

```
1
2 AVRASM ver. 2.2.7 F:\ESE_280\MyDocuments\Atmel Studio\7.0\lab_7
  \AssemblerApplication1\main.asm Wed Oct 21 11:23:43 2020
3
4 F:\ESE_280\MyDocuments\Atmel Studio\7.0\lab_7\AssemblerApplication1\main.asm
  (9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel
  \ATmega_DFP\1.2.209\avrasm\inc\m4809def.inc'
5 F:\ESE_280\MyDocuments\Atmel Studio\7.0\lab_7\AssemblerApplication1\main.asm
  (9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel
  \ATmega_DFP\1.2.209\avrasm\inc\m4809def.inc'
6
7
8 ; display_hex_digit_at_pos.asm
9 ;
10 ; Created: 10/20/2020 8:30:20 PM
11 ; Author : hp
12 ;
13
14 .list
15
16 start:
17 000000 e000 ldi r16, $00
18 000001 ef1f ldi r17, $FF
19 000002 b900 out VPORTA_DIR, r16
20 000003 b91c out VPORTD_DIR, r17
21 000004 b918 out VPORTC_DIR, r17
22 000005 b91d out VPORTD_OUT, r17
23
24 main_loop:
25 000006 b102 in r16, VPORTA_IN
26 000007 1710 cp r17, r16
27 000008 1720 cp r18, r16
28 000009 7c10 andi r17, $C0
29 00000a 700f andi r16, $0F
30 00000b 3010 cpi r17, $00
31 00000c f039 breq zero
32 00000d 3410 cpi r17, $40
33 00000e f039 breq one
34 00000f 3810 cpi r17, $80
35 000010 f039 breq two
36 000011 3c10 cpi r17, $C0
37 000012 f039 breq three
38 000013 cff2 rjmp main_loop
39
40 zero:
41 000014 984f cbi VPORTC_OUT, 7
42 000015 c005 rjmp hex_to_7seg
43
44 one:
```

```

45 000016 984e          cbi VPORTC_OUT, 6
46 000017 c003          rjmp hex_to_7seg
47
48                      two:
49 000018 984d          cbi VPORTC_OUT, 5
50 000019 c001          rjmp hex_to_7seg
51
52                      three:
53 00001a 984c          cbi VPORTC_OUT, 4
54
55
56
57                      ;*****
*****
58                      ;*
59                      ;* "hex_to_7seg" - Hexadecimal to Seven
Segment Conversion
60                      ;*
61                      ;* Description: Converts a right justified
hexadecimal digit to the seven
62                      ;* segment pattern required to display it.
Pattern is right justified a
63                      ;* through g. Pattern uses 0s to turn
segments on ON.
64                      ;*
65                      ;* Author:          Ken Short
66                      ;* Version:         1.0
67                      ;* Last updated:    101620
68                      ;* Target:         ATmega4809
69                      ;* Number of words:      8
70                      ;* Number of cycles:    13
71                      ;* Low registers modified: none
72                      ;* High registers modified: r16, r18,
ZL, ZH
73                      ;*
74                      ;* Parameters: r18: right justified hex
digit, high nibble 0
75                      ;* Returns: r18: segment values a through g
right justified
76                      ;*
77                      ;* Notes:
78                      ;*
79                      ;*****
*****
80                      hex_to_7seg:
81 00001b 702f          andi r18, 0x0F          ;clear ms
nibble
82 00001c e0f0          ldi ZH, HIGH(hextable * 2) ;set Z to

```

```

    point to start of table
83 00001d e4e6          ldi ZL, LOW(hextable * 2)
84 00001e e000          ldi r16, $00          ;add offset ↗
    to Z pointer
85 00001f 0fe2          add ZL, r18
86 000020 1ff0          adc ZH, r16
87 000021 9124          lpm r18, Z          ;load byte ↗
    from table pointed to by Z
88 000022 c008          rjmp output
89
90                      ;Table of segment values to display ↗
    digits 0 - F
91                      ;!!! seven values must be added - verify ↗
    all values
92 000023 4f01
93 000024 0612
94 000025 244c
95 000026 0f20
96 000027 0400
97 000028 6008
98 000029 4231
99 00002a 3830          hextable: .db $01, $4F, $12, $06, $4C, $24, ↗
    $20, $0F, $00, $04, $08, $60, $31, $42, $30, $38
100
101                      output:
102 00002b b92d          out VPORTD_OUT, r18
103
104
105 RESOURCE USE INFORMATION
106 -----
107
108 Notice:
109 The register and instruction counts are symbol table hit counts,
110 and hence implicitly used resources are not counted, eg, the
111 'lpm' instruction without operands implicitly uses r0 and z,
112 none of which are counted.
113
114 x,y,z are separate entities in the symbol table and are
115 counted separately from r26..r31 here.
116
117 .dseg memory usage only counts static data declared with .byte
118
119 "ATmega4809" register use summary:
120 x : 0 y : 0 z : 1 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0
121 r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
122 r13: 0 r14: 0 r15: 0 r16: 8 r17: 10 r18: 5 r19: 0 r20: 0
123 r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 0 r27: 0 r28: 0
124 r29: 0 r30: 2 r31: 2
125 Registers used: 6 out of 35 (17.1%)

```

126

127 "ATmega4809" instruction use summary:

```

128 .lds : 0 .sts : 0 adc : 1 add : 1 adiw : 0 and : 0
129 andi : 3 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
130 brcc : 0 brcs : 0 break : 0 breq : 4 brge : 0 brhc : 0
131 brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
132 brne : 0 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
133 brvs : 0 bset : 0 bst : 0 call : 0 cbi : 4 cbr : 0
134 clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
135 clt : 0 clv : 0 clz : 0 com : 0 cp : 2 cpc : 0
136 cpi : 4 cpse : 0 dec : 0 des : 0 eor : 0 fmul : 0
137 fmuls : 0 fmulsu : 0 icall : 0 ijmp : 0 in : 1 inc : 0
138 jmp : 0 ld : 0 ldd : 0 ldi : 5 lds : 0 lpm : 2
139 lsl : 0 lsr : 0 mov : 0 movw : 0 mul : 0 muls : 0
140 mulsu : 0 neg : 0 nop : 0 or : 0 ori : 0 out : 5
141 pop : 0 push : 0 rcall : 0 ret : 0 reti : 0 rjmp : 6
142 rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
143 sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
144 seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
145 sev : 0 sez : 0 sleep : 0 spm : 0 st : 0 std : 0
146 sts : 0 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0

```

147

148 Instructions used: 12 out of 114 (10.5%)

149

150 "ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00005a	74	16	90	49152	0.2%
[.dseg]	0x002800	0x002800	0	0	0	6144	0.0%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

156

157 Assembly complete, 0 errors, 0 warnings

158