```
###############################################################################
#
# IAR C/C++ Compiler V6.70.1.929 for Atmel AVR           29/Jan/2016  02:06:45
# Copyright 1996-2015 IAR Systems AB.
# Standalone license - IAR Embedded Workbench 4K Kickstart edition for Atmel AVR 6.70
#
#    Source file  =  G:\labs\0\2-PRELIMINARY\T4\sws_and.c
#    Command line =
#        G:\labs\0\2-PRELIMINARY\T4\sws_and.c --cpu=m128 -ms -o
#        G:\labs\0\2-Preliminary\T4\Debug\Obj -lCN
#        G:\labs\0\2-Preliminary\T4\Debug\List -y --initializers_in_flash
#        --no_cse --no_inline --no_code_motion --no_cross_call --no_clustering
#        --no_tbaa --debug -DENABLE_BIT_DEFINITIONS -e --eeprom_size 4096
#        --clib -On
#    List file    =  G:\labs\0\2-Preliminary\T4\Debug\List\sws_and.lst
#    Object file  =  G:\labs\0\2-Preliminary\T4\Debug\Obj\sws_and.r90
#
###############################################################################

G:\labs\0\2-PRELIMINARY\T4\sws_and.c
          /*
          *      title:         sws_and.c
          *      description:    performs AND bitwise between PD7-PD4 and PD3-PD0. Output
          *                      to PB3-PB0 as active low. PB7-PB4 are kept constant high.
          *      target:        ATMEGA128
          */

          #include <iom128.h>

   \                     In  segment ABSOLUTE, at 0x30
   \   union <unnamed> volatile __io _A_PIND
   \                     _A_PIND:
   \   00000000                      DS8 1

   \                     In  segment ABSOLUTE, at 0x31
   \   union <unnamed> volatile __io _A_DDRD
   \                     _A_DDRD:
   \   00000000                      DS8 1

   \                     In  segment ABSOLUTE, at 0x32
   \   union <unnamed> volatile __io _A_PORTD
   \                     _A_PORTD:
   \   00000000                      DS8 1

   \                     In  segment ABSOLUTE, at 0x37
   \   union <unnamed> volatile __io _A_DDRB
   \                     _A_DDRB:
   \   00000000                      DS8 1

   \                     In  segment ABSOLUTE, at 0x38
   \   union <unnamed> volatile __io _A_PORTB
   \                     _A_PORTB:
   \   00000000                      DS8 1
```

```
  9

\                          In  segment CODE, align 2, keep-with-next
 10        int main(void){
\                 main:
 11          //setup input and output pins
 12            DDRD =     0x00;
\  00000000   E000             LDI     R16, 0
\  00000002   BB01             OUT     0x11, R16
 13            PORTD =    0xFF;   //switches will use internal PUN
\  00000004   EF0F             LDI     R16, 255
\  00000006   BB02             OUT     0x12, R16
 14            DDRB =     0xFF;
\  00000008   EF0F             LDI     R16, 255
\  0000000A   BB07             OUT     0x17, R16
 15
 16          //define variables
 17            char tempA,tempB;
 18
 19          //while loop for AND process
 20            while(1){
 21              tempA = PIND;         //tempA = 0bABCDEFGH
\                 ??main_0:
\  0000000C   B300             IN      R16, 0x10
\  0000000E   2F10             MOV     R17, R16
 22              tempB = tempA>>4;      //tempB = 0b0000ABCD
\  00000010   2F01             MOV     R16, R17
\  00000012   9502             SWAP    R16
\  00000014   700F             ANDI    R16, 0x0F
\  00000016   2F20             MOV     R18, R16
 23
 24              //PORTB = ~( PIND & (PIND>>4) )   //Volatile solution
 25
 26              PORTB = ~( tempA & tempB );      //AND values and invert for active low
\  00000018   2F01             MOV     R16, R17
\  0000001A   2302             AND     R16, R18
\  0000001C   9500             COM     R16
\  0000001E   BB08             OUT     0x18, R16
\  00000020   CFF5             RJMP    ??main_0
\  00000022                 REQUIRE _A_DDRD
\  00000022                 REQUIRE _A_PORTD
\  00000022                 REQUIRE _A_DDRB
\  00000022                 REQUIRE _A_PIND
\  00000022                 REQUIRE _A_PORTB
 27            }
 28        }

Maximum stack usage in bytes:

RSTACK Function
------ --------
   2   main
   Segment part sizes:

   Bytes  Function/Label
   -----  --------------
       1  _A_DDRB
       1  _A_DDRD
       1  _A_PIND
       1  _A_PORTB
       1  _A_PORTD
      34  main


  5 bytes in segment ABSOLUTE
 34 bytes in segment CODE

 34 bytes of CODE memory
  0 bytes of DATA memory (+ 5 bytes shared)

Errors: none
Warnings: none
```