```
0.              addi $t1, $zero, 3          // $t1 = 3      // addi ok
1.              subi $t2, $zero, -2         // $t2 = 2      // subi ok
2.              add $t0, $t1, $t2           // $t0 = 5      // add ok
3.              sub $t3, $t1, $t2           // $t3 = 1      // sub ok
4.              nor $t4, $t0, $t2           // $t4 = -8     // nor ok, all regs ok
5.              sw $t1, 3($t2)              // m[5] = 3     // sw ok
6.              srl $t2, $t2, 1             // $t2 = 1      // srl ok
7.              beq $t2, $t3, label1        // branching will execute // beq ok
8.              j end
9. label1:      sll $t3, $t3, 1             // $t3 = 2      // sll ok
10.             lw $t2, 4($t2)              // $t2 = 3      // lw ok
12.             push $t1                    // m[FF] = 3 // push ok
13.             push $t2                    // m[FE] = 3 // push ok
14.             j label2                    // jump will execute // j ok
15. label3:     or $t0, $t0, $t2            // $t0 = 7      // or ok
16.             andi $t2, $t4, 1            // $t2 = 0      // and ok
17.             ori $t1, $t1, 5             // $t1 = 7      // ori ok
18.             pop $t2                     // $t2 = 3      // pop ok
19.             and $t1, $t2, $t4           // $t1 = 0      // andi ok
20.             pop $t1                     // $t1 = 3      // pop ok
21.             j end                       // jump will execute // j ok
22. label2:     bneq $t0, $t2, label3       // branching will execute // bneq ok
23. end:
```