

- 3.
- 1)  $\text{nop} \Leftrightarrow \text{addi } x0, x0, 0$  空指令
  - 2)  $\text{ret} \Leftrightarrow \text{jalr } x0, x1, 0$
  - 3)  $\text{call offset} \Leftrightarrow \text{auipc } x6, \text{offset}[31:12]$   
 $\text{jalr } x1, x6, \text{offset}[11:0]$
  - 4)  $\text{mv rd, rs} \Leftrightarrow \text{addi. rd, rs, 0}$
  - 5)  $\text{rdcycle rd} \Leftrightarrow \text{csrrs rd, cycle, } x0$
  - 6)  $\text{sext.w rd, rs} \Leftrightarrow \text{addiw rd, rs, 0}$

7. 1)  $\text{sub } t3, t0, t1$

$\text{mv } t4, t2$

2)  $\text{add. } t0, t1, t2$

$\text{slt } t3, t0, t1$

$\text{bne } t3, x0, \text{overflow.}$

3) x86 中采用 OF 标志位，如果执行加法指令后 OF 为 1，则溢出。

ARM 中采用 C 和 V 标志位检测。C 标志指示无符号加法是否溢出，V 标志有符号加法是否溢出。

8 (1).  $OP = DIV \quad rd = 0xFFFFFFFFFFFFFF$

$OP = REM \quad rd = 0$

$OP = DIV. \quad rd = 0xFFFFFFFFFFFFFF$

$OP = REM. \quad rd = 0$

为了避免程序错误执行除以 0 操作，通过检测引发异常来终止程序。

(2). NV → 发生不支持操作.

DZ → 发生某种动态非数值情况.

OF → 计算结果溢出

当  $f1/f0$  被置位，指令跳转后

UF → 计算结果下溢.

使处理器进入系统调用状态.

NX → 执行无效操作

(3). x86 中，整数除法使用 idiv 指令，被除数又有符号将相应添 0 第  
十六位置 1，ARM 中也置 1，由 VDIV 和 SDIV 指令执行.

12 w. 管理员 Mode

(2). 机器模式

(3) 管理员 Mode

(4). 管理员模式

(5). 用户模式

13. addi. a<sub>2</sub>, x<sub>0</sub>, 1

addi, a<sub>1</sub>, x<sub>0</sub>, 100

loop: bge a<sub>2</sub>, a<sub>3</sub>, exit.

sub a<sub>4</sub>, a<sub>2</sub>, 2.

add a<sub>5</sub>, a<sub>4</sub>, t<sub>0</sub>.

add a<sub>6</sub>, a<sub>4</sub>, t<sub>1</sub>

lw a<sub>5</sub>, 0(a<sub>5</sub>)

lw a<sub>6</sub>, 0(a<sub>6</sub>)

mul a<sub>5</sub>, a<sub>6</sub>, t<sub>2</sub>,

sw a<sub>5</sub>, 0(a<sub>5</sub>)

addi a<sub>2</sub>, a<sub>2</sub>, 1

j loop.

exit: lw t<sub>0</sub>, 0(t<sub>0</sub>)

ret

14. bgt a<sub>0</sub>, a<sub>1</sub>, big.

j small

big: add a<sub>2</sub>, a<sub>0</sub>, a<sub>1</sub>

small: sub a<sub>2</sub>, a<sub>0</sub>, a<sub>1</sub>

15. sw t<sub>0</sub>, 0(t<sub>0</sub>)

addi t<sub>1</sub>, x<sub>0</sub>, 3

sw t<sub>1</sub>, 4(t<sub>0</sub>)

sw t<sub>1</sub>, 12(t<sub>0</sub>)

16. mv t<sub>2</sub>, t<sub>0</sub>

mv t<sub>0</sub>, t<sub>1</sub>

mv t<sub>0</sub>, t<sub>2</sub>

17. 每次 loop 将 a<sub>1</sub> 左移 1 位，右移 3 位。且 a<sub>1</sub> = 2<sup>30</sup>.