

3. (1): call指令为调用子程序的指令

(2): ①取双指 PCout MOVb MARin

Read btl PCin
MDRout MOVb IRin

②取子程序

PCout MOVb MARin

Read btl Yin

MDRout MOVb PCin

③保存返回地址

SPout MOVb MARin

Yout MOVb MDRin

SPout b-1 spin

5. RegWt: 所有需将结果写入寄存器的指令 or/lu/R-Type等

RegDst: 存在两个寄存器的指令: R-Type

ALUSrc: 存在立即数的指令: i-Type, sw/lu等

Branch: 分支跳转指令

MemWt: sw指令

ExtOp: 需带符号扩展的指令: sw/lu oddi等

MemtoReg: lu指令

6. RegWt: 所有需将结果写入寄存器的指令: sw/lu等

RegDst: 只需要两个寄存器的指令: i-Type/sw/lu等

ALUSrc: 不存在立即数的指令: R-Type

Branch: 非分支跳转指令

MemWt: 非sw指令

ExtOp: 需无符号扩展的指令: ori/andi等

MemtoReg: 非lu指令

7. (1): xor \$rs \$rs \$rt

xor \$rt \$rs \$rt

xor \$rs \$rs \$rt

(2): 设占比为 X

使用伪指令: $3x + 1 - x = 1 + 2x$

使用硬码: 1-1

则 $1 + 2x > 1 - 1$

$x > 0.25$

即在大于25%时采用硬码

8. PCWt: 所有指令: 因为PC无法更新

MemtoReg: lu指令

IRWt: 所有指令: 因为IR无法写入

RegWt: 所有需将结果保存在寄存器中的指令如 R-Type

BrWt: Branch指令, 因为PC无法写入

MemWt: sw指令

PCWt cond: branch指令

R-Type: R-Type指令

9. PCWt: 执行过程中会出现问题

MemtoReg: 非lu指令

IRWt: 可能取错, 所有指令可能有问题

RegWt: 所有需保存结果的寄存器: sw等

BrWt: 非Branch指令

MemWt: 非sw指令

PCWt cond: 非branch指令有错误

R-Type: 非R-Type指令

扫码使用

夸克扫描王



10. loop: beq \$t3, \$zero, exit0
 lw \$t4, 0(\$t1)
 lw \$t5, 0(\$t2)
 bne \$t4, \$t5, exit1
 addi \$t1, \$t1, 4
 addi \$t2, \$t2, 4
 addi \$t3, \$t3, 1
 exit0: add \$t1, \$zero, \$t1
 exit1:

12. beq, lw, bne, addi 的时钟周期为 3.5, 3.4

假设有 x 段数据相同

① 总长为 x (执行 exit0)

loop: $3 + 5 + 5 + 3 + 4 + 4 + 4 = 28$

故需 $28x + 1$

② 总长大于 x (执行 exit1)

需 $28x$

13.

(1): 除数为 0 在取数/译码阶段检测

溢出在 R-型指令运算阶段检测

无效指令操作码在取数/译码阶段检测

无效指令地址在取指阶段检测

异常:

(2): 将异常原因写入异常查询寄存器 (cause), 根据异常类型跳转到相应异常处理程序

工作: 取异常, 处理异常

无效数据地址在存储器访问阶段检测

缺页在存储器访问与取指阶段检测

访问越权同上

异常中断在每一条指令执行完后检测

cause 寄存器判断

扫码使用

夸克扫描王

