第2章

4. 3.1) nop 空指令 addi x0, x0, 0

2) ret 用于从子程序返回到调用者，等价于将程序计数器PC设置为链接寄存器ra.
并使用该指令地址跳转，即 jalr x0, ra, 0

3) call offset 用于调用一个函数或子过程

auipc ra, %pcrel_hi(offset)

jalr ra, %pcrel_lo(offset)(ra)

4) mv rd, rs add rd, x0, rs

5) rdcycle rd csrr rd, cycle

6) sext.w rd, rs 符号扩展，32位有符号整数 → 64位有符号整数

slli rd, rs, 32

srai rd, rs, 31.

7. 检查加法溢出

1) add t0, t1, t2 # t0, t1, t2 都是有符号数，

slt t3, t1, zero #若 t1<0 则将 t3 设为1

slt t4, t0, t2 #若 t0<t2，则 t4 设为1

bne t3, t4, overflow #若 t3, t4 的值不相等，则跳转到 overflow

2) 若 add t0, t1, t2 # 无符号数，

sltu t3, t0, t1

bne t3, zero, overflow

3) x86架构：Overflow标志位是EFLAGS寄存器中的一个标志位，用于指示最近的一次算术操作是否溢出。在执行add指令时，若超过32位整数所能显示的范围，则设置为1

ARM：CPSR中的V标识符用于记录上一条算术操作是否溢出

8. 1) 2的xLEN次方 - 1 (即每一位都是1) x -1 (即每一位都是1) x

2) fflags是浮点异常标志寄存器.

NV: Invalid Operation (无效操作) 标志位, 表示执行了一个无效的浮点操作, 如0/0或sqrt(-1)

DF: Overflow (溢出) 标志位, 表示执行了一个浮点操作结果超出了浮点数的表示范围

UF: Underflow (下溢) 标志位, 表示执行了一个浮点操作结果小于浮点数的最小表示范围

NX: Inexact (不精确) 标志位, 表示结果不精确.

不会

3) x86: 处理器将操作权交给操作系统。异常会被映射到一个特定的异常向量.

ARM: 同上.

第

14. 1. 1) 最高特权等级(M模式)

2) 最低特权等级(M模式)

3) 较低特权等级(M或S模式)

4) 较低特权等级(M或S模式)

5) 用户态(U模式)

13. # int i=0, a2 储存常数100.

```
mv   a0, t0
li   a1, 0        # int i=0
li   a2, 100      #确定循环比较标志
loop:
bge  a1, a2, end.
lw   a3, t2
lw   a4, 0(t1)
mul  a4, a4, a3
sw   a4, 0(t0)

addi t0, t0, 4
addi t1, t1, 4
addi a1, a1, 1
j loop
end:
lw a0, 0(a0)
ret.
```

14. bgt a0, a1, part2

part1:

sub. a2, a0, a1

jx end

part2:

add a2, a0, a1

end.

15. sw t0, 0(t0)

li t1, 3

s

sw t1, 0(t2)

slli t2, t1, 2

add t2, t0, t2

sw t1, 0(t2).

16. lw t2, 0(t0) # int tmp=*a

lw t3, 0(t1) # *b

sw t3, 0(t0) # *a=*b

sw t2, 0(t1). # *b=tmp.

17. 将寄存器a1的值不断左移一位.