

CPU 设计报告

南京师范大学 1 队
张心妍 范泽华 蒋雨馨 牛籽心

一、设计简介

我队设计了一个 SRAM 接口的 5 级流水线 CPU，流水线分为取指令 (IF)、指令译码 (ID)、执行 (EX)、存储器访问 (Mem) 和寄存器写回 (Wr) 五级，满足大赛组委会要求的初阶 CPU 要求。该 CPU 能实现大赛组委会提供的 MIPS 指令集，即能运行大赛组委会要求的 57 条指令和响应中断与例外。

在 Vivado 2018.1 环境下，能正确运行 trace 比对，通过 89 个功能点。设置主频为 10MHz，综合后在龙芯试验箱上能正确上板测试，通过 89 个功能点。

二、设计方案

(一) 总体设计思路

5 级流水线 CPU 的结构^[1]如图 1 所示，分为 5 个流水段，分别为取指令 (IF)、指令译码 (ID)、执行 (EX)、存储器访问 (Mem) 和寄存器写回 (Wr)。PC 为程序计数器。IF/ID、ID/Ex、Ex/Mem 和 Mem/Wr 分别是各个流水段间的段寄存器，用于存放从当前段传到后面流水段的信息。每一个流水段有自己的功能部件，分别由相应的模块实现。下面详细叙述每个流水段的功能结构。

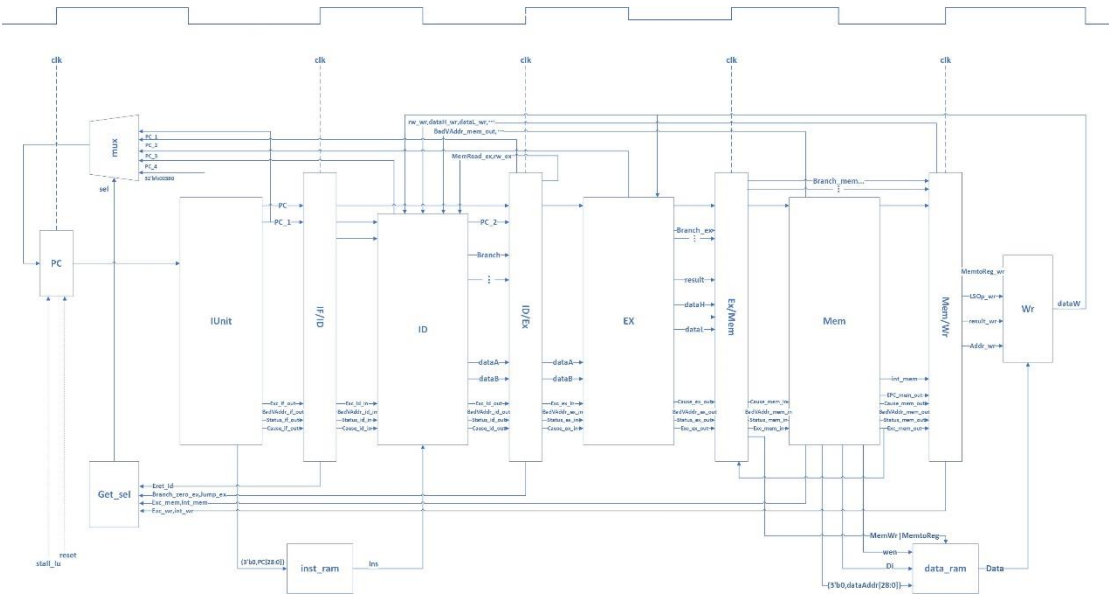


图 1.5 5 级流水线基本结构框图

(二) IF 段设计

IF 段包括 Ifetch.v, 基本功能部件有程序计数器 PC、Adder、GetSel、mux5to1 和 Exc_deal。IF 模块的结构如图 2 所示。程序计数器用于存储当前 PC 值; Adder 用于在指令顺序执行时实现 PC+4 操作; GetSel 用于得到 NewPC 的选择信号; mux5to1 用于选择 NewPC; Exc_deal 用于在发生例外时产生例外信息。IF 段执行的结果被送到 IF/ID 寄存器保存, 用于将数据和控制信号向下一段传递。

PC 的计算有 5 种情况:

- (1) 顺序执行时, 通过 PC+4 计算得到, 在 IF 段由 Adder 实现。
- (2) 对于分支指令, 通过 $PC+4+\{\{14\{imm16[15]\}\}, imm16, 2'b00\}$ 计算得到, 在 ID 段实现。
- (3) 对于跳转指令, 通过选择 $\{PC_1[31:28], ins[25:0], 2'b00\}$ 和寄存器 rs 中的值得到, 在 EX 段实现。
- (4) 从例外处理返回时, 通过取 CP0[EPC]中的值得到, 在 ID 段实现。
- (5) 产生例外时, 直接赋予例外入口地址 0xBFC00380, 在 IF 段实现。

IF 段的结构如图 2 所示。

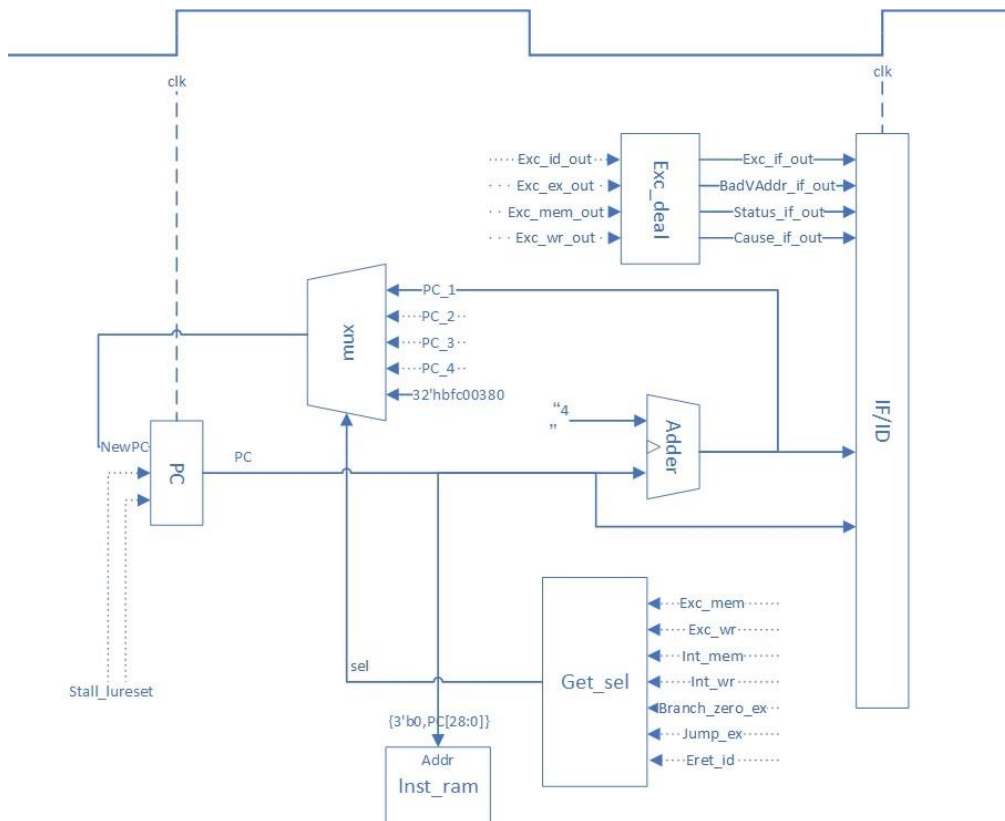


图 2. IF 段的结构示意图

表 1 列出了 IF 段的输入、输出信号。

表 1. IF 段输入输出信号

输入信号名称	输入信号解释
Clk	时钟信号
Reset	重置信号
stall_lu	是否发生 load-use
Branch_zero	是否进行分支跳转
Jump	是否进行跳转
Eret	是否从例外处理返回
PC_2	分支地址
PC_3	跳转地址
PC_4	例外返回地址
Int_mem	Mem 段是否存在中断
Int_wr	Wr 段是否存在中断
Exc_id	ID 段是否存在例外
Exc_ex	EX 段是否存在例外
Exc_mem	Mem 段是否存在例外
Exc_wr	Wr 段是否存在例外
输出信号名称	输出信号解释
Exc_out	该指令 IF 段是否产生例外
BadVAddr_out	发生地址错例外的虚地址
Status_out	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_out	描述最近一次例外的原因
PC_1	PC+4
PC	PC

（三）ID 段设计

ID 段包括 Trans.v 和 Registers.v。Trans.v 的功能部件有译码器 Decode、Load_use 检测

单元和 Exc_deal。Registers.v 的功能部件有通用寄存器组 Registers、HI/LO 寄存器、控制寄存器 CP0 和 Choose_data。Decode 用于指令译码；Load_use 检测单元用于检测是否发生了 Load_use 冒险；Exc_deal 用于在发生例外时产生例外信息；Choose_data 用于选择送往 ALU 的值。ID 段执行的结果被送到 ID/EX 寄存器保存，用于将数据和控制信号向下一段传递。

Registers 和 HI/LO 于 ID 段读数，Wr 段写数；CP0 于 ID 段读数，Mem 段写数。

ID 段的结构如图 3 所示。

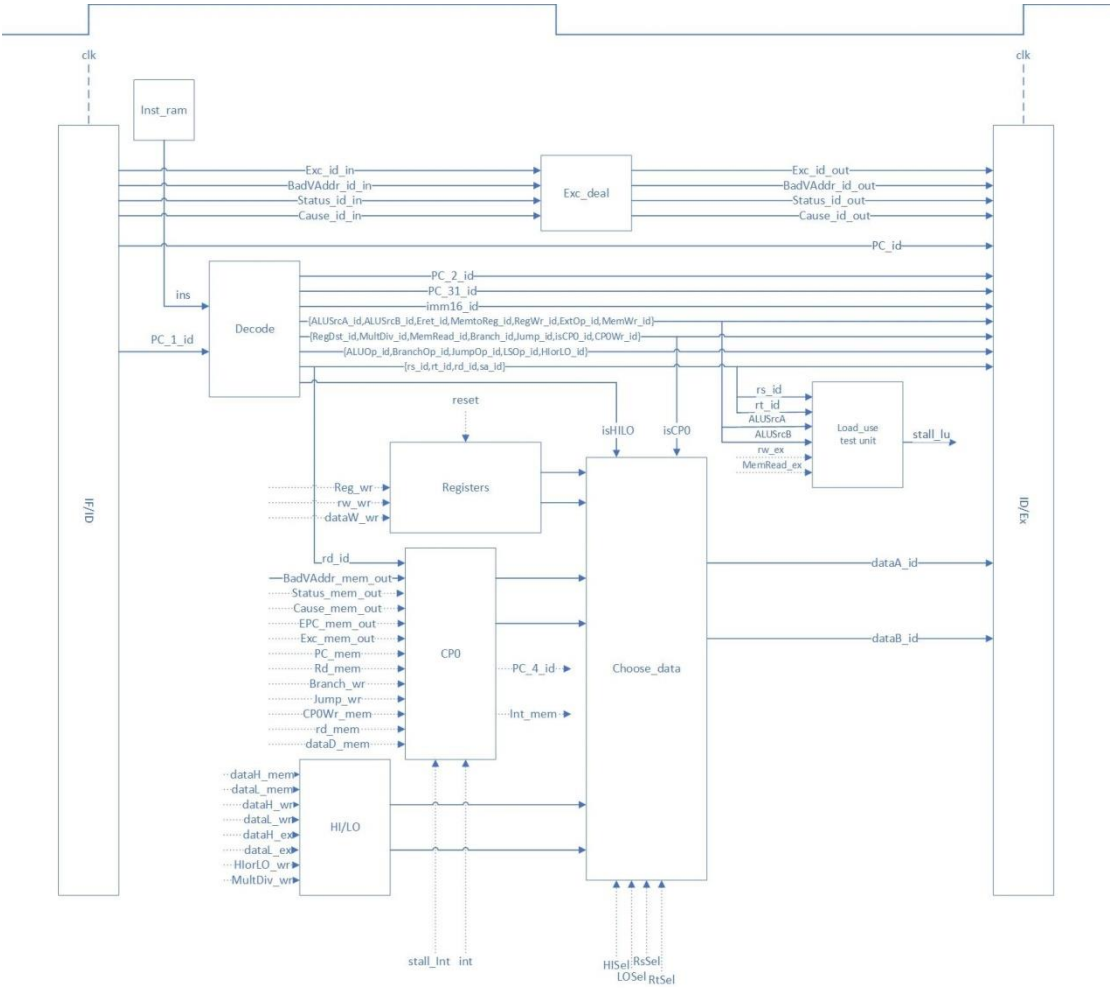


图 3. ID 段的结构示意图

表 2 列出了 ID 段的输入、输出信号。

表 2. ID 段输入输出信号

输入信号名称	输入信号解释
Trans.v	
ins	指令
MemRead_ex	EX 段是否为 Load 指令

rw_ex	EX 段目的寄存器
PC_1	PC+4
Exc_in	该指令于 IF 段是否产生例外
Exc_ex	EX 段是否存在例外
Exc_mem	Mem 段是否存在例外
BadVAddr_in	发生地址错例外的虚地址
Status_in	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_in	描述最近一次例外的原因
Registers.v	
clk	时钟信号
reset	重置信号
int	硬中断信号
Branch_wr	Wr 段是否为分支指令
Jump_wr	Wr 段是否为跳转指令
stall_Int	是否一直持续存在中断
RegWr	是否写通用寄存器
isHILO	是否读 HI/LO 寄存器
isCP0	是否读 CP0 寄存器
CP0Wr	是否写 CP0 寄存器
MultDiv	是否写 HI/LO 寄存器
HISeI	读 HI 寄存器是否存在转发
LOSeI	读 LO 寄存器是否存在转发
HIorLO	是否对 HI 及 LO 进行处理
RsSeI	读 rs 是否存在三阶转发
RtSeI	读 rt 是否存在三阶转发
rs_id	ID 段 rs
rt_id	ID 段 rt
rd_id	ID 段 rd
rd_mem	Mem 段 rd

rw	Wr 段 rw
dataH_ex	EX 段 dataH
dataL_ex	EX 段 dataL
dataH_mem	Mem 段 dataH
dataL_mem	Mem 段 dataL
dataH_wr	Wr 段 dataH
dataL_wr	Wr 段 dataL
PC	Mem 段 PC
dataW_wr	Wr 段写入通用寄存器的值
dataD	Mem 段备选运算端口 B 的值（满足转发）
Exc_mem	Mem 段是否存在例外
BadVAddr_mem	发生地址错例外的虚地址
Status_mem	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_mem	描述最近一次例外的原因
EPC_mem	例外处理完成后继续开始执行的指令的 PC
输出信号名称	输出信号解释
Trans.v	
Exc_out	该指令 ID 段及之前是否产生例外
BadVAddr_out	发生地址错例外的虚地址
Status_out	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_out	描述最近一次例外的原因
Branch	是否为分支指令
Jump	是否为跳转指令
RegDst	选择目的寄存器，1-rd，0-rt
ALUSrcA	选择运算端口 A 的值，1-sa，0-rs
ALUSrcB	选择运算端口 B 的值，1-imm16，0-rs
isHILO	是否读 HI/LO
MemtoReg	是否将从 data_ram 读取的数存入通用寄存器
MultDiv	是否写 HI/LO

RegWr	是否写通用寄存器
ExtOp	选择 imm16 扩展方式, 1-符号扩展, 2-零扩展
MemWr	是否写 data_ram
MemRead	是否为 Load 指令
Eret	是否为 Eret 指令
isCP0	是否读 CP0
CP0Wr	是否写 CP0
ALUOp	EX 段运算类型
BranchOp	分支类型
JumpOp	跳转类型
HIorLO	是否对 HI 及 LO 进行处理
LSOp	Load/Store 类型
rs	ID 段 rs
rt	ID 段 rt
rd	ID 段 rd
sa	ID 段 sa
imm16	ID 段 imm16
PC_2	分支地址
PC_31	跳转地址 1, {PC_1[31:28],ins[25:0],2'b00}
stall_lu	是否发生 load-use 冒险
Registers.v	
dataA	备选 EX 段运算端口 A 的值 (满足三阶转发)
dataB	备选 EX 段运算端口 B 的值 (满足三阶转发)
PC_4	例外返回地址
Int_mem	Mem 段是否产生中断

(四) EX 段设计

EX 段包括 Exec.v, 功能部件有 Get_data、MultDiv、ALU、Get_results、Branch_deal 和 Exc_deal。Get_data 用于选择参与运算的数值; MultDiv 用于计算送往 HI/LO 的数值; ALU

用于计算结果; Get_results 用于选择结果; Branch_deal 用于计算是否进行分支跳转; Exc_deal 用于在发生例外时产生例外信息。EX 段执行的结果被送到 EX/Mem 寄存器保存, 用于将数据和控制信号向下一段传递。

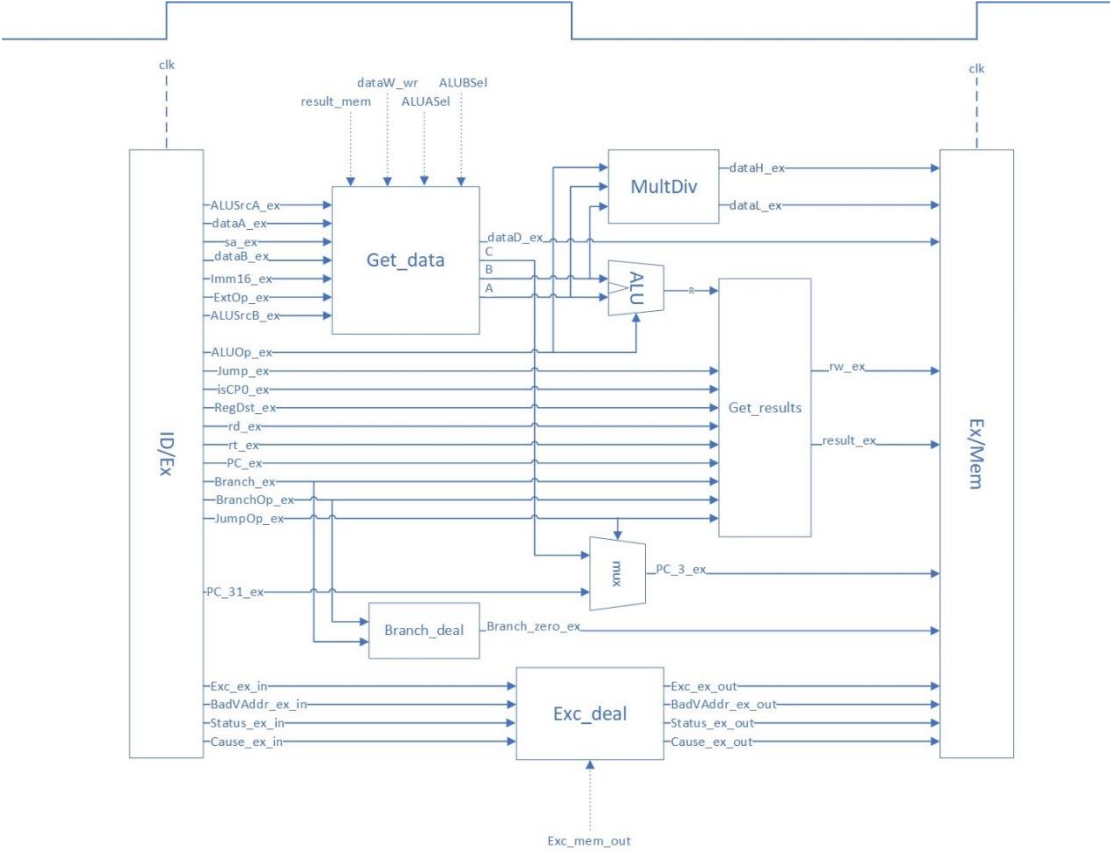


图 4. EX 模块的结构示意图

表 3 列出了 EX 段的输入、输出信号。

表 3. EX 段输入输出信号

输入信号名称	输入信号解释
PC	PC
PC_31	跳转地址 1, {PC_1[31:28],ins[25:0],2'b00}
dataA	备选运算端口 A 的值（满足三阶转发）
dataB	备选运算端口 B 的值（满足三阶转发）
result_mem	Mem 段运算结果值，用于一阶转发
dataW_wr	Wr 段写入通用寄存器的值，用于二阶转发
imm16	16 位立即数
Branch	是否为分支指令

Jump	是否为跳转指令
ExtOp	选择 imm16 扩展方式，1-符号扩展，0-零扩展
RegDst	选择目标寄存器，1-rd，0-rt
isCP0	是否读 CP0
ALUSrcA	选择运算端口 A 的值，1-sa，0-rs
ALUSrcB	选择运算端口 B 的值，1-imm16，0-rt
ALUASel	运算端口 A 是否存在转发
ALUBSel	运算端口 B 是否存在转发
ALUOp	运算类型
BranchOp	分支类型
JumpOp	跳转类型
rd	rd
rt	rt
sa	sa
Exc_in	该指令于 ID 段及之前是否产生例外
Exc_mem	Mem 段是否存在例外
BadVAddr_in	发生地址错例外的虚地址
Status_in	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_in	描述最近一次例外的原因
输出信号名称	输出信号解释
Exc_out	该指令 EX 段及之前是否产生例外
BadVAddr_out	发生地址错例外的虚地址
Status_out	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_out	描述最近一次例外的原因
Branch_zero	Mem 段产生的 Branch 信号与 Zero 标志得到，控制多路选择器的控制端
rw	目的寄存器编号
result	运算结果
dataH	存入 HI 的值
dataL	存入 LO 的值

D	备选运算端口 B 的值（满足转发）
PC_3	跳转地址

（五）Mem 段设计

Mem 段包括 Mem.v，功能部件有 data_deal、relay 和 Exc_deal。data_deal 用于得到数据存储器的地址、输入值和写使能信号；relay 用于计算是否需要转发；Exc_deal 用于在发生例外时产生例外信息。Mem 段执行的结果被送到 Mem/Wr 寄存器保存，用于将数据和控制信号向下一段传递。

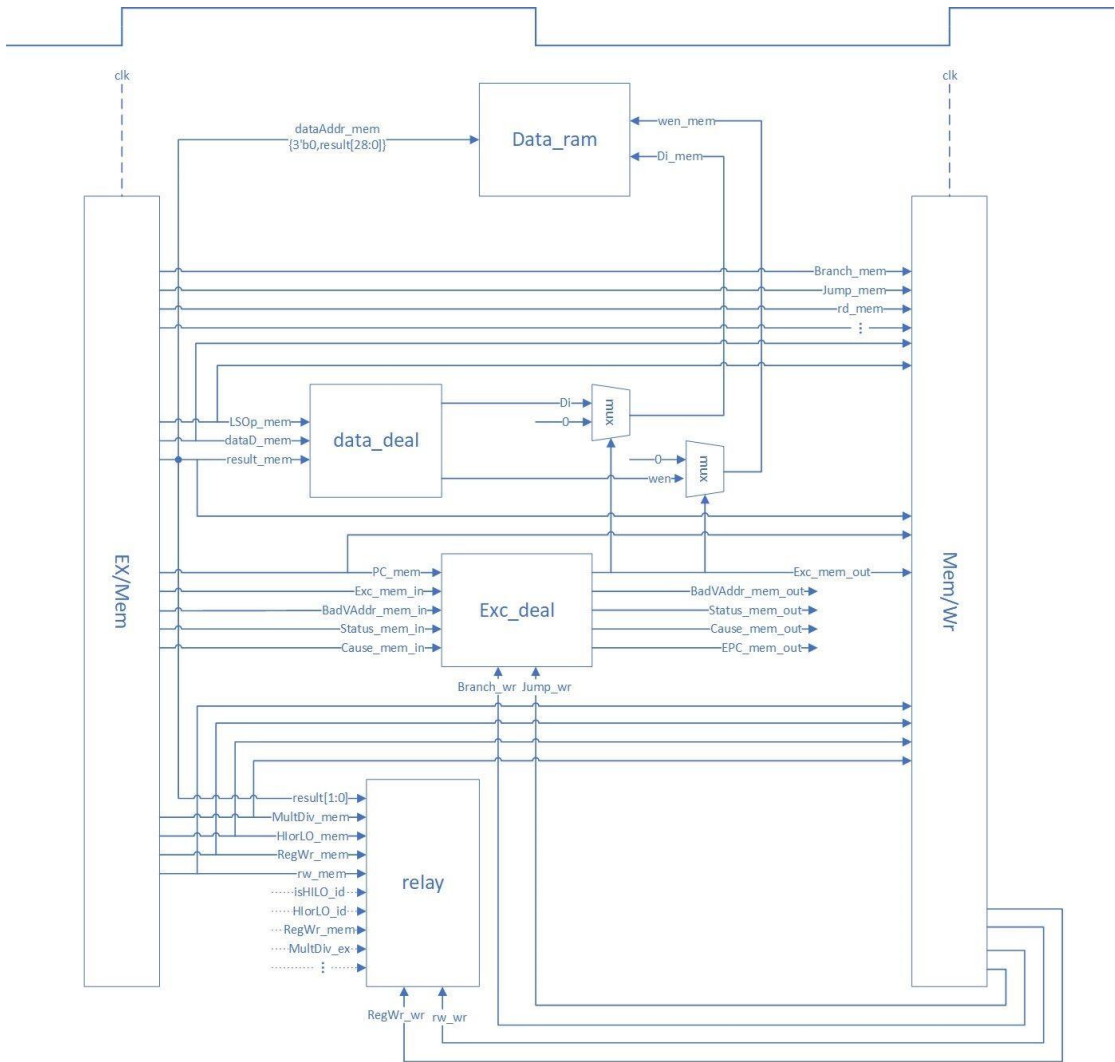


图 5. Mem 段的结构示意图

表 4 列出了 Mem 段的输入、输出信号。

表 4. Mem 段输入输出信号

输入信号名称	输入信号解释
--------	--------

PC	PC
Branch_wr	Wr 段是否为分支指令
Jump_wr	Wr 段是否为跳转指令
MultDiv_ex	EX 段是否写 HI/LO 寄存器
MultDiv_mem	Mem 段是否写 HI/LO 寄存器
MultDiv_wr	Wr 段是否写 HI/LO 寄存器
isHILO_id	ID 段是否读 HI/LO 寄存器
RegWr_mem	Mem 段是否写通用寄存器
RegWr_wr	Wr 段是否写通用寄存器
HIorLO_id	ID 段是否对 HI 及 LO 进行处理
HIorLO_ex	EX 段是否对 HI 及 LO 进行处理
HIorLO_mem	Mem 段是否对 HI 及 LO 进行处理
HIorLO_wr	Wr 段是否对 HI 及 LO 进行处理
LSOp_mem	Load/Store 类型
rs_id	ID 段 rs
rt_id	ID 段 rt
rs_ex	EX 段 rs
rt_ex	EX 段 rt
rw_mem	Mem 段目标寄存器编号
rw_wr	Wr 段目标寄存器编号
result	运算结果值
dataD	备选运算端口 B 的值（满足转发）
Exc_in	该指令于 EX 段及之前是否产生例外
BadVAddr_in	发生地址错例外的虚地址
Status_in	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_in	描述最近一次例外的原因
输出信号名称	输出信号解释
Exc_out	Mem 段及之前是否产生例外
BadVAddr_out	发生地址错例外的虚地址

Status_out	处理器操作模式、中断使能以及处理器状态诊断信息
Cause_out	描述最近一次例外的原因
EPC_out	例外处理完成后继续开始执行的指令的 PC
RsSel	读 rs 是否存在三阶转发
RtSel	读 rt 是否存在三阶转发
ALUASel	运算端口 A 是否存在转发
ALUBSel	运算端口 B 是否存在转发
HI Sel	读 HI 寄存器是否存在转发
LO Sel	读 LO 寄存器是否存在转发
wen	data_ram 写使能
Di	data_ram 输入值
dataAddr	data_ram 地址

（六）Wr 段设计

Wr 段包括 Write.v，功能部件有 Get_data 和 mux2to1。Get_data 用于处理从数据存储器得到的输出值；mux2to1 用于选择 data_ram 的输出值运算结果值作为通用寄存器存储值。

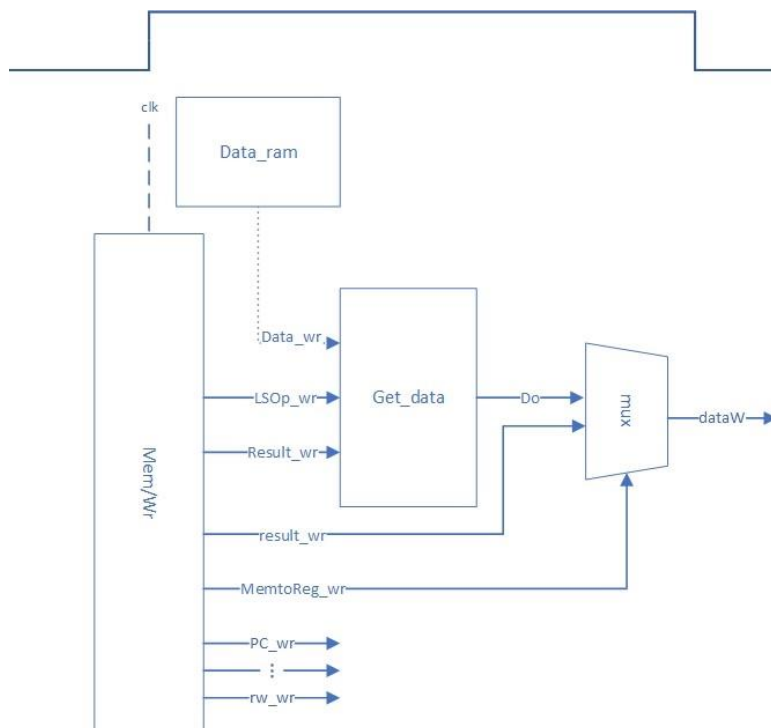


图 6. Wr 段的结构示意图

表 5 列出了 Wr 段的输入、输出信号。

表 5. Mem 段输入输出信号

输入信号名称	输入信号解释
MemtoReg	寄存器写入源的控制信号；1-源为 DM 输出，0-ALU 输出
LSOp_wr	Load/Store 类型
result	运算结果值
data	data_ram 输出值
Addr_wr	data_ram 地址
输出信号名称	输出信号解释
dataW	要写入目的寄存器的值

三、设计结果

（一）设计交付物说明

+--soc_sram_func/ : SRAM 接口的 CPU 运行环境，架构为 SoC_SRAM_Lite

+--rtl/

+--BRIDGE/

|--CONFREG/

|--myCPU/ :自己设计的 myCPU 文件

+--ALU.v

|--CPU.v :顶层文件

|--Exec.v

|--Extender.v

|--Ifetch.v

|--Mem.v

|--MultDiv.v

|--mux2to1.v

|--mux3to1.v

|--mux4to1.v

|--mux5to1.v

|--Registers.v

```

|--Trans.v

|--Write.v

|--xilinx_ip/

|--soc_lite_top.v

|--run vivado/

+--mycpu_prj1/

+--mycpu.xpr

|--func.bit      : mycpu 工程生成的 bit 流文件

|--soc_lite.xdc

|--testbench/

|--soft/          : 89 个功能点测试程序和记忆游戏测试程序，默认已包含编译好的结果

|--design.pdf      : 本文档

|--score.xls      : 分数登记表

```

myCPU 仿真：

- (1) 满足 mycpu 工程可以找到所需 inst_ram 文件及 golden_trace 文件；
- (2) 进入 soc_sram_func/run_vivado/mycpu/，打开 Vivado 工程 mycpu，将自己设计的 myCPU 文件添加到工程中，进行仿真，观察 Tcl Console 中输出结果。

myCPU 上板：

- (1) 降低 clk_pll 模块的输出端 cpu_clk 频率，将其修改为 10MHz；
- (2) 对 mycpu 工程进行综合处理并生成 bit 流；
- (3) 进入 Hardware Manager。连接 FPGA 开发板，加载 bit 流文件，进行上板验证，观察实验箱上数码管显示结果，判断是否正确。

(二) 设计演示结果

myCPU 进行功能仿真后通过 89 个功能点的结果如图 7 所示：

```

Test end!
——PASS!!!
$finish called at time : 15629995 ns : File "F:/1152833538/FileRecv/NDU_1_zhangxinyan/soc_sram_func/testbench/mycpu_tb.v" Line 261
run: Time (s): cpu = 00:01:17 ; elapsed = 00:06:22 . Memory (MB): peak = 3141.340 ; gain = 4.316

```

图 7.功能仿真结果

myCPU 上板后通过 89 个功能点最终显示的数码管结果（十六进制表示）如图 8 所示：



图 8.上板结果

由于降低了 clk_pll 模块的输出端频率，运行时数码管会出现数字闪烁现象，照片拍摄会出现有部分数码管未亮起现象。

WNS 值不为负，如图 9 所示：

Name	Constraints	WNS	Status
✓ synth_1 (active)	constrs_1		synth_design Complete!
✓ impl_1	constrs_1	2.934	write_bitstream Complete!

图 9. 最差负时序裕量

四、参考设计说明

主要参考资料为袁春风主编的《计算机组成与系统结构（第 2 版）》中的第六章——指令流水线。该书为我校教材，对指令流水线和五段流水线处理器做了初步讲解且书中无直接大段代码，我队仅结合书本理论知识、龙芯发布的资料包和 CSDN 网站上资料完成该设计。

- （1）模块划分参考书本 P196-197 中指令流水段分析。
- （2）数据通路参考书本 P198 中五段流水线数据通路基本框架。
- （3）控制信号和流水段功能参考书本 P198-P203 中对 IF 段、ID 段、Ex 段、Mem 段、Wr 段流水段功能的介绍。

（4）为解决数据冒险引入的转发技术和 Load-use 数据冒险的检测和处理参考书本 P207-P209 中对转发技术和 Load-use 数据冒险的概述，以及 P212 带转发和 Load-use 冒险处

理的部分流水线数据通路。

(5) 中断和例外处理参考书本 P216-P217 中异常或中断引起的控制冒险概述。

五、参考文献

[1]袁春风.计算机组成与系统结构(第 2 版)[M].北京:清华大学出版社,2015(2017.12 重印).