流水线CPU设计说明文档

一、数据通路设计

1、数据通路具体实现

见表格文档

二、模块规格

1、PC

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 同步复位信号 |
| npc[31:0] | I | 32位存储数据 |
| pc[31:0] | O | 输出32位PC存储的数据 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当reset为高电平时PC同步复位为0x0000\_3000 |
| 2 | 写入npc | 时钟上升沿写入npc端口传入的值 |

2、NPC

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| pc[31:0] | I | 当前指令PC值 |
| NPCOp[1:0] | I | 计算NPC功能选择 |
| Imm[25:0] | I | 26位立即数 |
| ra[31:0] | I | 32位寄存器值 |
| npc[31:0] | O | 输出32位NPC的值 |
| pc4[31:0] | O | 输出32位PC+4的值 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 次地址计算 | NPCOp=00，计算输出顺序地址（PC+4）  NPCOp=01，计算输出beq地址  NPCOp=10，计算输出jal地址  NPCOp=11，计算输出jr地址 |

3、IM

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| addr[31:0] | I | 输入32位地址 |
| Instr[31:0] | O | 输出32位指令 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 取出指令 | 根据addr的值从IM中取出指令 |

4、GRF

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| A1[4:0] | I | 第1个读出寄存器的编号 |
| A2[4:0] | I | 第2个读出寄存器的编号 |
| A3[4:0] | I | 写入寄存器的编号 |
| RD1[31:0] | O | A1指向寄存器的值 |
| RD2[31:0] | O | A2指向寄存器的值 |
| WD[31:0] | I | 写入寄存器的值 |
| WE | I | 写入使能 |
| clk | I | 时钟信号 |
| reset | I | 同步复位信号 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | reset=1时在时钟上升沿所有寄存器复位为0x0000\_0000 |
| 2 | 读出寄存器 | A1和A2对应的32位寄存器值分别通过RD1和RD2输出 |
| 3 | 写入寄存器 | WE=1时，WD的值写入A3所指的寄存器 |

5、ALU

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| A[31:0] | I | 第1个32位操作数 |
| B[31:0] | I | 第2个32位操作数 |
| C[31:0] | O | 32位计算结果 |
| ALUOp[1:0] | I | ALU功能选择 |
| zero | O | A和B相等比较结果 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | ALU数学逻辑计算 | ALUOp=00，C=A+B  ALUOp=01，C=A-B  ALUOp=10，C=A|B  ALUOp=11，保留 |
| 2 | 相等比较 | zero=1，A=B  zero=0，AB |

6、DM

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| addr[31:0] | I | 地址输入 |
| WD[31:0] | I | 写入数据 |
| RD[31:0] | O | 读出数据 |
| WE | I | 写入使能 |
| clk | I | 时钟信号 |
| reset | I | 异步复位信号 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | RST=1时，在时钟上升沿RAM复位为零 |
| 2 | 读出数据 | 将addr所表示的内存地址中的值读出 |
| 3 | 写入数据 | WE=1时，将WD的值写入addr所表示的内存地址中 |

7、EXT

（1）端口说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| Imm[15:0] | I | 16位立即数输入 |
| out[31:0] | O | 32位扩展结果输出 |
| EXTOp[1:0] | I | 扩展功能信号 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 扩展立即数 | EXTOp=00，进行零扩展  EXTOp=01，进行符号扩展  EXTOp=10，进行LUI扩展  EXTOp=11，保留 |

8、CMP

（1）端口定义

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| A[31:0] | I | 32位数据输入 |
| B[31:0] | I | 32位数据输入 |
| equal | O | 相等信号输出 |

（2）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 相等比较 | A==B, equal = 1  A!=B, equal = 0 |

9、Controller

（见第二部分：控制器设计）

三、控制器设计

1、端口定义及功能说明

|  |  |  |
| --- | --- | --- |
| 端口名 | 方向 | 描述 |
| opcode[5:0] | I | Instr[31:26] |
| funct[5:0] | I | Instr[5:0] |
| equal | I | 相等比较信号 |
| DMWr | O | DM写使能信号 |
| ALUOp[1:0] | O | ALU运算信号 |
| BSel | O | ALU的B端MUX选择信号：  0：寄存器RD2值  1：EXT扩展数据 |
| EXTOp[1:0] | O | EXT扩展信号 |
| RFWr | O | GRF写使能信号 |
| WDSel[1:0] | O | 寄存器的WD端MUX选择信号：  00：ALU运算结果  01：DM输出数据  10：PC + 4  11：保留 |
| WRSel[1:0] | O | 寄存器的A3端MUX选择信号：  00：Instr[20:16]  01：Instr[15:11]  10：0x1f  11：保留 |
| NPCOp[1:0] | O | IFU中NPC运算信号 |

2、真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr | addu | subu | ori | lw | sw | beq | lui | j | jal | jr |
| Op | 000000 | 000000 | 001101 | 100011 | 101011 | 000100 | 001111 | 000010 | 000011 | 000000 |
| Func | 100001 | 100011 | NA | NA | NA | NA | NA | NA | NA | 001000 |
| DMWr | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| ALUOp | 00 | 01 | 10 | 00 | 00 | XX | X0 | XX | XX | XX |
| BSel | 0 | 0 | 1 | 1 | 1 | X | 1 | X | X | X |
| EXTOp | XX | XX | 00 | 01 | 01 | XX | 10 | XX | XX | XX |
| RFWr | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| WDSel | 00 | 00 | 00 | 01 | XX | XX | 00 | XX | 10 | XX |
| WRSel | 01 | 01 | 00 | 00 | XX | XX | 00 | XX | 10 | XX |
| NPCOp | 00 | 00 | 00 | 00 | 00 | 0 Zero | 00 | 10 | 10 | 11 |

3、暂停控制设计表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | ID/EX级 | | | | EX/MEM级 | | | |
| 指令类型 | Calc\_R | Calc\_I | Load | JAL | Calc\_R | Calc\_I | Load | JAL |
| 目标寄存器 | rd | rt | rt | $31 | rd | rt | rt | $31 |
| Tnew | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 |
| 指令类型 | 源寄存器 | Tuse |  |  |  |  |  |  |  |  |  |
| Calc\_R | rs/rt | 1 |  |  |  | Stall |  |  |  |  |  |
| Calc\_I | rs | 1 |  |  |  | Stall |  |  |  |  |  |
| Load | rs | 1 |  |  |  | Stall |  |  |  |  |  |
| Beq | rs/rt | 0 |  | Stall | Stall | Stall |  |  |  | Stall |  |
| Store | rs | 1 |  |  |  | Stall |  |  |  |  |  |
| rt | 2 |  |  |  |  |  |  |  |  |  |
| JR | rs | 0 |  | Stall | Stall | Stall |  |  |  | Stall |  |

4、转发控制设计表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 流水级 | | 源寄存器 | | 涉及指令 | | | | 转发MUX | | | | | | | 控制信号 | | | | 输入0 | |
| IF/ID | | rs | | beq, jr | | | | MUX\_ForwardRS\_ID | | | | | | | ForwardRS\_ID | | | | GRF.RD1 | |
| rt | | beq | | | | MUX\_ForwardRT\_ID | | | | | | | ForwardRT\_ID | | | | GRF.RD2 | |
| ID/EX | | rs | | calc\_r, calc\_i,load,store | | | | MUX\_ForwardRS\_EX | | | | | | | ForwardRS\_EX | | | | RegData1\_ID\_EX | |
| rt | | calc\_r | | | | MUX\_ForwardRT\_EX | | | | | | | ForwardRT\_EX | | | | RegData2\_ID\_EX | |
| EX/MEM | | rt | | store | | | | MUX\_ForwardRT\_MEM | | | | | | | ForwardRT\_MEM | | | | DM\_WD\_EX\_MEM | |
|  | | |  | | ID/EX级 | | | | | | | | | | |
| 指令类型 | | Calc\_R | | | | Calc\_I | Load | | | JAL | | |
| 目标寄存器 | | rd | | | | rt | rt | | | $31 | | |
| Tnew | | 1 | | | | 1 | 2 | | | 0 | | |
| 流水级 | 源寄存器 | |  | |  | | | |  |  | | |  | | |
| IF/ID | rs | |  | | / | | | | / | / | | | PC8\_EX | | |
| rt | |  | | / | | | | / | / | | | PC8\_EX | | |
| ID/EX | rs | |  | | / | | | | / | / | | | / | | |
| rt | |  | | / | | | | / | / | | | / | | |
| EX/MEM | rt | |  | | / | | | | / | / | | | / | | |
|  | | | EX/MEM级 | | | | | | | | | | | | | |
| Calc\_R | | | Calc\_I | | | | | Load | | | JAL | | |
| rd | | | rt | | | | | rt | | | $31 | | |
| 0 | | | 0 | | | | | 1 | | | 0 | | |
| 流水级 | 源寄存器 | |  | | |  | | | | |  | | |  | | |
| IF/ID | rs | | ALUout\_MEM | | | ALUout\_MEM | | | | | / | | | PC8\_MEM | | |
| rt | | ALUout\_MEM | | | ALUout\_MEM | | | | | / | | | PC8\_MEM | | |
| ID/EX | rs | | ALUout\_MEM | | | ALUout\_MEM | | | | | / | | | PC8\_MEM | | |
| rt | | ALUout\_MEM | | | ALUout\_MEM | | | | | / | | | PC8\_MEM | | |
| EX/MEM | rt | | / | | | / | | | | | / | | | / | | |
|  | | | MEM/WB级 | | | | | | | | | | | | | | | | |
| Calc\_R | | | | Calc\_I | | | | | Load | | | | | | JAL | |
| rd | | | | rt | | | | | rt | | | | | | $31 | |
| 0 | | | | 0 | | | | | 0 | | | | | | 0 | |
| 流水级 | 源寄存器 | |  | | | |  | | | | |  | | | | | |  | |
| IF/ID | rs | | / | | | | / | | | | | / | | | | | | / | |
| rt | | / | | | | / | | | | | / | | | | | | / | |
| ID/EX | rs | | RegWriteData\_WB | | | | RegWriteData\_WB | | | | | RegWriteData\_WB | | | | | | RegWriteData\_WB | |
| rt | | RegWriteData\_WB | | | | RegWriteData\_WB | | | | | RegWriteData\_WB | | | | | | RegWriteData\_WB | |
| EX/MEM | rt | | RegWriteData\_WB | | | | RegWriteData\_WB | | | | | RegWriteData\_WB | | | | | | RegWriteData\_WB | |

四、测试模块

1、测试A

测试程序：

ori $24, $0, 12

ori $24, $0, 16

ori $24, $0, 18

sw $24, 0($0)

addu $24, $24, $24

addu $24, $24, $24

addu $24, $24, $24

sw $24, 4($0)

jal go

go:

lui $24, 0x8000

sw $24, 8($0)

lw $24, 8($0)

lw $24, 4($0)

lw $24, 0($0)

sw $24, 12($0)

beq $24, $24, to

to:

sw $24, 16($0)

ori $ra, $0, 0x304c

jr $ra

sw $24, 20($0)

期望结果：

//可能引起GRF内部转发时产生的错误转发（写使能为0时转发）

55@00003000: $24 <= 0000000c

65@00003004: $24 <= 00000010

75@00003008: $24 <= 00000012

75@0000300c: \*00000000 <= 00000012

95@00003010: $24 <= 00000024

105@00003014: $24 <= 00000048

115@00003018: $24 <= 00000090

115@0000301c: \*00000004 <= 00000090

135@00003020: $31 <= 00003028

145@00003024: $24 <= 80000000

155@00003024: $24 <= 80000000

155@00003028: \*00000008 <= 80000000

175@0000302c: $24 <= 80000000

185@00003030: $24 <= 00000090

195@00003034: $24 <= 00000012

195@00003038: \*0000000c <= 00000012

225@00003040: \*00000010 <= 00000012

235@00003040: \*00000010 <= 00000012

255@00003044: $31 <= 0000304c

275@0000304c: \*00000014 <= 00000012

285@0000304c: \*00000014 <= 00000012

2、测试B

测试程序：

ori $1, $0, 1

addu $0, $1, $1

addu $2, $0, $0

addu $3, $0, $0

addu $4, $0, $0

ori $1, $0, 4

addu $0, $1, $1

sw $0, 0($0)

sw $0, 0($0)

sw $0, 0($0)

期望结果：

//因为未判断rs、rt为零导致错误转发

$1 <= 1

$2 <= 0

$3 <= 0

$4 <= 0

$1 <= 4

\*0x00000000 <= 0

\*0x00000000 <= 0

\*0x00000000 <= 0

3、测试C (Calc\_R)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-M-RS | subu $2, $3, $4  addu $1, $2, $4 | $3 = 5  $4 = 2 | $2 <= 3  $1 <= 1 |
| 2 | R-M-RT | subu $2, $3, $4  addu $1, $3, $2 | $3 = 5  $4 = 2 | $2 <= 3  $1 <= 2 |
| 3 | R-W-RS | subu $2, $3, $4  nop  addu $1, $2, $4 | $3 = 5  $4 = 2 | $2 <= 3  $1 <= 1 |
| 4 | R-W-RT | subu $2, $3, $4  nop  addu $1, $3, $2 | $3 = 5  $4 = 2 | $2 <= 3  $1 <= 2 |
| 5 | I-M-RS | ori $2, $3, 8  addu $1, $2, $3 | $3 = 3 | $2 <= 11  $1 <= 14 |
| 6 | I-M-RT | ori $2, $3, 8  addu $1, $3, $2 | $3 = 3 | $2 <= 11  $1 <= 14 |
| 7 | I-W-RS | ori $2, $3, 8  nop  addu $1, $2, $3 | $3 = 3 | $2 <= 11  $1 <= 14 |
| 8 | I-W-RT | ori $2, $3, 8  nop  addu $1, $3, $2 | $3 = 3 | $2 <= 11  $1 <= 14 |
| 9 | LW-M-RS | lw $2, 0($3)  addu $1, $2, $3 | $3 = 4  \*4 = 10 | $2 <= 10  stall  $1 <= 14 |
| 10 | LW-M-RT | lw $2, 0($3)  addu $1, $3, $2 | $3 = 4  \*4 = 10 | $2 <= 10  stall  $1 <= 14 |
| 11 | LW-W-RS | lw $2, 0($3)  nop  addu $1, $2, $3 | $3 = 4  \*4 = 10 | $2 <= 10  $1 <= 14 |
| 12 | LW-W-RT | lw $2, 0($3)  nop  addu $1, $3, $2 | $3 = 4  \*4 = 10 | $2 <= 10  $1 <= 14 |
| 13 | JAL-M-RS | jal go  addu $1, $31, $0  go:… |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 |
| 14 | JAL-M-RT | jal go  addu $1, $0, $31  go:… |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 |
| 15 | JAL-W-RS | jal go  nop  go:  addu $1, $31, $0 |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 |
| 16 | JAL-W-RT | jal go  nop  go:  addu $1, $0, $31 |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 |

4、测试D(Calc\_I)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-M-RS | subu $2, $3, $4  ori $1, $2, 1 | $3 = 8  $4 = 2 | $2 <= 6  $1 <= 7 |
| 2 | R-W-RS | subu $2, $3, $4  nop  ori $1, $2, 1 | $3 = 8  $4 = 2 | $2 <= 6  $1 <= 7 |
| 3 | I-M-RS | ori $2, $3, 8  ori $1, $2, 1 | $3 = 2 | $2 <= 10  $1 <= 11 |
| 4 | I-W-RS | ori $2, $3, 8  nop  ori $1, $2, 1 | $3 = 2 | $2 <= 10  $1 <= 11 |
| 5 | LW-M-RS | lw $2, 0($3)  ori $1, $2, 1 | $3 = 4  \*4 = 10 | $2 <= 10  stall  $1 <= 11 |
| 6 | LW-W-RS | lw $2, 0($3)  nop  ori $1, $2, 1 | $3 = 4  \*4 = 10 | $2 <= 10  $1 <= 11 |
| 7 | JAL-M-RS | jal go  ori $1, $31, 1  go:… |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 + 1 |
| 8 | JAL-W-RS | jal go  nop  go:  ori $1, $31, 1 |  | $31 <= PC(jal) + 8  $1 <= PC(jal) + 8 + 1 |

4、测试E (Load)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-M-RS | subu $2, $3, $4  lw $1, 0($2) | $3 = 8  $4 = 4  \*4 = 127 | $2 <= 4  $1 <= 127 |
| 1 | R-M-RS | subu $2, $3, $4  nop  lw $1, 0($2) | $3 = 8  $4 = 4  \*4 = 127 | $2 <= 4  $1 <= 127 |
| 3 | I-M-RS | ori $2, $3, 8  lw $1, 0($2) | $3 = 0  \*8 = 127 | $2 <= 8  $1 <= 127 |
| 4 | I-W-RS | ori $2, $3, 8  nop  lw $1, 0($2) | $3 = 0  \*8 = 127 | $2 <= 8  $1 <= 127 |
| 5 | LW-M-RS | lw $2, 0($3)  lw $1, 0($2) | $3 = 4  \*4 = 10 | $2 <= 10  stall  $1 <= 11 |
| 6 | LW-W-RS | lw $2, 0($3)  nop  lw $1, 0($2) | $3 = 4  \*4 = 10 | $2 <= 10  $1 <= 11 |
| 7 | JAL-M-RS | jal go  lw $1, 0($31)  go:… | \*[ PC(jal) + 8] = 333 | $31 <= PC(jal) + 8  $1 <= 333 |
| 8 | JAL-W-RS | jal go  nop  go:  lw $1, 0($2) | \*[ PC(jal) + 8] = 333 | $31 <= PC(jal) + 8  $1 <= 333 |

5、测试F (Store)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-M-RS | subu $2, $3, $4  sw $4, -4($2) | $3 = 10  $4 = 2 | $2 <= 8  \*4 <= 2 |
| 2 | R-M-RT | subu $2, $3, $4  sw $2, -4($4) | $3 = 6  $4 = 4 | $2 <= 2  \*0 <= 2 |
| 3 | R-W-RS | subu $2, $3, $4  nop  sw $4, -4($2) | $3 = 10  $4 = 2 | $2 <= 8  \*4 <= 2 |
| 4 | R-W-RT | subu $2, $3, $4  nop  sw $2, -4($4) | $3 = 6  $4 = 4 | $2 <= 2  \*0 <= 2 |
| 5 | I-M-RS | ori $2, $3, 8  sw $4, -4($2) | $3 = 4  $4 = 11 | $2 <= 12  \*8 <= 11 |
| 6 | I-M-RT | ori $2, $3, 8  sw $2, -4($4) | $3 = 3  $4 = 24 | $2 <= 11  \*20 <= 11 |
| 7 | I-W-RS | ori $2, $3, 8  nop  sw $4, -4($2) | $3 = 4  $4 = 11 | $2 <= 12  \*8 <= 11 |
| 8 | I-W-RT | ori $2, $3, 8  nop  sw $2, -4($4) | $3 = 3  $4 = 24 | $2 <= 11  \*20 <= 11 |
| 9 | LW-M-RS | lw $2, 0($3)  sw $4, -4($2) | $3 = 4  \*4 = 12  $4 = 11 | $2 <= 12  stall  \*8 <= 11 |
| 10 | LW-M-RT | lw $2, 0($3)  sw $2, -4($4) | $3 = 4  \*4 = 10  $4 = 8 | $2 <= 10  \*4 <= 10 |
| 11 | LW-W-RS | lw $2, 0($3)  nop  sw $4, -4($2) | $3 = 4  \*4 = 12  $4 = 11 | $2 <= 12  \*8 <= 11 |
| 12 | LW-W-RT | lw $2, 0($3)  nop  sw $2, -4($4) | $3 = 4  \*4 = 10  $4 = 8 | $2 <= 10  \*4 <= 10 |
| 13 | JAL-M-RS | jal go  sw $1, 0($31)  go:… | $1 = 0 | $31 <= PC(jal) + 8  \*[PC(jal) + 8] <= 0  (可写入.text) |
| 14 | JAL-M-RT | jal go  sw $31, 0($0)  go:… |  | $31 <= PC(jal) + 8  \*0 <= PC(jal) + 8 |
| 15 | JAL-W-RS | jal go  nop  go:  sw $1, 0($31) | $1 = 0 | $31 <= PC(jal) + 8  \*[PC(jal) + 8] <= 0  (可写入.text) |
| 16 | JAL-W-RT | jal go  nop  go:  sw $31, 0($0) |  | $31 <= PC(jal) + 8  \*0 <= PC(jal) + 8 |

6、测试G (Beq)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-E-RS | subu $2, $3, $4  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 10  $4 = 2  $5 = 8 | $2 <= 8  stall  jump to go  $1 <= 0x11110000 |
| 2 | R-E-RT | subu $2, $3, $4  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 10  $4 = 2  $5 = 8 | $2 <= 8  stall  jump to go  $1 <= 0x11110000 |
| 3 | R-M-RS | subu $2, $3, $4  nop  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 10  $4 = 2  $5 = 8 | $2 <= 8  jump to go  $1 <= 0x11110000 |
| 4 | R-M-RT | subu $2, $3, $4  nop  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 10  $4 = 2  $5 = 8 | $2 <= 8  jump to go  $1 <= 0x11110000 |
| 5 | I-E-RS | ori $2, $0, 8  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $5 = 8 | $2 <= 8  stall  jump to go  $1 <= 0x11110000 |
| 6 | I-E-RT | ori $2, $0, 8  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $5 = 8 | $2 <= 8  stall  jump to go  $1 <= 0x11110000 |
| 7 | I-M-RS | ori $2, $0, 8  nop  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $5 = 8 | $2 <= 8  jump to go  $1 <= 0x11110000 |
| 8 | I-M-RT | ori $2, $0, 8  nop  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $5 = 8 | $2 <= 8  jump to go  $1 <= 0x11110000 |
| 9 | LW-E-RS | lw $2, 0($3)  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  stall  stall  jump to go  $1 <= 0x11110000 |
| 10 | LW-E-RT | lw $2, 0($3)  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  stall  stall  jump to go  $1 <= 0x11110000 |
| 11 | LW-M-RS | lw $2, 0($3)  nop  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  stall  jump to go  $1 <= 0x11110000 |
| 12 | LW-M-RT | lw $2, 0($3)  nop  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  stall  jump to go  $1 <= 0x11110000 |
| 13 | LW-W-RS | lw $2, 0($3)  nop  nop  beq $2, $5, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  jump to go  $1 <= 0x11110000 |
| 14 | LW-W-RT | lw $2, 0($3)  nop  nop  beq $5, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $3 = 4  \*4 = 12  $5 = 12 | $2 <= 12  jump to go  $1 <= 0x11110000 |
| 15 | JAL-E-RS | 未定义行为 |  |  |
| 16 | JAL-E-RT | 未定义行为 |  |  |
| 17 | JAL-M-RS | jal pass  nop  pass:  beq $ra, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $2 = 2 | $2 <= 12  stall  NOT jump to go  $1 <= 0xffff0000  $1 <= 0x11110000 |
| 18 | JAL-M-RT | jal pass  nop  pass:  beq $2, $ra, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $2 = 2 | $2 <= 12  stall  NOT jump to go  $1 <= 0xffff0000  $1 <= 0x11110000 |
| 19 | JAL-W-RS | jal pass  nop  pass:  nop  beq $ra, $2, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $2 = 2 | $2 <= 12  stall  NOT jump to go  $1 <= 0xffff0000  $1 <= 0x11110000 |
| 20 | JAL-W-RT | jal pass  nop  pass:  nop  beq $2, $ra, go  nop  lui $1, 0xffff  go:  lui $1, 0x1111 | $2 = 2 | $2 <= 12  stall  NOT jump to go  $1 <= 0xffff0000  $1 <= 0x11110000 |

7、测试H (Jr)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试类型 | 测试序列 | 预先赋值 | 期望结果 |
| 1 | R-E-RS | addu $2, $3, $4  jr $2 | $3 = 0x3000  $4 = 8 | $2 <= 0x000030008  stall  jump to 0x000030008 |
| 2 | R-M-RS | addu $2, $3, $4  nop  jr $2 | $3 = 0x3000  $4 = 8 | $2 <= 0x000030008  jump to 0x000030008 |
| 3 | R-W-RS | addu $2, $3, $4  nop  nop  jr $2 | $3 = 0x3000  $4 = 8 | $2 <= 0x000030008  jump to 0x000030008 |
| 4 | I-E-RS | ori $2, $0, 0x3008  jr $2 |  | $2 <= 0x000030008  stall  jump to 0x000030008 |
| 5 | I-M-RS | ori $2, $0, 0x3008  nop  jr $2 |  | $2 <= 0x000030008  jump to 0x000030008 |
| 6 | I-W-RS | ori $2, $0, 0x3008  nop  nop  jr $2 |  | $2 <= 0x000030008  jump to 0x000030008 |
| 7 | LW-E-RS | lw $2, 0($0)  jr $2 | \*0 = 0x3008 | $2 <= 0x000030008  stall  stall  jump to 0x000030008 |
| 8 | LW=M-RS | lw $2, 0($0)  nop  jr $2 | \*0 = 0x3008 | $2 <= 0x000030008  stall  jump to 0x000030008 |
| 9 | LW-W-RS | lw $2, 0($0)  nop  nop  jr $2 | \*0 = 0x3008 | $2 <= 0x000030008  jump to 0x000030008 |
| 10 | JAL-E-RS | 未定义行为 |  |  |
| 11 | JAL-M-RS | jal go  nop  go:  jr $ra |  | $ra <= PC(jal)+8  forever jump |
| 12 | JAL-W-RS | jal go  nop  go:  nop  jr $ra |  | $ra <= PC(jal)+8  forever jump |

五、思考题

1、在本实验中你遇到了哪些不同指令类型组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？

全部都在上面啦。

为手动构造特殊的指令进行冲突测试，根据暂停和转发的表格，对指令顺序进行有计划的排列，如上所写测试。