

第五周.

3. 1) `addi x0, x0, 0`

2) `jair x0, x1, 0`

3) `auipc x6, offset[31:12]`

`jair x1, x6, offset[11:0]`

4) `addi rd, rs, 0`

5) `rdcycle to`

`addi rd, to, 0`

6) `slli rd, rs, 32`

`srai rd, rd, 32`

7. 1) `sub t3, t0, t1`

`mv t4, t2`

2) `add t0, t1, t2`

`slt t3, t0, t1`

`bne t3, x0, overflow.`

3) 在x86架构中, 一般采用 `OF` 标志位, 是在执行加減指令后由处理器设置的一种标志位, 如果 `OF` 被设置为1, 则标志溢出.

在ARM架构中, 一般采用 `C` 和 `V` 两个标志位来检测加法是否溢出. `C` 标志指示无符号加法是否发生进位, `V` 标志位用于指示有符号加法是否发生溢出.



8. 1)  $Op = DIVU$  时  $rd = 0xFFFFFFFFFFFFFFFF$

$Op = REMU$  时  $rd = 0$

$Op = DIV$  时  $rd = 0xFFFFFFFFFFFFFFFF$

$Op = REM$  时  $rd = 0$

2) NV: 发生了某种不支持的操作

DZ: 发生了某种动态非数值情况.

OF: 计算结果溢出了

UF: 计算结果下溢

NX: 执行了无效操作

当 fflags 被置位时, 不会直接导致处理器陷入系统调用

而是可以通过指令跳转成处理器异常。在 RISC-V 中, 浮

点异常处理需要通过相应的浮点异常指令来进行显式处理。

3) 在 X86 架构中, 整数除法使用 `idiv` 指令, 被除数有符号

将相应浮点异常标志位置 1, ARM 也置 1, 但由

`UDIV` 和 `SDIV` 指令执行。

12. 1) 管理员模式

2) 机器模式

3) 管理员模式

4) 管理员模式

5) 用户模式



13. `addi a2, x0, 1`  
`addi a3, x0, 100`

`loop: bge a2, a3, exit`

`slli a4, a2, 2`

`add a5, a4, t0`

`add a6, a4, t1`

`lw a5, 0(a5)`

`lw a6, 0(a6)`

`mul a5, a6, t2`

`sw a5, 0(a5)`

`addi a2, a2, 1`

`j loop`

`exit: lw t0, 0(t0)`

`ret`

14. `bgt a0, a1, big`

`j small`

`big: add a2, a0, a1`

`small: sub a2, a0, a1`



15. sw t0, 0(t0).

addi t1, x0, 3.

sw t1, 4(t0).

sw t1, 12(t0).

16. mv t2, t0

mv t0, t1.

mv t1, t2.

17. 将寄存器 a 的值赋为  $2^{30}$