```
/*
 * ReadAccrossWordBoundary.s
 *
 *
 * Created by amaral on 2012-10-26.
 * Copyright 2012 __MyCompanyName__. All rights reserved.
 */

Pseudo-code for a function that reads a bit-field between two bit positions k1 and k2 in a bitvector of arbitrary size.
The function reads a bitfield of length smaller than 33

Parameters:
$a0: a, the address of the word containing the first position of vector A
$a1: k1, the first bit to be read
$a2: k2, the last bit to be read

Return value:
    if k2+1-k1 < 33
      then
        $v1 = 0
        $v0 = bitfield between k1 and k2
    else
        $v1 = 1

Conventions:

Bits within a word are numbered from the least significant
to the most significant as follows:

31        27        23        19              7 6 5 4   3 2 1 0
 b b b b   b b b b   b b b b   b b b b  ....   b b b b   b b b b


Example:

ReadField(0x80008000, 27, 36)


Assume that the following two words are found at addresses 0x80008000 and 0x80008004, respectively:
```

```
at 0x80008000:
31        27        23        19              7 6 5 4   3 2 1 0
 x x x x   x b b b   b b b b   b b b b  ....   b b b b   b b b b

at 0x80008004:
31        27        23        19              7 6 5 4   3 2 1 0
 b b b b   b b b b   b b b b   b b b b  ....   b b b y   y y y y

Then, ReadField return values are:
$v0 = 0000 0000 0000 0000 0000 00yy yyyx xxxx
$v1 = 0

ReadField:
    if(k2+1-k1 > 32)
      $v1 <-- 1
      return
    WordAddress_k1 <-- $a0 + (($a1 >> 5)<< 2)  # Computes address of word that contains k1
    FirstWord <-- Mem[WordAddress_k1]
    BitPosition_k1 <-- $a1 AND 0x0001F
    BitPosition_k2 <-- $a2 AND 0x0001F
    if(BitPosition_k2 >= BitPosition_k1)                # The entire bit field is in a single word
      SingleWordMask <-- 0x8000 >>_a (31-BitPosition_k1)  # arithmetic shift to right to make all leading bits 1 up to
      position of k1
                                                        # example: if k1 = 27, this produces 0xF800 0000
      t2 <-- 31-BitPosition_k2
      SingleWordMask <-- (SingleWordMask << t2) >> t2     # arithmetic shifts to zero leading bits
                                                        # example: if k2 = 29, in the example above, it produces 0x3800 0000
    else                                                # Bitfield spans over a word boundary
      # This case is left as an exercise
```