

3.

nop = addi zero, zero, 0

ret = jalr X0, X1, 0

call offset = auipc Xb, offset[31,12]

jalr X1, Xb, offset[11,0]

mv rd, rs = addi rd, rs, 0

rdcycle rd = csrrs rd, cycle[h], X0

sext.w rd, rs = addiw rd, rs, 0

7.

T: 1) add t₀, t₁, t₂

slti t₃, t₂, 0

slt t₄, t₀, t₁

bne t₃, t₄, overflow

2) add t₀, t₁, t₂

bltu t₀, t₁, overflow

bltu t₀, t₂, overflow

(3) 在X86架构下, FLAGS寄存器的第11位为OF标志, 当OF标志为1时, 表示发生了加法溢出。

在ARM架构下, 同CPSR或SPSR寄存器的28位V标志代表是否溢出。

8.

	Op=DIV时	Op=REMOV时	Op=DIV时	Op=REMOV时
Op rd, rs1, rs2	X	0	rd _{2^LEN-1}	X
			-1	X

不会引起地址异常，因为这样可以减少硬件成本和延时，提高计算效率。

(2) 异常编码：NV：非法操作 DZ：除以0
OF：上溢 UF：下溢

对 FFLAGS 寄存器进行置位不会使处理器陷入系统调用，因为这只是一个用户态的操作。但是，会影响异常处理程序的正常运行。

(3) X86：会导致 CPU 停止执行当前指令并跳转到异常处理程序。

ARM：处理器会自动发异常处理程序，处理器提供了一系列程序。

12.

(1) Linux kernel S 管理员模式

(2) Boot ROM M 机器模式

(3) BootLoader S 管理员模式

(4) USB Driver S 管理员模式

(5) vim U 用户模式

13.

13.

start:

```
addi sp, sp, -3b  
sd ra, 28(sp)  
sd s0, 20(sp)  
addi s0, sp, 3b  
mv t0, a0  
mv t1, a1  
mv t2, a2  
sw t0, -20(s0)  
sw t1, -24(s0)  
sw t2, -28(s0)  
addi a3, zero, 0 # a3 是 i
```

part1:

```
addi a4, a3, -100  
bge a4, zero, end  
be  
sebi t4, a3, 2  
lw t2, -28(s0)  
lw t5, t4(t2)  
mul t5, t9, t2  
sw t5, t4(t0)  
addi a3, a3, 1  
j part1
```

end:

```
lw a1, -20(s0)  
mv a0, 0(a1)  
ld ra, 28(sp)  
ld s0, 20(sp)  
addi sp, sp, 3b  
ret
```

14.

start:

```
addi sp, sp, -3b  
sd ra, 28(sp)  
sd s0, 20(sp)  
addi s0, sp, 3b  
mv a5, a0  
mv a4, a1  
sw a5, -20(s0)  
sw a4, -24(s0)
```

part1:

```
lw a5, -20(s0)  
lw a4, -24(s0)  
bge a4, a5, part2  
add a2, a5, a4  
j end
```

part2:

```
sub a2, a5, a4
```

j end

end:

```
mv a0, a2  
ld ra, 24(sp)  
ld s0, 20(sp)  
addi sp, sp, 3b  
ret
```

15. sw t0, 0(t0)

li t1, 3

sw t1, 4(t0)

sw t1, 12(t0)

17. addi a0, x0, 0

addi a1, x0, 1

addi a2, x0, 30

loop: beq a0, a2, done

slli a1, a1, 1

addi a0, a0, 1

j loop

done: #exit code

分析子以 C 代码:

Int a = 0, b = 1, c = 30;

for (a = 0; b < 30; b = ~~a++~~) {
 b = ~~b * 2~~
 a++; }

分析是取 $b = 2^{30}$

lw a5, -20(\$0)

lw a6, -24(\$0)

add a3, a5, zero

add a5, a4, zero

add a4, a3, zero

sw a5, -20(\$0)

sw a4, -24(\$0)

lw s0, 1b(\$p)

addi sp, sp, 32

jr ra.