

3/21

3. (1) addi \$t0,\$t0,0 (2) jalr \$t0,\$t1,0
 (3) addi \$t0,\$t0,offset(\$t1:\$t2) jalr \$t1,\$t0,offset(\$t1:\$t2)
 (4) addi \$t0,\$t0,0 (5) csrrs \$t0,cycle,\$t0
 (6) addiw \$t0,\$t0,0

7 (1) slti \$t3,\$t1,0

slt \$t4,\$t0,\$t2.

(2) add \$t0,\$t1,\$t2

bltu \$t0,\$t1,overflow

$$(3) \left\{ \begin{array}{l} \text{判断进位与借位, 末位 } V = A_3B_3S_3 + \bar{A}_3\bar{B}_3S_3 \\ V=0 \text{ 无溢出 } V=1 \text{ 有溢出} \end{array} \right.$$

根据进位情况判断	补首位进位	最高数位进位
上溢	0	1
下溢	1	0

双进位 正数加, 负数减,

首位进位异或大于V.

V=0无溢出 V=1有溢出

< 归入类作业

1 2 3 4 5 6 7 8 9 (10)

8-11) OR 指令的执行 X OR 指令的执行 X

不考虑寄存器。

使程序运行更加顺畅，并使得除法指令更加通用，减少初学者报错。

(2) NV: 那次操作

D2: 寄存器为0

OF: 上溢

UF: 下溢

NX: Inexact.

不会。高效率自己对操作数位检查。

(3) 除数为0，则跳过该指令或引起除0异常并退出异常处理程序。

12-11) S 12, M 13, M 14) S

15) U



