

3. (1) nop addi x0, x0, 0

(2) ret jalr x0, x10, 0

(3) call offset auipc x6, offset [31:12]
 jalr x1, x6, offset [11:0]

(4) mv rd, rs addi rd, rs, 0

(5) rdcycle rd csrrs rd, cycle, ~~x0~~ x0

(6) sextw rd, rs addiw rd, rs, 0

7. (1) ~~ret~~ slt t3, t1, x0
 slt t4, t0, t2

(2) add t0, t1, t2
 bgt t1, t0, overflow

8. ~~0xf~~⁽¹⁾ 0xff ..., X(rst1), 0xff ..., X(rst1)
 不会引起异常,

(2) NV: 无效操作异常	当 fflags 被置位时, 不会使处理器陷入系统调用。相反, 处理器将继续执行下一条指令,
DZ: 除以0异常	但在此之后, 如果使用了一条需要使用浮点结果的指令, 那么处理器将检查 fflags, 并在必要时生成浮点异常。
OF: 溢出异常	
UF: 下溢异常	
NX: 不精确异常	

(3) 在 x86 架构中, 当浮点除法的除数为 0 时, 会产生浮点异常并触发相应的浮点异常处理程序, 该处理程序可以选择继续执行或者终止程序的执行。

12. 1. 机器模式 2. 机器模式 3. 监管模式 4. 用户模式 5. 用户模式

13. add a2, x0, x0

addi a3, x0, 100

Loop:

bge a2, a3, exit

sll a4, a2, 2

add a5, a4, t0

add a4, a4, t1

lw a4, 0(a4)

~~add a4, a4, a5~~

tmul a4, a4, t2

sw a4, 0(a5)

addi a2, a2, 1

j Loop

exit:

~~add a0, x0~~

~~14~~ lw a0, 0(t0)

15. ~~stw~~

sw t0, 0(t0)

addi t1, t1, 3

add t2, t0, t1

sw t1, 4(t0)

sw t1, 0(t2)

16. ~~sw~~

lw t2, 0(t0)

lw t3, 0(t1)

sw t3, 0(t0)

sw t2, 0(t1)

17. a 存放于 a0, b 存放于 a1, c 存放于 a2

求出 $b = b * 2^{c-a}$ ($c > a$)

14. ~~bge a0, a1, a0~~

~~bge a0, a1, if~~

bge a1, a0, else

add a2, a1, a0

else:

~~addsub~~ a2, a0, a1