

3. 1) nop    addi x0, x0, 0
- 2) ret    jalr x0, x1, 0
- 3) call offset    amipc x6, offset [36:12]  
                    jalr x1, x6, offset [11:0]
- 4) mv rd, rs    addi rd, rs, 0
- 5) rdelaycycle rd    csrrs rd, cycle, x0
- 6) sext.w rd, rs    addi.w rd, rs, 0

7. 1) slti t3, t2, 0  
      slt t4, t0, t1
- 2) addl t0, t1, t2  
      bltu t0, t1, overflow

3) x86: 如果两个正数相加结果为负, 两个负数相加结果为正, 则溢出

ARM: 通过 CPSR 的状态寄存器反映溢出

8. 1) 指令                      rd 值

rs1	rs2	Op = DIVU 时	Op = RZMU 时	Op = DIV 时	Op = RZM 时
Op rd, rs1, rs2	x    0	0x ffffffff... 16Tf	rs1	0x ffffffff... 16Tf	rs1

整型除法中除数为 0 会引起异常。

0xf...f 代表 -1, 会报错, 而 RZM 是取余, rd = rs1 意味着没变化。

2) NV - 无效操作数异常: NV = 0, 没有产生操作数异常; NV = 1, 产生无效操作数异常

DZ - 除 0 异常: DZ = 0, 没有产生除 0 异常; DZ = 1, 产生除 0 异常

OF - 上溢异常: OF = 0, 没有...; OF = 1, 产生上溢异常

UF - 下溢异常: UF = 0, 没有...; UF = 1, 产生下溢异常

NX - 非精确异常: NX = 0, 没有...; NX = 1, 产生非精确异常

3) x86 除数为 0 时会显示出现异常, ARM 同样会导致用法异常

12. 1) Linux Kernel S

2) BootROM M

3) BootLoader M

4) USB Driver U

5) vim U

13. add x11, x0, x0 # i=0

addi x12, x0, 100 # x12=100

Loop:

bge x11, x12, exit

sll x13, x11, 2

add x13, x13, t0

lw x13, 0(x13)

sll x14, x11, 2

add x14, x14, t1

lw x14, 0(x14)

mul x13, x14, t2

addi x11, x11, 1

j Loop

exit:

sw a0, t0

lw a0, 0(a0)

ret.

14. blt a1, a0, if

sub a2, a0, a1

l:

add a2, a0, a1

15. lw x11, t0

add t0, x11, x0

lw x12, t1

addi x12, x0, 3

addi x13, t0, 4

sw x13, 0(x13)

add x14, t0, x12

sw x14, x12

16. lw x11, t0

add x12, x11, x0

lw x13, t1

add x11, x13, x0

sw x11, t0

add x13, x12, x0

sw x13, t1

ret

17. 计算  $2^{30}$  的结果.