

第二章 3, 7, 8, 12, 13, 14, 15, 16, 17

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

2. 1) sub t3, t0, t1
addi t4, t2, 0

2) ~~~~~

bgen ~~块~~ t1, t0, overflow
bgen ~~块~~ t3, t0, overflow

3) x86

CF 和 OF
无符号溢出 为1 有符号溢出 为1

对应 ARM C, V

} 处理中的状态标志



Q. 1) 不会

	DIVU	REMU	DIV	REM
Division by zero	X	0	$2^L - 1$	X
	X	-1		X

为了保护CPU和保证程序的正常运行

- 2) NV: 无效操作 b2: 被0除
OF: 上溢 UF: 下溢
NX: 不精确数值
不会陷入系统调用

3) 引发异常中断, 停止当前程序的执行并将控制权交给OS

12. User Mode / Supervisor Mode / Machine Mode
应用 操作系统 管理安全执行环境

- 1) 操作系统内核 S
2) M
3) ~~OS~~ M
4) S
5) U

13.



my function:

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

loop:

```
li a2, 1
li a3, 100
mul (t0), (t1), (t2)
addi a1, t0, 0
addi t0, t0, 4
addi t0, t0, 4
beq a2, a3, end
addi a2, a2, 1
j loop
```

end:

```
addi a0, (a1), 0
```

```
ld ra, 24(sp)
ld s0, 16(sp)
addi sp, sp, 32
ret
```



$a < b$

14. ^(c)
blt a0, a1, Part1
add a2, a1, a0
j end

Part1:

sub a2, a0, a1

end:

nop

15. addi t0, t0, 0 ;

addi t1, x0, 3 ;

addi t0, t0, 16

addi (t0), t0, 0 ;

~~addi t2, t1, 16~~

li t2, 16

mul t3, t1, t2

addi t0, t0, t3

addi (t0), t1, 0

16. addi t2, t0, 0

addi t0, t1, 0

addi t1, t2, 0



17, $a_0=0$ $a_1=1$ $a_2=3$

将 a_1 赋予 1, 并将其值左移 30 位, 即 2^{30}

