

# Multiplication

Xavier Perséguers

9th October 2002

```
1  .start 0
2  main:
3      # Init memory:
4      addiu    $t0, $zero, 0x8A13; # { mem[100] = 1
5      sw       $t0, 0x0100($zero); # }
6      addiu    $t0, $zero, 0xFFFF; # { mem[100] = 2
7      sw       $t0, 0x0104($zero); # }
9      # Init parameters:
10     addi     $a0, $zero, 0x100; # a0: Array memory address
11     addi     $a1, $zero, 2;      # a1: Number of elements
13     jal      proc;
14     nop; nop;
16  infini: j      infini;
17     nop; nop;
19  proc:   add     $v0, $zero, $zero; # v0 = 0
20     add     $t0, $zero, $zero; # t0 = 0
21  outer: sltu    $t2, $t0, $a1;      # t2 = (t0 < a1)
22     beq     $t2, $zero, fin;      # if (!t2) goto fin
23     nop; nop;
24     lw      $t3, 0($a0);          # t3 = mem[a0]
25     addi    $t4, $zero, 32;       # t4 = 32
26  inner: beq     $t4, $zero, next;  # if (!t4) goto next
27     nop; nop;
28     andi    $t1, $t3, 1;          # t1 = t3 & 1
29     add     $v0, $v0, $t1;        # v0 = v0 + t1
30     srl     $t3, $t3, 1;          # t3 = t3 >> 1
31     addi    $t4, $t4, -1;         # t4 = t4 - 1
32     j      inner;
33  next:  addi    $t0, $t0, 1;        # t0 = t0 + 1
34     addi    $a0, $a0, 4;          # a0 = a0 + 4
35     j      outer;                # goto outer
36     nop; nop;
37  fin:   jr     $ra;                # return to caller
38     nop; nop;
```