CPU 各模块及接口说明

喵喵喵喵喵？

December 11, 2017

Contents

1 PC

1.1 简介

取指模块。然后好像没什么好说的了。

1.2 接口定义

Table 1: PC 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	pause_i	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PAUSE_CTRL.pause_o	此模块是否暂停
in	branch_i	STD_LOGIC	1	ID.branch_o	是否跳转
in	branch_target_addr_i	STD_LOGIC_VECTOR	INST_ADDR_LEN	ID.branch_target_addr_o	如果跳转，跳到什么位置
out	en_o	STD_LOGIC	1	MIPS_CPU.rom_en_o	是否读指令
out	pc_o	STD_LOGIC_VECTOR	INST_ADDR_LEN	MIPS_CPU.rom_addr_o, IF/ID.pc_i	下一条指令的位置

2 IF/ID

2.1 简介

2.2 接口定义

Table 2: IF/ID 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	pc_i	STD_LOGIC_VECTOR	INST_ADDR_LEN	PC.pc_o	指令地址
in	inst_i	STD_LOGIC_VECTOR	INST_LEN	MIPS_CPU.inst_i	指令
in	pause_i	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PAUSE_CTRL.pause_o	是否暂停
out	pc_o	STD_LOGIC_VECTOR	INST_ADDR_LEN	ID.pc_i	指令地址
out	inst_o	STD_LOGIC_VECTOR	INST_LEN	ID.inst_i	指令

3 ID

3.1 简介

译码模块。主要工作流程是，先根据 op 确定指令类型，再根据 special_funct 确定具体是什么指令，然后根据指令类型进行译码。

SPECIAL 类指令 SHIFT 类指令 读 rt，扩展 shamt，写 rd

待写

待写

在译码结束之后，最后还有一段，根据 EX 和 MEM 阶段写寄存器的情况解决数据冲突，然后确定指令对应的两个操作数 operand_1_o 和 operand_2_o。

3.2 接口定义

Table 3: ID 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	pc_i	STD_LOGIC_VECTOR	INST_ADDR_LEN	IF/ID.pc_o	指令地址
in	inst_i	STD_LOGIC_VECTOR	INST_LEN	IF/ID.inst_o	指令
in	reg_rd_data_1_i	STD_LOGIC_VECTOR	REG_DATA_LEN	REGISTERS.reg_rd_data_-1_o	寄存器 1 读出数据
in	reg_rd_data_2_i	STD_LOGIC_VECTOR	REG_DATA_LEN	REGISTERS.reg_rd_data_-2_o	寄存器 2 读出数据
in	ex_reg_wt_en_i	STD_LOGIC	1	EX.reg_wt_en_o	EX 模块是否写寄存器
in	ex_reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	EX.reg_wt_addr_o	EX 模块写寄存器地址
in	ex_reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.reg_wt_data_o	EX 模块写寄存器数据
in	mem_reg_wt_en_i	STD_LOGIC	1	MEM.reg_wt_en_o	MEM 模块是否写寄存器
in	mem_reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	MEM.reg_wt_addr_o	MEM 模块写寄存器地址
in	mem_reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.reg_wt_data_o	MEM 模块写寄存器数据
in	is_in_delayslot_i	STD_LOGIC	1	ID/EX.is_in_delayslot_o	当前指令是否在延迟槽内
out	op_o	STD_LOGIC_VECTOR	OP_LEN	ID/EX.op_i	指令操作类型
out	funct_o	STD_LOGIC_VECTOR	FUNCT_LEN	ID/EX.funct_i	指令子操作类型
out	reg_rd_en_1_o	STD_LOGIC	1	REGISTERS.reg_rd_en_1_i	寄存器 1 读使能
out	reg_rd_en_2_o	STD_LOGIC	1	REGISTERS.reg_rd_en_2_i	寄存器 2 读使能
out	reg_rd_addr_1_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	REGISTERS.reg_rd_addr_-1_i	寄存器 1 读地址

接下页

方向	名称	类型	宽度	连接到	详细描述
out	reg_rd_addr_2_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	REGISTERS.reg_rd_addr_-2_i	寄存器 2 读地址
out	operand_1_o	STD_LOGIC_VECTOR	DATA_LEN	ID/EX.operand_1_i	指令操作数 1
out	operand_2_o	STD_LOGIC_VECTOR	DATA_LEN	ID/EX.operand_2_i	指令操作数 2
out	extended_offset_o	STD_LOGIC_VECTOR	DATA_LEN	ID/EX.extended_offset_i	扩展后立即数
out	reg_wt_en_o	STD_LOGIC	1	ID/EX.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	ID/EX.reg_wt_addr_i	寄存器写地址
out	pause_o	STD_LOGIC	1	PAUSE_CTRL.id_pause_i	是否需要暂停
out	branch_o	STD_LOGIC	1	PC.branch_i	当前是否为分支跳转指令
out	branch_target_-addr_o	STD_LOGIC_VECTOR	INST_ADDR_LEN	PC.branch_target_addr_i	跳转地址
out	is_in_delayslot_o	STD_LOGIC	1	ID/EX.is_in_delayslot_i	当前指令是否在延迟槽内
out	next_inst_in_de-layslot_o	STD_LOGIC	1	ID/EX.next_inst_in_de-layslot_i	下一条指令是否在延迟槽内
out	link_addr_o	STD_LOGIC_VECTOR	INST_ADDR_LEN	ID/EX.link_addr_i	跳转指令的返回地址

4 ID/EX

4.1 简介

4.2 接口定义

Table 4: ID/EX 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	op_i	STD_LOGIC_VECTOR	OP_LEN	ID.op_o	指令操作类型
in	funct_i	STD_LOGIC_VECTOR	FUNCT_LEN	ID.funct_o	指令子操作类型
in	operand_1_i	STD_LOGIC_VECTOR	REG_DATA_LEN	ID.operand_1_o	指令操作数 1
in	operand_2_i	STD_LOGIC_VECTOR	REG_DATA_LEN	ID.operand_2_o	指令操作数 2
in	extended_off-set_i	STD_LOGIC_VECTOR	DATA_LEN	ID.extended_offset_o	扩展后立即数
in	reg_wt_en_i	STD_LOGIC	1	ID.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	ID.reg_wt_addr_o	寄存器写地址
in	pause_i	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PAUSE_CTRL.pause_o	是否暂停
in	is_in_de-layslot_i	STD_LOGIC	1	ID.is_in_delayslot_o	当前指令是否在延迟槽中

接下页

方向	名称	类型	宽度	连接到	详细描述
in	next_inst_in_-delayslot_i	STD_LOGIC	1	ID.next_inst_in_de-layslot_o	下一条指令是否在延迟槽中
in	link_addr_i	STD_LOGIC_VECTOR	INST_ADDR_LEN	ID.link_addr_o	跳转指令的返回地址
out	op_o	STD_LOGIC_VECTOR	OP_LEN	EX.op_i	指令操作类型
out	funct_o	STD_LOGIC_VECTOR	FUNCT_LEN	EX.funct_i	指令子操作类型
out	operand_1_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.operand_1_i	指令操作数 1
out	operand_2_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.operand_2_i	指令操作数 2
out	extended_off-set_o	STD_LOGIC_VECTOR	DATA_LEN	EX.extended_offset_i	扩展后立即数
out	reg_wt_en_o	STD_LOGIC	1	EX.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	EX.reg_wt_addr_i	寄存器写地址
out	is_in_de-layslot_o	STD_LOGIC	1	EX.is_in_delayslot_i	当前指令是否在延迟槽中
out	next_inst_in_-delayslot_o	STD_LOGIC	1	ID.is_in_delayslot_i	下一条指令是否在延迟槽中
out	link_addr_o	STD_LOGIC_VECTOR	INST_ADDR_LEN	EX.link_addr_i	跳转指令的返回地址

5 EX

5.1 简介

5.2 接口定义

Table 5: EX 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	op_i	STD_LOGIC_VECTOR	OP_LEN	ID/EX.op_o	指令操作类型
in	funct_i	STD_LOGIC_VECTOR	FUNCT_LEN	ID/EX.funct_o	指令子操作类型
in	operand_1_i	STD_LOGIC_VECTOR	REG_DATA_LEN	ID/EX.operand_1_o	指令操作数 1
in	operand_2_i	STD_LOGIC_VECTOR	REG_DATA_LEN	ID/EX.operand_2_o	指令操作数 2
in	extended_offset_i	STD_LOGIC_VECTOR	DATA_LEN	ID/EX.extended_offset_o	扩展后立即数
in	reg_wt_en_i	STD_LOGIC	1	ID/EX.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	ID/EX.reg_wt_addr_o	寄存器写地址
in	hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	HI_LO.hi_o	HI 寄存器
in	lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	HI_LO.lo_o	LO 寄存器
in	mem_hilo_en_i	STD_LOGIC	1	MEM.hilo_en_o	MEM 阶段的指令是否写 HILO

接下页

方向	名称	类型	宽度	连接到	详细描述
in	mem_hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.hi_o	MEM 阶段的指令写 HI 的数据
in	mem_lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.lo_o	MEM 阶段的指令写 LO 的数据
in	wb_hilo_en_i	STD_LOGIC	1	MEM/WB.hilo_en_o	WB 阶段的指令是否写 HILO
in	wb_hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.hi_o	WB 阶段的指令写 HI 的数据
in	wb_lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.lo_o	WB 阶段的指令写 LO 的数据
in	clock_cycle_cnt_i	STD_LOGIC_VECTOR	ACCU_CNT_LEN	EX/MEM.clock_cycle_-cnt_o	进行到了乘累加指令的第几个周期
in	mul_cur_result_i	STD_LOGIC_VECTOR	DOUBLE_DATA_LEN	EX/MEM.mul_cur_result_o	乘累加指令当前结果
in	is_in_delayslot_i	STD_LOGIC	1	ID/EX.is_in_delayslot_o	当前指令是否在延迟槽内
in	link_addr_i	STD_LOGIC_VECTOR	INST_ADDR_LEN	ID/EX.link_addr_o	跳转指令的返回地址
out	reg_wt_en_o	STD_LOGIC	1	EX/MEM.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	EX/MEM.reg_wt_addr_i	寄存器写地址
out	reg_wt_data_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.reg_wt_data_i	寄存器写数据
out	is_load_store_o	STD_LOGIC	1	EX/MEM.is_load_store_i	当前指令是否为访存指令
out	funct_o	STD_LOGIC_VECTOR	FUNCT_LEN	EX/MEM.funct_i	访存指令子操作类型
out	load_store_addr_o	STD_LOGIC_VECTOR	ADDR_LEN	EX/MEM.load_store_-addr_i	访存指令访问的地址
out	store_data_o	STD_LOGIC_VECTOR	DATA_LEN	EX/MEM.store_data_i	store 指令要存储的数据
out	hilo_en_o	STD_LOGIC	1	EX/MEM.hilo_en_i	写 HILO 使能
out	hi_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.hi_i	写 HI 数据
out	lo_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.lo_i	写 LO 数据
out	pause_o	STD_LOGIC	1	PAUSE_CTRL.ex_pause_i	是否需要暂停
out	clock_cycle_cnt_o	STD_LOGIC_VECTOR	ACCU_CNT_LEN	EX/MEM.clock_cycle_-cnt_i	进行到了乘累加指令的第几个周期

接下页

方向	名称	类型	宽度	连接到	详细描述
out	mul_cur_result_o	STD_LOGIC_VECTOR	DOUBLE_DATA_LEN	EX/MEM.mul_cur_result_i	乘累加指令当前结果

6 EX/MEM

6.1 简介

6.2 接口定义

Table 6: PC 的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	reg_wt_en_i	STD_LOGIC	1	EX.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	EX.reg_wt_addr_o	寄存器写地址
in	reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.reg_wt_data_o	寄存器写数据
in	is_load_store_i	STD_LOGIC	1	EX.is_load_store_o	当前指令是否为访存指令
in	funct_i	STD_LOGIC_VECTOR	FUNCT_LEN	EX.funct_o	访存指令子操作类型
in	load_store_addr_i	STD_LOGIC_VECTOR	ADDR_LEN	EX.load_store_addr_o	访存指令访问的地址
in	store_data_i	STD_LOGIC_VECTOR	DATA_LEN	EX.store_data_o	store 指令要存储的数据
in	hilo_en_i	STD_LOGIC	1	EX.hilo_en_o	写 HILO 使能
in	hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.hi_o	写 HI 数据
in	lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.lo_o	写 LO 数据
in	pause_i	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PAUSE_CTRL.pause_o	流水线当前阶段是否需要暂停
in	clock_cycle_cnt_i	STD_LOGIC_VECTOR	ACCU_CNT_LEN	EX.clock_cycle_cnt_i	进行到了乘累加指令的第几个周期
in	mul_cur_result_i	STD_LOGIC_VECTOR	DOUBLE_DATA_LEN	EX.mul_cur_result_o	乘累加指令当前结果
out	reg_wt_en_o	STD_LOGIC	1	MEM.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	MEM.reg_wt_addr_i	寄存器写地址
out	reg_wt_data_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.reg_wt_data_i	寄存器写数据
out	is_load_store_o	STD_LOGIC	1	MEM.is_load_store_i	当前指令是否为访存指令

接下页

方向	名称	类型	宽度	连接到	详细描述
out	funct_o	STD_LOGIC_VECTOR	FUNCT_LEN	MEM.funct_i	访存指令子操作类型
out	load_store_addr_o	STD_LOGIC_VECTOR	ADDR_LEN	MEM.load_store_addr_i	访存指令访问的地址
out	store_data_o	STD_LOGIC_VECTOR	DATA_LEN	MEM.store_data_i	store 指令要存储的数据
out	hilo_en_o	STD_LOGIC	1	MEM.hilo_en_i	写 HILO 使能
out	hi_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.hi_i	写 HI 数据
out	lo_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.lo_i	写 LO 数据
out	clock_cycle_cnt_o	STD_LOGIC_VECTOR	ACCU_CNT_LEN	EX.clock_cycle_cnt_i	进行到了乘累加指令的第几个周期
out	mul_cur_result_o	STD_LOGIC_VECTOR	DOUBLE_DATA_LEN	EX.mul_cur_result_o	乘累加指令当前结果

7 MEM

7.1 简介

现在正在研究要不要写暂停。但是可能就不写了，因为可以抄别的组的文档。

7.2 接口定义

Table 7: PC 的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	reg_wt_en_i	STD_LOGIC	1	EX/MEM.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	EX/MEM.reg_wt_addr_o	寄存器写地址
in	reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.reg_wt_data_o	寄存器写数据
in	ram_rd_data_i	STD_LOGIC_VECTOR	DATA_LEN	MIPS_CPU.ram_data_o	从外部读取的数据
in	is_load_store_i	STD_LOGIC	1	EX/MEM.is_load_-store_o	当前指令是否为访存指令
in	funct_i	STD_LOGIC_VECTOR	FUNCT_LEN	EX/MEM.funct_o	访存指令子操作类型
in	load_store_addr_i	STD_LOGIC_VECTOR	ADDR_LEN	EX/MEM.load_store_-addr_o	访存指令访问的地址
in	store_data_i	STD_LOGIC_VECTOR	DATA_LEN	EX/MEM.store_data_o	store 指令要存储的数据
in	hilo_en_i	STD_LOGIC	1	EX/MEM.hilo_en_o	写 HILO 使能
in	hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.hi_o	写 HI 数据
in	lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	EX/MEM.lo_o	写 LO 数据

接下页

方向	名称	类型	宽度	连接到	详细描述
out	reg_wt_en_o	STD_LOGIC	1	MEM/WB.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	MEM/WB.reg_wt_addr_i	寄存器写地址
out	reg_wt_data_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.reg_wt_data_i	寄存器写数据
out	ram_en_o	STD_LOGIC	1	MIPS_CPU.ram_en_o	RAM 读写使能
out	ram_is_read_o	STD_LOGIC	1	MIPS_CPU.ram_is_-read_o	RAM 是否为读
out	ram_addr_o	STD_LOGIC_VECTOR	ADDR_LEN	MIPS_CPU.ram_addr_o	RAM 的访问地址
out	ram_data_o	STD_LOGIC_VECTOR	DATA_LEN	MIPS_CPU.ram_data_o	RAM 的写数据
out	ram_data_sel_o	STD_LOGIC_VECTOR	BYTE_IN_DATA	MIPS_CPU.ram_data_-sel_o	RAM 数据选择
out	hilo_en_o	STD_LOGIC	1	MEM/WB.hilo_en_i, EX.mem_hilo_en_i	写 HILO 使能
out	hi_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.hi_i, EX.mem_hi_i	写 HI 数据
out	lo_o	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.lo_i, EX.mem_lo_i	写 LO 数据

8 MEM/WB

8.1 简介

8.2 接口定义

Table 8: PC 的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	reg_wt_en_i	STD_LOGIC	1	MEM.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	MEM.reg_wt_addr_o	寄存器写地址
in	reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.reg_wt_data_o	寄存器写数据
in	hilo_en_i	STD_LOGIC	1	MEM.hilo_en_o	写 HILO 使能
in	hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.hi_o	写 HI 数据
in	lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM.lo_o	写 LO 数据
in	pause_i	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PAUSE_CTRL.pause_o	是否暂停
out	reg_wt_en_o	STD_LOGIC	1	REGISTERS.reg_wt_en_i	寄存器写使能
out	reg_wt_addr_o	STD_LOGIC_VECTOR	REG_ADDR_LEN	REGISTERS.reg_wt_addr_i	寄存器写地址
out	reg_wt_data_o	STD_LOGIC_VECTOR	REG_DATA_LEN	REGISTERS.reg_wt_data_i	寄存器写数据
out	hilo_en_o	STD_LOGIC	1	HI_LO.en, EX.wb_hilo_en_i	写 HILO 使能
out	hi_o	STD_LOGIC_VECTOR	REG_DATA_LEN	HI_LO.hi_i, EX.wb_hi_i	写 HI 数据
out	lo_o	STD_LOGIC_VECTOR	REG_DATA_LEN	HI_LO.lo_i, EX.wb_lo_i	写 LO 数据

接下页

方向	名称	类型	宽度	连接到	详细描述

9 REGISTERS

9.1 简介

9.2 接口定义

Table 9: REGISTERS 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	reg_rd_en_1_i	STD_LOGIC	1	ID.reg_rd_en_1_o	寄存器 1 读使能
in	reg_rd_en_2_i	STD_LOGIC	1	ID.reg_rd_en_2_o	寄存器 2 读使能
in	reg_rd_addr_1_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	ID.reg_rd_addr_1_o	寄存器 1 读地址
in	reg_rd_addr_2_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	ID.reg_rd_addr_2_o	寄存器 2 读地址
in	reg_wt_en_i	STD_LOGIC	1	MEM/WB.reg_wt_en_o	寄存器写使能
in	reg_wt_addr_i	STD_LOGIC_VECTOR	REG_ADDR_LEN	MEM/WB.reg_wt_addr_o	寄存器写地址
in	reg_wt_data_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.reg_wt_data_o	寄存器写数据
out	reg_rd_data_1_o	STD_LOGIC_VECTOR	REG_DATA_LEN	ID.reg_rd_data_1_i	寄存器 1 读出数据
out	reg_rd_data_2_o	STD_LOGIC_VECTOR	REG_DATA_LEN	ID.reg_rd_data_2_i	寄存器 2 读出数据

10 HI_LO

10.1 简介

10.2 接口定义

Table 10: HILO 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.rst	复位信号
in	clk	STD_LOGIC	1	MIPS_CPU.clk	时钟信号
in	en	STD_LOGIC	1	MEM/WB.hilo_en_o	使能
in	hi_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.hi_o	HI
in	lo_i	STD_LOGIC_VECTOR	REG_DATA_LEN	MEM/WB.lo_o	LO
out	hi_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.hi_i	HI
out	lo_o	STD_LOGIC_VECTOR	REG_DATA_LEN	EX.hi_o	LO

11 PAUSE_CTRL

11.1 简介

11.2 接口定义

Table 11: PAUSE_CTRL 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	MIPS_CPU.extended_offset_i	复位信号
in	id_pause_i	STD_LOGIC	1	ID.pause_o	ID 模块是否暂停
in	ex_pause_i	STD_LOGIC	1	EX.pause_o	EX 模块是否暂停
out	pause_o	STD_LOGIC_VECTOR	CTRL_PAUSE_LEN	PC.pause_i, IF/ID.pause_i, ID/EX.pause_i, EX/MEM.pause_-i, MEM/WB.pause_i	各模块是否暂停

12 MIPS_CPU

12.1 简介

12.2 接口定义

Table 12: MIPS_CPU 模块的接口

方向	名称	类型	宽度	连接到	详细描述
in	rst	STD_LOGIC	1	PC.rst, IF/ID.rst, ID.rst, ID/EX.rst, EX.rst, EX/MEM.rst, MEM.rst, MEM/WB.rst, REGISTERS.rst, HI_LO.rst, PAUSE_CTRL.rst	复位信号
in	clk	STD_LOGIC	1	PC.clk, IF/ID.clk, ID/EX.clk, EX/MEM.clk, MEM/WB.clk, REGISTERS.clk, HI_LO.clk	时钟信号
in	inst_i	STD_LOGIC_VECTOR	INST_LEN	IF/ID.inst_i, ROM	输入指令
in	ram_rd_data_i	STD_LOGIC_VECTOR	INST_LEN	MEM.ram_rd_data_i, RAM	输入数据
out	rom_en_o	STD_LOGIC	1	PC.en_o, ROM	ROM 使能
out	rom_addr_o	STD_LOGIC_VECTOR	INST_LEN	PC.pc_o, ROM	指令地址
out	ram_en_o	STD_LOGIC	1	MEM.ram_en_o, RAM	RAM 使能
out	ram_is_read_o	STD_LOGIC	1	MEM.ram_is_read_o, RAM	RAM 是否是读
out	ram_addr_o	STD_LOGIC_VECTOR	INST_LEN	MEM.ram_addr_o, RAM	读写 RAM 的地址

接下页

方向	名称	类型	宽度	连接到	详细描述
out	ram_data_o	STD_LOGIC_VECTOR	DATA_LEN	MEM.ram_data_o, RAM	写 RAM 的数据
out	ram_data_sel_o	STD_LOGIC_VECTOR	BYTE_IN_DATA	MEM.ram_data_sel_o, RAM	读写内容选择

A 常数和宏定义

A.1 INTEGER 类型的常数

Table 13: INTEGER 类型的常数

名称	内容	详细描述
INST_ADDR_LEN	32	指令的地址长度
ADDR_LEN	32	普通的地址长度
INST_LEN	32	指令长度
REG_ADDR_LEN	5	寄存器地址长度
REG_DATA_LEN	32	寄存器内数据长度
DATA_LEN	32	一般的数据长度
DOUBLE_DATA_LEN	64	乘法结果的数据长度
CTRL_PAUSE_LEN	6	暂停控制数据长度
OP_LEN	6	指令操作码长度
FUNCT_LEN	6	指令子操作码长度
ACCU_CNT_LEN	2	乘累加/减指令周期数
BYTE_IN_DATA	4	一个普通的数据中有多少字节