

7) $\text{slti } t3, t0, 0$

$\text{slli } t4, t0, 0$

2) $\text{add } t0, t1, t2$

$\text{sllr } t3, t0, t1$

$\text{bne } t3, t0, \text{overflow}$

(3) x86:

$\text{mov } \text{eax}, t1$

$\text{add } \text{eax}, t2$

$\text{cmp } \text{eax}, t1$

jb overflow

8) (1)

$\text{Op} = \text{DIVU}$

$\text{rd} = 0x\text{ffffffffffff}$

$\text{Op} = \text{REMU}$

$\text{rd} = X$

$\text{Op} = \text{DIV}$

$\text{rd} = 0x\text{ffffffffffff}$

$\text{Op} = \text{REM}$

$\text{rd} = X$

(2) 定点数除法:

NV

DZ

OF

UF

NX

非法操作

除以0

上溢

下溢

不精确

flags

被置位会起系统被置位

(3) x86 中 除数会引发 0 号中断

ARM 中没有除法指令, 除数为 0 的处理方法由软件定义.

等级

(2) Linux kernel	S S
BootROM	M M
BootLoader	M M
USB Driver	U U
vim	U U

(3) VecMul:

```
addi sp, sp, -32
sd ra, 24(sp)
sd s0, 16(sp)
addi s0, sp, 32
```

part1:

```
addi t3, zero, 0
addi t4, zero, 100
```

part2:

```
bge t3, t4, end ← muli t7, t3, 4,
add t5, t1, t7
lw t6, 0(t5)
mul t6, t6, t2
add t5, t0, t7
sw t6, 0(t5)
addi t3, t3, 1
j part2
```

end:

lw a0, 0(t0)

ld rd, 24(sp)

ld s0, 16(sp)

addi sp, sp, 32.

ret.

14.)

part1: bge a1, a0, part2

add a2, a0, a1

j Part3.

part2:

sub a2, a0, a1

part3:

15. A: sw t0, 0(t0)

addi t1, zero, 3

sw t1, 4(t0)

mulh t2, t1, 4.

add t2, t0, t2

sw t1, 0(t2)

16. swap:

addi sp, sp, -32

sd ra, 24(sp)

sd so, 16(sp)

addi so, sp, 32

lw t2, 0(t0) #

lw t3, 0(t1)

sw t3, 0(t0)

sw t2, 0(t1)

ld ra, 24(sp)

ld so, 16(sp)

addi sp, sp, 32

ret

17. A: 将1左移30位.