

3. 1) addi x0, x0, 0 2) ret: jalr x0, x1, 0

3) call offset : sllipc x6, offset [31:12]
 jalr x1, x6, offset [11:0]

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

5) rdcycle rd : csrrs rd, cycle, x0

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

7. 1) slti t3, t2, 0

 slti t4, t0, t1

2) add t0, t1, t2

 bltu t0, t1, overflow

3) 首先①负数+正数一定不会产生溢出

对于有符号数而言：②两正数相加得到负数或两负数相加得到正数，则溢出

③最高有效位产生进位，若结果为正数则溢出，若为负数则不溢出

无符号数：当结果小于两个加数时溢出。

8. (1) 指令 rs1 rs2 Op=DIVU时 Op=REMU时 Op=DIV时 Op=REM时
rd值 rd值 rd值 rd值

Op rd,rs1,rs2 X 0 0xfffffffffffffff X 0xfffffffffffffff X
=2⁶³-1 =-1

所以不会引起异常

原因：若会引发异常，则该异常会在大部分执行环境中会导致一个自陷。

而为了改进这一情况，会在除法操作后加一条分支指令。

(2) NV: 非法操作； DZ: 除以0； OF: 上溢； UF: 下溢； NX: 不精确

flags 被置位不会使处理器陷入系统调用，而是会设置异常标志，需要软件明确对该标志进行检查。

(3) 会产生一个内中断，暂停当下程序，跳转到对应的中断处理程序，执行完毕后再返回之前的程序执行。

12. (1) M 级，也可选 S 级，并在 M 级下运行

(2) M 级

(3) M 级 或 S 级

(4) M 级，要直接访问硬件资源

(5) U 级

13. li t₃, 0

li t₄, 100

lw a₅, 0(t₂)

j part1

part1: blt t₃, t₄, part2

j end

part2: addi t₃, t₃, 1
sll t₄, t₃, 2

add t₅, t₁, t₄

lw a₆, 0(t₅)

add t₅, t₀, t₄

mul a₇, a₆, a₅

sw a₇, 0(t₅)

j part2

end lw a₀, 0(t₀)

ret

14. part1: blt a₁, a₀, part2

add a₂, a₁, a₀

j end

part2: sub a₂, a₀, a₁

j end

end :

15. sw to , 0(t0)

li a1 , 3

sw t1 , 4(t0)

sll t2 , t1, 2

add t2 , t0, t2

sw t1 , 0(t2)

16. lw a1 , 0(t0)

lw a2 , 0(t1)

sw a2 , 0(t0)

sw a1 , 0(t1)

ret

17. int a;

int b = 1

int c = 30

for (a=0; a<=c; a++)

b = b * 2

return

实现的是计算 2^{30} .