

3. 解：1) `nop : addi zero,zero,0`

- 2) `ret : jalr x0,X1,0`
- 3) `call offset : jal label`
- 4) `mv rd,rs : addi rd,rs,0`
- 5) `rdcycle rd : rtime to
 csrw cycle,to
 addi X1,zero,0`
- 6) `sext.w rd,rs : slli rd,rs,0
 srli rd,rd,0`

7. 解：1) `sub t3,to,t1
 mv t4,t2`

- 2) `add to,t1,t2
 bltu to,t1,overflow`
- 3) X86架构：

X86架构中，用于检测加法溢出的指令是jno/jnoe指令，分别代表“跳转溢出”和“跳转不溢出”。
当寄存器溢出时，CPU将根据OF标志位的值设置jno指令的条件码，该指令将跳转到指定的目标地址。
对于无符号数，检测加法是否溢出的指令为jc/jnc，CPU会根据CF标志位的值设置jc指令的条件码，该指令将跳转到指定的目标地址。

ARM架构：
在ARM架构中，用于检测加法溢出的指令是四种“条件码移位指令”(CMP、CMN、TST和TEQ)，它们中的任意一条都可以用来比较两个操作数。

当使用CMP或CMN指令进行有符号数的比较时，V标志位被设置为1表示运算结果溢出，否则它将被清除。

当使用TST或TEQ指令进行无符号数比较时，C标志位被设置为1表示运算结果溢出，否则它将被清除。

8. 解：1)

指令	rs1	rs2	Op=DIVU 时 rd 值	Op=REMU 时 rd 值	Op=DIV 时 rd 值	Op=REM 时 rd 值
Op rd,rs1,rs2	x	0	2 ^{XLEN} -1	x	-1	x

除数为0会引起RISC-V异常，这样设计能保证系统的可靠性和安全性，防止因除数为0引起的崩溃或错误破坏系统的稳定运行。

- 2) NV：操作数错误，表示某个操作数不存在或不能代表一个实际数值

DZ：数据坏性，表示浮点运算结果非法，因为结果的指数太小而不能表示。

OF：上溢，表示某个操作的结果太大，超出了浮点数的范围。

UF：下溢，表示某个操作的结果太小，超出了浮点数的范围。

NX：不精确，表示浮点运算产生的结果是近似的，即数据上可能存在一些误差。

fflags被置位不会导致处理器陷入系统调用

- 3) X86架构：

X86指令集的除法指令在执行时会判断除数是否为0，如果除数为0，则会触发一个称为Divide Error的异常。

Divide Error异常用于表示在执行除法时除数为0的情况，处理器将生成异常类型为#DE的异常。

当Divide Error异常发生时，程序可以通过异常处理程序来恢复错误状态。

ARM架构：
无论除法中除数为0都会发生Undefined Instruction异常，该异常的异常代码为0x00000006，表示未定义操作。

对于浮点除法，当除数为0时会引发浮点异常，如下所示：

浮点除以0：“正无穷大(+∞)”或“负无穷大(-∞)”是一个无穷数，超出了浮点数表示范围。

在ARM架构中，发生异常时系统可以通过异常处理程序来恢复错误状态。

12. 解：1) M 等级

2) M 等级

3) S 等级

4) U 等级

5) U 等级

13. 解：
addi t0, A, 0
addi t1, B, 0
addi t2, C, 0
addi t3, zero, 0
lw t4, 0(t1)
mult t4, t2
mflo t5
sw t5, 0(t0)
Loop:
addi t3, t3, 1
bge t3, 100, Exit
addi t1, t1, 4
lw t4, 0(t1)
mult t4, t2
mflo t5
addi t0, t0, 4
sw t5, 0(t0)
j Loop
Exit:
lw t6, 0(t0)
jr ra

14. 解：
addi t0, zero, a
addi t1, zero, b
addi t2, zero, c
slt t3, t1, t0
beqz t3, Else
add t4, t0, t1
addi t5, t4, 0
sw t5, 0(t2)
j End
Else:
sub t4, t0, t1
addi t5, t4, 0
sw t5, 0(t2)
j End
End:

15. 解：
addi t₂, t₀, 0
sw t₂, 0(t₀)
addi t₃, t₁, 0
addi t₀, t₀, 4
sw t₃, 0(t₀)
slli t₄, t₃, 2
add t₅, t₀, t₄
sw t₃, 0(t₅)

16. 解：
lw t₂, 0(t₀)
lw t₃, 0(t₁)
sw t₂, 0(t₀)
sw t₂, 0(t₁)
jr ra

17. 解：计算 $2^{26} - 3^0$