Verilog流水线CPU设计文档

**一.数据通路设计（分布式）**

1.F级流水线

(1).PC(当前地址)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | NPC | Input | [31:0] | 要存到寄存器里的下一个PC的值 |
| 2 | CLK | Input | 1 | CLK时钟信号 |
| 3 | Reset | Input | 1 | Reset信号为1时，PC的值置0 |
| 4 | Stall | Input | 1 | 暂停信号 |
| 5 | IMaddr | Output | [11:2] | IM指令的地址 |
| 6 | PC | Output | [31:0] | 当前PC的值 |

(2).IM（指令存储器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IM\_in | Input | [9:0] | 取出ROM中地址为IM\_in的指令 |
| 2 | IM\_out | Output | [31:0] | 输出32位mips指令值 |

2.D级流水线

(1).F\_D（D级流水寄存器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IMcode\_F | Input | [31:0] | F级的32位指令码 |
| 2 | CLK | Input | 1 | CLK时钟信号 |
| 3 | Reset | Input | 1 | Reset信号为1时，寄存器 |
| 4 | Stall | Input | 1 | 暂停信号 |
| 5 | PC\_F | Input | [31:0] | F级的PC值 |
| 6 | IMcode\_D | Output | [31:0] | D级的32位指令码 |
| 7 | PC\_D | Output | [31:0] | D级的PC值 |

(2).GRF(同用寄存器堆)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | A1 | Input | [25:21] | 读出编号为A1的寄存器的值到输出端口RD1 |
| 2 | A2 | Input | [20:16] | 读出编号为A2的寄存器的值到输出端口RD2 |
| 3 | A3 | Input | [4:0] | 写入寄存器编号 |
| 4 | WD\_RF | Input | [31:0] | 写入寄存器的值 |
| 5 | PC | Input | [31:0] | 当前（D级）PC值 |
| 6 | CLK | Input | 1 | 时钟信号 |
| 7 | Reset | Input | 1 | Reset信号为1时，寄存器堆中寄存器的值置0 |
| 8 | RegWrite | Input | 1 | 为1时可以把值写入寄存器 |
| 9 | RD1 | Output | [31:0] | 输出端口RD1 |
| 10 | RD2 | Output | [31:0] | 输出端口RD2 |

（3）.CMP（比较器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | RF\_RD1 | Input | [31:0] | 转发后的GRF[rs]的值 |
| 2 | RF\_RD2 | Input | [31:0] | 转发后的GRF[rt]的值 |
| 3 | IF\_Equal | Output | 1 | 为1时二者相当，否则为0 |

（4）.EXT（拓展器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | EXT\_IN | Input | [15:0] | 输入的16位立即数 |
| 2 | OP | Input | [1:0] | 控制如何拓展立即数 |
| 3 | EXT\_OUT | Output | [31:0] | 输出拓展结果 |

（5）.NPC（PC计算单元）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | PC\_D | Input | [31:0] | D级PC的地址 |
| 2 | PC\_F | Input | [31:0] | F级PC的地址 |
| 3 | j\_if | Input | 1 | J或jal指令执行时为1(D级) |
| 4 | Jr\_if | Input | 1 | jr指令执行时为1(D级) |
| 5 | Bne\_if | Input | 1 | Bne指令执行时为1(D级) |
| 6 | Beq\_if | Input | 1 | Beq指令执行时为1(D级) |
| 7 | IF\_Equal | Input | 1 | CMP比较的结果，1时相等，0时不等 |
| 8 | rs | Input | [4:0] | Mips指令的25-21位(D级) |
| 9 | rt | Input | [4:0] | Mips指令的20-16位(D级) |
| 10 | imm | Input | [15:0] | Mips指令的15-0位(D级) |
| 11 | RD1 | Input | [31:0] | 转发后的GRF[rs]的值 |
| 12 | EXT\_OUT | Input | [31:0] | 扩展后的立即数 |
| 13 | NPC | Output | [31:0] | 下一个PC的值 |

（6）.FMUX\_rs\_D（D级选择转发信号rs多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rd1 | Input | [31:0] | D级GRF[rs]中的值 |
| 2 | Wtod\_rs\_data | Input | [31:0] | W级到D级rs转发的值 |
| 3 | Mtod\_rs\_data | Input | [31:0] | M级到D级rs转发的值 |
| 4 | Etod\_rs\_data | Input | [31:0] | E级到D级rs转发的值 |
| 5 | F\_rs\_D | Input | [1:0] | D级rs转发控制信号 |
| 6 | Forwar\_rd1\_d | Output | [31:0] | D级rs转发后的值 |

（7）.FMUX\_rt\_D（D级选择转发信号rt多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rd2 | Input | [31:0] | D级GRF[rt]中的值 |
| 2 | Wtod\_rt\_data | Input | [31:0] | W级到D级rt转发的值 |
| 3 | Mtod\_rt\_data | Input | [31:0] | M级到D级rt转发的值 |
| 4 | Etod\_rt\_data | Input | [31:0] | E级到D级rt转发的值 |
| 5 | F\_rt\_D | Input | [1:0] | D级rt转发控制信号 |
| 6 | Forwar\_rd2\_d | Output | [31:0] | D级rt转发后的值 |

3.E级流水线

(1).D\_E（E级流水寄存器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IMcode\_D | Input | [31:0] | D级的32位指令码 |
| 2 | RF\_RD1 | Input | [31:0] | D级传入的GRF[rs]的值 |
| 3 | RF\_RD2 | Input | [31:0} | D级传入的GRF[rt]的值 |
| 4 | ext\_imm | Input | [31:0] | D级传入的EXT拓展后imm的值 |
| 5 | CLK | Input | 1 | CLK时钟信号 |
| 6 | Reset | Input | 1 | Reset信号为1时，寄存器 |
| 7 | Stall | Input | 1 | 暂停信号 |
| 8 | PC\_D | Input | [31:0] | D级的PC值 |
| 9 | RS\_E | Output | [31:0] | 向E级传入的GRF[rs]的值 |
| 10 | RT\_E | Output | [31:0] | 向E级传入的GRF[rt]的值 |
| 11 | EXT\_E | Output | [31:0] | 向E级传入的EXT拓展后imm的值 |
| 12 | IMcode\_E | Output | [31:0] | E级的32位指令码 |
| 13 | PC\_E | Output | [31:0] | E级的PC值 |

（2）.ALU（逻辑运算单元）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | ALUOp | input | [2:0] | SrcA与SrcB的运算方式(见下表) |
| 2 | SrcA | input | [31:0] | SrcA的值 |
| 3 | SrcB | input | [31:0] | SrcB的值 |
| 4 | ALUOUT | output | [31:0] | SrcA与SrcB经过ALUop的方式运算后的结果 |

ALUOp含义

|  |  |  |  |
| --- | --- | --- | --- |
| ALUop编号 | 含义 | 位宽 | 功能描述 |
| 000 | “+” | [3:0] | A + B |
| 001 | “-” | [3:0] | A - B |
| 010 | “||” | [3:0] | A || B |
| 011 |  | [3:0] | 未定义 |
| 100 |  | [3:0] | 未定义 |
| 101 |  | [3:0] | 未定义 |
| 110 |  | [3:0] | 未定义 |
| 111 |  | [3:0] | 未定义 |

（3）.FMUX\_rs\_E（E级选择转发信号rs多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rf\_rd1\_e | Input | [31:0] | E级流水寄存器中GRF[rs]的值 |
| 2 | Wtoe\_rs\_data | Input | [31:0] | W级到E级rs转发的值 |
| 3 | Mtoe\_rs\_data | Input | [31:0] | M级到E级rs转发的值 |
| 4 | F\_rs\_E | Input | [1:0] | E级rs转发控制信号 |
| 5 | Forwar\_rd1\_e | Output | [31:0] | E级rs转发后的值 |

（4）.FMUX\_rt\_E（E级选择转发信号rt多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rf\_rd2\_e | Input | [31:0] | E级流水寄存器中GRF[rt]的值 |
| 2 | wtoe\_rt\_data | Input | [31:0] | W级到E级rt转发的值 |
| 3 | mtoe\_rt\_data | Input | [31:0] | M级到E级rt转发的值 |
| 4 | F\_rt\_E | Input | [1:0] | E级rt转发控制信号 |
| 5 | forwar\_rd2\_e | Output | [31:0] | E级rt转发后的值 |

（5）.ALU\_SrcB\_MUX（ALU的SrcB多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | forward\_rd2\_e | Input | [31:0] | E级rt转发后的值 |
| 2 | ext\_out\_e | Input | [31:0] | E级流水寄存器里的EXT扩展后的imm的值 |
| 3 | alu\_srcb\_e | Input | [1:0] | E级控制ALU的SrcB输入的控制信号 |
| 4 | alu\_inb | Output | [31:0] | E级要写入ALUSrcB的值 |

4.M级流水线

(1).E\_M（M级流水寄存器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IMcode\_E | Input | [31:0] | E级的32位指令码 |
| 2 | AO\_E | Input | [31:0] | E级传入的ALU\_OUT的值 |
| 3 | RT\_E | Input | [31:0} | E级传入的GRF[rt]的值 |
| 4 | CLK | Input | 1 | CLK时钟信号 |
| 5 | Reset | Input | 1 | Reset信号为1时，寄存器 |
| 6 | PC\_E | Input | [31:0] | E级的PC值 |
| 7 | PC\_M | Output | [31:0] | M级的PC值 |
| 8 | RT\_M | Output | [31:0] | 向M级传入的GRF[rt]的值 |
| 9 | AO\_M | Output | [31:0] | 向M级传入的ALU\_OUT的值 |
| 10 | IMcode\_M | Output | [31:0] | M级的32位指令码 |

（2）.DM（数据存储器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | AO\_E | Input | [31:0] | E级传入的ALU\_OUT |
| 2 | RT\_E | Input | [31:0] | 写入RAM的32位数据 |
| 3 | PC\_E | Input | [31:0] | E级PC的值 |
| 4 | CLK | Input | 1 | 时钟信号 |
| 5 | Reset | Input | 1 | 当Reset为1时，将RAM清零 |
| 6 | MemWrite | Input | 1 | 当且仅当MemWrite为1时，RAM允许写入 |
| 7 | DM\_OUT | Output | [31:0] | RAM输出的数据 |

（3）.FMUX\_rt\_M（M级选择转发信号rt多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rd2\_m | Input | [31:0] | M级流水寄存器中GRF[rt]的值 |
| 2 | wtom\_rt\_data | Input | [31:0] | W级到M级rt转发的值 |
| 3 | F\_rt\_M | Input | [1:0] | M级rt转发控制信号 |
| 4 | forwar\_rd1\_m | Output | [31:0] | M级rt转发后的值 |

（4）.MUX\_AO\_PC8\_M（M级向其他级转发的数据多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | ao\_m | Input | [31:0] | E级流水寄存器中ALU\_OUT的值 |
| 2 | pc8\_m | Input | [31:0] | W级PC+8的值 |
| 4 | j\_if\_m | Input | [1:0] | M级是否是j型指令 |
| 5 | forwar\_data\_m | Output | [31:0] | M级转发出去的数据 |

5.W级流水线

（5）.M\_W（W级流水寄存器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IMcode\_M | Input | [31:0] | M级的32位指令码 |
| 2 | AO\_M | Input | [31:0] | M级传入的ALU\_OUT的值 |
| 3 | DO\_M | Input | [31:0} | M级传入的DM的值 |
| 4 | CLK | Input | 1 | CLK时钟信号 |
| 5 | Reset | Input | 1 | Reset信号为1时，寄存器 |
| 6 | PC\_M | Input | [31:0] | M级的PC值 |
| 7 | PC\_W | Output | [31:0] | W级的PC值 |
| 8 | DO\_M | Output | [31:0] | 向W级传入的DM中的值 |
| 9 | AO\_M | Output | [31:0] | 向W级传入的ALU\_OUT的值 |
| 10 | IMcode\_W | Output | [31:0] | W级的32位指令码 |

（6）.MUX\_AO\_DM\_PC8\_W（W级向其他级转发数据多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | dm\_out\_w | Input | [31:0] | W级流水寄存器中DM\_OUT的值 |
| 2 | ao\_w | Input | [31:0] | W级流水寄存器中ALU\_OUT的值 |
| 4 | pc8\_w | Input | [31:0] | W级PC+8的值 |
| 5 | MemtoReg\_w | Input | [1:0] | W级寄存器堆写入的值选择信号 |
| 6 | forwar\_data\_w | Output | [31:0] | W级向其他级的转发数据，也是W级写入GRF的数据 |

（7）.A3\_MUX（W级写入GRF写入地址多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rt\_w | Input | [31:0] | W级的rt |
| 2 | rd\_w | Input | [31:0] | W级的rd |
| 4 | A3\_jal | Input | [31:0] | 31号寄存器 |
| 5 | RegDst\_w | Input | [1:0] | W级寄存器堆写入的地址的选择信号 |
| 6 | a3 | Output | [31:0] | W级写入GRF的地址 |

**二.控制模块设计（分布式）**

1.Splitter（D,E,M,W 各级指令的指令分离器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | A | Input | [31:0] | 32位mips指令 |
| 2 | op | Output | [31:26] | Op |
| 3 | rs | Output | [25:21] | Rs |
| 4 | rt | Output | [20:16] | Rt |
| 5 | rd | Output | [15:11] | Rd |
| 6 | immediate | Output | [15:0] | 立即数 |
| 7 | funct | output | [5:0] | Funct |

2.Control(D,E,M,W 各级指令的信号产生器)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | op | input | [5:0] | OP信号 |
| 2 | funct | input | [5:0] | FUNCT信号 |
| 3 | Jr\_if | Output | 1 | Jr指令执行时为1 |
| 4 | Bne\_if | Output | 1 | Bne指令执行时为1 |
| 5 | J\_if | Output | 1 | J或jal指令执行时为1 |
| 6 | RegDst | Output | [1:0] | 选择GRF的写入地址 |
| 7 | ALUSrc | Output | 1 | 选择传入 ALU 作为 ScrB 的数据 |
| 8 | MemtoReg | Output | [1:0] | 选择GRF的写入数据 |
| 9 | RegWrite | Output | 1 | GRF 的写入使能控制指令 |
| 10 | MemWrite | Output | 1 | DM 的写入使能控制指令 |
| 11 | Beq\_if | Output | 1 | Beq指令执行时为1 |
| 12 | ALUOp | Output | [2:0] | 对 ALU 的计算类型的选择信号 |
| 13 | EXTOp | Output | [1:0] | 对立即数拓展类型的选择信号 |

Control控制信号生成真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| op | 000000 | 000000 | 001101 | 100011 | 101011 | 000100 | 001111 | 000000 | 000011 | 000000 |
| funct | 100001 | 100011 | NULL | | | | | 000000 | NULL | 001000 |
| 指令名称 | addu | subu | ori | lw | sw | beq | lui | j | Jal | jr |
| RegDst | 01 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 |
| RegWrite | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| ALUsrc | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | X |
| ALUOp | 000"+" | 001"-" | 010"||" | 000"+" | 000"+" | 001"-" | 000"+" | 000“+” | 000“+” | 000“+” |
| Beq\_if | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| MemWrite | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| MemToReg | 01 | 01 | 01 | 00 | 00 | 00 | 01 | 00 | 10 | 00 |
| EXTOp | 00 | 00 | 00 | 01 | 01 | 01 | 10 | 00 | 00 | 00 |
| J\_if | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Jr\_if | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

3.Make\_TuseTnew（D,E,M,W 各级 TuseTnew产生器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | IMcode | Input | [31:0] | 32位mips指令 |
| 2 | Tuse\_rs | Output | [1:0] | 产生D级的Tuse\_rs |
| 3 | Tuse\_rt | Output | [1:0] | 产生D级的Tuse\_rt |
| 4 | Tnew\_E | Output | [1:0] | 产生E级的Tnew\_E |
| 5 | Tnew\_M | Output | [1:0] | 产生M级的Tnew\_M |
| 6 | Tnew\_W | Output | [1:0] | 产生W级的Tnew\_W |

4.A3\_MUX（D,E,M,W 各级 A3多选器）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | rt | Input | [4:0] | 对应级的rt |
| 2 | rd | Input | [4:0] | 对应级的rt |
| 3 | A3\_jal | Input | [4:0] | 31号寄存器 |
| 4 | RegDst | Input | [1:0] | 对应级的向GRF写入地址的选择信号 |
| 5 | a3 | Output | [4:0] | 对应级写入GRF的地址 |

5.Hazard（冲突单元）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能名称 | 输入/输出 | 位宽 | 功能描述 |
| 1 | Tuse\_rs | Input | [1:0] | 此时刻的Tuse\_rs |
| 2 | Tuse\_rt | Input | [1:0] | 此时刻的Tuse\_rt |
| 3 | Tnew\_E | Input | [1:0] | 此时刻的Tnew\_E |
| 4 | Tnew\_M | Input | [1:0] | 此时刻的Tnew\_M |
| 5 | Tnew\_W | Input | [1:0] | 此时刻的Tnew\_W |
| 6 | A1\_D | Input | [4:0] | D级用的rs |
| 7 | A2\_D | Input | [4:0] | D级用的rt |
| 8 | A1\_E | Input | [4:0] | E级用的rs |
| 9 | A2\_E | Input | [4:0] | E级用的rt |
| 10 | A2\_M | Input | [4:0] | M级用的rt |
| 11 | RegWrite\_E | Input | 1 | E级GRF写入使能 |
| 12 | RegWrite\_M | Input | 1 | M级GRF写入使能 |
| 13 | RegWrite\_W | Input | 1 | W级GRF写入使能 |
| 14 | A3\_E | Input | [4:0] | E级GRF写入地址 |
| 15 | A3\_M | Input | [4:0] | M级GRF写入地址 |
| 16 | A3\_W | Input | [4:0] | W级GRF写入地址 |
| 17 | Stall | Output | 1 | 暂停信号 |
| 18 | F\_rs\_D | Output | [1:0] | D级rs转发信号 |
| 19 | F\_rt\_D | Output | [1:0] | D级rt转发信号 |
| 20 | F\_rs\_E | Output | [1:0] | E级rs转发信号 |
| 21 | F\_rt\_E | Output | [1:0] | E级rt转发信号 |
| 22 | F\_rt\_M | Output | [1:0] | M级rt转发信号 |

**三.测试代码**

ori $29,$0,0

341d0000

341c0000

34090014

3c01000a

ac010000

34630011

ac030004

34840010

ac040008

00642823

ac05000c

8c06000c

8c070008

00c74021

ac080010

01064023

00000000

11030003

01064021

00000000

0c000c10

01064021

11090001

03e00008

00000000

ori $28,$0,0

ori $9,$0,20

lui $1,10

sw $1,0($0)

ori $3,$3,17

sw $3,4($0)

ori $4,$4,16

sw $4,8($0)

subu $5,$3,$4

sw $5,12($0)

lw $6,12($0)

lw $7,8($0)

addu $8,$6,$7

sw $8,16($0)

subu $8,$8,$6

one:

nop

beq $8,$3,yep

addu $8,$8,$6

nop

jal one

yep:

addu $8,$8,$6

beq $8,$9,end

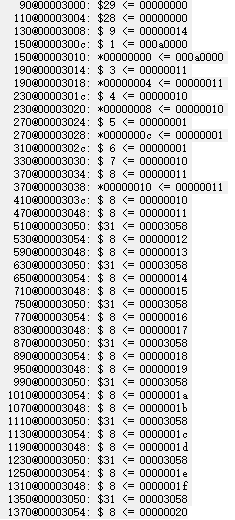
jr $ra

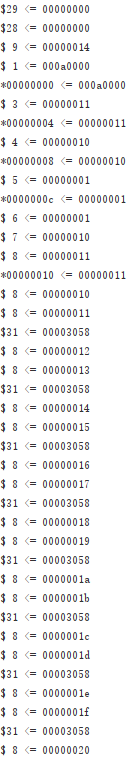
end:

nop

**测试结果**

预期结果 仿真结果





**四.思考题**

1、Cal\_i(ori)和Cal\_r(addu,subu)的冲突

测试代码 预期结果 实际结果

90@00003000: $ 9 <= 00000022

110@00003004: $ 9 <= 00000037

130@00003008: $10 <= 00000015

150@0000300c: $ 9 <= 0000003f

170@00003010: $ 9 <= 00000054

190@00003014: $10 <= 0000003f

210@00003018: $10 <= 00000093

230@0000301c: $ 9 <= 00000126

250@00003020: $ 9 <= 0000012f

270@00003024: $10 <= 00000126

290@00003028: $ 9 <= 0000012f

310@0000302c: $10 <= 0000012f 0000012f350@00003030: $10 <= 000000a8

390@00003038: $10 <= 00000150

$ 9 <= 00000022

$ 9 <= 00000037

$10 <= 00000015

$ 9 <= 0000003f

$ 9 <= 00000054

$10 <= 0000003f

$10 <= 00000093

$ 9 <= 00000126

$ 9 <= 0000012f

$10 <= 00000126

$ 9 <= 0000012f

$10 <= 0000012f

由于Cal\_i和Cal\_r型指令的Tuse都是1，且E级Tnew也是1，故所有冲突都可以通过转发解决。

ori $t1,34

ori $t1,53

ori $t2,21

ori $t1,13

addu $t1,$t2,$t1

subu $t2,$t1,$t2

addu $t2,$t1,$t2

addu $t1,$t2,$t2

ori $t1,15

addu $t2,$t2,$t2

ori $t1,15

ori $t2,15

2、Save(sw)与Load(lw)与Cal\_i(ori)的冲突

90@00003000: $ 9 <= 00000030

90@00003004: \*00000004 <= 00000030

130@00003008: $ 9 <= 00000034

130@0000300c: \*00000038 <= 00000034

150@00003010: \*00000030 <= 00000034

190@00003014: $10 <= 00000034

190@00003018: \*00000008 <= 00000034

210@0000301c: \*0000000c <= 00000034

250@00003020: $10 <= 00000030

290@00003024: $10 <= 0000003c

310@00003028: $10 <= 00000034

350@00003030: $10 <= 0000003c

ori $t1,48

sw $t1,4($t0)

ori $t1,36

sw $t1,4($t1)

sw $t1,-4($t1)

lw $t2,4($t1)

sw $t2,8($t0)

sw $t2,12($t0)

lw $t2,4($t0)

ori $t2,12

lw $t2,4($t1)

nop

ori $t2,12

测试代码 预期结果 实际结果

Load类在E级Tnew是2，当D级有Tuse为1的指令，若操作寄存器一样，则只能暂停，其他时候都可以转发。

$ 9 <= 00000030

\*00000004 <= 00000030

$ 9 <= 00000034

\*00000038 <= 00000034

\*00000030 <= 00000034

$10 <= 00000034

\*00000008 <= 00000034

\*0000000c <= 00000034

$10 <= 00000030

$10 <= 0000003c

$10 <= 00000034

$10 <= 0000003c

3、Save(sw)与Load(lw)与Cal\_r(addu)的冲突

测试代码 预期结果 实际结果

90@00003000: $ 9 <= 00000030

110@00003004: $10 <= 00000024

130@00003008: $ 9 <= 00000054

130@0000300c: \*00000004 <= 00000054

170@00003010: $ 9 <= 000000a8

170@00003014: \*000000ac <= 000000a8

190@00003018: \*000000a4 <= 000000a8

230@0000301c: $10 <= 000000a8

230@00003020: \*00000008 <= 000000a8

250@00003024: \*0000000c <= 000000a8

290@00003028: $10 <= 00000054

330@0000302c: $10 <= 000000a8

350@00003030: $10 <= 000000a8

390@00003038: $10 <= 00000150

$ 9 <= 00000030

$10 <= 00000024

$ 9 <= 00000054

\*00000004 <= 00000054

$ 9 <= 000000a8

\*000000ac <= 000000a8

\*000000a4 <= 000000a8

$10 <= 000000a8

\*00000008 <= 000000a8

\*0000000c <= 000000a8

$10 <= 00000054

$10 <= 000000a8

$10 <= 000000a8

$10 <= 00000150

ori $t1,48

ori $t2,36

addu $t1,$t1,$t2

sw $t1,4($t0)

addu $t1,$t1,$t1

sw $t1,4($t1)

sw $t1,-4($t1)

lw $t2,4($t1)

sw $t2,8($t0)

sw $t2,12($t0)

lw $t2,4($t0)

addu $t2,$t2,$t2

lw $t2,4($t1)

nop

addu $t2,$t2,$t2

Load类在E级Tnew是2，当D级有Tuse为1的指令，若操作寄存器一样，则只能暂停，其他时候都可以转发。

4、Cal\_r(addu)与beq，Cal\_i(ori)与beq的冲突

测试代码 预期结果 实际结果

90@00003000: $ 9 <= 00000001

110@00003004: $10 <= 00000002

130@00003008: $ 9 <= 00000003

210@00003014: $11 <= 00000004

230@00003018: $12 <= 00000004

$ 9 <= 00000001

$10 <= 00000002

$ 9 <= 00000003

$11 <= 00000004

$12 <= 00000004

beq的Tuse为0，Cal\_r和Cal\_i类在E级Tnew都是1，在对寄存器操作相同时只能暂停。其他时候都可以转发。

s0:

ori $t1,1

ori $t2,2

addu $t1,$t1,$t2

beq $t1,$t2,s0

nop

ori $t3,4

ori $t4,4

s1:

beq $t3,$t4,s1

nop

5、Load(lw)与beq的冲突

90@00003000: $ 9 <= 00000064

90@00003004: \*00000000 <= 00000064

130@00003008: $11 <= 00000064

230@00003014: $12 <= 00000064

250@00003018: $13 <= 00000064

$ 9 <= 00000064

\*00000000 <= 00000064

$11 <= 00000064

$12 <= 00000064

$13 <= 00000064

测试代码 预期结果 实际结果

由于 beq的Tuse为0，Load(lw)在E级的Tnew是2，所以只能暂停且要暂停两周期。但是当Load(lw)在W级时Tnew是0，可以转发。

s0:

ori $t1,100

sw $t1,0($t0)

lw $t3,0($t0)

beq $t2,$t3,s0

nop

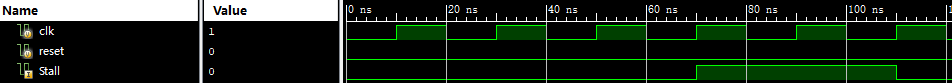
lw $t4,0($t0)

lw $t5,0($t0)

s1:

beq $t4,$t5,s1

nop



6、Jal、Jr与Cal\_r(addu)与Cal\_i(ori)的冲突

90@00003000: $ 9 <= 00000004

110@00003004: $31 <= 0000300c

130@00003008: $31 <= 00003010

210@00003010: $31 <= 00003010

230@00003014: $31 <= 00003018

$ 9 <= 00000004

$31 <= 0000300c

$31 <= 00003010

$31 <= 00003010

$31 <= 00003018

s0:

ori $t1,4

jal s1

addu $31,$31,$t1

ori $ra,0

ori $ra,0

ori $ra,12312

s1:

jr $ra

nop

当Jr在D级，Tuse为0，Cal\_r(addu)在E级Tnew为1，只能暂停，Cal\_i(ori)同理。其他情况都可转发。

测试代码 预期结果 实际结果

7、Jal、Jr与Sw与Lw的冲突

90@00003000: $ 9 <= 00000004

110@00003004: $31 <= 0000300c

110@00003008: \*00000000 <= 0000300c

150@00003014: $31 <= 0000300c

170@00003018: $31 <= 0000300c

270@0000300c: $31 <= 0000300c

290@00003010: $31 <= 0000301c

310@00003014: $31 <= 0000301c

330@00003018: $31 <= 0000300c

430@0000300c: $31 <= 0000300c

450@00003010: $31 <= 0000301c

470@00003014: $31 <= 0000301c

490@00003018: $31 <= 0000300c

590@0000300c: $31 <= 0000300c

$ 9 <= 00000004

$31 <= 0000300c

\*00000000 <= 0000300c

$31 <= 0000300c

$31 <= 0000300c

$31 <= 0000300c

$31 <= 0000301c

$31 <= 0000301c

$31 <= 0000300c

$31 <= 0000300c

$31 <= 0000301c

$31 <= 0000301c

$31 <= 0000300c

测试代码 预期结果 实际结果

由与jr在D级时Tues为0，lw的Tnew\_E是2，Tnew\_M是1，若操作相同寄存器，只能暂停，且lw在E级时要暂停两周期。

s0:

ori $t1,4

jal s1

sw $ra,-4($t1)

ori $ra,0

ori $ra,12312

s1:

ori $ra,0

lw $ra,-4($t1)

jr $ra

nop

**如下附上TnewTuse和暂停转发表格：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tuse** |  |  |  | **功能部件** | **Tnew** |  |  |
| **指令** | **rs** | **rt** |  |  | **E** | **M** | **W** |
| **addu** | **1** | **1** |  | **ALU** | **1** | **0** | **0** |
| **subu** | **1** | **1** |  | **ALU** | **1** | **0** | **0** |
| **ori** | **1** |  |  | **ALU** | **1** | **0** | **0** |
| **lw** | **1** |  |  | **DM** | **2** | **1** | **0** |
| **sw** | **1** | **2** |  |  |  |  |  |
| **beq** | **0** | **0** |  |  |  |  |  |
| **lui** |  |  |  | **ALU** | **1** | **0** | **0** |
| **j** |  |  |  |  |  |  |  |
| **jal** |  |  |  | **PC** | **0** | **0** | **0** |
| **jr** | **0** |  |  |  |  |  |  |
|  | {1,0} | {1,2,0} |  |  | {1,2,0} | {1,0} | {0} |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RS | E | | | M | | | W | | |
| Tuse/Tnew | ALU | DM | PC | ALU | DM | PC | ALU | DM | PC |
|  | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | S | S | F | F | S | F | F | F | F |
| 1 | F | S | F | F | F | F | F | F | F |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| RT | E | | | M | | | W | | |
| Tuse/Tnew | ALU | DM | PC | ALU | DM | PC | ALU | DM | PC |
|  | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | S | S | F | F | S | F | F | F | F |
| 1 | F | S | F | F | F | F | F | F | F |
| 2 | F | F | F | F | F | F | F | F | F |