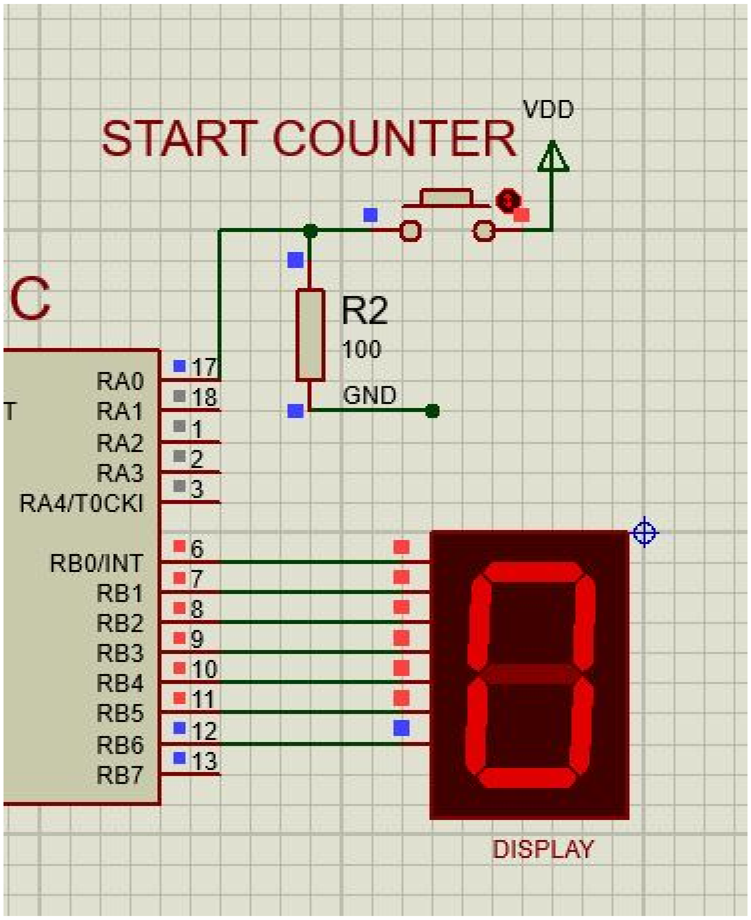


Fig 4. Button is pressed and starts counting from 9

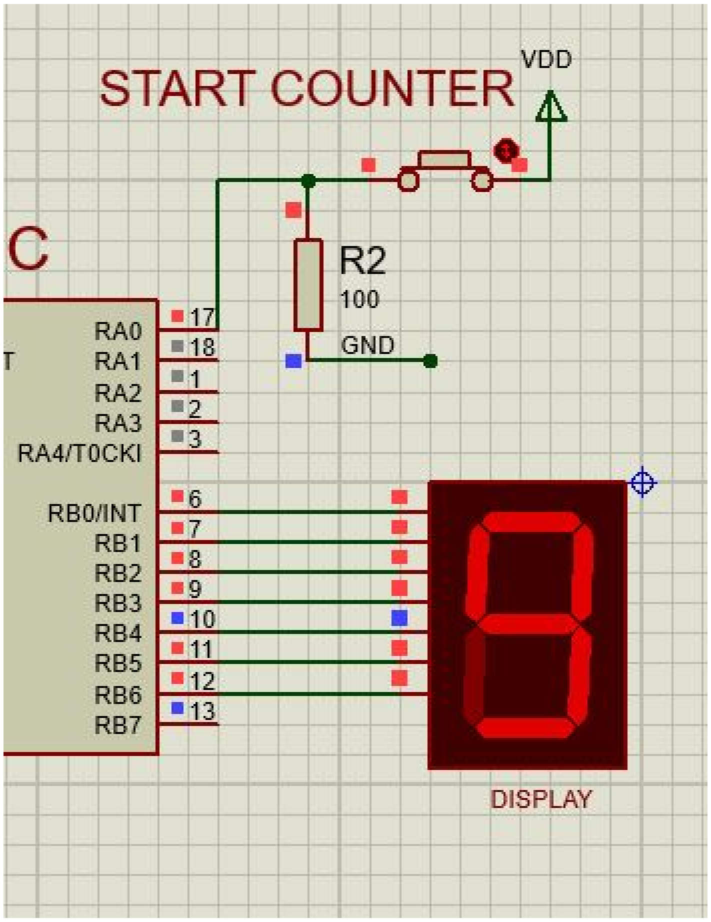
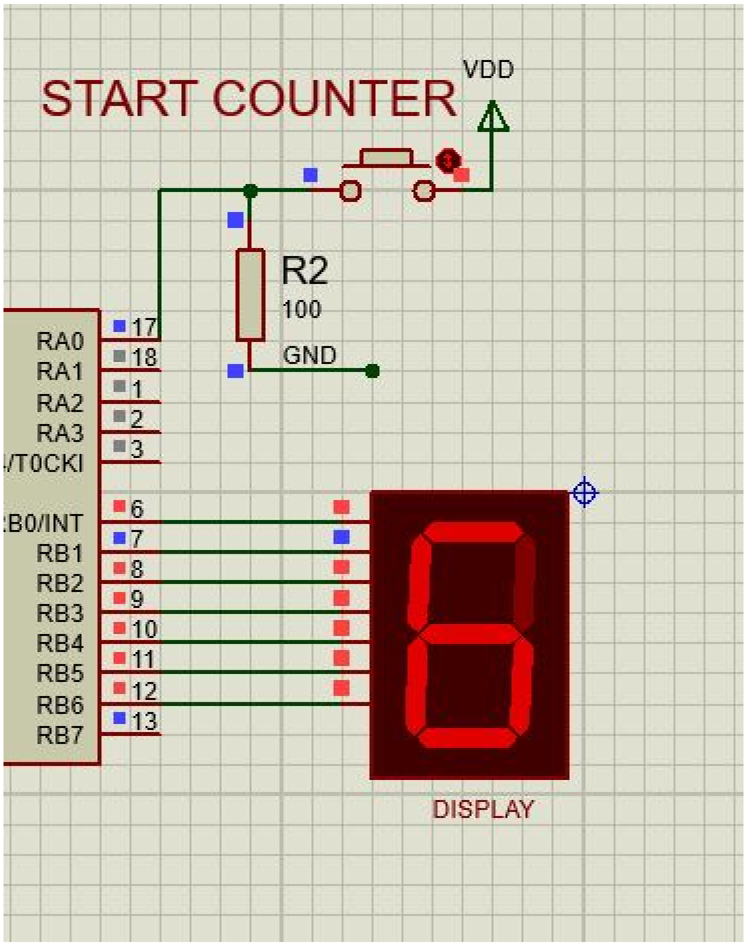


Fig 5. Counting Continues


START COUNTER
VDD
R2
100
GND
RA0
RA1
RA2
RA3
/T0CKI
B0/INT
RB1
RB2
RB3
RB4
RB5
RB6
RB7
17
18
1
2
3
6
7
8
9
10
11
12
13
DISPLAY