```
;
; table_lookup_seg_check.asm
;
; Created: 10/19/2020 5:42:32 PM
; Author : tyler ovenden
; 112122685
;
.nolist
.include "m4809def.inc"
.list

; Replace with your application code
start:
 ; configure I/O ports
    ldi r17, 0xFF  ;load r17 with all 1s
    out VPORTD_DIR, r17  ;PORTD - all pins configured as outputs
    out VPORTD_OUT, r17    ;clears display
    out VPORTE_DIR, r17   ;PORTE - all pins configured as outputs
    ldi r16, 0x00
    out VPORTA_DIR, r16   ;PORTA - all pins configured as inputs

    main_loop:
    sbis VPORTE_IN, 1      ;checks if flip flop is on, button is pushed
    rjmp main_loop         ;goes back to beginning of loop if button released
    rjmp display           ;goes to display if button pushed


    display:
    ldi r18, 0x00                     ;sets r18 to a blank register
    ldi r16, VPORTA_IN                ;loads r16 with switch inputs
    rcall reverse                     ;reverses r16
    rcall hex_to_7seg               ;converts hex number to 7 segment display pattern
    out VPORTD_OUT, r18             ;displays the
    out VPORTC_OUT, 0xEF            ;(1110 1111) sets display to 4th digit
    sbi VPORTE_IN , 1               ;clears flip flop
    rjmp main_loop                  ;goes back to main loop

;***************************************************************************
;*
;* "reverses" - reverses a register
;*
;* Description: Reverses a register using two different registers.
;* shifts r16 then moving that shifted bit into r17 8 times to reverse
;*
;* Author:                      Tyler Ovenden
;* Version:                     1.0
;* Last updated:                102120
;* Target:                      ATmega4809
;* Number of words:             11
;* Number of cycles:
;* Low registers modified:      none
;* High registers modified:     r16, r17
```

```
;*
;* Parameters: r16: input from switch
;* Returns: r16: reversed switch input, shifted 4 times to get only bits 7-4 from
  reversed bit
;*
;* Notes:
;*
;***************************************************************************

    reverse:
    lsr r16                      ;shifts r16 once putting msb in flag
    rol r17                      ;rotates r17 once placing carry bit from lsr into r17
    cpi r16, 0x00                ;checks if r16 is all 0
    brne reverse                 ;if r16 is not 0 then repeat loop
    mov r16, r17                 ;moves reversed number in r17 to r16
    lsr r16                       ;shifts r16 4 times to get only 4 bits
    lsr r16
    lsr r16
    lsr r16
    ret                          ;ends subroutine
    ;***************************************************************************
;*
;* "hex_to_7seg" - Hexadecimal to Seven Segment Conversion
;*
;* Description: Converts a right justified hexadecimal digit to the seven
;* segment pattern required to display it. Pattern is right justified a
;* through g. Pattern uses 0s to turn segments on ON.
;*
;* Author:                      Ken Short
;* Version:                     1.0
;* Last updated:                101620
;* Target:                      ATmega4809
;* Number of words:             8
;* Number of cycles:            13
;* Low registers modified:      none
;* High registers modified:     r16, r18, ZL, ZH
;*
;* Parameters: r18: right justified hex digit, high nibble 0
;* Returns: r18: segment values a through g right justified
;*
;* Notes:
;*
;***************************************************************************
hex_to_7seg:
    andi r18, 0x0F              ;clear ms nibble
    ldi ZH, HIGH(hextable * 2)  ;set Z to point to start of table
    ldi ZL, LOW(hextable * 2)
    ldi r16, $00                ;add offset to Z pointer
    add ZL, r18
    adc ZH, r16
    lpm r18, Z                  ;load byte from table pointed to by Z
    ret
    ;Table of segment values to display digits 0 - F
    ;!!! seven values must be added - verify all values
hextable: .db $01, $4F, $12, $06, $4C, $24, $20, $0F, $00, $04, $08, $60, $31, $32, $30, $38
```