```
;***********************************************************************
;*
;* Title:           table_lookup_seg_test
;* Author:          Jason Chen
;* Version:         2
;* Last updated:    10/12/2022 12:02:00
;* Target:          AVR128DB48
;*
;* DESCRIPTION
;*  Task 2
;*  Unconditionally reads a hexadecimal digit from DIP switches
;*  and displays to the right most digit on a 4-digit 7-segment display
;*
;* VERSION HISTORY
;* 1.0 Original version
;***********************************************************************

start:
    ldi r16, 0xFF       ; load r16 with all 1s
    out VPORTD_DIR, r16 ; VPORTD - all pins configured as outputs
    out VPORTA_DIR, r16 ; VPORTA - all pins configured as outputs
    ldi r16, 0x00       ; load r16 with all 0s
    out VPORTC_DIR, r16 ; VPORTC - all pins configured as inputs
    cbi VPORTE_DIR, 0   ; set direction for PE0 as input
    sbi VPORTE_DIR, 1   ; set direction for PE1 as output
    cbi VPORTA_OUT, 4   ; set PA4 to output 0, turns rightmost digit ON

main_loop:
    in r17, VPORTC_IN   ; read in switches
    rcall reverse_bits
    rcall hex_to_7seg
    out VPORTD_OUT, r18 ; output to 7-seg display
    rjmp main_loop

;***********************************************************************
;*
;* "reverse_bits" - Reverse bits
;*
;* Description: Reverse the order of bits register 17, which reads the input
;*              switches, into register 18.
;*
;* Author:          Jason Chen
;* Version:         1
;* Last updated:    10/13/2022
;* Target:          AVR128DB48
;* Number of words:
;* Number of cycles:        8
;* Low registers modified:  n/a
;* High registers modified: r16, r18
;*
;* Parameters:  r17 - switch input
;*
```

```
53  ;* Returns:     r18 - reversed bits
54  ;*
55  ;* Notes:
56  ;*
57  ;***************************************************************************
58
59  reverse_bits:           ; reverses bits from r17 into r18
60      ldi r18, 0x00
61      ldi r16, 0x08       ; 8 step counter
62      bits_loop:
63          lsl r17
64          ror r18
65          dec r16
66          cpi r16, 0x00
67          brne bits_loop
68      ret
69
70  ;***************************************************************************
71  ;*
72  ;* "hex_to_7seg" - Hexadecimal to Seven Segment Conversion
73  ;*
74  ;* Description: Converts a right justified hexadecimal digit to the seven
75  ;* segment pattern required to display it. Pattern is right justified a
76  ;* through g. Pattern uses 0s to turn segments on ON.
77  ;*
78  ;* Author:          Ken Short
79  ;* Version:         0.1
80  ;* Last updated:    10/03/2022
81  ;* Target:          AVR128DB48
82  ;* Number of words:         1
83  ;* Number of cycles:        1
84  ;* Low registers modified:  n/a
85  ;* High registers modified: r16, r18
86  ;*
87  ;* Parameters: r18: hex digit to be converted
88  ;* Returns: r18: seven segment pattern. 0 turns segment ON
89  ;*
90  ;* Notes:
91  ;*
92  ;***************************************************************************
93
94  hex_to_7seg:
95      ldi ZH, HIGH(hextable * 2)  ; set Z to point to start of table
96      ldi ZL, LOW(hextable * 2)
97      ldi r16, $00                ; add offset to Z pointer
98      andi r18, 0x0F              ; mask for low nibble
99      add ZL, r18
100     adc ZH, r16
101     lpm r18, Z                  ; load byte from table pointed to by Z
102     ret
103
104     ; Table of segment values to display digits 0 - F
```

```
105       ; !!! seven values must be added
106  hextable: .db $01, $4F, $12, $06, $4C, $24, $20, $0F, $00, $04, $08, $60, $31,
       $42, $30, $38
```