

C2

3.

1) $\text{nop} \rightarrow \text{addi } x_0, x_0, 0$

2) $\text{ret} \rightarrow \text{jalr } x_0, x_1, 0$

3) $\text{call offset} \rightarrow$

$\text{auipc } x_6, \text{offset}[31:12]$

$\text{jalr } x_1, x_6, \text{offset}[11:0]$

4) $\text{mv rd, rs} \rightarrow \text{addi rd, rs, 0}$

5) $\text{rdcycle rd} \rightarrow \text{csrrs rd, cycle, x0}$

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

8.

1) DIVU rd, x, 0

rd值为 $0xffffffffffffffff$

REMU rd, x, 0

rd值为 x

DIV rd, x, 0

rd值为 $0xffffffffffffffff$

REM rd, x, 0

rd值为 x

2) fflags 浮点异常标志域

若浮点运算单元在运算中出现相应异常，则将对应标志位置为1。且为一直保持累积，软件可以通过写0的方式单独清除某个异常标志位

7.

1) $\text{add } t_0, t_1, t_2$

$t_3, t_2, 0, 0$

$t_4, t_0, t_1, 1$

$t_3, t_4, \text{overflow}$

2) $\text{add } t_0, t_1, t_2$

$t_0, t_1, \text{overflow}$

NV Invalid Operation

DZ Divide by Zero

OF Overflow

UF Underflow

NX Inexact

3) X86: 当对有符号整数的add或sub指令溢出

处理器不会陷入系统调用

时，将设置溢出标志；而当对无符号整数的操作

3) X86: 系统会出现除法零中断

溢出时，将设置进位标志

DIV-0-TRP
ARM: 系统可配置控制寄存器CCR的DIV-0-TR

ARM: 通过CPSR的状态寄存器反映当前指令的

此位为0时，除以0操作不触发异常

溢出状态

此位为1时，除以0触发异常事件并产生相应中断

12.

1) Linux Kernel	1	S-Mode	part1: blt a ₁ , a ₀ , part2
2) BootROM	3	M-Mode	sub a ₂ , a ₀ , a ₁
3) BootLoader	3	M-Mode	j end
4) USB Driver	1	S-Mode	part2: add a ₂ , a ₀ , a ₁
5) Vim	0	U-Mode	end: mv a ₀ , a ₂

14.

13.	part1: lw a ₂ , 0(t2)	15.	part1: sw t0, 0(t0)
	mv a ₃ , t0		li t1, 3
	mv a ₄ , t1		sw t1, 4(t0)
	addi t1, t1, 400		li a ₂ , 4
part2:	beq a ₄ , t1, part3		mul a ₂ , a ₂ , t1
	lw a ₅ , 0(a3)		add a ₂ , a ₂ , t0
	lw a ₆ , 0(a4)		sw t1, 0(a2)
	mul a ₅ , a ₅ , a ₆		wolfrev, pt, et
	sw a ₅ , 0(a3)		sw a ₂ , 0(t1)
	addi a ₃ , a ₃ , 4	16.	part1: lw a ₂ , 0(t0)
	addi a ₄ , a ₄ , 4		lw a ₃ , 0(t1)
	j part2		sw a ₂ , 0(t1)
part3:	lw a ₀ , 0(t0)		sw a ₃ , 0(t0)

ret

17.

得到2的指数次幂(得到独热码)

本例：将 2^{30} 的值存入 a_1 寄存器

a_1 中有一个 $2^{(a_2 \text{ 的值})}$ 的值(独热码)