3.21 二/3/(1) nop等价于addi x0,x0,0

(2) ret等价为 jalr x0,0(ra)

(3) call offset等价为 auipc t0, offset[31:12]和 jalr t1,t0,offset[11:0]

(4) mv rd,rs等价为 addi rd,rs,0

(5) rdcycle rd等价为 csrrs rd,cycle,x0

16) sext.w rd rs等价为 slli rd,rs,0和 srli rd,rd,0x1f

7/(1) slti t3,t2,0

slt t4,t0,t1

(2) bltu t0,t1,overflow

(3) x86架构用jo指令，该指令检查EFLAGS寄存器中的OF位，若为1，则溢出

ARM架构可以用SBC (subtract with carry)指令，它将两个操作数相减，并将 Carry Flag和 Overflow Flag设为相应的值，若CarryFlag为1，则为无符号数溢出；若Overflow Flag为1，则为有符号数溢出

8/(1)

指令	rs1	rs2	Op=DIVU	Op=REMU	Op=DIV	Op=REM
Op rd,rs1,rs2	x	0	$2^{length(x)}-1$	x	$-2^{length(x)-1}$	x

(2) NV: Invalid Operation，无效操作　DZ: Divide by Zero：除0异常

OF: Overflow 结果溢出　UF (Underflow)：结果下溢　NX: Inexact 结果不精确

fflags被置为不会使处理器陷入系统调用

(3) x86中 整型除法为0会引起除0异常，可用异常处理程序来处理。浮点除法中除0会将FPU状态寄存器中标志位置位，不会引起异常。

ARM中，整型除法除数为0会引发除0异常，处理器会将异常的类型和地址信息保存在相关寄存器中并跳转到异常处理程序，处理程序中可以进行异常处理，如输出错误信息等。

12 (1) S (2) M (3) M (4) S (5) U

13
```
add a3, zero, zero
addi a4, zero, 100
lw, a5, 0(t2)
Loop:
  bge a3, a4, exit
  sll a6, a3, 2
  add a7, a0, a6
  add a6, a6, t1
  lw a6, 0(a6)
  mul a6, a6, a5
  sw a6, 0(a7)
  addi, a3, a3, 1
  j Loop
exit:
mv a0, t0
```

14
```
  bge a0, a1, part1
  beq a0, a1, part1
  add a2, a1, a0
part1:
  sub a2, a0, a1
```

15
```
sw ra, 0(t0)
addi t1, zero, 3
sw t1, 4(t0)
sw t1, 12(t0)
```

17 功能是计算 2^{30}。

16
```
lw a1, 0(t0)
lw a2, 0(t1)
sw a1, 0(t1)
sw a2, 0(t0)
```