

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

1) $\text{addi } X_0, X_0, 0$

2) let 等价于 $\text{jair } X_0, 0(r_a)$

(2) $\text{auipc } ra, \text{offset}[31:12]$ (取高20位)

$\text{jair } X_0, \text{offset}[11:0](ra)$

(4) $\text{add } rd, X_0, rs$

(5) $\text{rdcycle rd: csr rs } rd, \text{cycle}, X_0$

(6) $\text{sext.w } rd, rs: \text{addiw } rd, rs, 0$

2-7 1) 暂时不会

2) $\text{bltu } t_0, t_1, \text{overflow}$

3) x86和ARM架构指令集规定了检测溢出的标志寄存器。无符号数加法和进位时，其中的进位标志会被置1，有符号数加法发生溢出时，溢出标志会被置1

$op = \text{DIVU}$

$op = \text{REMU}$

$op = \text{DIV}$

$op = \text{REM}$

2-8 1) $\text{oxff} \dots \text{f}$
16个

X

$\text{oxf} \dots \text{f}$
16个

X

2) NV: 非法操作，表示发生了不合法的浮点运算，如0除以0

DZ: 除以零 OF: 上溢出，表示发生了超过最大正数或最小负数的运算结果

UF: 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,系统调用是那/应用主动发出的,除数是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) 除数加0时程序有可能中断运行并进行异常处理,也有可能根据指令集定义的特殊情况,返回 default 值或特殊值(如无穷大或 NaN)

2.12 1) Linux Kernel: M 2) BootRom: M

3) Boot loader: M 4) USB driver: S (5) Vim: V

13.

mv t3 zero

addi t4, zero, 100

Loop: bge t3, t4, end

slli t5, t3, 2

add t7, t0, t5

add t8, t1, t5

14. bge a1, a0, if

add a2, a0, a1

j end

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 t0, 0(t0)

addi t1, zero, 3.

sw t1, 4(t0).

mv t2, t1

slli t2, t2, 2

add a2, t0, t2

sw t1, 0(a2)

16.

lw t3, 0(t0)

lw t4, 0(t1)

sw t4, 0(t0)

sw t3, 0(t1).

ret.

17. 将 a_1 左移 30 次, 得到 $a_1 = 2^{30}$