***2021***

****

**计算机组成原理 ·实验报告·**

|  |  |
| --- | --- |
| 学 院： | 计算机与信息学院 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： |  |
| 电 话： |  |
| 邮 件： |  |
| 完成日期： | 2021-07-07 |
| 指导教师： | 阙 夏 |

目 录

[1 Verilog与vivado基础 2](#_Toc6665)

[1.1 设计要求 2](#_Toc4086)

[1.2 方案设计 2](#_Toc3295)

[1.3 实验步骤 4](#_Toc17540)

[1.4 故障与调试 4](#_Toc29478)

[1.5 仿真及分析 5](#_Toc1155)

[2 单周期CPU设计与实现——单指令CPU 6](#_Toc24271)

[2.1 设计要求 6](#_Toc12847)

[2.2 方案设计 6](#_Toc10489)

[2.3 实验步骤 8](#_Toc27342)

[2.4 故障与调试 14](#_Toc2019)

[2.5 仿真及分析 14](#_Toc18773)

[3 总结与心得 16](#_Toc14752)

[3.1 实验总结 16](#_Toc3791)

[3.2 实验心得 16](#_Toc4420)

[参考文献 17](#_Toc26446)

# Verilog与vivado基础

## 设计要求

熟悉并掌握 Verilog HDL 与 vivado 的使用。

请使用 Verilog 完成4选1 多路选择器（MUX） 、4 位全加器、8 位比较器、74138 译码器等模块设计，然后编写测试文件进行仿真验证。

## 方案设计

多路选择器（MUX）是一种在多路数据传送过程中，能够根据需要将其中任意一路选出来的电路，其原理图和真值表如图 1.1所示。

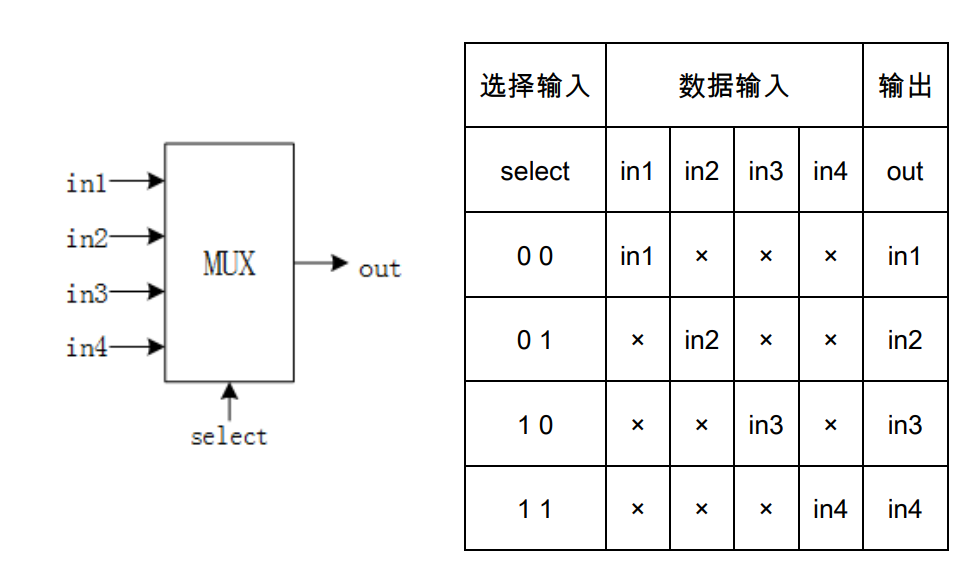
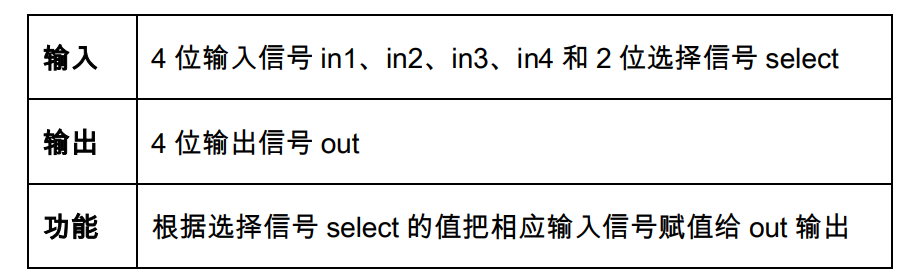


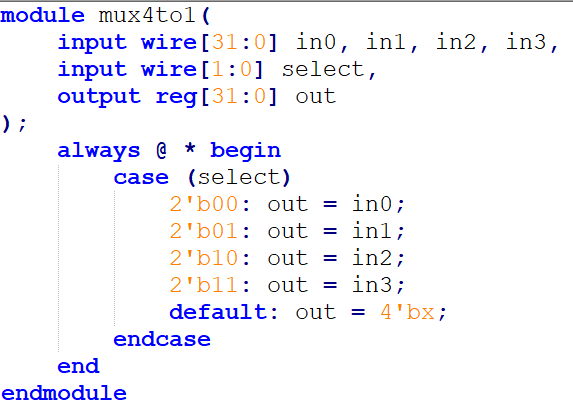
图 1.1 4选1多路选择器及其真值表

### Verilog 关键代码实现

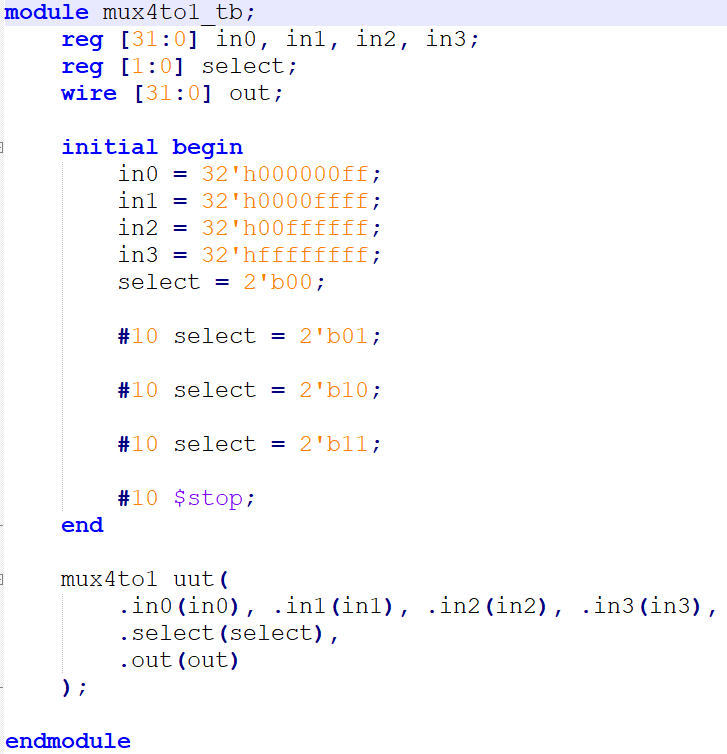
表 1.1 MUX模块功能描述

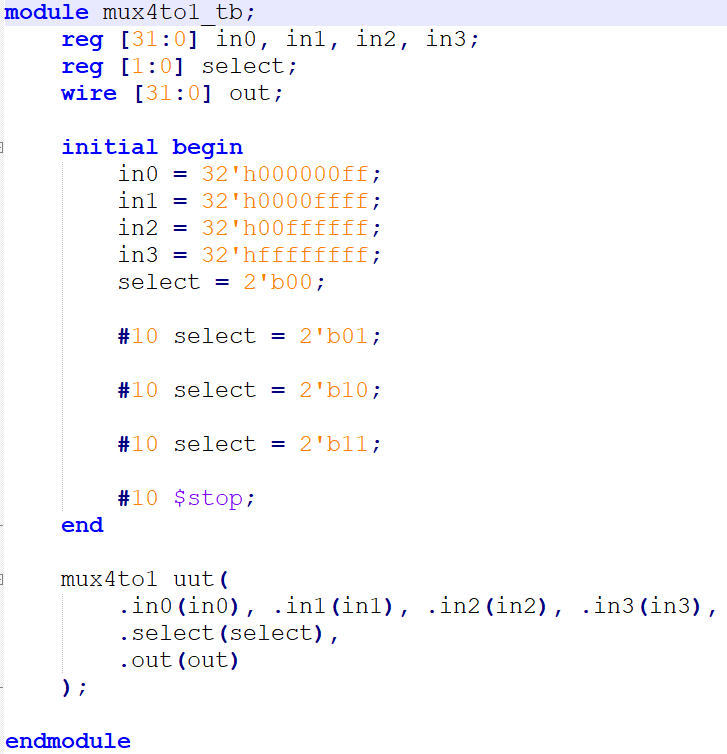


MUX 模块的功能描述如表 1.1所示，模块的verilog 代码如下：



### 测试文件(TestBench)关键代码描述





## 实验步骤

### 建立ModelSim工程

打开 ModelSim，选择 File->New->Project，出现 Create Project 对话框，填写工程名（Project Name），选择保存目录（Project Location），注意保存目录中不要有中文，如下图所示：

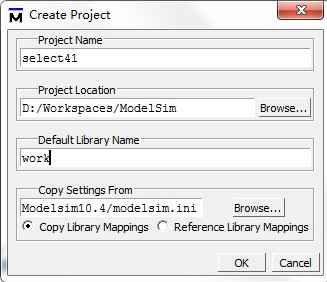


图1.2 ModelSim创建工程

单击 OK 按钮后，会出现下图界面：

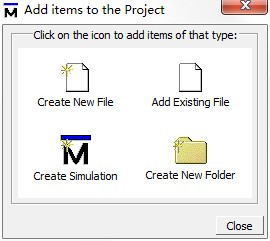


图1.3 ModelSim添加工程文件

现在可以选 Create New File（新建文件）或者 Add Existing File（添加已存在文件）。这里我们选择 Add Existing File，也就是添加 5.1 和 5.2 中的 Verilog 代码，会出现下图界面：

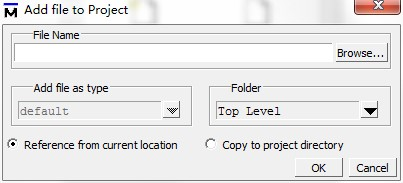


图1.4 ModelSim添加工程

点击 Browse 按钮，添加 5.1 中的 mux41.v 和 5.2 中的测试文件 mux41\_tb.v，会出现下图界面：

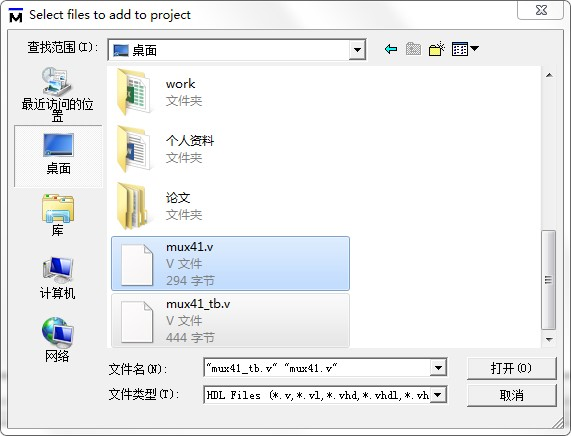


图1.5 选择文件

选择要添加的文件后，单击“打开”按钮，即添加完成，会出现下图界面，在其中选择 copy to project directory，这样就会将 mux41.v 和 mux41\_tb.v 文件复制到新的工程目录下，单击OK 按钮。

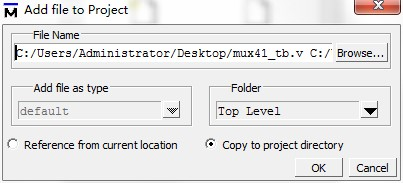


图1.6 选择文件2

文件添加完成后，ModelSim 主界面会显示所有文件的状态。选中任意一个文件，右键单击，选择 Compile->Compile All，即开始编辑所有文件，会出现下图界面。没有出错，文件状态应该都是绿色的对号，否则点击屏幕下方的 Transcript，查看出错信息，直至无误。

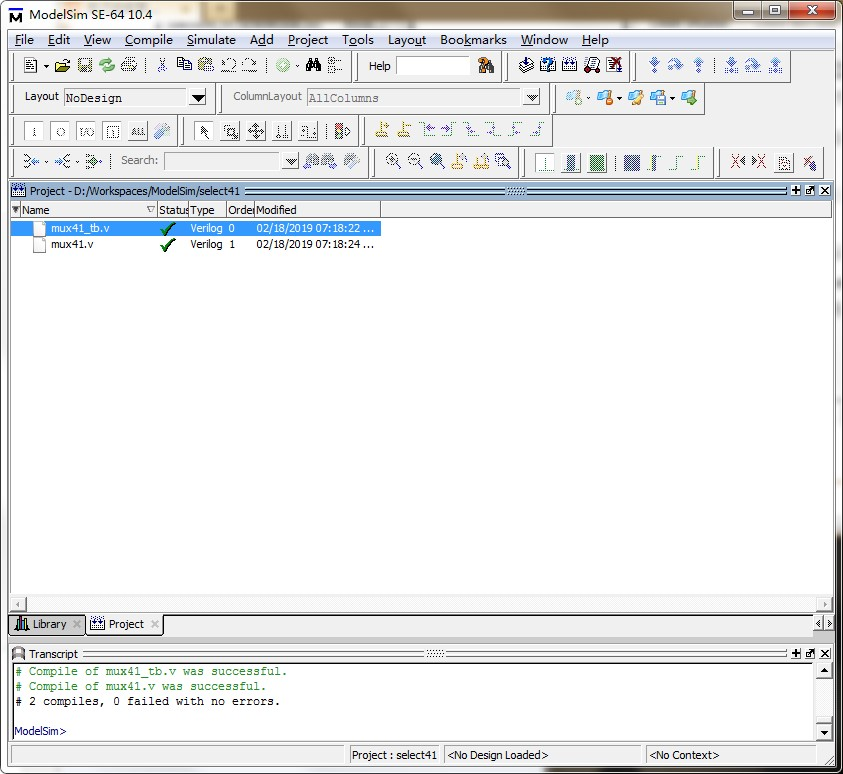


图1.7 选择文件

### 开始仿真

切换到 Library，然后展开 work 目录，在 mux41\_tb.v 文件上单击右键，在弹出菜单中选择 Simulate （without Optimization），如下图界面：

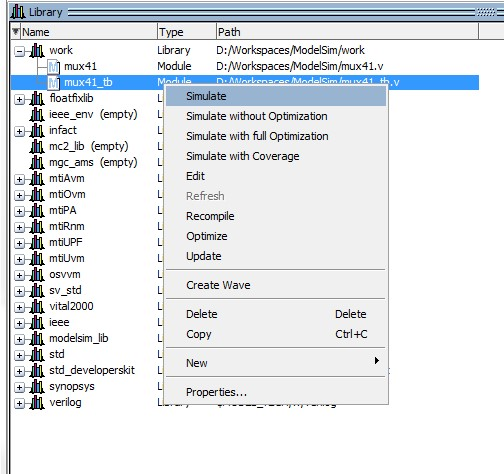


图1.8 选择Library中文件

此时会出现一个名为 sim 的界面，展开其中的 mux41\_tb 节点，选择 uut，会在 Objects 窗口显示所有信号，如下图所示：（若没有出现 Objects 窗口，可以通过菜单 View->Objects调出该窗口）

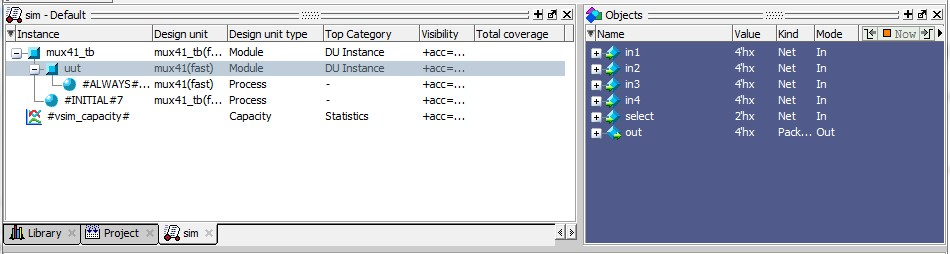


图1.9 展开节点

选择 Objects 窗口的所有信号（Ctrl+A），然后单击右键，在弹出菜单中选择 Add

to->Wave->Selected Signals，如下图所示：

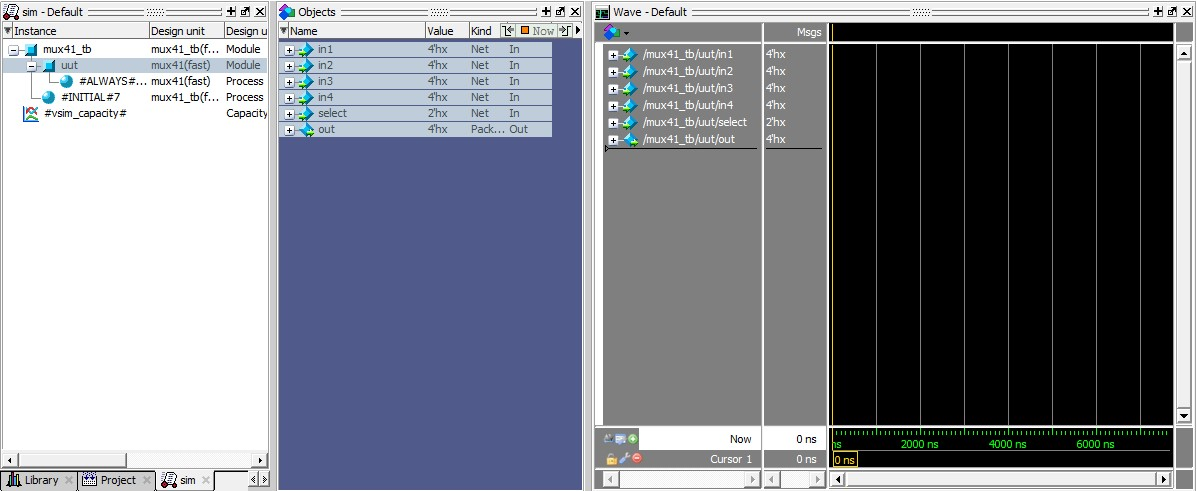


图1.10 选择展示波形

单击工具栏中的 Run-All 按钮，便开始仿真，如下图所示：

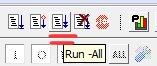


图1.11 选择文件

结束后，请在 Transcript 中输入 quit –sim 命令退出，如下图所示：

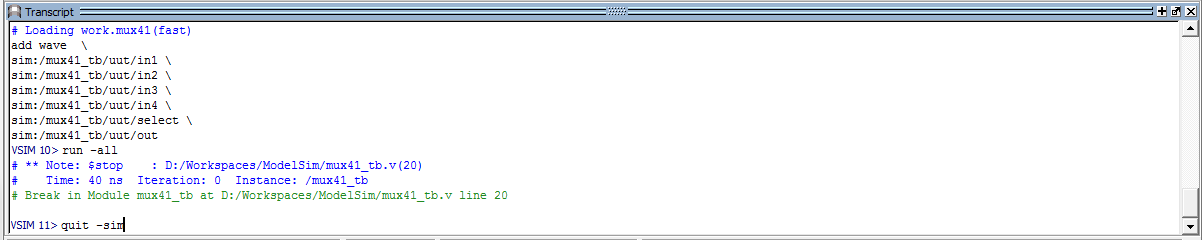


图1.12 结束仿真

## 故障与调试

### ModelSim编译出现问题

**故障现象：** .v文件编译不通过

**原因分析：文件内容出现错误。**

**解决方案：对.v文件内部进行差错，发现错误并改正**

### 带有Include的文件编译出错

**故障现象：只要用到include，编译就出错，爆出Cannotopen `include file \*.v**

**原因分析：文件调用出现错误。**

**解决方案：将编译文件放在同一个文件夹中，并编译时要在文件区中进行。**

## 仿真及分析

仿真测试用例见表 1.2。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | in1 | in2 | in3 | in4 | select | out |
| 10 | 0001 | 0011 | 0111 | 1111 | 00 | 0001 |
| 20 | 0001 | 0011 | 0111 | 1111 | 01 | 0011 |
| 30 | 0001 | 0011 | 0111 | 1111 | 10 | 0111 |
| 40 | 0001 | 0011 | 0111 | 1111 | 11 | 1111 |

表 1.2 仿真测试用例

仿真的结果如下：

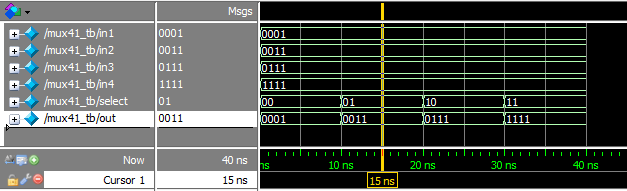


图1.13 仿真结果

从图1.13中可以看到，根据select选择器的选择结果，选择器所选择的数字对应的输入情况将在输出中反应出来，这符合4选1数据选择器的结果。实验结果较为成功。

# 单周期CPU设计与实现——单指令CPU

## 设计要求

通过设计并实现支持一条指令的 CPU，理解和掌握 CPU 设计的基本原理和过程。

## 方案设计

### 指令设计

设计和实现一个支持加法指令的单周期 CPU。要求该加法指令（表示为 add r1，r2，r3）格式约定如下：

1. 采用寄存器寻址，r1，r2，r3 为寄存器编号，r1 和 r2 存放两个源操作数，r3为目标寄存器，其功能为[r1] + [r2] -> r3；
2. 指令字长 16 位，操作码和地址码字段分配如下所示：

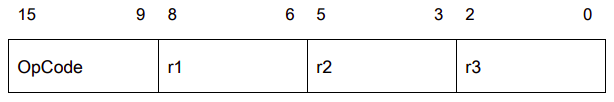


图2.1 加法指令结构设计

### CPU数据通路的设计

本实验需要设计的 CPU 只需要支持一条加法指令，而该指令的功能是在一个时钟周期内从寄存器组中 r1 和 r2 中取出两个操作数，然后送到 ALU 进行加法运算，最后把计算结果保存到 r1 寄存器中。下图给出了改加法指令的数据通路图。

此外，还需要确定各个部件的位数，为了简单起见，我们假设目标 CPU 的机器字长、存储字长和指令字长相等均为 16 位，存储单元个数假设为 256，按字寻址，并取 PC 位数为 8。

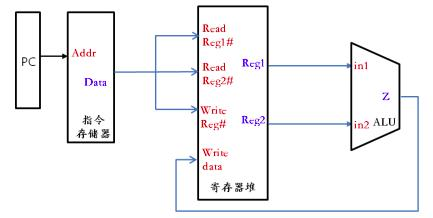


图2.2 CPU数据通路设计图

### 控制单元的设计

控制单元的功能是为当前要执行的指令产生微操作命令从而完成该指令的执行。为了能够完成加法指令的执行，结合图 1，控制单元需要在取出指令后根据指令操作码（本例中是 加法指令），控制 ALU（参考实验二）做加法（通过给 alu\_op 信号线相应赋值），并把结果 写回寄存器组（参考实验三）中（通过给 wr\_en 赋值为 true）。图 2.3 给出了整合控制单元后 目标 CPU 的原理图，系统时钟信号也已标注。

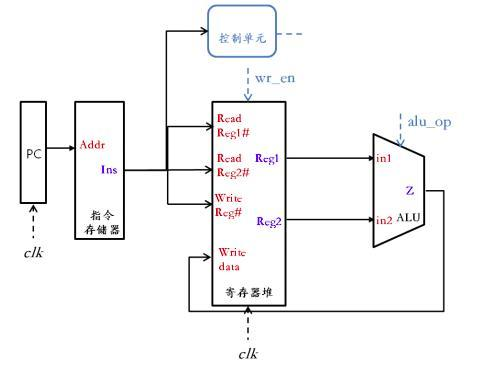


图2.3 控制单元设计

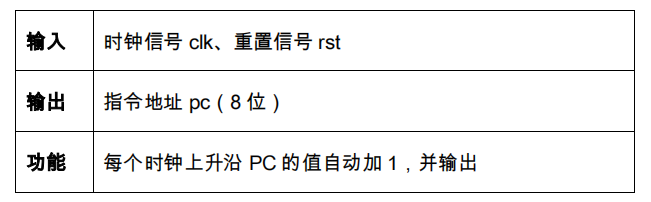
## 实验步骤

在第三部分通过对该 CPU 实现细节的分析、设计，并得到该 CPU 的原理图后，就可以依次实现各个模块，并进行仿真验证了。

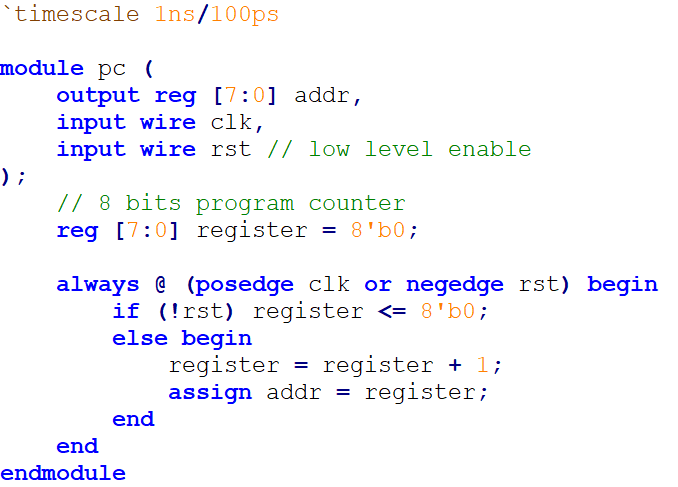
### CPU各模块Verilog实现

1. PC模块

表2.1 PC模块功能描述

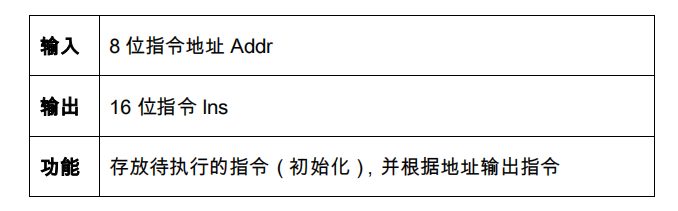


Verilog关键代码：

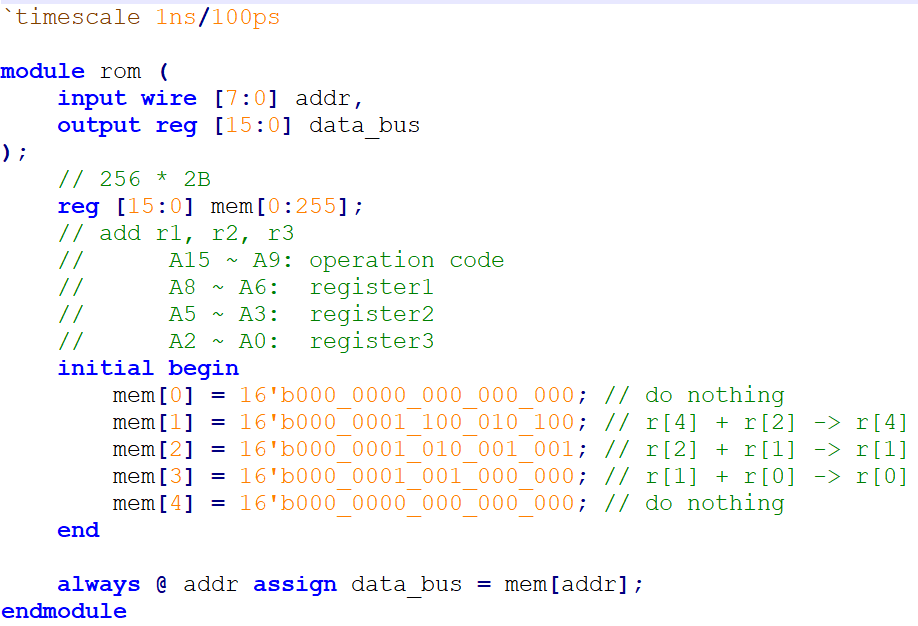


1. 指令存储器模块

表2.2 指令存储器模块功能描述

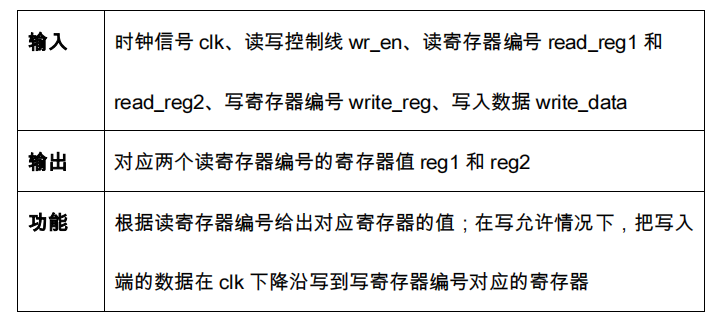


Verilog关键代码：

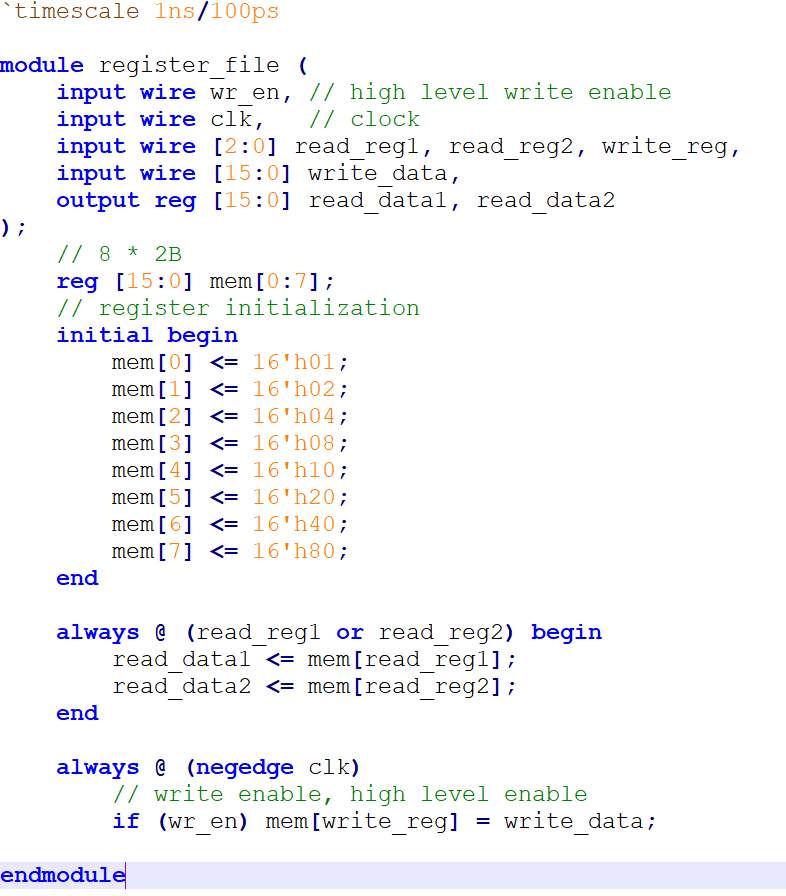


1. 寄存器堆

表2.3 寄存器堆模块功能描述

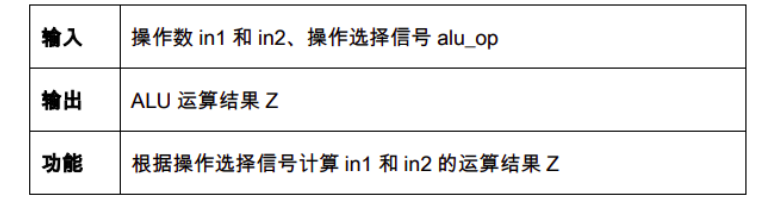


Verilog关键代码：

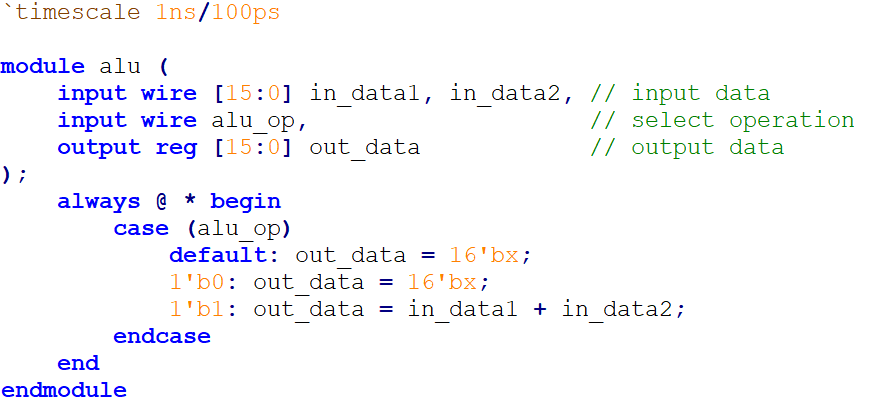


1. ALU

表2.4 ALU模块功能描述

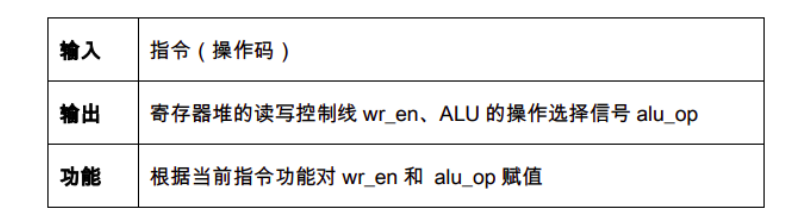


Verilog关键代码：

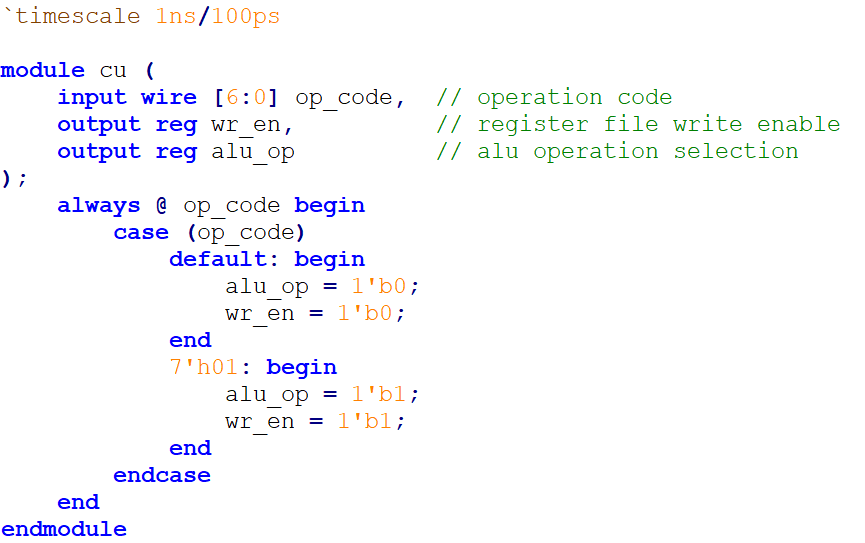


1. 控制单元

表2.5 控制单元模块功能描述



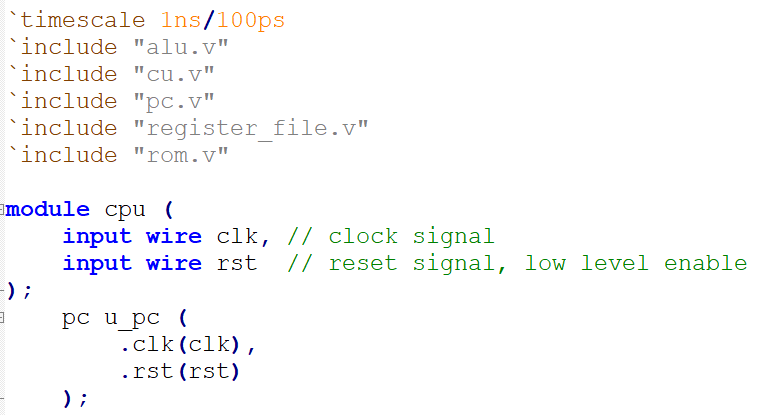
Verilog关键代码：

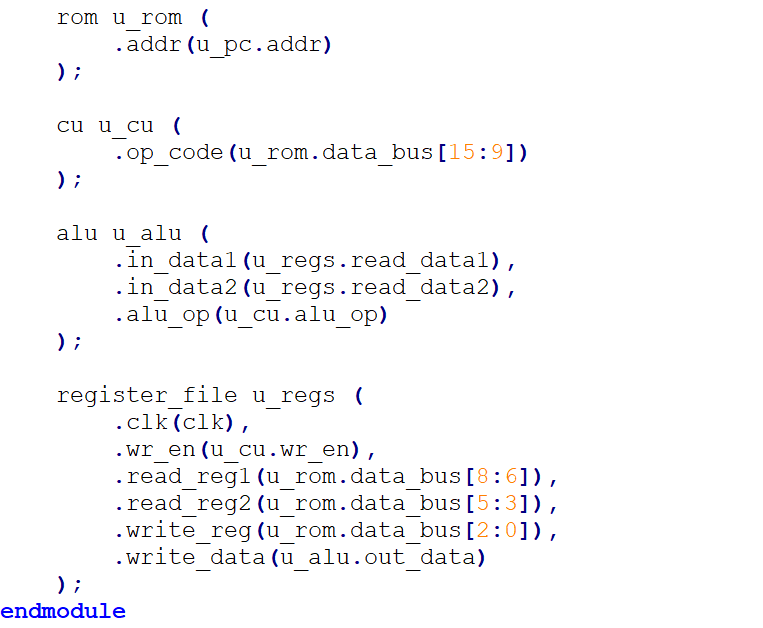


### CPU顶层文件封装实现

通过根据图 2 将以上定义的模块进行连接、封装就得到了目标 CPU，该 CPU 的输入为 系统时钟信号 clk 和重置信号 reset。

Verilog关键代码：

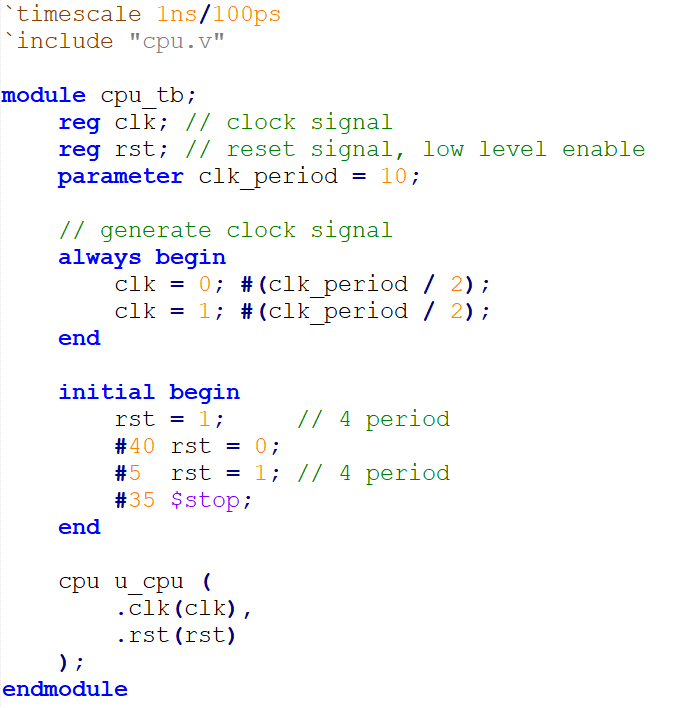


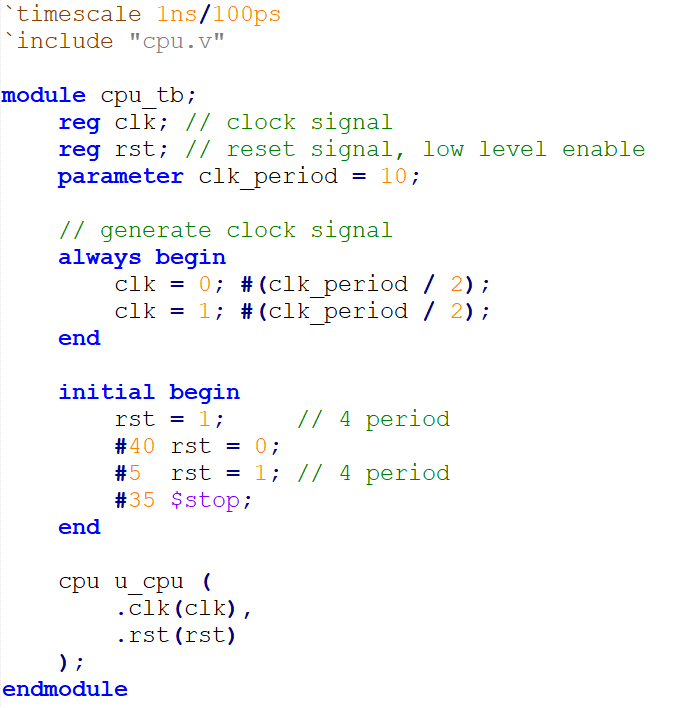


### CPU模拟仿真

为了仿真验证所实现的 CPU，需要定义测试文件并在测试文件中对指令存储器和寄存 器堆中的相应寄存器的值进行初始化，并通过仿真波形图查看是否指令得到了正确执行。

1. TestBench关键代码：





## 故障与调试

### ModelSim编译出现问题

**故障现象：** .v文件编译不通过

**原因分析：文件内容出现错误。**

**解决方案：对.v文件内部进行差错，发现错误并改正**

### 带有Include的文件编译出错

**故障现象：只要用到include，编译就出错，爆出Cannotopen `include file \*.v**

**原因分析：文件调用出现错误。**

**解决方案：将编译文件放在同一个文件夹中，并编译时要在文件区中进行。**

## 仿真及分析

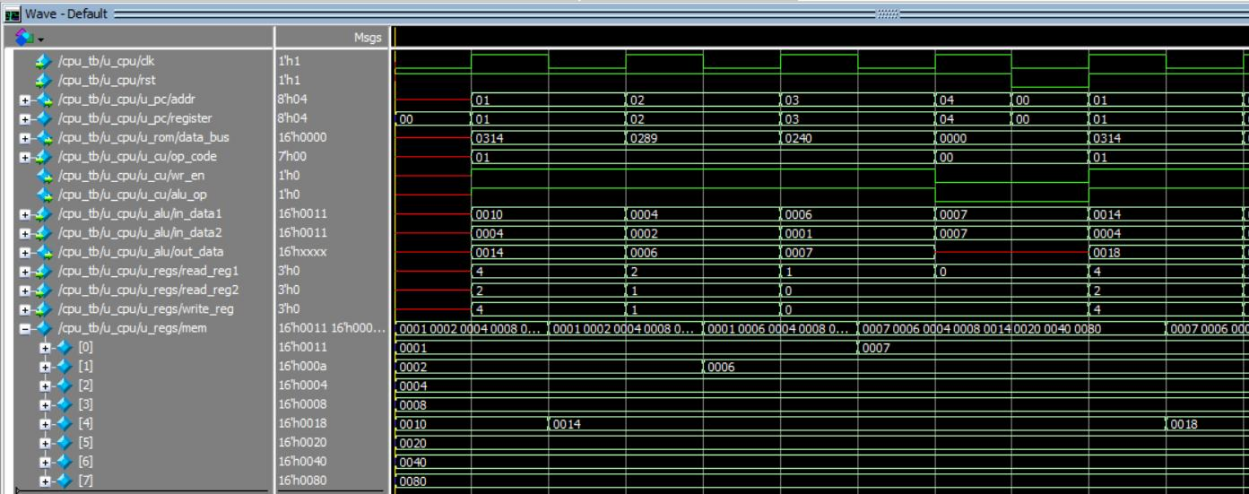


图2-4 ModelSim仿真结果

通过仿真波形可以看到，前4个时钟周期，alu都根据指令算出了正确的结果，并存储到了正确的寄存器中，而第5个时钟周期的时候输入了复位信号，PC寄存器又回到初始地址，程序又开始重复执行，并且所有器件均工作正常。

# 总结与心得

## 实验总结

本次实验主要完成了如下几点工作：

1. 完成方案总结
   1. 完成实验1：Verilog与Vivado基础的方案设计
   2. 完成实验4：单周期CPU的设计——单指令CPU
2. 功能总结
   1. 成功实现了4选1数据选择器的功能
   2. 成功实现了CPU中ALU的功能
   3. 成功实现了CPU中PC的功能
   4. 成功实现了CPU中指令存储器的功能
   5. 成功实现了CPU中寄存器堆的功能
3. 其他需要总结的内容。
   1. 较为熟练的掌握了Verilog的语法
   2. 较为熟练的掌握了Vivado、ModelSim等软件的使用

## 实验心得

1. 在本次实验中，对于环境的配置和软件的设置尤为艰难。在老师的实验指导书、同学演讲分享经验以及从网上寻找答案，经过多次尝试后，完美的配置了环境，以及调试好了软件。这是一份非常宝贵的经验。
2. 另一个非常重要的是Verilog代码的书写。这一个部分我从老师提供的实例代码开始，一步步的研究，找到了不同功能的实现方式。同时，互联网也为我提供了很多的帮助，一些人求助的博客也解决了我在编译、调试过程中出现的问题。

# 参考文献

1. DAVID A.PATTERSON(美).计算机组成与设计硬件/软件接口(原书第5版).北京:机械工业出版社.
2. David Money Harris(美).数字设计和计算机体系结构（第二版）. 机械工业出版社
3. 谭志虎,秦磊华,胡迪青.计算机组成原理实践教程.北京:清华大学出版社，2018年.
4. 秦磊华，吴非，莫正坤.计算机组成原理. 北京:清华大学出版社，2011年.
5. 袁春风编著. 计算机组成与系统结构. 北京:清华大学出版社，2011年.
6. 张晨曦，王志英. 计算机系统结构. 高等教育出版社，2008年.

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  特此声明！  **作者签字: 嵌入签名图片** |
| 二、对课程实验的学术评语（教师填写） |
|  |
| 三、对课程实验的评分（教师填写） |
| |  |  |  |  | | --- | --- | --- | --- | | 评分项目  （分值） | 报告撰写  （30分） | 课设过程  （70分） | 最终评定  （100分） | | 得分 |  |  |  | |
| **指导教师签字:** |