**实验3报告**

学号 2018K8009929021

姓名 袁欣怡

箱子号 10

一、实验任务（10%）

学习了解无阻塞五级流水线CPU设计思路，并调试有错误的CPU设计代码。本次实验包括两个子任务：

子任务一：阅读代码，画出流水线CPU的设计结果框图。

子任务二：结合仿真结果和上板测试，对代码进行调试，找出其中的7处错误。

二、实验设计（40%）

1. 总体设计思路

图示

描述已自动生成

图1 硬件结构设计图

代码设计中，主要包括了七个模块，包括ALU、寄存器堆、取指模块、译码模块、执行模块、访存模块、写回模块。该设计还使用了两个IP核，Inst\_RAM和Data\_RAM核。

每当时钟信号到来的时候，IF模块负责更新PC值，从inst\_sram中取指令，并把指令和当前PC值放到fs\_to\_ds\_bus总线上。同时，IF模块还会将fs\_to\_ds\_valid信号拉高，目的是告诉ID模块数据已经准备好，等到ID接收数据时即可完成数据传递。

ID模块会对接收到的指令进行译码，并判断指令的类型。如果有分支指令，则需要判断是否需要跳转，否则就将译码结果全部送到ds\_to\_es\_bus 总线上，并把ds\_to\_es\_valid拉高，告诉EXE模块数据已经准备好。此外，ID模块还需要接受来自WB模块的写回寄存器的数据，在译码的同时调用reg\_file模块，写回数据。

EXE模块接收到ID的数据后，根据指令调用ALU进行计算。同时EXE还需要判断该指令是否需要写内存（e.g. sw指令），并调整内存写使能信号data\_sram\_wen。EXE模块需要将计算结果等输送到es\_to\_ms\_bus总线。并把es\_to\_ms\_valid拉高，等到MEM模块接收数据，同时给data\_sram发送读内存的信号。

MEM模块，数据传入MEM模块，同时接受data\_sram传来的数据。MEM需要判断最终写入内存的数据是哪个，EXE模块从ALU中得到的结果还是data\_sram发来的数据。确定后MEM模块将这个数据传到ms\_to\_ws\_bus总线上，并把ms\_to\_ws\_valid拉高，等待WB模块读数据。

WB模块允许数据进入后，需要根据传进来的数据，判断是否要写寄存器，同时确认写寄存器的地址。WB模块需要将数据传送到ws\_to\_rf\_bus总线上，让ID模块接收。

1. 重要模块1：IF模块
2. 工作原理

从inst\_ram中读取指令，处理PC的值，并将指令传递给bus，下一周期再传递给ID模块。

1. 接口定义

表1 IF模块的接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 方向 | 位宽 | 功能描述 |
| ds\_allowin | IN | 1 | IDstage 流水线接收允许信号，高电平有效 |
| br\_bus | IN | 33 | 分支指令总线，包括是否有分支指令跳转信号(1 位)、分支指 令跳转目标(32 位) |
| fs\_to\_ds\_valid | OUT | 1 | IFstage 到 IDstage 是否有效，高电平有效 |
| fs\_to\_ds\_bus | OUT | 64 | IFstage 到 IDstage 总线，包含从 sram 取的指令(高 32 位)、以 及 IFstage 的当前 PC |
| inst\_sram\_en | OUT | 1 | 向 sram 中取指使能，高电平有效 |
| inst\_sram\_wen | OUT | 4 | 向 sram 中取指写使能 |
| inst\_sram\_addr | OUT | 32 | IF 过程向 sram 读写数据的地址 |
| inst\_sram\_wdata | OUT | 32 | IF 过程向 sram 写的数据 |
| inst\_sram\_rdata | IN | 32 | IF 过程读出的指令数据 |

1. 功能描述

从sram取出当前PC地址存储的指令。

1. 重要模块2：ID模块
2. 工作原理

对指令进行译码，并且判断是否需要跳转。同时接收WB模块的数据，传递给寄存器堆。

1. 接口定义

表2 ID模块的接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 方向 | 位宽 | 功能描述 |
| es\_allowin | IN | 1 | EXEstage 流水线接收允许信号，高电平有效 |
| ds\_allowin | OUT | 1 | IDstage 流水线接收允许信号，高电平有效 |
| fs\_to\_ds\_valid | IN | 1 | IFstage 到 IDstage 是否有效，高电平有效 |
| fs\_to\_ds\_bus | IN | 64 | IFstage 到 IDstage 总线，包含从 sram 取的指令(高 32 位)、以 及 IFstage 的当前 PC |
| ds\_to\_es\_valid | OUT | 1 | IDstage 到 EXEstage 方向有效信号，高电平有效 |
| ds\_to\_es\_bus | OUT | 136 | IDstage 到 EXEstage 数据总线，包含 alu 操作信号(12 位)、从 内存加载数信号(1 位)、操作数 1 的来源(2 位)、操作数 2 的来源(2 位)、寄存器堆写使能(1 位)、内存写使能(1 位)、寄存器写地址(5 位)、立即数(16 位)、rs 寄存器的值 (32 位)、rt 寄存器的值(32 位)、IDstage 的当前 PC(32 位) |
| br\_bus | OUT | 33 | 分支指令总线，包括是否有分支指令跳转信号(1 位)、分支指 令跳转目标(32 位) |
| ws\_to\_rf\_bus | IN | 38 | 写寄存器总线，包含寄存器写使能(1 位)、寄存器写地址(5 位)、寄存器写数据(32 位) |

1. 功能描述

对IF传来的指令进行译码，大致确定指令类型（是否需要加载，是否需要跳转等）。在译码的数据和控制信号在下一时钟周期传递给EXE模块，并接受WB模块返回的数据。

1. 重要模块 3 :EXE 模块
2. 工作原理

当 EXEstage 流水线接收允许信号为高电平且向 EXEstage 流水线方向有效(ds\_to\_es\_valid)时，调用 ALU 模 块进行计算。

1. 接口定义

表3 EXE模块的接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 方向 | 位宽 | 功能描述 |
| ms\_allowin | IN | 1 | MEMstage 流水线接收允许信号，高电平有效 |
| es\_allowin | OUT | 1 | EXEstage 流水线接收允许信号，高电平有效 |
| ds\_to\_es\_valid | IN | 1 | IDstage 到 EXEstage 方向有效信号，高电平有效 |
| ds\_to\_es\_bus | IN | 136 | IDstage 到 EXEstage 数据总线，包含 alu 操作信号(12 位)、从 内存加载数信号(1 位)、操作数 1 的来源(2 位)、操作数 2 的来源(2 位)、寄存器堆写使能(1 位)、内存写使能(1 位)、寄存器写地址(5 位)、立即数(16 位)、rs 寄存器的值 (32 位)、rt 寄存器的值(32 位)、IDstage 的当前 PC(32 位) |
| es\_to\_ms\_valid | OUT | 1 | EXEstage 到 MEMstage 是否有效，高电平有效 |
| es\_to\_ms\_bus | OUT | 71 | EXEstage 到 MEMstage 总线，包含是否从内存加载数至寄存器 信号(1 位)、寄存器写使能(1 位)、寄存器写地址(5 位)、ALU 计算结果(32 位)、EXEstage 的当前 PC(32 位) |
| data\_sram\_en | OUT | 1 | 向 sram 中读数据使能，高电平有效 |
| date\_sram\_wen | OUT | 4 | 向 sram 中写数据使能，高电平有效 |
| data\_sram\_addr | OUT | 32 | 向 sram 中读写数据的地址，地址为 ALU 计算的结果 |
| data\_sram\_rdata | OUT | 32 | 从 sram 中读取的数据 |

1. 功能描述

根据 ID模块得到的不同的指令译码执行不同的指令，调用ALU。需要注意的是 sw 指令也在该阶段完成。

1. 重要模块 4 设计:MEM 模块
2. 工作原理

当MEM模块的允许接收，且向MEM模块的流水线有效时，向WB模块输出写回阶段使用的数据和地址等。

1. 接口定义

表4 MEM模块的接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 方向 | 位宽 | 功能描述 |
| ws\_allowin | IN | 1 | MEMstage 流水线接收允许信号，高电平有效 |
| ms\_allowin | OUT | 1 | EXEstage 流水线接收允许信号，高电平有效 |
| es\_to\_ms\_valid | IN | 1 | EXEstage 到 MEMstage 是否有效，高电平有效 |
| es\_to\_ms\_bus | IN | 71 | EXEstage 到 MEMstage 总线，包含是否从内存加载数至寄存器 信号(1 位)、寄存器写使能(1 位)、寄存器写地址(5 位)、ALU 计算结果(32 位)、EXEstage 的当前 PC(32 位) |
| ms\_to\_ws\_valid | OUT | 1 | MEMstage 到 WBstage 方向有效信号，高电平有效 |
| ms\_to\_ws\_bus | OUT | 70 | MEMstage 到 WBstage 数据总线，包含寄存器写使能(1 位)、 寄存器写地址(5 位)、最终传输的数据(从 ALU 来的或从内 存中来的，取决于 es\_to\_ms\_bus 中的是否从内存加载数至寄存 器的信号，32 位)、MEMstage 的当前 PC(32 位) |
| data\_sram\_rdata | IN | 32 | 从 sram 中读取的数据 |

1. 功能描述

向 WB 模块传输可能要写回寄存器的数据，它可能是来自于内存，也可能来自于 EXE模块计算出的 alu\_result，这需要根据ms\_res\_from\_mem 信号来判断。

1. 重要模块 5 :WB 模块
2. 工作原理

当WBstage流水线接收允许信号为高电平且向WBstage流水线方向有效(ms\_to\_ws\_valid)时，向给定地址的寄存器写回数据。

1. 接口定义

表5 WB模块的接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 方向 | 位宽 | 功能描述 |
| ws\_allowin | OUT | 1 | MEMstage 流水线接收允许信号，高电平有效 |
| ms\_to\_ws\_valid | IN | 1 | MEMstage 到 WBstage 方向有效信号，高电平有效 |
| ms\_to\_ws\_bus | IN | 70 | MEMstage 到 WBstage 数据总线，包含寄存器写使能(1 位)、 寄存器写地址(5 位)、最终传输的数据(从 ALU 来的或从内 存中来的，取决于 es\_to\_ms\_bus 中的是否从内存加载数至寄存 器的信号，32 位)、MEMstage 的当前 PC(32 位) |
| ws\_to\_rf\_bus | OUT | 38 | 写寄存器总线，包含寄存器写使能(1 位)、寄存器写地址(5 位)、寄存器写数据(32 位) |
| debug\_wb\_pc | OUT | 32 | WBstage 的 PC 值，用于 debug |
| debug\_wb\_rf\_wen | OUT | 4 | WBstage 寄存器堆写使能，用于 debug |
| debug\_wb\_rf\_wnum | OUT | 5 | WBstage 写寄存器堆地址，用于 debug |
| debug\_wb\_rf\_wdata | OUT | 32 | WBstage 写寄存器堆数据，用于 debug |

1. 功能描述

向寄存器写回数据，数据来自MEM模块到WB模块的数据总线。

三、实验过程（50%）

1. 实验流水账

2020.9.27 16:00-18:00 阅读讲义，完成任务一

2020.9.27 18:30-22:30 进行任务二

2020.9.28 20:00-22:00 完成任务二和实验报告

1. 错误记录
2. 错误1：信号为X
3. 错误现象

信号ds\_valid为X，fs\_to\_ds\_valid等几个valid信号均为X。



图2 信号ds\_valid

1. 分析定位过程

由于几个valid信号均为X值，考虑到流水线层层递进的关系，确定为ds\_to\_es\_valid信号赋值时出现问题。在源码中ID模块中，从给ds\_to\_es\_valid的赋值语句中可以判定时ds\_valid的值出现问题，因此发现没有给它赋值。

1. 错误原因

变量ds\_valid没有赋值。

1. 修正效果

可以仿照其他模块给ds\_valid信号赋值。赋值成功后信号都恢复正常。

1. 错误2：信号为Z
2. 错误现象

信号load\_op没有赋初值，因此显示为Z。

1. 分析定位过程

发现ds\_to\_es\_bus中间一段值出错。翻看源码时，将鼠标放在变量上来查看变量的值，于是发现load\_op的值为Z。

表格

描述已自动生成

图3 ds\_to\_es\_bus赋值语句

1. 错误原因

变量load\_op没有赋值。

1. 修正效果

添加给load\_op赋值的语句assign load\_op = inst\_lw。修改之后load\_op恢复正常。

1. 错误3:信号为Z
2. 错误现象

变量op\_d和func\_d的值最高位均为Z。

1. 分析定位过程

这两个信号由decoder\_6\_64负责译码，这个模块在tools.v中。译码部分使用了循环语句，而且只有最高位出错，所以怀疑循环语句的判断条件错误。

1. 错误原因

循环语句generate for中，变量i的变化范围为0至62，没有包括最高位。

1. 修正效果

修改循环条件，使i变化范围为0到63。

图片包含 表格

描述已自动生成

图4 修改后的循环条件

修改后变量赋值成功。

1. 错误4：ALU模块调用错误
2. 错误现象

控制台报错，提示wb\_rf\_wdata出错。

page7image64650480

图5 控制台报错信息

1. 分析定位过程

经过追溯发现出错的数据为alu\_result。查看调用ALU模块的部分时，发现ALU模块的输入alu\_src1与错误的端口es\_alu\_src2相连。

1. 错误原因

模块ALU调用时连接到了错误的端口。

1. 修正效果

修改连接的端口，修改后同一指令处不再报错。

1. 错误5:组合环
2. 错误现象

控制台在PC=0xbfc00000时报错（报错信息见图5），在检查ALU模块时，发现or\_result和alu\_result之间形成了组合环。

1. 分析定位过程

检查ALU模块的源码时发现其中的or\_result和alu\_resut之间形成了环。

1. 错误原因

变量or\_result的赋值语句出错。

1. 修正效果

因为or\_result只和alu\_src1和alu\_src2有关，因此将or\_result的赋值语句中的alu\_result删除。

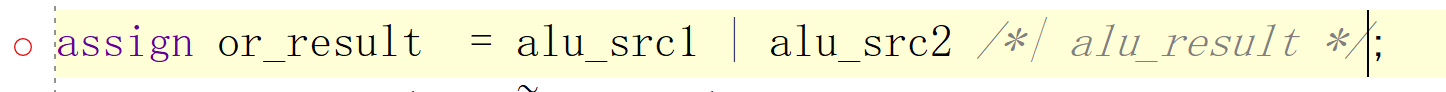


图6 修改or\_result的赋值语句

修改后同一指令处不再报错。

1. 错误6:跳转指令beq出错
2. 错误现象

控制台报错，提示PC出错。

page7image64647568

图7 控制台报错信息

1. 分析定位过程

从报错位置的前一条指令开始看，程序没有跳转。从PC可以判断前一指令为beq指令。这一步本来应该发生跳转，但是却没有，于是判断跳转条件没有满足，其中信号赋值错误。

1. 错误原因

变量br\_bus和br\_target位宽均为32，变量br\_taken位宽为1，所以IF模块和ID模块中对这三个变量的赋值会导致数据丢失。

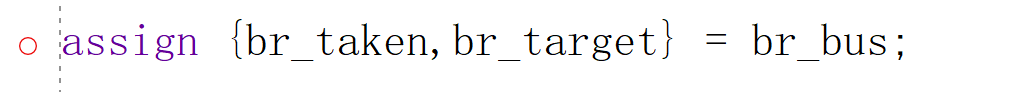


图8 IF模块中赋值语句



图9 ID模块中赋值语句

1. 修正效果

给br\_bus增加一位位宽。可以修改mycpu.h文件中的宏定义`BR\_BUS\_WD为33.

修改后同一指令处不再报错。

1. 错误7：移位指令srl出错
2. 错误现象

控制台报错，提示wb\_rf\_wdata出错。

page8image64620160

图10 控制台报错信息

1. 分析定位过程

先看PC值，确认此时是移位指令，也就是说此时wb\_rf\_wdata为alu\_result，ALU模块中移位操作出错。

1. 错误原因

ALU模块中，移位指令的计算结果只保留了低31位，导致出错。

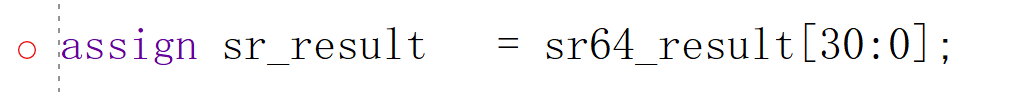


图11 移位指令赋值出错

1. 修正效果

将上图中30改为31。修改后同一指令处不再报错。

1. 错误8：越沿赋值
2. 错误现象

这一错误在此任务中不会影响结果，但是为了代码规范，防止以后出错，所以必须修改。

1. 分析定位过程

在检查MEM模块时，发现给信号es\_to\_ms\_bus\_r赋值时采用的是阻塞赋值（图中光标处）。

图形用户界面, 文本, 应用程序

描述已自动生成

图12 阻塞赋值

1. 错误原因

没有采用非阻塞赋值。

1. 修正效果

修改为非阻塞赋值。

四、实验总结（可选）

修改所有错误后，上板成功（因为绿色led灯亮度太高，所以，为了看清显示的数字，调低了整张照片的亮度）。

错误太多了，我裂开了.jpg。



图13 上板成功结果