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

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .data         00000000  00804000  00804000  0000018c  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  1 .text         00000138  00000000  00000000  00000054  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .comment      00000030  00000000  00000000  0000018c  2**0
                  CONTENTS, READONLY
  3 .note.gnu.avr.deviceinfo 00000040  00000000  00000000  000001bc  2**2
                  CONTENTS, READONLY
  4 .debug_aranges 00000020  00000000  00000000  000001fc  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_info   00003403  00000000  00000000  0000021c  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev 00002d85  00000000  00000000  0000361f  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   00000346  00000000  00000000  000063a4  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_frame  00000024  00000000  00000000  000066ec  2**2
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_str    000019d2  00000000  00000000  00006710  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_ranges 00000010  00000000  00000000  000080e2  2**0
                  CONTENTS, READONLY, DEBUGGING

Disassembly of section .text:

00000000 <__vectors>:
   0:   0c 94 7a 00     jmp 0xf4    ; 0xf4 <__ctors_end>
   4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
   8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
   c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  10:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  14:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  18:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  1c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  20:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  24:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  28:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  2c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  30:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  34:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  38:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  3c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  40:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  44:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
```

```
  48:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  4c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  50:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  54:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  58:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  5c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  60:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  64:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  68:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  6c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  70:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  74:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  78:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  7c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  80:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  84:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  88:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  8c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  90:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  94:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  98:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  9c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  ac:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  bc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  cc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  dc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  ec:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  f0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>

000000f4 <__ctors_end>:
  f4:   11 24           eor r1, r1
  f6:   1f be           out 0x3f, r1    ; 63
  f8:   cf ef           ldi r28, 0xFF   ; 255
  fa:   cd bf           out 0x3d, r28   ; 61
```

```
 fc:   df e7           ldi r29, 0x7F   ; 127
 fe:   de bf           out 0x3e, r29   ; 62
100:   0e 94 86 00     call    0x10c   ; 0x10c <main>
104:   0c 94 9a 00     jmp 0x134   ; 0x134 <_exit>

00000108 <__bad_interrupt>:
108:   0c 94 00 00     jmp 0   ; 0x0 <__vectors>

0000010c <main>:

#define CLKCTRL_PDIV_enable 0x01

int main(void)
{
    PORTA.DIRSET = PIN7_bm;
10c:   90 e8           ldi r25, 0x80   ; 128
10e:   90 93 01 04     sts 0x0401, r25 ; 0x800401 <__TEXT_REGION_LENGTH__
  +0x7e0401>
    CPU_CCP = CCP_IOREG_gc;
112:   88 ed           ldi r24, 0xD8   ; 216
114:   84 bf           out 0x34, r24   ; 52
    CLKCTRL.MCLKCTRLA = CLKCTRL_CLKOUT_bm | CLKSEL_OSCHF_gc;
          // main clock enabled on CLKOUT, main clock set to internal high
      frequency oscillator.
116:   e0 e6           ldi r30, 0x60   ; 96
118:   f0 e0           ldi r31, 0x00   ; 0
11a:   90 83           st  Z, r25
    CPU_CCP = CCP_IOREG_gc;
11c:   84 bf           out 0x34, r24   ; 52
    CLKCTRL.MCLKCTRLB = CLKCTRL_PDIV_enable | CLKCTRL_PDIV_4X_gc;
          // prescalar division enabled, prescaler set to 2
11e:   93 e0           ldi r25, 0x03   ; 3
120:   91 83           std Z+1, r25    ; 0x01
    CPU_CCP = CCP_IOREG_gc;
122:   84 bf           out 0x34, r24   ; 52
    CLKCTRL.MCLKCTRLC = CLKCTRL_CFDSRC_CLKMAIN_gc | CLKCTRL_CFDEN_bm;
          // clock failure source set to main clock, clock failure detection
      enabled.
124:   91 e0           ldi r25, 0x01   ; 1
126:   92 83           std Z+2, r25    ; 0x02
    CPU_CCP = CCP_IOREG_gc;
128:   84 bf           out 0x34, r24   ; 52
    CLKCTRL.MCLKINTCTRL = CLKCTRL_INTTYPE_INT_gc;
          // regular interrupt type.
12a:   13 82           std Z+3, r1 ; 0x03
    CPU_CCP = CCP_IOREG_gc;
12c:   84 bf           out 0x34, r24   ; 52
    CLKCTRL.OSCHFCTRLA = CLKCTRL_RUNSTBY_bm | CLKCTRL_AUTOTUNE_bm;
          // run on standby enabled, autotune enabled.
```

```
 12e:   81 e8           ldi r24, 0x81    ; 129
 130:   80 87           std Z+8, r24     ; 0x08
 132:   ff cf           rjmp    .-2          ; 0x132 <main+0x26>

00000134 <_exit>:
 134:   f8 94           cli

00000136 <__stop_program>:
 136:   ff cf           rjmp    .-2          ; 0x136 <__stop_program>
```