MIPS流水线处理器设计文档

MIPS流水线处理器设计文档由顶层模块、数据通路、控制器、冲突模块、测试程序组成。

1. 顶层模块mips.v

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 功能描述 |
| 1 | clk | 1 | I | pc、grf、dm的时钟信号 |
| 2 | reset | 1 | I | 当reset=1且时钟为上跳沿时，pc被设置为0x00003000，gpr所有寄存器置零，dm所有数据置零 |

顶层模块将数据通路模块datapath.v与控制器模块control.v,hazard.v相连。

1. 数据通路datapath.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | reset | 1 | I | 复位信号 |
| 2 | clk | 1 | I | 时钟信号 |
| 3 | PCOP | 2 | I | muxPCOP模块控制信号 |
| 4 | enable | 1 | I | PC,IF\_ID使能端 |
| 5 | clear | 1 | I | ID\_EX清零信号 |
| 6 | RegDst | 2 | I | muxRegDst模块控制信号 |
| 7 | ExtOP | 1 | I | Ext模块控制信号 |
| 8 | RegWrite | 1 | I | 寄存器文件写使能 |
| 9 | RegWData | 1 | I | muxRegWData模块控制信号 |
| 10 | ALUSrc | 1 | I | muxALUSrc模块控制信号 |
| 11 | ALUOP | 4 | I | 选择ALU运算类型的信号 |
| 12 | MemWrite | 1 | I | dm写入控制信号 |
| 13 | Instr\_original | 32 | O | IM输出指令 |
| 14 | Instr\_D | 32 | O | D级指令 |
| 15 | Instr\_E | 32 | O | E级指令 |
| 16 | Instr\_W | 32 | O | W级指令 |
| 17 | MF\_RS\_D\_OP | 2 | O | MF\_RS\_D模块控制信号 |
| 18 | MF\_RT\_D\_OP | 2 | O | MF\_RT\_D模块控制信号 |
| 19 | MF\_RS\_E\_OP | 2 | O | MF\_RS\_E模块控制信号 |
| 20 | MF\_RT\_E\_OP | 2 | O | MF\_RT\_E模块控制信号 |
| 21 | MF\_RT\_M\_OP | 2 | O | MF\_RT\_M模块控制信号 |
| 22 | Equal\_out | 1 | O | equal模块输出信号 |
| 23 | bgez\_out | 1 | O | bgez\_out信号 |
| 24 | bgtz\_out | 1 | O | bgtz\_out信号 |
| 25 | bltz\_out | 1 | O | bltz\_out信号 |
| 26 | blez\_out | 1 | O | blez\_out信号 |
| 27 | busy | 1 | O | Busy 信号 |

datapath.v将pc.v, dm.v , im.v , add4.v,npc.v,alu.v , grf.v , ext.v , muxPCOP.v , muxRegWData.v , muxRegDst.v , muxALUSrc.v 等模块连接在一起。下面将介绍这些模块。

1. grf.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | RA1 | 5 | I | 读寄存器文件时第一个寄存器的下标 |
| 2 | RA2 | 5 | I | 读寄存器文件时第二个寄存器的下标 |
| 3 | Waddr | 5 | I | 写入文件的寄存器下标 |
| 4 | WData | 32 | I | 寄存器文件写入数据 |
| 5 | clk | 1 | I | 时钟信号 |
| 6 | reset | 1 | I | 复位信号 |
| 7 | RegWrite | 1 | I | 寄存器文件写使能 |
| 8 | RData1 | 32 | O | 读寄存器文件时第一个寄存器的输出 |
| 9 | RData2 | 32 | O | 读寄存器文件时第二个寄存器的输出 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读寄存器 | RData1输出RA1所寻址的寄存器的文件数据，RData2输出RA2所寻址的寄存器的文件数据 |
| 2 | 写寄存器 | 当时钟上升沿到来时，并且reset=0且RegWrite有效时，WD被写入WA所寻址的寄存器 |
| 3 | 复位 | reset=1 且 clk为上升沿时gpr所有寄存器置零 |

1. alu.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALU\_A | 32 | I | ALU的第一个操作数 |
| 2 | ALU\_B | 32 | I | ALU的第二个操作数 |
| 3 | ALUOP | 4 | I | 选择ALU运算类型的信号 |
| 4 | ALU\_C | 32 | O | ALU的运算结果 |
| 5 | equal | 1 | O | 判断ALU\_A与ALU\_B是否相等，若相等输出1，否则输出0 |
| 6 | s | 5 | I | 逻辑左移、右移位数 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加法 | 输出端ALU\_C=ALU\_A+ALU\_B |
| 2 | 减法 | 输出端ALU\_C=ALU\_A-ALU\_B |
| 3 | 或 | 输出端ALU\_C=ALU\_A|ALU\_B |
| 4 | 判断相等 | 若ALU\_A=ALU\_B，equal信号输出1，否则输出0 |
| 5 | 置高位 | 输出端ALU\_C={{ALU\_B[15:0]},{16{1'b0}}} |
| 6 | 逻辑左移 | 输出端ALU\_C=ALU\_A<<s |
| 7 | 逻辑右移 | 输出端ALU\_C=ALU\_A>>s |
| 8 | 与 | 输出端ALU\_C=ALU\_A&ALU\_B |
| 9 | 异或 | 输出端ALU\_C=ALU\_A^ALU\_B |
| 10 | nor | 输出端ALU\_C=ALU\_A nor ALU\_B |
| 11 | 算术右移 | 输出端ALU\_C={{32{ALU\_B[31]}},ALU\_B}>>s |
| 12 | 算术可变右移 | 输出端ALU\_C={{32{ALU\_B[31]}},ALU\_B}>>ALU\_A[4:0] |
| 13 | 逻辑可变左移 | 输出端ALU\_C=ALU\_B<<ALU\_A[4:0] |
| 14 | 逻辑可变右移 | 输出端ALU\_C=ALU\_B>>ALU\_A[4:0] |
| 15 | 小于置一（有符号） | 输出端ALU\_C=$signed(ALU\_A)<$signed(ALU\_B)?32'b1:32'b0 |
| 16 | 小于置一（无符号） | 输出端ALU\_C=(ALU\_A<ALU\_B?32'b1:32'b0):32'b0 |

1. dm.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | addr | 32 | I | 写入数据储存器的数据的地址/读取数据储存器的数据的地址 |
| 2 | din | 32 | I | 写入数据储存器的数据 |
| 3 | MemWrite | 1 | I | 写入控制信号 |
| 4 | clk | 1 | I | 时钟信号 |
| 5 | reset | 1 | I | 复位信号 |
| 6 | dout | 32 | O | 从数据存储器读取的数据 |
| 7 | DMOP | 2 | I | Sb，sh，sw信号 |
| 8 | Be\_out | 4 | I | be模块输出 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数据储存器 | dout输出数据储存器在addr[11:2]地址的数据 |
| 2 | 写数据储存器 | 当MemWrite有效,reset=0,并且时钟为上升沿时，din被写入数据存储器地址为addr[11:2]的区域 |
| 3 | 复位 | reset=1并且时钟为上升沿时,DM所有数据置零 |

1. ext.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ext\_in | 16 | I | 位扩展输入信号 |
| 2 | EXTOP | 1 | I | 信号为1时，进行符号扩展，信号为0时，进行0扩展 |
| 3 | ext\_out | 32 | O | 位扩展输出信号 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 符号扩展 | 当EXTOP信号为1时，ext\_in符号扩展后由ext\_out输出 |
| 2 | 0扩展 | 当EXTOP信号为0时，在ext\_in前置16个0后由ext\_out输出 |

1. im.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | address | 32 | I | 指令地址 |
| 2 | instr | 32 | O | 指令 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 取指令 | 取出位于(address-0x00003000)[12:2]处指令 |

1. pc.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PC\_in | 32 | I | 下一条指令PC值 |
| 2 | clk | 1 | I | 时钟信号 |
| 3 | reset | 1 | I | 复位信号 |
| 4 | PC\_out | 32 | O | 当前PC值 |
| 5 | Enable | 1 | I | 使能端 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出指令地址 | 时钟上升沿且reset=0，enable=1，PC输出指令地址 |
| 2 | 复位 | 时钟上升沿且reset=1，将PC置为0x00003000 |

1. muxPCOP.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PCOP | 2 | I | muxPCOP控制信号 |
| 2 | PC4 | 32 | I | PC+4 |
| 3 | NPC | 32 | I | NPC结果 |
| 4 | newPC | 32 | O | 下条指令地址 |
| 5 | RD1 | 32 | I | rs值 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出下条指令地址 | PCOP==2’b00:newPC<=PC4;  PCOP==2’b01:newPC<=NPC;  PCOP==2’b10:newPC<=RD1; |

1. muxRegDst.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | RegDst | 2 | I | muxRegDst控制信号 |
| 2 | IR\_W | 32 | I | W级指令 |
| 3 | WAddr | 5 | O | 写入文件的寄存器下标 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出写入文件的寄存器下标 | RegDst=2’b00，WAddr=rd  RegDst=2’b01，WAddr=rt  RegDst=2’b10，WAddr=5’d31 |

1. muxRegWData.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALUC\_W | 32 | I | W级ALU结果 |
| 2 | DM\_W | 32 | I | W级从数据存储器读取的数据 |
| 3 | RegWData | 2 | I | muxRegWData的控制信号 |
| 4 | WData | 32 | O | 寄存器文件写入数据 |
| 5 | IR\_W | 32 | I | W级指令 |
| 6 | PC4\_W | 32 | I | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出寄存器文件写入数据 | RegWData=2’b00:WData<=ALUC\_W;  RegWData=2’b01:WData<=DM\_W;  RegWData=2’b10:WData<=PC4\_W+4; |

1. muxALUSrc.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALUSrc | 1 | I | muxAlUSrc的控制信号 |
| 2 | RD2 | 32 | I | 读寄存器文件时第二个寄存器的下标 |
| 3 | EXT\_out | 32 | I | 位扩展输出信号 |
| 4 | ALU\_B | 32 | O | ALU的第二个操作数 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU的第二个操作数 | ALUSrc=0时：ALU\_B=RD2  ALUSrc=1时：ALU\_B=EXT\_out |

11、add4.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PC | 32 | I | PC |
| 3 | PC4 | 32 | O | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | PC+4 | PC4=PC+4 |

1. IF\_ID.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_IM | 32 | I | IM输出指令 |
| 4 | PC4\_ADD4 | 32 | I | PC+4 |
| 5 | PC4\_D | 32 | O | PC+4 |
| 6 | IR\_D | 32 | O | D级指令 |
| 7 | enable | 1 | I | 使能端 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | D级流水线寄存器 | 传递IR与数据 |
| 2 | 使能端置零 | 暂停时enable=0 |

1. ID\_EX.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_D | 32 | I | D级指令 |
| 4 | RD1 | 32 | I | 读寄存器文件时第一个寄存器的输出 |
| 5 | RD2 | 32 | O | 读寄存器文件时二个寄存器的输出 |
| 6 | EXT\_out | 32 | O | D级指令 |
| 7 | PC4\_D | 32 | O | PC+4 |
| 8 | IR\_E | 32 | O | E级指令 |
| 9 | RD1\_E | 32 | O | 读寄存器文件时第一个寄存器的输出 |
| 10 | RD2\_E | 32 | O | 读寄存器文件时第二个寄存器的输出 |
| 11 | EXT\_E | 32 | O | EXT结果 |
| 12 | PC4\_E | 32 | O | PC+4 |
| 13 | Clear | 1 | I | 清零信号 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | E级流水线寄存器 | 传递IR与数据 |
| 2 | 清零 | ID\_EX清零 |

1. EX\_MEM.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_E | 32 | I | E级指令 |
| 4 | RD2\_E | 32 | I | 读寄存器文件时二个寄存器的输出 |
| 5 | ALU\_out | 32 | I | ALU结果 |
| 6 | PC4\_E | 32 | I | PC+4 |
| 7 | IR\_M | 32 | O | M级指令 |
| 8 | ALUC\_M | 32 | O | ALU结果 |
| 9 | PC4\_M | 32 | O | PC+4 |
| 10 | RD2\_M | 32 | O | 读寄存器文件时二个寄存器的输出 |
| 11 | muxHILO\_out | 32 | I | muxHILO输出值 |
| 12 | muxHILO\_M | 32 | I | muxHILO输出值 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | M级流水线寄存器 | 传递IR与数据 |

1. MEM\_WB.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_M | 32 | I | M级指令 |
| 4 | DM\_Data | 32 | I | 从数据存储器读取的数据 |
| 5 | ALU\_M | 32 | I | ALU结果 |
| 6 | PC4\_M | 32 | I | PC+4 |
| 7 | IR\_W | 32 | O | W级指令 |
| 8 | ALUC\_W | 32 | O | ALU结果 |
| 9 | PC4\_W | 32 | O | PC+4 |
| 10 | DM\_W | 32 | O | 从数据存储器读取的数据 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | W级流水线寄存器 | 传递IR与数据 |

1. MF\_RS\_E.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RS\_E\_OP | 2 | I | MF\_RS\_E控制信号 |
| 2 | RD1\_E | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | WData | 32 | I | muxRegWData结果 |
| 5 | MF\_RS\_E\_out | 32 | O | 转发结果 |
| 6 | PC4\_M | 32 | I | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10,11时转发 |

1. MF\_RT\_E.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RT\_E\_OP | 2 | I | MF\_RT\_E控制信号 |
| 2 | RD2\_E | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | WData | 32 | I | muxRegWData结果 |
| 5 | RD2 | 32 | O | 转发结果 |
| 6 | PC4+M | 32 | I | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10,11时转发 |

1. MF\_RS\_D.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RS\_D\_OP | 2 | I | MF\_RS\_D控制信号 |
| 2 | RData1 | 32 | I | 读寄存器文件时第1个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | WData | 32 | O | muxRegWData结果 |
| 5 | MF\_RS\_D\_out | 32 | I | 转发结果 |
| 6 | PC4\_M | 32 | I | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10,11时转发 |

1. MF\_RT\_D.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RT\_D\_OP | 2 | I | MF\_RS\_D控制信号 |
| 2 | RData2 | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | WData | 32 | O | muxRegWData结果 |
| 5 | MF\_RT\_D\_out | 32 | I | 转发结果 |
| 6 | PC4\_M | 32 | I | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10,11时转发 |

1. MF\_RT\_M.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RT\_M\_OP | 2 | I | MF\_RT\_M控制信号 |
| 2 | RD2\_M | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | WData | 32 | O | muxRegWData结果 |
| 4 | MF\_RT\_M\_out | 32 | I | 转发结果 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01时转发 |

21、compare.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | RD1 | 32 | I | 读寄存器文件时第一个寄存器的值 |
| 2 | RD2 | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | Equal\_out | 1 | O | RD1=RD2，则为1，否则为0 |
| 4 | bgez\_out | 1 | O | $signed(RD1)>=0?1'b1:1'b0; |
| 5 | bgtz\_out | 1 | O | $signed(RD1)>0?1'b1:1'b0; |
| 6 | bltz\_out | 1 | O | $signed(RD1)<0?1'b1:1'b0; |
| 7 | blez\_out | 1 | O | $signed(RD1)<=0?1'b1:1'b0; |

1. Be.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | Addr1\_0 | 2 | I | 地址低两位 |
| 2 | DMOP | 2 | I | DMOP控制信号 |
| 3 | be\_out | 1 | O | 字节使能 |

1. Expand.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | expandOP | 3 | I | expand控制信号 |
| 2 | Addr1\_0 | 2 | I | 地址低两位 |
| 3 | DM\_W | 32 | I | 原始数据 |
| 4 | Expand\_out | 32 | O | 数据扩展结果 |

1. Multdiv.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | start | 2 | I | 乘除法开始信号 |
| 4 | multdivOP | 2 | I | Multdiv控制信号 |
| 5 | multdiv\_A | 32 | I | Rs值 |
| 6 | multdiv\_B | 32 | I | Rt值 |
| 7 | HIWrite | 1 | I | HI写使能 |
| 8 | LOWrite | 1 | I | LO写使能 |
| 9 | HI | 32 | O | HI寄存器输出 |
| 10 | LO | 32 | O | LO寄存器输出 |
| 11 | Busy | 1 | O | Busy信号 |

1. muxHILO.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | HILOOP | 1 | I | muxHILO控制信号 |
| 2 | HI | 32 | I | HI寄存器输出 |
| 3 | LO | 32 | I | LO寄存器输出 |
| 4 | muxHILO\_out | 32 | O | muxHILO结果 |

1. muxSelect.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | Select | 1 | I | MuxSelect控制信号 |
| 2 | ALUC\_M | 32 | I | ALU结果 |
| 3 | muxHILO\_M | 32 | I | muxHILO结果 |
| 4 | muxSelect\_out | 32 | O | MuxSelect结果 |

1. 控制器control.v

control.v信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | instr | 32 | I | 指令 |
| 3 | PCOP | 2 | O | muxPCOP控制信号 |
| 4 | RegDst | 2 | O | muxRegDst控制信号 |
| 5 | ExtOP | 1 | O | ext控制信号 |
| 6 | RegWrite | 1 | O | gpr写入控制信号 |
| 7 | RegWData | 2 | O | muxRegWData的控制信号 |
| 8 | ALUSrc | 1 | O | muxAlUSrc的控制信号 |
| 9 | ALUOP | 4 | O | alu控制信号 |
| 10 | MemWrite | 1 | O | dm写入控制信号 |
| 11 | Equal | 1 | I | Equal信号 |
| 12 | bgez\_out | 1 | I | bgez\_out信号 |
| 13 | bgtz\_out | 1 | I | bgtz\_out信号 |
| 14 | bltz\_out | 1 | I | bltz\_out信号 |
| 15 | blez\_out | 1 | I | blez\_out信号 |
| 16 | start | 2 | O | 乘除法开始信号 |
| 17 | multdivOP | 2 | O | 乘除控制信号 |
| 18 | HIWrite | 1 | O | HI写使能 |
| 19 | LOWrite | 1 | O | LO写使能 |
| 20 | Select | 1 | O | MuxSelect控制信号 |
| 21 | DMOP | 2 | O | Sb，sh，sw信号 |
| 22 | expandOP | 2 | O | expand控制信号 |
| 23 | HILOOP | 1 | O | MuxHILO控制信号 |

1. 冲突模块

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | IR\_D | 32 | I | D级指令 |
| 2 | IR\_E | 32 | I | E级指令 |
| 3 | IR\_M | 32 | I | M级指令 |
| 4 | IR\_W | 32 | I | W级指令 |
| 5 | MF\_RT\_E\_OP | 2 | O | MF\_RT\_E控制信号 |
| 6 | MF\_RS\_E\_OP | 2 | O | MF\_RS\_E控制信号 |
| 7 | MF\_RS\_D\_OP | 2 | O | MF\_RS\_D控制信号 |
| 8 | MF\_RT\_D\_OP | 2 | O | MF\_RT\_D控制信号 |
| 9 | MF\_RT\_M\_OP | 2 | O | MF\_RT\_M控制信号 |
| 10 | enable | 1 | O | PC,IF\_ID使能信号 |
| 11 | clear | 1 | O | ID\_EX清零信号 |

1. 测试代码

lui $1,0x1234

ori $1,$1,0x6789

sw $1,0($0)

sh $1,6($0)

sb $1,5($0)

lb $2,6($0)

lbu $3,6($0)

add $4,$3,$2

sb $4,8($0)

lh $5,8($0)

lhu $6,4($0)

sub $1,$0,$1

sh $1,10($0)

lui $7,0x1010

lhu $8,10($0)

sra $9,$8,2

sllv $10,$9,$5

nor $11,$10,$9

slt $12,$10,$11

sltu $12,$10,$11

srlv $13,$11,$12

srav $14,$13,$12

andi $15,$14,0x9234

xori $16,$15,0x5678

slti $17,$16,0x7000

sltiu $18,$16,0x4562

lh $13,2($0)

lw $14,4($0)

divu $14,$13

jal label

mult $13,$14

mfhi $15

mflo $16

mthi $2

mtlo $2

srlv $16,$16,$4

div $13,$16

j end

mfhi $17

mflo $18

label:

jalr $1,$31

nop

end:

nop

$ 1 <= 12340000

$ 1 <= 12346789

\*00000000 <= 12346789

\*00000006 <= 6789

\*00000005 <= 89

$ 2 <= ffffff89

$ 3 <= 00000089

$ 4 <= 00000012

\*00000008 <= 12

$ 5 <= 00000012

$ 6 <= 00008900

$ 1 <= edcb9877

\*0000000a <= 9877

$ 7 <= 10100000

$ 8 <= 00009877

$ 9 <= 0000261d

$10 <= 98740000

$11 <= 678bd9e2

$12 <= 00000001

$12 <= 00000000

$13 <= 678bd9e2

$14 <= 678bd9e2

$15 <= 00009020

$16 <= 0000c658

$17 <= 00000000

$18 <= 00000000

$13 <= 00001234

$14 <= 67898900

$31 <= 0000307c

$ 1 <= 000030a8

$15 <= 0000075c

$16 <= b391d400

$16 <= 00002ce4

$17 <= 00001234

六、思考题

1、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？

整合进ALU会导致正在进行乘除法时，后续指令不能使用ALU，暂停的时钟周期变多，同时导致E级流水耗时过长，流水线时钟周期变大。

1. 参照你对延迟槽的理解，试解释“乘除槽”。

乘除槽指乘法指令后五个周期，除法指令后十个周期。在乘除槽中的与乘除、HI、LO相关的指令都应该被阻塞，其他指令可以运行。

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?

DM本身耗时大，如果在DM后，会导致M级流水耗时更大，导致流水线时钟周期因此变大，降低性能。

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）

访问char类型变量时按字节访问内存相对于按字访问内存性能上更有优势，因为char类型占一个字节。

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？

规划者。将指令归类为CAL\_R,CAL\_I,B,md,mf,mt等类型，对同类型指令进行统一处理。

1. 你对流水线CPU设计风格有何见解？

规划者和侦测者各有优劣，规划者一目了然但是代码量太大。侦测者代码简洁，偏向于基础原理。