

2023-3-21 第五周 Chapter 2: 3.7.8.12.13.14.15.16.17

3. (1) nop : addi x0, x0, 0

(2) ret : jalr x0, x1, 0

(3) call offset: auipc x6, offset[31:12] [jalr的立即数由offset[12:1]决定]

jalr x1, x6, offset[11:0]. 跳转至 4KB-4GB 的函数。

(4) mv rd, rs : addi rd, rs, 0

(5) rdcycle rd: csrrs rd, cycle, x0. [control status register: read and set. 周期数的读取]

(6) sext,w rd, rs : addiw rd, rs, 0. 符号位扩展

7. (1) add t0, t1, t2

slti t3, t2, 0. ($t_2 < 0$)

slt t4, t0, t1. ($t_4 = t_0 < t_1$)

bne t3, t4, overflow

(2) addu t0, t1, t2

bltu t0, t1, overflow

(3) x86: 用进位标志 CF 检查无符号加法的溢出：若 $t = a+b$, $CF = (t \leq a)$

用溢出标志 OF 检查有符号加法的溢出： $OF = (a < 0 == b < 0) \& \& (t < 0 != a < 0)$

arm: 用 C 标志检查无符号溢出.. $C = (t \text{ 超出操作数范围})$

用 V 标志检查有符号溢出 $V = (t \text{ 超出有符号数范围})$



扫描全能王 创建

8.11) 指令 ① $O_p = DIVU$ ② $O_p = REMU$ ③ $O_p = DIV$ ④ REM

$O_p rd, rs_1, rs_2 \quad rd = 2^{x_{LEN}-1}$ $rd = x$ $rd = -1$ $rd = x$

$(rs_1 = x)$

$(rs_2 = 0)$

整数除以0不会抛出异常。取消异常可以减少与陷阱处理程序的交互开销，也可以让除法器硬件设计更简化。

(2) $+flags$: 各位为1. 说明异常发生。

NX: 非精确异常位

DE: 除0异常位

UF: 下溢异常位

NV: 无效操作数异常位

OF: 上溢异常位

在捕获异常时，不会陷入系统调用，而是在软件中显示检查标志。

(3) Arm架构中先检查除数是否为0. 若为0. 会触发“Divide by Zero”异常。这是一个软件异常，不会陷入系统调用。

x86架构中. 除数为0. 会触发“#DE”，可以是内核的一部分. 也可以是用户程序处理. 会把异常信息保存在异常描述符表中. 以便异常处理程序进行处理。

12 1> Linux Kernel: M 47 USB driver: S

2> BootROM: M 5> vim : D

3> BootLoader: M

17. 求 $a_1 = 2^3$ 是多少



扫描全能王 创建

mk13.s

```
.text
.globl vecMul
.type vecMul,@function
```

vecMul:

```
    addi sp,sp,-32
    sd ra,24(sp)
    sd s0,16(sp)
    addi s0,sp,32
```

```
addi a5,zero,0 # i=0
addi a4,zero,99 # imax=99
```

```
mv a0,t0 # A
```

```
mv a1,t1 # B
```

loop:

```
mul (a0),(a1),t2
```

```
addi a0,a0,4
```

```
addi a1,a1,4
```

```
addi a5,a5,1
```

```
bne a5,a4,loop
```

```
j end
```

end:

```
mv a0,(t0)
```

I

```
ld ra,24(sp)
```

```
ld s0,16(sp)
```

```
addi sp,sp,32
```

```
ret
```



扫描全能王 创建

draftc

C exe3_2A1.c

fas

hmk13s

hwk14s

X

hwk14s

hwk14.s

```
1 bgt a0,a1,L1 # a>b
2 L1:
3 add a2,a1,a0 # c=a+b
4 ble a0,a1,L2 # a<=b
5 L2:
6 add a2,a0,a1 # c=a-b
7
```



扫描全能王 创建

C exe3_2A1.c

ASM fac.s

ASM hmk13.s

ASM hwk14.s

ASM hwk15.s

X

ASM hwk

ASM hwk15.s

```
1 mv ($t0),t0
2 addi t1,zero,3
3 addi t0,t0,4
4 mv ($t0),t1
5 addi t0,t0,8
6 mv ($t0),t1
```

I



扫描全能王 创建

运行(R) ...

hwk16.s - exercise - Visual Studio Code

□ □ □ 0: -

C exe3_2A1.c

ASM fac.s

ASM hmk13.s

ASM hwk14.s

ASM hwk15.s

ASM hwk16.s

X



ASM hwk16.s

```
1 .text
2 .globl swap
3 .type swap,@function
4 swap:
5     addi sp,sp,-32
6     sd ra,24(sp)
7     sd s0,16(sp)
8     addi s0,sp,32
9
10    mv t2,(t0)
11    mv (t0),(t1)
12    mv (t1),t2
13    j end
14
15 end:
16    ld ra,24(sp)
17    ld s0,16(sp)
18    addi sp,sp,32
19    ret
```



扫描全能王 创建