3. 1) nop <=> addi x0, x0, 0

2) ret <=> jalr x0, 0(x1)

3) call offset <=> auipc x1, offset[31:12] + offset[11];
jalr x1, offset[11:0](x1)

4) mv rd, rs <=> addi rd, rs, 0

5) rdcycle rd <=> csrrs rd, cycle x0

6) sext.w rd, rs <=> addiw rd, rs, 0

7. 1) ~~sext~~ sub t3, t0, t1
mv t4, t2

2) add t0, t1, t2
blt t0, t1, overflow

3) x86使用"OF"标志位来检测加法溢出，当有符号整数相加时，若溢出则"OF标志位为1；ARM使用"V"标志位检测加法溢出。

8. 1) Op = DIVU时，rd为 0xffffffffffffffff
Op = REMU时，rd为 x
Op = DIV时，rd为 0xffffffffffffffff
Op = REM时，rd为 x

2) NV: Invalid Operation
DZ: Divide by Zero
OF: Overflow
UF: Underflow
NX: Inexact

(3) x86中除数为0一般会触发除以零异常并引发CPU中断
ARM中除数为0也会触发异常，但会根据设置的异常处理方式进行相应处理.

12. 1) ~~##~~ N/A
2) M
3) S
4) M
5) U

13. part 1: add a5, x0 < mv a6, a0
mul ~~a4~~ a4 (a2)+, a3
mv (a0), a4
addi a0, a0, 4
addi a5, a5, 1
bge i, 100, end
j part 1
end: mv a0, (a6)

14. bge a1, a0, part2
~~part1:~~ sub a2, a0, a1
~~part~~ j end
part2: add a2, a0, a1
end: nop

15.
```
mv (t0), t0
addi t1, x0, 3
addi t2, t0, 4
mv (t2), t1
mul t3, t1, 4
add t2, t0, t3
mv (t2), t1
```

16.
```
mv t2, (t0)
mv (t0), (t1)
mv (t1), t2
```

17. 令 a0=0, a1=1, a2=30

将 a1 每次左移1位，循环30次。

得到 2^{30}.