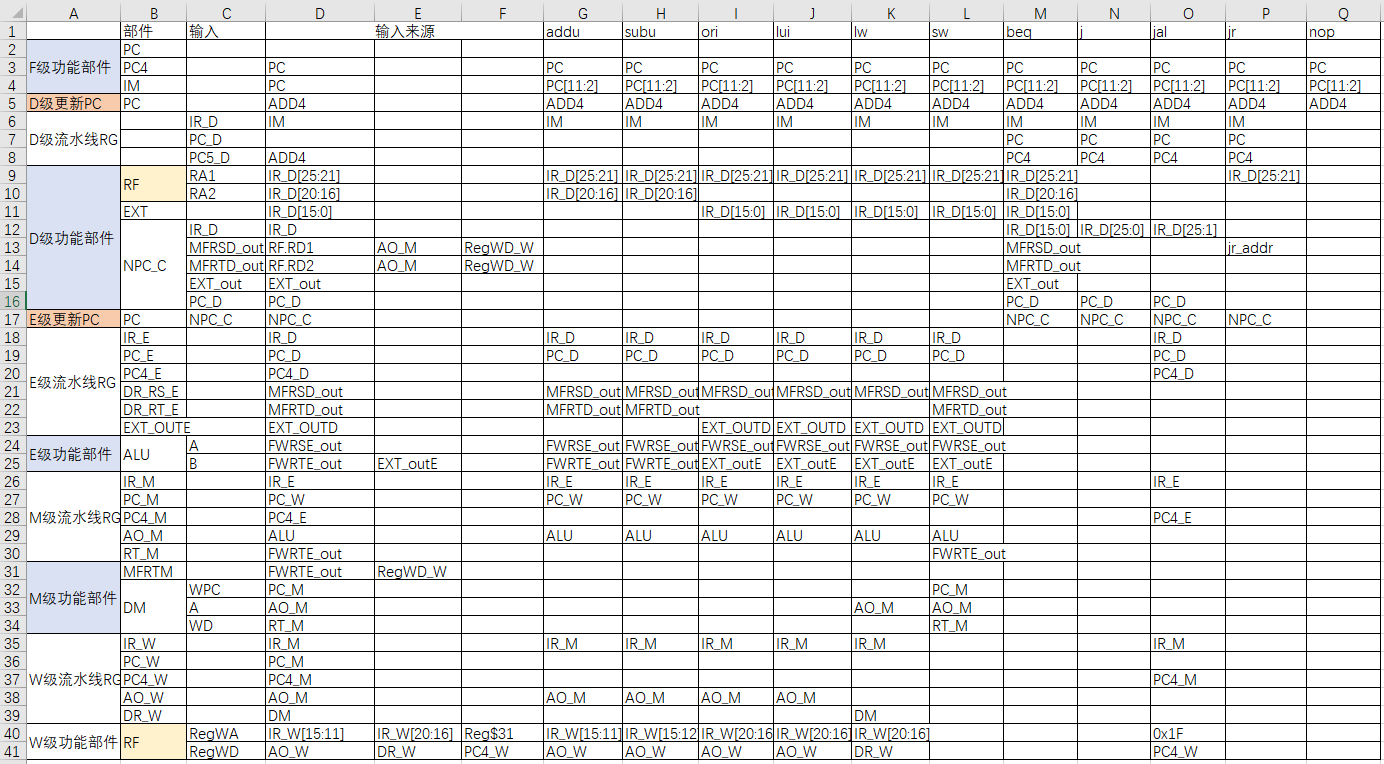
verilog单周期CPU设计文档

1. 设计与测试说明
   1. 处理器应支持的指令集为：MIPS-lite2={ addu, subu, ori, lw, sw, beq, lui, j, jal, jr, nop }
   2. 处理器为流水线设计。
2. 数据通路设计
3. 控制器设计

Controller模块整体进行设计，输入指令，输出指令相应的信号。mips.v中进行模块实例化和连接时，在每个流水级分别输出该流水级需要的信号。STALL和FORWARD单独设计。

1.Controller模块

CONTROLLER(INSTR,ALU\_OP,ALU\_Src,EXT\_OP,RegDst,RegWrite,MemtoReg,b\_pc,j\_pc,NPC\_sel,MemWrite,cal\_r,cal\_i,load,store,jal,jr);

其中ALU\_OP, RegWrite, MemWrite等信号控制ALU，GRF，DM等功能部件的功能以及多选器对于输入数据和地址的选择。b\_pc, j\_pc, cal\_r等信号用于识别指令的类型。jal, jr信号识别具体指令jal, jr.

Table 指令对应信号值

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | addu | subu | ori | sw | lw | lui | beq | j | jal | jr |
| IR[31:26] | 000000 | 000000 | 001101 | 101011 | 100011 | 001111 | 000100 | 000010 | 000011 | 000000 |
| IR[5:0] | 100001 | 100011 | x | x | x | x | x | x | x | 001000 |
| ALU\_Src | 0 | 0 | 1 | 1 | 1 | 1 | x | x | x | x |
| ALU\_OP | 000 | 001 | 010 | 000 | 000 | 000 | x | x | x | x |
| RegDst | 01 | 01 | 00 | x | 00 | 00 | x | x | 10 | xx |
| MemtoReg | 00 | 00 | 00 | xx | 01 | 00 | xx | xx | 10 | Xx |
| EXT\_op | x | x | 00 | 10 | 10 | 01 | 11 | x | x | x |
| ALU\_Src | 00 | 00 | 01 | 01 | 01 | 01 | xx | xx | xx | xx |
| NPC\_sel | x | x | x | x | x | x | 0 | 01 | 01 | 10 |
| RegWrite | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| MemWrite | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| b\_pc | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| j\_pc | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| cal\_i | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| cal\_r | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| load | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| store | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| jal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| jr | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 部分信号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号 | 含义 | 值 | 选择/功能 | 对应指令 |
| RegDst | 选择写入寄存器的地址 | 0 | rt | lui, ori, lw |
| 1 | rd | addu, subu |
| 2 | $31 | jal |
| MemtoReg | 选择写入寄存器的数据 | 0 | ALU\_out | lui, ori, addu, subu |
| 1 | DM\_RD | lw |
| 2 | PC+4(PC+8) | jal |
| ALU\_OP | ALU运算符 | 0 | 加 | addu, lw, sw, lui |
| 1 | 减 | subu |
| 2 | 或 | ori |
| ALU\_Src | 选择输入ALU\_B数据 | 0 | RF\_RD2 | addu, subu |
| 1 | EXT\_out | lui, ori, lw, sw |
| EXT\_OP | 扩展符号 | 0 | zero\_ext | ori |
| 1 | imm移至高位 | lui |
| 2 | sign\_ext | lw, sw |
| 3 | beq\_ext | beq |
| NPC\_sel | 选择NPC | 0 | PC+4+EXT\_out | beq |
| 1 | PC[31:28] || IR[25:21] ||00 | jal, j |
| 2 | RF\_RD1 | jr |

1. STALL模块

根据Tnew和Tuse分析需要暂停的情况，暂停序列有: 1. 1.beq\_D(rs/rt)-cal\_rD(rd)/cal\_iD(rt)/loadE(rt)/loadM(rt)

2.cal\_rD(rs/rt)-loadM(rt), cal\_iD(rs)-loadE(rt)

3.loadD(rs)-loadE(rt)

4.storeD(rs)-loadE(rt)

5.jr\_D(rs)-cal\_rD(rd)/cal\_iD(rt)/loadE(rt)/loadM(rt)

1. FORWARD模块

直接根据流水级指令由分布式控制器输出的RegWrite, RegDst, MemtoReg 信号，再设置流水级多选器RegWA\_M和RegWA\_W，输出流水级写寄存器地址M\_WA和W\_WA.根据寄存器写入地址和读地址来判断转发情况，避免转发多选器信号定义过于复杂。

Table 转发信号

|  |  |  |  |
| --- | --- | --- | --- |
| 转发信号 | 取值 | 条件 | 数据 |
| FWRSD | 0 | else | RF\_RD1 |
| 1 | RegWriteM && M\_WA!=0 && D\_rs==M\_WA | AO\_M |
| 2 | RegWriteW && W\_WA!=0 && D\_rs==W\_WA | RegWD\_W |
| FWRTD | 0 | else | RF\_RD2 |
| 1 | RegWriteM && M\_WA!=0 && D\_rt==M\_WA | AO\_M |
| 2 | RegWriteW && W\_WA!=0 && D\_rt==W\_WA | RegWD\_W |
| FWRSE | 0 | else | RS\_E |
| 1 | RegWriteM && M\_WA!=0 && E\_rs==M\_WA | AO\_M |
| 2 | RegWriteW && W\_WA!=0 && E\_rs==W\_WA | RegWD\_W |
| FWRTE | 0 | else | RT\_E |
| 1 | RegWriteM && M\_WA!=0 && E\_rt==M\_WA | AO\_M |
| 2 | RegWriteW && W\_WA!=0 && E\_rt==W\_WA | RegWD\_W |
| FWRTM | 0 | else | RT\_M |
| 1 | loadW && W\_WA!=0 && M\_rt==W\_WA | RegWD\_W |

除了ID级需要用到寄存器数据的beq, jr指令，对于下一级需要用到寄存器值的指令比如cal\_r也进行了转发，再将转发的结果传入流水线寄存器作为RF\_RD1和RF\_RD2. 此处考虑到以下特殊的几点：

1. loadM且与D级指令数据冲突，以cal\_rD为例，转发到ID的数据为AO\_M, 并不符合cal\_r读寄存器的要求，但当cal\_r到达E级时，对于ALU\_A和ALU\_B会进行重新选择，发生冲突的寄存器转发信号为2，实际参加运算的数据为将要写入寄存器的数据。
2. 在一些满足暂停条件的情况下也满足转发条件，但因pc\_en的优先级较高，并不会以错误的转发数据进行运算，不影响最终结果。以jr\_D和loadM且数据冲突为例，转发到ID的RF\_RD1的数据为AO\_M, 但此时满足了暂停条件，暂停条件的优先级较高，PC停止计数，下一时钟周期jr\_D && loadW, 转发到ID的数据为RegWD\_W.
3. FWRTM写寄存器只考虑loadW，因为其他写寄存器的指令都在前一时钟周期经ALU得出了结果，且转发至EX做为RF\_RD2E.之后在下一时钟周期存入M流水级寄存器。
4. 测试程序

Table addu冲突测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 | 初始化 |
| r-M-rs | subu | rs | subu $1, $2, $3 | 430@00003044: $ 1 <= ffff0000 | lui $1,1 |
| addu $4, $1, $2 | 450@00003048: $ 4 <= 00010000 | lui $2,2 |
| r-M-rs | subu | rt | subu $5, $6, $7 | 470@0000304c: $ 5 <= ffff0000 | lui $3,3 |
| addu $8, $6, $5 | 490@00003050: $ 8 <= 00050000 | lui $4,4 |
| r-W-rs | subu | rs | subu $9, $10, $11 | 510@00003054: $ 9 <= ffff0000 | lui $5,5 |
| lui $12 1 | 530@00003058: $12 <= 00010000 | lui $6,6 |
| addu $13, $9, $10 | 550@0000305c: $13 <= 00090000 | lui $7,7 |
| r-W-rt | subu | rt | subu $14, $15, $16 | 570@00003060: $14 <= ffff0000 | lui $8,8 |
| lui $17 1 | 590@00003064: $17 <= 00010000 | lui $9,9 |
| addu $17,$15,$14 | 610@00003068: $17 <= 000e0000 | lui $10,10 |
| i-M-rs | ori | rs | ori $1, $2, 1 | 630@0000306c: $ 1 <= 00020001 | lui $11,11 |
| addu $4, $1, $2 | 650@00003070: $ 4 <= 00040001 | lui $12,12 |
| i-M-rt | ori | rt | ori $5, $6, 1 | 670@00003074: $ 5 <= 00060001 | lui $13,13 |
| addu $7, $6, $5 | 690@00003078: $ 7 <= 000c0001 | lui $14,14 |
| i-W-rs | ori | rs | ori $3, $8, 1 | 710@0000307c: $ 3 <= 00050001 | lui $15,15 |
| lui $9, 9 | 730@00003080: $ 9 <= 00090000 | lui $16,16 |
| addu $10, $3, $8 | 750@00003084: $10 <= 000a0001 | lui $17,17 |
| i-W-rs | ori | rt | ori $11, $8, 2 | 770@00003088: $11 <= 00050002 |  |
| lui $9, 10 | 790@0000308c: $ 9 <= 000a0000 |  |
| addu $10, $11, $8 | 810@00003090: $10 <= 000a0002 |  |
| ld-W-rs | lw | rs | lw $6,0($1) | 270@00003024: $ 6 <= 00030000 | lui $2,1 |
| addu $6,$6,$2 | 310@00003028: $ 6 <= 00040000 | lui $3,3 |
| ld-W-rt | lw | rt | lw $8,4($1) | 330@0000302c: $ 8 <= 00040000 | lui $4,4 |
| addu $9,$7,$8 | 370@00003030: $ 9 <= 000b0000 | lui $5,5 |
| ld-M-rs | lw | rs | lw $10,8($1) | 390@00003034: $10 <= 00050000 | lui $7,7 |
| lui $17,17 | 410@00003038: $17 <= 00110000 | sw $3,0($0) |
| addu $11,$10,$7 | 430@0000303c: $11 <= 000c0000 | sw $4,4($0) |
| ld-M-rt | lw | rt | lw $12,0($1) | 450@00003040: $12 <= 00030000 | sw $5,8($0) |
| lui $17,18 | 470@00003044: $17 <= 00120000 | sw $2,12($0) |
| addu $13,$7,$12 | 490@00003048: $13 <= 000a0000 |  |

Table ori冲突测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 | 初始化 |
| i-M-rs | ori | rs | ori $2,$1,1 | 250@00003020: $ 2 <= 00010001 | lui $1,1 |
| ori $3,$2,1300 | 270@00003024: $ 3 <= 00010515 | lui $2,2 |
| i-W-rs | ori | rs | ori $4,$5,100 | 290@00003028: $ 4 <= 00050064 | lui $3,3 |
| lui $9,9 | 310@0000302c: $ 9 <= 00090000 | lui $4,4 |
| ori $5,$6,0x1ff | 330@00003030: $ 5 <= 000601ff | lui $5,5 |
| r-M-rs | addu | rs | addu $1,$6,$2 | 350@00003034: $ 1 <= 00070001 | lui $6,6 |
| ori $7,$1,0xfff | 370@00003038: $ 7 <= 00070fff | lui $7,7 |
| r-W-rs | subu | rs | subu $2,$3,$1 | 390@0000303c: $ 2 <= fffa0514 | lui $8,8 |
| lui $9,10 | 410@00003040: $ 9 <= 000a0000 |  |
| ori $4,$9,0xfff | 430@00003044: $ 4 <= 000a0fff | lui $2,1 |
| ld-M-rs | lw | rs | lw $6,0($1) | 270@00003024: $ 6 <= 00030000 | lui $3,3 |
| ori $6,$6,0xf | 310@00003028: $ 6 <= 0003000f | lui $4,4 |
| ld-W-rs | lw | rs | lw $10,8($1) | 330@0000302c: $10 <= 00050000 | lui $5,5 |
| lui $17,17 | 350@00003030: $17 <= 00110000 | lui $7,7 |
| ori $11,$10,0xff | 370@00003034: $11 <= 000500ff | sw $3,0($0) |
|  |  |  |  |  | sw $4,4($0) |
|  |  |  |  |  | sw $5,8($0) |
|  |  |  |  |  | sw $2,12($0) |

Table lw冲突测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 | 初始化 |
| i-M-rs | ori | rs | ori $1,$0,4 | 310@0000302c: $ 1 <= 00000004 | lui $1,1 |
| lw $2,0($1) | 330@00003030: $ 2 <= 00020000 | lui $2,2 |
| r-M-rs | addu | rs | addu $2,$0,$0 | 350@00003034: $ 2 <= 00000000 | lui $3,3 |
| lw $3,4($2) | 370@00003038: $ 3 <= 00020000 | lui $4,4 |
| i-W-rs | ori | rs | ori $3,$0,0 | 390@0000303c: $ 3 <= 00000000 | lui $5,5 |
| lui $8,8 | 410@00003040: $ 8 <= 00080000 | lui $6,6 |
| lw $7,8($3) | 430@00003044: $ 7 <= 00030000 | lui $7,7 |
| r-W-rs | subu | rs | subu $5,$4,$4 | 450@00003048: $ 5 <= 00000000 | sw $1,0($0) |
| lui $9,9 | 470@0000304c: $ 9 <= 00090000 | sw $2,4($0) |
| lw $9,12($5) | 490@00003050: $ 9 <= 00040000 | sw $3,8($0) |
|  |  |  |  |  | sw $4,12($0) |
| lw-M-rs | lw | rs | lw $5,12($0) | 130@00003008: $ 5 <= 00000000 | lui $5,5 |
| sw $7,0($5) | 170@0000300c: $ 6 <= 00000000 | lui $7,7 |
| ld-W-rs | lw | rs | lw $6,12($0) | 190@00003010: $ 7 <= 00000000 |  |
| lui $7,7 | 210@00003014: $ 8 <= 00080000 |  |
| sw $2,0($6) | 230@00003018: $ 9 <= 00000000 |  |

Table sw冲突测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 | 初始化 |
| i-M-rs | ori | rs | ori $1,$0,4 | 90@00003000: $ 2 <= 00010000 | lui $2,1 |
| sw $2,0($1) | 110@00003004: $ 3 <= 00030000 | lui $3,3 |
| r-M-rs | addu | rs | addu $2,$0,$1 | 130@00003008: $ 4 <= 00040000 | lui $4,4 |
| sw $3,0($2) | 150@0000300c: $ 5 <= 00050000 | lui $5,5 |
| i-W-rs | ori | rs | ori $3,$0,0 | 170@00003010: $ 7 <= 00070000 | lui $7,7 |
| lui $8,8 | 190@00003014: $ 1 <= 00000004 |  |
| sw $7,0($3) | 190@00003018: \*00000004 <= 00010000 |
| r-W-rs | subu | rs | subu $5,$4,$4 | 230@0000301c: $ 2 <= 00000004 |
| lui $9,9 | 230@00003020: \*00000004 <= 00030000 |
| sw $9,4($5) | 270@00003024: $ 3 <= 00000000 |
| i-M-rt | ori | rt | ori $7,$6,0xff | 290@00003028: $ 8 <= 00080000 |
| sw $7,4($0) | 290@0000302c: \*00000000 <= 00070000 |
| r-M-rt | addu | rt | addu $5,$4,$6 | 330@00003030: $ 5 <= 00000000 |
| sw $5,0($0) | 350@00003034: $ 9 <= 00090000 |
| ld-M-rs | lw | rs | lw $5,12($0) | 270@00003024: $ 5 <= 00000004 | ori $1,$0,1 |
| sw $7,0($5) | 290@00003028: \*00000004 <= 00070000 | ori $2,$0,2 |
| ld-W-rs | lw | rs | lw $6,12($0) | 330@0000302c: $ 6 <= 00000004 | ori $3,$0,3 |
| lui $7,7 | 350@00003030: $ 7 <= 00070000 | ori $4,$0,4 |
| sw $2,0($6) | 350@00003034: \*00000004 <= 00000002 | lui $7,7 |
| lw-M-rt | lw | rt | lw $9,8($0) | 390@00003038: $ 9 <= 00000003 | sw $1,0($0) |
| sw $9,12($0) | 390@0000303c: \*0000000c <= 00000003 | sw $2,4($0) |
| lw-W-rt | lw | rt | lw $8,0($0) | 430@00003040: $ 8 <= 00000001 | sw $3,8($0) |
| lui $9,9 | 450@00003044: $ 9 <= 00090000 | sw $4,12($0) |
| sw $8,4($0) | 450@00003048: \*00000004 <= 00000001 |  |

Table beq冲突测试-1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 | 测试代码 | |
| i-E-rs | ori | rs | ori $2,$1,0 | 跳转 | label0: | label6: |
| beq $2,$1,label1 | lui $1,1 | ori $7,$1,0 |
| r-E-rs | addu | rs | addu $4,$1,$0 | 跳转 | ori $2,$1,0 | nop |
| beq $4,$1,label2 | beq $2,$1,label1 | beq $1,$7,label7 |
| i-M-rs | ori | rs | ori $3,$1,0 | 跳转 | nop | nop |
| nop | label2: | label8: |
| beq $3,$1,label3 | ori $3,$1,0 | lw $7,0($0) |
| r-M-rs | subu | rs | subu $4,$1,$3 | 跳转 | nop | beq $7,$0,label9 |
| nop | beq $3,$1,label3 | nop |
| beq $4,$0,label4 | nop | label7: |
| i-E-rt | ori | rt | ori $4,$1,0 | 跳转 | label1: | addu $8,$1,$0 |
| beq $1,$4,label5 | addu $4,$1,$0 | nop |
| r-E-rt | addu | rt | addu $5,$1,$0 | 跳转 | beq $4,$1,label2 | beq $1,$8,label8 |
| beq $1,$5,label6 | nop | nop |
| i-M-rt | ori | rt | ori $7,$1,0 | 跳转 | label3: | label9: |
| nop | subu $4,$1,$3 | lw $8,0($0) |
| beq $1,$7,label7 | nop | beq $0,$8,label10 |
| r-M-rt | addu | rt | addu $8,$1,$0 | 跳转 | beq $4,$0,label4 | nop |
| nop | nop | label11: |
| beq $1,$8,label8 | label5: | lw $4,0($0) |
| lw-E-rs | lw | rs | lw $7,0($0) | 跳转 | addu $5,$1,$0 | nop |
| beq $7,$0,label9 | beq $1,$5,label6 | beq $0,$4,label12 |
| lw-E-rt | lw | rt | lw $8,0($0) | 跳转 | nop | nop |
| beq $0,$8,label10 | label4: | label10: |
| lw-M-rs | lw | rs | lw $5,0($0) | 跳转 | ori $4,$1,0 | lw $5,0($0) |
| nop | beq $1,$4,label5 | nop |
| beq $5,$0,label11 | nop | beq $5,$0,label11 |
| lw-M-rt | lw | rt | lw $4,0($0) | 跳转 |  | nop |
| nop |  | label12: |
| beq $0,$4,label12 |  | lui $12,12 |

Table beq冲突测试-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 |
| i-W-rs | ori | rs | ori $2,$1,0 | 跳转正确 |
| nop |
| nop |
| beq $2,$1,label1 |
| r-W-rs | addu | rs | addu $3,$0,$1 | 跳转正确 |
| nop |
| nop |
| beq $3,$1,label2 |
| lw-W-rs | lw | rs | lw $4,0($0) | 跳转正确 |
| nop |
| nop |
| beq $4,$1,label3 |
| i-W-rs | ori | rt | ori $5,$1,0 | 跳转正确 |
| nop |
| nop |
| beq $1,$5,label4 |
| r-W-rs | addu | rt | lw $7,4($0) | 跳转正确 |
| nop |
| nop |
| beq $1,$7,end |
| lw-W-rs | lw | rt | subu $6,$1,$0 | 跳转正确 |
| nop |
| nop |
| beq $1,$6,label5 |

label5:

sw $1,4($0)

lw $7,4($0)

nop

nop

beq $1,$7,end

nop

label4:

subu $6,$1,$0

nop

nop

beq $1,$6,label5

nop

end:

lui $8,8

nop

beq $4,$1,label3

nop

label1:

addu $3,$0,$1

nop

nop

beq $3,$1,label2

nop

label3:

ori $5,$1,0

nop

nop

beq $1,$5,label4

nop

测试程序：

lui $1,1

ori $2,$1,0

nop

nop

beq $2,$1,label1

nop

label2:

sw $1,0($0)

lw $4,0($0)

nop

Table jr冲突测试

label2:

addu $5,$31,$0

jr $5

nop

label3:

ori $6,$31,0

nop

jr $6

nop

label31:

subu $12,$31,$0

nop

jr $12

nop

label4:

sw $31,0($0)

lw $7,0($0)

jr $7

nop

label5:

sw $31,4($0)

lw $8,4($0)

nop

jr $8

nop

label6:

ori $9,$31,0

nop

nop

jr $9

nop

label7:

subu $10,$31,$0

nop

nop

jr $10

nop

label8:

sw $31,8($0)

lw $11,8($0)

nop

nop

jr $11

nop

end:

ori $31,$0,0

lui $1,1

lui $2,2

lui $3,3

lui $4,4

jal label1

nop

addu $2,$1,$0

jal label2

nop

addu $3,$2,$1

jal label3

nop

ori $4,$0,100

jal label31

nop

ori $18,1

jal label4

nop

lui $5,5

jal label5

nop

lui $6,6

jal label6

nop

lui $7,7

jal label7

nop

lui $8,8

jal label8

nop

lui $9,9

j end

label1:

ori $4,$31,0

jr $4

nop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试类型 | 前序指令 | 冲突寄存器 | 测试序列 | 实际结果 |
| i-E-rs | ori | rs | ori $4,$31,0 | 跳转正确 |
| jr $4 |
| r-E-rs | addu | rs | addu $5,$31,$0 | 跳转正确 |
| jr $5 |
| i-M-rs | ori | rs | ori $6,$31,0 | 跳转正确 |
| nop |
| jr $6 |
| r-M-rs | subu | rs | subu $12,$31,$0 | 跳转正确 |
| nop |
| jr $12 |
| lw-E-rs | lw | rs | lw $7,0($0) | 跳转正确 |
| jr $7 |
| lw-M-rs | lw | rs | lw $8,4($0) | 跳转正确 |
| nop |
| jr $8 |
| i-W-rs | ori | rs | ori $9,$31,0 | 跳转正确 |
| nop |
| nop |
| jr $9 |
| r-W-rs | addu | rs | subu $10,$31,$0 | 跳转正确 |
| nop |
| nop |
| jr $10 |
| lw-W-rs | lw | rs | lw $11,8($0) | 跳转正确 |
| nop |
| nop |
| jr $11 |

label8:

sw $31,8($0)

lw $11,8($0)

nop

nop

jr $11

nop

end:

ori $31,$0,0

1. 思考题

1. 在本实验中你遇到了哪些不同指令组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？请有条理的罗列出来。(**非常重要**)