3.

1) addi x0, x0, 0 (nop)

2) pop eip (ret)

3) jmp offset (call offset)

4) mov rd, rs (mv rd, rs)

5) rdtsc (rdcycle rd)

6) movsx rd, word ptr[rs] (sext.w rd, rs)

7. 1) xor t3, t1, t2

and t3, t3, t0

2) add t0, t1, t2

blt t0, t1, overflow

3) 用 adds 执行加法操作，根据标志位来判断加法是否溢出

12.

1) Linux Kernel
最高级别的机器模式

2) BootROM
机器模式

3) BootLoader
机器模式，通常为特权模式

4) USB Driver
特权模式

5) Vim
用户模式

13.

```
vecMul:
    addi  sp, sp, -16
    sw    s0, 0(sp)
    sw    ra, 4(sp)
    li    s0, 0
loop:
    bge   s0, 100, done
    slli  t3, s0, 2
    lw    t4, 0(t0+t3)
    mul   t5, t4, a2
    sw    t5, 0(t1+t3)
    addi  s0, s0, 1
    j     loop
done:
    lw    a0, 0(t0)
    lw    a0, 0(sp)
    lw    ra, 4(sp)
    addi  sp, sp, 16
    jr    ra
```

14

```
slt   t0, a0, a1
bne   t0, zero, else
add   a2, a0, a1
j     end
else:
  sub   a2, a0, a1
```

15.

```
li    a0, 16
li    a7, 9
ecall
mv    t1, t0
sw    t0, 0(t0)
li    t2, 3
sw    t2, 4(t0)
sw    t2, 12(t0)
slli  t3, t2, 2
add   t4, t0, t3
sw    t2, 0(t4)
```

16.

```
lw      t0, 0(a)
lw      t1, 0(b)
lw      t2, 0(t0)
sw      t1, 0(t0)
sw      t2, 0(t1)
ret
```

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:
```