Week 5.

3. 1) nop: addi x_0, x_0, 0

2) ret: jr ra.

3) call offset:
auipc t_1, offset [31:12]
addi t_1, t_1, offset [11:0]
jalr ra, t_1, 0

4) mv rd, rs: addi rd, rs, 0

5) rdcycle rd: csrrs rd, cycle, x_0

6) sext.w rd, rs: addiw rd, rs, 0

7. 1) srai t_3, t_0, 对31
srai t_4, t_1, 对31

2) add t_0, t_1, t_2
blt t_0, t_1, overflow

3) 在x86架构中，加法指令会设置进位标记CF和溢出标记OF用于观测是否溢出.
在ARM架构中，有专门的带进位加法ADC和带进位减SBC指令，内设CF/OF，用法同x86.

8. 1)

DIVU	REMU	DIV	REM
2^n-1	2^n-1	异常	异常.

异常现象

因为无符号数的除/求余得不到2^n-1，故可借此表达
但有符号数中，诸如-2^n除以-1时大于2^n-1，不如用
它表示异常，只能直接显示异常.

2) ①NV：执行无效浮点操作时，它会被置位 ② DZ：浮点数除以0时将被置位.

③OF：浮点运算且除数为0时将被置位 ④UF：浮点运算的结果精度超出上限时将被置位

⑤NX：浮点运算的结果无法精确表示为浮点数时被置位；fflag被置位时，处理器不会陷入系统调用，但程序员可用浮点异常处理机制处理异常.

3) x86, ARM 等指令集架构在除数为0时会抛出一个异常，称为"除数为0异常"/"浮点除0异常"并跳转到相应的异常处理.

12: 1) Machine mode 2) Machine mode 3) Supervisor mode 4) User mode 5) User mode.

13:
```
    li   a3, 0
    li   a4, 100
loop: bge  a3, a4, end
    mul  0(t0), 0(t1), t2
    beqz a3, place
    addi t0, t0, 4
    addi t1, t1, 4
    j    loop
    addi a3, a3, 1
    j    loop
place: mv  a5, 0(t0)
    addi t0, t0, 4
    addi t1, t1, 4
    addi a3, a3, 1
    j    loop
end: mv  a0, a5
    ret
```

14:
```
    bge  a0, a1, part2
    sub  a2, a0, a1
part2: add  a0, a0, a1
```

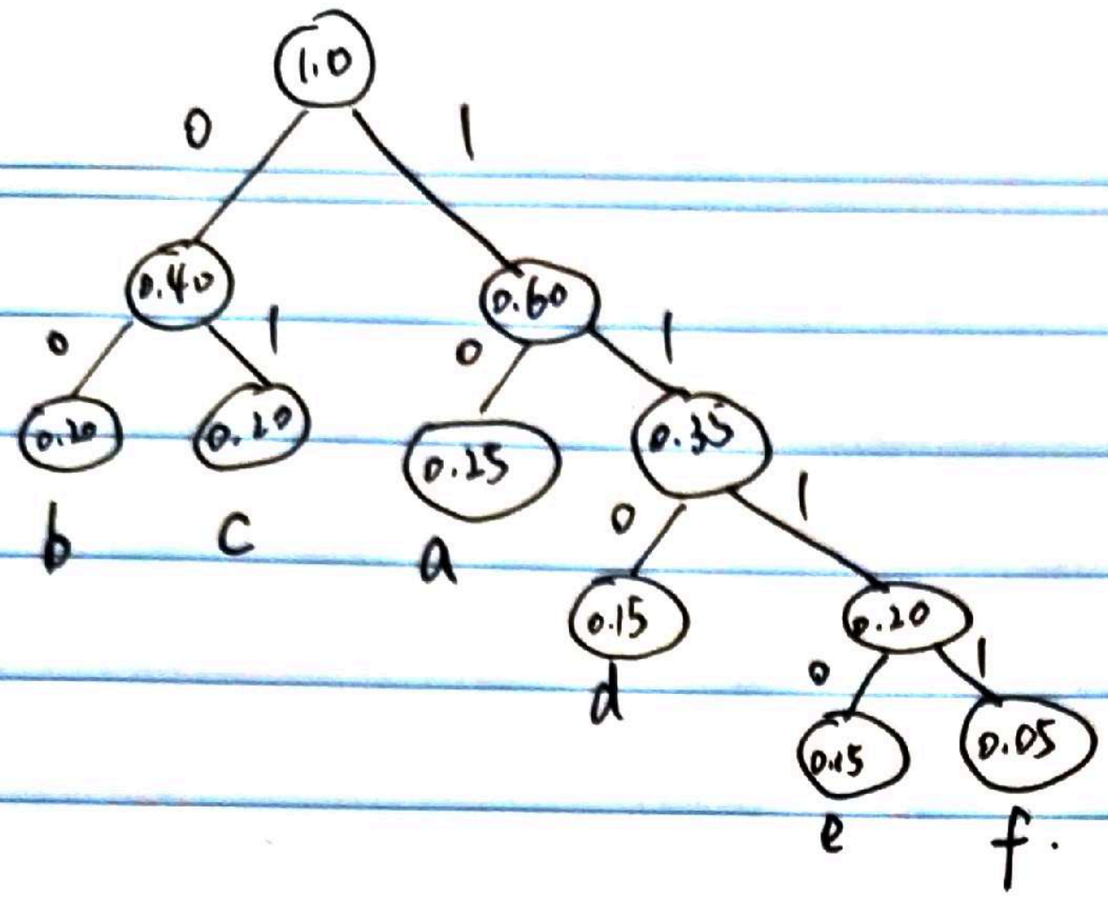
15:
```
    lw   t0, 0(t0)
    addi t1, zero, 3
    mv   1(t0), t1
    mv   t1(t0), t1
```

```
lb: addi sp, sp, -32
    sw   ra, 0(sp)
    sw   t0, 4(sp)
    sw   t1, 8(sp)
    lw   t2, 0(t0)
    lw   t0, 0(t1)
    lw   t1, 0(t2)
    lw   ra, 0(sp)
    addi sp, sp, 32
    ret.
```

```
lb: lw   t2, (t0)
    lw   t3, (t1)
    mv   t4, t2
    mv   t2, t3
    mv   t3, t4
    ret.
```

门：将寄存器 a1 中存的1 翻倍30次得到 2^{30}.

18.



a → 10 2×0.25

b → 00 2×0.2

c → 01 2×0.2

d → 110 3×0.15

e → 1110 4×0.15

f → 1111 4×0.05

$\bar{l} = \sum_{i=1}^{6} p_i l_i =$ [illegible] 2.55

$R = 1 - \frac{2.55}{3} = 0.15$.

$R = 1 - \frac{-\sum_{i=1}^{6} p_i \log_2 p_i}{\lceil \log_2 6 \rceil}$