P6 实验报告

14061075 修闽珂

1. 功能测试
2. 加减部分

#ADD、ADDU、SUB、SUBU、ADDI、ADDIU

#进行了正数对正数，负数对负数，正负数相对的测试

addi $1 $1 21512 #$1 is 21512

addi $3 $3 156 #$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

add $5 $1 $3#$5 is 21688

add $6 $2 $4 #$6 is -1801

add $7 $2 $3#$7 is -1100

addu $9 $1 $3#$9 is 21668

addu $10 $1 $2#$10 is 20256

addu $11 $2 $4#$11 is -1801

sub $12 $1 $2#$12 is 22768

sub $13 $2 $1#$13 is -22768

sub $14 $2 $4#$14 is -711

sub $15 $4 $2#$15 is 711

sub $16 $1 $3#$16 is 21356

subu $17 $3 $1#$17 is -21356

subu $18 $1 $2#$18 is 22768

subu $19 $2 $1#$19 is -22768

subu $20 $2 $4#$20 is -711

subu $21 $4 $2#$21 is 711

subu $22 $1 $3#$22 is 21356

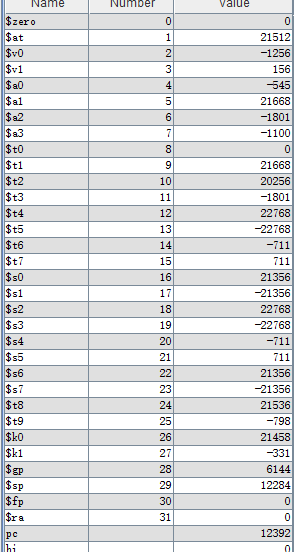
subu $23 $3 $1#$23 is -21356

addiu $24 $1 24#$24 is 21524

addiu $25 $2 458#$25 is -1238

addiu $26 $1 -54#$26 is 22437

addiu $27 $3 -487#$27 941

期望结果：

1. 分支部分

#BEQ、BNE、BLEZ、BGTZ、BLTZ、BGEZ

addi $1 $1 21512#$1 is 21512

addi $3 $3 156#$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

addi $5 $1 21512#$5 is 43024

#beq

beq $5 $1 \_beq#branch if $5 and $1 is equal

nop

subu $6 $5 $3#$6 is 7864320

\_beq: addi $8 $8 10 #$8 is 10

lui $7 120#$7 is 7864320

#bne

bne $1 $3 \_bne#branch if $1 is equal to $3

nop

xori $8 $2 10

\_bne: ori $9 $2 126#$9 is -1154

#blez

blez $2 \_blez#branch if $2 is smaller than or equal to 0

nop

add $9 $5 $3

\_blez: sll $10 $3 26#$10 is 1879048192

#bgtz

bgtz $3 \_bgtz#branch if $3 is greater than 0

nop

srlv $11 $9 $3

\_bgtz: srav $12 $7 $2#$12 is 0

#bltz

bltz $4 \_bltz#branch if $4 is smaller than 0

nop

addi $13 $13 20

\_bltz: addiu $14 $14 158#$14 is 158

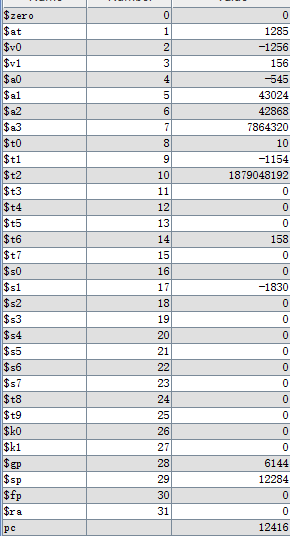
#bgez

bgez $0 \_bgez#branch if $0 is equal to or greater than 0

nop

addiu $15 $16 157

\_bgez: subiu $17 $4 1285 #$17 is -1830

期望结果：

1. 存储指令

#sb, sh, sw

addi $1 $1 510032#$1 is 510032

subu $2 $2 84135#$2 is 84135

#using @ as the symbol of data memory

sw $1 0($0)#@0 is 510032

sw $2 4($0)#@1 is 84135

addi $3 $3 5152#$3 is 5152

add $4 $3 $1#$4 is 89287

sub $5 $3 $4#$5 is -84135

ori $6 $7 10235#6 is 10235

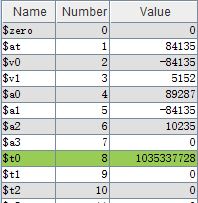
lui $8 15798#$8 is 1035337728

sb $6 8($0)#@2 is 251

sh $8 12($0)#@3 is 0

sb $6 12($0)#@3 is 251

sh $4 8($0)#@2 is 23571

期望结果：

1. 左右移

#SLL、SRL、SRA、SLLV、SRLV、SRAV、

addi $1 $1 21512#$1 is 21512

addi $3 $3 156#$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

sll $5 $1 10#$5 is 22028288

sll $6 $1 16#$6 is 54080000

sll $7 $2 10#$7 is ffec6000

sll $8 $2 19#$8 is d8c00000

srl $9 $1 10#$9 is 00000015

srl $10 $1 17#$10 is 00000000

srl $11 $2 10#$11 is 003ffffe

srl $12 $2 15#$12 is 0001ffff

sra $13 $1 10#$13 is 00000015

sra $14 $1 26#$14 is 00000000

sra $15 $2 10#$15 is fffffffe

sra $16 $2 23#$16 ffffffff

sllv $17 $1 $3#$17 80000000

sllv $18 $2 $1#$18 fffb1800

srlv $19 $1 $3#$19 00000000

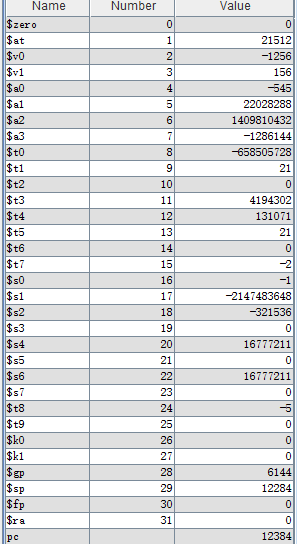
srlv $20 $2 $1#$20 00fffffb

srlv $21 $1 $3#$21 00000000

srlv $22 $2 $1 #$22 00fffffb

srav $23 $1 $3#$23 00000000

srav $24 $2 $1 #$24 fffffffb

期望结果：

1. 装载指令

#LB、LBU、LH、LHU、LW

addi $1 $1 510032#$1 is 510032

subu $2 $2 84135#$2 is 84135

#using @ as the symbol of data memory

sw $1 0($0)#@0 is 510032

sw $2 4($0)#@2 is 84135

addi $3 $3 5152#$3 is 5152

add $4 $3 $1#$4 is 89287

sub $5 $3 $4#$5 is -84135

ori $6 $7 10235#6 is 10235

lui $8 15798#$8 is 1035337728

sw $3 8($0) #@3 is 5152

sw $4 12($0)#@4 is 89287

sw $5 16($0)#@5 is -84135

sw $6 20($0)#@6 is 10235

sw $7 24($0)#@7 is 0

sw $8 28($0)#@8 is 1035337728

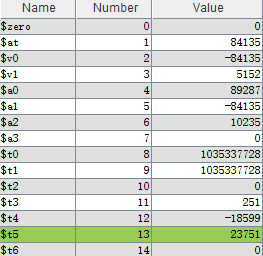
lw $9 28($0)#$9 is 1035337728

lb $10 24($0)#$10 is 0

lbu $11 20($0)#$11 is 251

lh $12 16($0)#$12 is -18599

lhu $13 12($0)#$13 is 23751

期望结果：

1. 转跳指令

addi $1 $1 21512#$1 is 21512

addi $3 $3 156#$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

#jump forward

j \_j

sll $6 $1 16#$6 is 54080000

sll $7 $2 10#$7 is ffec6000

#jump to later

\_j: sll $8 $2 19#$8 is d8c00000

srl $9 $1 10#$9 is 00000015

\_jal: srl $10 $1 17#$10 is 00000000

srl $12 $2 15#$12 is 0001ffff

sllv $17 $1 $3#$17 80000000

beq $30 $31 to

addi $30 $30 12356

sra $13 $1 10#$13 is 00000015

jal \_jal#jump back

sra $14 $1 26#$14 is 00000000

to: nop

addi $31 $0 12376

jr $31#jump register

sub $15 $4 $2#$15 is 711

sub $16 $1 $3#$16 is 21356

subu $17 $3 $1#$17 is -21356

addiu $24 $1 4542#$24 is 21524

jal \_jal1 #jump and link

addiu $25 $2 -48#$25 is -1238

addiu $26 $1 -785#$26 is 22437

\_jal1: subu $19 $2 $1#$19 is -22768

subu $20 $2 $4#$20 is -711

subu $21 $4 $2#$21 is 711

addi $31 $0 12424

jalr $27 $31#jump to $31 while store the pc + 8 to $27

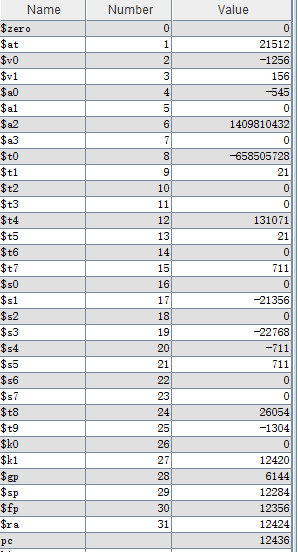
srl $10 $1 17#$10 is 00000000

srl $11 $2 10#$11 is 003ffffe

srl $12 $2 15#$12 is 0001ffff

sra $13 $1 10#$13 is 00000015

nop

期望结果：

1. 乘除法相关指令

#MULT、MULTU、DIV、DIVU、MFHI、MFLO、MTHI、MTLO

addi $1 $1 21512#$1 is 21512

addi $3 $3 156#$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

#mult

mult $1 $3#21512 \* 156

mfhi $5#$5 is 0

mflo $6#$6 is 3355872

mult $1 $2#21512 \* 156

mfhi $7#$7 is -1

mflo $8#$8 is -27019072

mult $2 $4#-1256 \* -545

mfhi $9#$9 is 0

mflo $10#$10 is 000a71e8

#multu

multu $1 $3 #21512 \* 156

mfhi $11#$11 is 0000000

mflo $12#$003334e0

multu $1 $2 #21512 \* -1256

mfhi $13#$13 is 00005407

mflo $14#$14 is fe63b8c0

multu $2 $4

mfhi $15#$15 is fffff8f7

mflo $16#$16 is 000a71e8

#div

div $1 $3

mfhi $17#$17 is 00000008c

mflo $18#$18 is 00000089

div $1 $2

mfhi $19#$19 is 000000a0

mflo $20#$20 is ffffffef

div $2 $4

mfhi $21#$21 is ffffff5a

mflo $22#$22 is 00000002

#divu

divu $1 $3

mfhi $23#$23 0000008c

mflo $24#$24 is 00000089

divu $1 $2

mfhi $25#$25 is 00005408

mflo $26#$26 is 0

divu $2 $4

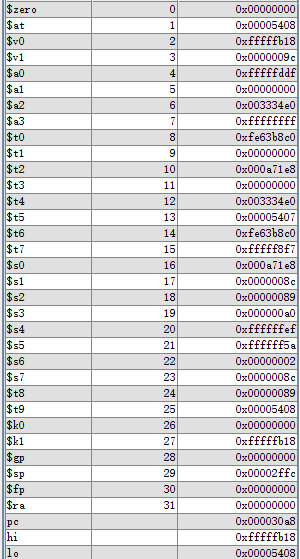
mfhi $27#$27 is fffffb18

mflo $28#$$28 is 0

#mtlo, mthi

mtlo $1#$hi is fffffb18

mthi $2#$lo is 00005408

期望结果： 

1. 逻辑指令

#AND、OR、XOR、NOR、ORI、XORI、LUI、SLT、SLTI、SLTIU、SLTU

addi $1 $1 21512#$1 is 21512

addi $3 $3 156#$2 is -1256

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

and $5 $1 $3#$5 00000008

and $6 $1 $2#$6 00005008

and $7 $2 $4#$7 fffff918

#ori

or $8 $1 $3

or $9 $1 $2

or $10 $2 $4

#xor

xor $11 $1 $3

xor $12 $2 $4

xor $13 $1 $2

#nor

nor $14 $1 $3

nor $15 $2 $4

nor $16 $1 $2

#nor

nor $17 $1 $3

nor $18 $2 $4

nor $19 $1 $2

#slt

slt $20 $1 $3

slt $21 $2 $4

slt $22 $1 $2

#sslti

slti $23 $1 156189

slti $24 $1 125

#

sltiu $25 $2 1521

sltiu $26 $2 -1300

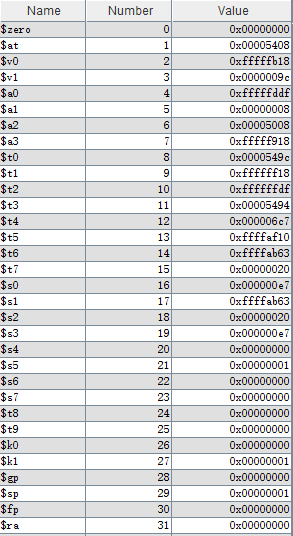
#sltiu

sltu $27 $3 $1

sltu $28 $1 $3

sltu $29 $3 $4

#

期望结果： 

1. 冒险测试

#harzard test

#指令被分成以下几类，每一类都有一个代表。

#装载sw、存储lw、alu指令addi(ALU指令无论结果是存入rt还是rd，在仿真的时候总是会根据控制信号选择

#最终存入的寄存器是rt还是rd，并将其视为Rd，最终存入的寄存器)

#两个对比数的分支beq、转跳的r型指令jr、乘除法mult、mthi、mfhi

#addi vs sw

#冒险在rt

addi $1 $1 23

sw $1 0($0)

addi $2 $2 21

nop

sw $2 4($0)

addi $3 $3 21

nop

nop

sw $3 8($0)

#冒险在rs

addi $4 $4 16

sw $3 ($4)

addi $5 $5 20

#lw vs addi

#id-ex

lw $6 ($16)

addi $7 $6 153

#id-mem

lw $7 8($0)

nop

addi $8 $7 564

#id-wb

lw $8 4($0)

nop

nop

addi $9 $8 48

#beq vs lw

#如果有冲突要暂停两个周期

lw $8 4($0)

beq $8 $9 heheda

nop

#暂停一个周期

lw $8 4($0)

nop

beq $8 $6 kengdiea

nop

#

lw $8 16($0)

nop

nop

beq $9 $8 douruimi

nop

#beq vs addi

#暂停一个周期，等addi到mem阶段转发

addi $9 $8 2132

beq $9 $8 \_beqaddi0

nop

#jr vs lw

#暂停两个周期

addi $10 $0 12464

sw $10 16($0)

lw $10 16($0)

jr $10

nop

#暂停一个周期

lw $12 16($0)

nop

jr $12

#乘除法

#乘除法结果要在10或5个周期后才会写入到hi和lo寄存器中，然而这两个寄存器的值如果没有mflo和mfhi就没有意义

#所以关键的暂停在于能否在mult和div等指令与其紧邻的mflo等指令之间插入暂停

#mflo和mfhi其实和其他的ALU指令一样，也要写入rd寄存器，经过选择器选择WData的来源就可以使写入到GPR的值为$HI或者$LO

#如果有指令需要用到mfhi和mflo的结果，依然可以用原来的的转发机制。

#暂停5周期

mult $2 $1

mfhi $13

#暂停

mult $10 $7

addi $3 $3 156#$2 is -1256

mflo $15

#不暂停

mult $10 $9

addi $2 $2 -1256#$3 is 156

addi $4 $4 -545#$4 is -545

add $5 $1 $3#$5 is 21688

add $6 $2 $4 #$6 is -1801

addu $10 $1 $2#$10 is 20256

mfhi $16

#mult和mthi

#暂停五个周期，其他暂停机制同上，因为乘除法的busy信号有效的时候与乘除法器有关的一切在；指令都会被暂停

mult $15 $6

mtlo $8

#mthi vs mfhi

mthi $9

mfhi $8

#mthi vs alu

#转发

addi $15 $0 1561

mtlo $15

sub $14 $2 $4#$14 is -711

sub $15 $4 $2#$15 is 711

sub $16 $1 $3#$16 is 21356

subu $17 $3 $1#$17 is -21356

heheda: subu $18 $1 $2#$18 is 22768

subu $19 $2 $1#$19 is -22768

subu $20 $2 $4#$20 is -711

subu $21 $4 $2#$21 is 711

kengdiea: subu $22 $1 $3#$22 is 21356

subu $23 $3 $1#$23 is -21356

addiu $24 $1 24#$24 is 21524

douruimi:addiu $25 $2 458#$25 is -1238

addiu $26 $1 -54#$26 is 22437

\_beqaddi0: addiu $27 $3 -487#$27 941

期望结果：