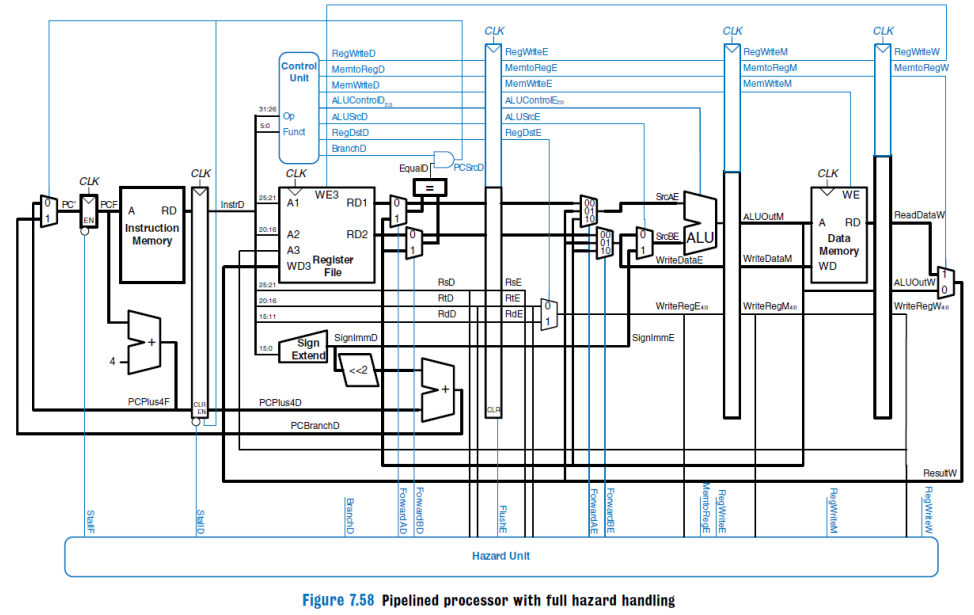
P6利用Verilog开发MIPS流水线处理器plus

一．整体结构：

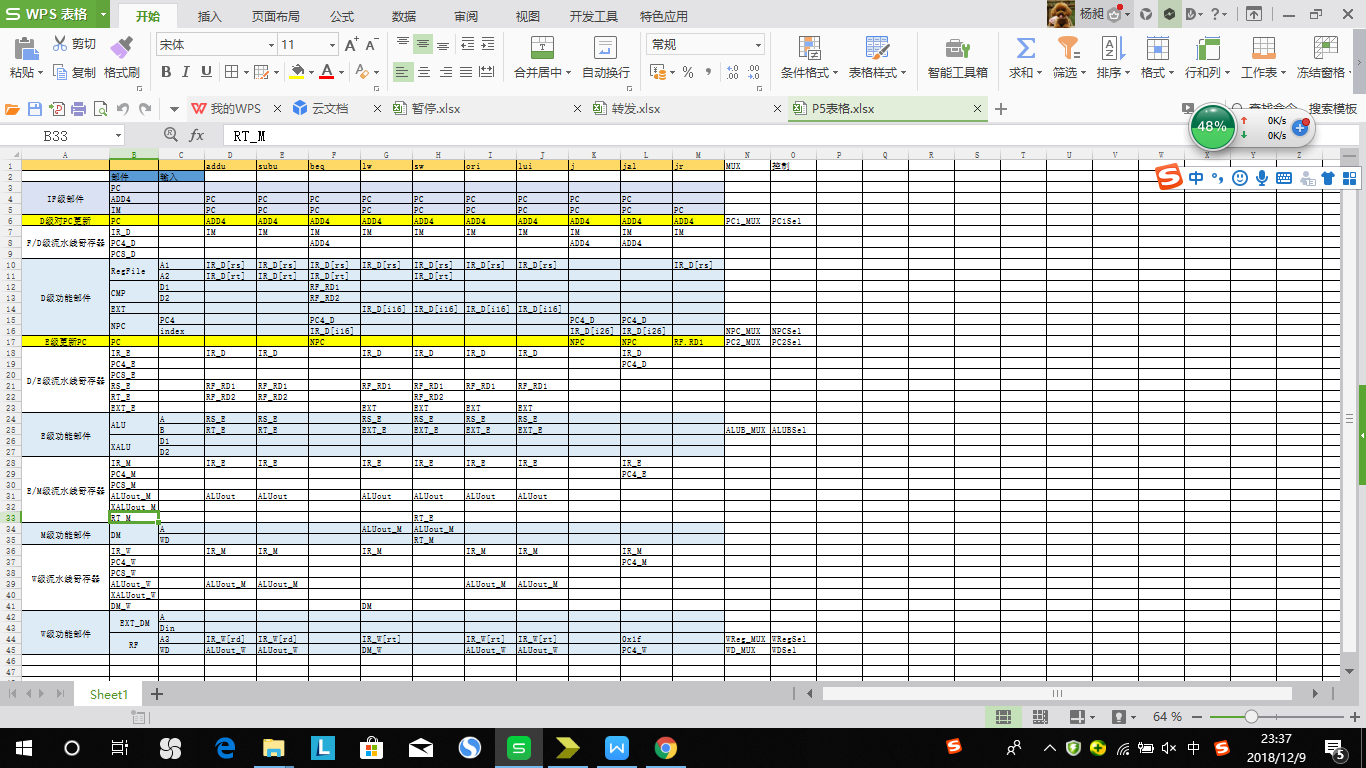
流水线处理器包括流水寄存器、各级组合逻辑以及各级控制器三大部分

它们均放在mips.v层次下，其中code.txt中存储相应指令码

处理器为32位处理器，支持的指令集为：**addu,subu, ori, lw, sw, beq, lui, j,jal, jr,nop**



1. **数据通路部分**



**1.IF级组合逻辑：**

**（1）PC.V**

**模块定义：**

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| **clk** | **I** | **时钟信号** |
| **reset** | **I** | **复位信号**  **0：无效**  **1：有效** |
| **stall** | **I** | **阻塞/暂停信号：**  **0：pc=npc**  **1：pc保持不变** |
| **npc[31:0]** | **I** | **输入的PC地址** |
| **pc** | **O** | **输出当前PC地址** |

功能定义：

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 功能定义 |
| 1 | 复位 | 当时钟上升沿来临时，若复位信号有效，PC=0x00003000 |
| 2 | 取地址 | 时钟上升沿来临输出读取地址 |

1. IM.V

容量为16KB(32bit/word×4096word)

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| pc[31:0] | I | 当前PC地址 |
| Instr[31:0] | O | 当前读取的指令 |

（3）ADD4.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| pc[31:0] | I | 当前pc地址 |
| pcplus4[31:0] | O | 输出数据为地址加4 |

（4）ADD8.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| pc[31:0] | I | 当前pc地址 |
| Pcplus8[31:0] | O | 输出数据为地址加8 |

2.IF/ID级流水寄存器：

IF\_ID\_register.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号  0:无效  1：有效 |
| en | I | 写使能信号  0：不可写流水寄存器  1：可写流水寄存器 |
| IR\_D\_in[31:0] | I | 传入该寄存器的指令 |
| PC4\_D\_in[31:0] | I | 传入该寄存器的PC+4 |
| PC8\_D\_in[31:0] | I | 传入该寄存器的PC+8 |
| IR\_D\_out[31:0] | O | 传出该寄存器的指令 |
| PC4\_D\_out[31:0] | O | 传出该寄存器的PC+4 |
| PC8\_D\_out[31:0] | O | 传出该寄存器的PC+8 |

功能定义：

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 功能定义 |
| 1 | 复位 | 当时钟上升沿来临时，若复位信号有效，寄存器内容全为零 |
| 2 | 取地址 | 时钟上升沿来临时输出读取地址 |
| 3 | 取指令 | 时钟上升沿来临时取出当前指令 |

3.ID级组合逻辑：

（1）GRF.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号，将32个寄存器中的值全部清零  1：有效  0：无效 |
| pc[31:0] | I | W级PC地址（PC4\_W-4） |
| RegWrite\_W | I | W级写使能信号  1：可向GRF中写入数据  0：不能向GRF中写入数据 |
| Read\_register1[4:0] | I | 5位地址输入信号，指定32个寄存器中的一个，将其中存储的数据读出到D1 |
| Read\_register2[4:0] | I | 5位地址输入信号，指定32个寄存器中的一个，将其中存储的数据读出到D2 |
| Write\_register\_W | I | 5位地址输入信号，指定32个寄存器中的一个作为写入的目标寄存器 |
| Write\_data\_W[31:0] | I | 向写入寄存器写入的数据 |

功能定义：

|  |  |  |
| --- | --- | --- |
| **序号** | **功能名称** | **功能描述** |
| 1 | 复位 | reset信号有效时，所有寄存器存储的数值清零 |
| 2 | 读数据 | 读出Read\_register1,Read\_register2地址对应寄存器中所存储的数据到RF.RD1,RF.RD2 |
| 3 | 写数据 | 当WE有效且时钟上升沿来临时，将Write\_data\_W写入Write\_register\_W所对应的寄存器中 |

（2）EXT.v:

功能：选择立即数扩展方式

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| imm[15:0] | I | 输入数据 |
| Extop[1:0] | I | 选择信号：  00：无符号扩展  01：有符号扩展  10：加载至高位，低位补零 |
| after\_ext[31:0] | O | 符号扩展后输出数据 |

1. CMP.v

功能：比较器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| D1[31:0] | I | 第一个比较的数 |
| D2[31:0] | I | 第二个比较的数 |
| equal | O | 判断信号  1：D1=D2  0:D1!=D2 |
| g\_or\_e | O | 判断信号  1：D1>=D2  0:D1<D2 |
| greater | O | 判断信号  1：D1>D2  0:D1<=D2 |

1. PC\_beq.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| after\_ext[31:0] | I | EXT扩展后的数 |
| PC4\_D[31:0] | I | PC+4的值 |
| equal | I | 相等信号 |
| g\_or\_e | I | 大于等于信号 |
| greater | I | 大于信号 |
| pc\_beq | O | B类指令跳转地址 |

1. PC\_jal.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Instr[31:0] | I | 指令 |
| PC4\_D[31:0] | I | PC+4的值 |
| pc\_jal | O | jal指令跳转地址 |

1. MFRSD.v

功能：D级rs转发多选器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| RF\_RD1[31:0] | I | rs寄存器里面内容 |
| ALUout\_M[31:0] | I | M级ALUout数据 |
| Write\_data\_W[31:0] | I | W级多选器的输出内容 |
| ForwardRSD[1:0] | I | 选择信号  00：RF\_RD1\_trans=RF\_RD1  01：RF\_RD1\_trans=ALUout\_M  10：RF\_RD1\_trans=Write\_data\_W |
| RF\_RD1\_trans[31:0] | O | 选择出来的数据 |

1. MFRTD.v

功能：D级rt转发多选器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| RF\_RD2[31:0] | I | rs寄存器里面内容 |
| ALUout\_M[31:0] | I | M级ALUout数据 |
| Write\_data\_W[31:0] | I | W级多选器的输出内容 |
| ForwardRTD[1:0] | I | 选择信号  00：RF\_RD2\_trans=RF\_RD2  01：RF\_RD2\_trans=ALUout\_M  10：RF\_RD2\_trans=Write\_data\_W |
| RF\_RD2\_trans[31:0] | O | 选择出来的数据 |

1. nextpc\_2.v

功能：跳转pc的选择

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| pc\_jal[31:0] | I | jal跳转的地址 |
| pc\_beq[31:0] | I | beq跳转的地址 |
| RF\_RD1\_trans[31:0] | I | jr跳转的地址 |
| pc\_sel2[1:0] | I | 选择信号  00：nextpc=pc\_jal  01：nextpc=pc\_beq  10：nextpc=RF\_RD1\_trans |
| nextpc[31:0] | O | 选择出来的nextpc |

1. Wreg\_D.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Instr[20:16] | I | rt寄存器 |
| Instr[15:11] | I | rd寄存器 |
| 5'b11111 | I | 31号（$ra）寄存器 |
| RegDst[1:0] | I | 写寄存器选择信号  00：write\_register\_D=rt  01：write\_register\_D=rd  10：write\_register\_D=$ra |
| write\_register\_D[4:0] | O | 选择出来的写寄存器 |

4.ID/EX级流水寄存器：

ID\_EX\_register.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号  0:无效  1：有效 |
| stall | I | 阻塞/暂停信号 |
| IR\_E\_in[31:0] | I | 传入该寄存器的指令 |
| PC4\_E\_in[31:0] | I | 传入该寄存器的PC+4 |
| PC8\_E\_in[31:0] | I | 传入该寄存器的PC+8 |
| RS\_E\_in[31:0] | I | 由rs寄存器传出，传入该寄存器的值 |
| RT\_E\_in[31:0] | I | 由rt寄存器传出，传入该寄存器的值 |
| EXT\_E\_in[31:0] | I | 传入该寄存器的立即数扩展之后的值 |
| write\_register\_E\_in[4:0] | I | 传入该寄存器的写寄存器 |
| Start\_E\_in | I | 传入该寄存器的触发信号 |
| IR\_E\_out[31:0] | O | 传出该寄存器的指令 |
| PC4\_E\_out[31:0] | O | 传出该寄存器的PC+4 |
| PC8\_E\_out[31:0] | O | 传出该寄存器的PC+8 |
| RS\_E\_out[31:0] | O | 由rs寄存器传出，传出该寄存器的值 |
| RT\_E\_out[31:0] | O | 由rt寄存器传出，传出该寄存器的值 |
| EXT\_E\_out[31:0] | O | 传出该寄存器的立即数扩展之后的值 |
| write\_register\_E\_out[4:0] | O | 传出该寄存器的写寄存器 |
| Start\_E\_out | O | 传出该寄存器的触发信号 |

5.EX级组合逻辑：

（1）ALU\_data\_B.v

功能：选择进入ALU的第二个数据值

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| RF\_RD2\_trans[31:0] | I | 经过转发选择后的1寄存器的值 |
| after\_ext[31:0] | I | 立即数扩展之后的值 |
| ALUSrc | I | 选择信号 |
| ALUB | O | 输入ALU的第二个数据值 |

（2）ALU.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A[31:0] | I | 输入A数据 |
| B[31:0] | I | 输入B数据 |
| shamt[4:0] | I | 输入的5位移位的位数 |
| ALUop[3:0] | I | 选择信号：  4'b0000:Result=A+B;  4'b0001:Result=A-B;  4'b0010:Result=A|B;  4'b0011:Result=A&B;  4'b0100:Result=A^B;  4'b0101:Result=~(A|B);  4'b0110:Result=B<<shamt;  4'b0111:Result=B<<A[4:0];  4'b1000:Result=B>>shamt;  4'b1001:Result=B>>A[4:0];  4'b1010:Result=$signed(B)>>>shamt;  4'b1011:Result=$signed(B)>>>A[4:0];  4'b1100:Result=($signed(A)<$signed(B))?32'b1:32'b0;  4'b1101:Result=(A<B)?32'b1:32'b0; |
| Result[31:0] | O | 计算后输出数据 |

1. MFRSE.v

功能：E级转发多选器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| RF\_RD1[31:0] | I | 第一个寄存器传出来的值 |
| ALUout\_M[31:0] | I | M级ALUout数据 |
| Write\_data\_W[31:0] | I | W级多选器的输出内容 |
| ForwardRSE[1:0] | I | 选择信号  00：RF\_RD1\_trans=RF\_RD1  01：RF\_RD1\_trans=ALUout\_M  10：RF\_RD1\_trans=Write\_data\_W |
| RF\_RD1\_trans[31:0] | O | rs转发多选器选出来的值 |

1. MFRTE.v

功能：E级转发多选器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| RF\_RD2[31:0] | I | 第二个寄存器传出来的值 |
| ALUout\_M[31:0] | I | M级ALUout数据 |
| Write\_data\_W[31:0] | I | W级多选器的输出内容 |
| ForwardRTE[1:0] | I | 选择信号  00：RF\_RD2\_trans=RF\_RD2  01：RF\_RD2\_trans=ALUout\_M  10：RF\_RD2\_trans=Write\_data\_W |
| RF\_RD2\_trans[31:0] | O | rt转发多选器选出来的值 |

（5）XALU.v

功能：有限状态机

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 |
| Instr[31:0] | I | 31位指令 |
| D1 | I | 输入的第一个乘除数据 |
| D2 | I | 输入的第二个乘除数据 |
| Start | I | 乘除类指令开始信号 |
| HI[31:0] | O | HI寄存器中的值 |
| LO[31:0] | O | LO寄存器中的值 |
| Busy | O | 阻塞信号 |

6.EX/MEM级流水寄存器：

EX\_MEM\_register.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号  0:无效  1：有效 |
| IR\_M\_in[31:0] | I | 传入该寄存器的指令 |
| PC4\_M\_in[31:0] | I | 传入该寄存器的PC+4 |
| PC8\_M\_in[31:0] | I | 传入该寄存器的PC+8 |
| ALUout\_M\_in[31:0] | I | 由ALU传出，传入该寄存器的值 |
| RT\_M\_in[31:0] | I | 由rt寄存器传出，传入该寄存器的值 |
| RegWrite\_M\_in | I | 传入该寄存器的写信号 |
| write\_register\_M\_in[4:0] | I | 传入该寄存器的写寄存器 |
| IR\_M\_out[31:0] | O | 传出该寄存器的指令 |
| PC4\_M\_out[31:0] | O | 传出该寄存器的PC+4 |
| PC8\_M\_out[31:0] | O | 传出该寄存器的PC+8 |
| ALUout\_M\_out[31:0] | O | 由ALU传出，传出该寄存器的值 |
| RT\_M\_out[31:0] | O | 由rt寄存器传出，传出该寄存器的值 |
| RegWrite\_M\_out | O | 传出该寄存器的写信号 |
| write\_register\_M\_out[4:0] | O | 传出该寄存器的写寄存器 |

7.MEM级组合逻辑

（1）DM.v

功能：对内存进行读写操作

容量为16KB(32bit/word×4096word)

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号  0：无效  1：有效 |
| pc[31:0] | I | pc现在地址 |
| addr[31:0] | I | 存数据的地址 |
| MemWrite | I | 写内存信号  0：不可写内存  1：可写内存 |
| MemData[31:0] | I | 存入的数据 |
| BE[3:0] | I | 字节有效信号 |
| ext\_op | I | 扩展方式信号 |
| DMout[31:0] | O | 读出对应内存位置的数据 |

|  |  |  |
| --- | --- | --- |
| **序号** | **功能名称** | **功能描述** |
| 1 | 读数据 | 读出pc地址对应内存中所存储的数据到DMout |
| 2 | 写数据 | 当MemWrite有效且时钟上升沿来临时，将MemData写入addr所对应的内存位置 |

功能定义：

1. MFRTM.v

功能：M级rt转发选择器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| WD[31:0] | I | M级ALUout\_M的数据 |
| Write\_data\_W[31:0] | I | W级多选器的输出内容 |
| ForwardRTM | I | 选择信号  0：Write\_data\_trans=WD  1：Write\_data\_trans=Write\_data\_W |
| Write\_data\_trans[31:0] | O | 输出传至DMWD端口数据 |

1. ByteEnable.v

功能：判断哪些字节有效

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| s\_l\_op[1:0] | I | 指令信号  01：sw、lw  10：sh、lh、lhu  11---sb、lb、lbu |
| addr[31:0] | I | 31位指令 |
| BE[3:0] | O | 输出哪些字节有效 |

8.MEM/WB级流水寄存器：

MEM\_WB\_register.v

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号  0:无效  1：有效 |
| IR\_W\_in[31:0] | I | 传入该寄存器的指令 |
| PC4\_W\_in[31:0] | I | 传入该寄存器的PC+4 |
| PC8\_W\_in[31:0] | I | 传入该寄存器的PC+8 |
| ALUout\_W\_in[31:0] | I | 由ALU传出，传入该寄存器的值 |
| DM\_W\_in[31:0] | I | 由DM传出，传入该寄存器的值 |
| RegWrite\_W\_in | I | 传入该寄存器的写信号 |
| write\_register\_W\_in[4:0] | I | 传入该寄存器的写寄存器 |
| IR\_W\_out[31:0] | O | 传出该寄存器的指令 |
| PC4\_W\_out[31:0] | O | 传出该寄存器的PC+4 |
| PC8\_W\_out[31:0] | O | 传出该寄存器的PC+8 |
| ALUout\_W\_out[31:0] | O | 由ALU传出，传出该寄存器的值 |
| DM\_W\_out[31:0] | O | 由DM传出，传出该寄存器的值 |
| RegWrite\_W\_out | O | 传出该寄存器的写信号 |
| write\_register\_W\_out[4:0] | O | 传出该寄存器的写寄存器 |

9.WB级组合逻辑：

（1）DATAtoREG.v

功能：选择回写寄存器堆的数据来源

模块定义：

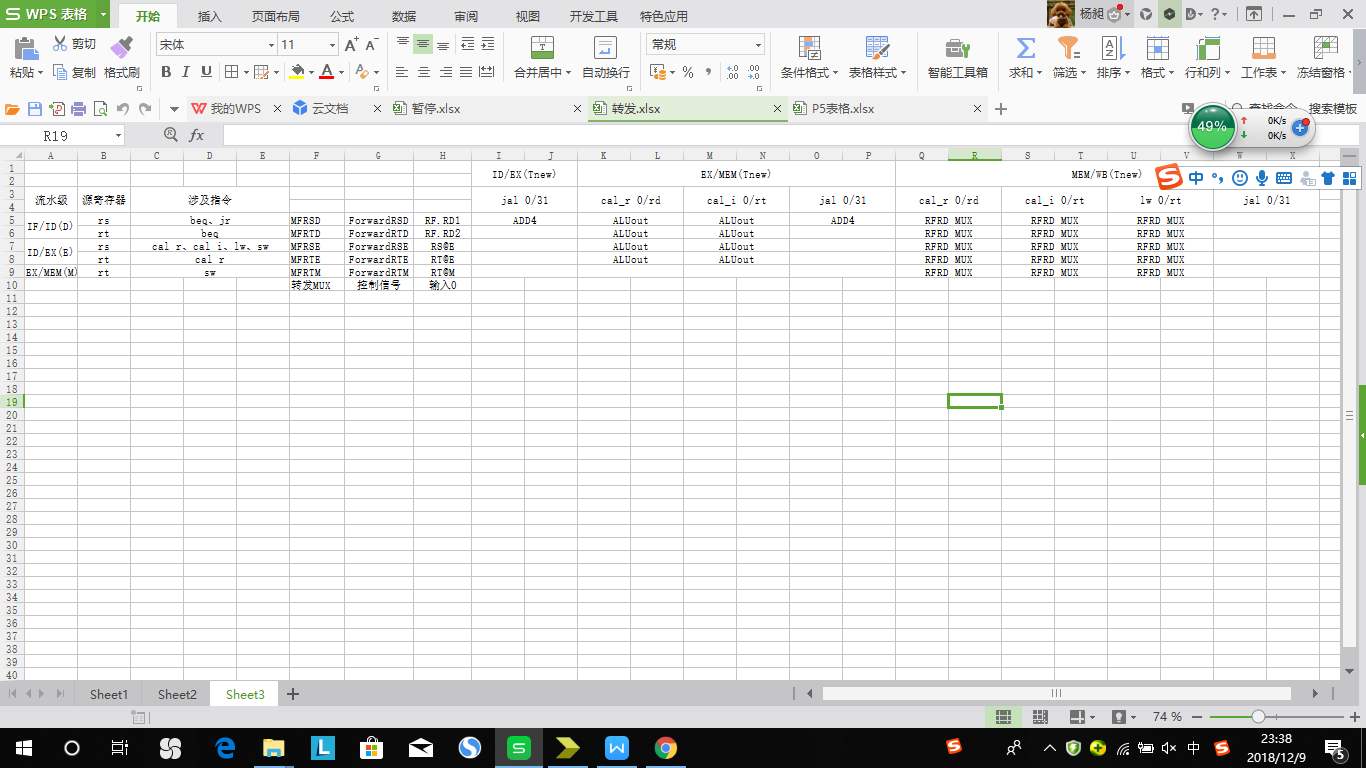
|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| ALUout\_W[31:0] | I | 从ALU出来的数据 |
| DMout[31:0] | I | 从DM出来的数据 |
| MemtoReg | I | 选择信号：  0：writeback\_data=ALUout\_W  1：writeback\_data=DMout |
| writeback\_data[31:0] | O | 输出数据作为回写寄存器内容 |

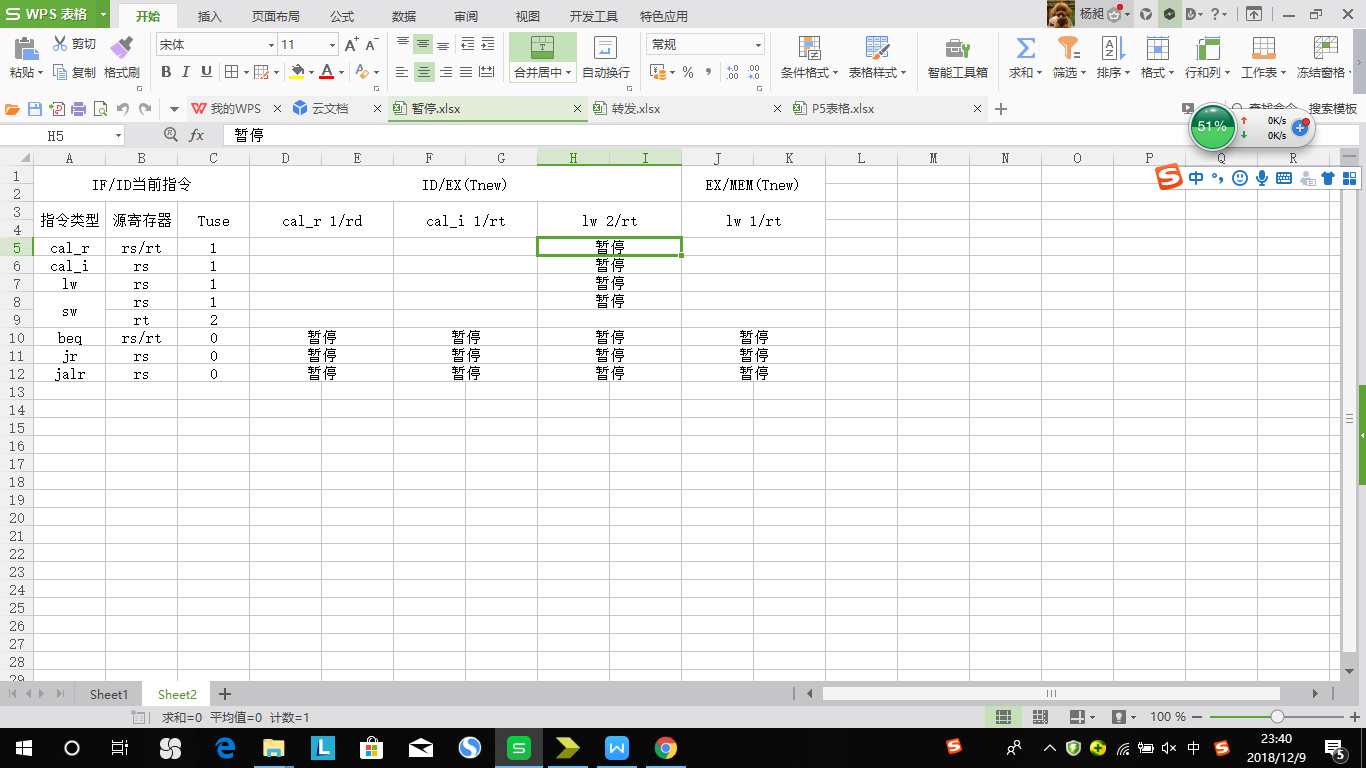
三．控制器

模块定义：

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| func[5:0] | I | 6位func |
| op[5:0] | I | 6位op |
| RegDst[1:0] | O | 写寄存器选择信号 |
| ALUSrc | O | 进入ALU的第二个值选择信号 |
| MemtoReg | O | 写回写寄存器的数据选择信号 |
| RegWrite | O | 写入寄存器信号 |
| MemWrite | O | 写入DM信号 |
| Extop[1:0] | O | 位扩展信号 |
| ALUop[1:0] | O | ALU功能选择信号 |
| pc\_sel1 | O | 是否跳转信号 |
| pc\_sel2[1:0] | O | 哪种跳转方式选择信号 |
| Cal\_r | O | Cal\_r类信号 |
| Cal\_i | O | Cal\_i类信号 |
| B | O | Beq类信号 |
| Load | O | Load类信号 |
| Save | O | Save类信号 |
| J | O | J类信号 |
| mf\_type[1:0] | O | mf类型指令信号  00:非mf类型  01：对应HI  10：对应LO |
| s\_l\_op[1:0] | O | 存取指令的类型信号 |
| ext\_op | O | 存取指令进行扩展方式  0：0扩展  1：符号扩展 |

1. 转发和暂停





对于暂停信号stall

stall\_B\_Calr=(B\_D&&Cal\_r\_E&&((IR\_D\_out[25:21]==IR\_E\_out[15:11])|(IR\_D\_out[20:16]==IR\_E\_out[15:11])));

stall\_B\_Cali=(B\_D&&Cal\_i\_E&&((IR\_D\_out[25:21]==IR\_E\_out[20:16])|(IR\_D\_out[20:16]==IR\_E\_out[20:16])));

stall\_B\_Load1=(B\_D&&Load\_E&&((IR\_D\_out[25:21]==IR\_E\_out[20:16])|(IR\_D\_out[20:16]==IR\_E\_out[20:16])));

stall\_B\_Load2=(B\_D&&Load\_M&&((IR\_D\_out[25:21]==IR\_M\_out[20:16])|(IR\_D\_out[20:16]==IR\_M\_out[20:16])));

stall\_Calr\_Load=(Cal\_r\_D&&Load\_E&&((IR\_D\_out[25:21]==IR\_E\_out[20:16])|(IR\_D\_out[20:16]==IR\_E\_out[20:16])));

stall\_Cali\_Load=(Cal\_i\_D&&Load\_E&&(IR\_D\_out[25:21]==IR\_E\_out[20:16]));

stall\_Load\_Load =(Load\_D&&Load\_E&&(IR\_D\_out[25:21]==IR\_E\_out[20:16]));

stall\_Save\_Load =(Save\_D&&Load\_E&&(IR\_D\_out[25:21]==IR\_E\_out[20:16]));

stall=stall\_m|stall\_B\_Calr|stall\_B\_Cali|stall\_B\_Load1|stall\_B\_Load2|stall\_Calr\_Load|stall\_Cali\_Load|stall\_Load\_Load|stall\_Save\_Load;

1. 思考题
2. 为什么需要有单独的乘除法部件而不是整合进ALU？为何需要有独立的HI、LO寄存器？

答：乘除法运算需要延迟，我们假定乘/除部件的执行乘法的时间为5个cycle，执行除法的时间为10个cycle，在乘除部件需要模拟这个延迟，这会导致CPU运算乘除法延迟时间较长。除此之外，乘除法运算结果会超出32位。为了提高CPU效率，在进行乘除法时不影响其他指令执行，因此需要单独设立模块。

HI和LO寄存器不能被编码。它们在乘除法时被使用，计算结果的不同位储存在不同的寄存器中。避免在简单流水线中从高延迟指令回写寄存器文件的问题。例如：当multu产生计算值后，要写入两个值，与此同时也有可能流水线其他指令也要写入寄存器，这样会产生冲突。而建立HI、LO寄存器，则可以避免冲突。

1. 参照你对延迟槽的理解，试解释“乘除槽”。

答：引入延时槽的主要目的是提高流水线的效率。跟延迟槽类似，乘除槽是指在进行乘除法运算时，和HI或LO寄存器相关的指令全部被冻结在ID级，而其他的指令可以顺利向前进行，不会被冻结阻塞。这样可以保证HI和LO寄存器的值稳定。

1. 为何上文文末提到的lb等指令使用的数据扩展模块应在 MEM/WB 之后，而不能在 DM 之后?

答：放在DM之后一定会使该级延迟增加一个拓展器时间，可能会导致紊乱而时序出错。而放在MEM/WB之后对于一些不需要写寄存器的指令，相对于前一种情况，延时减少了这个拓展器时间。

1. 举例说明并分析何时按字节访问内存相对于按字访问内存性能上更有优势。（Hint： 考虑C语言中字符串的情况）

答：对于C语言中字符串的情况，如果是按照字节寻址，则可以直接找到地址，可以是不是4的倍数的字符。而如果按照字寻址，则还需要首先将寻找地址右移两位（即除以4）后细找，增加了运算时间而且操作起来更加麻烦，这时按字节访问内存相对于按字访问内存性能上更有优势。

1. 如何概括你所设计的CPU的设计风格？为了对抗复杂性你采取了哪些抽象和规范手段？

答：DETECTOR型。hazard单元像是一位各种冲突的侦测者，它本身遵循着自己的一套逻辑，根据对不同阶段流水指令的解析结果，判断出哪个寄存器同时有读入和输出（数据冒险），是否会对程序控制产生影响（控制冒险）。

在转发操作时并没有将指令进行类型划分，而是在代码级对流水线的冒险控制进行了处理。在暂停操作，通过指令类型划分，得出暂停信号的表达。

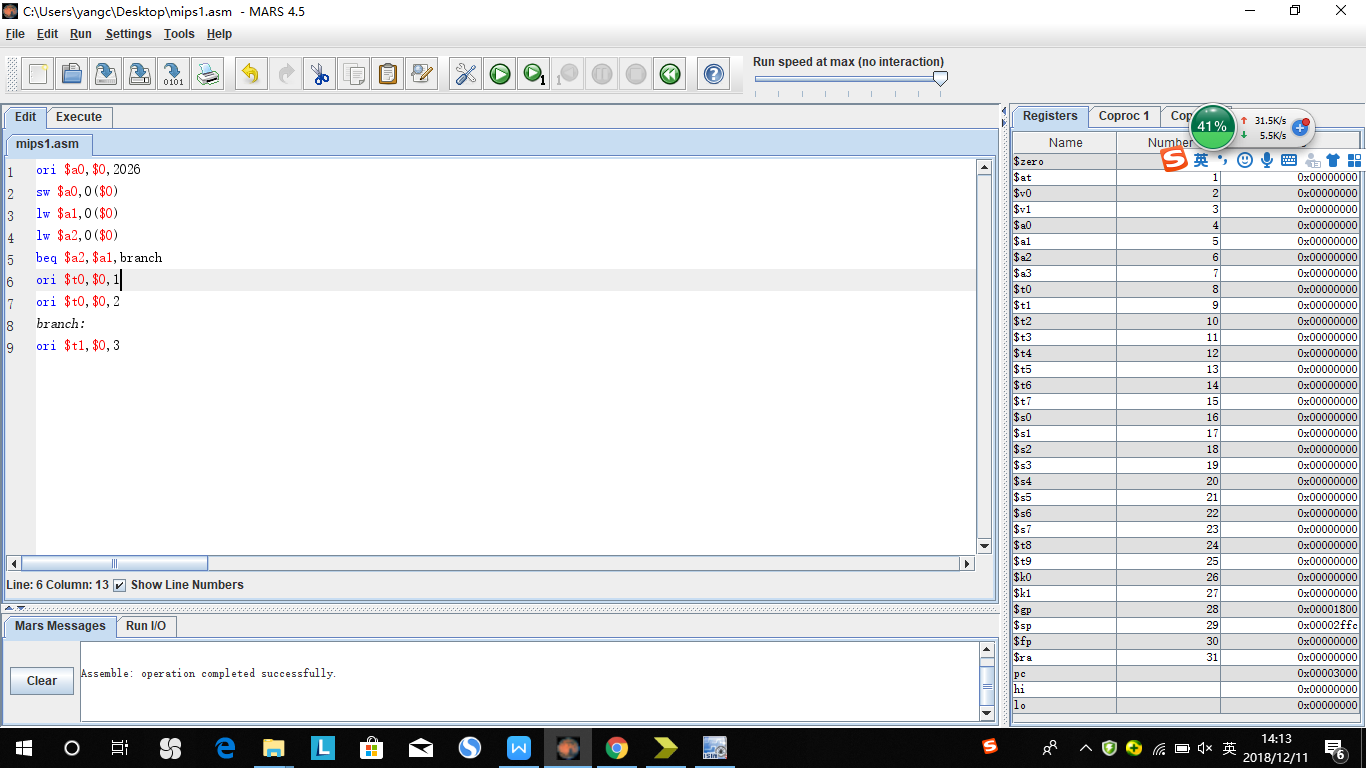
1. 你对流水线CPU设计风格有何见解？

（如果你觉得你的思考值得分享，不妨请在讨论发表你自己的观点和文章，我们会从中发掘优秀文本以飨后辈并予以分数上的鼓励。）

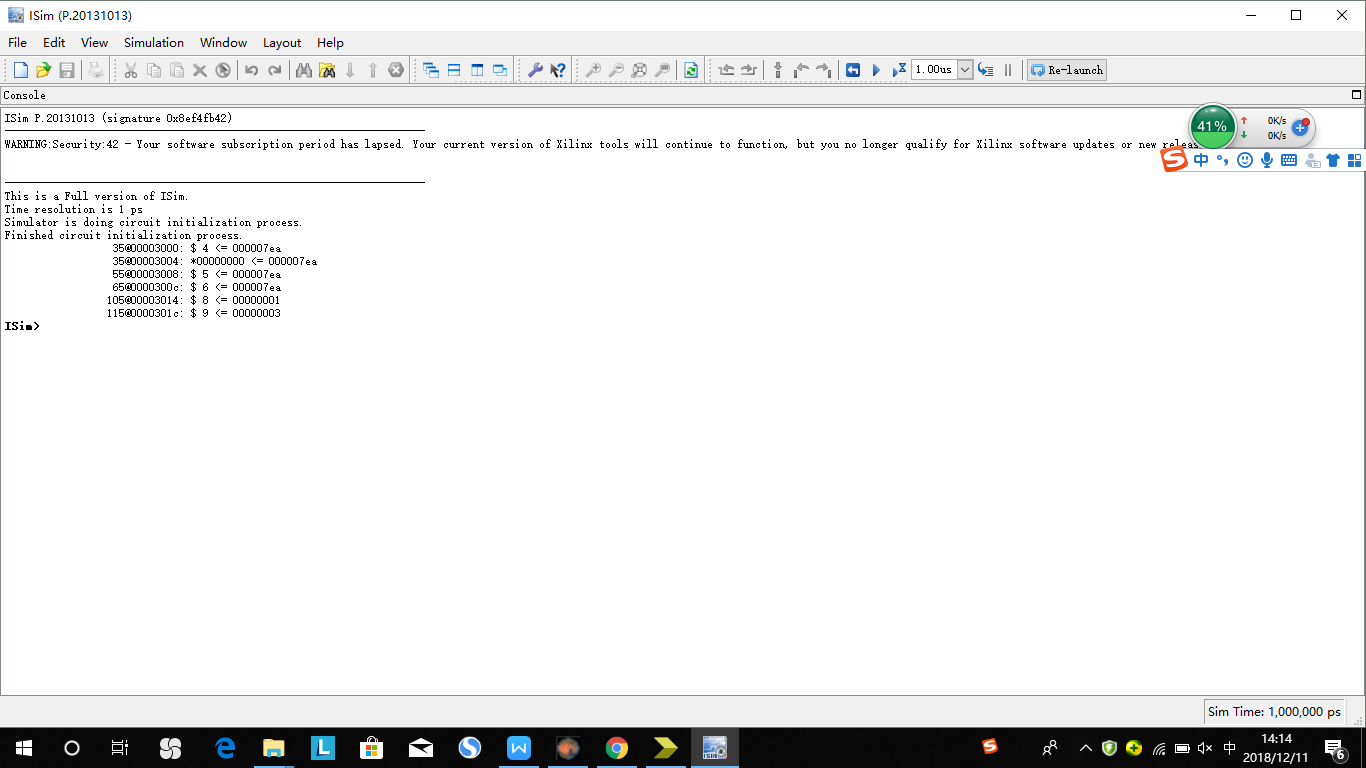
答：这种写法控制代码短小精悍且具有一定的代表性。同段代码适用于多类指令，面对添加指令，很多时候不用修改代码，避免了很多麻烦也降低了出错概率。除此之外，这种方式从冲突产生的原因着手，丛原理上理解。

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)
2. lw后加beq指令出现异常，lw后加beq应该暂停两次。

测试程序：

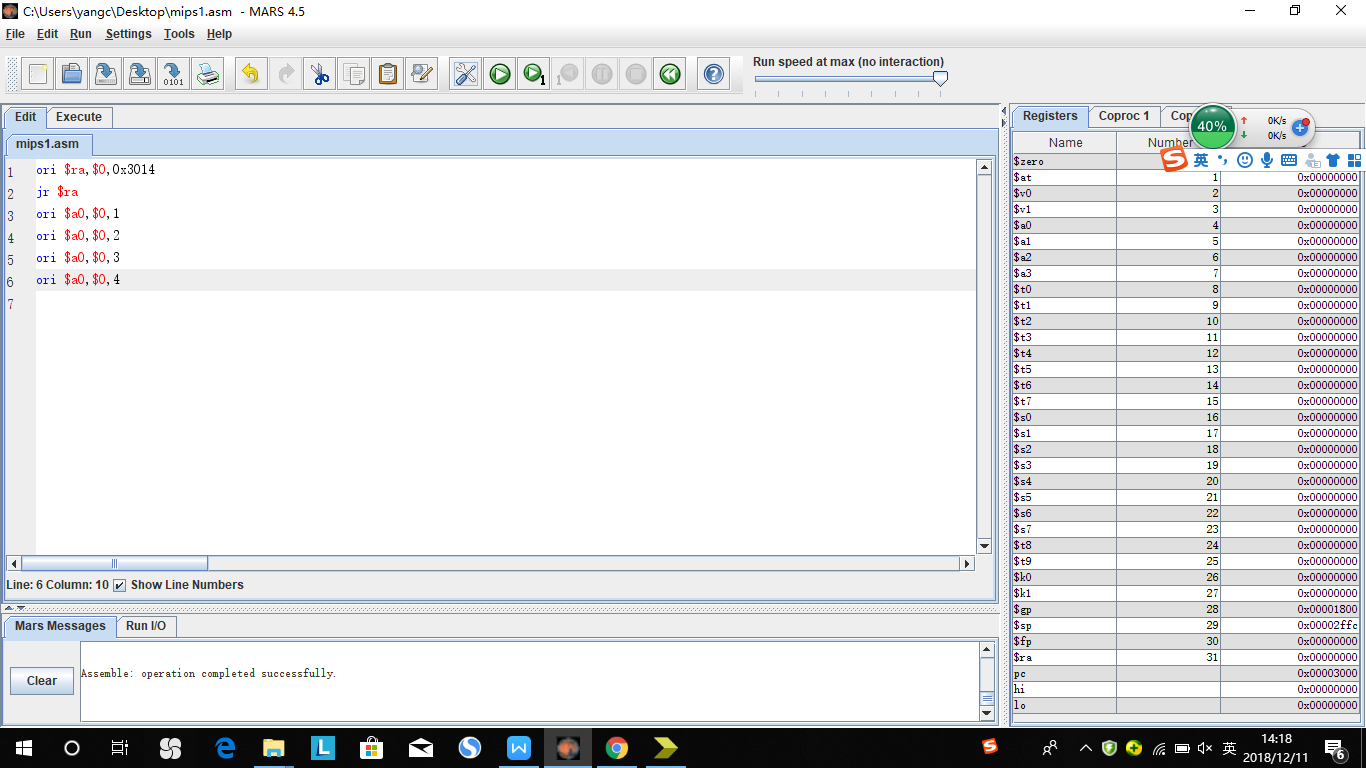


预期结果：

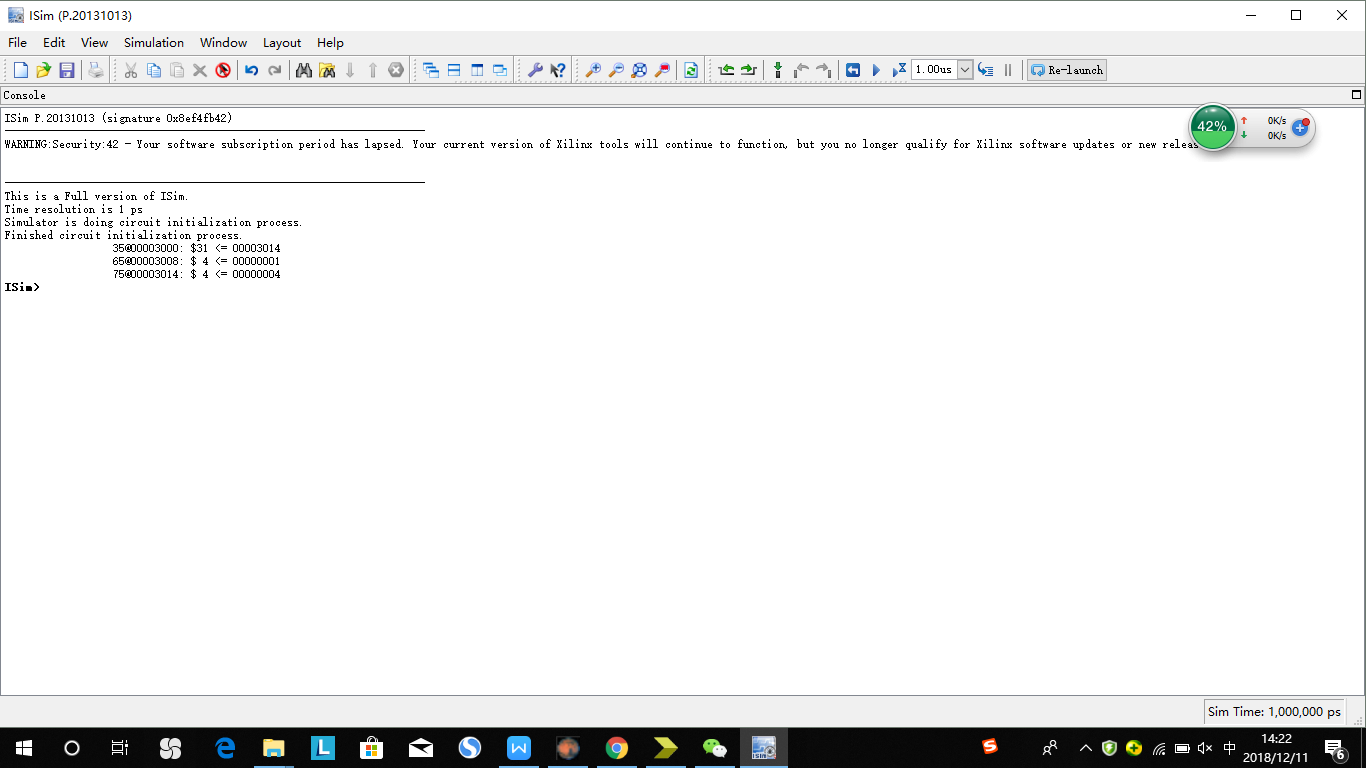


1. jr在D级需要用到rs寄存器的值，需要通过转发来解决，不然中强测前两个点过不去。

测试程序：

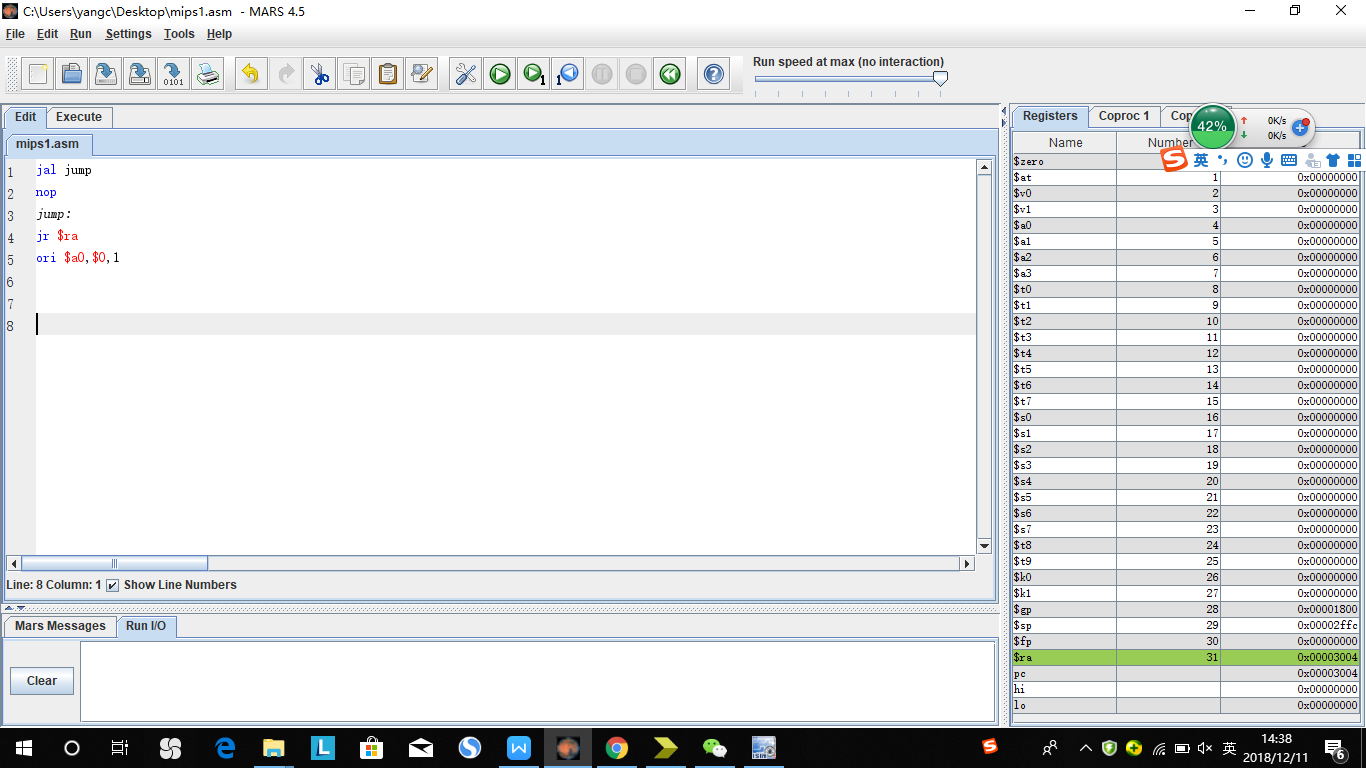


预期结果：

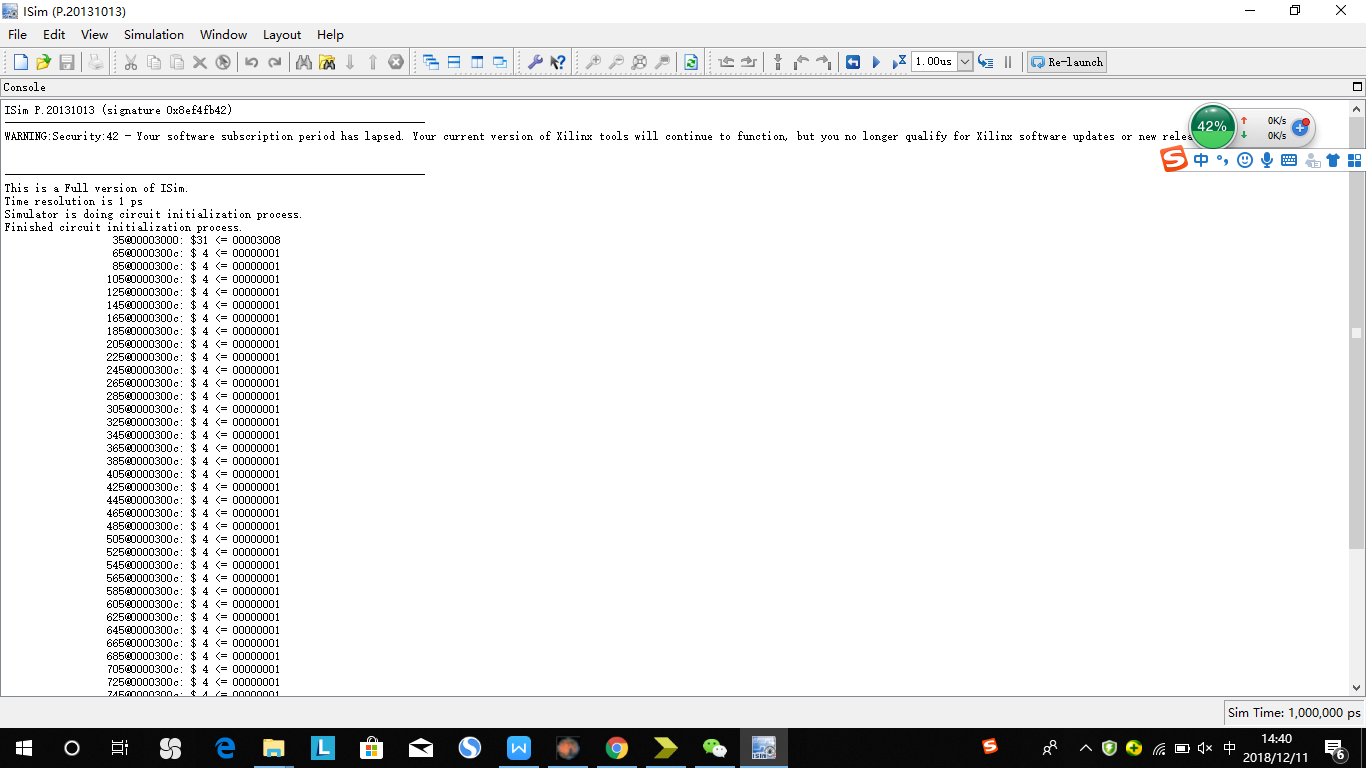


1. 在D级部件将通过转发后的RF\_RD1\_trans和RF\_RD2\_trans转发到了下一级寄存器中，实际上应该将转发前的RF\_RD1和RF\_RD2传到下一级。
2. jal延迟槽后跟jr发生冲突，在E级增加多路选择器，，选择传出的地址是ALU计算出来的值还是PC+8

测试程序：

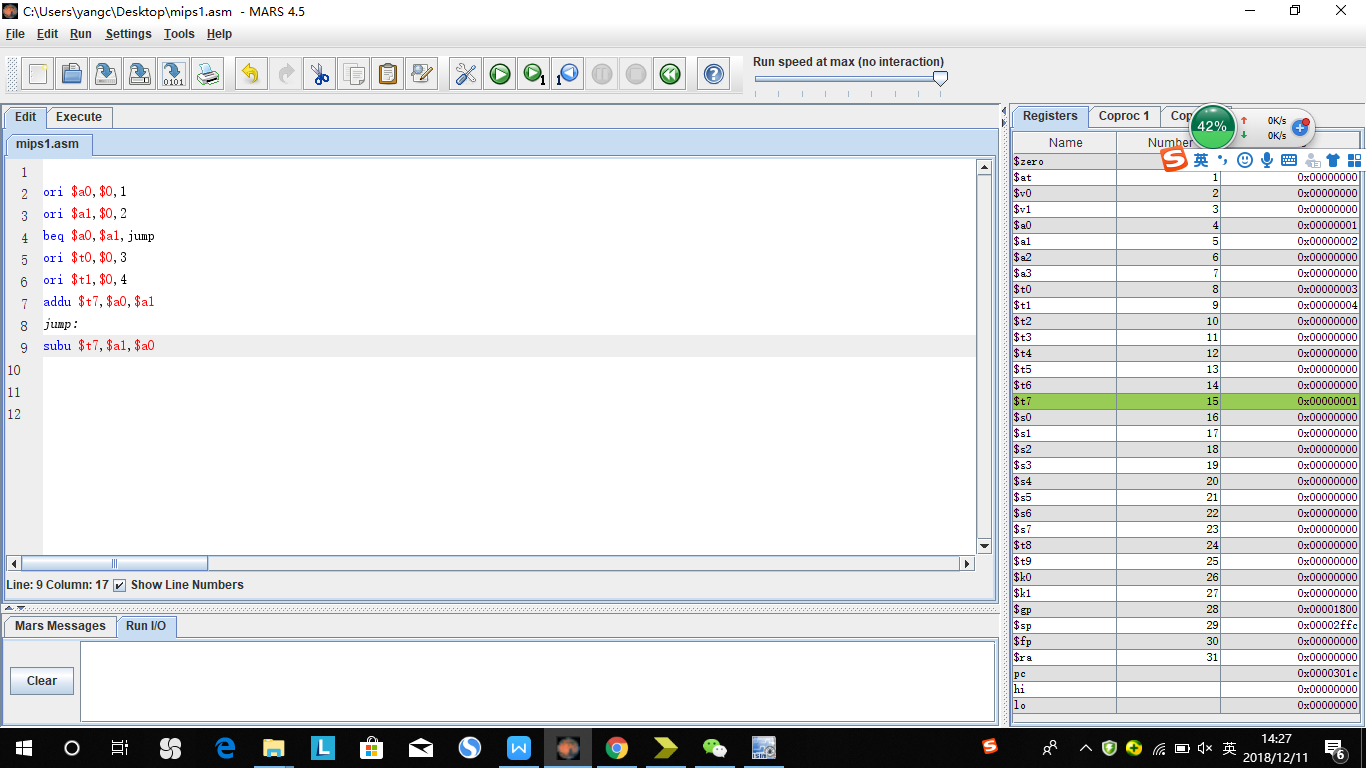


预期结果：

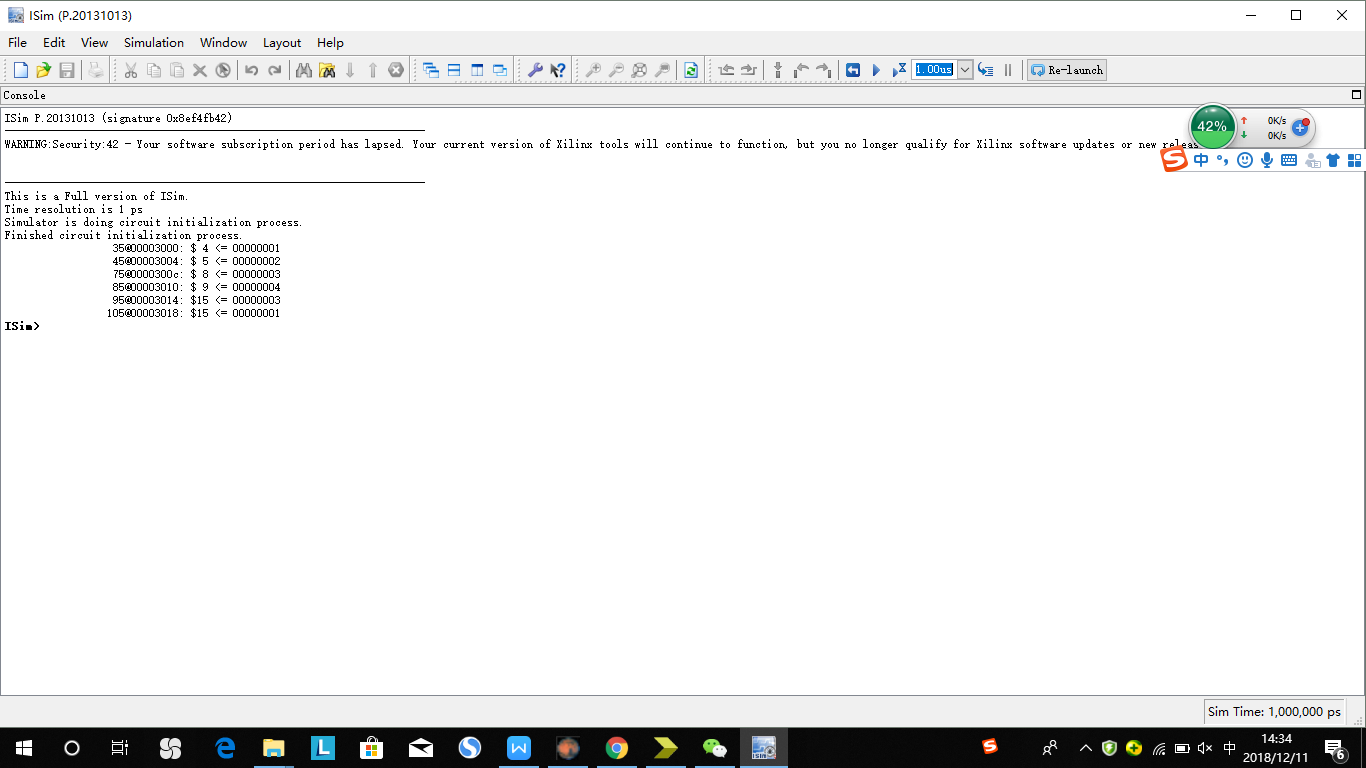


1. beq指令当跳转条件不满足时，应传回PC+8的值，而不是PC+4，不然中强测第二个点过不去。

测试程序：



预期结果：



1. 测试程序

X-Y-Z

X: 冲突前序指令类型

Y: 后续指令位于D级时，前序指令的级数

Z: 产生冲突的寄存器

**Addu:**

**Ori\_E\_RS(addu):**

ori $t0, $zero, 8

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000008

**Ori\_M\_RS(addu)**

ori $t0, $zero, 8

ori $t2, $zero, 12

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 0000000c

260@00003008: $ 9 <= 00000008

**Ori\_W\_RS(addu)**

ori $t0, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 13

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 0000000c

260@00003008: $11 <= 0000000d

300@0000300c: $ 9 <= 00000008

**Ori\_E\_RT(addu)**

ori $t0, $zero, 8

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000008

**Ori\_M\_RT(addu)**

ori $t0, $zero, 8

ori $t2, $zero, 12

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 0000000c

260@00003008: $ 9 <= 00000008

**Ori\_W\_RT(addu)**

ori $t0, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 13

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 0000000c

260@00003008: $11 <= 0000000d

300@0000300c: $ 9 <= 00000008

**Lui\_E\_RS(addu)**

lui $t0, 8

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00080000

220@00003004: $ 9 <= 00080000

**Lui\_M\_RS(addu)**

lui $t0, 8

lui $t2, 12

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00080000

220@00003004: $10 <= 000c0000

260@00003008: $ 9 <= 00080000

**Lui\_W\_RS(addu)**

lui $t0, 8

lui $t2, 12

lui $t3, 14

addu $t1, $t0, $zero

180@00003000: $ 8 <= 00080000

220@00003004: $10 <= 000c0000

260@00003008: $11 <= 000e0000

300@0000300c: $ 9 <= 00080000

**Lui\_W\_RT(addu)**

lui $t0, 8

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00080000

220@00003004: $ 9 <= 00080000

**Lui\_w\_RT(addu)**

lui $t0, 8

lui $t2, 12

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00080000

220@00003004: $10 <= 000c0000

260@00003008: $ 9 <= 00080000

**Lui\_W\_RT(addu)**

lui $t0, 8

lui $t2, 12

lui $t3, 14

addu $t1, $zero, $t0

180@00003000: $ 8 <= 00080000

220@00003004: $10 <= 000c0000

260@00003008: $11 <= 000e0000

300@0000300c: $ 9 <= 00080000

**R\_E\_RS(addu)**

lui $t0, 8

addu $t1, $zero, $t0

addu $t2, $t1, $zero

180@00003000: $ 8 <= 00080000

220@00003004: $ 9 <= 00080000

260@00003008: $10 <= 00080000

**R\_M\_RS(addu)**

lui $t0, 8

addu $t1, $zero, $t0

addu $t3, $t0, $t0

addu $t2, $t1, $zero

180@00003000: $ 8 <= 00080000

220@00003004: $ 9 <= 00080000

260@00003008: $11 <= 00100000

300@0000300c: $10 <= 00080000

**R\_W\_RS(RT)(addu)**

lui $t0, 8

addu $t1, $zero, $t0

addu $t3, $t0, $t0

addu $t4, $t0, $t0

addu $t2, $t1, $t1

180@00003000: $ 8 <= 00080000

220@00003004: $ 9 <= 00080000

260@00003008: $11 <= 00100000

300@0000300c: $12 <= 00100000

340@00003010: $10 <= 00100000

**Ld\_E\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

ori $t3, $zero, 20

ori $t4, $zero, 24

ori $t5, $zero, 28

sw $t1, 0($t0)

ori $s0, $zero, 4

ori $s1, $zero, 8

ori $s2, $zero, 12

lw $t6, 0($t0)

addu $t7, $t6, $t6

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 0000000c

260@00003008: $10 <= 00000010

300@0000300c: $11 <= 00000014

340@00003010: $12 <= 00000018

380@00003014: $13 <= 0000001c

380@00003018: \*00000008 <= 0000000c

460@0000301c: $16 <= 00000004

500@00003020: $17 <= 00000008

540@00003024: $18 <= 0000000c

580@00003028: $14 <= 0000000c

660@0000302c: $15 <= 00000018

**Ld\_M\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

ori $t3, $zero, 20

ori $t4, $zero, 24

sw $t1, 0($t0)

ori $s0, $zero, 4

ori $s1, $zero, 8

ori $s2, $zero, 12

lw $t6, 0($t0)

ori $t5, $zero, 28

addu $t7, $t6, $t6

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 0000000c

260@00003008: $10 <= 00000010

300@0000300c: $11 <= 00000014

340@00003010: $12 <= 00000018

340@00003014: \*00000008 <= 0000000c

420@00003018: $16 <= 00000004

460@0000301c: $17 <= 00000008

500@00003020: $18 <= 0000000c

540@00003024: $14 <= 0000000c

580@00003028: $13 <= 0000001c

620@0000302c: $15 <= 00000018

**Ld\_W\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

ori $t3, $zero, 20

ori $t4, $zero, 24

sw $t1, 0($t0)

ori $s0, $zero, 4

ori $s1, $zero, 8

ori $s2, $zero, 12

lw $t6, 0($t0)

ori $t5, $zero, 28

ori $t8, $zero, 32

addu $t7, $t6, $t6

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 0000000c

260@00003008: $10 <= 00000010

300@0000300c: $11 <= 00000014

340@00003010: $12 <= 00000018

340@00003014: \*00000008 <= 0000000c

420@00003018: $16 <= 00000004

460@0000301c: $17 <= 00000008

500@00003020: $18 <= 0000000c

540@00003024: $14 <= 0000000c

580@00003028: $13 <= 0000001c

620@0000302c: $24 <= 00000020

660@00003030: $15 <= 00000018

**Jal\_E\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

jal change1

addu $t3, $ra, $ra

ori $t4, $zero, 20

ori $t5, $zero, 24

change1:

ori $t6, $zero, 20

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 0000000c

260@00003008: $10 <= 00000010

300@0000300c: $31 <= 00003014

340@00003010: $11 <= 00006028

380@0000301c: $14 <= 00000014

**Jal\_M\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

jal change1

ori $t4, $zero, 20

ori $t5, $zero, 24

change1:

addu $t3, $ra, $ra

ori $t6, $zero, 20

ori $t7, $zero, 24

$ 8 <= 00000008

$ 9 <= 0000000c

$10 <= 00000010

$31 <= 00003014

$12 <= 00000014

$11 <= 00006028

$14 <= 00000014

$15 <= 00000018

**Jal\_W\_RS(RT)(addu)**

ori $t0, $zero, 8

ori $t1, $zero, 12

ori $t2, $zero, 16

jal change1

ori $t4, $zero, 20

ori $t5, $zero, 24

change1:

ori $t6, $zero, 20

addu $t3, $ra, $ra

ori $t6, $zero, 20

ori $t7, $zero, 24

$ 8 <= 00000008

$ 9 <= 0000000c

$10 <= 00000010

$31 <= 00003014

$12 <= 00000014

$14 <= 00000014

$11 <= 00006028

$14 <= 00000014

$15 <= 00000018

**Ori\_E\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $t0, 12

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 0000000c

**Ori\_M\_RS(ori)**

ori $t0, $zero, 8

ori $t2, $zero, 20

ori $t1, $t0, 12

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 00000014

260@00003008: $ 9 <= 0000000c

**Ori\_W\_RS(ori)**

ori $t0, $zero, 8

ori $t2, $zero, 20

ori $t3, $zero, 24

ori $t1, $t0, 12

180@00003000: $ 8 <= 00000008

220@00003004: $10 <= 00000014

260@00003008: $11 <= 00000018

300@0000300c: $ 9 <= 0000000c

**Subu\_E\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 24

ori $t3, $zero, 12

ori $t4, $zero, 16

subu $t5, $t0, $t1

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000018

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

380@00003014: $13 <= fffffff4

420@00003018: $14 <= fffffffd

**Subu\_M\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 24

ori $t3, $zero, 12

ori $t4, $zero, 16

subu $t5, $t0, $t1

ori $t7, $zero, 20

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000018

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

380@00003014: $13 <= fffffff4

420@00003018: $15 <= 00000014

460@0000301c: $14 <= fffffffd

**Subu\_W\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 24

ori $t3, $zero, 12

ori $t4, $zero, 16

subu $t5, $t0, $t1

ori $t7, $zero, 20

ori $t8, $zero, 24

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000018

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

380@00003014: $13 <= fffffff4

420@00003018: $15 <= 00000014

460@0000301c: $24 <= 00000018

500@00003020: $14 <= fffffffd

**Ld\_E\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 4

ori $t3, $zero, 12

ori $t4, $zero, 16

sw $t0, 0($t1)

lw $t5, 0($t1)

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000004

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

340@00003014: \*00000014 <= 00000008

420@00003018: $13 <= 00000008

500@0000301c: $14 <= 0000000d

**Ld\_M\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 4

ori $t3, $zero, 12

ori $t4, $zero, 16

sw $t0, 0($t1)

lw $t5, 0($t1)

ori $t7, $zero, 20

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000004

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

340@00003014: \*00000014 <= 00000008

420@00003018: $13 <= 00000008

460@0000301c: $15 <= 00000014

500@00003020: $14 <= 0000000d

**Ld\_W\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

ori $t2, $zero, 4

ori $t3, $zero, 12

ori $t4, $zero, 16

sw $t0, 0($t1)

lw $t5, 0($t1)

ori $t7, $zero, 20

ori $t8, $zero, 24

ori $t6, $t5, 13

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $10 <= 00000004

300@0000300c: $11 <= 0000000c

340@00003010: $12 <= 00000010

340@00003014: \*00000014 <= 00000008

420@00003018: $13 <= 00000008

460@0000301c: $15 <= 00000014

500@00003020: $24 <= 00000018

540@00003024: $14 <= 0000000d

**Jal\_E\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

jal change1

ori $t2, $ra, 8

ori $t3, $zero, 14

change1:

ori $t4, $zero, 18

ori $t5, $zero, 22

ori $t6, $zero, 26

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $31 <= 00003010

300@0000300c: $10 <= 00003018

340@00003014: $12 <= 00000012

380@00003018: $13 <= 00000016

420@0000301c: $14 <= 0000001a

**Jal\_M\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

jal change1

ori $t2, $zero, 4

ori $t3, $zero, 14

change1:

ori $t4, $ra, 18

ori $t5, $zero, 22

ori $t6, $zero, 26

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $31 <= 00003010

300@0000300c: $10 <= 00000004

340@00003014: $12 <= 00003012

380@00003018: $13 <= 00000016

420@0000301c: $14 <= 0000001a

**Jal\_W\_RS(ori)**

ori $t0, $zero, 8

ori $t1, $zero, 20

jal change1

ori $t2, $zero, 4

ori $t3, $zero, 14

change1:

ori $t7, $zero, 6

ori $t4, $ra, 18

ori $t5, $zero, 22

ori $t6, $zero, 26

180@00003000: $ 8 <= 00000008

220@00003004: $ 9 <= 00000014

260@00003008: $31 <= 00003010

300@0000300c: $10 <= 00000004

340@00003014: $15 <= 00000006

380@00003018: $12 <= 00003012

420@0000301c: $13 <= 00000016

460@00003020: $14 <= 0000001a

**Lw:**

**R\_E/M/W\_RS(lw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 9

ori $t3, $zero, 12

sw $t1, 0($t0)

sw $t2, 4($t0)

sw $t3, 8($t0)

occasion1: #R\_E\_RS

subu $t4, $t1, $t0

lw $t5, 0($t4)

occasion2: #R\_M\_RS

subu $t5, $t3, $t0

ori $zero, $zero, 5

lw $t6, 0($t5)

occasion3: #R\_W\_RS

addu $t6, $t0, $t1

ori $s0, $zero, 12

ori $s1, $zero, 16

lw $t7, 0($t6)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00000009

$11 <= 0000000c

\*00000004 <= 00000008

\*00000008 <= 00000009

\*0000000c <= 0000000c

$12 <= 00000004

$13 <= 00000008

$13 <= 00000008

$14 <= 00000009

$14 <= 0000000c

$16 <= 0000000c

$17 <= 00000010

$15 <= 0000000c

**I\_E/M/W\_RS(lw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 9

ori $t3, $zero, 12

sw $t1, 0($t0)

sw $t2, 4($t0)

sw $t3, 8($t0)

occasion1: #R\_E\_RS

ori $t4, $zero, 4

lw $t5, 0($t4)

occasion2: #R\_M\_RS

ori $t5, $zero, 8

ori $zero, $zero, 5

lw $t6, 0($t5)

occasion3: #R\_W\_RS

ori $t6, $zero, 12

ori $s0, $zero, 12

ori $s1, $zero, 16

lw $t7, 0($t6)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00000009

$11 <= 0000000c

\*00000004 <= 00000008

\*00000008 <= 00000009

\*0000000c <= 0000000c

$12 <= 00000004

$13 <= 00000008

$13 <= 00000008

$14 <= 00000009

$14 <= 0000000c

$16 <= 0000000c

$17 <= 00000010

$15 <= 0000000c

**Ld\_E/M/W\_RS(lw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

sw $t0, 0($zero)

sw $t1, 0($t0)

sw $t2, 0($t1)

sw $t3, 4($t1)

occasion1: #ld\_E\_RS

lw $t4, 0($t0)

lw $t5, 0($t4)

occasion2: #ld\_M\_RS

lw $t5, -4($t0)

addu $zero, $zero, $t1

lw $t6, 0($t5)

occasion: #ld\_W\_RS

lw $t6, 4($t0)

ori $s0, $zero, 1

ori $s1, $zero, 2

lw $t7, 0($t6)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

\*00000000 <= 00000004

\*00000004 <= 00000008

\*00000008 <= 0000000c

\*0000000c <= 00000010

$12 <= 00000008

$13 <= 0000000c

$13 <= 00000004

$14 <= 00000008

$14 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$15 <= 00000010

由于DM的容量只有4KB，因此31号寄存器中的值不可能作为lw指令中的寻址

**Sw:**

**R\_E/M/W\_RS/RT(sw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

ori $t4, $zero, 20

occasion1: #R\_E\_RS

addu $t5, $t0, $t1

sw $t2, 0($t5)

occasion2: #R\_M\_RS

subu $t6, $t2, $t1

ori $s0, $zero, 12

sw $t3, 4($t6)

occasion3: #R\_W\_RS

subu $t7, $t4, $t0

ori $s1, $zero, 4

ori $s2, $zero, 8

sw $t4, 0($t7)

occasion4: #R\_E\_RT

subu $t5, $t1, $t0

sw $t5, 0($t0)

occasion5: #R\_M\_RT

subu $t5, $t2, $t0

ori $s0, $zero, 2

sw $t5, 4($t2)

occasion6: #R\_W\_RT

addu $t6, $t3, $t4

ori $s0, $zero, 1

ori $s1, $zero, 2

sw $t6, 0($t1)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= 00000014

$13 <= 0000000c

\*0000000c <= 0000000c

$14 <= 00000004

$16 <= 0000000c

\*00000008 <= 00000010

$15 <= 00000010

$17 <= 00000004

$18 <= 00000008

\*00000010 <= 00000014

$13 <= 00000004

\*00000004 <= 00000004

$13 <= 00000008

$16 <= 00000002

\*00000010 <= 00000008

$14 <= 00000024

$16 <= 00000001

$17 <= 00000002

\*00000008 <= 00000024

**I\_E/M/W\_RS/RT(sw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

ori $t4, $zero, 20

occasion1: #I\_E\_RS

ori $t5, $zero, 12

sw $t2, 0($t5)

occasion2: #I\_M\_RS

ori $t6, $zero, 4

ori $s0, $zero, 12

sw $t3, 4($t6)

occasion3: #I\_W\_RS

ori $t7, $zero, 16

ori $s1, $zero, 4

ori $s2, $zero, 8

sw $t4, 0($t7)

occasion4: #I\_E\_RT

lui $t5, 3

sw $t5, 0($t0)

occasion5: #I\_M\_RT

lui $t5, 1

ori $s0, $zero, 2

sw $t5, 4($t2)

occasion6: #I\_W\_RT

ori $zero, $zero, 9

ori $s0, $zero, 1

ori $s1, $zero, 2

sw $zero, 0($t1)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= 00000014

$13 <= 0000000c

\*0000000c <= 0000000c

$14 <= 00000004

$16 <= 0000000c

\*00000008 <= 00000010

$15 <= 00000010

$17 <= 00000004

$18 <= 00000008

\*00000010 <= 00000014

$13 <= 00030000

\*00000004 <= 00030000

$13 <= 00010000

$16 <= 00000002

\*00000010 <= 00010000

$16 <= 00000001

$17 <= 00000002

\*00000008 <= 00000000

**Ld\_E/M/W\_RS\_RT(sw)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

ori $t4, $zero, 20

sw $t0, 0($zero)

sw $t1, 0($t0)

sw $t2, 4($t0)

sw $t3, 8($t0)

occasion1: #ld\_E\_RS

lw $t5, 0($t0)

sw $t2, 0($t5)

occasion2: #ld\_M\_RS

lw $t6, 0($t0)

ori $s0, $zero, 12

sw $t3, 4($t6)

occasion3: #ld\_W\_RS

lw $t7, 4($t0)

ori $s1, $zero, 4

ori $s2, $zero, 8

sw $t4, 0($t7)

occasion4: #ld\_E\_RT

lw $t5, 0($t2)

sw $t5, 0($t0)

occasion5: #ld\_M\_RT

lw $t5, 4($t2)

ori $s0, $zero, 2

sw $t5, 4($t2)

occasion6: #ld\_W\_RT

lw $t6, 4($t0)

ori $s0, $zero, 1

ori $s1, $zero, 2

sw $t6, 0($t1)

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= 00000014

\*00000000 <= 00000004

\*00000004 <= 00000008

\*00000008 <= 0000000c

\*0000000c <= 00000010

$13 <= 00000008

\*00000008 <= 0000000c

$14 <= 00000008

$16 <= 0000000c

\*0000000c <= 00000010

$15 <= 0000000c

$17 <= 00000004

$18 <= 00000008

\*0000000c <= 00000014

$13 <= 00000014

\*00000004 <= 00000014

$13 <= 00000000

$16 <= 00000002

\*00000010 <= 00000000

$14 <= 0000000c

$16 <= 00000001

$17 <= 00000002

\*00000008 <= 0000000c

**(jal\_sw的情况)**

**Beq:**

**R\_E/M/W\_RS/RT(beq)(equal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

subu $t4, $t0, $t1

occasion1: #R\_E\_RS

addu $t5, $t0, $t1

beq $t5, $t2, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion2: #R\_M\_RS

addu $t6, $t0, $t1

ori $s0, $zero, 1

beq $t6, $t2, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion3: #R\_W\_RS

addu $t7, $t0, $t1

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t7, $t2, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion4: #R\_E\_RT

subu $t5, $t1, $t2

beq $t4, $t5, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion5: #R\_M\_RT

subu $t6, $t1, $t2

ori $s0, $zero, 1

beq $t4, $t6, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion6: #R\_W\_RT

subu $t7, $t1, $t2

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t4, $t7, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 3

ori $s3, $zero, 4

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= fffffffc

$13 <= 0000000c

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$14 <= 0000000c

$16 <= 00000001

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$15 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$13 <= fffffffc

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$14 <= fffffffc

$16 <= 00000001

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$15 <= fffffffc

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

**R\_E/M/W\_RS/RT(beq)(unequal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

subu $t4, $t0, $t1

occasion1: #R\_E\_RS

addu $t5, $t0, $t1

beq $t5, $t3, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion2: #R\_M\_RS

addu $t6, $t0, $t1

ori $s0, $zero, 1

beq $t6, $t3, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion3: #R\_W\_RS

addu $t7, $t0, $t1

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t7, $t3, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion4: #R\_E\_RT

subu $t5, $t1, $t2

beq $t3, $t5, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion5: #R\_M\_RT

subu $t6, $t1, $t2

ori $s0, $zero, 1

beq $t3, $t6, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion6: #R\_W\_RT

subu $t7, $t1, $t2

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t3, $t7, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 3

ori $s3, $zero, 4

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= fffffffc

$13 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$14 <= 0000000c

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$15 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$13 <= fffffffc

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$14 <= fffffffc

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$15 <= fffffffc

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

**I\_E/M/W\_RS/RT(beq)(equal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

subu $t4, $t0, $t1

occasion1: #I\_E\_RS

ori $t5, $zero, 16

beq $t5, $t3, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion2: #I\_M\_RS

ori $t6, $zero, 16

ori $s0, $zero, 1

beq $t6, $t3, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion3: #I\_W\_RS

ori $t7, $zero, 16

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t7, $t3, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion4: #I\_E\_RT

ori $t5, $zero, 8

beq $t1, $t5, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion5: #I\_M\_RT

ori $t6, $zero, 8

ori $s0, $zero, 1

beq $t1, $t6, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion6: #I\_W\_RT

ori $t7, $zero, 8

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t1, $t7, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 3

ori $s3, $zero, 4

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= fffffffc

$13 <= 00000010

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$14 <= 00000010

$16 <= 00000001

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$15 <= 00000010

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$13 <= 00000008

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$14 <= 00000008

$16 <= 00000001

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

$15 <= 00000008

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$18 <= 00000003

$19 <= 00000004

**I\_E/M/W\_RS/RT(beq)(unequal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $t3, $zero, 16

subu $t4, $t0, $t1

occasion1: #I\_E\_RS

ori $t5, $zero, 16

beq $t5, $t2, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion2: #I\_M\_RS

ori $t6, $zero, 16

ori $s0, $zero, 1

beq $t6, $t2, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion3: #I\_W\_RS

ori $t7, $zero, 16

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t7, $t2, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion4: #I\_E\_RT

ori $t5, $zero, 8

beq $t2, $t5, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion5: #I\_M\_RT

ori $t6, $zero, 8

ori $s0, $zero, 1

beq $t2, $t6, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 3

ori $s3, $zero, 4

occasion6: #I\_W\_RT

ori $t7, $zero, 8

ori $s0, $zero, 1

ori $s1, $zero, 2

beq $t2, $t7, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 3

ori $s3, $zero, 4

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$11 <= 00000010

$12 <= fffffffc

$13 <= 00000010

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$14 <= 00000010

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$15 <= 00000010

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$13 <= 00000008

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$14 <= 00000008

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

$15 <= 00000008

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$19 <= 00000004

**Ld\_E/M/W\_RS/RT(beq)(equal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $s0, $zero, 1

sw $t0, 0($zero)

sw $t1, 4($zero)

sw $t2, 8($zero)

ori $s0, $zero, 1

ori $s1, $zero, 2

occasion1: #ld\_E\_RS

lw $t3, 0($t0)

beq $t3, $t1, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion2: #ld\_M\_RS

lw $t4, 0($t0)

ori $s2, $zero, 2

beq $t4, $t1, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion3: #ld\_W\_RS

lw $t5, 0($t0)

ori $s2, $zero, 2

ori $s3, $zero, 3

beq $t5, $t1, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion4: #ld\_E\_RT

lw $t6, 0($t0)

beq $t6, $t1, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion5: #ld\_M\_RT

lw $t7, 0($t0)

ori $s2, $zero, 2

beq $t7, $t1, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion6: #ld\_W\_RT

lw $t8, 0($t0)

ori $s2, $zero, 2

ori $s3, $zero, 3

beq $t8, $t1, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 2

ori $s3, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$16 <= 00000001

\*00000000 <= 00000004

\*00000004 <= 00000008

\*00000008 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$11 <= 00000008

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

$12 <= 00000008

$18 <= 00000002

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

$13 <= 00000008

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

$14 <= 00000008

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

$15 <= 00000008

$18 <= 00000002

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

$24 <= 00000008

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$18 <= 00000002

$19 <= 00000003

**Ld\_E/M/W\_RS/RT(beq)(unequal)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 12

ori $s0, $zero, 1

sw $t0, 0($zero)

sw $t1, 4($zero)

sw $t2, 8($zero)

ori $s0, $zero, 1

ori $s1, $zero, 2

occasion1: #ld\_E\_RS

lw $t3, 0($t0)

beq $t3, $t0, change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion2: #ld\_M\_RS

lw $t4, 0($t0)

ori $s2, $zero, 2

beq $t4, $t0, change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion3: #ld\_W\_RS

lw $t5, 0($t0)

ori $s2, $zero, 2

ori $s3, $zero, 3

beq $t5, $t0, change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion4: #ld\_E\_RT

lw $t6, 0($t0)

beq $t0, $t6, change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion5: #ld\_M\_RT

lw $t7, 0($t0)

ori $s2, $zero, 2

beq $t0, $t7, change5

ori $s0, $zero, 1

ori $s1, $zero, 2

change5:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion6: #ld\_W\_RT

lw $t8, 0($t0)

ori $s2, $zero, 2

ori $s3, $zero, 3

beq $t0, $t8, change6

ori $s0, $zero, 1

ori $s1, $zero, 2

change6:

ori $s2, $zero, 2

ori $s3, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 0000000c

$16 <= 00000001

\*00000000 <= 00000004

\*00000004 <= 00000008

\*00000008 <= 0000000c

$16 <= 00000001

$17 <= 00000002

$11 <= 00000008

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$12 <= 00000008

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$13 <= 00000008

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$14 <= 00000008

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$15 <= 00000008

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$24 <= 00000008

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

**Jal\_M/W\_RS/RT(beq)(equal)**

ori $t0, $zero, 4

ori $t1, $zero, 0x00003014

ori $t2, $zero, 0x00003034

ori $t3, $zero, 0x00003058

ori $t4, $zero, 0x00003078

occasion1: #jal\_M\_RS

jal change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

beq $ra, $t1, change11

ori $s0, $zero, 1

ori $s1, $zero, 2

change11:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion2: #jal\_W\_RS

jal change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 2

beq $ra, $t2, change21

ori $s0, $zero, 1

ori $s1, $zero, 2

change21:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion3: #jal\_M\_RT

jal change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

beq $t3, $ra, change31

ori $s0, $zero, 1

ori $s1, $zero, 2

change31:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion4: #jal\_W\_RT

jal change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 2

beq $t4, $ra, change41

ori $s0, $zero, 1

ori $s1, $zero, 2

change41:

ori $s2, $zero, 2

ori $s3, $zero, 3

$ 8 <= 00000004

$ 9 <= 00003014

$10 <= 00003034

$11 <= 00003058

$12 <= 00003078

$31 <= 0000301c

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 0000303c

$16 <= 00000001

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 00003060

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 00003080

$16 <= 00000001

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

**Jal\_M/W\_RS/RT(beq)(unequal)**

ori $t0, $zero, 4

ori $t1, $zero, 0x00003010

ori $t2, $zero, 0x00003030

ori $t3, $zero, 0x00003050

ori $t4, $zero, 0x00003070

occasion1: #jal\_M\_RS

jal change1

ori $s0, $zero, 1

ori $s1, $zero, 2

change1:

beq $ra, $t1, change11

ori $s0, $zero, 1

ori $s1, $zero, 2

change11:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion2: #jal\_W\_RS

jal change2

ori $s0, $zero, 1

ori $s1, $zero, 2

change2:

ori $s2, $zero, 2

beq $ra, $t2, change21

ori $s0, $zero, 1

ori $s1, $zero, 2

change21:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion3: #jal\_M\_RT

jal change3

ori $s0, $zero, 1

ori $s1, $zero, 2

change3:

beq $t3, $ra, change31

ori $s0, $zero, 1

ori $s1, $zero, 2

change31:

ori $s2, $zero, 2

ori $s3, $zero, 3

occasion4: #jal\_W\_RT

jal change4

ori $s0, $zero, 1

ori $s1, $zero, 2

change4:

ori $s2, $zero, 2

beq $t4, $ra, change41

ori $s0, $zero, 1

ori $s1, $zero, 2

change41:

ori $s2, $zero, 2

ori $s3, $zero, 3

$ 8 <= 00000004

$ 9 <= 00003010

$10 <= 00003030

$11 <= 00003050

$12 <= 00003070

$31 <= 0000301c

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 0000303c

$16 <= 00000001

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 00003060

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$31 <= 00003080

$16 <= 00000001

$18 <= 00000002

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

**Jal\_M\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 0x00003010

ori $t2, $zero, 0x00003030

ori $t3, $zero, 0x00003050

ori $t4, $zero, 0x00003070

occasion1: #jal\_M\_RS

jal change1

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 2

ori $s3, $zero, 3

change1:

jr $ra

ori $s0, $zero, 1

ori $s1, $zero, 2

$ 8 <= 00000004

$ 9 <= 00003010

$10 <= 00003030

$11 <= 00003050

$12 <= 00003070

$31 <= 0000301c

$16 <= 00000001

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$16 <= 00000001

**Jal\_W\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 0x00003010

ori $t2, $zero, 0x00003030

ori $t3, $zero, 0x00003050

ori $t4, $zero, 0x00003070

occasion1: #jal\_W\_RS

jal change1

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 2

ori $s3, $zero, 3

change1:

ori $s3, $zero, 3

jr $ra

ori $s0, $zero, 1

ori $s1, $zero, 2

$ 8 <= 00000004

$ 9 <= 00003010

$10 <= 00003030

$11 <= 00003050

$12 <= 00003070

$31 <= 0000301c

$16 <= 00000001

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$19 <= 00000003

$16 <= 00000001

$17 <= 00000002

$18 <= 00000002

$19 <= 00000003

$19 <= 00000003

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_E\_RS

addu $t3, $t0, $t2

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$ 9 <= 00000008

$10 <= 00003000

**R\_M\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_M\_RS

addu $t3, $t0, $t2

ori $s0, $zero, 1

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$16 <= 00000001

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$16 <= 00000001

$ 9 <= 00000008

$10 <= 00003000

**R\_W\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_W\_RS

addu $t3, $t0, $t2

ori $s0, $zero, 1

ori $s1, $zero, 2

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003004

$16 <= 00000001

$17 <= 00000002

$16 <= 00000001

**I\_E\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_E\_RS

ori $t3, $t2, 0

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003000

$16 <= 00000001

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003000

$16 <= 00000001

$ 8 <= 00000004

**I\_M\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_M\_RS

ori $t3, $t2, 0

ori $s0, $zero, 1

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003000

$16 <= 00000001

$16 <= 00000001

$ 8 <= 00000004

$ 9 <= 00000008

$10 <= 00003000

$16 <= 00000001

$17 <= 00000002

$18 <= 00000003

$11 <= 00003000

$16 <= 00000001

$16 <= 00000001

$ 8 <= 00000004

**I\_W\_RS(jr)**

ori $t0, $zero, 4

ori $t1, $zero, 8

ori $t2, $zero, 0x00003000

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

occasion1: #R\_W\_RS

ori $t3, $t2, 0

ori $s0, $zero, 1

ori $s1, $zero, 2

jr $t3

ori $s0, $zero, 1

ori $s1, $zero, 2

ori $s2, $zero, 3

$ 8 <= 00000004

$ 9 <= 00000008

乘除测试代码：

lui $t0, 0x7000

lui $t1, 0xc000

ori $t0, $t0, 0x1234

ori $t1, $t1, 0x1234

mult $t0, $t1

mfhi $t2

mflo $t3

multu $t0, $t1

mfhi $t2

mflo $t3

div $t1, $t0

mfhi $t2

mflo $t3

divu $t1, $t0

mfhi $t2

mflo $t3

mult $t0, $t1

mtlo $t2

mthi $t3

mfhi $t2

mflo $t3

预期输出：

45@00003000: $ 8 <= 70000000

55@00003004: $ 9 <= c0000000

65@00003008: $ 8 <= 70001234

75@0000300c: $ 9 <= c0001234

145@00003014: $10 <= e4000369

155@00003018: $11 <= c14b5a90

225@00003020: $10 <= 5400159d

235@00003024: $11 <= c14b5a90

355@0000302c: $10 <= c0001234

365@00003030: $11 <= 00000000

485@00003038: $10 <= 50000000

495@0000303c: $11 <= 00000001

585@0000304c: $10 <= 00000001

595@00003050: $11 <= 50000000

测试程序：

# R test

li $1, 100

li $2, 60

add $3, $1, $2

sw $3, -4($3)

sw $3,0($0)

li $3, 12

sb $3, -2($3)

li $1, -1

li $2, -3

add $3, $2, $1

li $2, 0x80000000

lui $1, 0x8000

addu $3, $1, $2

sw $1, 4($3)

sh $2, 2($0)

#add $3, $1, $2

lui $1, 0x7700

lui $2, 0x7001

addu $3, $1, $2

#add $3, $1, $2

li $1, 100

li $2, 150

sub $3, $1, $2

addu $3, $3, $1

sh $3,2($3)

lui $1, 0x7fff

lui $1, 0x8000

subu $3, $1, $2

#sub $3, $1, $2

li $1, 100

li $2, 50

mult $1, $2

mfhi $3

mflo $4

sb $2, -19($4)

sb $2, 7($3)

lui $1, 0x8000

lui $2, 100

mult $1, $2

mfhi $3

mflo $4

lui $1, 0x8000

lui $2, 0x8000

mult $1, $2

mfhi $3

mflo $4

lui $1, 100

lui $2, 0x8000

mult $1, $2

mfhi $3

mflo $4

li $1, 100

li $2, 50

multu $1, $2

mfhi $3

mflo $4

lui $1, 0x8000

lui $2, 100

multu $1, $2

mfhi $3

mflo $4

sw $3, 0($0)

sb $3, 100($0)

lui $1, 0x8000

lui $2, 0x8000

multu $1, $2

mfhi $3

li $5, 200

sw $3, 0($5)

mflo $4

lui $1, 100

lui $2, 0x8000

multu $1, $2

mfhi $3

mflo $4

li $1, -8

li $2, -4

div $1, $2

mfhi $3

li $2, 64

sw $3, -16($2)

mflo $4

li $1, 16

li $2, -6

div $1, $2

mfhi $3

li $3, 100

mflo $4

sh $4, 2($3)

li $1, 16

li $2, 5

div $1, $2

mfhi $3

li $5, 100

mflo $4

sh $4, 8($5)

li $1, -16

li $2, 4

div $1, $2

mfhi $3

mflo $4

li $1, -8

li $2, -4

divu $1, $2

mfhi $3

mflo $4

li $1, 16

li $2, -6

divu $1, $2

mfhi $3

li $5, 100

mflo $4

sh $4, 8($5)

li $1, 16

li $2, 5

divu $1, $2

mfhi $3

mflo $4

li $1, -16

li $2, 4

divu $1, $2

mfhi $3

mflo $4

li $1, 80

li $2, 70

slt $3, $1, $2

slt $4, $2, $1

li $1, -10

li $2, 100

slt $3, $1, $2

slt $3, $2, $1

li $1, -100

li $2, -50

slt $3, $1, $2

slt $3, $2, $1

li $1, -20

li $2, 100

sltu $3, $1, $2 # -20<100

sltu $3, $2, $1 #

sltu $3, $1, $0

sltu $3, $0, $1

li $1, 0

sltu $3, $1, $1

sltu $3, $1, $0

li $1, 100

sltu $3, $1, $0

li $9, 128

sll $10, $9, 3

lui $12, 0xff00

srl $13, $12, 2

lui $5, 0xff00

sra $6, $5, 2

li $8, 12

li $5, 4

srav $6, $8, $5

li $9, 13

srlv $1, $9, $8

li $10, 6

li $5, 3

sllv $3, $10, $5

li $1, 0x1008

li $2, 0xdff7

xor $3, $1, $2

xori $3, $0, 0x8fff

li $1, 0xffff

li $2, 0x1000

or $2, $1, $2

li $1, 100

li $2, 50

sw $2, 0($1)

li $1, 50

li $2, 100

sw $1, 4($2)

预期结果：

45@00003000: $ 1 <= 00000064

55@00003004: $ 2 <= 0000003c

65@00003008: $ 3 <= 000000a0

65@0000300c: \*0000009c <= 000000a0

75@00003010: \*00000000 <= 000000a0

95@00003014: $ 3 <= 0000000c

95@00003018: \*00000008 <= 000c0000

115@0000301c: $ 1 <= ffffffff

125@00003020: $ 2 <= fffffffd

135@00003024: $ 3 <= fffffffc

145@00003028: $ 1 <= 80000000

155@0000302c: $ 2 <= 80000000

165@00003030: $ 1 <= 80000000

175@00003034: $ 3 <= 00000000

175@00003038: \*00000004 <= 80000000

185@0000303c: \*00000000 <= 000000a0

205@00003040: $ 1 <= 77000000

215@00003044: $ 2 <= 70010000

225@00003048: $ 3 <= e7010000

235@0000304c: $ 1 <= 00000064

245@00003050: $ 2 <= 00000096

255@00003054: $ 3 <= ffffffce

265@00003058: $ 3 <= 00000032

265@0000305c: \*00000034 <= 00000032

285@00003060: $ 1 <= 7fff0000

295@00003064: $ 1 <= 80000000

305@00003068: $ 3 <= 7fffff6a

315@0000306c: $ 1 <= 00000064

325@00003070: $ 2 <= 00000032

395@00003078: $ 3 <= 00000000

405@0000307c: $ 4 <= 00001388

405@00003080: \*00001374 <= 00003200

415@00003084: \*00000004 <= 32000000

435@00003088: $ 1 <= 80000000

445@0000308c: $ 2 <= 00640000

515@00003094: $ 3 <= ffce0000

525@00003098: $ 4 <= 00000000

535@0000309c: $ 1 <= 80000000

545@000030a0: $ 2 <= 80000000

615@000030a8: $ 3 <= 40000000

625@000030ac: $ 4 <= 00000000

635@000030b0: $ 1 <= 00640000

645@000030b4: $ 2 <= 80000000

715@000030bc: $ 3 <= ffce0000

725@000030c0: $ 4 <= 00000000

735@000030c4: $ 1 <= 00000064

745@000030c8: $ 2 <= 00000032

815@000030d0: $ 3 <= 00000000

825@000030d4: $ 4 <= 00001388

835@000030d8: $ 1 <= 80000000

845@000030dc: $ 2 <= 00640000

915@000030e4: $ 3 <= 00320000

925@000030e8: $ 4 <= 00000000

925@000030ec: \*00000000 <= 00320000

935@000030f0: \*00000064 <= 00000000

955@000030f4: $ 1 <= 80000000

965@000030f8: $ 2 <= 80000000

1035@00003100: $ 3 <= 40000000

1045@00003104: $ 5 <= 000000c8

1045@00003108: \*000000c8 <= 40000000

1065@0000310c: $ 4 <= 00000000

1075@00003110: $ 1 <= 00640000

1085@00003114: $ 2 <= 80000000

1155@0000311c: $ 3 <= 00320000

1165@00003120: $ 4 <= 00000000

1175@00003124: $ 1 <= fffffff8

1185@00003128: $ 2 <= fffffffc

1305@00003130: $ 3 <= 00000000

1315@00003134: $ 2 <= 00000040

1315@00003138: \*00000030 <= 00000000

1335@0000313c: $ 4 <= 00000002

1345@00003140: $ 1 <= 00000010

1355@00003144: $ 2 <= fffffffa

1475@0000314c: $ 3 <= 00000004

1485@00003150: $ 3 <= 00000064

1495@00003154: $ 4 <= fffffffe

1495@00003158: \*00000064 <= fffe0000

1515@0000315c: $ 1 <= 00000010

1525@00003160: $ 2 <= 00000005

1645@00003168: $ 3 <= 00000001

1655@0000316c: $ 5 <= 00000064

1665@00003170: $ 4 <= 00000003

1665@00003174: \*0000006c <= 00000003

1685@00003178: $ 1 <= fffffff0

1695@0000317c: $ 2 <= 00000004

1815@00003184: $ 3 <= 00000000

1825@00003188: $ 4 <= fffffffc

1835@0000318c: $ 1 <= fffffff8

1845@00003190: $ 2 <= fffffffc

1965@00003198: $ 3 <= fffffff8

1975@0000319c: $ 4 <= 00000000

1985@000031a0: $ 1 <= 00000010

1995@000031a4: $ 2 <= fffffffa

2115@000031ac: $ 3 <= 00000010

2125@000031b0: $ 5 <= 00000064

2135@000031b4: $ 4 <= 00000000

2135@000031b8: \*0000006c <= 00000000

2155@000031bc: $ 1 <= 00000010

2165@000031c0: $ 2 <= 00000005

2285@000031c8: $ 3 <= 00000001

2295@000031cc: $ 4 <= 00000003

2305@000031d0: $ 1 <= fffffff0

2315@000031d4: $ 2 <= 00000004

2435@000031dc: $ 3 <= 00000000

2445@000031e0: $ 4 <= 3ffffffc

2455@000031e4: $ 1 <= 00000050

2465@000031e8: $ 2 <= 00000046

2475@000031ec: $ 3 <= 00000000

2485@000031f0: $ 4 <= 00000001

2495@000031f4: $ 1 <= fffffff6

2505@000031f8: $ 2 <= 00000064

2515@000031fc: $ 3 <= 00000001

2525@00003200: $ 3 <= 00000000

2535@00003204: $ 1 <= ffffff9c

2545@00003208: $ 2 <= ffffffce

2555@0000320c: $ 3 <= 00000001

2565@00003210: $ 3 <= 00000000

2575@00003214: $ 1 <= ffffffec

2585@00003218: $ 2 <= 00000064

2595@0000321c: $ 3 <= 00000000

2605@00003220: $ 3 <= 00000001

2615@00003224: $ 3 <= 00000000

2625@00003228: $ 3 <= 00000001

2635@0000322c: $ 1 <= 00000000

2645@00003230: $ 3 <= 00000000

2655@00003234: $ 3 <= 00000000

2665@00003238: $ 1 <= 00000064

2675@0000323c: $ 3 <= 00000000

2685@00003240: $ 9 <= 00000080

2695@00003244: $10 <= 00000400

2705@00003248: $12 <= ff000000

2715@0000324c: $13 <= 3fc00000

2725@00003250: $ 5 <= ff000000

2735@00003254: $ 6 <= ffc00000

2745@00003258: $ 8 <= 0000000c

2755@0000325c: $ 5 <= 00000004

2765@00003260: $ 6 <= 00000000

2775@00003264: $ 9 <= 0000000d

2785@00003268: $ 1 <= 00000000

2795@0000326c: $10 <= 00000006

2805@00003270: $ 5 <= 00000003

2815@00003274: $ 3 <= 00000030

2825@00003278: $ 1 <= 00001008

2835@0000327c: $ 2 <= 0000dff7

2845@00003280: $ 3 <= 0000cfff

2855@00003284: $ 3 <= 00008fff

2865@00003288: $ 1 <= 0000ffff

2875@0000328c: $ 2 <= 00001000

2885@00003290: $ 2 <= 0000ffff

2895@00003294: $ 1 <= 00000064

2905@00003298: $ 2 <= 00000032

2905@0000329c: \*00000064 <= 00000032

2925@000032a0: $ 1 <= 00000032

2935@000032a4: $ 2 <= 00000064

2935@000032a8: \*00000068 <= 00000032

测试程序：  
  
ori $t0, $0, 0x1234  
mthi $t0  
mtlo $t0  
mfhi $t1  
mflo $t2

预期结果：

45@00003000: $ 8 <= 00001234

75@0000300c: $ 9 <= 00001234

85@00003010: $10 <= 00001234

测试程序：

.text

ori $t1, $0, 3

jal lable1

nop

add $t1, $t1, $t1

ori $t2, $0, 0x3024

jr $t2

lable1:

ori $t0, $0, 2

jalr $a0, $31

nop

nop

预期结果：

45@00003000: $ 9 <= 00000003

55@00003004: $31 <= 0000300c

75@00003018: $ 8 <= 00000002

85@0000301c: $31 <= 00003024

105@0000300c: $ 9 <= 00000006

115@00003010: $10 <= 00003024

145@00003018: $ 8 <= 00000002

测试程序：

ori $t1, $0, 3

jal lable1

nop

add $t1, $t1, $t1

ori $t2, $0, 0x3024

jr $t2

lable1:

ori $t0, $0, 2

jalr $31, $31

nop

Nop

预期结果：

45@00003000: $ 9 <= 00000003

55@00003004: $31 <= 0000300c

75@00003018: $ 8 <= 00000002

85@0000301c: $31 <= 00003024

105@0000300c: $ 9 <= 00000006

115@00003010: $10 <= 00003024

145@00003018: $ 8 <= 00000002