```
;***********************************************************************
;*
;* Title:           display_hex_digit_at_pos
;* Author:          Jason Chen
;* Version:         2
;* Last updated:    10/13/2022 18:47:00
;* Target:          AVR128DB48
;*
;* DESCRIPTION
;*  Task 3
;*  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
    ldi r21, 0xFF
    mov r22, r21
    mov r23, r21
    mov r24, r21        ; set r21 - r24 to all 1s, all segments are initially OFF

main_loop:
    sbi VPORTE_OUT, 1   ; unclear DFF
    rcall turn_off_all
    ldi r19, 0xEF       ; load r19 with 1110 1111

digit_4_blink:          ; display hex at digit 4 / pos 0 ON (rightmost)
    out VPORTD_OUT, r24
    out VPORTA_OUT, r19
    rcall var_delay
    rcall turn_off_all

digit_3_blink:
    lsl r19             ; r19 is now 1101 1110, digit 3 / pos 1 will turn ON
    out VPORTD_OUT, r23
    out VPORTA_OUT, r19
    rcall var_delay
    rcall turn_off_all

digit_2_blink:
    lsl r19             ; r19 is now 1011 1100, digit 2 / pos 2 will turn ON
    out VPORTD_OUT, 22
    out VPORTA_OUT, r19
```

```
53      rcall var_delay
54      rcall turn_off_all
55
56  digit_1_blink:
57      lsl r19             ; digit 1 / pos 3 will turn ON (leftmost)
58      out VPORTD_OUT, 21
59      out VPORTA_OUT, r19
60      rcall var_delay
61      rcall turn_off_all
62
63  check_if_1:             ; wait for the Q from the DFF
64      sbis VPORTE_IN, 0
65      rjmp main_loop
66      cbi VPORTE_OUT, 1   ; clear DFF
67
68  update_digits:
69      in r17, VPORTC_IN   ; read switches
70      rcall reverse_bits
71      mov r17, r18
72      rcall hex_to_7seg
73      andi r17, 0xC0      ; mask for assigning digit
74      cpi r17, 0xC0
75      breq update_digit_1 ; update digit 1 / pos 3 with new hex
76      cpi r17, 0x80
77      breq update_digit_2
78      cpi r17, 0x40
79      breq update_digit_3
80
81  update_digit_4:         ; update digit 4 / pos 0 (default)
82      mov r24, r18
83      rjmp main_loop
84
85  update_digit_1:
86      mov r21, r18
87      rjmp main_loop
88
89  update_digit_2:
90      mov r22, r18
91      rjmp main_loop
92
93  update_digit_3:
94      mov r23, r18
95      rjmp main_loop
96
97  ;***************************************************************************
98  ;*
99  ;* "turn_off_all" - Turn OFF All Segments and Digit's Transistors
100 ;*
101 ;* Description: Delays a variable time that is adjusted by need basis.
102 ;*
103 ;* Author:          Jason Chen
104 ;* Version:         1
```

```
105 ;* Last updated:    10/18/2022
106 ;* Target:          AVR128DB48
107 ;* Number of words:         4
108 ;* Number of cycles:        1
109 ;* Low registers modified:  n/a
110 ;* High registers modified: r16
111 ;*
112 ;* Parameters:  r16 - set all 1s
113 ;*
114 ;* Returns:     r16 - set all 1s
115 ;*
116 ;* Notes:
117 ;*
118 ;***************************************************************************
119
120 turn_off_all:
121     ldi r16, 0xFF
122     out VPORTD_OUT, r16 ; turn all segments OFF
123     out VPORTA_OUT, r16 ; turn all transistors/digits OFF
124     ret
125
126 ;***************************************************************************
127 ;*
128 ;* "var_delay" - Variable Delay
129 ;*
130 ;* Description: Delays r16 * 1ms (approx.)
131 ;*
132 ;* Author:          Jason Chen
133 ;* Version:         1
134 ;* Last updated:    10/13/2022
135 ;* Target:          AVR128DB48
136 ;* Number of words:         7
137 ;* Number of cycles:        r16 * r17
138 ;* Low registers modified:  n/a
139 ;* High registers modified: r16, r17
140 ;*
141 ;* Parameters:  r16 - outer loop counter
142 ;*              r17 - inner loop counter
143 ;* Returns:     r16 and r17 set to all 0s
144 ;*
145 ;* Notes:
146 ;*
147 ;***************************************************************************
148
149 var_delay:
150     ldi r16, 0x01
151     outer_loop:
152         ldi r17, 133
153     inner_loop:
154         dec r17
155         brne inner_loop
156         dec r16
```

```
157         brne outer_loop
158     ret
159
160 ;***********************************************************************
161 ;*
162 ;* "reverse_bits" - Reverse bits
163 ;*
164 ;* Description: Reverse the order of bits register 17, which reads the input
165 ;*              switches, into register 18.
166 ;*
167 ;* Author:          Jason Chen
168 ;* Version:         1
169 ;* Last updated:    10/13/2022
170 ;* Target:          AVR128DB48
171 ;* Number of words:
172 ;* Number of cycles:        8
173 ;* Low registers modified:  n/a
174 ;* High registers modified: r16, r18
175 ;*
176 ;* Parameters:  r17 - switch input to be read and reversed
177 ;*              r16 - 8 step counter
178 ;* Returns:     r18 - reversed bits
179 ;*
180 ;* Notes:
181 ;*
182 ;***********************************************************************
183
184 reverse_bits:           ; reverses bits from r17 into r18
185     ldi r18, 0x00
186     ldi r16, 0x08       ; 8 step counter
187     bits_loop:
188         lsl r17
189         ror r18
190         dec r16
191         cpi r16, 0x00
192         brne bits_loop
193     ret
194
195 ;***********************************************************************
196 ;*
197 ;* "hex_to_7seg" - Hexadecimal to Seven Segment Conversion
198 ;*
199 ;* Description: Converts a right justified hexadecimal digit to the seven
200 ;* segment pattern required to display it. Pattern is right justified a
201 ;* through g. Pattern uses 0s to turn segments on ON.
202 ;*
203 ;* Author:          Ken Short
204 ;* Version:         0.1
205 ;* Last updated:    10/03/2022
206 ;* Target:          AVR128DB48
207 ;* Number of words:         1
208 ;* Number of cycles:        1
```

```
209 ;* Low registers modified:  n/a
210 ;* High registers modified: r16, r18
211 ;*
212 ;* Parameters:  r18 - hex digit to be converted
213 ;* Returns:     r18 - seven segment pattern. 0 turns segment ON
214 ;*
215 ;* Notes:
216 ;*
217 ;***************************************************************************
218
219 hex_to_7seg:
220     ldi ZH, HIGH(hextable * 2)  ; set Z to point to start of table
221     ldi ZL, LOW(hextable * 2)
222     ldi r16, $00                ; add offset to Z pointer
223     andi r18, 0x0F              ; mask for low nibble
224     add ZL, r18
225     adc ZH, r16
226     lpm r18, Z                  ; load byte from table pointed to by Z
227     ret
228
229     ; Table of segment values to display digits 0 - F
230     ; !!! seven values must be added
231 hextable: .db $01, $4F, $12, $06, $4C, $24, $20, $0F, $00, $04, $08, $60, $31,    ↵
      $42, $30, $38   ; dp a b c d e f g
232 ;hextable: .db $40, $79, $24, $30, $19, $12, $02, $78, $00, $10, $08, $03, $46,  ↵
      $21, $06, $0E   ; dp g f e d c b a
```