

# 流水线 CPU 的 verilog 设计

北京航空航天大学二队

流水线 CPU 的 verilog 设计 .....	1
1.概述 .....	1
2.优点 .....	2
2.1 七级流水.....	2
2.2 将读写字节使能信号的生成与流水线控制分离.....	2
2.3 PC 回写机制 .....	3
2.4 指令衔接机制.....	3
2.5 除法采用 17 周期实现.....	3
3.缺点 .....	3
4.模块介绍.....	4
4.1 IF_1_HALF .....	4
IF_1_HALF 子模块: .....	4
4.2 IF_2_HALF .....	5
4.3 ID and WB.....	5
子模块: .....	6
D 级 .....	6
W 级 .....	8
4.4 EX.....	8
EX 子模块: .....	9
4.5 MEM_1_HALF.....	10
MEM_1_HALF 子模块: .....	11
4.6 MEM_2_HALF.....	12
MEM_2_HALF 子模块.....	12
4.7 转发暂停控制单元.....	13

## 1.概述

我们的作品是一款**单发射**的流水线 CPU，功能及特点如下：

1. 支持延迟槽
2. 支持大赛要求的 57 条指令
3. 支持暂停和转发
4. 支持大赛要求的中断和异常类型
5. 七级流水
6. 将字节读写使能信号的生成与流水线控制信号分离
7. 新增 PC 回写机制
8. 加入指令衔接机制
9. 除法 17 个周期实现

其中 5-9 在第三部分特点中详细说明。

本 cpu 的七个流水级由经典五级流水切分而来，分别为 IF\_1\_HALF, IF\_2\_HALF, ID, EX, MEM\_1\_HALF, MEM\_2\_HALF, WB, 其中 IF\_1\_HALF 和 IF\_2\_HALF 由 IF 级切分而来，MEM\_1\_HALF 和 MEM\_2\_HALF 由 MEM 级切分而来，另设一个 Hazard\_unit 转发控制单元控制整个流水线。

## 2. 优点

**时钟周期 16.368ns，同等约束下较 gs132 快 103%。**在尽力减少运行周期数并优化时钟周期的过程中，我们尝试了下列方法：

### 2.1 七级流水

我们的 cpu 设计为七级流水，相较于经典的五级流水，**将取指(IF)和访存(MEM)两级分别切分为两段**，分别命名为 IF\_1\_HALF 和 IF\_2\_HALF，MEM\_1\_HALF，MEM\_2\_HALF。

针对所有指令，IF\_1\_HALF 级负责向外发送取指请求，拿到地址握手信号后进入 IF\_2\_HALF 级。IF\_2\_HALF 级负责接收由 IF\_1\_HALF 级发出的请求所返回的数据。

除此之外，针对访存指令，MEM\_1\_HALF 级负责向外发送访存请求，拿到地址握手信号后进入 MEM\_2\_HALF 级。MEM\_2\_HALF 级负责接收由 MEM\_1\_HALF 级发出的请求所返回的数据。

### 2.2 将读写字节使能信号的生成与流水线控制分离

只要 pc 低两位为 0，if\_ben(取指的字节读写使能信号)恒为 4'b1111。mem\_ben(访存的字节读写使能信号)与 MEM\_1\_HALF 级译码得到的字节使能信号一致。以**接收数据后将数据舍弃**的方式取代**不发送请求**的方式，消除了先前由 if\_data\_ok 到 if\_ben 和由 mem\_data\_ok 到 if\_ben 的关键路径，将理想时钟周期从 29.2ns 降至 18.85ns，降幅达 35.4%。

## 2.3 PC 回写机制

若取值请求的数据返回时，cpu 正处于暂停状态，ID 级无法接收新指令，则将该指令舍弃，并将该指令的 pc 会写入 pc 寄存器中，待流水线重新流动时，重新发出请求。

我们还尝试了在相同情况下将该指令保存在寄存器中，待流水线重新流动时直接进入 ID 级，以性能测试为参考，发现不回写机制所用周期比回写机制少约 **1.5%**，但是理想时钟周期会提高约 **4%**，综合来看，回写机制略胜于不回写机制，因此我们决定采用回写机制。

## 2.4 指令衔接机制

每条指令在 D 级都会存有它下一条指令的 pc，若某指令保存的它下一条指令的 pc 与实际它后面指令的 pc 不符，则该实际在后面的 pc 为非法 pc，非法 pc 取回的数据将舍弃。主要应用于**舍弃延迟槽后的指令**

## 2.5 除法采用 17 周期实现

我们对比了两种实现除法的方式：第一种为 33 周期实现（32 周期运算，1 周期写 HI,LO 寄存器），第二种为 17 周期实现（16 周期运算，1 周期写 HI,LO 寄存器），即第二种方法每周期运算两位。

以性能测试和时钟综合结果来看，采用第一种方法时，理想时钟周期为 18.845ns，性能测试平均周期数为龙芯 gs132 的 98.0%，采用第二种方法时，理想时钟周期为 18.850ns，性能测试平均周期数为龙芯 gs132 的 97.6%。综合来看，以万分之二点六的时钟周期代价换取周期数减少千分之四是划算的，除法采用 17 周期性能更好。

## 3.缺点

- 1.十个性能测试的几何平均周期数仅比 gs132 少了 2.6%，仍有提升空间。
- 2.执行过程中插入气泡较多

## 4.模块介绍

### 4.1 IF\_1\_HALF

相当于经典五级流水取指模块的前半部分，功能如下：

1. 放置 pc 寄存器
2. 向指令 ram 发出取指请求并接收地址握手信号
3. 检测取指地址未对齐异常

表一 IF\_1\_HALF 接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
back_pc	I	pc 回退信号
PC_en	I	pc 使能信号
int_PC_sel	I	中断时的 pc 选择信号
ERET_PC_sel	I	eret 指令执行时的 pc 选择信号
JUMP_sel	I	跳转指令执行时的 pc 选择信号
b_j_r_tgt[31:0]	I	跳转目标地址
EPC[31:0]	I	异常处理程序的返回地址
PC_IFMID_out[31:0]	I	pc 回退地址
PC4_D[31:0]	I	D 级指令对应的 pc+4
inst_sram_addr[31:0]	O	取指地址
PC[31:0]	O	当前 PC
EXC_D[6:2]	O	传给下一级的异常码
pc_valid	O	pc 是否有效

### IF\_1\_HALF 子模块：

PC 模块：根据控制信号从众多地址中选择一个作为 pc 值。

表二 PC 模块接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
back_pc	I	pc 回退信号
PC_en	I	pc 使能信号
int_PC_sel	I	中断时的 pc 选择信号
ERET_PC_sel	I	eret 指令执行时的 pc 选择信号
JUMP_sel	I	跳转指令执行时的 pc 选择信号
PC4[31:0]	I	pc+4

b_jjr_tgt[31:0]	I	跳转目标地址
EPC[31:0]	I	异常处理程序的返回地址
PC_IFMID_out	I	pc 回退地址
PC	O	当前 pc 值

pc 自增模块：将输入的 pc 值加 4 后输出

表三 pc 自增模块接口表

端口名称	方向	端口描述
PC	I	当前 pc
PC4	O	当前 pc+4

## 4.2 IF\_2\_HALF

相当于经典五级流水取指模块的后半部分，功能如下：

1. 若 data\_ok 信号返回，则接收返回的数据
2. 若 data\_ok 信号未返回，则向 IF/ID 级流水寄存器输出 0

表四 IF\_2\_HALF 接口表

端口名称	方向	端口描述
inst_sram_rrdy	I	表示指令是否收到
int_eret_refuse	I	控制信号，用于控制是否舍弃本周期接收到的指令
KEEP	I	控制信号，用于控制是否舍弃本周期接收到的指令
PC_IFMID[31:0]	I	来自 IFMID 流水线寄存器的 pc 值
inst_data[31:0]	I	收到的指令
PC4_D	O	输出给下一级的 pc+4
IR_D	O	输出给下一级的指令
EXC_IFMID	I	上一级传下来的异常码
EXC_D	O	输出给下一级的异常码
NPC_IFMID	I	上一级传下来的下一条指令地址
NPC_D	O	输出给下一级的下一条指令地址
NPC_D_out	I	D 级指令的下一条指令的地址

## 4.3 ID and WB

功能如下：

1. 译码并读取通用寄存器堆(D 级)
2. 如果跳转，产生跳转后的地址(D 级)
3. 检测保留指令异常、break 异常、syscall 异常(D 级)

#### 4. 将数据写回寄存器堆(W 级)

表五 IDandWB 级接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
Exc_in[6:2]	I	由上级传下来的异常码
Exc_out[6:2]	O	传给下一级的异常码
IR_D[31:0]	I	D 级指令
PC4_D[31:0]	O	D 级指令的 pc+4
IR_E[31:0]	O	输出给 E 级的指令
RS_E[31:0]	O	输出给 E 级的 rs 字段对应寄存器的值
RT_E[31:0]	O	输出给 E 级的 rt 字段对应寄存器的值
EXT_E[31:0]	O	输出给 E 级的立即数扩展结果
PC8_E[31:0]	O	输出给 E 级的 pc+8
b_jjr_tgt[31:0]	O	跳转目标地址
PC_sel	O	PC 跳转地址选择信号
IR_W [31:0]	I	W 级指令
PC8_W[31:0]	I	W 级指令 pc+8
ALUOUT_W[31:0]	I	传至 W 级的常规运算模块输出
DMOUT_W[31:0]	I	传至 W 级的从数据 ram 返回的数据
XALUOUT_W	I	传至 W 级的乘除运算模块输出
Wdata[31:0]	O	待写入寄存器堆的数据
Forward_RS_D[2:0]	I	D 级 rs 转发信号
Forward_RT_D[2:0]	I	D 级 rt 转发信号
ALUOUT_M_out[31:0]	I	来自 MEM_1_HALF 级的 ALUOUT
PC8_M_out[31:0]	I	来自 MEM_1_HALF 级的 pc+8
XALUOUT_M_out[31:0]	I	来自 MEM_1_HALF 级的 XALUOUT
ALUOUT_MMID_out[31:0]	I	来自 MEM_2_HALF 的 ALUOUT
PC8_MMID_out[31:0]	I	来自 MEM_2_HALF 级的 pc+8
XALUOUT_MMID_out[31:0]	I	来自 MEM_2_HALF 级的 XALUOUT

### 子模块:

#### D 级

D 级转发模块：将还未写入寄存器堆的数据传递给 D 级的需求者。

表六 D 级转发模块接口表

端口名称	方向	端口描述
PRE[31:0]	I	无转发时的原始数据
ALUOUT_M[31:0]	I	MEM_1_HALF 级的 ALUOUT
PC8_M[31:0]	I	MEM_1_HALF 级的 PC8

XALUOUT_M[31:0]	I	MEM_1_HALF 级的 XALUOUT
ALUOUT_MMID[31:0]	I	MEM_2_HALF 级的 ALUOUT
PC8_MMID[31:0]	I	MEM_2_HALF 级的 PC8
XALUOUT_MMID[31:0]	I	MEM_2_HALF 级的 XALUOUT
Forward_sel[2:0]	I	选择信号
sel_result[31:0]	O	选择结果

寄存器堆：放置 32 个通用寄存器

表七 寄存器堆接口表

端口名称	方向	端口描述
Rreg1[4:0]	I	第一个读寄存器的地址
Rreg2[4:0]	I	第二个读寄存器的地址
Rdata1	O	第一个读寄存器的内容
Rdata2	O	第二个读寄存器的内容
Wreg[4:0]	I	写寄存器的地址
Wdata[31:0]	I	待写入的数据
GRF_WE	I	寄存器写使能信号
clk	I	时钟信号
reset	I	复位信号

立即数扩展器：对输入的 16 位立即数进行符号扩展或无符号扩展

表八 立即数扩展接口表

端口名称	方向	端口描述
imm16[15:0]	I	16 位立即数
extop	I	扩展控制信号，决定是符号扩展还是无符号扩展
extout[31:0]	O	扩展结果

跳转地址生成器：生成分支跳转指令的目的地址

表九 跳转地址生成接口表

端口名称	方向	端口描述
PC4_D[31:0]	I	D 级 pc+4
imm26[25:0]	I	26 位立即数
jr_tgt[31:0]	I	jr/jalr 跳转目的地址
b_j_jr_sel[1:0]	I	跳转地址选择信号
npc[31:0]	O	生成的跳转地址

D 级控制器：现仅用于检测是否是保留指令例外

表十 D 级控制器接口表

端口名称	方向	端口描述
IR_D[31:0]	I	D 级指令
Illegal	O	非法指令信号

此外，还是用了 pc 自增模块，将输入的 PC4 加四后输出位 PC8，接口已在 IF\_1\_HALF

中给出。

## W 级

数据扩展器：将 lb, lbu, lh, lhu 等指令返回的有效的字节或半字扩展为字

表十一 W 级数据扩展器接口表

端口名称	方向	端口描述
ADDR[1:0]	I	访存地址
DMOUT[31:0]	I	数据 ram 返回的数据
XEXT_OP[2:0]	I	扩展控制信号
XEXTOUT[31:0]	O	扩展结果

W 级控制器

表十二 W 级控制器接口表

端口名称	方向	端口描述
IR[31:0]	I	指令
Wreg_sel	O	Wreg 选择信号
Wdata_sel	O	Wdata 选择信号
GRF_WE	O	GRF 写使能信号
XEXT_OP[2:0]	O	XEXT 模块控制信号

## 4.4 EX

功能如下：

1. 运算
2. 对于 add, addi, sub，检测计算结果是否溢出

表十三 E 级接口表

端口名称	方向	端口描述
IR_E [31:0]	I	E 级指令
RS_E [31:0]	I	上一级传下来的 rs 字段对应寄存器的值
RT_E [31:0]	I	上一级传下来的 rt 字段对应寄存器的值
EXT_E [31:0]	I	上一级传下来的扩展结果
PC8_E [31:0]	I	上一级传下来的 pc+8
IR_M [31:0]	O	传给 MEM_1_HALF 级的指令
PC8_M [31:0]	O	传给 MEM_1_HALF 级的 pc+8
ALUOUT_M [31:0]	O	传给 MEM_1_HALF 级的 ALUOUT
RT_M [31:0]	O	传给 MEM_1_HALF 级的 rt 字段对应寄存器的值
Forward_RS_E [2:0]	I	E 级 rs 转发信号
Forward_RT_E [2:0]	I	E 级 rt 转发信号
ALUOUT_M_out[31:0]	I	来自 MEM_1_HALF 级的 ALUOUT



mux_Wdata_out[31:0]	I	来自 W 级的 Wdata
PC8_M_out[31:0]	I	来自 MEM_1_HALF 级的 pc+8
PC8_W_out[31:0]	I	来自 W 级的 pc+8
clk	I	时钟信号
reset	I	复位信号
Exc_in[6:2]	I	上一级传下来的异常码
Exc_out[6:2]	O	传给下一级的异常码
XALUOUT_M[31:0]	I	传给 MEM_1_HALF 级的 XALUOUT
XALUOUT_M_out[31:0]	I	来自 MEM_1_HALF 级的 XALUOUT
XALUOUT_W_out[31:0]	I	来自 W 级的 XALUOUT
ALUOUT_MMID_out[31:0]	I	来自 MEM_2_HALF 级的 ALUOUT
PC8_MMID_out[31:0]	I	来自 MEM_2_HALF 级的 pc+8
XALUOUT_MMID_out[31:0]	I	来自 MEM_2_HALF 级的 XALUOUT
BUSY	O	乘除模块忙碌信号

## EX 子模块:

### E 级转发模块

表十四 E 级转发模块接口表

端口名称	方向	端口描述
PRE[31:0]	I	无转发时的原始数据
PC8_M[31:0]	I	MEM_1_HALF 级的 pc+8
ALUOUT_M[31:0]	I	MEM_1_HALF 级的 ALUOUT
mux_Wdata[31:0]	I	W 级的 Wdata
PC8_W[31:0]	I	W 级的 pc+8
XALUOUT_M[31:0]	I	MEM_1_HALF 级的 XALUOUT
XALUOUT_W[31:0]	I	W 级的 XALUOUT
ALUOUT_MMID[31:0]	I	MEM_2_HALF 级的 ALUOUT
PC8_MMID[31:0]	I	MEM_2_HALF 级的 pc+8
XALUOUT_MMID[31:0]	I	MEM_2_HALF 级的 XALUOUT
Forward_sel[2:0]	I	转发选择信号
sel_result[31:0]	O	选择结果

### ALU

表十五 ALU 接口表

端口名称	方向	端口描述
IR_E[10:6]	I	指令的 shamt 字段
A[31:0]	I	操作数一
B[31:0]	I	操作数二
ALU_OP[3:0]	I	操作信号
C[31:0]	O	运算结果

## 乘除运算模块(XALU)

表十六 XALU 接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
HI_WE	I	HI 寄存器写使能信号
LO_WE	I	LO 寄存器写使能信号
XALUOUT_sel	I	输出选择信号
XALU_OP[3:0]	I	乘除运算控制信号
A[31:0]	I	操作数一
B[31:0]	I	操作数二
XALU_Wdata[31:0]	I	待写入数据
XALUOUT[31:0]	O	乘除模块输出
BUSY	O	乘除模块忙碌信号
div0	O	除 0 异常信号

## E 级控制器

表十七 E 级控制器接口表

端口名称	方向	端口描述
IR[31:0]	I	E 级指令
A_B_SIGN_less	I	有符号比较, A 是否小于 B
A_B_UNSIGN_less	I	无符号比较, A 是否小于 B
ALU_B_sel	O	ALU 操作数二的选择信号
ALU_OP[3:0]	O	ALU 控制信号
XALU_OP[3:0]	O	XALU 控制信号
HI_WE	O	HI 寄存器写使能信号
LO_WE	O	LO 寄存器写使能信号
HI_LO_sel	O	HI,LO 寄存器选择信号
add_addi	O	指令是否为 add/addi
_sub	O	指令是否为 sub

## 4.5 MEM\_1\_HALF

相当于经典五级流水 MEM 级的前半部分, 功能如下:

1. 若为访存指令, 则向数据 ram 发送请求并接收地址握手信号
2. 检测访存地址不对齐异常
3. 将虚拟地址转换为物理地址

表十八 MEM\_1\_HALF 级接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号

IR_M[31:0]	I	MEM_1_HALF 级指令
PC8_M[31:0]	I	MEM_1_HALF 级 PC+8
ALUOUT_M[31:0]	I	MEM_1_HALF 级 ALUOUT
RT_M[31:0]	I	MEM_1_HALF 级 rt 字段对应寄存器的值
XALUOUT_M[31:0]	I	MEM_1_HALF 级 XALUOUT
EXC_M[6:2]	I	上一级传下来的异常码
IR_MMID[31:0]	O	传给 MEM_2_HALF 级的指令
PC8_MMID[31:0]	O	传给 MEM_2_HALF 级的 pc+8
ALUOUT_MMID[31:0]	O	传给 MEM_2_HALF 级的 ALUOUT
RT_MMID[31:0]	O	传给 MEM_2_HALF 级的 rt 字段对应寄存器的值
XALUOUT_MMID[31:0]	O	传给 MEM_2_HALF 级的 XALUOUT
EXC_MMID[6:2]	O	传给 MEM_2_HALF 级的异常码
Forward_RT_M_1_HALF[2:0]	I	M_1_HALF 级转发控制信号
mux_Wdata_out[31:0]	I	W 级的 Wdata
PC8_W_out[31:0]	I	W 级的 pc+8
XALUOUT_W_out[31:0]	I	W 级的 XALUOUT
ALUOUT_MMID_out[31:0]	I	MEM_2_HALF 级的 ALUOUT
PC8_MMID_out[31:0]	I	MEM_2_HALF 级的 pc+8
XALUOUT_MMID_out[31:0]	I	MEM_2_HALF 级的 XALUOUT
data_sram_addr[31:0]	O	访问数据 ram 的地址
data_sram_wdata[31:0]	O	待写入数据
data_sram_addr_illegal	O	访存地址非法信号
byte_en	O	访存字节使能信号

## MEM\_1\_HALF 子模块:

MEM\_1\_HALF 转发控制模块:

表十九 MEM\_1\_HALF 转发控制模块接口表

端口名称	方向	端口描述
PRE[31:0]	I	无转发时的原始数据
mux_Wdata[31:0]	I	W 级的 Wdata
PC8_W[31:0]	I	W 级的 pc+8
XALUOUT_W[31:0]	I	W 级的 XALUOUT
ALUOUT_MMID[31:0]	I	MMID 级的 ALUOUT
PC8_MMID[31:0]	I	MMID 级的 pc+8
XALUOUT_MMID[31:0]	I	MMID 级的 XALUOUT
Forward_sel[2:0]	I	转发选择信号
sel_result[31:0]	O	选择结果

## 4.6 MEM\_2\_HALF

相当于经典五级流水 MEM 级后半部分，功能如下：

1. 接收访存的结果
2. 包含 CP0 模块，处理异常

表二十 MEM\_2\_HALF 级接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
delay_groove	I	延迟槽信号
EXC_MMID[6:2]	I	上一级传下来的异常码
IR_MMID[31:0]	I	MEM_2_HALF 级指令
PC8_MMID[31:0]	I	上一级传下来的 pc+8
ALUOUT_MMID[31:0]	I	上一级传下来的 ALUOUT
RT_MMID[31:0]	I	上一级传下来的 rt 字段对应寄存器的值
XALUOUT_MMID[31:0]	I	上一级传下来的 XALUOUT
IR_W[31:0]	O	传给 W 级的指令
PC8_W[31:0]	O	传给 W 级的 pc+8
ALUOUT_W[31:0]	O	传给 W 级的 ALUOUT
DMOUT_W[31:0]	O	传给 W 级的数据 ram 返回的数据
XALUOUT_W[31:0]	O	传给 W 级的 XALUOUT
Forward_RT_M[1:0]	I	MEM_2_HALF 级转发信号
mux_Wdata_out[31:0]	I	W 级的 Wdata
PC8_W_out[31:0]	I	W 级的 pc+8
XALUOUT_W_out[31:0]	I	W 级的 XALUOUT
HWInt[7:2]	I	来自外部的硬件中断信号
int_clr	O	异常产生信号
EPC[31:0]	O	中断处理程序返回地址
DMOUT[31:0]	I	从数据 ram 返回的数据

## MEM\_2\_HALF 子模块

MEM\_2\_HALF 级转发模块

表二十一 MEM\_2\_HALF 级转发模块接口表

端口名称	方向	端口描述
PRE	I	无转发时的原始数据
mux_Wdata	I	W 级的 Wdata
PC8_W	I	W 级的 pc+8
XALUOUT_W	I	W 级的 XALUOUT
Forward_sel	I	转发控制信号
sel_result	O	选择结果

## MEM\_2\_HALF 级控制器

表二十二 MEM\_2\_HALF 级控制器接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
IR	I	MEM_2_HALF 级指令
CP0_WE	O	CP0 写使能信号
EXL_clr	O	CP0 的 exl 位清零信号
CP0_sel	O	CP0 选择信号
SL	O	指令是否为 load 或 store 类型的指令

## CP0

表二十三 CP0 接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
SL	I	指令是否为 load 或 store 类型的指令
epc_sel	I	异常指令是否为延迟槽指令
CP0_RWreg[4:0]	I	读写 CP0 寄存器编号
CP0_Wdata[31:0]	I	待写入的数据
PC[31:0]	I	待写入 epc 寄存器的中断处理程序的返回地址
Bad_PC8[31:0]	I	为取指地址例外准备的非法 pc+8
Bad_addr[31:0]	I	非法访存地址
HWint[7:2]	I	硬件中断信号
CP0_WE	I	CP0 寄存器写使能信号
EXL_clr	I	exl 清零信号
int_clr	O	异常信号
EPC	O	中断处理程序的返回地址
CP0_Dataout	O	从 CP0 中读出的数据
Exc_in	I	异常码

## 4.7 转发暂停控制单元

功能如下：

1. 控制各级流水线寄存器的使能信号，清空信号
2. 控制流水线的转发与暂停

表二十四 转发暂停控制单元主要接口表

端口名称	方向	端口描述
clk	I	时钟信号
reset	I	复位信号
int_clr	I	中断异常信号

data_sram_addr_illegal	I	访存地址非法信号
IR_D[31:0]	I	D 级指令
IR_E[31:0]	I	E 级指令
IR_M[31:0]	I	M_1_HALF 级指令
IR_MMID[31:0]	I	M_2_HALF 级指令
IR_W[31:0]	I	W 级指令
PC_IFMID_out[31:0]	I	IFMID 级 pc
PC[31:0]	I	当前 pc
PC4_D[31:0]	I	D 级 pc+4
EPC[31:0]	I	当前 CP0 的 epc 寄存器的值
NPC_D_out[31:0]	I	D 级指令的下一条指令地址
IF_GET_ADDR_OK	I	取指阶段地址握手信号
IF_GET_DATA_OK	I	取指阶段数据握手信号
MEM_GET_ADDR_OK	I	访存阶段地址握手信号
MEM_GET_DATA_OK	I	访存阶段数据握手信号
PC_sel	I	初级跳转信号
BUSY	I	乘除模块忙碌信号
if_ben[3:0]	I	取指阶段字节读使能信号
mem_ben[3:0]	I	访存阶段字节读使能信号
KEEP	I	标记当前 IFMID 级 pc 是否要保留
Forward_RS_D[2:0]	O	D 级 rs 转发信号
Forward_RT_D[2:0]	O	D 级 rt 转发信号
Forward_RS_E[3:0]	O	E 级 rs 转发信号
Forward_RT_E[3:0]	O	E 级 rt 转发信号
Forward_RT_M_1_HALF[1:0]	O	MEM_1_HALF 级 rt 转发信号
Forward_RT_M_2_HALF[1:0]	O	MEM_2_HALF 级 rtd 转发信号
int_eret_refuse	O	因中断异常或 eret 的执行而导致拒收当前从指令 ram 中返回的数据的信号
JUMP_sel	O	终极跳转信号
ERET_PC_sel	O	eret 指令执行时的 pc 选择信号
delay_groove_IFMID	O	发送给 IFMID 级流水线寄存器的延迟槽标记
delay_groove_IF_ID	O	发送给 IF/ID 级流水线寄存器的延迟槽标记
back_pc	O	pc 回退信号
PC_en	O	Pc 使能信号
IF_MID_en	O	IFMID 级流水线寄存器使能信号
IF_MID_clr	O	IFMID 级流水线寄存器清空信号
IF_ID_en	O	IF/ID 级流水线寄存器使能信号
IF_ID_clr	O	IF/ID 级流水线寄存器清空信号
ID_EX_en	O	ID/EX 级流水线寄存器使能信号
EX_MEM_en	O	EX/MEM 级流水线寄存器使能信号
MEM_MMID_en	O	MEM_MID 级流水线寄存器使能信号
MEM_MMID_clr	O	MEM_MID 级流水线寄存器清空信号

MMID_WB_en	O	MEM/WB 级流水线寄存器使能信号
MMID_WB_clr	O	MEM/WB 级流水线寄存器清空信号
ID_EX_clr	O	ID/EX 级流水线寄存器清空信号
data_sram_wr	O	数据 ram 读写信号