

C2

3.

1) nop → addi x0, x0, 0

2) ret → jalr x0, x1, 0

3) call offset → auipc x6, offset[31:12]
jalr x1, x6, offset[11:0]

4) mv rd, rs → addi rd, rs, 0

5) rdcycle rd → csr rs rd, cycle, x0

6) sext.w rd, rs → addiw rd, rs, 0

7.

1) add to, t1, t2
slti t3, t2, 0
slt t4, to, t1
bne t3, t4, overflow

2) add to, t1, t2
bltu to, t1, overflow

3) X86: 当对有符号整数的add或sub指令溢出时, 将设置溢出标志; 而当对无符号整数的操作溢出时, 将设置进位标志

ARM: 通过CPSR的状态寄存器反映当前指令的溢出状态

8.

1) DIVU rd, x, 0
~~rd~~ rd值为 0xffffffffffffffff

REMU rd, x, 0
rd值为 x

DIV rd, x, 0
rd值为 0xffffffffffffffff

REM rd, x, 0
rd值为 x

2) fflags 浮点异常标志域

若浮点运算单元在运算中出现相应异常, 则将对标志位置为1. 且为一直保持累积, 软件可以通过写0的方式单独清除某个异常标志位

NV Invalid Operation

DZ Divide by Zero

OF Overflow

UF Underflow

NX Inexact

处理器不会陷入系统调用

3) X86: 系统会出现除法零中断

ARM: 系统可配置控制寄存器CCR的DIV-0-TRP

此位为0时, 除以0操作不触发异常

此位为1时, 除以0触发异常事件并产生相应中断

12.

- | | | |
|-----------------|---|--------|
| 1) Linux Kernel | 1 | S-Mode |
| 2) BootROM | 3 | M-Mode |
| 3) Boot Loader | 3 | M-Mode |
| 4) USB Driver | 1 | S-Mode |
| 5) Vim | 0 | U-Mode |

14.

```

part 1:  blt    a1, a0, part 2
          sub    a2, a0, a1
          j      end
part 2:  add    a2, a0, a1
end:     mv     a0, a2
  
```

13.

```

part 1:  lw     a2, 0(t2)
          mv     a3, t0
          mv     a4, t1
          addi   t1, t1, 400
part 2:  beq    a4, t1, part 3
          lw     a5, 0(a3)
          lw     a6, 0(a4)
          mul    a5, a5, a6
          sw     a5, 0(a3)
          addi   a3, a3, 4
          addi   a4, a4, 4
          j      part 2
part 3:  lw     a0, 0(t0)
  
```

15.

```

part 1:  sw     t0, 0(t0)
          li     t1, 3
          sw     t1, 4(t0)
          li     a2, 4
          mul    a2, a2, t1
          add    a2, a2, t0
          sw     t1, 0(a2)
  
```

16.

```

part 1:  lw     a2, 0(t0)
          lw     a3, 0(t1)
          sw     a2, 0(t1)
          sw     a3, 0(t0)
          ret
  
```


17.

得到2的指数次幂(得到独热码)

本例: 将 2^{30} 的值存入 Q_1 寄存器

Q_1 中有一个 $2^{(Q_2 \text{ 的值})}$ 的值(独热码)
