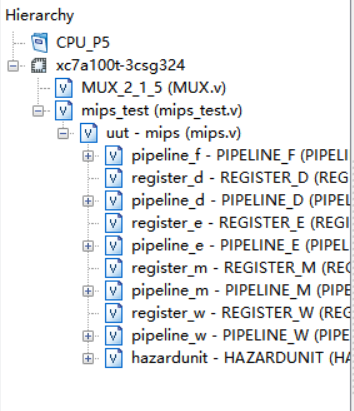
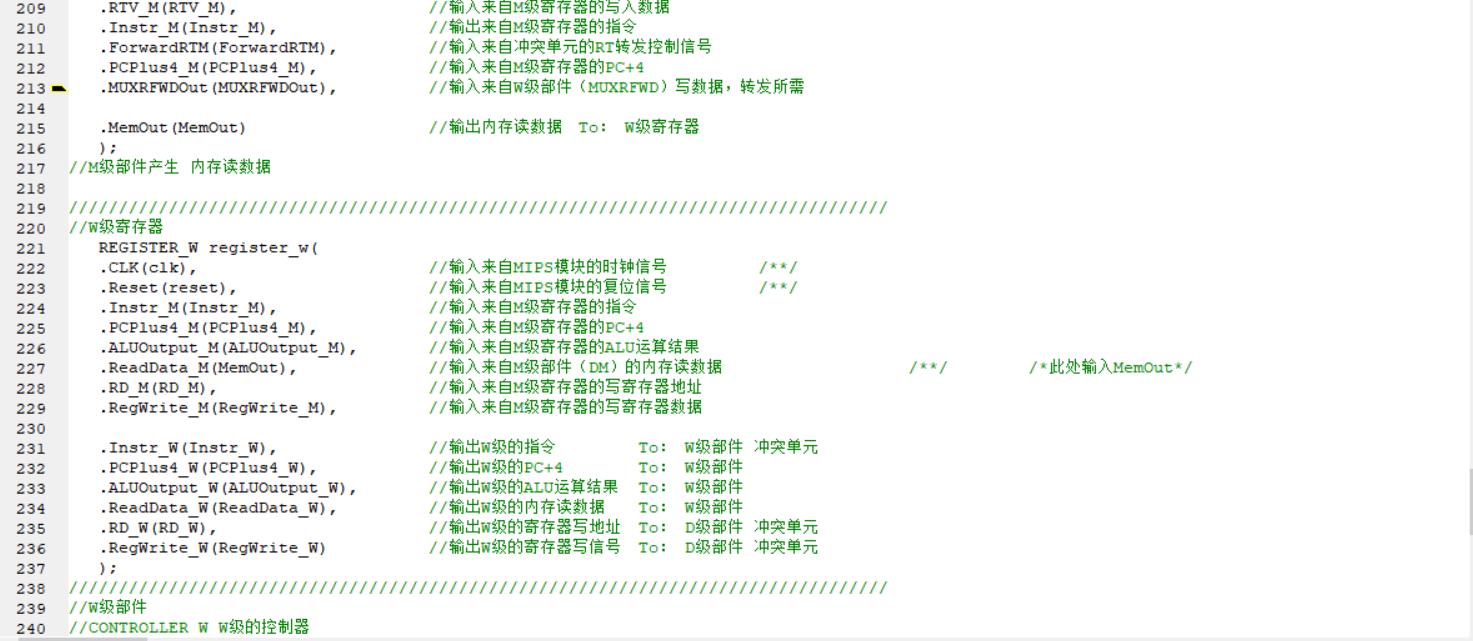
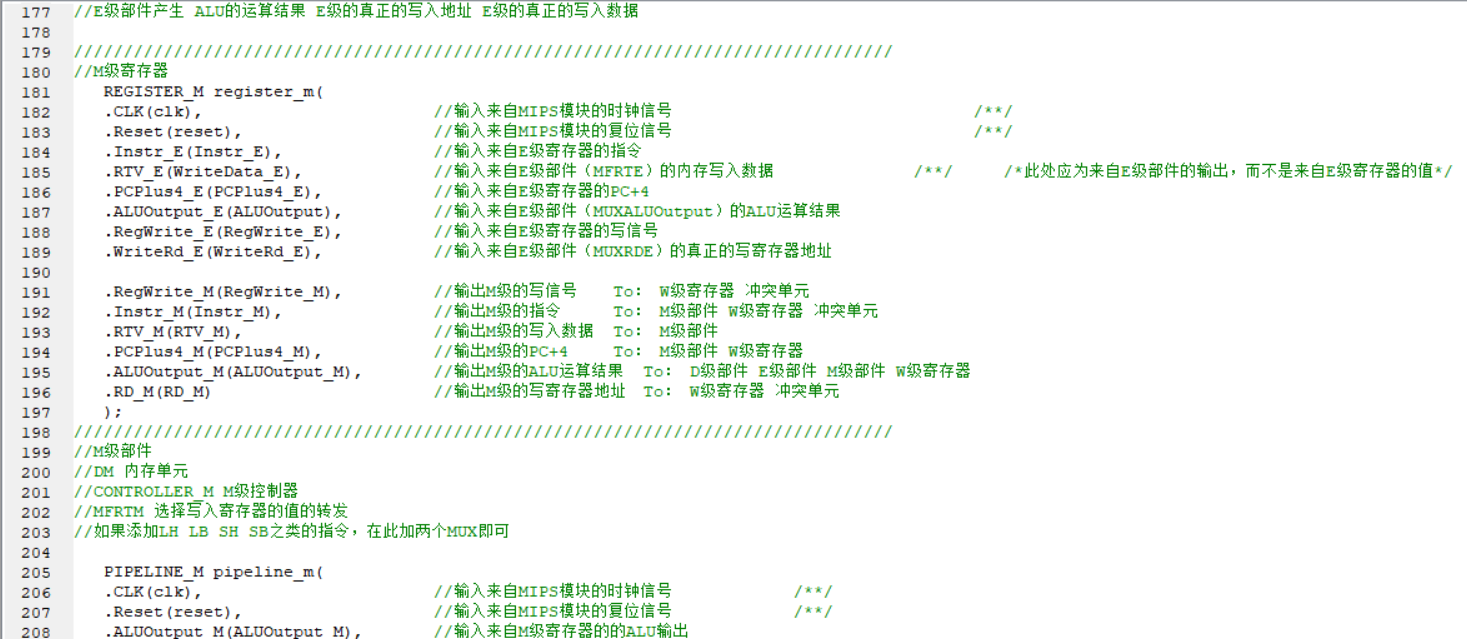
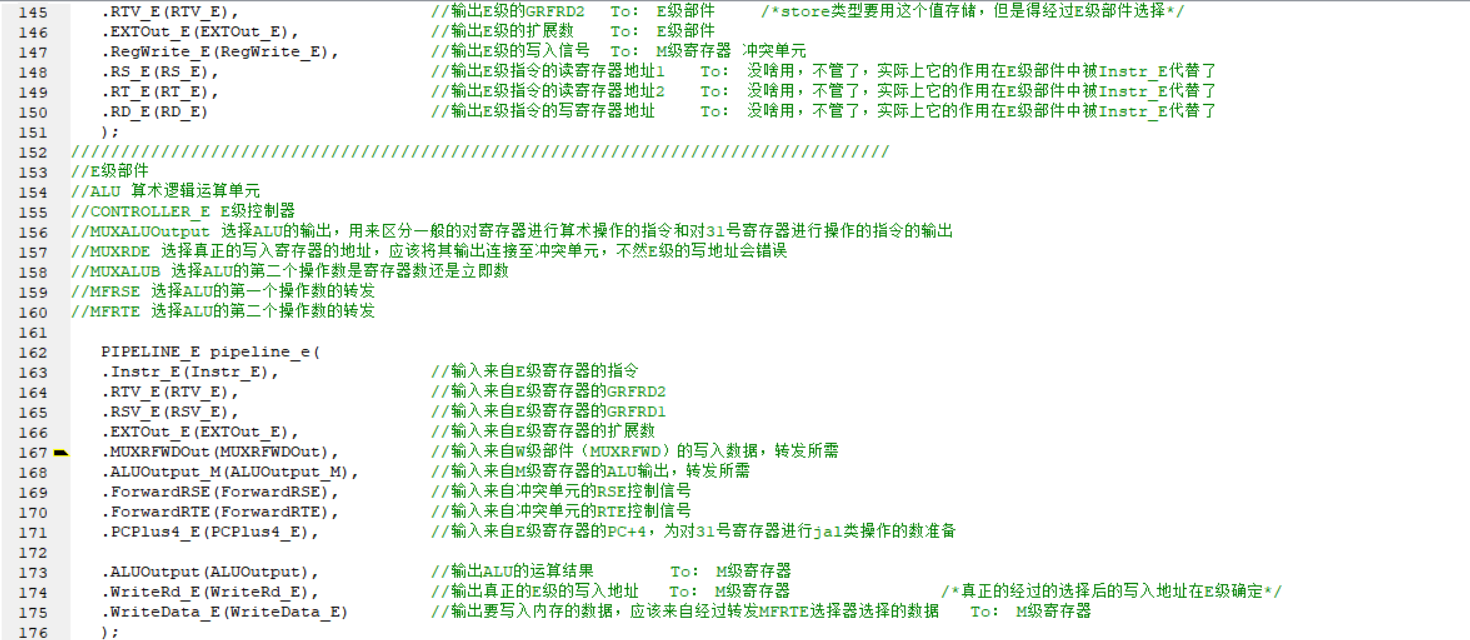
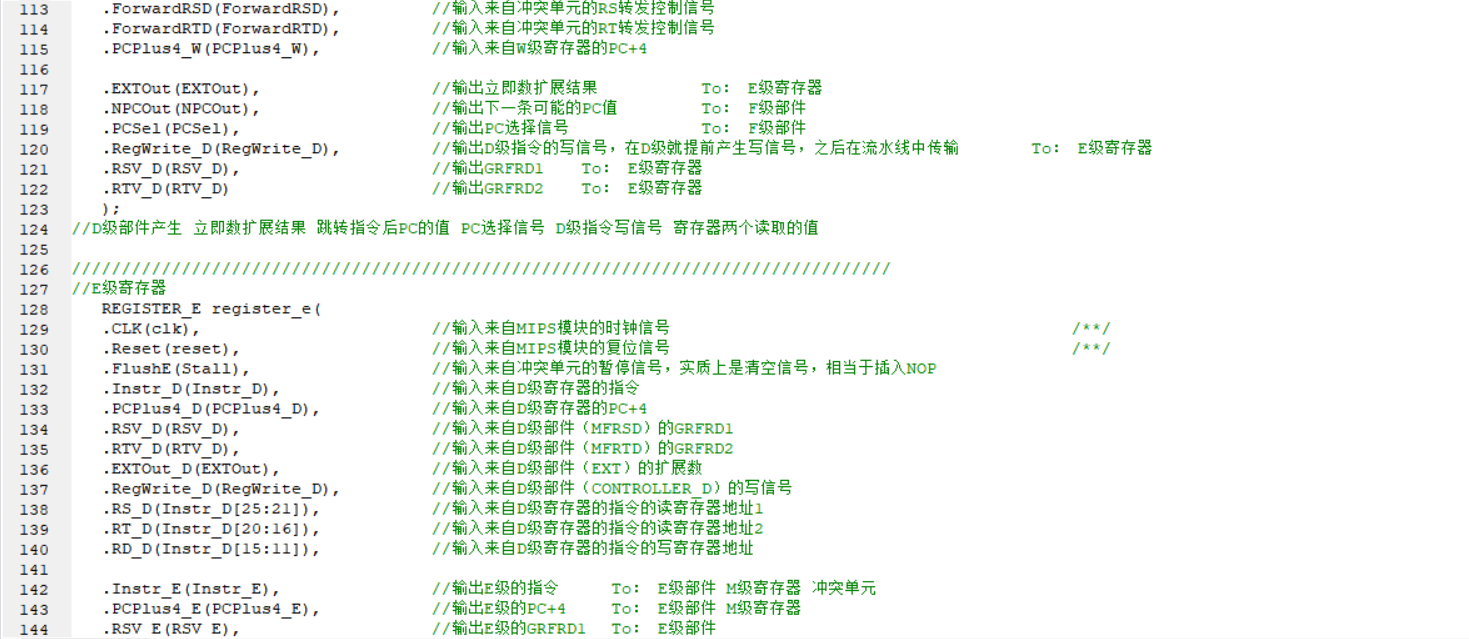
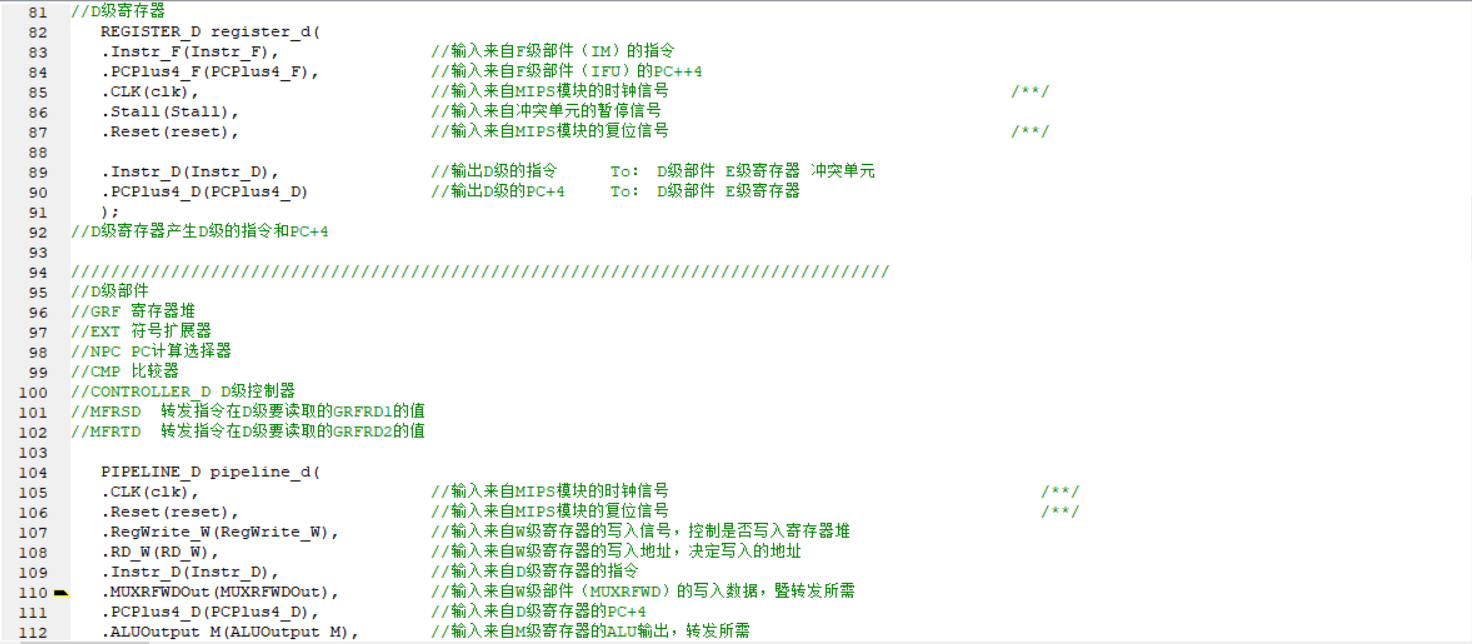
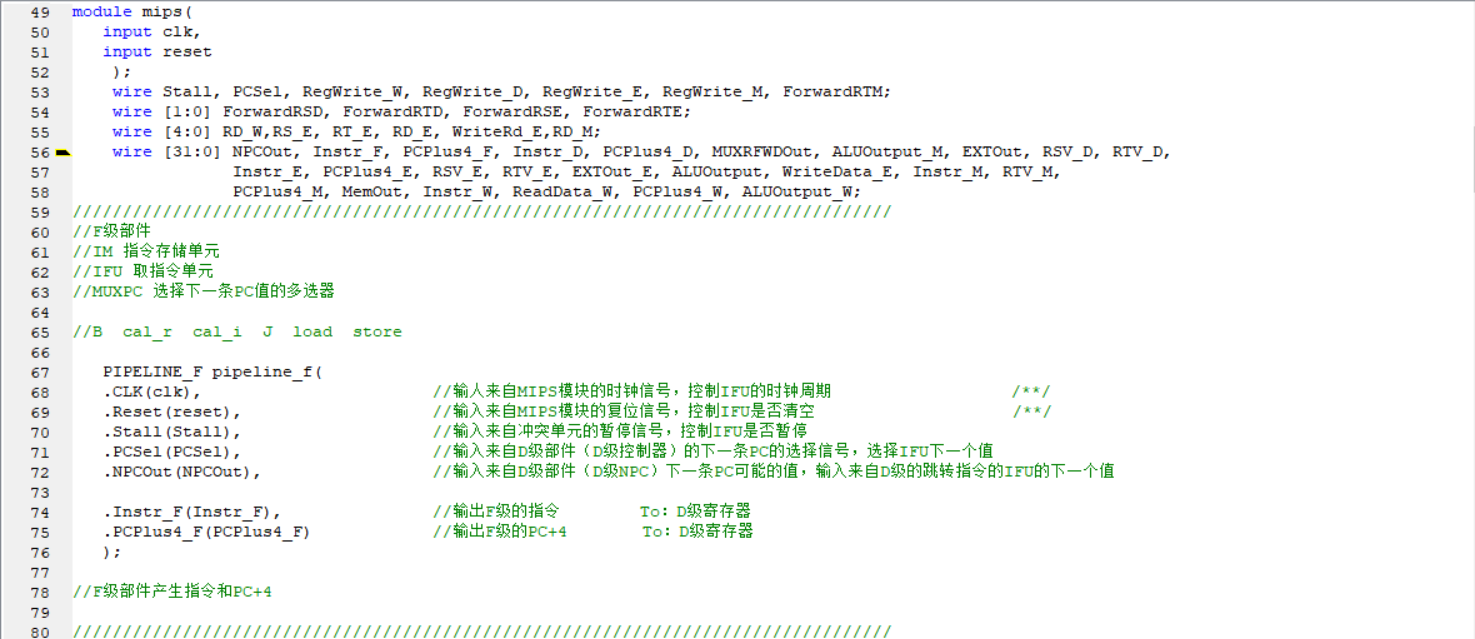
**Project5 Verilog完成流水线CPU开发**

1. **顶层设计**





1. **模块设计**
2. **PIPELINE\_F**

**（1）端口定义**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 时钟信号 |
| Reset | I | 复位信号 |
| Stall | I | 输入来自冲突单元的暂停信号 |
| PCSel | I | 来自D级部件的PC选择信号 |
| NPCOut | I | 来自D级部件的NPC输出 |
| Instr\_F | O | 输出F级的指令 |
| PCPlus4\_F | O | 输出F级的PC+4 |

1. **功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 内部部件 | 功能描述 |
| 1 | IM | 存储和输出当前指令 |
| 2 | IFU | 存储和输出当前指令地址PC |
| 3 | MUXPC | 选择下一条指令地址 |

1. **内部部件**
2. **IM**



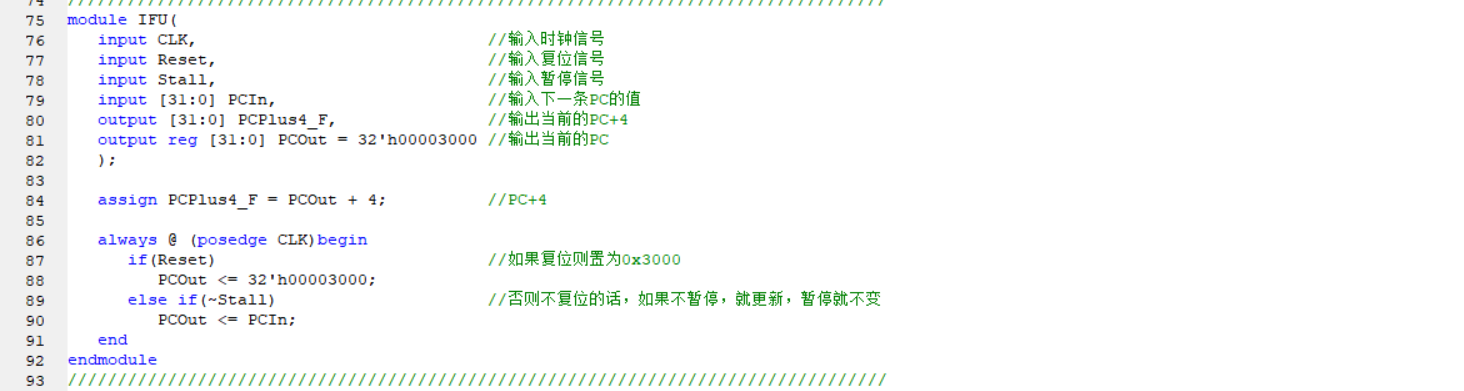
**①端口定义**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| InstrAddr | I | 输入指令地址 |
| Instr\_F | O | 输出当前指令 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 取指令 | 根据InstrAddr指定的地址从IM中取出指令 |

1. **IFU**



**①端口定义**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 时钟信号 |
| Reset | I | 复位信号 |
| Stall | I | 来自冲突单元的暂停信号 |
| PCIn | I | 来自MUXPC的下一个PC值 |
| PCOut | O | 输出PC值 |
| PCPlus4\_F | O | 输出当前PC+4 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当Reset信号有效时，PC被置为0x00003000 |

1. **MUX\_PC（MUX\_2\_1\_32）**



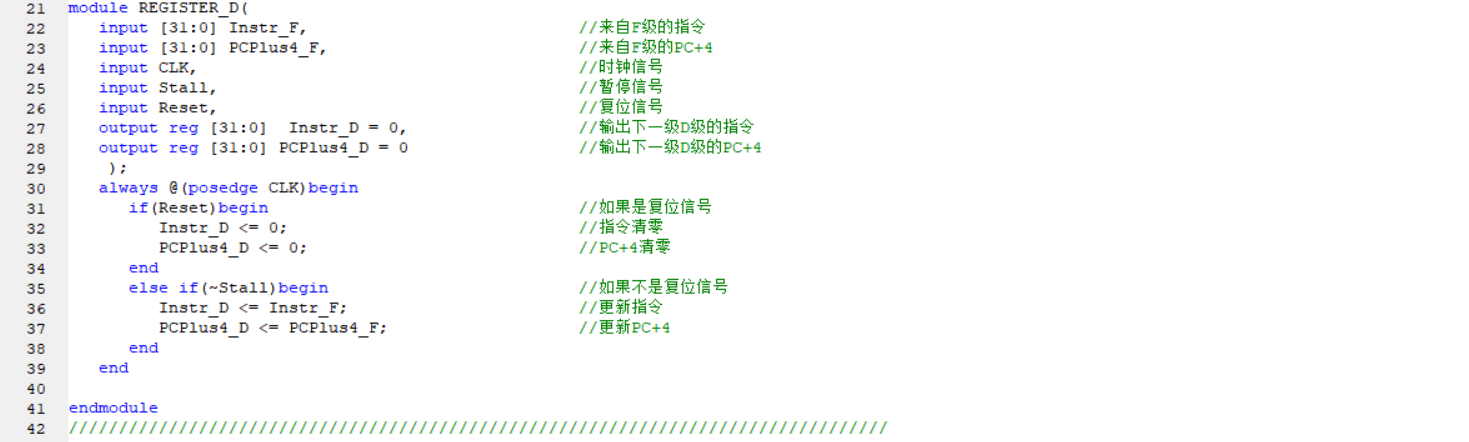
**①端口定义**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| PCPlus4\_F | I | 输入来自IFU的PC+4 |
| NPCOut | I | 输入来自D级部件的B或J类指令对应的PC |
| PCSel | I | 输入来自D级部件的PC选择信号 |
| PCIn | O | 输出下一条要更新的PC值 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择下一条指令地址 | 当PCSel有效时选择NPCOut,即跳转指令的地址 |

1. **Regsiter\_D**



1. **模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_F | I | 来自F级部件的指令 |
| PCPlus4\_F | I | 来自F级部件的PC+4 |
| CLK | I | 时钟信号 |
| Stall | I | 来自冲突单元的暂停信号 |
| Reset | I | 复位信号 |
| Instr\_D | O | 输出D级寄存器的指令 |
| PCPlus4\_D | O | 输出D级寄存器的PC+4 |

1. **PIPELINE\_D**
2. **端口定义**

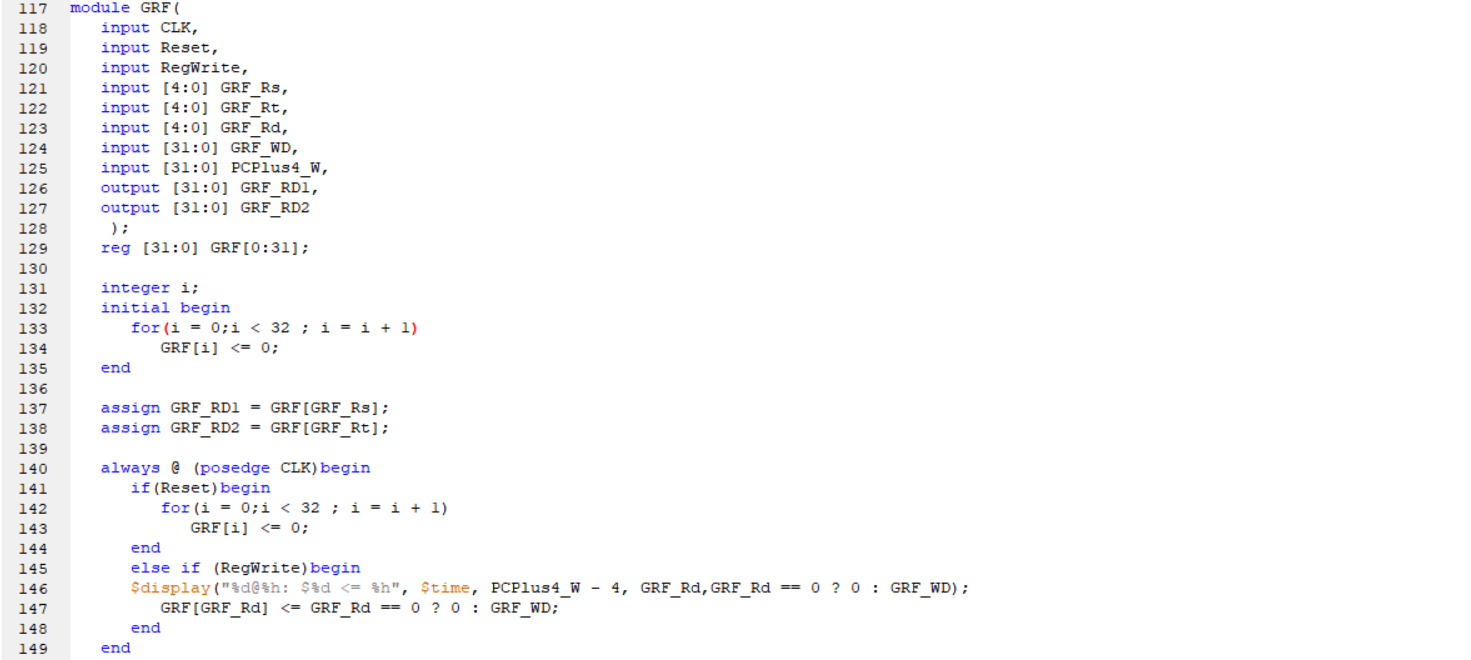
|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 时钟信号 |
| Reset | I | 复位信号 |
| RegWrite\_W | I | 输入来自W级寄存器的写信号 |
| RD\_W | I | 输入来自W级寄存器的写地址 |
| Instr\_D | I | 输入来自D级寄存器的指令 |
| MUXRFWDOut | I | 输入来自W级部件的写数据，也是转发所需要的数据 |
| PCPlus4\_D | I | 输入来自D级寄存器的PC+4 |
| ALUOutput\_M | I | 输入来自M级寄存器的ALU运算结果，也是转发所需要的数据 |
| ForwardRSD | I | 输入来自冲突单元的RSD转发控制信号 |
| ForwardRTD | I | 输入来自冲突单元的RTD转发控制信号 |
| PCPlus4\_W | I | 输入来自W级寄存器的PC+4 |
| EXTOut | O | 输出D级部件产生的立即数扩展结果 |
| NPCOut | O | 输出D级部件的跳转指令产生的PC值 |
| PCSel | O | 输出D级部件产生的PC选择信号 |
| RegWrite\_D | O | 输出D级部件的指令的写信号 |
| RSV\_D | O | 输出寄存器堆读数据1 |
| RTV\_D | O | 输出寄存器堆读数据2 |

1. **功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数据 | 输出指令要读的rs、rt的寄存器对应的值 |
| 2 | 扩展立即数 | 扩展指令中的立即数作为E级的ALU操作数 |
| 3 | 计算跳转地址 | 计算跳转指令对应的跳转地址 |
| 4 | 控制跳转 | 判断当前指令是否为跳转指令 |
| 5 | 写数据 | 将指令要写的数据写入rd寄存器中 |

**（3）内部部件**

**1）GRF**



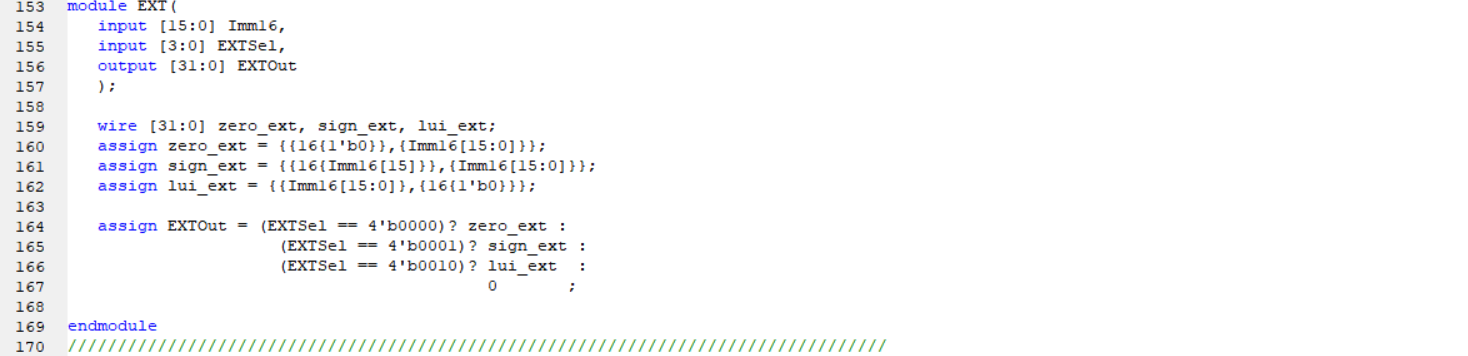
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| GRF\_Rs | I | 读寄存器地址1 |
| GRF\_Rt | I | 读寄存器地址2 |
| GRF\_Rd | I | 写寄存器地址 |
| GRF\_WD | I | 写寄存器数据 |
| CLK | I | 时钟信号 |
| Reset | I | 复位信号 |
| RegWrite | I | 寄存器写信号 |
| PCPlus4\_W | I | 写指令的地址 |
| GRF\_RD1 | O | 读寄存器数据1 |
| GRF\_RD2 | O | 读寄存器数据2 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数据 | 读出GRF\_Rs和GRF\_Rt地址对应寄存器中的数据到GRF\_RD1、GRF\_RD2 |
| 2 | 写数据 | 当RegWrite信号有效且时钟上升沿到来时，将GRF\_WD写入GRF\_Rd对应的寄存器中 |

1. **EXT**



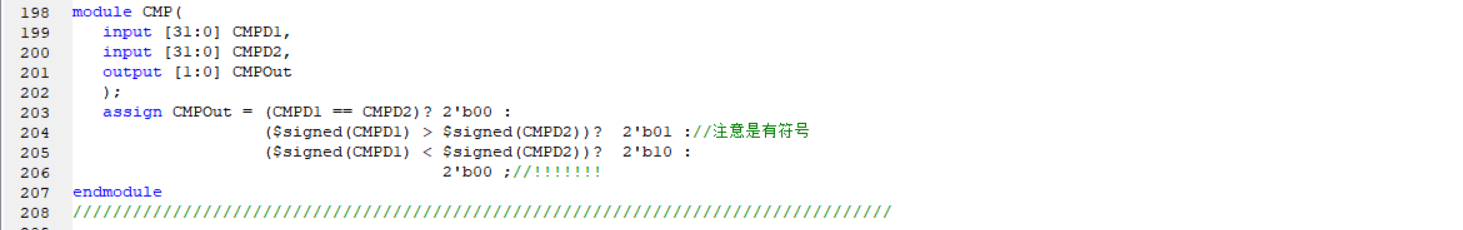
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Imm16 | I | 进行扩展的立即数 |
| EXTSel | I | 扩展选择信号 |
| EXTOut | O | 扩展结果 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 零扩展 | 对立即数进行高位补0扩展 |
| 2 | 符号扩展 | 对立即数进行符号扩展至32位 |
| 3 | 加载到高位 | 将立即数加载到高位，低位补0 |

1. **CMP**



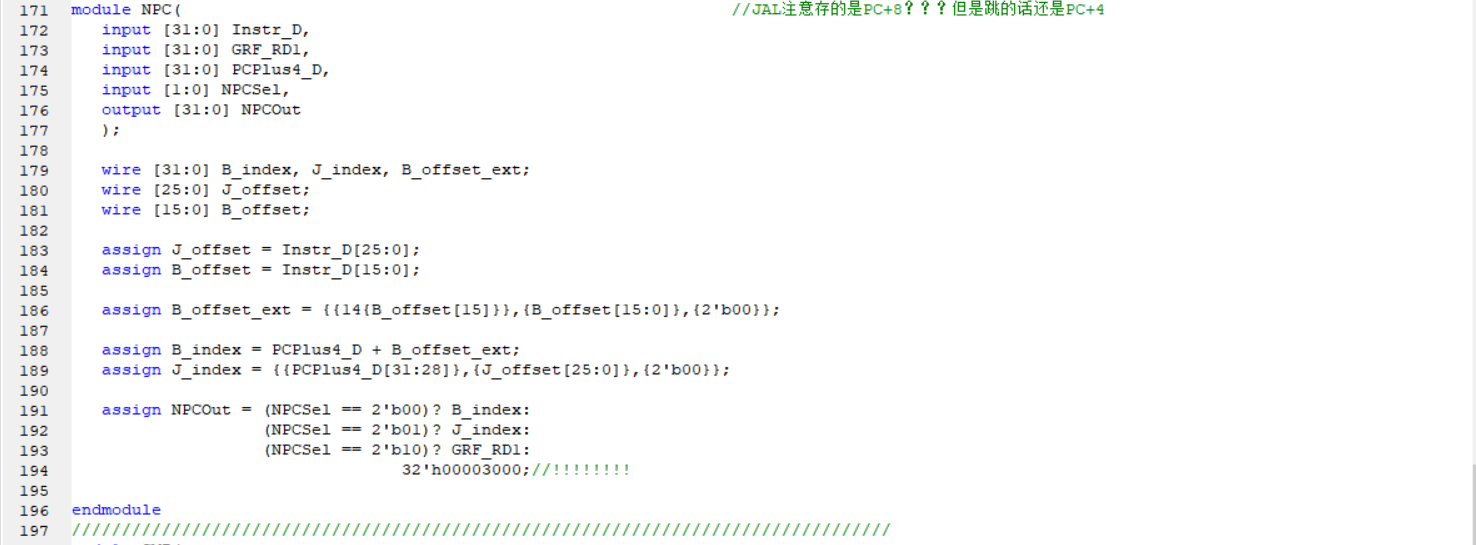
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CMPD1 | I | 比较的第一个数 |
| CMPD2 | I | 比较的第二个数 |
| CMPOut | O | 比较的结果 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 比较两个操作数 | 00：CMPD1 == CMPD2  01：CMPD1>CMPD2  10:CMPD1<CMPD2 |

1. **NPC**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| GRF\_RD1(MFRSDOut) | I | 输入Ra中存储的地址 |
| PCPlus4\_D | I | 输入来自D级寄存器的PC+4 |
| Instr\_D | I | 输入D级寄存器的指令，实质上是传输指令中的偏移量 |
| NPCSel | I | 输入NPC选择信号 |
| NPCOut | O | 输出B J JR指令对应的PC |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出跳转地址 | NPCSel00：B型指令地址  NPCSel01：J型指令地址  NPCSel10：JR指令地址 |

1. **MFRSD（MUX\_4\_1\_32）**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| GRF\_RD1 | I | 输入GRF的读数据1 |
| MUXRFWDOut | I | 输入来自W级部件的GRF要写入的数据 |
| ALUOutput\_M | I | 输入来自M级寄存器的ALU运算结果 |
| ForwardRSD | I | 输入来自冲突单元的RSD转发信号 |
| MFRSDOut | O | 输出选择后的指令要用的GRF操作数1 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出转发选择后的GRF读数据1 | ForwardRSD：  00 GRF\_RD1  01 MUXRFWDOut  10 ALUOutput\_M |

1. **MFRTD（MUX\_4\_1\_32）**





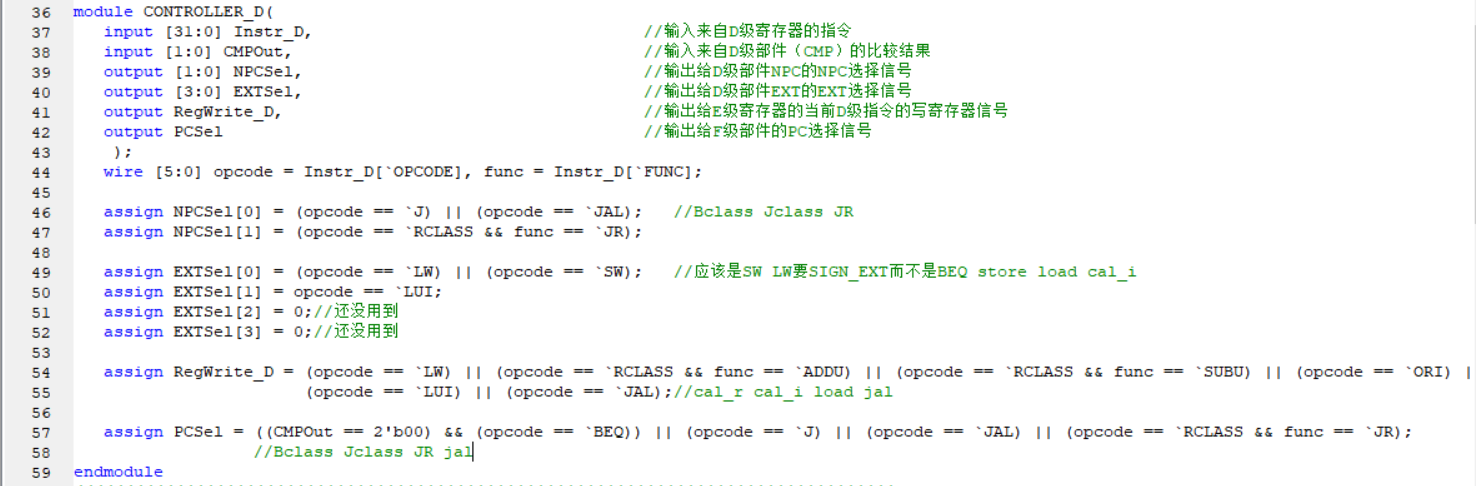
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| GRF\_RD2 | I | 输入GRF的读数据2 |
| MUXRFWDOut | I | 输入来自W级部件的GRF要写入的数据 |
| ALUOutput\_M | I | 输入来自M级寄存器的ALU运算结果 |
| ForwardRTD | I | 输入来自冲突单元的RTD转发信号 |
| MFRTDOut | O | 输出选择后的指令要用的GRF操作数2 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出转发选择后的GRF读数据2 | ForwardRTD：  00 GRF\_RD2  01 MUXRFWDOut  10 ALUOutput\_M |

1. **Controller\_D**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_D | I | 输入来自D级寄存器的指令 |
| CMPOut | I | 输入来自D级部件CMP的比较结果 |
| NPCSel | O | 输出给NPC的选择信号 |
| EXTSel | O | 输出给EXT的选择信号 |
| RegWrite\_D | O | 输出给E级寄存器的GRF写信号 |
| PCSel | O | 输出给F级部件MUXPC的选择信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择指令跳转类型 | 通过NPCSel选择指令跳转地址 |
| 2 | 选择是否跳转 | 通过PCSel选择是否跳转 |
| 3 | 选择立即数扩展类型 | 通过EXTSel选择扩展类型 |
| 4 | 判断指令是否进行写寄存器操作 | 通过RegWrite\_D进行判断 |

1. **Register\_E**
2. **模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 输入时钟信号 |
| Reset | I | 输入复位信号 |
| FlushE | I | 来自冲突单元的清空信号，实质上是Stall |
| Instr\_D | I | 输入来自D级寄存器的指令 |
| PCPlus4\_D | I | 输入来自D级寄存器的PC+4 |
| RSV\_D | I | 输入来自D级部件的RS的数据 |
| RTV\_D | I | 输入来自D级部件的RT的数据 |
| EXTOut\_D | I | 输入来自D级部件的EXT结果 |
| RegWrite\_D | I | 输入来自D级部件的GRF写信号 |
| RS\_D | I | 输入来自D级寄存器的RS地址 |
| RT\_D | I | 输入来自D级寄存器的RT |
| RD\_D | I | 输入来自D级寄存器的RD |
| Instr\_E | O | 输出E级的指令 |
| PCPlus4E | O | 输出E级的PC+4 |
| RSV\_E | O | 输出E级的RS对应的GRF读数据1 |
| RTV\_E | O | 输出E级的RD对应的GRF读数据2 |
| EXTOut\_E | O | 输出E级的EXT结果 |
| RegWrite\_E | O | 输出E的GRF写信号 |
| RS\_E | O | 输出E级的RS地址 |
| RT\_E | O | 输出E的RT地址 |
| RD\_E | O | 输出E的RD地址 |

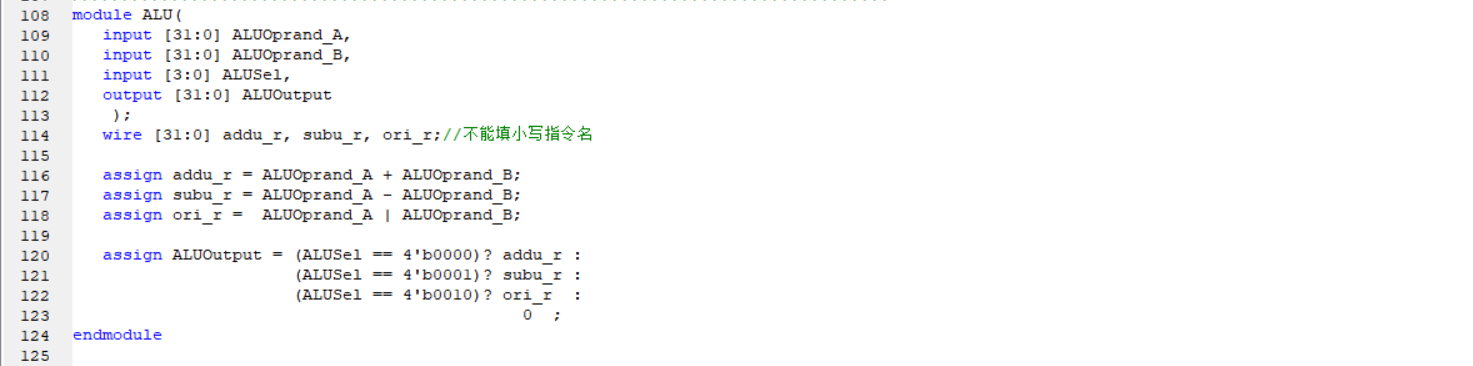
1. **PIPELINE\_E**
2. **模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_E | I | 输入来自E级寄存器的指令 |
| RTV\_E | I | 输入来自E级寄存器的GRF读数据2 |
| RSV\_E | I | 输入来自E级寄存器的GRF读数据1 |
| EXTOut\_E | I | 输入来自E级寄存器的扩展数 |
| MUXRFWDOut | I | 输入来自W级部件的GRF写数据，是转发所需数据 |
| ALUOutput\_M | I | 输入来自M级寄存器的ALU运算结果，是转发所需数据 |
| ForwardRSE | I | 输入来自冲突单元的RSE转发选择信号 |
| ForwardRTE | I | 输入来自冲突单元的RTE转发选择信号 |
| PCPlus4\_E | I | 输入来自E级寄存器的PC+4，也是jal类指令要写入31寄存器的数 |
| ALUOutput | O | 输出ALU运算结果 |
| WriteRd\_E | O | 输出寄存器写地址 |
| WriteData\_E | O | 输出内存写数据 |

1. **功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | ALU运算 | 通过ALU进行指令要求的运算 |
| 2 | 选择ALU输出，ALU输出可作为load store类指令的地址，可作为cal运算的结果，可作为jal指令的写入值 | 如果是JAL类对31号寄存器进行地址写入的指令，置为PCPlus4\_E+4，否则如果是正常的ALU运算，就置为ALU运算结果 |
| 3 | 选择写入寄存器地址 | 如果是三寄存器操作指令，置为rd，如果是二寄存器操作指令，置为rt，如果是jal类指令，置为31 |
| 4 | 选择写入内存数据，仅仅针对store类指令 | store类指令要写入内存的数据可在E级部件通过转发获取 |

1. **内部部件**
2. **ALU**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ALUOprand\_A | I | ALU操作数A |
| ALUOprand\_B | I | ALU操作数B |
| ALUSel | I | ALU选择信号 |
| ALUOutput | O | ALU运算结果 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加运算 | ALUOutput=ALUOprand\_A+ALUOprand\_B |
| 2 | 减运算 | ALUOutput=ALUOprand\_A-ALUOprand\_B |
| 3 | 或运算 | ALUOutput=ALUOprand\_A|ALUOprand\_B |

1. **Controller\_E**



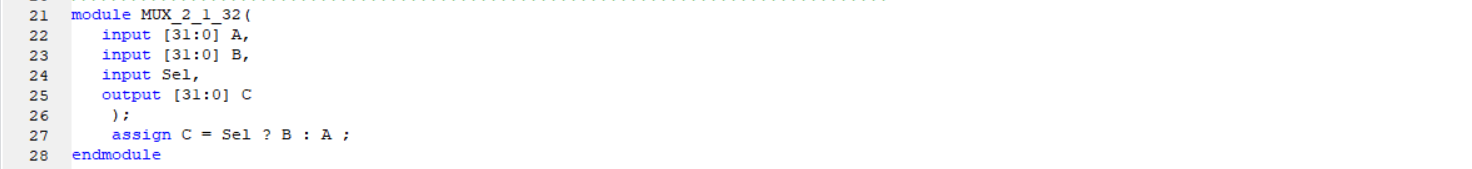
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_E | I | 输入来自E级寄存器的指令 |
| ALUSel | O | 输出ALU操作选择信号 |
| MUXALUBSel | O | 输出ALUB功能选择信号，选择ALU操作数B是寄存器数还是立即数 |
| RegDst | O | 输出当前指令的写寄存器选择信号 |
| ALUOutputSel | O | 输出ALU的结果选择信号，是jal型对应的PC+8还是正常的ALU运算结果 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU运算类型 | 通过ALUSel选择当前E级指令对应的ALU运算 |
| 2 | 选择ALU运算数 | 通过MUXALUBSel选择当前E级指令对应的ALU运算数是寄存器数还是立即数 |
| 3 | 选择写寄存器地址 | 通过RegDst选择当前E级指令写入寄存器的地址是rt 还是rd还是31 |
| 4 | 选择ALU输出 | 通过ALUOutputSel选择当前E级指令要用的ALU输出是PC+8（JAL）还是正常的运算结果（正常运算指令） |

1. **MUXALUB**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| MFRTEOut | I | 输入ALUB转发多选器的选择结果，为寄存器操作数 |
| EXTOut\_E | I | 输入来自E级寄存器的扩展数 |
| MUXALUBSel | I | 输入来自E级控制器的MUXALUB的选择信号 |
| MUXALUBOut | O | 输出ALUB |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU的第二个操作数 | 当MUXALUBSel为1时选择立即数，为0时选择寄存器操作数 |

1. **MFRSE**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| RSV\_E | I | 来自E级寄存器的的GEF读数据1 |
| MUXRFWDOut | I | 来自W级部件的GRF写回数据 |
| ALUOutput\_M | I | 来自M级寄存器的ALU的输出 |
| ForwardRSE | I | 来自E级控制器的MFRSE的选择信号 |
| MFRSEOut | O | 输出MFRSE选择后的数据 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU的第一个操作数 | ForwardRSE:  00 RSV\_E  01 MUXRFWDOut  10 ALUOutput\_M |

1. **MFRTE**





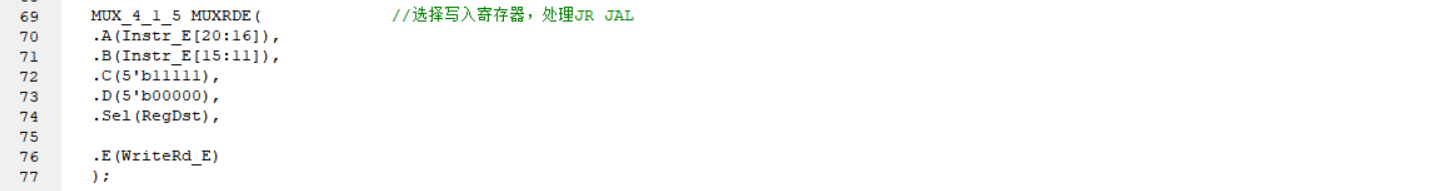
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| RTV\_E | I | 来自E级寄存器的GRF的读数据2 |
| MUXRFWDOut | I | 来自W级寄存器的GRF写回数据 |
| ALUOutput\_M | I | 来自M级寄存器的ALU的输出 |
| ForwardRTE | I | 来自E级控制器的MFRTE的选择信号 |
| MFRTEOut | O | 输出MFRTE选择后的数据 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU的第二个操作数中的寄存器数 | ForwardRTE:  00 RTV\_E  01 MUXRFWDOut  10 ALUOutput\_M |

**6）MUXRDE**





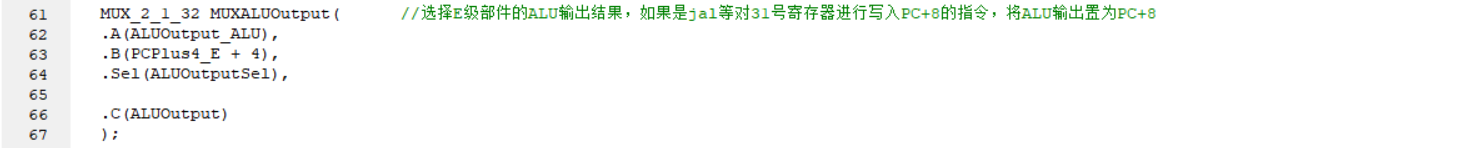
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| RT\_E | I | 来自E级寄存器的GRF写地址1 |
| RD\_E | I | 来自E级寄存器的GRF写地址2 |
| 5’b11111 | I | 31号寄存器 |
| RegDst | I | 来自E级控制器的写寄存器选择信号 |
| WriteRd\_E | O | 输出E级当前指令要写的寄存器地址 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择当前指令要写入的寄存器地址 | RegDst:  00 RT\_E  01 RD\_E  10 31 |

**7）MUXALUOutput**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ALUOutput\_ALU | I | 输入来自ALU的运算结果 |
| PCPlus4\_E+4 | I | 输入来自E级寄存器的PCPlus4\_E+4，作为JAL指令对应要输入的值 |
| ALUOutputSel | I | 输入来自E级寄存器E级的ALU输出选择信号 |
| ALUOutput | O | 输出E级的ALU运算结果，其可能是正常指令的运算结果，也可能是对应JAL指令的PC+8 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择E级部件的ALU运算输出是正常ALU运算指令的运算结果还是JAL指令对应的PC+8 | ALUOutputSel  0 ALUOutput\_ALU  1 PCPlus4\_E+4 |

1. **Register\_M**

**（1）模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 输入时钟信号 |
| Reset | I | 输入复位信号 |
| Instr\_E | I | 输入来自E级寄存器的指令 |
| RTV\_E | I | 输入来自E级部件的要写入内存的数据 |
| PCPlus4\_E | I | 输入来自E级寄存器的的PC+4 |
| ALUOutput\_E | I | 输入来自E级部件的ALU运算结果 |
| RegWrite\_E | I | 输入来自E级寄存器的GRF写信号 |
| WriteRd\_E | I | 输入来自E级部件的当前指令对应的写寄存器地址 |
| RegWrite\_M | O | 输出M级寄存器的GRF写信号 |
| Instr\_M | O | 输出M级寄存器的指令 |
| RTV\_M | O | 输出M级寄存器的要写入内存的数据 |
| PCPlus4\_M | O | 输出M级寄存器的PC+4 |
| ALUOutput\_M | O | 输出M级寄存器的ALU运算结果 |
| RD\_M | O | 输出M级寄存器的写寄存器地址 |

1. **PIPELINE\_M**
2. **模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 输入时钟信号 |
| Reset | I | 输入复位信号 |
| ALUOutput\_M | I | 输入M级寄存器的ALU运算结果，作为存取内存的地址 |
| RTV\_M | I | 输入来自M级寄存器的写入内存数据 |
| Instr\_M | I | 输入来自M级寄存器的指令 |
| ForwardRTM | I | 输入来自冲突单元的RTM转发控制信号 |
| PCPlus4\_M | I | 输入来自M级寄存器的PC+4，用来display |
| MUXRFWDOut | I | 输入来自W级部件的GRF写入数据，是转发所需数据 |
| MemOut | O | 输出M级部件读出的内存数据 |

1. **功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读内存 | 根据输入的ALUOutput\_M作为内存地址读出内存中的数据 |
| 2 | 写内存 | 当写内存信号有效且在时钟上升沿将数据写入对应的内存地址 |
| 3 | 复位 | 当Reset信号有效且时钟上升沿到来时将所有内存数据置零 |

1. **内部部件**
2. **DM**



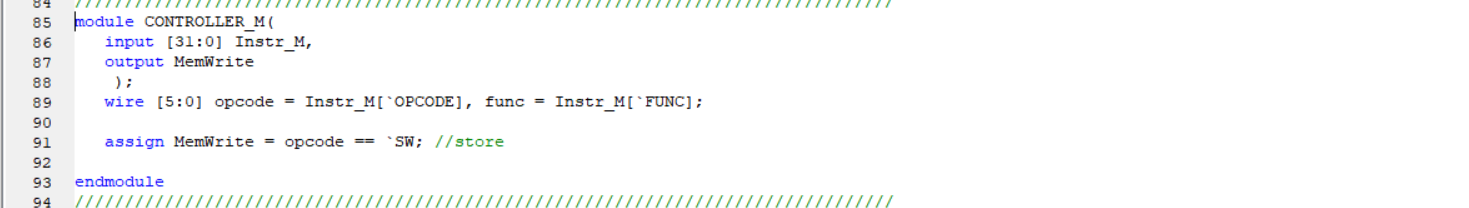
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 输入时钟信号 |
| Reset | I | 输入复位信号 |
| MemWrite | I | 输入来自M级控制器的内存写使能信号 |
| MemAddr | I | 输入来自M级寄存器额内存写地址，即ALU运算结果 |
| MemWD | I | 输入来自MFRTM的内存写数据 |
| PCPlus4\_M | I | 输入来自M级寄存器的PC+4，用来display |
| MemOut | O | 输出内存读数据 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读内存 | 根据输入的ALUOutput\_M作为内存地址读出内存中的数据 |
| 2 | 写内存 | 当写内存信号有效且在时钟上升沿将数据写入对应的内存地址 |
| 3 | 复位 | 当Reset信号有效且时钟上升沿到来时将所有内存数据置零 |

1. **Controller\_M**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_M | I | 来自M级的指令 |
| MemWrite | O | 输出内存写使能信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 判断是否写内存 | 根据当前的M级指令输出MemWrite信号控制是否写内存 |

1. **MFRTM**



**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| RTV\_M | I | 输入来自M级寄存器的GRF读数据2 |
| MUXRFWDOut | I | 输入来自W级部件的GRF写回数据，是转发所需数据 |
| ForwardRTM | I | 输入来自冲突单元的MFRTM选择信号 |
| MFRTMOut | O | 输出MFRTM的转发选择结果，作为写入内存的数据 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择写入内存的数据 | ForwardRTM:  0 RTV\_M  1 MUXRFWDOut |

1. **Register\_W**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| CLK | I | 输入时钟信号 |
| Reset | I | 输入复位信号 |
| Instr\_M | I | 输入来自M级寄存器的指令 |
| PCPlus4\_M | I | 输入来自M级寄存器的PC+4 |
| ALUOutput\_M | I | 来自M级寄存器的ALU输出 |
| ReadData\_M | I | 输入来自M级部件的内存读数据 |
| RD\_M | I | 输入来自M级寄存器的GRF写地址 |
| RegWrite\_M | I | 输入来自M级寄存器的GRF写信号 |
| Instr\_W | O | 输出W级寄存器的指令 |
| PCPlus4\_W | O | 输出W级寄存器的PC+4 |
| ALUOutput\_W | O | 输出W级寄存器的ALU输出 |
| ReadData\_W | O | 输出W级寄存器的内存读数据 |
| RD\_W | O | 输出W级寄存器的GRF写地址地址 |
| RegWrite\_W | O | 输出W级寄存器的GRF写信号 |

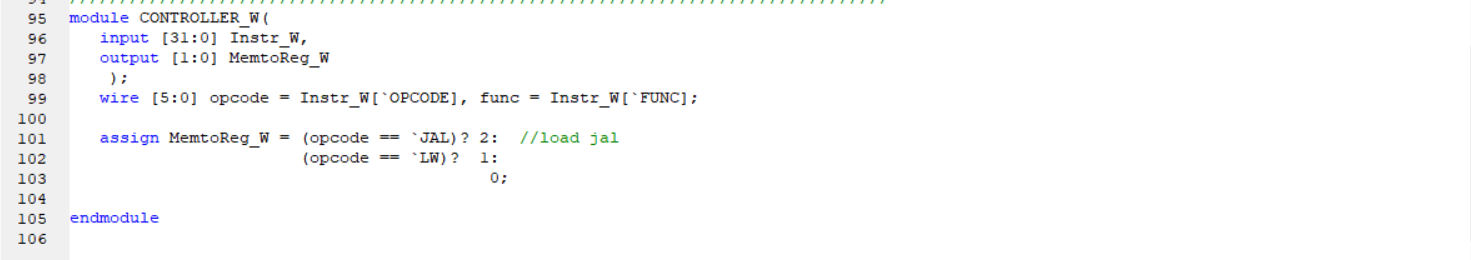
1. **PIPELINE\_W**
2. **模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_W | I | 输入来自W级寄存器的指令 |
| ReadData\_W | I | 输入来自W级寄存器的内存读数据 |
| ALUOutput\_W | I | 输入来自W级寄存器的ALU运算结果 |
| PCPlus4\_W | I | 输入来自W级寄存器的PC+4 |
| MFRDWDOut | O | 输出经过选择后的GRF要写入的数据 |

1. **功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择写入GRF的数据 | MemtoReg\_W:  0 ALUOutput\_W  1 ReadData\_W |

1. **内部部件**
2. **Controller\_W**



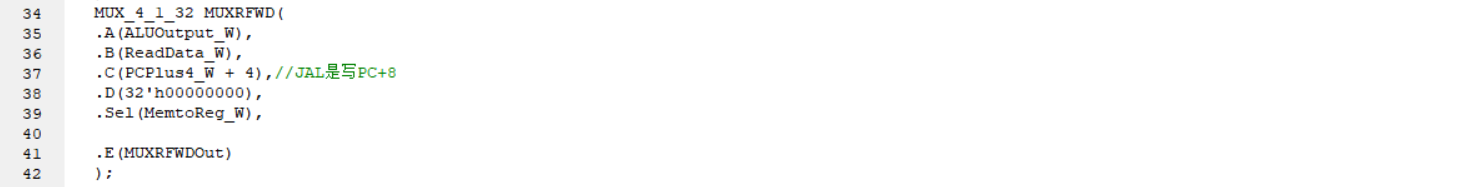
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_W | I | 来自W级的指令 |
| MemtoReg\_W | O | MUXRFWD的选择信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出选择写入GRF的数据的控制信号 | load指令1，其他指令0 |

1. **MUXRFWD**





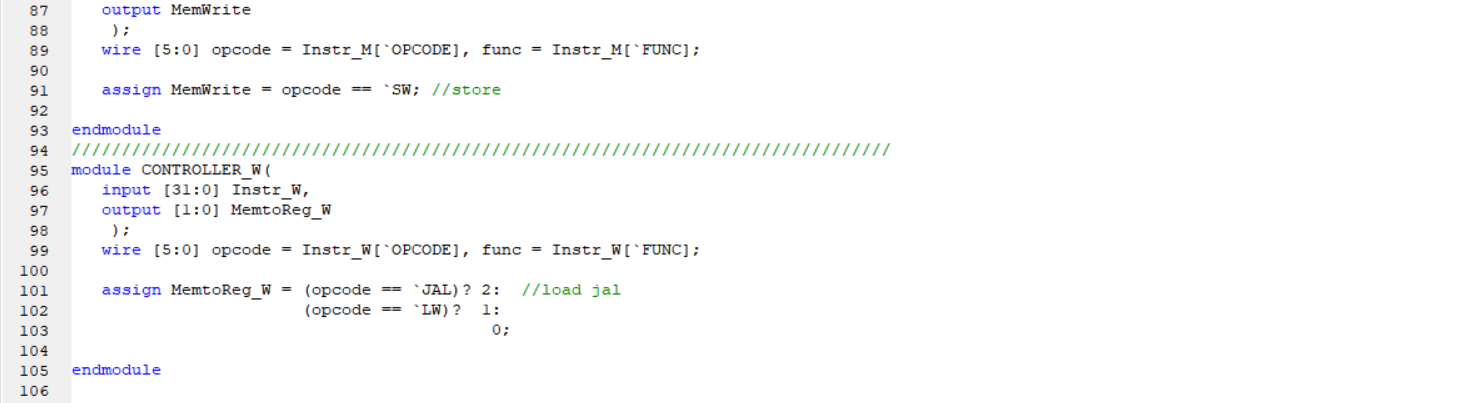
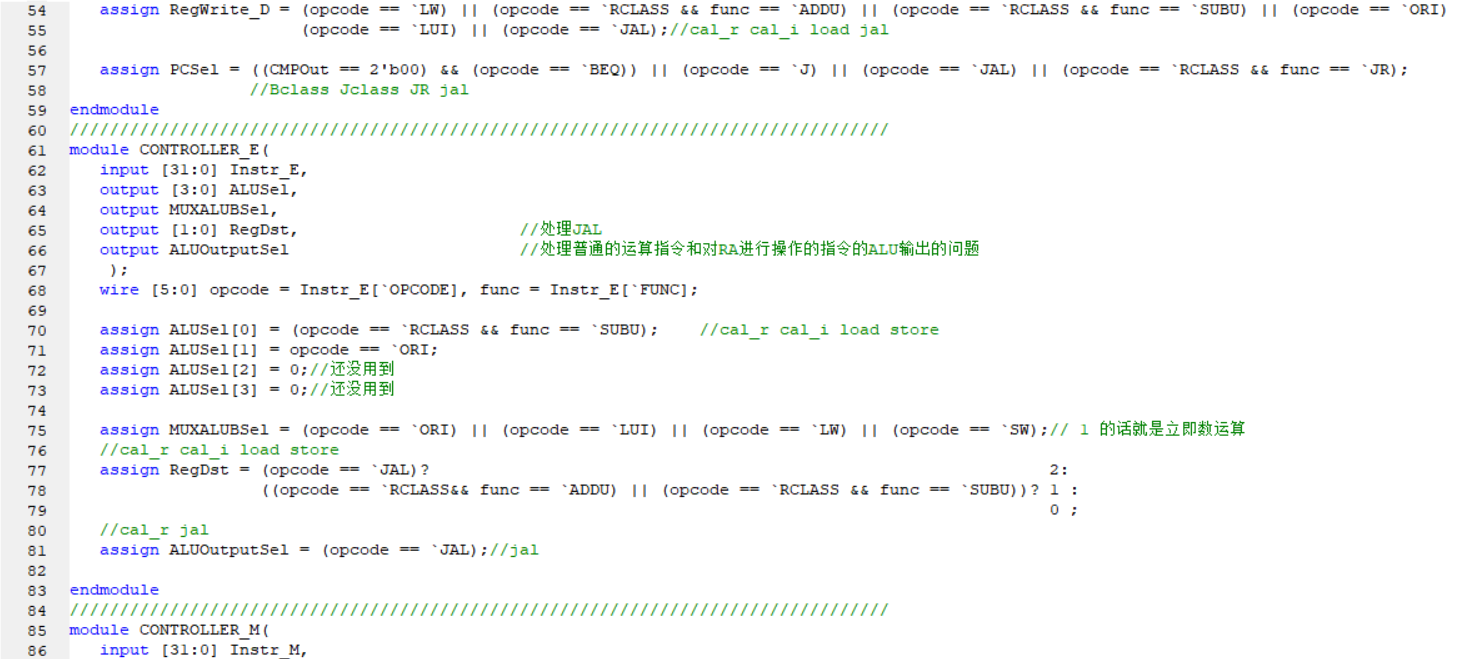
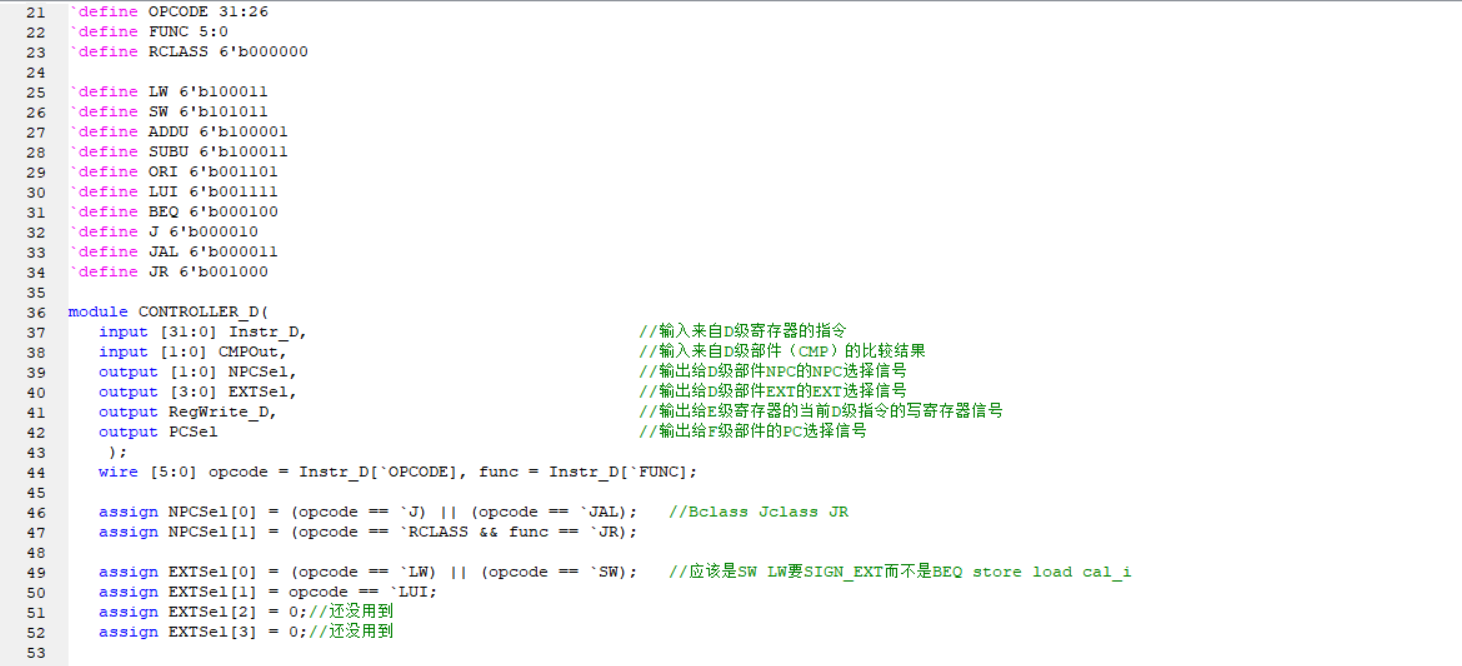
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ReadData\_W | I | 来自W级的内存数据 |
| ALUOutput\_W | I | 来自W级的ALU输出 |
| MemtoReg\_W | I | MUXRFWD选择信号 |
| MUXRFWDOut | O | 输出寄存器写入数据 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择写入GRF的数据 | MemtoReg\_W:  0 ALUOutput\_W  1 ReadData\_W |

1. **CONTROLLER**



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **op**  **code** | **func** | **NPC**  **Sel** | **EXT**  **Sel** | **PC**  **Sel** | **Reg**  **Write** | **Reg**  **Dst** | **ALU**  **Out**  **put**  **Sel** | **ALU**  **Sel** | **MUXALUB**  **Sel** | **Mem**  **Write** | **Mem**  **toReg** |
| **addu** | **000000** | **100001** | **X** | **X** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **subu** | **000000** | **100011** | **X** | **X** | **0** | **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **ori** | **001101** | **N/A** | **X** | **0** | **0** | **1** | **0** | **0** | **2** | **1** | **0** | **0** |
| **lw** | **100011** | **N/A** | **X** | **1** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **1** |
| **sw** | **101011** | **N/A** | **X** | **1** | **0** | **0** | **X** | **X** | **0** | **1** | **1** | **0** |
| **beq** | **000100** | **N/A** | **0** | **X** | **1** | **0** | **X** | **X** | **0** | **0** | **0** | **0** |
| **j** | **000010** | **N/A** | **1** | **X** | **1** | **0** | **X** | **X** | **0** | **X** | **0** | **0** |
| **jr** | **000000** | **001000** | **2** | **X** | **1** | **0** | **X** | **X** | **0** | **X** | **0** | **0** |
| **jal** | **000011** | **N/A** | **1** | **X** | **1** | **1** | **2** | **1** | **0** | **X** | **0** | **0** |
| **lui** | **001111** | **N/A** | **X** | **2** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** |

**（1）Controller\_D**

**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_D | I | 输入来自D级寄存器的指令 |
| CMPOut | I | 输入来自D级部件CMP的比较结果 |
| NPCSel | O | 输出给NPC的选择信号 |
| EXTSel | O | 输出给EXT的选择信号 |
| RegWrite\_D | O | 输出给E级寄存器的GRF写信号 |
| PCSel | O | 输出给F级部件MUXPC的选择信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择指令跳转类型 | 通过NPCSel选择指令跳转地址 |
| 2 | 选择是否跳转 | 通过PCSel选择是否跳转 |
| 3 | 选择立即数扩展类型 | 通过EXTSel选择扩展类型 |
| 4 | 判断指令是否进行写寄存器操作 | 通过RegWrite\_D进行判断 |

**（2）Controller\_E**

**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_E | I | 输入来自E级寄存器的指令 |
| ALUSel | O | 输出ALU操作选择信号 |
| MUXALUBSel | O | 输出ALUB功能选择信号，选择ALU操作数B是寄存器数还是立即数 |
| RegDst | O | 输出当前指令的写寄存器选择信号 |
| ALUOutputSel | O | 输出ALU的结果选择信号，是jal型对应的PC+8还是正常的ALU运算结果 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU运算类型 | 通过ALUSel选择当前E级指令对应的ALU运算 |
| 2 | 选择ALU运算数 | 通过MUXALUBSel选择当前E级指令对应的ALU运算数是寄存器数还是立即数 |
| 3 | 选择写寄存器地址 | 通过RegDst选择当前E级指令写入寄存器的地址是rt 还是rd还是31 |
| 4 | 选择ALU输出 | 通过ALUOutputSel选择当前E级指令要用的ALU输出是PC+8（JAL）还是正常的运算结果（正常运算指令） |

**（3）Controller\_M**

**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_M | I | 来自M级的指令 |
| MemWrite | O | 输出内存写使能信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 判断是否写内存 | 根据当前的M级指令输出MemWrite信号控制是否写内存 |

**（4）Controller\_W**

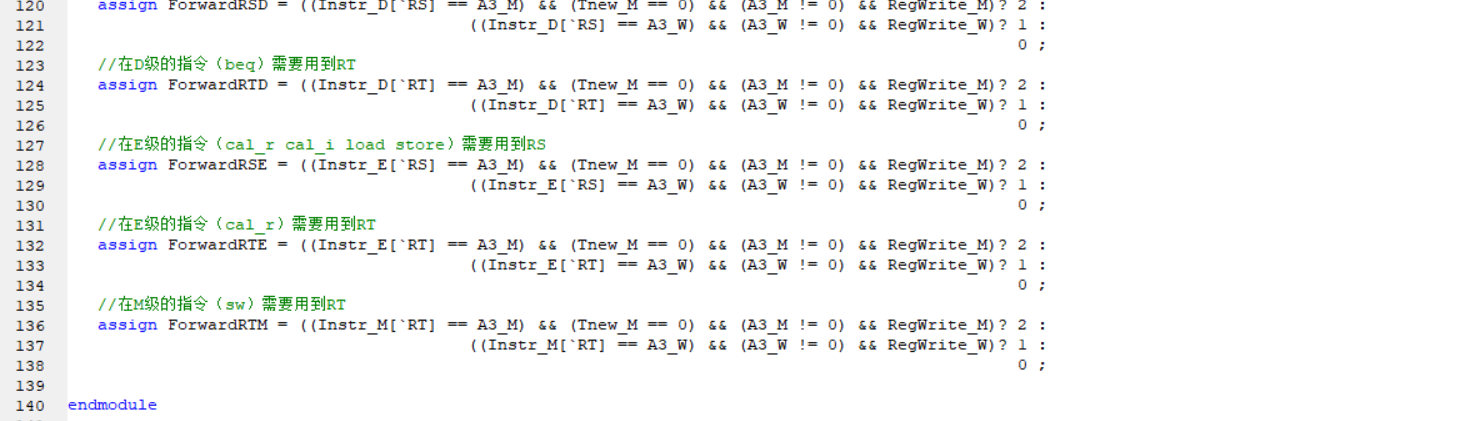
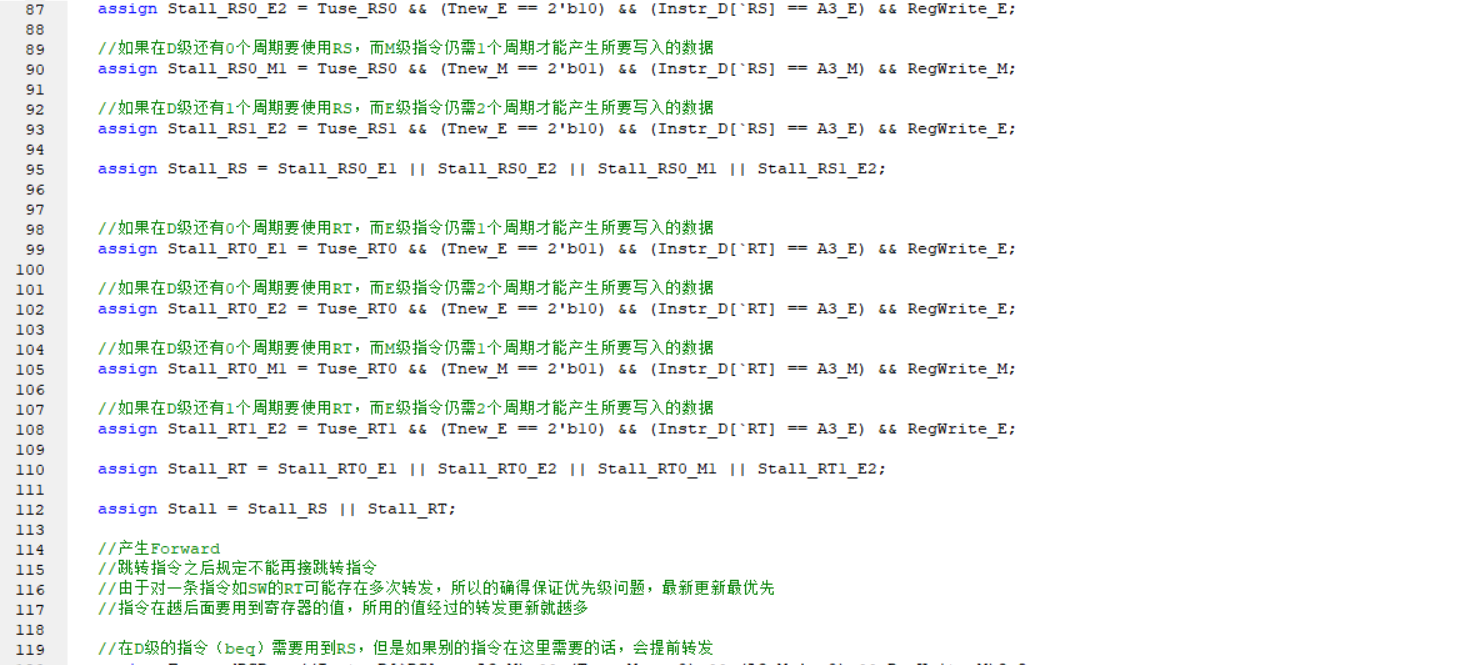
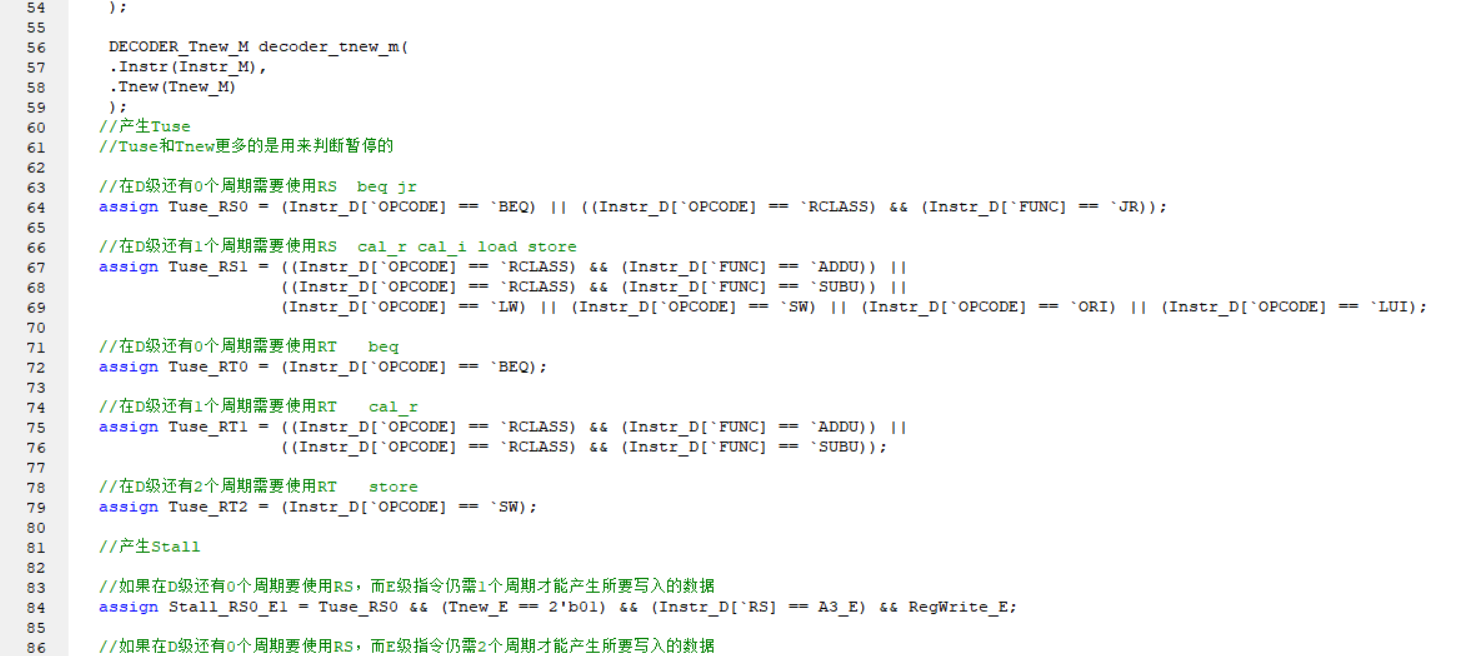
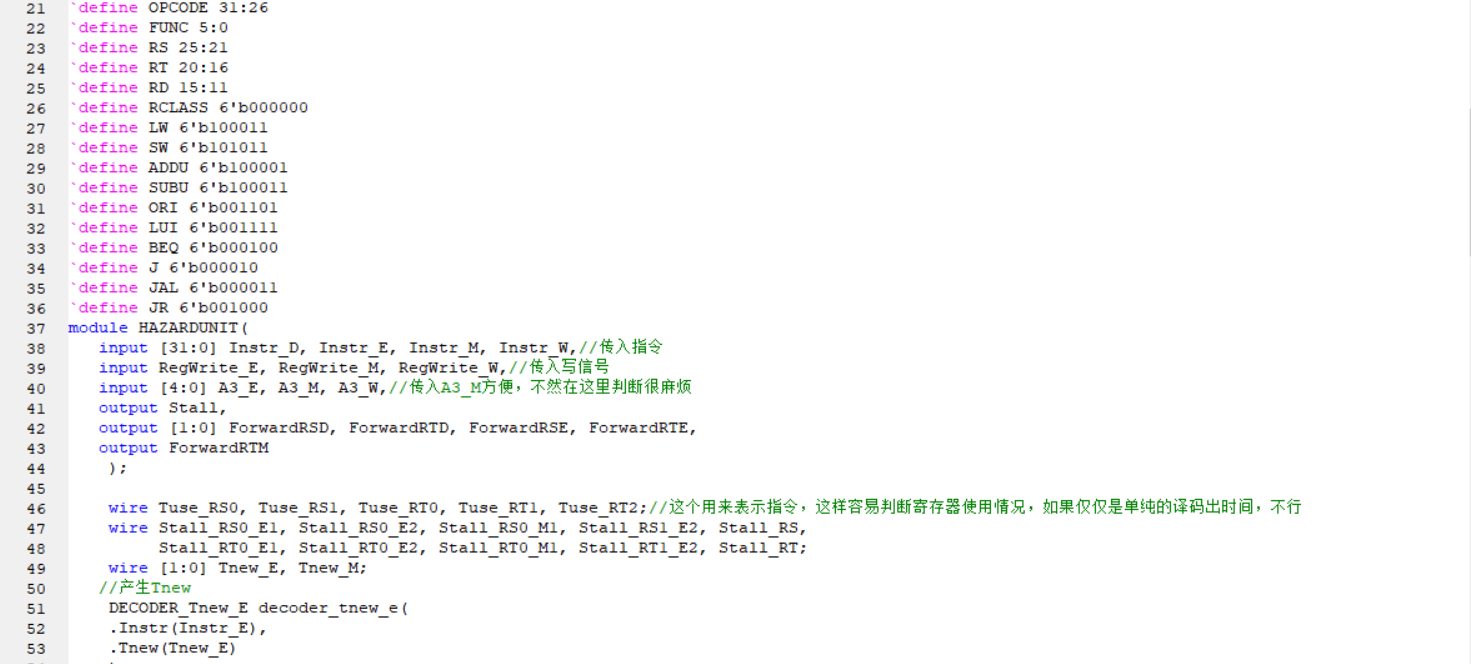
**①模块接口**

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_W | I | 来自W级的指令 |
| MemtoReg\_W | O | MUXRFWD的选择信号 |

**②功能定义**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出选择写入GRF的数据的控制信号 | load指令1，其他指令0 |

1. **HAZARDUNIT**



1. **模块接口**

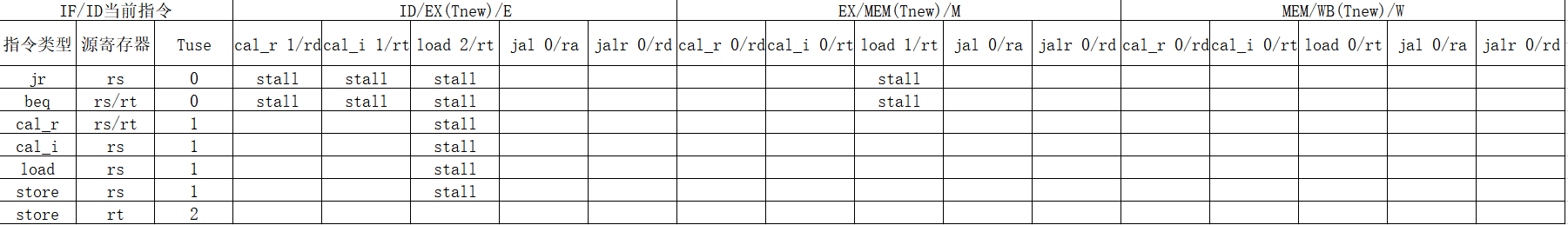
|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr\_D | I | 输入来自D级寄存器的指令 |
| Instr\_E | I | 输入来自E级寄存器的指令 |
| Instr\_M | I | 输入来自M级寄存器的指令 |
| Instr\_W | I | 输入来自W级寄存器的指令 |
| RegWrite\_E | I | 输入来自E级的写使能信号 |
| RegWrite\_M | I | 输入来自M级的写使能信号 |
| RegWrite\_W | I | 输入来自W级的写使能信号 |
| A3\_E | I | 输入来自E级的写地址 |
| A3\_M | I | 输入来自M级的写地址 |
| A3\_W | I | 输入来自W级的写地址 |
| Stall | O | 输出暂停信号 |
| ForwardRSD | O | 输出RSD转发控制信号 |
| ForwardRTD | O | 输出RTD转发控制信号 |
| ForwardRSE | O | 输出RSE转发控制信号 |
| ForwardRTE | O | 输出RTE转发控制信号 |
| ForwardRTM | O | 输出RTM转发控制信号 |

1. **功能定义**

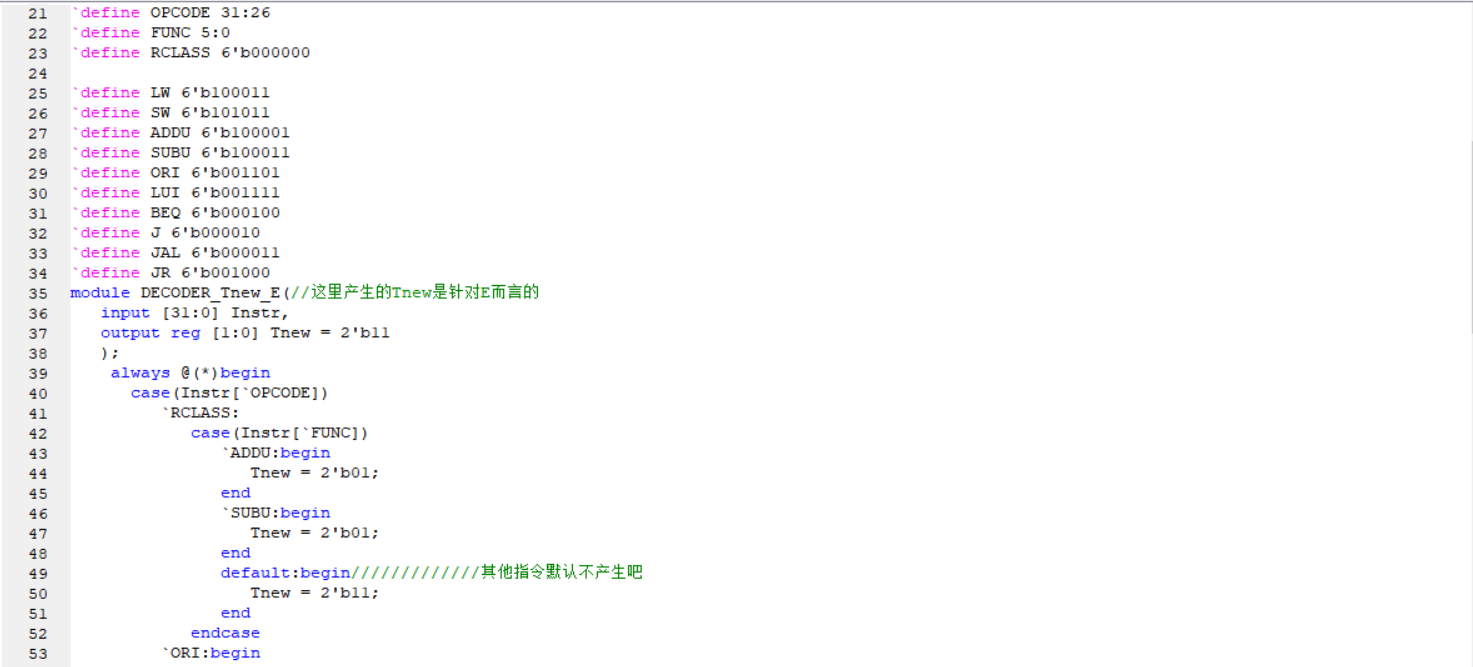
|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 判断暂停 | 当前D级指令所需使用的寄存器的时间周期大于E、M、W级相同的寄存器写入指令的产生写入数据时间周期时，需要暂停。有两类寄存器和八种情况  RS0\_E1 RS0\_E2  RS0\_M1 RS1\_E2  RT0\_E1 RT0\_E2  RT0\_M1 RT1\_E2 |
| 2 | 判断转发 | 当前D级指令所需使用的寄存器的时间周期小于或等于E、M、W级相同的寄存器写入指令的产生写入数据时间周期时，可以转发。有两类寄存器和五种情况  FRSD FRTD  FRSE FRTE  FRTM |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | cal\_r | | cal\_i | | store | load |  |  |  |  |  |
| addu | subu | ori | lui | sw | lw | beq | jr | jal | nop | j |
| Tuse\_RS0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | X | X | X |
| Tuse\_RS1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | X | X | X |
| Tuse\_RT0 | 0 | 0 | X | X | 0 | X | 1 | X | X | X | X |
| Tuse\_RT1 | 1 | 1 | X | X | 0 | X | 0 | X | X | X | X |
| Tuse\_RT2 | 0 | 0 | X | X | 1 | X | 0 | X | X | X | X |
| Tnew | 1 | 1 | 1 | 1 | 3 | 2 | 3 | 3 | 3 | 3 | 3 |

注：Tnew=3 代表不产生写入寄存器数据



1. **内部部件DECODER**



1. **MUX**



1. **测试程序**

ori $0,$0,0 #测试ori

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 #测试lui

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 #测试冲突

subu $t1,$t1,$t2

ori $t1,$0,8

sw $t1,0($0) #测试冲突

lw $t1,0($0)

ori $t1,$0,12 #测试冲突

sw $t1,-4($t1) #测试冲突

sw $t1,4($t1)

ori $t1,$0,4

loop1:

lw $t1,4($t1) #测试冲突

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

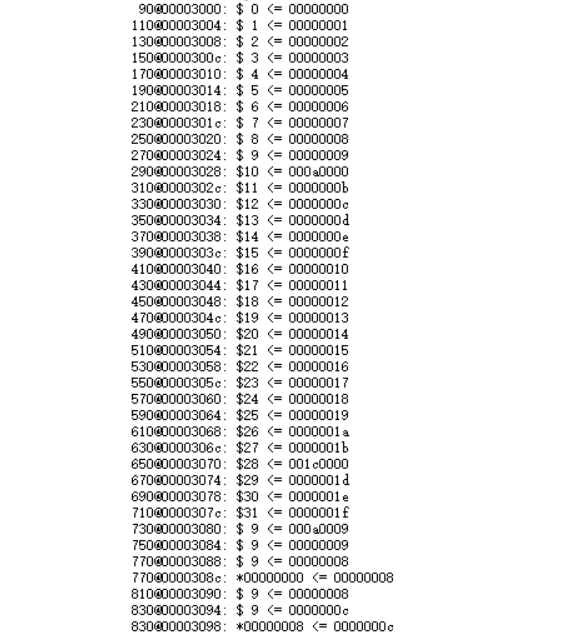
subu $t3,$t3,$t4

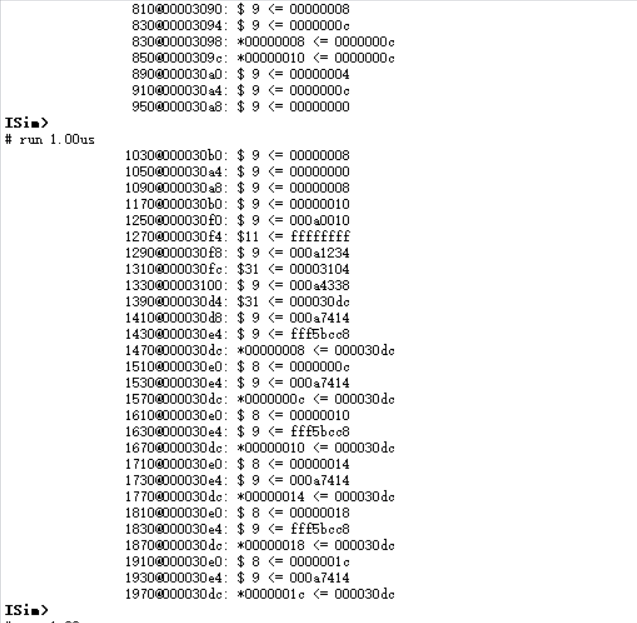
ori $t1,$t1,0x00001234

jal a3

addu $t1,$ra,$t1

期望输出：





该程序最后是将所有DM中的值都赋为0x000030dc的死循环

**四、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**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 类型 | Tuse\_RS | Tuse\_RT | Tnew\_E | 冲突寄存器 | 解决方案 | 测试程序 |
| 1 | R-J | 0 |  | 1 | RS | 暂停1个时钟周期后从M到D转发 | ori $ra $0,0x0000300c  jr $ra |
| 2 | L-J | 0 |  | 2 | RS | 暂停2个时钟周期后从W到D转发 | lw $ra,0($0)  jr $ra |
| 3 | R-B | 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 |
| 4 | L-B | 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 |
| 5 | R-R | 1 | 1 | 1 | RS或RT | 从M到E转发 | addu $t1,$t1,$t2  subu $t1,$t1,$t2 |
| 6 | L-R | 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 |
| 7 | R-R | 1 |  | 1 | RS | 从M到E级转发 | ori $t1,$0,1  lui $t1,1 |
| 8 | L-R | 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 |
| 9 | R-S | 1 | 2 | 1 | RS | 从M到E级转发 | lui $t1,1  sw $t1,0($0) |
| 10 | L-S | 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) |
| 11 | L-S | 1 | 2 | 2 | RT | 从W到M级转发 | lw $t1,0($0)  sw $t1,0($0) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 类型 | Tuse\_RS | Tuse\_RT | Tnew\_M | 冲突寄存器 | 解决方案 | 测试程序 |
| 1 | L-R-J | 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 |
| 2 | L-R-B | 0 | 0 | 1 | RT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 |
| 3 | L-R-R | 1 | 1 | 1 | RS或RT | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  addu $t1,$t3,$t1 |
| 4 | L-R-R | 1 |  | 1 | RS | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  ori $t1,$t1,1 |
| 5 | L-R-S | 1 | 2 | 1 | RS | 从W到E级转发 | lw $t1,0($0)  lui $t2,1  sw $t1,0($0) |
| 6 | R-R-J | 0 |  | 0 | RS | 从M级到D级转发 | addu $ra,$t1,$t2  lui $t2,1  jr $ra |
| 7 | R-R-B | 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 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 类型 | Tuse\_RS | Tuse\_RT | Tnew\_W | 冲突寄存器 | 解决方案 | 测试程序 |
| 1 | R-N-N-J | 0 |  | 0 | RS | 从W到D级转发 | addu $ra,$t1,$t2  nop  nop  jr $ra |
| 2 | R-N-N-B | 0 | 0 | 0 | RS或RT | 从W到D级转发 | addu $t1,$t1,$t2  nop  nop  beq $t1,$t2,loop |