

2023.3.21

- 3) addi x0,x0,0 2) jalr x0,x0
3) auipc x1,offset[31:17] jalr x1,x1,offset
5) srs rs,rdr,rd,cycle[h].x0 6) addiw rd,r5,0
7.1) sli t3,t2 sli t4,t0,1

2) addu t0,t1,t2 bltu t0,t1,overflow

3) 在x86架构中，当执行算术指令时条件代码将设置在CPU状态字中。有些指令执行后将取消条件代码。

在ARM架构中，通过CPSR的状态寄存器反映先前指令的溢出状态。

8.1) $\frac{x}{-1}$ x -1 x

不会抛出异常，结果会是Infinity。

>IN代表invalid operation, >Z代表divide by zero, OF代表overflow, UF代表underflow, NX代表inexact。

不会转入系统调用。

3) 在x86架构中会产生0号中断，会产生CoreDump，由应用程序自己处理，或运行时压制或运行时包装再抛出。

在ARM架构中会进入HardFault中断。

1) 管理员模式

2) 机器模式

3) 机器模式

4) 用户模式

5) 用户模式

B.vecMul:

addi \$p,\$p,-32

sw ra,>4(\$p)

sw so,>4(\$p)

addi so,\$p,32

sw a0,-12(\$o)

sw a1,-16(\$o)

sw a2,>0(\$o)

li a0,0

sw a0,>4(\$o)

loop:

lw a1,>4(\$o)

li a0,99

blt a0,a1,end

lw a0,-16(\$o)

lw a1,>4(\$o)

shl a2,a1,2

add a0,a0,a2

lw a0,0(\$o)

lw a1,>0(\$o)

mul a0,a0,a1

lw a1,-12(\$o)

add a1,a1,a2

sw a0,0(\$o)

lw \$0,-24(\$0)

addi \$0,\$0,1

sw \$0,-24(\$0)

j loop

end:

lw \$0,-12(\$0)

lw \$0,0(\$0)

lw \$0,28(\$0)

lw \$0,24(\$0)

addi \$0,\$0,32

ret

16. snap: addi \$P,\$P,-32

sw ra,>8(\$P)

sw \$0,>4(\$P)

addi \$0,\$P,32

sw a0,-12(\$0)

sw a1,-16(\$0)

lw a0,-12(\$0)

lw a0,a0(\$0)

sw a0,>0(\$0)

lw a0,-16(\$0)

lw \$0,0(\$0)

lw \$1,-12(\$0)

sw \$0,0(\$1)

lw \$0,-20(\$0)

lw \$1,-16(\$0)

sw \$0,0(\$1)

lw \$a,28(\$p)

lw \$0,24(\$p)

addi \$p,\$p,3

ret

15 start: sw t0, octo)

addi t1,x0,3

sw t1,4(to)

slli t2,t1,2

add t2,t2,t1

sw t1,octo)

17. $a_1 = \geq 30$

H. start:beg Q1,a0,f
add a2,a0,a1

I. sub a2,a0,a1