

3. (1). `addi x0, x0, 0` (2). `jalu x0, x, 0`

(3). `auipc x1, offset [31:12]` `jalu x1, x1, offset`

(4). `addi rdrs, 0` (5). `csrrs rd, cycle [h], x0`

(6). `addiw rdrs, 0`

7. (1). `sltu t3, t2, 0` `slt t4, t0, 1`

(2). `addu t0, t1, t2` `bltu t0, t1, overflow`

8. (1). `-1 x -1 x` 不会

(2). NV代表 invalid operation. DZ: divide by zero

OF: overflow, UF: underflow, NX代表 inexact

tf flags 被置位不会使处理器陷入系统调用

12. (1). 管理员模式 (2). 机器模式 (3). 机器模式

(4). 用户模式 (5). 用户模式.

13. vecMul:

`addi sp, sp, -32`

`sw ra, 24(sp)`

`sw s0, 24(sp)`

`addi s0, sp, 32`

`sw a0, -12(s0)`

`sw a1, -16(s0)`

`sw a2, -20(s0)`

`li a0, 0`

`sw a0, -24(s0)`

`if`

`lw a1, -24(s0)`

`li a0, 99`

`blt a0, a1, end`

`lw a0, -16(s0)`

`lw a1, -24(s0)`

`slli a2, a1, 2`

`add a0, a0, a2`

`lw a0, 0(a0)`

`lw a1, -20(s0)`

`mul a0, a0, a1`

`lw a1, -12(s0)`

lw a0, -24(sp)

addi a0, a0, 1

sw a0, -24(sp)

j if

end:

lw a0, -12(sp)

lw a0, 0(a0)

lw ra, 28(sp)

lw s0, 24(sp)

addi sp, sp, 32

ret

14. start: beg a1, a0. if.

add a2, a0, a1

l: sub a2, a0, a1

15. start = sw t0, 0(t0)

addi t1, x0, 3

sw t1, 4(t0)

slli t2, t1, 2

add t2, t2, t1

sw t1, 0(t2)

17. 计算  $a_1 = 2^{20}$