

## 第五周作业

### Chapter 2.3. RISC-V 中的伪指令

- (1) nop: `addi x0, x0, 0`
- (2) ret: `jalr x0, x1, 0`
- (3) call offset: `auipc x6, offset[31:12] jalr x1, x6, offset[11:0]`
- (4) mv rd, rs: `addi rd, rs, 0`
- (5) rdcycle rd: `csrrs rd, cycle, x0`
- (6) sext.w rd, rs: `addiw rd, rs, 0`

7. 1) RISC-V 判断有符号数加法是否溢出:

```
add t0, t1, t2
slti t3, t2, 0
slt t4, t0, t1
bne t3, t4, overflow.
```

2) 无符号数加法:

```
add t0, t1, t2
bltu t0, t1, overflow.
```

- 3) ① ARM 指令集架构通过 CPSR 状态寄存器 V 位表示溢出标志  
② MIPS 指令集架构通过指令触发中断的方式产生溢出信号, 通知处理器.  
③ x86 指令集设置溢出标志 (OF) 和进/借位标志 (CF) 表示有符号数和无符号数加减法的溢出.

8. RISC-V 对除数为 0 的除法/取余指令处理:



## (1) 整型除法:

指令	rs1	rs2	DIVU rd	REMU rd	DIV rd	REM rd
Op rd, rs1, rs2	x	0	$2^{XLEN}-1$	x	-1	x

① 不会引起 RISC-V 抛出异常。

② 设计原因: 其他算术操作均不引发异常, 若整数除零引发异常, 需实现执行环境的异常处理机制; 值设为全 1 是有/无符号除法电路的自然结果, 简化硬件设计。

## (2) 浮点除法:

NV: Invalid Operation 无效操作

DZ: Divide by Zero 除以 0

OF: Overflow 上溢

UF: Underflow 下溢

NX: Inexact 不精确

不使得处理器陷入系统调用, 而要求上层软件使用时显示检查这些标志, 以保持 ISA 简洁。

(3) x86, ARM 对除数为 0 产生异常, 中断处理后继续执行。

## 12. 特权等级:

(1) Linux Kernel: Machine

(2) BootROM: Machine

(3) BootLoader: Machine

(4) USB Driver: Supervisor

(5) vim: User



13.

```
15      addi t3,x0,0
16      addi t4,x0,100
17
18      lw t6,0(t1)
19      mul t5,t6,t2
20      sw t5,0(t0)
21      mv a0,t5
22      addi t0,t0,4
23      addi t1,t1,4
24      addi t3,t3,1
25
26  loop:
27      bge t3,t4,out
28      lw t6,0(t1)
29      mul t5,t6,t2
30      sw t5,0(t0)
31      addi t0,t0,4
32      addi t1,t1,4
33      addi t3,t3,1
34      j loop
35
36  out:
37      nop
```

14-16.

```
1  #problem 14
2  blt b,a,l2
3  sub a2,a0,a1
4  j l3
5  l2:
6  add a2,a1,a0
7  l3:
8  nop
9
10 #problem 15
11 sw t0,0(t0)
12 addi t1,x0,3
13 sw t1,4(t0)
14 sw t1,12(t0)
15
16 #problem 16
17 lw t2,0(t0)
18 lw t3,0(t1)
19 sw t3,0(t0)
20 sw t2,0(t1)
```

17. 实现功能：计算  $2^{30}$  的值，存放在 a1 中。