

作业5

3. 1) nop addi x0, x0, 0 无操作
 2) ret ~~jalr~~ jalr x0, x1, 0 从子过程返回
 3) call offset auipc x1, offset [31:12] 远程调用子过程
 jalr x1, x1, offset [11:0]
 4) mv rd, rs addi rd, rs, 0 复制寄存器
 5) rdcycleh rd csrrs rd, cycleh, x0 读取周期计数器
 6) sext.w rd, rs addiw rd, rs, 0 有符号扩展字

7. ~~addi t3, t2, 100~~ 11) SLTI *t3, t2, 0

~~addi t4, t2, 100~~ SLT t4, t2, t1

12) ~~BLTZ~~ bltu t2, t1, overflow

13) x86通过DF检测加法是否溢出

rd值

8. 11)	指令	rs1	rs2	DIVU	REM	DIV	REM
	Op rd, rs1, rs2	X	0	0xfffffffffffffff	X	0xfffffffffffffff	X

不会。

允许除数为0, 可更好地扩大 RISC-V 应用范围

- 12) NX: 非精确异常 UF: 下溢异常 OF: 上溢异常
 DZ: 除0异常 NV: 无效操作数异常

不会

13) ①强制进入管理模式

②强制进入到 ARM 状态

③跳转至绝对地址 PC = 0x00000000 处执行

④禁止 IRQ 中断和 FIQ 中断, 复位后 CPSR 中最后4位状态为 0011, 并且进入管理模式, 执行操作系统程序, 然后切换回用户模式, 开始执行正常的用户程序



No.

Date.

12. (1) 用户模式

(2) 管理员模式

(3) 管理员模式

(4) 用户模式

(5) 机器模式

3. add x4, x0, x0 # i=0

addi x5, x0, 100

Loop:

bge x4, x5, exit

~~addi~~

~~addi~~ x6, x4, 2 # i x 4

~~addi~~

add x6, x6, a10

add x7, x6, a11

lw x6, 0(x6)

lw x7, 0(x7)

~~mul~~ x6, x7, a12

~~addi~~ x4, x4, 1

~~addi~~ x8, x0, a10

~~addi~~ x8, x0, a10

mul x6, x7, a12

addi x4, x4, 1

~~exit~~

exit:

add x8, x0, a10

~~addi~~ x8, x0, a10

mv a0, x8

addi sp, sp, -32

sd ra, 24(sp)

sd s0, 16(sp)

addi s0, sp, 32

mv a10, t0

mv a11, t1

mv a12, t2

sw a10, -20(sp)

sw a11, -20(sp)

ld ra, 24(sp)

ld s0, 16(sp)

addi sp, sp, 32

ret



扫描全能王 创建

14. `addi sp, sp, -32`

`sd ra, 24(sp)`

`sd s0, 16(sp)`

`addi s0, sp, 32`

`mv x10, a0`

`mv x11, a1`

`mv x12, a2`

`sw a10, -20(s0)`

`sw a11, -20(s0)`

`sw a12, -20(s0)`

`addi x13, x11, 1`

`bge a10, a13, if`

`sub x12 = x10 - x11`

1:

`add x12 = x10 + x11`

`mv a2, x12`

`mv a0, a2`

`ld ra, 24(sp)`

`ld s0, 16(sp)`

`addi sp, sp, 32`

`ret`

`lw x11, 0(x11)`

`addi x11, x0, 3`

`sll t5, t12, 2`

`add t5, t5, t0`

`lw t5, 0(t5)`

`addi t5, x0, 3`

16. `mv x10, t0`

`mv x11, t1`

`lw x10, 0(x10)`

`lw x11, 0(x11)`

`mv x9, x10`

`mv x10, x11`

`mv x11, x9`

`ret`

17. 求出 2^{30} 的取值

15. `addi t12, x0, 3`

`mv x10, t1`

`addi x11, x0, 1`

`sll x11, x11, 2`

`add x11, x11, t0`

