

Chapter 2

答: 1) nop: addi x0,x0,0

2) ret: jalr x0,x1,0

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[h],x0

6) sext.w rd,rs: addiw rd,rs,0

答: 1) slti t₃,t₂,0

slt t₀,t₀,t₁

2) bltu x5,x6,overflow

3) x86 通过 OF 标志位检测溢出

ARM 通过 CPSR 的状态寄存器反映当前指令的溢出状态。

答: 1) $2^{XLEN} - 1$ ($XLEN$ 为除数和被除数位数); x ; -1 ; x

整型除法中除数为0不会引起 RIS-V 抛出异常。如果触发异常，在绝大多数执行环境里会导致一个自陷。此时需要语言实现者与执行函数的自陷处理函数交互。如果不触发，只要在每个除法操作时增加一些分支指令即可，且该分支指令通常不跳转，对性能影响极小。

2) NV: 非法操作 ; DZ: 除以零 ; OF: 上溢 ; UF: 下溢 ; NX: 不精确

fflags 被置位不会使处理器陷入系统调用。

3) x86 通过标志位判断中断并执行相应的中断服务程序，然后返回执行原程序。

12. 答: 1) M 2) M 3) M 4) S 5) U

13. 解: vecMul: add x10, x0, x0

addi x11, x0, 100

lw x12, 0(t2)

loop: bge x10, x11, exit

sll x13, x10, 2

add x14, x13, t0

add x15, x13, t0

lw x15, 0(x15)

mul x13, x15, x12

sw x13, 0(x14)

addi x10, x10, 1

j loop

exit: lw a0, 0(t0)

ret

14. 解: part1: bge a1, a0, part2

sub a2, a0, a1

part2: add a2, a0, a1

sw t0, 0(t0)

li t1, 3

sw t1, 4(t0)

sll 05, t1, 2

add a5, a5, t0

sw t1, 0(a5)

16答: swap:
lw a4, 0(t0)
lw a5, 0(t1)
sw a5, 0(t0)
sw a4, 0(t1)
ret

17答: 功能为计算 2^{30}