

Judah Ben-Eliezer

112352727

10/27/2020

Prelab 8:

Memory Reference Instructions and Using Subroutines for Program Modularity

```
1
2 AVRASM ver. 2.2.7 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8
  \enable_pullups_inven\enable_pullups_inven\main.asm Tue Oct 27 18:39:56 2020
3
4 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8\enable_pullups_inven
  \enable_pullups_inven\main.asm(20): Including file 'C:/Program Files (x86)
  \Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
5 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8\enable_pullups_inven
  \enable_pullups_inven\main.asm(20): Including file 'C:/Program Files (x86)
  \Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
6
7
8      ;*
9      ;* Title: enable_pullups_inven.asm
10     ;* Author: Judah Ben-Eliezer
11     ;* Version: 1.0
12     ;* Last updated: 10/27/2020
13     ;* Target:
14     ;*
15     ;* DESCRIPTION
16     ;*
17     ;*
18     ;*
19     ;*
20     ;* VERSION HISTORY
21     ;* 1.0 Original version
22     ;* *****
    *****
23
24
25     .list
26
27
28     ; Replace with your application code
29     start:
30         ldi r16, $00
31         out VPORTA_DIR, r16
32         ldi XH, HIGH(PORTA_PIN0CTRL)
33         ldi XL, LOW(PORTA_PIN0CTRL)
34         ldi r17, 8
35
36     main_loop:
37         rcall pullups
38         rjmp main_loop
39
40     ;*****
    *****
41
42     ;*
43     ;* "pullups" - title
```

```

43      ;*
44      ;* Description:
45      ;*
46      ;* Author:  Judah Ben-Eliezer
47      ;* Version:   1.0
48      ;* Last updated:  10/27/2020
49      ;* Target:  AtMega4809
50      ;* Number of words:
51      ;* Number of cycles:
52      ;* Low registers modified:
53      ;* High registers modified:
54      ;*
55      ;* Parameters: none
56      ;* Returns:   none
57      ;*
58      ;* Notes:
59      ;*
60      ;*****
        *****
61
62      pullups:
63      000007 910c      ld r16, X
64      000008 6808      ori r16, $88
65      000009 930d      st X+, r16
66      00000a 951a      dec r17
67      00000b f7d9      brne pullups
68      00000c cff8      rjmp main_loop
69
70
71
72
73
74
75  RESOURCE USE INFORMATION
76  -----
77
78  Notice:
79  The register and instruction counts are symbol table hit counts,
80  and hence implicitly used resources are not counted, eg, the
81  'lpm' instruction without operands implicitly uses r0 and z,
82  none of which are counted.
83
84  x,y,z are separate entities in the symbol table and are
85  counted separately from r26..r31 here.
86
87  .dseg memory usage only counts static data declared with .byte
88
89  "ATmega4809" register use summary:
90  x  :   2  y  :   0  z  :   0  r0 :   0  r1 :   0  r2 :   0  r3 :   0  r4 :   0

```

```

91 r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
92 r13: 0 r14: 0 r15: 0 r16: 5 r17: 2 r18: 0 r19: 0 r20: 0
93 r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 1 r27: 1 r28: 0
94 r29: 0 r30: 0 r31: 0

```

95 Registers used: 5 out of 35 (14.3%)

96

97 "ATmega4809" instruction use summary:

```

98 .lds : 0 .sts : 0 adc : 0 add : 0 adiw : 0 and : 0
99 andi : 0 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
100 brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
101 brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
102 brne : 1 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
103 brvs : 0 bset : 0 bst : 0 call : 0 cbi : 0 cbr : 0
104 clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
105 clt : 0 clv : 0 clz : 0 com : 0 cp : 0 cpc : 0
106 cpi : 0 cpse : 0 dec : 1 des : 0 eor : 0 fmul : 0
107 fmul : 0 fmul : 0 icall : 0 ijmp : 0 in : 0 inc : 0
108 jmp : 0 ld : 1 ldd : 0 ldi : 4 lds : 0 lpm : 0
109 lsl : 0 lsr : 0 mov : 0 movw : 0 mul : 0 muls : 0
110 mulsu : 0 neg : 0 nop : 0 or : 0 ori : 1 out : 1
111 pop : 0 push : 0 rcall : 1 ret : 0 reti : 0 rjmp : 2
112 rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
113 sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
114 seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
115 sev : 0 sez : 0 sleep : 0 spm : 0 st : 1 std : 0
116 sts : 0 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0

```

117

118 Instructions used: 9 out of 114 (7.9%)

119

120 "ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00001a	26	0	26	49152	0.1%
[.dseg]	0x002800	0x002800	0	0	0	6144	0.0%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

126

127 Assembly complete, 0 errors, 0 warnings

128

```

1
2 AVRASM ver. 2.2.7 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8
  \multiplex_display\multiplex_display\main.asm Tue Oct 27 20:41:12 2020
3
4 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8\multiplex_display
  \multiplex_display\main.asm(20): Including file 'C:/Program Files (x86)\Atmel
  \Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
5 E:\ESE_280\MyDocuments$\Atmel Studio\7.0\lab_8\multiplex_display
  \multiplex_display\main.asm(20): Including file 'C:/Program Files (x86)\Atmel
  \Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
6
7
8 ;*
9 ;* Title: multiplex_display.asm
10 ;* Author: Judah Ben-Eliezer
11 ;* Version: 1.0
12 ;* Last updated: 10/27/2020
13 ;* Target: ATmega4809
14 ;*
15 ;* DESCRIPTION
16 ;*
17 ;*
18 ;*
19 ;*
20 ;* VERSION HISTORY
21 ;* 1.0 Original version
22 ;* *****
    *****
23
24
25 .list
26
27
28 .dseg
29
30 002800 led_display: .byte 4
31 002804 digit_num: .byte 1
32
33
34 .cseg
35
36 start:
37 000000 ef0f ldi r16, $FF
38 000001 e010 ldi r17, $00
39 000002 b90c out VPORTD_DIR, r16
40 000003 b909 out VPORTC_OUT, r16
41 000004 9310 2804 sts digit_num, r17
42 000006 e2b8 ldi XH, HIGH(led_display)
43 000007 e0a0 ldi XL, LOW(led_display)

```

```

44
45             main_loop:
46 000008 d001             rcall multiplex_display
47 000009 cffe             rjmp main_loop
48
49             ;*****
*****
50             ;*
51             ;* "multiplex_display" - Multiplex the Four Digit LED Display
52             ;*
53             ;* Description: Updates a single digit of the display and increments the
54             ;* digit_num to the value of the digit position to be displayed next.
55             ;*
56             ;* Author: Judah Ben-Eliezer
57             ;* Version: 1.0
58             ;* Last updated: 10/27/2020
59             ;* Target: ;ATmega4809 @ 3.3MHz
60             ;* Number of words: 39
61             ;* Number of cycles: 30
62             ;* Low registers modified: none
63             ;* High registers modified: none
64             ;*
65             ;* Parameters:
66             ;* led_display: a four byte array that holds the segment values
67             ;* for each digit of the display. led_display[0] holds the segment pattern
68             ;* for digit 0 (the rightmost digit) and so on.
69             ;* digit_num: a byte variable, the least significant two bits provide the
70             ;* index of the next digit to be displayed.
71             ;*
72             ;* Returns: Outputs segment pattern and turns on digit driver for the next
73             ;* position in the display to be turned ON in the multiplexing sequence.
74             ;*
75             ;* Notes:
76             ;*
77             ;*****
*****
78
79
80             multiplex_display:

```

```

81 00000a 9110 2804      lds r17, digit_num
82 00000c 9140 2804      lds r20, digit_num
83 00000e 7013          andi r17, $03
84 00000f 0fa1          add XL, r17
85 000010 912c          ld r18, X
86 000011 e056          ldi r21, $06
87                          loop:
88 000012 0f44          lsl r20
89 000013 955a          dec r21
90 000014 f7e9          brne loop
91
92 000015 d006          rcall hex_to_7seg
93 000016 b92d          out VPORTD_OUT, r18
94 000017 b949          out VPORTC_OUT, r20
95 000018 9513          inc r17
96 000019 9310 2804      sts digit_num, r17
97 00001b 9508          ret
98
99
100
101
102                          ;*****
*****
103                          ;*
104                          ;* "hex_to_7seg" - Hexadecimal to Seven
Segment Conversion
105                          ;*
106                          ;* Description: Converts a right justified
hexadecimal digit to the seven
107                          ;* segment pattern required to display it.
Pattern is right justified a
108                          ;* through g. Pattern uses 0s to turn segments
on ON.
109                          ;*
110                          ;* Author:                      Ken Short
111                          ;* Version:                      1.0
112                          ;* Last updated:                101620
113                          ;* Target:                      ATmega4809
114                          ;* Number of words:              8
115                          ;* Number of cycles:             13
116                          ;* Low registers modified:       none
117                          ;* High registers modified:      r16, r18,
ZL, ZH
118                          ;*
119                          ;* Parameters: r18: right justified hex digit,
high nibble 0
120                          ;* Returns: r18: segment values a through g
right justified

```

```

121                                     ;*
122                                     ;* Notes:
123                                     ;*
124                                     ;*****
                                     *****
125
126                                     hex_to_7seg:
127 00001c 702f                         andi r18, 0x0F                ;clear ms
                                     nibble
128 00001d e0f0                         ldi ZH, HIGH(hextable * 2) ;set Z to
                                     point to start of table
129 00001e e4e8                         ldi ZL, LOW(hextable * 2)
130 00001f e000                         ldi r16, $00                ;add offset to
                                     Z pointer
131 000020 0fe2                         add ZL, r18
132 000021 1ff0                         adc ZH, r16
133 000022 9124                         lpm r18, Z                ;load byte
                                     from table pointed to by Z
134 000023 9508                         ret
135
136                                     ;Table of segment values to display digits
                                     0 - F
137                                     ;!!! seven values must be added - verify
                                     all values
138 000024 4f01
139 000025 0612
140 000026 244c
141 000027 0f20
142 000028 0400
143 000029 6008
144 00002a 4231
145 00002b 3830                         hextable: .db $01, $4F, $12, $06, $4C, $24,
                                     $20, $0F, $00, $04, $08, $60, $31, $42, $30, $38
146
147
148 RESOURCE USE INFORMATION
149 -----
150
151 Notice:
152 The register and instruction counts are symbol table hit counts,
153 and hence implicitly used resources are not counted, eg, the
154 'lpm' instruction without operands implicitly uses r0 and z,
155 none of which are counted.
156
157 x,y,z are separate entities in the symbol table and are
158 counted separately from r26..r31 here.
159
160 .dseg memory usage only counts static data declared with .byte
161

```


162 "ATmega4809" register use summary:

```
163 x : 1 y : 0 z : 1 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0
164 r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
165 r13: 0 r14: 0 r15: 0 r16: 5 r17: 7 r18: 5 r19: 0 r20: 3
166 r21: 2 r22: 0 r23: 0 r24: 0 r25: 0 r26: 2 r27: 1 r28: 0
167 r29: 0 r30: 2 r31: 2
```

168 Registers used: 11 out of 35 (31.4%)

169

170 "ATmega4809" instruction use summary:

```
171 .lds : 0 .sts : 0 adc : 1 add : 2 adiw : 0 and : 0
172 andi : 2 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
173 brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
174 brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
175 brne : 1 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
176 brvs : 0 bset : 0 bst : 0 call : 0 cbi : 0 cbr : 0
177 clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
178 clt : 0 clv : 0 clz : 0 com : 0 cp : 0 cpc : 0
179 cpi : 0 cpse : 0 dec : 1 des : 0 eor : 0 fmul : 0
180 fmul : 0 fmul : 0 icall : 0 ijmp : 0 in : 0 inc : 1
181 jmp : 0 ld : 1 ldd : 0 ldi : 8 lds : 2 lpm : 2
182 lsl : 1 lsr : 0 mov : 0 movw : 0 mul : 0 muls : 0
183 mulsu : 0 neg : 0 nop : 0 or : 0 ori : 0 out : 4
184 pop : 0 push : 0 rcall : 2 ret : 2 reti : 0 rjmp : 1
185 rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
186 sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
187 seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
188 sev : 0 sez : 0 sleep : 0 spm : 0 st : 0 std : 0
189 sts : 2 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0
```

190

191 Instructions used: 16 out of 114 (14.0%)

192

193 "ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x000058	72	16	88	49152	0.2%
[.dseg]	0x002800	0x002805	0	5	5	6144	0.1%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

199

200 Assembly complete, 0 errors, 0 warnings

201

Verification Strategy:

For Task 2:

1. In the watch window, add variables `led_display` and `digit_num`.
2. Test all values of `digit_num` with all bytes in `led_display` set to zero
3. Repeat for all bytes in `led_display` set to `$FF`