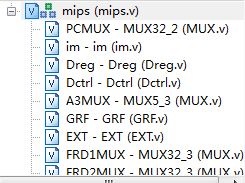
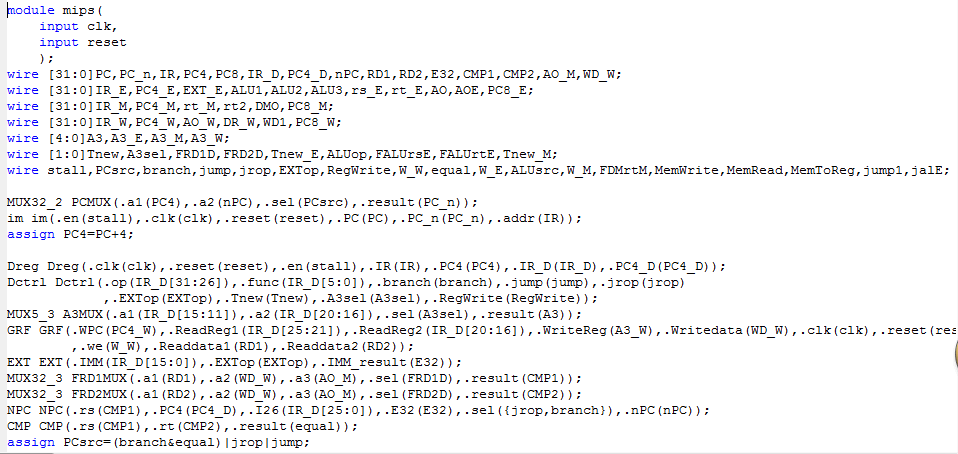
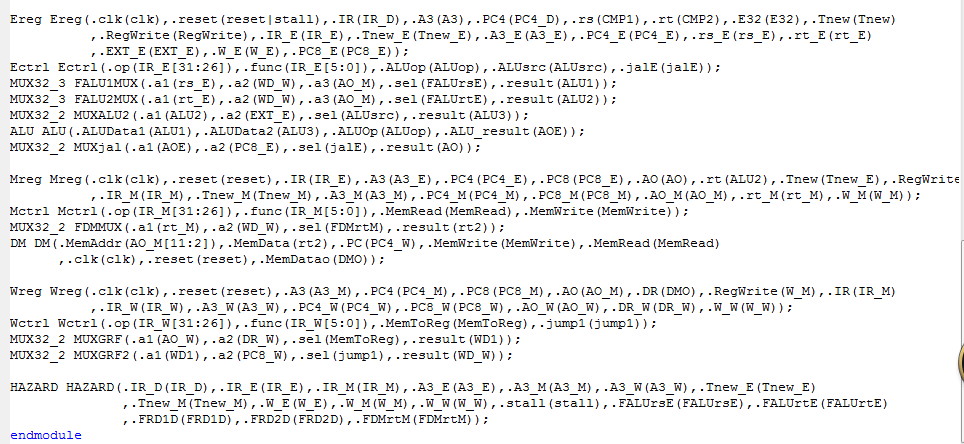
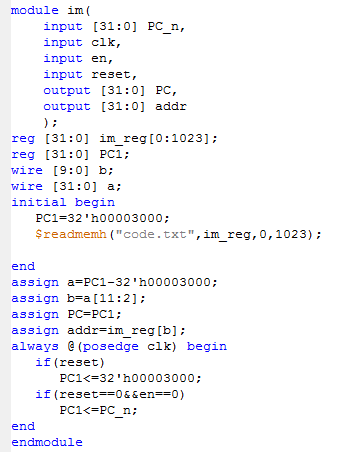
CPU设计文档

一、总体设计

  二、数据通路设计

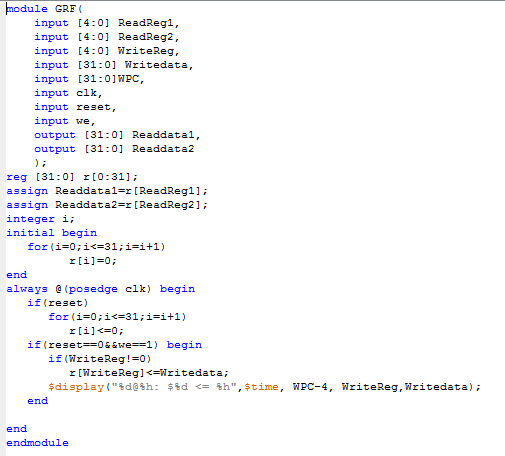
1、im



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| reset | I | 复位信号,当复位信号为1时，PC被设置为0x00003000 |
| [31:0]PC\_n | I | 读入下一周期PC的值 |
| clk | I | 时钟信号 |
| en | I | PC使能端 |
| [31:0]PC | O | 当前PC的值 |
| [31:0]addr | O | 当前32位指令值 |

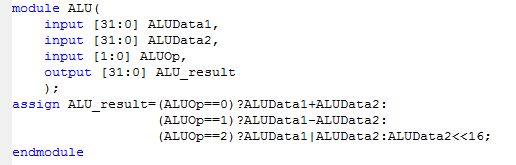
2、GRF



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| reset | I | 复位信号，当复位信号为1时，所有寄存器被设置为0x00000000 |
| we | I | 寄存器使能信号，当使能信号为1时，寄存器可以被写入数据 |
| clk | I | 时钟信号 |
| [4:0]ReadReg1 | I | 寄存器1的地址 |
| [4:0]ReadReg2 | I | 寄存器2的地址 |
| [4:0]WriteReg | I | 需要写入数据的寄存器的地址 |
| [31:0]Wirtedata | I | 写入的数据 |
| [31:0]WPC | I | 当前PC的值 |
| [31:0]Readdata1 | O | 寄存器1中的数据 |
| [31:0]Readdata2 | O | 寄存器2中的数据 |

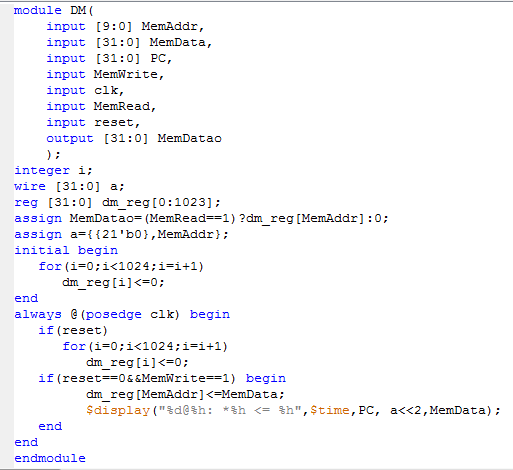
3、ALU



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [31:0]ALUData1 | I | ALU数据1 |
| [31:0]ALUData2 | I | ALU数据2 |
| [1:0]ALUOp | I | ALU控制信号  00:加法运算 01:减法运算（Data1-Data2）  10:或运算 11:左移16位 |
| [31:0]ALU\_result | O | 输出ALU计算结果 |

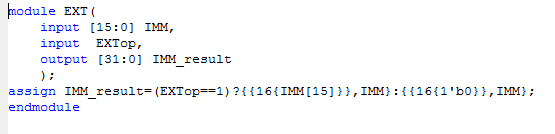
4、DM



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| reset | I | 复位信号,当复位信号为1时，所有寄存器被设置为0x00000000 |
| clk | I | 时钟信号 |
| MemWrite | I | 内存寄存器写使能信号，当该信号为1时，可将数据写入内存寄存器中 |
| MemRead | I | 内存寄存器读使能信号，当该信号为1时，可输出内存寄存器中的数据 |
| [9:0]MemAddr | I | 内存寄存器的地址值 |
| [31:0]MemData | I | 需要写入内存寄存器的数据 |
| [31:0]PC | I | 当前PC值 |
| [31:0]MemData\_o | O | 输出内存寄存器中的数据 |

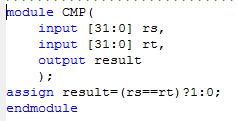
5、EXT



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [15:0]IMM | I | 16为待扩展数据 |
| EXTop | I | 扩展方式控制信号  0:无符号扩展 1:有符号扩展 |
| [31:0]IMM\_result | O | 扩展后数据 |

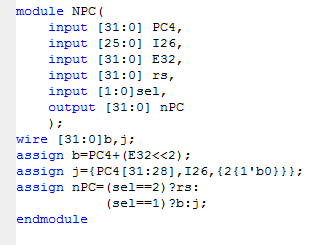
6、CMP



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [31:0]rs | I | 寄存器rs的值 |
| [31:0]rt | I | 寄存器rt的值 |
| result | O | 若rs==rt，则输出1 |

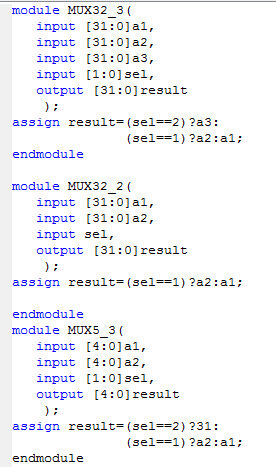
7、NPC



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [31:0]PC4 | I | PC+4的值 |
| [25:0]I26 | I | J型指令后26位 |
| [31:0]E32 | I | B型指令扩展后的32位数 |
| [31:0]rs | I | rs寄存器中的值 |
| [1:0]sel | I | 选择信号  2：选择rs寄存器中的值 1：b型指令 0：j型指令 |
| [31:0]nPC | O | 下一周期PC的值 |

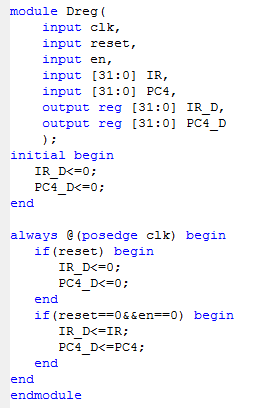
8、MUX



模块接口

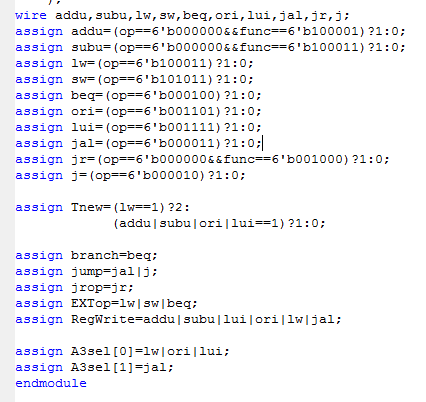
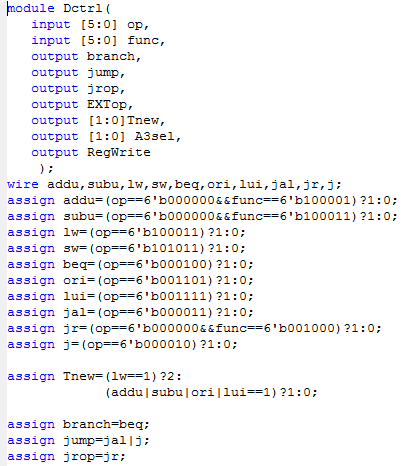
|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [31:0]a1 | I | 多选器0接口 |
| [31:0]a2 | I | 多选器1接口 |
| [31:0]a3 | I | 多选器2接口 |
| [1:0]sel | I | 选择信号 |
| [31:0]result | O | 选择结果 |
| [31:0]a1 | I | 多选器0接口 |
| [31:0]a2 | I | 多选器1接口 |
| sel | I | 选择信号 |
| [31:0]result | O | 选择结果 |
| [4:0]a1 | I | 多选器0接口 |
| [4:0]a2 | I | 多选器1接口 |
| [1:0]sel | I | 选择信号（多选器2接口为$31） |
| [4:0]result | O | 选择结果 |

9、Dreg



模块接口

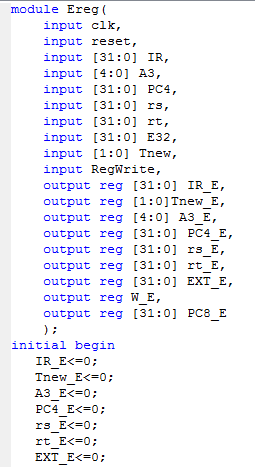
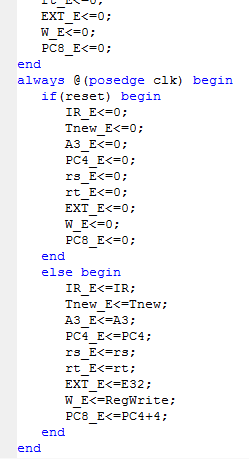
|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时钟信号 |
| en | I | D级流水线寄存器使能信号 |
| reset | I | D级流水线寄存器复位信号 |
| [31:0]IR | I | 传入的32位指令 |
| [31:0]PC4 | I | 传入的PC+4的值 |
| [31:0]IR\_D | O | 输出D级寄存器的指令 |
| [31:0]PC4\_D | O | 输出D级寄存器的PC+4 |

10、Dctrl

模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [5:0]op | I | 指令的[31:26]位 |
| [5:0]func | I | 指令的[5:0]位 |
| branch | O | B型指令控制信号 |
| jump | O | J型指令控制信号 |
| Jrop | O | jr指令控制信号 |
| EXTop | O | 扩展控制信号 |
| [1:0]Tnew | O | 当前指令产生结果所需要的周期数 |
| [1:0]A3sel | O | 写回寄存器的选择信号 |
| RegWrite | O | 写寄存器信号 |

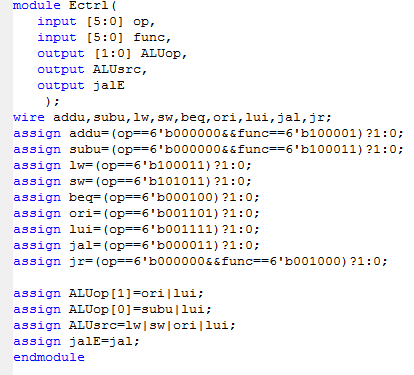
11、Ereg



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时钟信号 |
| reset | I | E级寄存器复位信号 |
| [31:0]IR | I | 传入的指令 |
| [4:0]A3 | I | 要写入的寄存器的地址值 |
| [31:0]PC4 | I | 传入的PC+4的值 |
| [31:0]rs | I | 传入的rs寄存器的值 |
| [31:0]rt | I | 传入的rt寄存器的值 |
| [31:0]E32 | I | 传入的扩展后的立即数值 |
| [1:0]Tnew | I | 传入的Tnew值 |
| RegWrite | I | 写寄存器信号 |
| [31:0]IR\_E | O | 输出指令值 |
| [1:0]Tnew\_E | O | 输出Tnew值 |
| [4:0]A3\_E | O | 输出写回寄存器地址值 |
| [31:0]PC4\_E | O | 输出PC+4的值 |
| [31:0]rs\_E | O | 输出rs寄存器的值 |
| [31:0]rt\_E | O | 输出rt寄存器的值 |
| [31:0]EXT\_E | O | 输出扩展后立即数的值 |
| W\_E | O | 输出寄存器写使能信号 |
| [31:0]PC8\_E | O | 输出PC+8的值 |

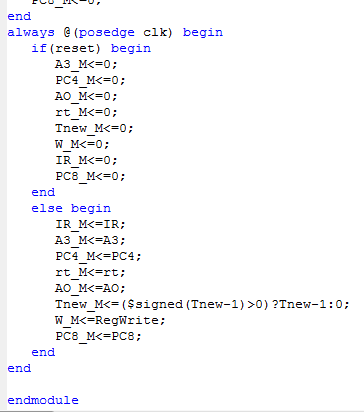
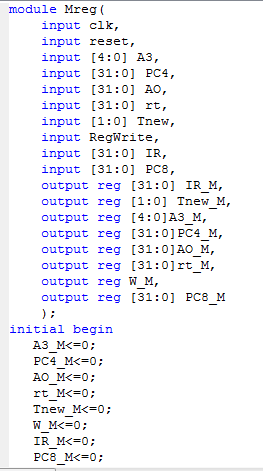
12、Ectrl



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [5:0]op | I | 指令[31:26]位 |
| [5:0]func | I | 指令[5:0]位 |
| [1:0]ALUop | O | ALU控制信号 |
| ALUsrc | O | ALU接口2数据选择信号 |
| jalE | O | jal指令控制信号 |

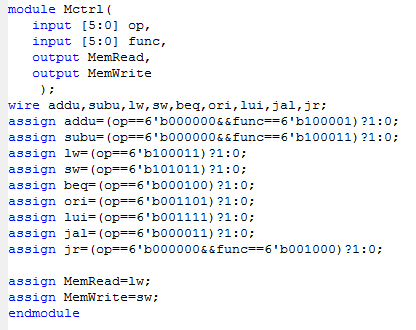
13、Mreg



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时钟信号 |
| reset | I | M级寄存器复位信号 |
| [31:0]IR | I | 传入的指令 |
| [4:0]A3 | I | 要写入的寄存器的地址值 |
| [31:0]PC4 | I | 传入的PC+4的值 |
| [31:0]AO | I | 传入的ALU结果 |
| [31:0]rt | I | 传入的rt寄存器的值 |
| [31:0]PC8 | I | 传入的PC+8的值 |
| [1:0]Tnew | I | 传入的Tnew值 |
| RegWrite | I | 写寄存器信号 |
| [31:0]IR\_M | O | 输出指令值 |
| [1:0]Tnew\_M | O | 输出Tnew值 |
| [4:0]A3\_M | O | 输出写回寄存器地址值 |
| [31:0]PC4\_M | O | 输出PC+4的值 |
| [31:0]AO\_M | O | 输出ALU结果 |
| [31:0]rt\_M | O | 输出rt寄存器的值 |
| W\_M | O | 输出寄存器写使能信号 |
| [31:0]PC8\_M | O | 输出PC+8的值 |

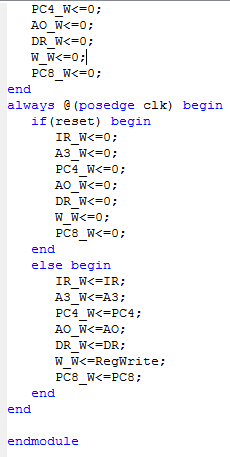
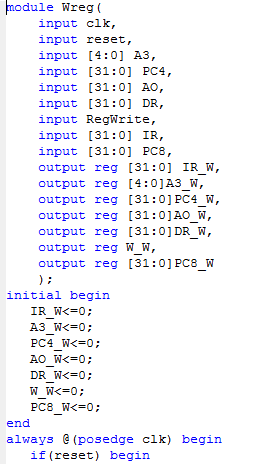
14、Mctrl



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [5:0]op | I | 指令[31:26]位 |
| [5:0]func | I | 指令[5:0]位 |
| MemRead | O | 寄存器读使能信号 |
| MemWrite | O | 寄存器写使能信号 |

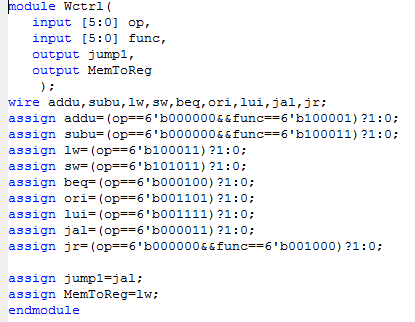
15、Wreg



模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时钟信号 |
| reset | I | W级寄存器复位信号 |
| [31:0]IR | I | 传入的指令 |
| [4:0]A3 | I | 要写入的寄存器的地址值 |
| [31:0]PC4 | I | 传入的PC+4的值 |
| [31:0]AO | I | 传入的ALU结果 |
| [31:0]DR | I | 传入的DM输出的值 |
| [31:0]PC8 | I | 传入的PC+8的值 |
| RegWrite | I | 写寄存器信号 |
| [31:0]IR\_W | O | 输出指令值 |
| [4:0]A3\_W | O | 输出写回寄存器地址值 |
| [31:0]PC4\_W | O | 输出PC+4的值 |
| [31:0]AO\_W | O | 输出ALU结果 |
| [31:0]DR\_W | O | 输出传入的DM的值 |
| W\_M | O | 输出寄存器写使能信号 |
| [31:0]PC8\_M | O | 输出PC+8的值 |

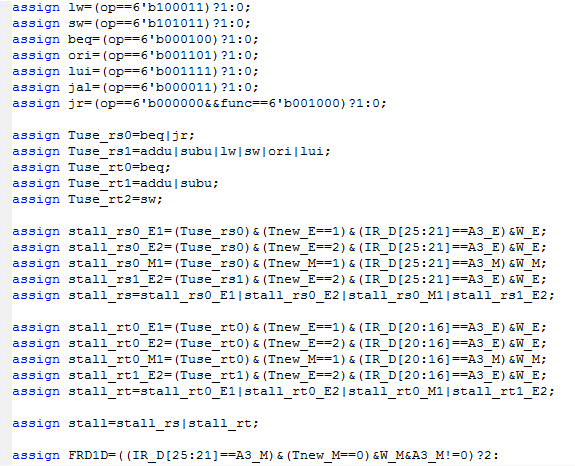
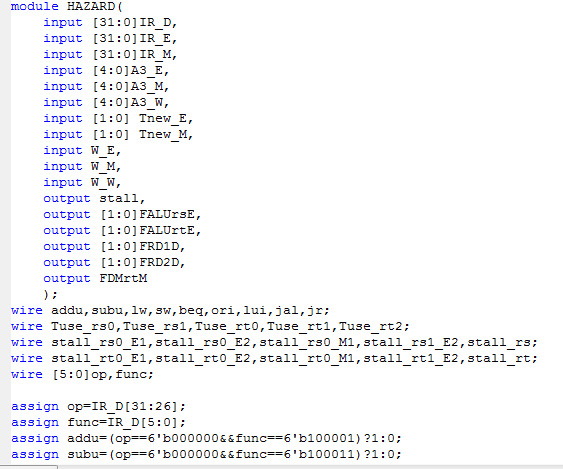
15、Wctrl

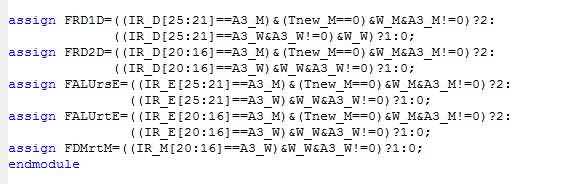


模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [5:0]op | I | 指令[31:26]位 |
| [5:0]func | I | 指令[5:0]位 |
| jump1 | O | Jal跳转控制信号 |
| MemToReg | O | 寄存器写回数据控制信号 |

16、HAZARD





模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| [31:0]IR\_D | I | D级寄存器的指令 |
| [31:0]IR\_E | I | E级寄存器的指令 |
| [31:0]IR\_M | I | M级寄存器的指令 |
| [4:0]A3\_E | I | E级要写入的寄存器地址 |
| [4:0]A3\_M | I | M级要写入的寄存器地址 |
| [4:0]A3\_W | I | W级要写入的寄存器地址 |
| [1:0]Tnew\_E | I | E级Tnew的值 |
| [1:0]Tnew\_M | I | M级Tnew的值 |
| W\_E | I | E级寄存器写使能信号 |
| W\_M | I | M级寄存器写使能信号 |
| W\_W | I | W级寄存器写使能信号 |
| stall | O | 暂停信号 |
| [1:0]FALUrsE | O | ALUrs寄存器转发控制信号 |
| [1:0]FALUrtE | O | ALUrt寄存器转发控制信号 |
| [1:0]FRD1D | O | 寄存器堆rd1转发控制信号 |
| [1:0]FRD2D | O | 寄存器堆rd2转发控制信号 |
| FDMrtM | O | DM写入数据转发控制信号 |

三、控制器及冲突设计

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| func[5:0] | 100001 | 100011 | N/A | | | | | | 001000 | 000000 | N/A |
| op[5:0] | 000000 | 000000 | 100011 | 101011 | 000100 | 001101 | 001111 | 000011 | 000000 | 000000 | 000010 |
|  | addu | subu | lw | sw | beq | ori | lui | jal | jr | nop | j |
| RegDst | 1 | 1 | 0 | X | X | 0 | 0 | X | X | X | X |
| RegWrite | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | X | 0 |
| ALUOp[1:0] | 00 | 01 | 00 | 00 | 01 | 10 | 11 | XX | XX | XX | XX |
| ALUsrc | 0 | 0 | 1 | 1 | 0 | 1 | 1 | X | X | X | X |
| MemToReg | 0 | 0 | 1 | X | X | 0 | 0 | X | X | X | X |
| MemRead | X | X | 1 | X | X | X | X | X | X | X | X |
| MemWrite | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | X | 0 |
| Branch | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | X | 0 |
| EXTop | X | X | 1 | 1 | 1 | 0 | 0 | X | X | X | X |
| jal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X | 0 |
| jrop | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 0 |
| jump | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | addu | subu | lw | sw | beq | ori | lui | jal | jr | nop | j |
| Tuse\_rs0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | X | 1 | X | X |
| Tuse\_rs1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | X | 0 | X | X |
| Tuse\_rt0 | 0 | 0 | X | 0 | 1 | X | X | X | X | X | X |
| Tuse\_rt1 | 1 | 1 | X | 0 | 0 | X | X | X | X | X | X |
| Tuse\_rt2 | 0 | 0 | X | 1 | 0 | X | X | X | X | X | X |
| [1:0]Tnew | 01 | 01 | 10 | 00 | 00 | 01 | 01 | 00 | 00 | 00 | 00 |

四、测试程序

ori $0,$0,0

ori $1,$0,1

ori $2,$0,2

ori $3,$0,3

ori $4,$0,4

ori $5,$0,5

ori $6,$0,6

ori $7,$0,7

ori $8,$0,8

ori $9,$0,9

lui $10,10

ori $11,$0,11

ori $12,$0,12

ori $13,$0,13

ori $14,$0,14

ori $15,$0,15

ori $16,$0,16

ori $17,$0,17

ori $18,$0,18

ori $19,$0,19

ori $20,$0,20

ori $21,$0,21

ori $22,$0,22

ori $23,$0,23

ori $24,$0,24

ori $25,$0,25

ori $26,$0,26

ori $27,$0,27

lui $28,28

ori $29,$0,29

ori $30,$0,30

ori $31,$0,31

addu $t1,$t1,$t2

subu $t1,$t1,$t2

ori $t1,$0,8

sw $t1,0($0)

lw $t1,0($0)

ori $t1,$0,12

sw $t1,-4($t1)

sw $t1,4($t1)

ori $t1,$0,4

loop1:

lw $t1,4($t1)

lw $t1,0($t1)

beq $t1,$0,loop1

addu $t1,$t1,$8

beq $t1,$s0,loop2

nop

a3:

j a1

nop

nop

nop

nop

nop

a1:

jal a2

addu $t1,$ra,$t1

sw $ra,0($8)

addu $8,$8,$4

a2:

subu $t1,$ra,$t1

jr $ra

nop

loop2:

addu $t1,$t1,$t2

subu $t3,$t3,$t4

ori $t1,$t1,0x00001234

jal a3

addu $t1,$ra,$t1

期望输出：

45@00003000: $ 0 <= 00000000

55@00003004: $ 1 <= 00000001

65@00003008: $ 2 <= 00000002

75@0000300c: $ 3 <= 00000003

85@00003010: $ 4 <= 00000004

95@00003014: $ 5 <= 00000005

105@00003018: $ 6 <= 00000006

115@0000301c: $ 7 <= 00000007

125@00003020: $ 8 <= 00000008

135@00003024: $ 9 <= 00000009

145@00003028: $10 <= 000a0000

155@0000302c: $11 <= 0000000b

165@00003030: $12 <= 0000000c

175@00003034: $13 <= 0000000d

185@00003038: $14 <= 0000000e

195@0000303c: $15 <= 0000000f

205@00003040: $16 <= 00000010

215@00003044: $17 <= 00000011

225@00003048: $18 <= 00000012

235@0000304c: $19 <= 00000013

245@00003050: $20 <= 00000014

255@00003054: $21 <= 00000015

265@00003058: $22 <= 00000016

275@0000305c: $23 <= 00000017

285@00003060: $24 <= 00000018

295@00003064: $25 <= 00000019

305@00003068: $26 <= 0000001a

315@0000306c: $27 <= 0000001b

325@00003070: $28 <= 001c0000

335@00003074: $29 <= 0000001d

345@00003078: $30 <= 0000001e

355@0000307c: $31 <= 0000001f

365@00003080: $ 9 <= 000a0009

375@00003084: $ 9 <= 00000009

385@00003088: $ 9 <= 00000008

385@0000308c: \*00000000 <= 00000008

405@00003090: $ 9 <= 00000008

415@00003094: $ 9 <= 0000000c

415@00003098: \*00000008 <= 0000000c

425@0000309c: \*00000010 <= 0000000c

445@000030a0: $ 9 <= 00000004

455@000030a4: $ 9 <= 0000000c

475@000030a8: $ 9 <= 00000000

515@000030b0: $ 9 <= 00000008

525@000030a4: $ 9 <= 00000000

545@000030a8: $ 9 <= 00000008

585@000030b0: $ 9 <= 00000010

625@000030f0: $ 9 <= 000a0010

635@000030f4: $11 <= ffffffff

645@000030f8: $ 9 <= 000a1234

655@000030fc: $31 <= 00003104

665@00003100: $ 9 <= 000a4338

695@000030d4: $31 <= 000030dc

705@000030d8: $ 9 <= 000a7414

715@000030e4: $ 9 <= fff5bcc8

735@000030f0: \*00000008 <= 000030dc

755@000030e0: $ 8 <= 0000000c

765@000030e4: $ 9 <= 000a7414

785@000030f0: \*0000000c <= 000030dc

805@000030e0: $ 8 <= 00000010

815@000030e4: $ 9 <= fff5bcc8

835@000030f0: \*00000010 <= 000030dc

855@000030e0: $ 8 <= 00000014

865@000030e4: $ 9 <= 000a7414

885@000030f0: \*00000014 <= 000030dc

905@000030e0: $ 8 <= 00000018

915@000030e4: $ 9 <= fff5bcc8

935@000030f0: \*00000018 <= 000030dc

955@000030e0: $ 8 <= 0000001c

965@000030e4: $ 9 <= 000a7414

985@000030f0: \*0000001c <= 000030dc

该程序最后为死循环，将所有DM中的值都赋为0x000030dc

五、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**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tuse\_rs | Tuse\_rt | Tnew\_E | 冲突寄存器 | 解决方案 | 测试程序 |
| 0 |  | 1 | rs | 暂停1个时钟周期后从M到D转发 | ori $ra $0,0x0000300c  jr $ra |
| 0 |  | 2 | rs | 暂停2个时钟周期后从W到D转发 | lw $ra,0($0)  jr $ra |
| 0 | 0 | 1 | rs或rt | 暂停1个时钟周期后从M到D转发 | addu $t1,$t1,$t2  beq $t1,$t2,loop |
| 0 | 0 | 2 | rs或rt | 暂停两个时钟周期后从W到D转发 | lw $t1,0($0)  beq $t1,$t2,loop |
| 1 | 1 | 1 | rs或rt | 从M到E转发 | addu $t1,$t1,$t2  subu $t1,$t1,$t2 |
| 1 | 1 | 2 | rs或rt | 暂停一个时钟周期后从W到E转发 | lw $t1,0($0)  addu $t1,$t1,$t2 |
| 1 |  | 1 | rs | 从M到E级转发 | ori $t1,$0,1  lui $t1,1 |
| 1 |  | 2 | rs | 暂停一个时钟周期后从W到E转发 | lw $t1,0($0)  lui $t1,1 |
| 1 | 2 | 1 | rs | 从M到E级转发 | lui $t1,1  sw $t1,0($0) |
| 1 | 2 | 2 | rs | 暂停1个周期后从W到E级转发 | lw $t1,0($0)  sw $t2,0($t1) |
| 1 | 2 | 2 | rt | 从W到M级转发 | lw $t1,0($0)  sw $t1,0($0) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tuse\_rs | Tuse\_rt | Tnew\_M | 冲突寄存器 | 解决方案 | 测试程序 |
| 0 |  | 1 | rs | 暂停一个周期后从W到D级转发 | lw $ra,0($0)  lui $t2,1  jr $ra |
| 0 | 0 | 1 | rs或rt | 暂停一个周期后从W到D级转发 | lw $t1,0($0)  lui $t2,1  beq $t1,$t2,loop |
| 1 | 1 | 1 | rs或rt | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  addu $t1,$t3,$t1 |
| 1 |  | 1 | rs | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  ori $t1,$t1,1 |
| 1 | 2 | 1 | rs | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  sw $t1,0($0) |
| 0 |  | 0 | rs | 从M级到D级转发 | addu $ra,$t1,$t2  lui $t2,1  jr $ra |
| 0 | 0 | 0 | rs或rt | 从M级到D级转发 | addu $t1,$t1,$t2  lui $t3,1  beq $t1,$t2,loop |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tuse\_rs | Tuse\_rt | Tnew\_W | 冲突寄存器 | 解决方案 | 测试程序 |
| 0 |  | 0 | rs | 从W到D级转发 | addu $ra,$t1,$t2  nop  nop  jr $ra |
| 0 | 0 | 0 | rs或rt | 从W到D级转发 | addu $t1,$t1,$t2  nop  nop  beq $t1,$t2,loop |