

第二章 3, 7, 8, 12, 13, 14, 15, 16, 17

3. 1) ~~nop~~ → addi x0, x0, 0  
2) ~~ret~~ → ret jalr x0, 0(x1)  
3) call offset → auipc x1, offset[31:12] + offset[11:0],  
jalr x1, offset[11:0](x1)

4) mv rd, rs → addi rd, rs, 0

5) ~~rdcycle~~ rd → addi rd, ~~rdcycle~~, 0

6) sext.w rd, rs → addi w rd, rs, 0

7. 1) sub t3, t0, t1,

addi t4, t2, 0

2) ~~~~~

bgeu ~~t3~~ t1, t0, overflow

bgeu ~~t3~~ t3, t0, overflow

3) x86

C 和 OF  
无符号溢出 为1      有符号溢出 为1

} 处理中的状态标志

对应 ARM C, V



扫描全能王 创建

Q. 1) 不会

DIVU REMU DIV REM

Division by zero X, 0,  $2^l - 1$  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: 不准角数值

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把控制权交给OS

12. User Mode / Supervisor Mode / Machine Mode

应用 操作系统 管理安全执行环境

1) 系统内核 S

2) M

3) BM

4) S

5) U

13.



扫描全能王 创建

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  
bgeq a2, a3, end  
addi a2, a2, 1  
j loop

end:

addi d0, (a1), 0

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



扫描全能王 创建

$a < b$

14.  $\leftarrow$  blt  $a_0, a_1, \text{Part1}$   
 addl  $a_2, a_1, a_0$   
 j end

Part1:

~~add~~ sub  $a_2, a_0, a_1$   
~~sub~~

end :

nop

15. addi  $t_0, t_0, 0$  ;  
 addi  $t_1, x_0, 3$  ;  
 addi  $t_0, t_0, 16$   
 addi  $(t_0), t_0, 0$  ;  
~~addi  $t_0, t_1, t_2, 16$~~   
~~mul~~

li  $t_2, 16$

mul  $t_3, t_1, t_2$

addi  $t_0, t_0, t_3$

addi  $(t_0), t_1, 0$

16. addi  $t_2, t_0, 0$   
 addi  $t_0, t_1, 0$   
 addi  $t_1, t_2, 0$



扫描全能王 创建

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  $2^{30}$



扫描全能王 创建