# 1. 实验目的与背景

## 1.1 课程目标与实验意义

本次实验的内容是设计并实现一个简易的五级流水线CPU，并通过64个测试点，主要目的是让我们更加深入地理解流水线技术及其在提升CPU性能中的应用。同时学习并使用Verilog语言，掌握如何实现指令的逐级处理。本次实验将不止能增强我们对计算机体系结构的理解，同时还提高了硬件设计和调试能力，为进一步学习复杂的计算机系统设计打下了基础。

## 1.2 实验相关技术背景

流水线技术是现代CPU设计中的核心优化方法，它将指令执行过程分解为多个阶段进行，分别是取指、解码、执行、内存访问、写回，好处是使得多个指令能够并行处理，从而提高CPU的吞吐量。五级流水线结构是经典的处理器设计，通过将复杂操作分解为简单阶段，有效减少了处理时间。流水线设计能够显著提升性能。

# 2. 实验任务与分工

## 2.1 实验任务概述

本实验中要完成CPU流水线的测试，需要深入理解计算机系统的硬件运行机制，掌握流水线技术在提高 CPU 性能方面的原理和应用。要构建构建一个包含五级流水线，需要使用verilog完成取指 IF、译码 ID、执行 EX、访存 MEM、写回 WB的 CPU 模型等代码，以实现了一个具备基本功能的 CPU 流水线系统。

## 2.2 小组成员分工详情

|  |  |  |
| --- | --- | --- |
| 姓名 | 任务分工 | 任务量占比 |
| 王新宇 | 实现IF，ID，MEM，WBHILO，CTRL等指令以及主要代码编写，参加实验报告编写 | 70% |
| 刘力瑞 | 主要负责在流水线中的EX阶段编写，参与实验报告的编写 | 30% |

# 3. 实验设计与实现

## 3.1 CPU 总体架构设计

CPU总体围绕一条五级流水线展开，包含多个功能模块，各模块协同工作实现指令的取指、译码、执行、访存和写回等操作，同时具备对流水线的控制以及特殊寄存器的处理功能，具体架构可以分为以下部分：

1. 流水线模块：包括IF（取指）、ID（译码）、EX（执行）、MEM（访存）、WB（写回）模块，构建了CPU指令执行的五级流水线基本框架，负责指令在不同阶段的处理和流转。

2.运算与寄存器模块：ALU相关位于/lib目录下的alu.v等文件构建了ALU，用于执行算术和逻辑运算，为EX模块提供运算支持。

3.寄存器相关：regfile.v构建了通用寄存器，hi\_lo\_reg.v实现了hilo寄存器（用于乘法和除法结果的存储等），这些寄存器在指令执行过程中用于数据的存储和传递。

4.控制模块：CTRL模块接收各阶段的请求信号，控制流水线各阶段的暂停和运行，协调各模块工作节奏。

5.数据传输总线：各模块之间通过特定宽度的数据总线连接，如IF模块到ID模块通过if\_to\_id\_bus（33位）传输数据，ID模块到EX模块通过id\_to\_ex\_bus（159位）传输数据等，确保数据在不同阶段的准确传递。

6.指令处理流程：

取指（IF）：从指令内存中获取指令，根据时钟、复位、暂停和跳转等信号控制指令地址的生成和指令的读取，将读取的指令传递给ID模块。

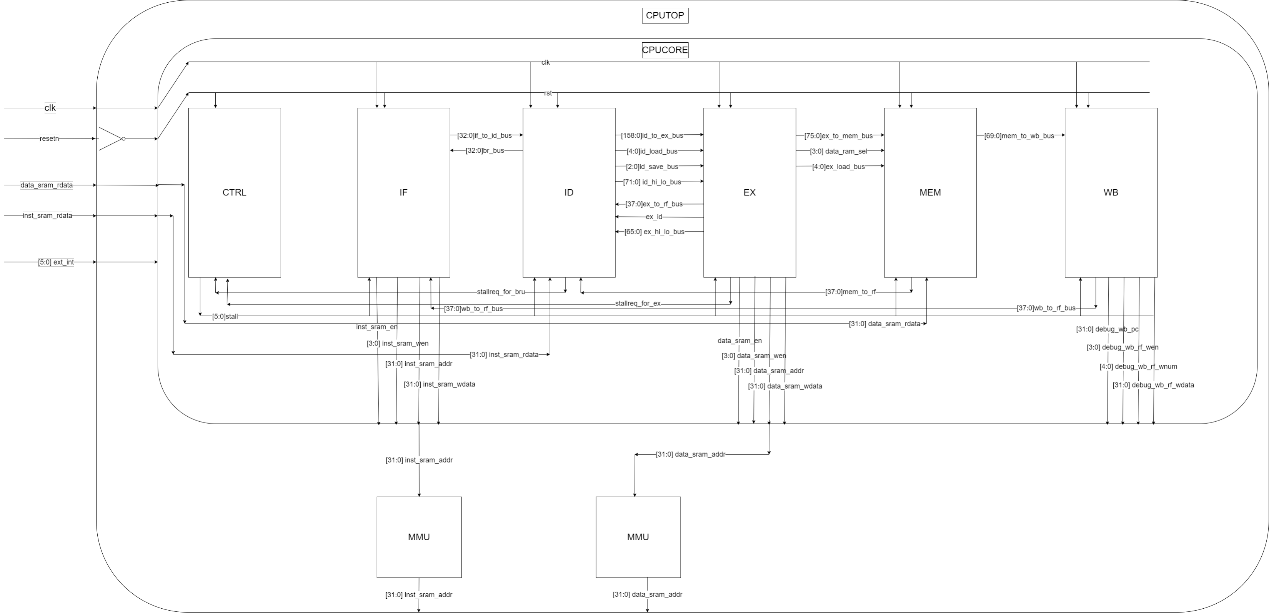
译码（ID）：对指令进行译码，判断指令类型（一般指令或跳转指令等），读取通用寄存器的值，处理数据相关，确定操作数来源，将译码结果和相关数据传递给EX模块，同时处理流水线暂停请求。

执行（EX）：执行运算操作，包括算术、逻辑运算，计算地址，处理访存请求，将运算结果和访存相关信息传递给MEM模块，并向ID模块反馈部分数据。

访存（MEM）：根据EX模块传来的地址进行内存访问操作，处理load和store指令，将访存结果传递给WB模块。

写回（WB）：将结果写回寄存器堆，完成指令执行的最后一步。

4.特殊寄存器处理：hilo寄存器用于乘法和除法结果的高位（hi）和低位（lo）存储，其读写操作有别于通用寄存器，在执行阶段读取（mfhi、mflo），写回阶段写入（mthi、mtlo），通过特定的控制信号实现数据的正确读写。



## 3.2 代码文件概述

项目包含多个模块文件，共同构建了一条流水线的基本框架以及相关运算和寄存器等功能。包括 IF.v、ID.v、EX.v、MEM.v、WB.v、hi\_lo\_reg.v、mycpu\_core.v、mycpu\_top.v 等文件搭建流水线框架；位于 /lib 目录下的 alu.v、decoder\_2\_4.v、decoder\_5\_32.v、decoder\_6\_64.v、defines.vh、div.v、mmu.v、regfile.v 构建 ALU 和寄存器并定义头文件；位于 /lib/mul 目录下的 add.v、fa.v、mul.v 实现乘法运算。

## 3.3 流水线各阶段详细设计

### 3.3.1 IF（取指令）阶段

### 3.3.2 ID（指令译码）阶段

### 3.3.3 EX（执行）阶段

输入端口：

|  |  |  |
| --- | --- | --- |
| 接口名 | 宽度 | 数据 |
| clk | 1 | 时钟信号 |
| rst | 1 | 复位信号 |
| stall | 6 | 控制暂停信号 |
| id\_to\_ex\_bus | 169 | ID 段传给 EX 段的数据 |
| id\_load\_bus | 5 | ID段传递读的数据 |
| id\_save\_bus | 3 | ID段传递写的数据 |
| id\_hi\_lo\_bus | 72 | ID 段传给hilo段的数据 |

### 3.3.4 MEM（访存）阶段

### 3.3.5 WB（写回）阶段

## 3.4 关键模块实现细节

### 3.4.1 HILO 寄存器模块

### 3.4.2 CTRL 模块

# 4. 实验测试与结果分析

## 4.1 测试环境与方法

测试平台：装有Vivado的windows系统。

使用VSCode编写代码，提供代码编辑功能。

利用Vivado进行模拟仿真，用于验证设计的正确性。

借助git进行版本管理，便于代码的更新、回溯和团队协作。

通过GitHub搭建项目仓库，实现代码的存储、共享和团队间的协同开发。

测试方法

debug

在实验的debug过程中，通过在波形图中添加可能有问题的数值，查看提示pc值附近目标的波形图来定位出错的位置及原因。通过观察波形图中信号的变化、数据的传输以及各模块之间的交互情况，分析指令执行过程中是否存在异常，如数据错误、信号时序问题等，从而找出导致错误的具体指令或模块操作，以便进行针对性的修改和优化。

2. \*\*功能测试（推测）\*\*

- 可能会编写一系列测试程序，涵盖各种指令类型（如算术运算、逻辑运算、数据移动、跳转、访存等指令）以及不同的操作数组合和边界情况，将这些测试程序加载到设计的CPU中运行。

- 观察CPU的执行结果，包括寄存器值的变化、内存数据的读写正确性、程序的执行流程是否符合预期等方面，以此来验证CPU在各种情况下的功能正确性，确保各个模块能够协同工作，正确执行指令集的各项功能，满足设计要求。例如，对于算术运算指令，检查计算结果是否准确；对于跳转指令，验证程序是否能正确跳转到指定地址；对于访存指令，确认数据的读写操作是否正确无误等。

## 4.2 测试结果展示

Test begin!

----[ 14025 ns] Number 8'd01 Functional Test Point PASS!!!

[ 22000 ns] Test is running, debug\_wb\_pc = 0xbfc5e4d4

[ 32000 ns] Test is running, debug\_wb\_pc = 0xbfc5f474

----[ 40475 ns] Number 8'd02 Functional Test Point PASS!!!

[ 42000 ns] Test is running, debug\_wb\_pc = 0xbfc89440

----[ 49355 ns] Number 8'd03 Functional Test Point PASS!!!

[ 52000 ns] Test is running, debug\_wb\_pc = 0xbfc3ad58

[ 62000 ns] Test is running, debug\_wb\_pc = 0xbfc3c260

----[ 71115 ns] Number 8'd04 Functional Test Point PASS!!!

[ 72000 ns] Test is running, debug\_wb\_pc = 0xbfc23898

[ 82000 ns] Test is running, debug\_wb\_pc = 0xbfc24c40

[ 92000 ns] Test is running, debug\_wb\_pc = 0xbfc2621c

[ 102000 ns] Test is running, debug\_wb\_pc = 0xbfc2776c

----[ 104845 ns] Number 8'd05 Functional Test Point PASS!!!

[ 112000 ns] Test is running, debug\_wb\_pc = 0xbfc4a0ac

----[ 117885 ns] Number 8'd06 Functional Test Point PASS!!!

[ 122000 ns] Test is running, debug\_wb\_pc = 0xbfc6a68c

[ 132000 ns] Test is running, debug\_wb\_pc = 0xbfc6b62c

[ 142000 ns] Test is running, debug\_wb\_pc = 0xbfc6c5cc

----[ 144265 ns] Number 8'd07 Functional Test Point PASS!!!

[ 152000 ns] Test is running, debug\_wb\_pc = 0xbfc509e4

[ 162000 ns] Test is running, debug\_wb\_pc = 0xbfc51984

----[ 167675 ns] Number 8'd08 Functional Test Point PASS!!!

[ 172000 ns] Test is running, debug\_wb\_pc = 0xbfc03bb0

[ 182000 ns] Test is running, debug\_wb\_pc = 0xbfc04b50

----[ 185575 ns] Number 8'd09 Functional Test Point PASS!!!

[ 192000 ns] Test is running, debug\_wb\_pc = 0xbfc3e008

[ 202000 ns] Test is running, debug\_wb\_pc = 0xbfc3efa8

----[ 203475 ns] Number 8'd10 Functional Test Point PASS!!!

[ 212000 ns] Test is running, debug\_wb\_pc = 0xbfc6f800

[ 222000 ns] Test is running, debug\_wb\_pc = 0xbfc707a0

----[ 222735 ns] Number 8'd11 Functional Test Point PASS!!!

[ 232000 ns] Test is running, debug\_wb\_pc = 0xbfc02648

----[ 237365 ns] Number 8'd12 Functional Test Point PASS!!!

[ 242000 ns] Test is running, debug\_wb\_pc = 0xbfc3faec

[ 252000 ns] Test is running, debug\_wb\_pc = 0xbfc40f3c

----[ 261915 ns] Number 8'd13 Functional Test Point PASS!!!

[ 262000 ns] Test is running, debug\_wb\_pc = 0xbfc00d58

[ 272000 ns] Test is running, debug\_wb\_pc = 0xbfc64ab8

[ 282000 ns] Test is running, debug\_wb\_pc = 0xbfc65db8

[ 292000 ns] Test is running, debug\_wb\_pc = 0xbfc67094

----[ 296425 ns] Number 8'd14 Functional Test Point PASS!!!

[ 302000 ns] Test is running, debug\_wb\_pc = 0xbfc84484

[ 312000 ns] Test is running, debug\_wb\_pc = 0xbfc85814

[ 322000 ns] Test is running, debug\_wb\_pc = 0xbfc86b8c

----[ 330855 ns] Number 8'd15 Functional Test Point PASS!!!

[ 332000 ns] Test is running, debug\_wb\_pc = 0xbfc7a0f0

----[ 333825 ns] Number 8'd16 Functional Test Point PASS!!!

----[ 336795 ns] Number 8'd17 Functional Test Point PASS!!!

----[ 338455 ns] Number 8'd18 Functional Test Point PASS!!!

----[ 340625 ns] Number 8'd19 Functional Test Point PASS!!!

[ 342000 ns] Test is running, debug\_wb\_pc = 0xbfc87f18

----[ 342805 ns] Number 8'd20 Functional Test Point PASS!!!

[ 352000 ns] Test is running, debug\_wb\_pc = 0xbfc8006c

[ 362000 ns] Test is running, debug\_wb\_pc = 0xbfc8100c

----[ 364615 ns] Number 8'd21 Functional Test Point PASS!!!

[ 372000 ns] Test is running, debug\_wb\_pc = 0xbfc0b1c8

[ 382000 ns] Test is running, debug\_wb\_pc = 0xbfc0c168

----[ 385015 ns] Number 8'd22 Functional Test Point PASS!!!

[ 392000 ns] Test is running, debug\_wb\_pc = 0xbfc332a8

[ 402000 ns] Test is running, debug\_wb\_pc = 0xbfc34248

----[ 406375 ns] Number 8'd23 Functional Test Point PASS!!!

[ 412000 ns] Test is running, debug\_wb\_pc = 0xbfc61798

[ 422000 ns] Test is running, debug\_wb\_pc = 0xbfc62738

[ 432000 ns] Test is running, debug\_wb\_pc = 0xbfc636d8

----[ 432785 ns] Number 8'd24 Functional Test Point PASS!!!

[ 442000 ns] Test is running, debug\_wb\_pc = 0xbfc7b564

[ 452000 ns] Test is running, debug\_wb\_pc = 0xbfc7c504

----[ 456195 ns] Number 8'd25 Functional Test Point PASS!!!

[ 462000 ns] Test is running, debug\_wb\_pc = 0xbfc4d470

[ 472000 ns] Test is running, debug\_wb\_pc = 0xbfc4e410

----[ 481535 ns] Number 8'd26 Functional Test Point PASS!!!

[ 482000 ns] Test is running, debug\_wb\_pc = 0xbfc6cfc8

[ 492000 ns] Test is running, debug\_wb\_pc = 0xbfc6df68

----[ 499435 ns] Number 8'd27 Functional Test Point PASS!!!

[ 502000 ns] Test is running, debug\_wb\_pc = 0xbfc8a360

[ 512000 ns] Test is running, debug\_wb\_pc = 0xbfc8b300

[ 522000 ns] Test is running, debug\_wb\_pc = 0xbfc8c2a0

----[ 525825 ns] Number 8'd28 Functional Test Point PASS!!!

[ 532000 ns] Test is running, debug\_wb\_pc = 0xbfc78914

[ 542000 ns] Test is running, debug\_wb\_pc = 0xbfc798b4

----[ 546225 ns] Number 8'd29 Functional Test Point PASS!!!

[ 552000 ns] Test is running, debug\_wb\_pc = 0xbfc475a4

[ 562000 ns] Test is running, debug\_wb\_pc = 0xbfc48544

[ 572000 ns] Test is running, debug\_wb\_pc = 0xbfc494e4

----[ 572635 ns] Number 8'd30 Functional Test Point PASS!!!

[ 582000 ns] Test is running, debug\_wb\_pc = 0xbfc09260

[ 592000 ns] Test is running, debug\_wb\_pc = 0xbfc0a200

----[ 593035 ns] Number 8'd31 Functional Test Point PASS!!!

[ 602000 ns] Test is running, debug\_wb\_pc = 0xbfc76b40

[ 612000 ns] Test is running, debug\_wb\_pc = 0xbfc77ae0

----[ 615165 ns] Number 8'd32 Functional Test Point PASS!!!

[ 622000 ns] Test is running, debug\_wb\_pc = 0xbfc42d9c

[ 632000 ns] Test is running, debug\_wb\_pc = 0xbfc43d3c

----[ 634365 ns] Number 8'd33 Functional Test Point PASS!!!

[ 642000 ns] Test is running, debug\_wb\_pc = 0xbfc0d4ac

[ 652000 ns] Test is running, debug\_wb\_pc = 0xbfc0e44c

----[ 656735 ns] Number 8'd34 Functional Test Point PASS!!!

[ 662000 ns] Test is running, debug\_wb\_pc = 0xbfc06b28

[ 672000 ns] Test is running, debug\_wb\_pc = 0xbfc07ac8

----[ 676065 ns] Number 8'd35 Functional Test Point PASS!!!

[ 682000 ns] Test is running, debug\_wb\_pc = 0xbfc5bc14

[ 692000 ns] Test is running, debug\_wb\_pc = 0xbfc5cbb4

----[ 698445 ns] Number 8'd36 Functional Test Point PASS!!!

[ 702000 ns] Test is running, debug\_wb\_pc = 0xbfc565fc

[ 712000 ns] Test is running, debug\_wb\_pc = 0xbfc57bfc

[ 722000 ns] Test is running, debug\_wb\_pc = 0xbfc59250

[ 732000 ns] Test is running, debug\_wb\_pc = 0xbfc5a8d4

----[ 736645 ns] Number 8'd37 Functional Test Point PASS!!!

[ 742000 ns] Test is running, debug\_wb\_pc = 0xbfc1eaec

[ 752000 ns] Test is running, debug\_wb\_pc = 0xbfc20170

[ 762000 ns] Test is running, debug\_wb\_pc = 0xbfc217e4

[ 772000 ns] Test is running, debug\_wb\_pc = 0xbfc22e94

----[ 774605 ns] Number 8'd38 Functional Test Point PASS!!!

[ 782000 ns] Test is running, debug\_wb\_pc = 0xbfc718dc

[ 792000 ns] Test is running, debug\_wb\_pc = 0xbfc72f24

[ 802000 ns] Test is running, debug\_wb\_pc = 0xbfc7456c

[ 812000 ns] Test is running, debug\_wb\_pc = 0xbfc75be4

----[ 812805 ns] Number 8'd39 Functional Test Point PASS!!!

[ 822000 ns] Test is running, debug\_wb\_pc = 0xbfc53414

[ 832000 ns] Test is running, debug\_wb\_pc = 0xbfc548ec

[ 842000 ns] Test is running, debug\_wb\_pc = 0xbfc55d4c

----[ 842485 ns] Number 8'd40 Functional Test Point PASS!!!

[ 852000 ns] Test is running, debug\_wb\_pc = 0xbfc29d0c

[ 862000 ns] Test is running, debug\_wb\_pc = 0xbfc2b0e0

[ 872000 ns] Test is running, debug\_wb\_pc = 0xbfc2c480

[ 882000 ns] Test is running, debug\_wb\_pc = 0xbfc2d884

----[ 886735 ns] Number 8'd41 Functional Test Point PASS!!!

[ 892000 ns] Test is running, debug\_wb\_pc = 0xbfc18e80

[ 902000 ns] Test is running, debug\_wb\_pc = 0xbfc1a2a0

[ 912000 ns] Test is running, debug\_wb\_pc = 0xbfc1b658

[ 922000 ns] Test is running, debug\_wb\_pc = 0xbfc1c9f4

[ 932000 ns] Test is running, debug\_wb\_pc = 0xbfc1ddcc

----[ 933015 ns] Number 8'd42 Functional Test Point PASS!!!

[ 942000 ns] Test is running, debug\_wb\_pc = 0xbfc121b8

[ 952000 ns] Test is running, debug\_wb\_pc = 0xbfc13318

[ 962000 ns] Test is running, debug\_wb\_pc = 0xbfc14478

[ 972000 ns] Test is running, debug\_wb\_pc = 0xbfc155d8

[ 982000 ns] Test is running, debug\_wb\_pc = 0xbfc16728

----[ 983765 ns] Number 8'd43 Functional Test Point PASS!!!

[ 992000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1002000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1012000 ns] Test is running, debug\_wb\_pc = 0xbfc7e440

[1022000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1032000 ns] Test is running, debug\_wb\_pc = 0xbfc7ed14

[1042000 ns] Test is running, debug\_wb\_pc = 0x00000000

----[1043835 ns] Number 8'd44 Functional Test Point PASS!!!

[1052000 ns] Test is running, debug\_wb\_pc = 0xbfc0ef50

[1062000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1072000 ns] Test is running, debug\_wb\_pc = 0xbfc0f824

[1082000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1092000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1102000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1112000 ns] Test is running, debug\_wb\_pc = 0x00000000

[1122000 ns] Test is running, debug\_wb\_pc = 0xbfc10e7c

----[1131265 ns] Number 8'd45 Functional Test Point PASS!!!

[1132000 ns] Test is running, debug\_wb\_pc = 0xbfc349e4

[1142000 ns] Test is running, debug\_wb\_pc = 0xbfc35984

[1152000 ns] Test is running, debug\_wb\_pc = 0xbfc36924

----[1154805 ns] Number 8'd46 Functional Test Point PASS!!!

[1162000 ns] Test is running, debug\_wb\_pc = 0xbfc8240c

[1172000 ns] Test is running, debug\_wb\_pc = 0xbfc833ac

----[1176105 ns] Number 8'd47 Functional Test Point PASS!!!

[1182000 ns] Test is running, debug\_wb\_pc = 0xbfc88c94

----[1185635 ns] Number 8'd48 Functional Test Point PASS!!!

[1192000 ns] Test is running, debug\_wb\_pc = 0xbfc17f80

----[1195145 ns] Number 8'd49 Functional Test Point PASS!!!

[1202000 ns] Test is running, debug\_wb\_pc = 0xbfc60c04

----[1204035 ns] Number 8'd50 Functional Test Point PASS!!!

----[1209165 ns] Number 8'd51 Functional Test Point PASS!!!

[1212000 ns] Test is running, debug\_wb\_pc = 0xbfc0337c

----[1215135 ns] Number 8'd52 Functional Test Point PASS!!!

----[1221745 ns] Number 8'd53 Functional Test Point PASS!!!

[1222000 ns] Test is running, debug\_wb\_pc = 0xbfc00d94

----[1228035 ns] Number 8'd54 Functional Test Point PASS!!!

[1232000 ns] Test is running, debug\_wb\_pc = 0x00000000

----[1234645 ns] Number 8'd55 Functional Test Point PASS!!!

----[1241885 ns] Number 8'd56 Functional Test Point PASS!!!

[1242000 ns] Test is running, debug\_wb\_pc = 0xbfc00d60

----[1248485 ns] Number 8'd57 Functional Test Point PASS!!!

[1252000 ns] Test is running, debug\_wb\_pc = 0xbfc4fa0c

----[1253335 ns] Number 8'd58 Functional Test Point PASS!!!

[1262000 ns] Test is running, debug\_wb\_pc = 0xbfc37e08

[1272000 ns] Test is running, debug\_wb\_pc = 0xbfc38d3c

----[1277255 ns] Number 8'd59 Functional Test Point PASS!!!

[1282000 ns] Test is running, debug\_wb\_pc = 0xbfc67fc4

[1292000 ns] Test is running, debug\_wb\_pc = 0xbfc68ee0

[1302000 ns] Test is running, debug\_wb\_pc = 0xbfc69e00

----[1302835 ns] Number 8'd60 Functional Test Point PASS!!!

[1312000 ns] Test is running, debug\_wb\_pc = 0xbfc2ef90

----[1321725 ns] Number 8'd61 Functional Test Point PASS!!!

[1322000 ns] Test is running, debug\_wb\_pc = 0xbfc00ae8

[1332000 ns] Test is running, debug\_wb\_pc = 0xbfc4b900

[1342000 ns] Test is running, debug\_wb\_pc = 0xbfc4c7e8

----[1342985 ns] Number 8'd62 Functional Test Point PASS!!!

[1352000 ns] Test is running, debug\_wb\_pc = 0xbfc44da4

[1362000 ns] Test is running, debug\_wb\_pc = 0xbfc45cb4

----[1371175 ns] Number 8'd63 Functional Test Point PASS!!!

[1372000 ns] Test is running, debug\_wb\_pc = 0xbfc2ff28

[1382000 ns] Test is running, debug\_wb\_pc = 0xbfc30dd0

[1392000 ns] Test is running, debug\_wb\_pc = 0xbfc31cf8

----[1397715 ns] Number 8'd64 Functional Test Point PASS!!!

--------------------------------------------------------------

[1398137 ns] Error!!!

reference: PC = 0xbfc00380, wb\_rf\_wnum = 0x1a, wb\_rf\_wdata = 0x00004000

mycpu : PC = 0xbfc6a074, wb\_rf\_wnum = 0x09, wb\_rf\_wdata = 0x41000000

--------------------------------------------------------------

--------------------------------------------------------------

[1398155 ns] Error( 0)!!! Occurred in number 8'd65 Functional Test Point!

## 4.3 结果分析与讨论

从日志输出中可以看到，在大多数测试点（如8'd01、8'd02、8’d60等），测试都顺利通过，功能测试点标记为 PASS，且没有错误信息。每个测试点后跟着的是 debug\_wb\_pc 的值，表示当前测试的程序计数器值。这些信息意味着在每个测试点，CPU的工作是正常的，且没有异常发生。且在调试过程中，debug\_wb\_pc 的值逐渐变化，符合预期。

# 5. 实验总结与展望

## 5.1 实验成果总结

## 5.2 实验不足与改进方向

## 5.3 未来展望