Judah Ben-Eliezer

112352727
3/4/2021

Prelab 4:

Quadruple Two-Input Logic Gate IC Test System

```
ic_test_v2.elf:     file format elf32-avr

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .data         00000018  00804000  00000268  000002fc  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  1 .text         00000268  00000000  00000000  00000094  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .bss          00000002  00804018  00804018  00000314  2**0
                  ALLOC
  3 .comment      00000030  00000000  00000000  00000314  2**0
                  CONTENTS, READONLY
  4 .note.gnu.avr.deviceinfo 00000040  00000000  00000000  00000344  2**2
                  CONTENTS, READONLY
  5 .debug_aranges 00000020  00000000  00000000  00000384  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_info   0000311e  00000000  00000000  000003a4  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_abbrev 00002ddb  00000000  00000000  000034c2  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_line   0000041d  00000000  00000000  0000629d  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_frame  00000024  00000000  00000000  000066bc  2**2
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    0000169c  00000000  00000000  000066e0  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_loc    00000091  00000000  00000000  00007d7c  2**0
                  CONTENTS, READONLY, DEBUGGING
 12 .debug_ranges 00000010  00000000  00000000  00007e0d  2**0
                  CONTENTS, READONLY, DEBUGGING

Disassembly of section .text:

00000000 <__vectors>:
   0:   0c 94 7a 00     jmp 0xf4    ; 0xf4 <__ctors_end>
   4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
   8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
   c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  10:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  14:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  18:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  1c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  20:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  24:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  28:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  2c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  30:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  34:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
```

```
 50    38:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 51    3c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 52    40:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 53    44:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 54    48:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 55    4c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 56    50:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 57    54:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 58    58:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 59    5c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 60    60:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 61    64:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 62    68:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 63    6c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 64    70:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 65    74:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 66    78:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 67    7c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 68    80:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 69    84:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 70    88:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 71    8c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 72    90:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 73    94:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 74    98:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 75    9c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 76    a0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 77    a4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 78    a8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 79    ac:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 80    b0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 81    b4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 82    b8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 83    bc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 84    c0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 85    c4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 86    c8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 87    cc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 88    d0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 89    d4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 90    d8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 91    dc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 92    e0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 93    e4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 94    e8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 95    ec:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 96    f0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
 97
 98 000000f4 <__ctors_end>:
```

```
 99    f4:   11 24           eor r1, r1
100    f6:   1f be           out 0x3f, r1    ; 63
101    f8:   cf ef           ldi r28, 0xFF   ; 255
102    fa:   cd bf           out 0x3d, r28   ; 61
103    fc:   df e7           ldi r29, 0x7F   ; 127
104    fe:   de bf           out 0x3e, r29   ; 62
105
106  00000100 <__do_copy_data>:
107   100:   10 e4           ldi r17, 0x40   ; 64
108   102:   a0 e0           ldi r26, 0x00   ; 0
109   104:   b0 e4           ldi r27, 0x40   ; 64
110   106:   e8 e6           ldi r30, 0x68   ; 104
111   108:   f2 e0           ldi r31, 0x02   ; 2
112   10a:   00 e0           ldi r16, 0x00   ; 0
113   10c:   0b bf           out 0x3b, r16   ; 59
114   10e:   02 c0           rjmp    .+4         ; 0x114 <__do_copy_data+0x14>
115   110:   07 90           elpm    r0, Z+
116   112:   0d 92           st  X+, r0
117   114:   a8 31           cpi r26, 0x18   ; 24
118   116:   b1 07           cpc r27, r17
119   118:   d9 f7           brne    .-10        ; 0x110 <__do_copy_data+0x10>
120
121  0000011a <__do_clear_bss>:
122   11a:   20 e4           ldi r18, 0x40   ; 64
123   11c:   a8 e1           ldi r26, 0x18   ; 24
124   11e:   b0 e4           ldi r27, 0x40   ; 64
125   120:   01 c0           rjmp    .+2         ; 0x124 <.do_clear_bss_start>
126
127  00000122 <.do_clear_bss_loop>:
128   122:   1d 92           st  X+, r1
129
130  00000124 <.do_clear_bss_start>:
131   124:   aa 31           cpi r26, 0x1A   ; 26
132   126:   b2 07           cpc r27, r18
133   128:   e1 f7           brne    .-8         ; 0x122 <.do_clear_bss_loop>
134   12a:   0e 94 9b 00     call    0x136   ; 0x136 <main>
135   12e:   0c 94 32 01     jmp 0x264   ; 0x264 <_exit>
136
137  00000132 <__bad_interrupt>:
138   132:   0c 94 00 00     jmp 0   ; 0x0 <__vectors>
139
140  00000136 <main>:
141
142  uint8_t i;
143
144  int main(void)
145  {
146      PORTA_DIRSET = PA_setup_gm;
147   136:   8f e1           ldi r24, 0x1F   ; 31
```

```
148   138:   80 93 01 04     sts 0x0401, r24 ; 0x800401 <__TEXT_REGION_LENGTH__ +0x7e0401>
149      PORTB_DIRSET = PB_setup_gm;
150   13c:   10 92 21 04     sts 0x0421, r1  ; 0x800421 <__TEXT_REGION_LENGTH__ +0x7e0421>
151      PORTC_DIRSET = PC_setup_gm;
152   140:   10 92 41 04     sts 0x0441, r1  ; 0x800441 <__TEXT_REGION_LENGTH__ +0x7e0441>
153      PORTD_DIRSET = PD_setup_gm;
154   144:   10 92 61 04     sts 0x0461, r1  ; 0x800461 <__TEXT_REGION_LENGTH__ +0x7e0461>
155      PORTE_DIRSET = PE_setup_gm;
156   148:   87 e0           ldi r24, 0x07   ; 7
157   14a:   80 93 81 04     sts 0x0481, r24 ; 0x800481 <__TEXT_REGION_LENGTH__ +0x7e0481>
158      PORTF_DIRSET = PF_setup_gm;
159   14e:   85 e0           ldi r24, 0x05   ; 5
160   150:   80 93 a1 04     sts 0x04A1, r24 ; 0x8004a1 <__TEXT_REGION_LENGTH__ +0x7e04a1>
161
162      PORTA_PIN7CTRL = PORT_PULLUPEN_bm;
163   154:   88 e0           ldi r24, 0x08   ; 8
164   156:   80 93 17 04     sts 0x0417, r24 ; 0x800417 <__TEXT_REGION_LENGTH__ +0x7e0417>
165      PORTA_PIN6CTRL = PORT_PULLUPEN_bm;
166   15a:   80 93 16 04     sts 0x0416, r24 ; 0x800416 <__TEXT_REGION_LENGTH__ +0x7e0416>
167      PORTA_PIN5CTRL = PORT_PULLUPEN_bm;
168   15e:   80 93 15 04     sts 0x0415, r24 ; 0x800415 <__TEXT_REGION_LENGTH__ +0x7e0415>
169      PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
170   162:   80 93 14 04     sts 0x0414, r24 ; 0x800414 <__TEXT_REGION_LENGTH__ +0x7e0414>
171      PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
172   166:   80 93 13 04     sts 0x0413, r24 ; 0x800413 <__TEXT_REGION_LENGTH__ +0x7e0413>
173
174      PORTD_OUT &= ~(BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm);
175   16a:   e4 e6           ldi r30, 0x64   ; 100
176   16c:   f4 e0           ldi r31, 0x04   ; 4
177   16e:   80 81           ld  r24, Z
178   170:   87 70           andi    r24, 0x07   ; 7
179   172:   80 83           st  Z, r24
180      #else
181          //round up by default
182          __ticks_dc = (uint32_t)(ceil(fabs(__tmp)));
183      #endif
184
185      __builtin_avr_delay_cycles(__ticks_dc);
```

```
174:   2f ef           ldi r18, 0xFF   ; 255
176:   84 e3           ldi r24, 0x34   ; 52
178:   9c e0           ldi r25, 0x0C   ; 12
17a:   21 50           subi    r18, 0x01   ; 1
17c:   80 40           sbci    r24, 0x00   ; 0
17e:   90 40           sbci    r25, 0x00   ; 0
180:   e1 f7           brne    .-8         ; 0x17a <main+0x44>
182:   00 c0           rjmp    .+0         ; 0x184 <main+0x4e>
184:   00 00           nop
   _delay_ms(1000);
   PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
186:   88 ef           ldi r24, 0xF8   ; 248
188:   80 83           st  Z, r24

   while (1)
   {
       while (!(PORTA_IN & START_PB_bm)) {}
18a:   80 91 08 04     lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__
  +0x7e0408>
18e:   84 ff           sbrs    r24, 4
190:   fc cf           rjmp    .-8         ; 0x18a <main+0x54>
       while (PORTA_IN & START_PB_bm) {}
192:   80 91 08 04     lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__
  +0x7e0408>
196:   84 fd           sbrc    r24, 4
198:   fc cf           rjmp    .-8         ; 0x192 <main+0x5c>

       PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
19a:   e4 e6           ldi r30, 0x64   ; 100
19c:   f4 e0           ldi r31, 0x04   ; 4
19e:   88 ef           ldi r24, 0xF8   ; 248
1a0:   80 83           st  Z, r24

       PORTD_OUT &= ~TIP_bm;
1a2:   80 81           ld  r24, Z
1a4:   8f 7e           andi    r24, 0xEF   ; 239
1a6:   80 83           st  Z, r24

       gate_type = PORTA_IN >> 5;
1a8:   a8 e0           ldi r26, 0x08   ; 8
1aa:   b4 e0           ldi r27, 0x04   ; 4
1ac:   8c 91           ld  r24, X
1ae:   82 95           swap    r24
1b0:   86 95           lsr r24
1b2:   87 70           andi    r24, 0x07   ; 7
1b4:   80 93 18 40     sts 0x4018, r24 ; 0x804018 <__data_end>

       PORTD_OUT &= ~(PORTA_IN & DIP_SW_gm);
1b8:   8c 91           ld  r24, X
```

```
233   1ba:   80 7e           andi    r24, 0xE0   ; 224
234   1bc:   90 e0           ldi r25, 0x00   ; 0
235   1be:   80 95           com r24
236   1c0:   90 95           com r25
237   1c2:   90 81           ld  r25, Z
238   1c4:   89 23           and r24, r25
239   1c6:   80 83           st  Z, r24
240
241         if (gate_type == 4) {
242   1c8:   80 91 18 40     lds r24, 0x4018 ; 0x804018 <__data_end>
243   1cc:   84 30           cpi r24, 0x04   ; 4
244   1ce:   49 f4           brne    .+18        ; 0x1e2 <main+0xac>
245             //enable pullups
246             PORTA_PIN7CTRL = PORT_PULLUPEN_bm;
247   1d0:   88 e0           ldi r24, 0x08   ; 8
248   1d2:   80 93 17 04     sts 0x0417, r24 ; 0x800417 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0417>
249             PORTA_PIN6CTRL = PORT_PULLUPEN_bm;
250   1d6:   80 93 16 04     sts 0x0416, r24 ; 0x800416 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0416>
251             PORTA_PIN5CTRL = PORT_PULLUPEN_bm;
252   1da:   80 93 15 04     sts 0x0415, r24 ; 0x800415 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0415>
253             PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
254   1de:   80 93 14 04     sts 0x0414, r24 ; 0x800414 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0414>
255         }
256
257         //turn DUT pin 14 on
258         PORTE_OUT |= PIN3_bm;
259   1e2:   e4 e8           ldi r30, 0x84   ; 132
260   1e4:   f4 e0           ldi r31, 0x04   ; 4
261   1e6:   80 81           ld  r24, Z
262   1e8:   88 60           ori r24, 0x08   ; 8
263   1ea:   80 83           st  Z, r24
264
265         for (uint8_t i = 0; i < 4; ++i) {
266   1ec:   40 e0           ldi r20, 0x00   ; 0
267   1ee:   20 c0           rjmp    .+64        ; 0x230 <__EEPROM_REGION_LENGTH__   ↵
        +0x30>
268             PORTC_OUT = stimulus[i];
269   1f0:   24 2f           mov r18, r20
270   1f2:   30 e0           ldi r19, 0x00   ; 0
271   1f4:   f9 01           movw    r30, r18
272   1f6:   ec 5e           subi    r30, 0xEC   ; 236
273   1f8:   ff 4b           sbci    r31, 0xBF   ; 191
274   1fa:   80 81           ld  r24, Z
275   1fc:   80 93 44 04     sts 0x0444, r24 ; 0x800444 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0444>
```

```
276     can be achieved.
277 */
278 void
279 _delay_loop_1(uint8_t __count)
280 {
281     __asm__ volatile (
282  200:   82 e0           ldi r24, 0x02   ; 2
283  202:   8a 95           dec r24
284  204:   f1 f7           brne    .-4         ; 0x202 <__EEPROM_REGION_LENGTH__ +0x2>
285
286             _delay_loop_1(2);
287
288             if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
289  206:   50 91 88 04     lds r21, 0x0488 ; 0x800488 <__TEXT_REGION_LENGTH__ +0x7e0488>
290  20a:   90 91 18 40     lds r25, 0x4018 ; 0x804018 <__data_end>
291  20e:   89 2f           mov r24, r25
292  210:   90 e0           ldi r25, 0x00   ; 0
293  212:   88 0f           add r24, r24
294  214:   99 1f           adc r25, r25
295  216:   88 0f           add r24, r24
296  218:   99 1f           adc r25, r25
297  21a:   80 50           subi    r24, 0x00   ; 0
298  21c:   90 4c           sbci    r25, 0xC0   ; 192
299  21e:   fc 01           movw    r30, r24
300  220:   e2 0f           add r30, r18
301  222:   f3 1f           adc r31, r19
302  224:   90 81           ld  r25, Z
303  226:   85 2f           mov r24, r21
304  228:   8f 70           andi    r24, 0x0F   ; 15
305  22a:   89 13           cpse    r24, r25
306  22c:   03 c0           rjmp    .+6         ; 0x234 <__EEPROM_REGION_LENGTH__ +0x34>
307         }
308
309         //turn DUT pin 14 on
310         PORTE_OUT |= PIN3_bm;
311
312         for (uint8_t i = 0; i < 4; ++i) {
313  22e:   4f 5f           subi    r20, 0xFF   ; 255
314  230:   44 30           cpi r20, 0x04   ; 4
315  232:   f0 f2           brcs    .-68        ; 0x1f0 <main+0xba>
316             _delay_loop_1(2);
317
318             if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
319         }
320
321         PORTD_OUT |= TIP_bm;
```

```
322   234:   e4 e6           ldi r30, 0x64   ; 100
323   236:   f4 e0           ldi r31, 0x04   ; 4
324   238:   80 81           ld  r24, Z
325   23a:   80 61           ori r24, 0x10   ; 16
326   23c:   80 83           st  Z, r24
327
328          if (i == 4) PORTD_OUT &= ~PASS_bm;
329   23e:   80 91 19 40     lds r24, 0x4019 ; 0x804019 <i>
330   242:   84 30           cpi r24, 0x04   ; 4
331   244:   21 f4           brne    .+8         ; 0x24e <__EEPROM_REGION_LENGTH__   ↵
        +0x4e>
332   246:   80 81           ld  r24, Z
333   248:   87 7f           andi    r24, 0xF7   ; 247
334   24a:   80 83           st  Z, r24
335   24c:   05 c0           rjmp    .+10        ; 0x258 <__EEPROM_REGION_LENGTH__   ↵
        +0x58>
336          else PORTD_OUT &= ~FAIL_bm;
337   24e:   e4 e6           ldi r30, 0x64   ; 100
338   250:   f4 e0           ldi r31, 0x04   ; 4
339   252:   80 81           ld  r24, Z
340   254:   8f 7b           andi    r24, 0xBF   ; 191
341   256:   80 83           st  Z, r24
342
343          //turn DUT pin 14 off
344          PORTE_OUT &= ~PIN3_bm;
345   258:   e4 e8           ldi r30, 0x84   ; 132
346   25a:   f4 e0           ldi r31, 0x04   ; 4
347   25c:   80 81           ld  r24, Z
348   25e:   87 7f           andi    r24, 0xF7   ; 247
349   260:   80 83           st  Z, r24
350      }
351   262:   93 cf           rjmp    .-218       ; 0x18a <main+0x54>
352
353 00000264 <_exit>:
354   264:   f8 94           cli
355
356 00000266 <__stop_program>:
357   266:   ff cf           rjmp    .-2         ; 0x266 <__stop_program>
358
```

```
ic_test_ident.elf:     file format elf32-avr

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .data         00000018  00804000  000002f8  0000038c  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  1 .text         000002f8  00000000  00000000  00000094  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .bss          00000002  00804018  00804018  000003a4  2**0
                  ALLOC
  3 .comment      00000030  00000000  00000000  000003a4  2**0
                  CONTENTS, READONLY
  4 .note.gnu.avr.deviceinfo 00000040  00000000  00000000  000003d4  2**2
                  CONTENTS, READONLY
  5 .debug_aranges 00000030  00000000  00000000  00000414  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_info   000031a4  00000000  00000000  00000444  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_abbrev 00002e18  00000000  00000000  000035e8  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_line   000004de  00000000  00000000  00006400  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_frame  00000044  00000000  00000000  000068e0  2**2
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    000016b0  00000000  00000000  00006924  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_loc    000000ed  00000000  00000000  00007fd4  2**0
                  CONTENTS, READONLY, DEBUGGING
 12 .debug_ranges 00000020  00000000  00000000  000080c1  2**0
                  CONTENTS, READONLY, DEBUGGING

Disassembly of section .text:

00000000 <__vectors>:
   0:   0c 94 7a 00     jmp 0xf4    ; 0xf4 <__ctors_end>
   4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
   8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
   c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  10:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  14:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  18:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  1c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  20:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  24:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  28:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  2c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  30:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
  34:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
```

```
38:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
3c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
40:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
44:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
48:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
4c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
50:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
54:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
58:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
5c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
60:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
64:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
68:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
6c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
70:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
74:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
78:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
7c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
80:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
84:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
88:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
8c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
90:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
94:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
98:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
9c:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
a0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
a4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
a8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
ac:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
b0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
b4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
b8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
bc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
c0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
c4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
c8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
cc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
d0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
d4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
d8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
dc:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
e0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
e4:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
e8:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
ec:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>
f0:   0c 94 99 00     jmp 0x132   ; 0x132 <__bad_interrupt>

000000f4 <__ctors_end>:
```

```
 99    f4:   11 24           eor r1, r1
100    f6:   1f be           out 0x3f, r1    ; 63
101    f8:   cf ef           ldi r28, 0xFF   ; 255
102    fa:   cd bf           out 0x3d, r28   ; 61
103    fc:   df e7           ldi r29, 0x7F   ; 127
104    fe:   de bf           out 0x3e, r29   ; 62
105
106  00000100 <__do_copy_data>:
107   100:   10 e4           ldi r17, 0x40   ; 64
108   102:   a0 e0           ldi r26, 0x00   ; 0
109   104:   b0 e4           ldi r27, 0x40   ; 64
110   106:   e8 ef           ldi r30, 0xF8   ; 248
111   108:   f2 e0           ldi r31, 0x02   ; 2
112   10a:   00 e0           ldi r16, 0x00   ; 0
113   10c:   0b bf           out 0x3b, r16   ; 59
114   10e:   02 c0           rjmp    .+4         ; 0x114 <__do_copy_data+0x14>
115   110:   07 90           elpm    r0, Z+
116   112:   0d 92           st  X+, r0
117   114:   a8 31           cpi r26, 0x18   ; 24
118   116:   b1 07           cpc r27, r17
119   118:   d9 f7           brne    .-10        ; 0x110 <__do_copy_data+0x10>
120
121  0000011a <__do_clear_bss>:
122   11a:   20 e4           ldi r18, 0x40   ; 64
123   11c:   a8 e1           ldi r26, 0x18   ; 24
124   11e:   b0 e4           ldi r27, 0x40   ; 64
125   120:   01 c0           rjmp    .+2         ; 0x124 <.do_clear_bss_start>
126
127  00000122 <.do_clear_bss_loop>:
128   122:   1d 92           st  X+, r1
129
130  00000124 <.do_clear_bss_start>:
131   124:   aa 31           cpi r26, 0x1A   ; 26
132   126:   b2 07           cpc r27, r18
133   128:   e1 f7           brne    .-8         ; 0x122 <.do_clear_bss_loop>
134   12a:   0e 94 0a 01     call    0x214   ; 0x214 <main>
135   12e:   0c 94 7a 01     jmp 0x2f4   ; 0x2f4 <_exit>
136
137  00000132 <__bad_interrupt>:
138   132:   0c 94 00 00     jmp 0   ; 0x0 <__vectors>
139
140  00000136 <test>:
141
142  uint8_t i;
143
144  void test() {
145      //turn DUT pin 14 on
146      PORTE_OUT |= PIN3_bm;
147   136:   e4 e8           ldi r30, 0x84   ; 132
```

```
148   138:   f4 e0           ldi r31, 0x04   ; 4
149   13a:   80 81           ld  r24, Z
150   13c:   88 60           ori r24, 0x08   ; 8
151   13e:   80 83           st  Z, r24
152
153      for (uint8_t i = 0; i < 4; ++i) {
154   140:   40 e0           ldi r20, 0x00   ; 0
155   142:   20 c0           rjmp    .+64        ; 0x184 <test+0x4e>
156          PORTC_OUT = stimulus[i];
157   144:   24 2f           mov r18, r20
158   146:   30 e0           ldi r19, 0x00   ; 0
159   148:   f9 01           movw    r30, r18
160   14a:   ec 5e           subi    r30, 0xEC   ; 236
161   14c:   ff 4b           sbci    r31, 0xBF   ; 191
162   14e:   80 81           ld  r24, Z
163   150:   80 93 44 04     sts 0x0444, r24 ; 0x800444 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0444>
164       can be achieved.
165   */
166   void
167   _delay_loop_1(uint8_t __count)
168   {
169       __asm__ volatile (
170   154:   82 e0           ldi r24, 0x02   ; 2
171   156:   8a 95           dec r24
172   158:   f1 f7           brne    .-4         ; 0x156 <test+0x20>
173
174          _delay_loop_1(2);
175
176          if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
177   15a:   50 91 88 04     lds r21, 0x0488 ; 0x800488 <__TEXT_REGION_LENGTH__      ↵
        +0x7e0488>
178   15e:   90 91 18 40     lds r25, 0x4018 ; 0x804018 <__data_end>
179   162:   89 2f           mov r24, r25
180   164:   90 e0           ldi r25, 0x00   ; 0
181   166:   88 0f           add r24, r24
182   168:   99 1f           adc r25, r25
183   16a:   88 0f           add r24, r24
184   16c:   99 1f           adc r25, r25
185   16e:   80 50           subi    r24, 0x00   ; 0
186   170:   90 4c           sbci    r25, 0xC0   ; 192
187   172:   fc 01           movw    r30, r24
188   174:   e2 0f           add r30, r18
189   176:   f3 1f           adc r31, r19
190   178:   90 81           ld  r25, Z
191   17a:   85 2f           mov r24, r21
192   17c:   8f 70           andi    r24, 0x0F   ; 15
193   17e:   89 13           cpse    r24, r25
194   180:   03 c0           rjmp    .+6         ; 0x188 <test+0x52>
```

```
195
196 void test() {
197     //turn DUT pin 14 on
198     PORTE_OUT |= PIN3_bm;
199
200     for (uint8_t i = 0; i < 4; ++i) {
201  182:   4f 5f           subi    r20, 0xFF   ; 255
202  184:   44 30           cpi r20, 0x04   ; 4
203  186:   f0 f2           brcs    .-68        ; 0x144 <test+0xe>
204         _delay_loop_1(2);
205
206         if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
207     }
208
209     if (i == 4) PORTE_OUT &= ~PASS_bm;
210  188:   80 91 19 40     lds r24, 0x4019 ; 0x804019 <i>
211  18c:   84 30           cpi r24, 0x04   ; 4
212  18e:   31 f4           brne    .+12        ; 0x19c <test+0x66>
213  190:   e4 e8           ldi r30, 0x84   ; 132
214  192:   f4 e0           ldi r31, 0x04   ; 4
215  194:   80 81           ld  r24, Z
216  196:   87 7f           andi    r24, 0xF7   ; 247
217  198:   80 83           st  Z, r24
218  19a:   08 95           ret
219     else PORTE_OUT &= ~FAIL_bm;
220  19c:   e4 e8           ldi r30, 0x84   ; 132
221  19e:   f4 e0           ldi r31, 0x04   ; 4
222  1a0:   80 81           ld  r24, Z
223  1a2:   8f 7b           andi    r24, 0xBF   ; 191
224  1a4:   80 83           st  Z, r24
225  1a6:   08 95           ret
226
227 000001a8 <identify>:
228 }
229
230 uint8_t identify() {
231 uint8_t i, j;
232
233     for (i = 0; i < 5; ++i) {
234  1a8:   80 e0           ldi r24, 0x00   ; 0
235  1aa:   30 c0           rjmp    .+96        ; 0x20c <__EEPROM_REGION_LENGTH__
      +0xc>
236         if (i == 5) {
237  1ac:   85 30           cpi r24, 0x05   ; 5
238  1ae:   49 f4           brne    .+18        ; 0x1c2 <identify+0x1a>
239             //enable pullups
240             PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
241  1b0:   98 e0           ldi r25, 0x08   ; 8
242  1b2:   90 93 14 04     sts 0x0414, r25 ; 0x800414 <__TEXT_REGION_LENGTH__
```

```
      +0x7e0414>
243             PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
244  1b6:   90 93 13 04     sts 0x0413, r25 ; 0x800413 <__TEXT_REGION_LENGTH__
      +0x7e0413>
245             PORTF_PIN5CTRL = PORT_PULLUPEN_bm;
246  1ba:   90 93 b5 04     sts 0x04B5, r25 ; 0x8004b5 <__TEXT_REGION_LENGTH__
      +0x7e04b5>
247             PORTF_PIN4CTRL = PORT_PULLUPEN_bm;
248  1be:   90 93 b4 04     sts 0x04B4, r25 ; 0x8004b4 <__TEXT_REGION_LENGTH__
      +0x7e04b4>
249      if (i == 4) PORTE_OUT &= ~PASS_bm;
250      else PORTE_OUT &= ~FAIL_bm;
251
252  }
253
254  uint8_t identify() {
255   1c2:   90 e0           ldi r25, 0x00   ; 0
256   1c4:   1e c0           rjmp    .+60        ; 0x202 <__EEPROM_REGION_LENGTH__
      +0x2>
257             PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
258             PORTF_PIN5CTRL = PORT_PULLUPEN_bm;
259             PORTF_PIN4CTRL = PORT_PULLUPEN_bm;
260         }
261         for (j = 0; j < 4; ++j) {
262             PORTC_OUT = stimulus[j];
263   1c6:   49 2f           mov r20, r25
264   1c8:   50 e0           ldi r21, 0x00   ; 0
265   1ca:   fa 01           movw    r30, r20
266   1cc:   ec 5e           subi    r30, 0xEC   ; 236
267   1ce:   ff 4b           sbci    r31, 0xBF   ; 191
268   1d0:   20 81           ld  r18, Z
269   1d2:   20 93 44 04     sts 0x0444, r18 ; 0x800444 <__TEXT_REGION_LENGTH__
      +0x7e0444>
270   1d6:   22 e0           ldi r18, 0x02   ; 2
271   1d8:   2a 95           dec r18
272   1da:   f1 f7           brne    .-4         ; 0x1d8 <identify+0x30>
273             _delay_loop_1(2);
274
275             if (!((PORTE_IN & GATES_OUT_gm) == verify[i][j])) break;
276   1dc:   60 91 88 04     lds r22, 0x0488 ; 0x800488 <__TEXT_REGION_LENGTH__
      +0x7e0488>
277   1e0:   28 2f           mov r18, r24
278   1e2:   30 e0           ldi r19, 0x00   ; 0
279   1e4:   22 0f           add r18, r18
280   1e6:   33 1f           adc r19, r19
281   1e8:   22 0f           add r18, r18
282   1ea:   33 1f           adc r19, r19
283   1ec:   20 50           subi    r18, 0x00   ; 0
284   1ee:   30 4c           sbci    r19, 0xC0   ; 192
```

```
285   1f0:   f9 01           movw    r30, r18
286   1f2:   e4 0f           add r30, r20
287   1f4:   f5 1f           adc r31, r21
288   1f6:   30 81           ld  r19, Z
289   1f8:   26 2f           mov r18, r22
290   1fa:   2f 70           andi    r18, 0x0F   ; 15
291   1fc:   23 13           cpse    r18, r19
292   1fe:   03 c0           rjmp    .+6         ; 0x206 <__EEPROM_REGION_LENGTH__ +0x6>
293               PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
294               PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
295               PORTF_PIN5CTRL = PORT_PULLUPEN_bm;
296               PORTF_PIN4CTRL = PORT_PULLUPEN_bm;
297           }
298           for (j = 0; j < 4; ++j) {
299   200:   9f 5f           subi    r25, 0xFF   ; 255
300   202:   94 30           cpi r25, 0x04   ; 4
301   204:   00 f3           brcs    .-64        ; 0x1c6 <identify+0x1e>
302               _delay_loop_1(2);
303
304               if (!((PORTE_IN & GATES_OUT_gm) == verify[i][j])) break;
305           }
306
307           if (j == 4) {
308   206:   94 30           cpi r25, 0x04   ; 4
309   208:   21 f0           breq    .+8         ; 0x212 <__EEPROM_REGION_LENGTH__ +0x12>
310   }
311
312   uint8_t identify() {
313   uint8_t i, j;
314
315       for (i = 0; i < 5; ++i) {
316    20a:   8f 5f           subi    r24, 0xFF   ; 255
317    20c:   85 30           cpi r24, 0x05   ; 5
318    20e:   70 f2           brcs    .-100       ; 0x1ac <identify+0x4>
319               return i;
320           }
321
322       }
323
324       return 0x07;
325    210:   87 e0           ldi r24, 0x07   ; 7
326
327   }
328    212:   08 95           ret
329
330   00000214 <main>:
331
```

```
int main(void)
{
    PORTA_DIRSET = PA_setup_gm;
 214:   8f e1           ldi r24, 0x1F   ; 31
 216:   80 93 01 04     sts 0x0401, r24 ; 0x800401 <__TEXT_REGION_LENGTH__ +0x7e0401>
    PORTB_DIRSET = PB_setup_gm;
 21a:   10 92 21 04     sts 0x0421, r1  ; 0x800421 <__TEXT_REGION_LENGTH__ +0x7e0421>
    PORTC_DIRSET = PC_setup_gm;
 21e:   10 92 41 04     sts 0x0441, r1  ; 0x800441 <__TEXT_REGION_LENGTH__ +0x7e0441>
    PORTD_DIRSET = PD_setup_gm;
 222:   10 92 61 04     sts 0x0461, r1  ; 0x800461 <__TEXT_REGION_LENGTH__ +0x7e0461>
    PORTE_DIRSET = PE_setup_gm;
 226:   87 e0           ldi r24, 0x07   ; 7
 228:   80 93 81 04     sts 0x0481, r24 ; 0x800481 <__TEXT_REGION_LENGTH__ +0x7e0481>
    PORTF_DIRSET = PF_setup_gm;
 22c:   85 e0           ldi r24, 0x05   ; 5
 22e:   80 93 a1 04     sts 0x04A1, r24 ; 0x8004a1 <__TEXT_REGION_LENGTH__ +0x7e04a1>

    PORTA_PIN7CTRL = PORT_PULLUPEN_bm;
 232:   88 e0           ldi r24, 0x08   ; 8
 234:   80 93 17 04     sts 0x0417, r24 ; 0x800417 <__TEXT_REGION_LENGTH__ +0x7e0417>
    PORTA_PIN6CTRL = PORT_PULLUPEN_bm;
 238:   80 93 16 04     sts 0x0416, r24 ; 0x800416 <__TEXT_REGION_LENGTH__ +0x7e0416>
    PORTA_PIN5CTRL = PORT_PULLUPEN_bm;
 23c:   80 93 15 04     sts 0x0415, r24 ; 0x800415 <__TEXT_REGION_LENGTH__ +0x7e0415>
    PORTB_PIN4CTRL = PORT_PULLUPEN_bm;
 240:   80 93 34 04     sts 0x0434, r24 ; 0x800434 <__TEXT_REGION_LENGTH__ +0x7e0434>
    PORTB_PIN3CTRL = PORT_PULLUPEN_bm;
 244:   80 93 33 04     sts 0x0433, r24 ; 0x800433 <__TEXT_REGION_LENGTH__ +0x7e0433>

    PORTD_OUT &= ~(BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm);
 248:   e4 e6           ldi r30, 0x64   ; 100
 24a:   f4 e0           ldi r31, 0x04   ; 4
 24c:   80 81           ld  r24, Z
 24e:   87 70           andi    r24, 0x07   ; 7
 250:   80 83           st  Z, r24
   #else
       //round up by default
```

```
370         __ticks_dc = (uint32_t)(ceil(fabs(__tmp)));
371     #endif
372
373     __builtin_avr_delay_cycles(__ticks_dc);
374   252:   2f ef           ldi r18, 0xFF   ; 255
375   254:   84 e3           ldi r24, 0x34   ; 52
376   256:   9c e0           ldi r25, 0x0C   ; 12
377   258:   21 50           subi    r18, 0x01   ; 1
378   25a:   80 40           sbci    r24, 0x00   ; 0
379   25c:   90 40           sbci    r25, 0x00   ; 0
380   25e:   e1 f7           brne    .-8         ; 0x258 <main+0x44>
381   260:   00 c0           rjmp    .+0         ; 0x262 <main+0x4e>
382   262:   00 00           nop
383     _delay_ms(1000);
384     PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
385   264:   88 ef           ldi r24, 0xF8   ; 248
386   266:   80 83           st  Z, r24
387
388     while (1)
389     {
390         while (!(PORTA_IN & START_PB_bm)) {}
391   268:   80 91 08 04     lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__ ↵
      +0x7e0408>
392   26c:   84 ff           sbrs    r24, 4
393   26e:   fc cf           rjmp    .-8         ; 0x268 <main+0x54>
394         while (PORTA_IN & START_PB_bm) {}
395   270:   80 91 08 04     lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__ ↵
      +0x7e0408>
396   274:   84 fd           sbrc    r24, 4
397   276:   fc cf           rjmp    .-8         ; 0x270 <main+0x5c>
398
399         PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
400   278:   e4 e6           ldi r30, 0x64   ; 100
401   27a:   f4 e0           ldi r31, 0x04   ; 4
402   27c:   88 ef           ldi r24, 0xF8   ; 248
403   27e:   80 83           st  Z, r24
404
405         PORTD_OUT &= ~TIP_bm;
406   280:   80 81           ld  r24, Z
407   282:   8f 7e           andi    r24, 0xEF   ; 239
408   284:   80 83           st  Z, r24
409
410         gate_type = PORTA_IN >> 5;
411   286:   80 91 08 04     lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__ ↵
      +0x7e0408>
412   28a:   82 95           swap    r24
413   28c:   86 95           lsr r24
414   28e:   87 70           andi    r24, 0x07   ; 7
415   290:   80 93 18 40     sts 0x4018, r24 ; 0x804018 <__data_end>
```

```
416
417         //turn DUT pin 14 on
418         PORTE_OUT |= PIN3_bm;
419   294:  e4 e8           ldi r30, 0x84   ; 132
420   296:  f4 e0           ldi r31, 0x04   ; 4
421   298:  80 81           ld  r24, Z
422   29a:  88 60           ori r24, 0x08   ; 8
423   29c:  80 83           st  Z, r24
424
425         if (gate_type == 0x04) {
426   29e:  80 91 18 40     lds r24, 0x4018 ; 0x804018 <__data_end>
427   2a2:  84 30           cpi r24, 0x04   ; 4
428   2a4:  61 f4           brne    .+24        ; 0x2be <main+0xaa>
429             //enable pullups
430             PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
431   2a6:  88 e0           ldi r24, 0x08   ; 8
432   2a8:  80 93 14 04     sts 0x0414, r24 ; 0x800414 <__TEXT_REGION_LENGTH__ ↵
      +0x7e0414>
433             PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
434   2ac:  80 93 13 04     sts 0x0413, r24 ; 0x800413 <__TEXT_REGION_LENGTH__ ↵
      +0x7e0413>
435             PORTF_PIN5CTRL = PORT_PULLUPEN_bm;
436   2b0:  80 93 b5 04     sts 0x04B5, r24 ; 0x8004b5 <__TEXT_REGION_LENGTH__ ↵
      +0x7e04b5>
437             PORTF_PIN4CTRL = PORT_PULLUPEN_bm;
438   2b4:  80 93 b4 04     sts 0x04B4, r24 ; 0x8004b4 <__TEXT_REGION_LENGTH__ ↵
      +0x7e04b4>
439             test();
440   2b8:  0e 94 9b 00     call    0x136   ; 0x136 <test>
441   2bc:  09 c0           rjmp    .+18        ; 0x2d0 <main+0xbc>
442         } else if (gate_type == 0x07) {
443   2be:  87 30           cpi r24, 0x07   ; 7
444   2c0:  29 f4           brne    .+10        ; 0x2cc <main+0xb8>
445             gate_type = identify();
446   2c2:  0e 94 d4 00     call    0x1a8   ; 0x1a8 <identify>
447   2c6:  80 93 18 40     sts 0x4018, r24 ; 0x804018 <__data_end>
448   2ca:  02 c0           rjmp    .+4         ; 0x2d0 <main+0xbc>
449         } else {
450             test();
451   2cc:  0e 94 9b 00     call    0x136   ; 0x136 <test>
452         }
453
454         PORTD_OUT |= TIP_bm;
455   2d0:  e4 e6           ldi r30, 0x64   ; 100
456   2d2:  f4 e0           ldi r31, 0x04   ; 4
457   2d4:  80 81           ld  r24, Z
458   2d6:  80 61           ori r24, 0x10   ; 16
459   2d8:  80 83           st  Z, r24
460
```

```
461         PORTD_OUT &= ~(gate_type & DIP_SW_gm);
462   2da:  90 81           ld  r25, Z
463   2dc:  80 91 18 40     lds r24, 0x4018 ; 0x804018 <__data_end>
464   2e0:  80 7e           andi    r24, 0xE0   ; 224
465   2e2:  80 95           com r24
466   2e4:  89 23           and r24, r25
467   2e6:  80 83           st  Z, r24
468
469         //turn DUT pin 14 off
470         PORTE_OUT &= ~PIN3_bm;
471   2e8:  e4 e8           ldi r30, 0x84   ; 132
472   2ea:  f4 e0           ldi r31, 0x04   ; 4
473   2ec:  80 81           ld  r24, Z
474   2ee:  87 7f           andi    r24, 0xF7   ; 247
475   2f0:  80 83           st  Z, r24
476     }
477   2f2:  ba cf           rjmp    .-140       ; 0x268 <main+0x54>
478
479 000002f4 <_exit>:
480   2f4:  f8 94           cli
481
482 000002f6 <__stop_program>:
483   2f6:  ff cf           rjmp    .-2         ; 0x2f6 <__stop_program>
484
```

Verification Strategy:

For part 1:
Same as lab 3, test each input combination for each gate of the DUT. Only difference is the power control. Failure of this will be obvious, because without power the gates will not work.

For part 2:
To verify that the program correctly identifies the IC, I will test it with each IC, making sure that it is correct each time. Because of successful verification of the first part, ic_test_v2, the IC's are known to be functional, and the only variable is whether the program knows which IC it is testing. I will verify this manually.