

3/7/8/12/13/14/15/16/17

1) addi $X_0, X_0, 0$

2) ret 等价于 jalr $X_0, 0(r_a)$

(3) auipc ra, offset[31:12] (取高20位)

jalr $X_0, \text{offset}[11:0](r_a)$

(4) add rd, X₀, rs

(5) rdcycle rd: csrrs rd, cycle, X₀

(6) sext, w rd, rs : addiw rd, rs, 0

2-7 1) 短时间不会

2) bltu t₀, t₁, overflow

3) x86 和 ARM 架构 指令集规定了检测溢出的标志寄存器、无符号数
加法发生进位时，其中的进位标志会被置「有符号数加法发生溢出时，溢出标志会被置」

op = DIVU op = REMV op = DIV op = REM

2-8 1) $\underbrace{0x\text{ff}\dots f}_{16\text{个}}$ X $\underbrace{0x f\dots f}_{16\text{个}}$ X

2) NV：非法操作，表示发生了不合法的操作运算，如0除以0

VZ：除以零 OF：上溢出，表示发生了超过最大正数或最小负数的界限

NF：下溢出，表示发生了小于最小正数或最大负数的运算结果

NX：不精确，表示发生了舍入误差

不会，系统调用是用户应用主动发起的，除数为0仅仅是一种异常，操作系统会调用异常处理程序进行处理

3) 除数为0时程序有可能中断执行并进行异常处理，也有可能根据指
定的特殊情况，返回 default 值或特殊值(如无穷大或 NaN)

2.12.1) Linux Kernel: M 2) BootROM: M
3) Bootloader: M 4) USB driver: S (5) Vim: V

13.

mv t₃ zero
addi t₄, zero, 100

14. bge a₁, a₀, If

add a₂, a₀, a₁
j end

Loop: bge t₃, t₄, end
slli t₅, t₃, 2
add t₇, t₀, t₅
add t₈, t₁, t₅

l: sub a₂, a₀, a₁
end: nop

lw t₈, 0(t₈)
lw t₂, 0(t₂)
mul t₉, t₈, t₂
sw t₉, 0(t₇)
addi t₃, t₃, 1
j Loop

end: lw t₀, 0(t₀)

ret -

15. SW $t_0, 0(t_0)$

addi $t_1, zero, 3$.

SW $t_1, 4(t_0)$.

MV t_2, t_1

SLI $t_2, t_2, 2$

ADD a_2, t_0, t_2

SW $t_1, 0(a_2)$

Tb.

LW $t_3, 0(t_0)$

LW $t_4, 0(t_1)$

SW $t_4, 0(t_0)$

SW $t_3, 0(t_1)$.

RET.

17. 将 a_1 左移 30 次，得到 $a_1 = 2^{30}$