

# 软工计原联合项目

## 设计文档

NonExist 组

张钰晖, 杨一滨, 周正平

# 目录

<b>1</b>	<b>文档说明</b>	<b>4</b>
<b>2</b>	<b>指令流水</b>	<b>5</b>
2.1	pc_reg . . . . .	5
2.1.1	简介概述 . . . . .	5
2.1.2	接口定义 . . . . .	5
2.1.3	设计细节 . . . . .	6
2.2	if_id . . . . .	6
2.2.1	简介概述 . . . . .	6
2.2.2	接口定义 . . . . .	6
2.2.3	设计细节 . . . . .	6
2.3	id . . . . .	6
2.3.1	简介概述 . . . . .	6
2.3.2	接口定义 . . . . .	7
2.3.3	设计细节 . . . . .	8
2.4	id_ex . . . . .	8
2.4.1	简介概述 . . . . .	8
2.4.2	接口定义 . . . . .	9
2.4.3	设计细节 . . . . .	9
2.5	ex . . . . .	10
2.5.1	简介概述 . . . . .	10
2.5.2	接口定义 . . . . .	12
2.5.3	设计细节 . . . . .	13
2.6	ex_mem . . . . .	13
2.6.1	简介概述 . . . . .	13
2.6.2	接口定义 . . . . .	13
2.6.3	设计细节 . . . . .	13
2.7	mem . . . . .	13
2.7.1	简介概述 . . . . .	13
2.7.2	接口定义 . . . . .	13

2.7.3	设计细节	13
2.8	mem_wb	13
2.8.1	简介概述	13
2.8.2	接口定义	13
2.8.3	设计细节	13
2.9	regfile	14
2.9.1	简介概述	14
2.9.2	接口定义	14
2.9.3	设计细节	14
2.10	hilo_reg	14
2.10.1	简介概述	14
2.10.2	接口定义	14
2.10.3	设计细节	14
<b>3</b>	<b>控制模块</b>	<b>15</b>
3.1	ctrl	15
3.1.1	简介概述	15
3.1.2	接口定义	15
3.1.3	设计细节	15
3.2	cp0_reg	15
3.2.1	简介概述	15
3.2.2	接口定义	15
3.2.3	设计细节	15
<b>4</b>	<b>内存管理</b>	<b>16</b>
4.1	tlb_reg	16
4.1.1	简介概述	16
4.1.2	接口定义	16
4.1.3	设计细节	16
4.2	cp0_reg	16
4.2.1	简介概述	16
4.2.2	接口定义	16
4.2.3	设计细节	16
4.3	openmips	17
4.3.1	简介概述	17
4.3.2	接口定义	17
4.3.3	设计细节	17

<b>5 外设连接</b>	<b>18</b>
5.1 ROM	18
5.1.1 简介概述	18
5.1.2 接口定义	18
5.1.3 设计细节	18
5.2 RAM	18
5.2.1 简介概述	18
5.2.2 接口定义	18
5.2.3 设计细节	18
5.3 Flash	19
5.3.1 简介概述	19
5.3.2 接口定义	19
5.3.3 设计细节	19
5.4 串口	19
5.4.1 简介概述	19
5.4.2 接口定义	19
5.4.3 设计细节	19
5.5 VGA	19
5.5.1 简介概述	19
5.5.2 接口定义	19
5.5.3 设计细节	19
5.6 七段数码管	20
5.6.1 简介概述	20
5.6.2 接口定义	20
5.6.3 设计细节	20
5.7 LED 灯	20
5.7.1 简介概述	20
5.7.2 接口定义	20
5.7.3 设计细节	20
5.8 开关	20
5.8.1 简介概述	20
5.8.2 接口定义	20
5.8.3 设计细节	20

# Chapter 1

## 文档说明

本文档是 NonExist 组软工计原联合项目的设计文档，作为设计文档，将会尽可能详细的覆盖到所有的设计方面和设计细节。

但是，设计文档呈现的是最终版 CPU，因此每个模块此时都已经经历了无数次蜕变，在这个过程中，每个模块的功能越来越多，也越来越复杂。本文档呈现的是最终版的设计，缺少了循序渐进的过程，因此读者初读起来可能感到困难，不建议将此文档作为前期主要参考文档。

本文档正确的使用方式是，开发初期通读全文，站在更高的层面上俯视了解整个项目的设计；开发后期精读细节，实现和完善具体的功能。

设计文档将项目分成了以下部分：

1. **指令流水**：本阶段实现 CPU 五级流水线及绝大部分基本指令。
2. **控制模块**：本阶段实现协处理器和异常处理
3. **内存管理**：本阶段实现内存管理。
4. **外设连接**：本阶段完成外设连接。
5. **仿真调试**：本阶段加入模拟的硬件模块完成仿真调试。

设计文档每个章节遵从以下介绍流程：

1. **简介概述**：简单介绍所实现元件的功能。
2. **接口定义**：实现的接口及其含义。
3. **设计细节**：详细而完善的设计思路与细节。

希望本文档能给读者带来裨益。

## Chapter 2

# 指令流水

本章介绍了 MIPS 标准五级流水线的实现，同时实现了绝大部分需要基本指令。

### 2.1 pc\_reg

#### 2.1.1 简介概述

pc\_reg 阶段实现程序计数，是五级流水线的第一级，记录了当前指令地址，同时对下一条指令地址进行计算与选择，是一个简单的时序逻辑电路。

#### 2.1.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	clk	CPU 外部	时钟信号
input	wire	1	rst	CPU 外部	复位信号
input	wire	6	stall	ctrl	流水线暂停使能
input	wire	1	tlb_hit	tlb_reg	TLB 是否命中
input	wire	32	physical_pc	tlb_reg	物理地址
input	wire	1	flush	ctrl	流水线清空使能
input	wire	32	new_pc	ctrl	下一指令地址
input	wire	1	branch_flag_i	id	跳转使能信号
input	wire	32	branch_target_addr_i	id	分支跳转地址
output	reg	32	virtual_pc	tlb_reg、if_id	虚拟地址
output	reg	32	pc	CPU 外部	指令地址
output	reg	1	ce	CPU 外部	访存使能信号
output	reg	32	excepttype_o	if_id	异常类型

### 2.1.3 设计细节

初始时 `virtual_pc` 指向 `0xbfc00000` ( ROM 起始地址 ), 之后 `virtual_pc` 每个时钟周期加 4, 若上一条指令为分支跳转指令, 则 `virtual_pc` 置为分支跳转地址。

## 2.2 if\_id

### 2.2.1 简介概述

`if_id` 用于衔接五级流水线第一阶段 `if` 和第二阶段 `id`, 在时钟上升沿储存 `if` 阶段数据, 传递给 `id` 阶段, 将 `if` 阶段所得到的指令传递至 `id` 阶段进行译码, 是一个简单的时序逻辑电路。

### 2.2.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	clk	CPU 外部	时钟信号
input	wire	1	rst	CPU 外部	复位信号
input	wire	32	if_pc	pc_reg	指令地址
input	wire	32	if_inst	pc_reg	指令内容
input	wire	32	if_excepttype	pc_reg	异常类型
input	wire	6	stall	ctrl	流水线暂停使能
input	wire	1	flush	ctrl	流水线清空使能
output	reg	32	id_pc	id	指令地址
output	reg	32	id_inst	id	指令内容
output	reg	32	id_excepttype	id	异常类型

### 2.2.3 设计细节

当 `if` 阶段没有被暂停, 置 `id_pc` 为 `if_id`, `id_excepttype` 为 `if_excepttype`, 此时如果没有发生 TLB 缺失异常, 置 `id_inst` 为 `if_inst`, 其余情况下 `id_pc`、`id_inst` 和 `id_excepttype` 均置 0。

## 2.3 id

### 2.3.1 简介概述

`id` 阶段实现指令译码, 是五级流水线的第二级, 主要是识别指令类型和各字段、读取通用寄存器值、产生流水线控制信号, 除此之外, `id` 阶段还需要实现数据旁路、分支判断处理等, 是一个复杂的组合逻辑电路。

### 2.3.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	rst	CPU 外部	复位信号
input	wire	32	pc_i	if_id	指令地址
input	wire	32	inst_i	if_id	指令内容
input	wire	32	reg1_data_i	regfile	通用寄存器读端口 1 数据
input	wire	32	reg2_data_i	regfile	通用寄存器读端口 2 数据
input	wire	1	ex_wreg_i	ex	旁路信号, EX 阶段是否写回通用寄存器
input	wire	32	ex_wdata_i	ex	旁路信号, EX 阶段写回通用寄存器数据
input	wire	5	ex_wd_i	ex	旁路信号, EX 阶段写回通用寄存器地址
input	wire	8	ex_aluop_i	ex	旁路信号, EX 阶段指令类型
input	wire	1	mem_wreg_i	mem	旁路信号, MEM 阶段是否写回通用寄存器
input	wire	32	mem_wdata_i	mem	旁路信号, MEM 阶段写回通用寄存器数据
input	wire	5	mem_wd_i	mem	旁路信号, MEM 阶段写回通用寄存器地址
input	wire	1	is_in_delay_slot_i	id_ex	当前指令是否位于分支延迟槽中
input	wire	32	excepttype_i	if_id	异常类型
output	reg	1	next_inst_in_delay_slot_o	id_ex	下一条指令是否位于分支延迟槽中
output	reg	1	branch_flag_o	pc_reg	跳转使能信号
output	reg	32	branch_target_addr_o	pc_reg	分支跳转地址
output	reg	32	link_addr_o	<TODO>	<TODO>
output	reg	1	is_in_delay_slot_o	id_ex	当前指令是否位于分支延迟槽中
output	reg	1	reg1_read_o	regfile	通用寄存器读端口 1 使能
output	reg	1	reg2_read_o	regfile	通用寄存器读端口 2 使能
output	reg	5	reg1_addr_o	regfile	通用寄存器读端口 1 地址
output	reg	5	reg2_addr_o	regfile	通用寄存器读端口 2 地址
output	reg	8	aluop_o	id_ex	ALU 运算类型
output	reg	3	alusel_o	id_ex	ALU 选择类型
output	reg	32	reg1_o	id_ex	ALU 第一个操作数
output	reg	32	reg2_o	id_ex	ALU 第二个操作数
output	reg	5	wd_o	id_ex	通用寄存器写端口地址
output	reg	1	wreg_o	id_ex	通用寄存器写端口使能
output	wire	32	inst_o	id_ex	指令内容
output	wire	1	stallreq	<TODO>	<TODO>
output	wire	32	excepttype_o	id_ex	异常类型
output	wire	32	current_inst_address_o	<TODO>	<TODO>



### 2.3.3 设计细节

ID 阶段将识别指令的操作码和各个字段，根据指令操作码向 EX 阶段传入其所需信号。<TODO>

## 2.4 id\_ex

### 2.4.1 简介概述

id\_ex 用于衔接五级流水线第二阶段 id 和第三阶段 ex，在时钟上升沿储存 id 阶段数据，传递给 ex 阶段，将 id 阶段译码阶段传递至 ex 阶段进行算数逻辑运算，是一个简单的时序逻辑电路。

### 2.4.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	clk	CPU 外部	时钟信号
input	wire	1	rst	CPU 外部	复位信号
input	wire	8	id_aluop	id	ALU 运算类型
input	wire	3	id_alusel	id	ALU 选择类型
input	wire	32	id_reg1	id	ALU 第一个操作数
input	wire	32	id_reg2	id	ALU 第二个操作数
input	wire	5	id_wd	id	通用寄存器写端口地址
input	wire	1	id_wreg	id	通用寄存器写端口使能
input	wire	32	id_link_addr	id	<TODO>
input	wire	1	id_is_in_delay_slot	id	当前指令是否位于分支延迟槽中
input	wire	1	next_inst_in_delay_slot_i	id	下一条指令是否位于分支延迟槽中
input	wire	32	id_inst	id	指令内容
input	wire	32	id_current_inst_address	id	<TODO>
input	wire	32	id_excepttype	id	异常类型
input	wire	6	stall	ctrl	流水线暂停使能
input	wire	1	flush	ctrl	流水线清空使能
output	reg	8	ex_aluop	ex	ALU 运算类型
output	reg	3	ex_alusel	ex	ALU 选择类型
output	reg	32	ex_reg1	ex	ALU 第一个操作数
output	reg	32	ex_reg2	ex	ALU 第二个操作数
output	reg	5	ex_wd	ex	通用寄存器写端口地址
output	reg	1	ex_wreg	ex	通用寄存器写端口使能
output	reg	32	ex_link_addr	ex	<TODO>
output	reg	1	ex_is_in_delay_slot	ex	当前指令是否位于分支延迟槽中
output	reg	1	is_in_delay_slot_o	ex	<TODO>
output	reg	32	ex_inst	ex	指令内容
output	reg	32	ex_current_inst_address	ex	指令地址
output	reg	32	ex_excepttype	ex	异常类型

### 2.4.3 设计细节

<TODO>

## 2.5 ex

### 2.5.1 简介概述

ex 阶段实现算数逻辑运算，是五级流水线的第三级，主要是进行各种算数逻辑运算，例如加法、减法、乘法、移位、与或非等操作，除此之外，ex 阶段还需要实现数据旁路、分支判断处理等，是一个复杂的组合逻辑电路。



## 2.5.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	rst	CPU 外部	复位信号
input	wire	8	aluop_i	id_ex	ALU 运算类型
input	wire	3	alusel_i	id_ex	ALU 选择类型
input	wire	32	reg1_i	id_ex	ALU 第一个操作数
input	wire	32	reg2_i	id_ex	ALU 第二个操作数
input	wire	5	wd_i	id_ex	通用寄存器写端口地址
input	wire	1	wreg_i	id_ex	通用寄存器写端口使能
input	wire	32	inst_i	id_ex	指令内容
input	wire	32	hi_i	hilo_reg	HI 寄存器读出数据
input	wire	32	lo_i	hilo_reg	LO 寄存器读出数据
input	wire	32	wb_hi_i	mem_wb	旁路信号, WB 阶段写回 HI 寄存器数据
input	wire	32	wb_lo_i	mem_wb	旁路信号, WB 阶段写回 LO 寄存器数据
input	wire	1	wb_whilo_i	mem_wb	旁路信号, WB 阶段是否写回 HILO 寄存器
input	wire	32	mem_hi_i	ex_mem	旁路信号, MEM 阶段写回 HI 寄存器数据
input	wire	32	mem_lo_i	ex_mem	旁路信号, MEM 阶段写回 LO 寄存器数据
input	wire	1	mem_whilo_i	ex_mem	旁路信号, MEM 阶段是否写回 HILO 寄存器
input	wire	32	link_addr_i	id_ex	LINK 地址
input	wire	1	is_in_delay_slot_i	id_ex	当前指令是否位于分支延迟槽中
input	wire	1	mem_cp0_reg_we	ex_mem	旁路信号, MEM 阶段是否写回 CP0 寄存器
input	wire	5	mem_cp0_reg_write_addr	ex_mem	旁路信号, MEM 阶段写回 CP0 寄存器地址
input	wire	32	mem_cp0_reg_data	ex_mem	旁路信号, MEM 阶段写回 CP0 寄存器数据
input	wire	1	wb_cp0_reg_we	mem_wb	旁路信号, WB 阶段是否写回 CP0 寄存器
input	wire	5	wb_cp0_reg_write_addr	mem_wb	旁路信号, WB 阶段写回 CP0 寄存器地址
input	wire	32	wb_cp0_reg_data	mem_wb	旁路信号, WB 阶段写回 CP0 寄存器数据
input	wire	32	cp0_reg_data_i	cp0_reg	CP0 协处理器寄存器读出数据
input	wire	32	excepttype_i	id_ex	异常类型
input	wire	32	current_inst_address_i	id_ex	指令地址
output	reg	5	wd_o	ex_mem	通用寄存器写端口地址
output	reg	1	wreg_o	ex_mem	通用寄存器写端口使能
output	reg	32	wdata_o	ex_mem	通用寄存器写端口数据
output	wire	32	inst_o	ex_mem	指令内容
output	reg	32	hi_o	ex_mem	HI 寄存器写入数据
output	reg	32	lo_o	ex_mem	LO 寄存器写入数据
output	reg	1	whilo_o	ex_mem	HILO 寄存器写使能
output	wire	8	aluop_o	ex_mem	ALU 运算类型
output	wire	32	mem_addr_o	ex_mem	<TODO>
output	wire	32	reg2_o	ex_mem	ALU 第二个操作数
output	wire	1	stallreq	<TODO>	<TODO>
output	reg	5	cp0_reg_read_addr_o	cp0_reg	CP0 协处理器寄存器读出地址
output	reg	1	cp0_reg_we_o	ex_mem	CP0 协处理器寄存器写使能
output	reg	5	cp0_reg_write_addr_o	ex_mem	CP0 协处理器寄存器写入地址

### 2.5.3 设计细节

<TODO>

## 2.6 ex\_\_mem

### 2.6.1 简介概述

ex\_\_mem 用于衔接五级流水线第三阶段 ex 和第四阶段 mem，在时钟上升沿储存 ex 阶段数据，传递给 mem 阶段，将 ex 阶段需要写入的数据传递至 mem 阶段进行访存操作，是一个简单的时序逻辑电路。

## 2.6.2 接口定义

信号类型	信号规格	信号位宽	信号名称	来源/去向	详细描述
input	wire	1	clk	CPU 外部	时钟信号
input	wire	1	rst	CPU 外部	复位信号
input	wire	5	ex_wd	ex	通用寄存器写端口地址
input	wire	1	ex_wreg	ex	通用寄存器写端口使能
input	wire	32	ex_wdata	ex	通用寄存器写端口数据
input	wire	32	ex_hi	ex	HI 寄存器写入数据
input	wire	32	ex_lo	ex	LO 寄存器写入数据
input	wire	1	ex_whilo	ex	HILO 寄存器写使能
input	wire	8	ex_aluop	ex	ALU 运算类型
input	wire	32	ex_mem_addr	ex	<TODO>
input	wire	32	ex_reg2	ex	ALU 第二个操作数
input	wire	1	ex_cp0_reg_we	ex	CP0 协处理器寄存器写使能
input	wire	5	ex_cp0_reg_write_addr	ex	CP0 协处理器寄存器写入地址
input	wire	32	ex_cp0_reg_data	ex	CP0 协处理器寄存器写入数据
input	wire	32	ex_excepttype	ex	异常类型
input	wire	1	ex_is_in_delay_slot	ex	当前指令是否位于分支延迟槽中
input	wire	32	ex_current_inst_address	ex	指令地址
input	wire	32	ex_inst	ex	指令内容
input	wire	6	stall	ctrl	流水线暂停使能
input	wire	1	flush	ctrl	流水线清空使能
output	reg	5	mem_wd	mem	通用寄存器写端口地址
output	reg	1	mem_wreg	mem	通用寄存器写端口使能
output	reg	32	mem_wdata	mem	通用寄存器写端口数据
output	reg	32	mem_hi	mem	HI 寄存器写入数据
output	reg	32	mem_lo	mem	LO 寄存器写入数据
output	reg	1	mem_whilo	mem	HILO 寄存器写使能
output	reg	8	mem_aluop	mem	ALU 运算类型
output	reg	32	mem_mem_addr	mem	
output	reg	32	mem_reg2	mem	ALU 第二个操作数
output	reg	1	mem_cp0_reg_we	mem	CP0 协处理器寄存器写使能
output	reg	5	mem_cp0_reg_write_addr	mem	CP0 协处理器寄存器写入地址
output	reg	32	mem_cp0_reg_data	mem	CP0 协处理器寄存器写入数据
output	reg	32	mem_excepttype	mem	异常类型
output	reg	1	mem_is_in_delay_slot	mem	当前指令是否位于分支延迟槽中
output	reg	32	mem_current_inst_address	mem	指令地址
output	reg	32	mem_inst	mem	指令内容

### 2.6.3 设计细节

<TODO>

## 2.7 mem

### 2.7.1 简介概述

a

### 2.7.2 接口定义

a

### 2.7.3 设计细节

a

## 2.8 mem\_\_wb

### 2.8.1 简介概述

a

### 2.8.2 接口定义

a

### 2.8.3 设计细节

a

## 2.9 regfile

### 2.9.1 简介概述

a

### 2.9.2 接口定义

a

### 2.9.3 设计细节

a



## **2.10 hilo\_reg**

### **2.10.1 简介概述**

a

### **2.10.2 接口定义**

a

### **2.10.3 设计细节**

a

# Chapter 3

## 控制模块

### 3.1 ctrl

#### 3.1.1 简介概述

a

#### 3.1.2 接口定义

a

#### 3.1.3 设计细节

a

### 3.2 cp0\_\_reg

#### 3.2.1 简介概述

a

#### 3.2.2 接口定义

a

#### 3.2.3 设计细节

a

# Chapter 4

## 内存管理

### 4.1 tlb\_reg

#### 4.1.1 简介概述

a

#### 4.1.2 接口定义

a

#### 4.1.3 设计细节

a

### 4.2 cp0\_reg

#### 4.2.1 简介概述

a

#### 4.2.2 接口定义

a

#### 4.2.3 设计细节

a

## 4.3 openmips

### 4.3.1 简介概述

a

### 4.3.2 接口定义

a

### 4.3.3 设计细节

a

# Chapter 5

## 外设连接

### 5.1 ROM

#### 5.1.1 简介概述

a

#### 5.1.2 接口定义

a

#### 5.1.3 设计细节

a

### 5.2 RAM

#### 5.2.1 简介概述

a

#### 5.2.2 接口定义

a

#### 5.2.3 设计细节

a

## **5.3 Flash**

### **5.3.1 简介概述**

a

### **5.3.2 接口定义**

a

### **5.3.3 设计细节**

a

## **5.4 串口**

### **5.4.1 简介概述**

a

### **5.4.2 接口定义**

a

### **5.4.3 设计细节**

a

## **5.5 VGA**

### **5.5.1 简介概述**

a

### **5.5.2 接口定义**

a

### **5.5.3 设计细节**

a

## 5.6 七段数码管

### 5.6.1 简介概述

a

### 5.6.2 接口定义

a

### 5.6.3 设计细节

a

## 5.7 LED 灯

### 5.7.1 简介概述

a

### 5.7.2 接口定义

a

### 5.7.3 设计细节

a

## 5.8 开关

### 5.8.1 简介概述

a

### 5.8.2 接口定义

a

### 5.8.3 设计细节

a