

pc_reg.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位（同步）	
PCchange_enable_i	in	1	取指使能	来自 mem.v 的 mem_ce_o
branch_flag_i	in	1	标志是否跳转	来自 ctrl.v
branch_addr_i	in	31:0 `RegBus	跳转目标地址	
stalled_i	in	5:0	流水线停顿控制信号	
pc_o	out	31:0 `InstAddrBus	要读取的指令地址 PC	接入 if_id.v 同时作为 rom_addr_o 接入 cpu_ahb_if.v
ce_o	out	1	指令存储器使能信号	接入 cpu_ahb_if.v

if_id.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位（同步）	
pc_i	in	31:0 `InstAddrBus	取指阶段读取的指令地址	来自 pc_reg.v
inst_i	in	31:0 `InstBus	取指阶段取得的指令	来自 rom 经过 cpu_ahb_if.v
ex_branch_flag_i	in	1	标志是否跳转	来自 ctrl.v
stalled_i	in	5:0	流水线停顿控制信号	
pc_o	out	31:0 `InstAddrBus	译码阶段的指令地址	接入 id.v
inst_o	out	31:0 `InstBus	译码阶段的指令	

Regsfile.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位（同步）	
int_assert_i	in	1	中断发生标志	interrupt_ctrl.v
we_i	in	1	写使能	来自 mem_wb.v
waddr_i	in	4:0 `RegAddrBus	要写入的寄存器地址	
wdata_i	in	31:0 `RegBus	要写入的数据	
re1_i	in	1	port1 读使能	来自 id.v
raddr1_i	in	4:0 `RegAddrBus	port1 要读取的寄存器地址	
rdata1_o	out	31:0 `RegBus	port1 读出的值	接入 id.v
re2_i	in	1	port2 读使能	来自 id.v
raddr2_i	in	4:0 `RegAddrBus	port2 要读取的寄存器地址	
rdata2_o	out	31:0 `RegBus	port2 读出的值	接入 id.v

id.v

接口名	方向	位宽	描述	接线
pc_i	in	31:0 `InstAddrBus	译码阶段的指令地址	来自 if_id.v
inst_i	in	31:0 `InstBus	译码阶段的指令	
reg1_data_i	in	31:0 `RegBus	读入的寄存器值 1	来自 Regsfile.v
reg2_data_i	in	31:0 `RegBus	读入的寄存器值 2	
ex_wreg_i	in	1	写寄存器使能	来自 ex.v
ex_wreg_data_i	in	31:0 `RegBus	写寄存器地址	
ex_wreg_addr_i	in	4:0 `RegAddrBus	写寄存器数据	
ex_branch_flag_i	in	1	标志是否跳转	
ex_aluop_i	in	5:0 `AluOpBus	运算的类型	
mem_wreg_i	in	1	写寄存器使能	来自 mem.v
mem_wreg_data_i	in	31:0 `RegBus	写寄存器数据	
mem_wreg_addr_i	in	4:0 `RegAddrBus	写寄存器地址	
csr_data_i	in	31:0 `RegBus		来自 csr_reg.v
rcsr_addr_o	out	31:0 `DataAddrBus		接入 csr_reg.v
reg1_read_o	out	1	RF port1 读使能	接入 Regsfile.v
reg2_read_o	out	1	RF port2 读使能	
reg1_addr_o	out	4:0 `RegAddrBus	RF port1 读地址	
reg2_addr_o	out	4:0 `RegAddrBus	RF port2 读地址	
stallreq	out	1	译码阶段发出的停止指令	接入 ctrl.v
pc_o	out	31:0 `InstAddrBus	指令地址	接入 id_ex.v
inst_o	out	31:0 `InstBus	指令	
aluop_o	out	5:0 `AluOpBus	运算的子类型	
alusel_o	out	3:0 `AluSelBus	运算的类型	
operand1_o	out	31:0 `RegBus	运算的源操作数 1	
operand2_o	out	31:0 `RegBus	运算的源操作数 2	
wreg_addr_o	out	4:0 `RegAddrBus	目的寄存器地址	
wreg_o	out	1	标志是否写入目的寄存器	
wcsr_o	out	1	写 csr 使能	
csr_data_o	out	31:0 `RegBus	写 csr 数据	
wcsr_addr_o	out	31:0 `DataAddrBus	写 csr 地址	

id_ex.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
id_pc_i	in	31:0 `InstAddrBus	指令地址	来自 id.v
id_inst_i	in	31:0 `InstBus	指令	
id_aluop_i	in	5:0 `AluOpBus	运算的子类型	
id_alusel_i	in	3:0 `AluSelBus	运算的类型	
id_operand1_i	in	31:0 `RegBus	运算的源操作数 1	
id_operand2_i	in	31:0 `RegBus	运算的源操作数 2	
id_wreg_addr_i	in	5:0 `RegAddrBus	目的寄存器地址	
id_wreg_i	in	1	标志是否写入目的寄存器	
id_wcsr_i	in	1	csr 使能	
id_csr_data_i	in	5:0 `RegBus	写 csr 数据	
id_wcsr_addr_i	in	1:0 `DataAddrBus	写 csr 地址	
ex_branch_flag_i	in	1	分支标志	来自 ex.v
stalled_i	in	5:0	流水线停顿控制信号	来自 ctrl.v
ex_pc_o	out	31:0 `InstAddrBus	执行阶段的指令地址	接入 ex.v
ex_inst_o	out	31:0 `InstBus	执行阶段的指令	
ex_aluop_o	out	5:0 `AluOpBus	运算的子类型	
ex_alusel_o	out	3:0 `AluSelBus	运算的类型	
ex_operand1_o	out	31:0 `RegBus	运算的源操作数 1	
ex_operand2_o	out	31:0 `RegBus	运算的源操作数 2	
ex_wreg_addr_o	out	5:0 `RegAddrBus	要写入的目的寄存器地址	
ex_wreg_o	out	1	是否有要写的目的寄存器	
ex_wcsr_o	out	1	写 csr 使能	
ex_csr_data_o	out	31:0 `RegBus	写 csr 数据	
ex_wcsr_addr_o	out	31:0 `DataAddrBus	写 csr 地址	

ex.v

接口名	方向	位宽	描述	接线
int_assert_i	in	1	中断发生标志	
int_addr_i	in	31:0 `InstAddrBus	中断跳转地址	
pc_i	in	31:0 `InstAddrBus	执行阶段的指令地址	来自 id_ex.v
inst_i	in	31:0 `InstBus	执行阶段的指令	
aluop_i	in	5:0 `AluOpBus	运算的子类型	
alusel_i	in	3:0 `AluSelBus	运算的类型	
operand1_i	in	31:0 `RegBus	运算的源操作数 1	
operand2_i	in	31:0 `RegBus	运算的源操作数 2	
wreg_addr_i	in	5:0 `RegAddrBus	要写入的目的寄存器地址	
wreg_i	in	1	是否有要写的目的寄存器	
wcsr_i	in	1	写 csr 使能	
csr_data_i	in	31:0 `RegBus	写 csr 地址	
wcsr_addr_i	in	31:0 `DataAddrBus	写 csr 数据	
muldiv_result_i	in	63:0 `DoubleRegBus	表示运算的结果	来自 mul_div
muldiv_done	in	1	指示乘除法是否完成	
int_assert_i	in	1	中断发生标志	来自 interrupt_ctrl.v
int_addr_i	in	31:0 `InstAddrBus	中断跳转地址	
muldiv_start_o	out	1	运算使能信号	接入 mul_div
muldiv_dividend_o	out	31:0 `RegBus	被除数/乘数 1	
muldiv_divisor_o	out	31:0 `RegBus	除数/乘数 2	
mul_or_div	out	1	乘除法选择	
muldiv_reg1_sign_o	out	1	有无符号数选择 1	
muldiv_reg2_sign_o	out	1	有无符号数选择 2	
wreg_addr_o	out	5:0 `RegAddrBus	写寄存器地址	接入 ex_mem.v
wreg_o	out	1	写寄存器使能	
wreg_data_o	out	31:0 `RegBus	写寄存器数据	
aluop_o	out	5:0 `AluOpBus	运算的子类型	
memaddr_o	out	31:0 `DataAddrBus	加载或存储指令中需要访问的内存地址	
operand2_o	out	31:0 `RegBus	(存储指令中被存储的)源操作数 2	
wcsr_o	out	1	写 csr 使能	接入 csr_reg.v
wcsr_addr_o	out	31:0 `DataAddrBus	写 csr 地址	
wcsr_data_o	out	31:0 `RegBus	写 csr 数据	
branch_flag_o	out	1	标志是否跳转	接入 ctrl.v
branch_addr_o	out	31:0 `RegBus	跳转目标地址	

ex_mem.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
ex_wreg_addr_i	in	5:0 `RegAddrBus	执行阶段寄存器地址	来自 ex.v
ex_wreg_i	in	1	执行阶段寄存器使能	
ex_wreg_data_i	in	31:0 `RegBus	执行阶段寄存器数据	
ex_aluop_i	in	5:0 `AluOpBus	运算的子类型	
ex_memaddr_i	in	31:0 `DataAddrBus	访存地址	
ex_operand2_i	in	31:0 `RegBus	源操作数 2	
stalled_i	in	5:0	流水线停顿控制信号	来自 ctrl.v
mem_wd	out	5:0 `RegAddrBus	访存阶段寄存器地址	来自 ex.v
mem_wreg	out	1	访存阶段寄存器使能	
mem_wdata	out	31:0 `RegBus	访存阶段寄存器数据	
mem_aluop_o	out	5:0 `AluOpBus	运算的子类型	
mem_memaddr_o	out	31:0 `DataAddrBus	访存地址	
mem_operand2_o	out	31:0 `RegBus	运算的源操作数 2	

mem.v

接口名	方向	位宽	描述	接线
wreg_addr_i	in	5:0 `RegAddrBus	访存阶段寄存器地址	来自 ex_mem.v
wreg_i	in	1	访存阶段寄存器使能	
wreg_data_i	in	31:0 `RegBus	访存阶段寄存器数据	
aluop_i	in	5:0 `AluOpBus	运算的子类型	
memaddr_i	in	31:0 `DataAddrBus	访存地址	
operand2_i	in	31:0 `RegBus	源操作数 2	
int_assert_i	in	1	中断发生标志	
ram_data_i	in	31:0 `DataBus	内存信号的输入	来自 ram
ram_addr_o	out	31:0 `DataAddrBus	内存的地址信号	接入 ram
ram_we_o	out	1	内存的使能信号	
ram_sel_o	out	2:0	内存的片选信号	
ram_data_o	out	31:0 `DataBus	写入内存的数据	
ram_ce_o	out	1	内存的片使能信号	
wreg_addr_o	out	5:0 `RegAddrBus	写寄存器地址	接入 mem_wb.v 和 id.v
wreg_o	out	1	写寄存器使能	
wreg_data_o	out	31:0 `RegBus	写寄存器数据	

mem_wb.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
mem_wreg_addr_i	in	5:0 `RegAddrBus	写寄存器地址	来自 mem.v
mem_wreg_i	in	1	写寄存器使能	
mem_wreg_data_i	in	31:0 `RegBus	写寄存器数据	
stalled_i	in	5:0	流水线停顿控制信号	来自 ctrl.v
wb_wreg_addr_o	out	5:0 `RegAddrBus	回写阶段寄存器地址	接入 regsfile.v
wb_wreg_o	out	1	回写阶段寄存器使能	
wb_wreg_data_o	out	31:0 `RegBus	回写阶段寄存器数据	

ctrl.v

接口名	方向	位宽	描述	接线
stallreq_from_id	in	1	译码阶段流水行停顿请求	来自 id.v
stallreq_from_ex	in	1	执行阶段流水行停顿请求	来自 ex.v 的 muldiv_start_o
stallreq_from_if	in	1	取指阶段流水行停顿请求	来自 cpu_ahb_if.v
stallreq_from_mem	in	1	访存阶段流水行停顿请求	来自 cpu_ahb_mem.v
stallreq_from_interrupt	in	1	中断流水线停顿请求	来自 interrupt_ctrl.v
stallreq_from_jtag	in	1		
branch_flag_i	in	1	分支跳转标志	来自 ex.v
branch_addr_i	in	31:0 `InstAddrBus	分支跳转地址	
branch_flag_o	out	1	分支跳转标志	接入 pc_reg.v
branch_addr_o	out	31:0 `InstAddrBus	分支跳转地址	
stalled_o	out	5:0	流水线中断信号	接入流水线各个阶段

Interrupt_ctrl.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
int_flag_i	in	13:0 `INT_BUS		外部输入
inst_i	in	31:0 `InstBus	当前指令	来自 id.v
inst_addr_i	in	31:0 `InstAddrBus	当前指令地址	
branch_flag_i	in	1	标志是否跳转	来自 ctrl.v
branch_addr_i	in	31:0 `RegBus	跳转的地址	
div_i	in	1	乘除法使能	来自 ex.v
global_int_en_i	in	1	全局中断使能标志	来自 csr_reg.v
csr_mtvec	in	31:0 `RegBus		来自 csr_reg.v
csr_mepc	in	31:0 `RegBus		
csr_mstatus	in	31:0 `RegBus		
stallreq_interrupt_o	out	1	中断请求信号	来自 ctrl.v
we_o	out	1	中断写使能	接入 csr_reg.v
waddr_o	out	31:0 `DataAddrBus	中断写地址	
raddr_o	out	31:0 `DataAddrBus	中断读地址	
data_o	out	31:0 `RegBus	(中断信息)	
int_addr_o	out	31:0 `InstAddrBus		接入 ex.v
int_assert_o	out	1		

csr_reg.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
we_i	in	1	csr 写使能	来自 ex.v
waddr_i	in	31:0`DataAddrBus	写 csr 寄存器的地址	
wdata_i	in	31:0`RegBus	写 csr 寄存器的数据	
raddr_i	in	31:0`DataAddrBus	读 csr 寄存器的地址	来自 id.v
interrupt_we_i	in	1	中断写使能信号	来自 interrupt_ctrl.v
interrupt_raddr_i	in	31:0`DataAddrBus	中断读 csr 寄存器的地址	
interrupt_waddr_i	in	31:0`DataAddrBus	中断写 csr 寄存器的地址	
interrupt_data_i	in	31:0`RegBus	中断写 csr 寄存器的数据	
interrupt_data_o	out	31:0`RegBus	中断读 csr 寄存器的数据	
interrupt_csr_mtvec	out	31:0`RegBus		接入 interrupt_ctrl.v
interrupt_csr_mepc	out	31:0`RegBus		
interrupt_csr_mstatus	out	31:0`RegBus		
global_int_en_o	out	1	全局中断使能标志	
rdata_o	out	31:0`RegBus	ex 模块读寄存器数据	接入 id.v

cpu_ahb_if.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
cpu_addr_i	in	[31:0]	cpu 请求的指令地址	来自 pc_reg.v
cpu_ce_i	in	1	使能信号	
cpu_we_i	in	1	读写选择	1'b0
cpu_writedata_i	in	[31:0]	写数据(ZeroWord)	`ZeroWord
cpu_sel_i				3'b010
in	[2:0]	片选信号 (3'b010)	3'b010	
cpu_readdate_o				
out	[31:0]	读出的指令	接入 if_id.v	
M_HGRANT	in	1	总线授权信号	总线 直接接入顶层 io 作为 Master 1
M_HRDATA	in	[31:0]	总线读回数据	
M_HBUSREQ	out	1	主机 0 请求总线, 1 使能	
M_HADDR	out	[31:0]	地址	
M_HTRANS	out	[1:0]	主机 0 传输状态	
M_HSIZE	out	[2:0]	主机 0 数据大小	
M_HBURST	out	[2:0]	传输模式 (单次)	
M_HWRITE	out	1	1 为写, 0 为读	
M_HWDATA	out	[31:0]	写数据	
stallreq	out	1	取指流水线停顿请求	接入 ctrl.v

cpu.ahb_mem.v

接口名	方向	位宽	描述	接线
clk	in	1	时钟	
rst_n	in	1	复位	
cpu_addr_i	in	[31:0]	cpu 端指令地址	来自 mem.v
cpu_ce_i	in	1	使能信号	
cpu_we_i	in	1	写使能信号	
cpu_writedata_i	in	[31:0]	写数据	
cpu_sel_i	in	[2:0]	片选信号	
cpu_readdata_o	out	[31:0]	读出的指令	接入 if_id.v
M_HGRANT	in	1	总线授权信号	总线 直接接入顶层 io 作为 Master 0
M_HRDATA	in	[31:0]	总线读回数据	
M_HREADY	in	1	指示数据是否完成	
M_HBUSREQ	out	1	主机 0 请求总线，1 使能	
M_HADDR	out	[31:0]	主机 0 地址	
M_HTRANS	out	[1:0]	主机 0 传输状态	
M_HSIZE	out	[2:0]	主机 0 数据大小	
M_HBURST	out	[2:0]	传输模式（单次）	
M_HWRITE	out	1	1 为写，0 为读	
M_HWDATA	out	[31:0]	写数据	
stallreq	out	1	取指流水线停顿请求	接入 ctrl.v

三个模块间的连线如下：

源模块	信号	目标模块	描述
ahb_arbiter	HGRANT	ahb_m2s_m3	仲裁结果，授权信号，指示给请求总线的主设备，告诉它已经获得了对总线的控制权。
	ahb_arbiter HMASTER		主设备编号信号，标示当前控制总线的主设备。
	ahb_arbiter HLOCK		主设备的锁定信号，用于表示当前总线周期是否被锁定。
ahb_arbiter	HGRANT	amba_ahb_m3s5	总线授权信号反馈至顶级模块，用于进一步控制。
	ahb_arbiter HMASTER		主设备编号信号反馈至顶级模块，用于进一步处理。
ahb_m2s_m3	HADDR, HTRANS, HPROT, HWRITE, HSIZE, HBURST, HWDATA	ahb_lite_s5	主设备的地址、事务类型、保护位、写控制、大小、突发类型和数据，传送至从设备。
ahb_m2s_m3	HADDR, HTRANS, HPROT, HWRITE, HSIZE, HBURST, HWDATA	amba_ahb_m3s5	主设备接口信号反馈至顶级模块，用于进一步控制。
ahb_lite_s5	M_HRDATA, M_HRESP, M_HREADY	ahb_m2s_m3	从设备返回的数据、响应和就绪信号，传送回主设备。