P5实验报告

14061075 修闽珂

Ori 指令只有RT 和RS寄存器，与立即数的与的和存在于rt中。经过RegDst的选择可以选择rt与rd（在立即数中，没有rd，所以rd在这里的值没有意义），而在此情况下RegDst只能是选择rt作为存入地址的。所以后来的流水级别中，rd为EX级的rt。

Lui指令中只有一个寄存器，此时选择立即数的扩展做为输入。通过ALU来决定输出。

Addu和subu很类似，没有什么区别。都利用了三个寄存器。RegDst决定存入的寄存器是指令中包含的rd段落所对应的寄存器。

这三类指令应该互为前提并充分考虑冒险机制。冲突位置可以是RT或者RS， 中间插有n条无关指令时情况都不一样。Ori后应该分别连接lui，3种冲突，addu/subu 6种；lui后面跟着ori 2\*3 = 6种和addu/subu2 \* 3 = 6种；addu/subu后面连接ori指令有2 \* 3 =6和lui指令有1 \* 3 = 3种。共30种

测试代码：

ori $1, $1, 10# $1 =10

ori $2, $2, 20#$2 = 20

ori $3, $3, 30#$3 = 30

addu $4, $1, $2#$4 = 30

subu $5, $2, $3#$5 = -10

lui $6, 50#$6 = 3276800

addu $4, $5, $2#$4 = 10

addu $7, $0, $3#$7 = 30

addu $1, $0, $6#$1 = 3276800

j nima#

subu $1, $2 $5

lui $1, 26

nima: ori $3 $2 28#$3 = 28

addu $2, $3, $4#$2 = 38

addu $6 $3 $2#$6 = 66

ori $7, $8, 56#$7 = 56

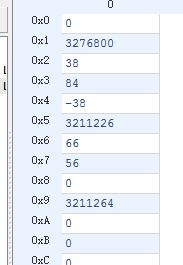
addu $3, $7, $3#$3 = 84

subu $4, $9 $2#$4 = -38

lui $9, 49#$9= 3211264

addu $5 $4 $9#$5 = 3211226

期望结果：



测试数据：

ori $1, $1, 10# $1 =10

ori $2, $2, 20#$2 = 20

ori $3, $3, 30#$3 = 30

addu $4, $1, $2#$4 = 30

subu $5, $2, $3#$5 = -10

lui $6, 50#$6 = 3276800

addu $4, $5, $2#$4 = 10

addu $7, $0, $3#$7 = 30

addu $1, $0, $6#$1 = 3276800

j nima#

subu $1, $2 $5

lui $1, 26

nima: ori $3 $2 28#$3 = 28

addu $2, $3, $4#$2 = 38

addu $6 $3 $2#$6 = 66

ori $7, $8, 56#$7 = 56

addu $3, $7, $3#$3 = 84

subu $4, $9 $2#$4 = -38

lui $9, 49#$9= 3211264

addu $5 $4 $9#$5 = 3211226

addu $10 $5 $3#$10 = 3211310

addu $11 $10 $9#$11 6422574

lui $20 47#3080192

addu $12 $11 $3#6422658

subu $30 $8 $7#-56

subu $29 $4 $6#-104

ori $13 $12 23#6422679

subu $28 $6 $3#-18

ori $27, $8 34#34

ori $14 $10 42#3211310

lui $15 19#1245184

ori $26 $8 14#14

addu $16 $15 $6#1245250

ori $6 $5 20 #3211230

ori $7 $8 59#59

期望结果

