

**<<基于 Verilog\_HDL 的 FPGA 设计基础>>  
实验指导书**

电子工程学院 微电子系

## 说 明

- 1、本实验指导书用于《基于 Verilog\_HDL 的 FPGA 设计基础》课程内实验指导。
- 2、实验指导书主要内容只包括大纲所列课内基本实验，学生自选实验题目可由学生自由选择，并参考本课程网站所提供的复杂电路设计实例完成。
- 3、实验所用 EDA 工具的使用方法均配有屏幕录像视频，可配合实验指导书一起使用。

## 实验报告格式和要求

实验报告要求采用按“设计规范”的方式撰写，具体格式和要求如下：

第一页：

实验题目

版本 v1.0

专业：

班级：

姓名：

学号：

日期

第二页开始：

一、功能描述：

应简要，但全面的描述实验题目要完成的功能。

二、设计方案：（如果题目复杂，可分为总体设计方案和详细设计方案）

对设计思想，方法以及所设计电路的基本结构进行描述，辅以总体设计框图，子模块框图（多模块的复杂设计情况）以及管脚列表进行说明。

三、验证方案：

给出验证设计的基本思想和方法，仿真用例等。如果是需要下载到开发板上的实验，要给出开发板型号、管脚分配的.udf 文件

四、设计代码：

带有注释的设计和仿真代码。

五、仿真结果分析

1、逻辑仿真结果：在 Modelsim 中对设计仿真，对照验证方案对仿真结果进行分析说明。

2、下载验证：对下载到开发板上的设计仿真结果进行说明，最好配有结果照片。

## 实验一 熟悉 Modelsim 仿真环境

### 一、 实验目的

熟悉 Modelsim 仿真环境，为后续的电路设计、仿真等工作打好基础。

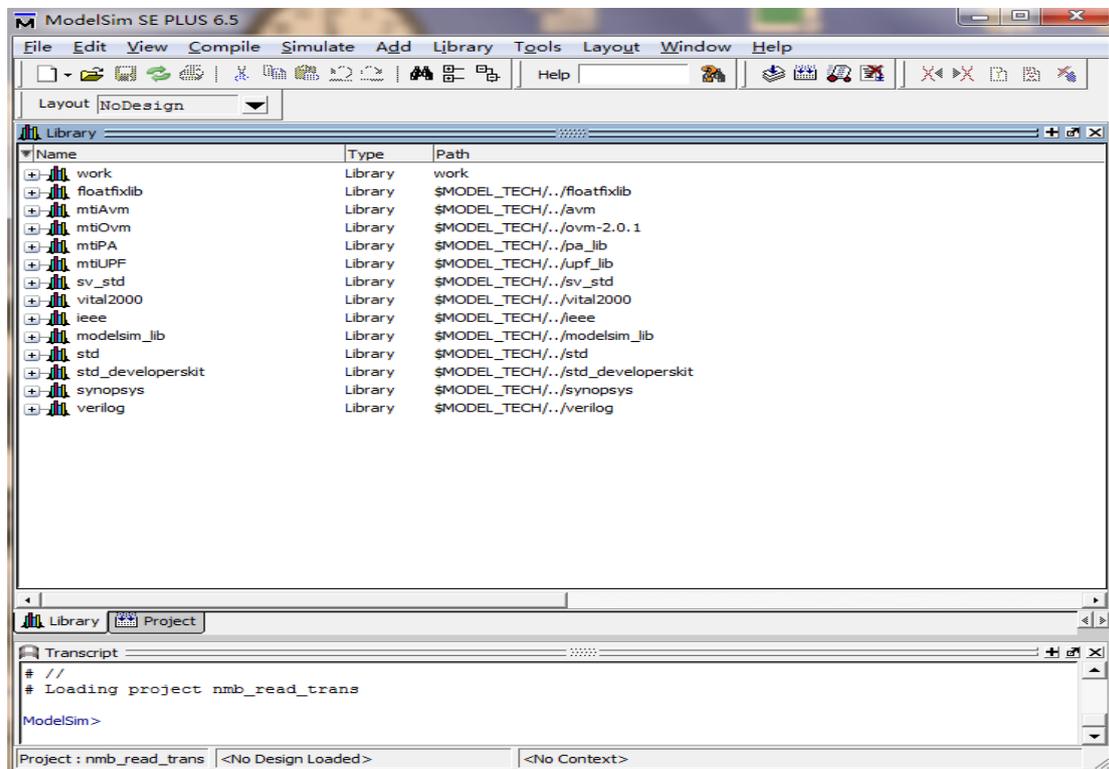
### 二、 实验要求

- 1、学会使用 Modelsim 仿真器的基本功能；
- 2、按照仿真程序的结构编写仿真程序。

### 三、 实验原理

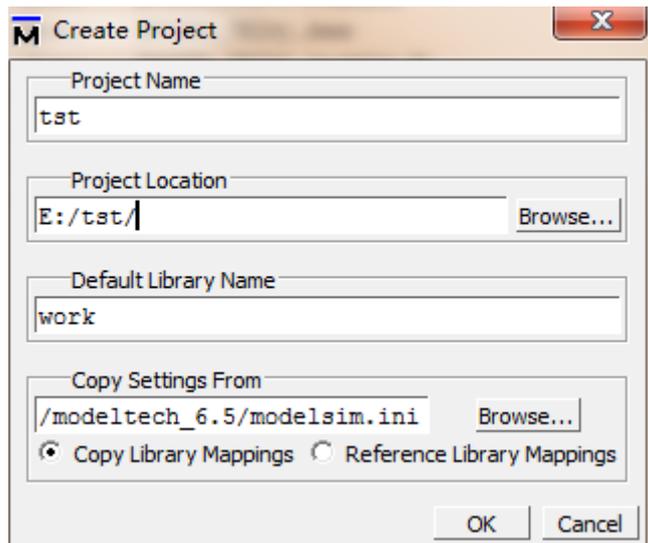
ModelSim HDL 仿真工具同时支持 VHDL 和 verilog 两种硬件描述语言的仿真及混合仿真。它们的仿真步骤基本一致。在此只给出 verilog 的仿真步骤。

- 1、打开 Modelsim 工具，显示如下界面：

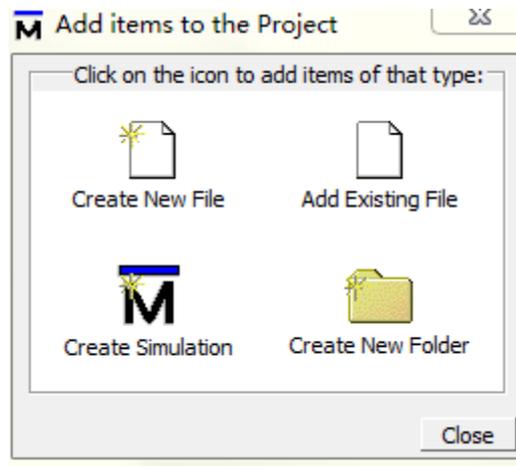


### 2、创建工程

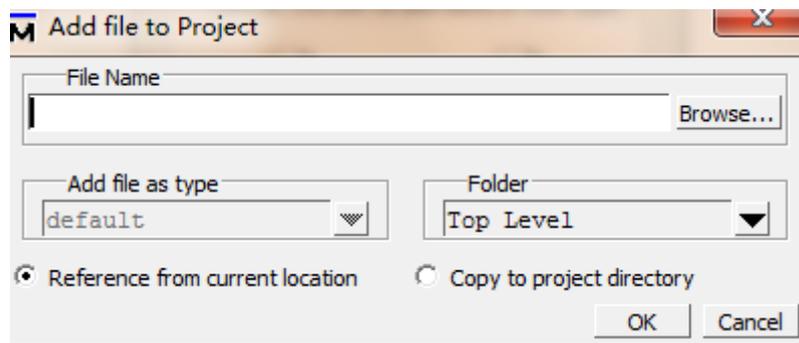
- 1) 点击 File->new->project，弹出下面窗口：



2) 在窗口中敲入工程名和工程存储的目录地址，点击 OK 按钮，弹出如下窗口：

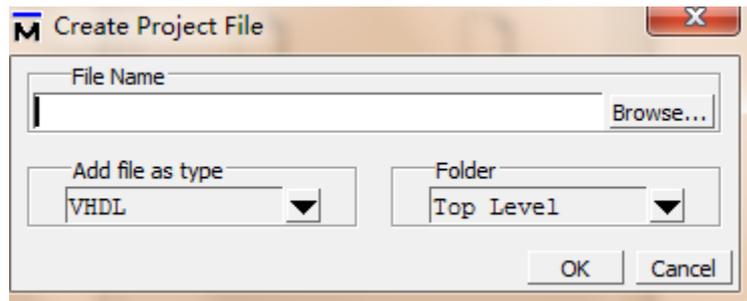


3) 如果已经编写好程序代码，直接点击“add existing file”，弹出如下窗口：

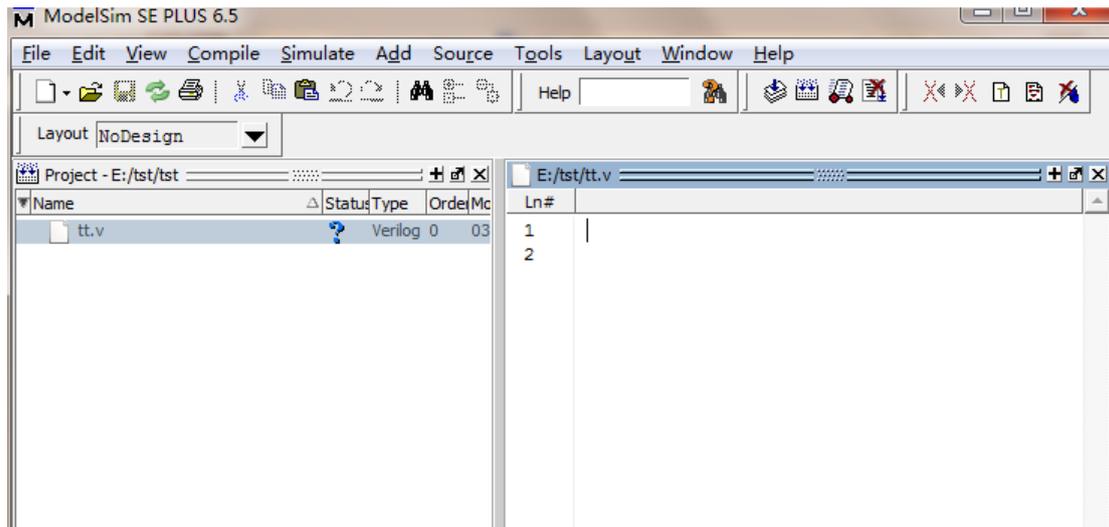


点击 Browse 按钮选择编辑好的代码文件，然后点击 OK。

如果没有编辑好的代码，需要在仿真工具中编辑，点击“create new file”，弹出如下窗口：



敲入代码文件名，并在“add file as type”栏选择 verilog 格式，点击 OK，出现如下画面：



然后双击文件名，对文件进行编辑。

### 3、编译文件

- 1) 点击按钮  编译所有工程下的文件；
- 2) 点击按钮  对单个文件进行编译。

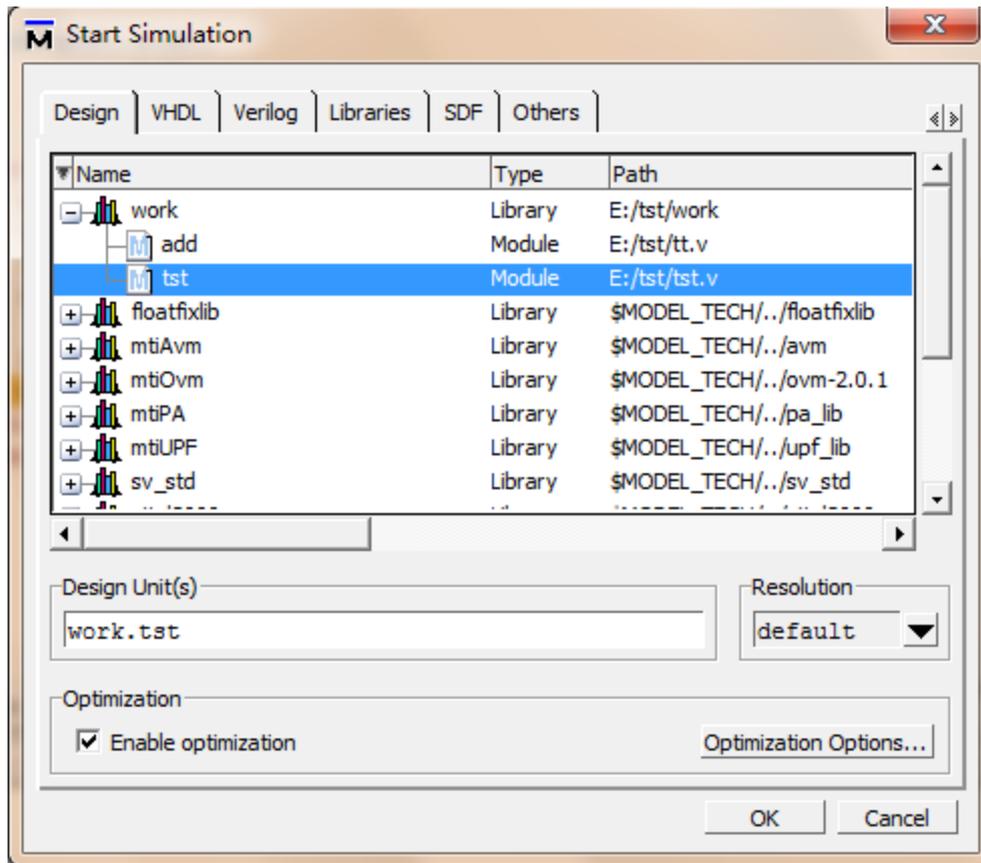
编译结果如下窗口所示：

```
# Compile of tt.v was successful.
# Compile of tst.v failed with 1 errors.
# 2 compiles, 1 failed with 1 error.
# Compile of tt.v was successful.
# Compile of tst.v was successful.
# 2 compiles, 0 failed with no errors.
ModelSim>
```

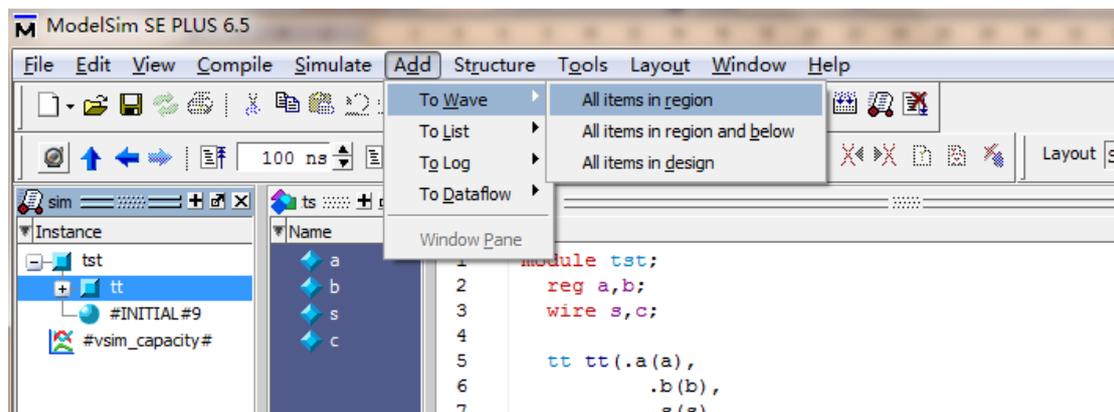
如果编译成功，显示绿色成功指示；如果不成功，报告错误个数，双击红色错误即可显示错误的具体细节。

### 4、开始仿真

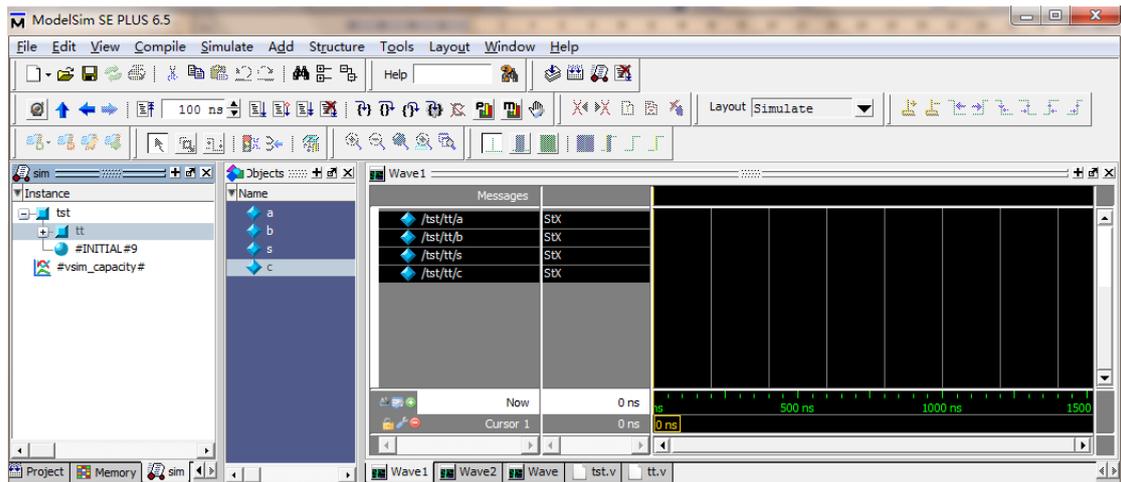
1) 点击 simulation->start simulation，展开 work 工作库，选择测试文件为顶层文件，点击 OK。

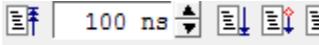


2) 点击 add->to wave->all items in region and below, 添加波形文件, 将顶层目录下的所有信号都添加到波形文件中。也可以选择 all items in region, 只添加顶层文件中的信号, 或者添加 all items in design 添加设计中的信号!

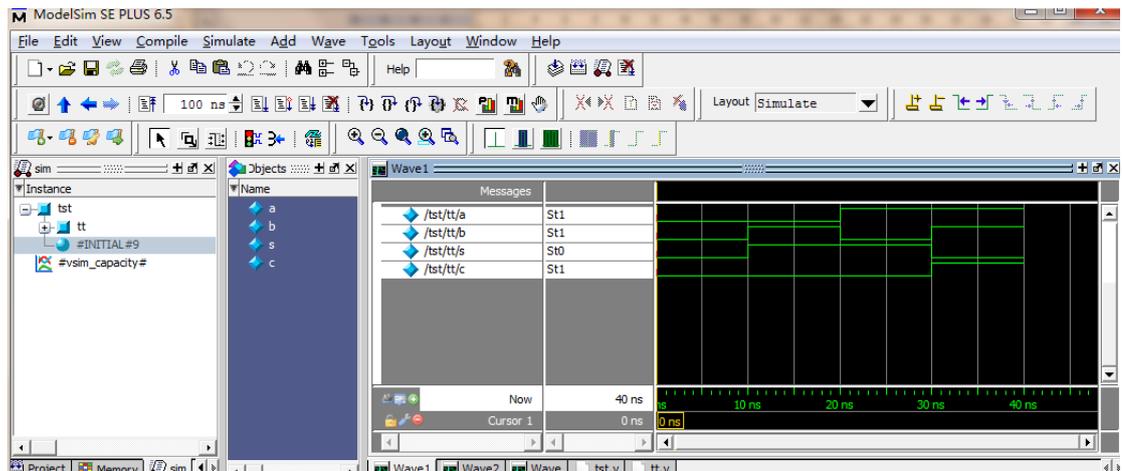


点击后弹出如下窗口:



3) 点击按钮  运行，运行时间如按钮左边窗口显示  100 ns   ，运行时间可以修改。继续运行可以再次点击按钮 。重新开始运行点击按钮 。中断运行点击按钮 。中断后，从中断点继续运行点击按钮 。一直运行点击按钮 ，注意点击一直运行按钮后，除非测试文件中有\$stop 或者\$finish，否则仿真不会自动停止。

4) 点击运行按钮后，显示如下画面：



缩小波形图点击按钮 ，放大波形图点击按钮 。

## 四、 实验内容及步骤

### 1、 实验内容

用 Modelsim 工具仿真一个 4 位同步计数器电路。

### 2、 实验步骤

- 1) 按照实验原理的步骤要求完成工程的创建工作；
- 2) 编写 4 位同步计数器电路；
- 3) 编写 4 位同步计数器电路仿真程序；
- 4) 按照实验原理的步骤完成电路的编译及仿真工作，观察输出波形，检验结果是否正确，如果正确完成实验，如果结果出错，通过查看电路中间变量等方

法，纠错，并重新编译，仿真，直到结果正确为止。

## 五、 撰写报告

按附录要求撰写总结报告

## 实验二 设计一个四位的先行进位全加器

### 一、 实验目的

- 1、掌握门级电路描述的方法；
- 2、掌握电路仿真程序编写技巧。

### 二、 实验要求

- 1、可综合的门级电路描述一个 4 位的先行进位全加器；
- 2、编写电路仿真程序，要求测试输入的所有组合情况；
- 3、观察仿真结果。

### 三、 实验原理

串行进位加法器需要一级一级的进位，进位延迟很大。先行进位加法器（也叫超前进位加法器）可以有效的减少进位延迟。

设二进制加法器的第  $i$  位输入为  $X_i, Y_i$ ，输出为  $S_i$ ，进位输入为  $C_i$ ，进位输出为  $C_{i+1}$

则有

$$S_i = X_i \oplus Y_i \oplus C_i$$

$$C_{i+1} = X_i \cdot Y_i + X_i \cdot C_i + Y_i \cdot C_i = X_i \cdot Y_i + (X_i + Y_i) \cdot C_i$$

令  $G_i = X_i \cdot Y_i$ ， $P_i = X_i + Y_i$

则  $C_{i+1} = G_i + P_i \cdot C_i$

当  $X_i$  和  $Y_i$  都为 1 时， $G_i = 1$ ，产生进位  $C_{i+1} = 1$

当  $X_i$  和  $Y_i$  有一个为 1 时， $P_i = 1$ ，传递进位  $C_{i+1} = C_i$

因此  $G_i$  定义为进位产生信号， $P_i$  定义为进位传递信号。 $G_i$  的优先级比  $P_i$  高，也就是说：当  $G_i = 1$  时（当然此时也有  $P_i = 1$ ），无条件产生进位，而不管  $C_i$  是多少；

当  $G_i=0$  而  $P_i=1$  时，进位输出为  $C_i$ ，跟  $C_i$  之前的逻辑有关。

下面推导 4 位超前进位加法器。设 4 位加数和被加数为  $A$  和  $B$ ，进位输入为  $C_{in}$ ，进位输出为  $C_{out}$ ，对于第  $i$  位的进位产生  $G_i = A_i \cdot B_i$ ，进位传递  $P_i = A_i + B_i$ ， $i=0, 1, 2, 3$

于是这各级进位输出，递归的展开  $C_i$ ，有：

$$C_0 = C_{in}$$

$$C_1 = G_0 + P_0 \cdot C_0$$

$$C_2 = G_1 + P_1 \cdot C_1 = G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0) = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0$$

$$C_3 = G_2 + P_2 \cdot C_2 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0$$

$$C_4 = G_3 + P_3 \cdot C_3 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 +$$

$$P3 \cdot P2 \cdot P1 \cdot P0 \cdot C0$$

$$Cout=C4$$

由此可以看出，各级的进位彼此独立产生，只与输入数据和  $C_{in}$  有关，将各级间的进位级联传播给去掉了，因此减小了进位产生的延迟。

同时由真值表可以简单的得出第  $i$  位的和为：

$$S_i = X_i \oplus Y_i \oplus C_i = (X_i \cdot Y_i) \oplus (X_i + Y_i) \oplus C_i = G_i \oplus P_i \oplus C_i$$

根据上面的式子便可以设计出超前进位加法器。用 Verilog 编写了 4 位超前进位加法器。

#### 四、 实验内容及步骤

- 1、 根据先行进位全加器的工作原理，设计并绘制出门级电路图；
- 2、 根据门级电路图，使用 verilog HDL 语言，进行门级电路描述；
- 3、 编写电路的仿真激励，要求给出 4 位加数及被加数的所有组合情况，观察仿真结果。

#### 五、 撰写报告

按附录要求撰写总结报告

## 实验二 设计一个三八译码器

### 一、 实验目的

- 1、掌握组合电路设计方法；
- 2、熟悉电路仿真程序编写技巧。

### 二、 实验要求

- 1、用可综合的组合电路描述一个三八译码器；
- 2、编写电路仿真程序，要求测试输入的所有组合情况；
- 3、观察仿真结果。

### 三、 实验原理

译码器属于组合逻辑电路，它的逻辑功能是将二进制代码按其编码时的原意译成对应的输出高、底电平信号，又叫解码器。在数字电子技术中，它具有非常重要的地位，应用也很广泛。

三八译码器将三位的数据译码为八位的信号，其真值表为：

输入			输出
0	0	0	00000001
0	0	1	00000010
0	1	0	00000100
0	1	1	00001000
1	0	0	00010000
1	0	1	00100000
1	1	0	01000000
1	1	1	10000000

### 四、 实验内容及步骤

- a) 根据先行进位全加器的工作原理，设计并绘制出门级电路图；
  - b) 根据门级电路图，使用 verilog HDL 语言，进行电路描述；
- 3、编写电路的仿真激励，要求给出输入的所有组合情况，观察仿真结果。

### 五、 撰写报告

按附录要求撰写总结报告

## 实验三 熟悉 FPGA 设计流程

### 一、 实验目的

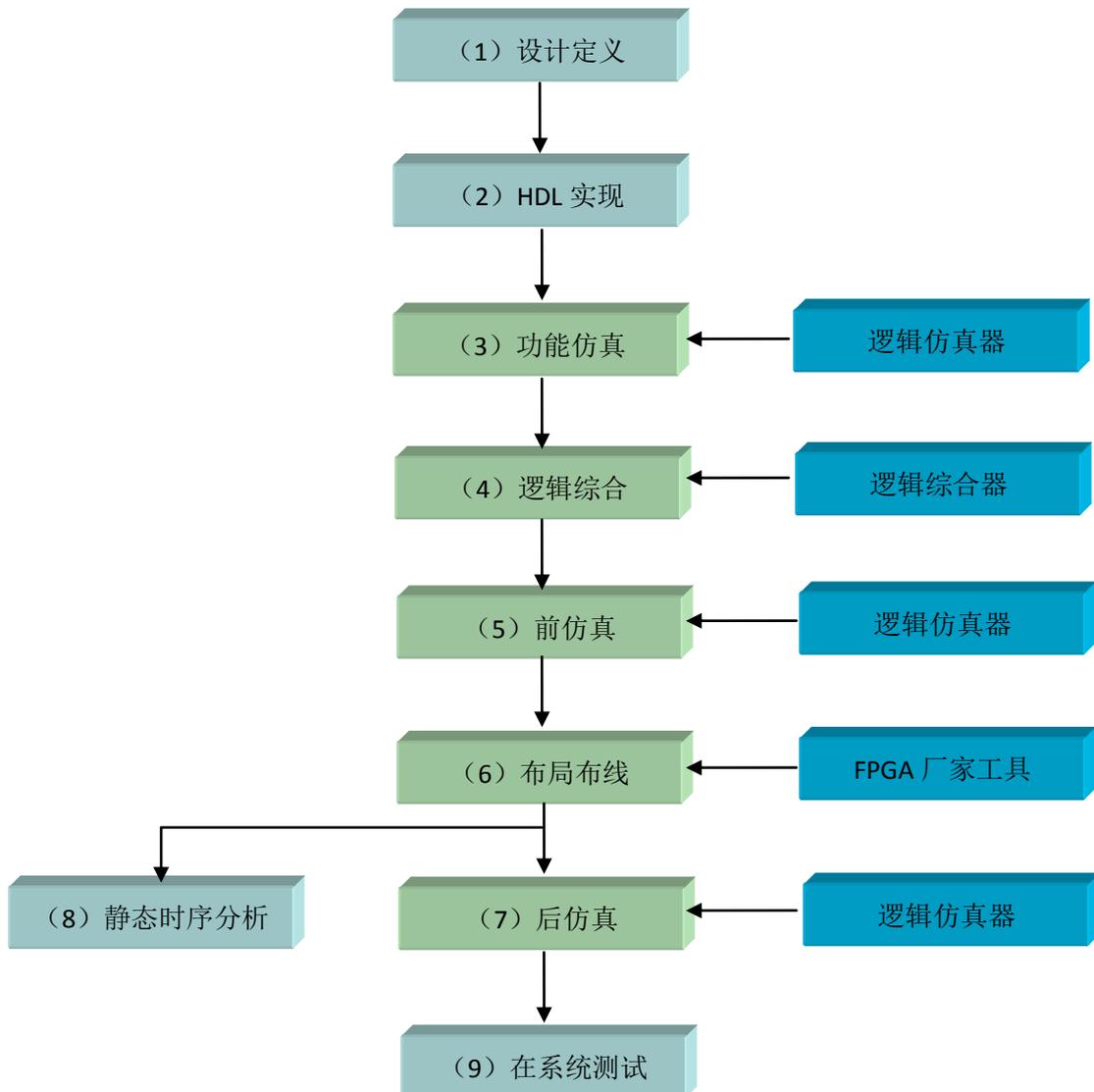
熟悉 Xilinx 公司的 ISE 设计和综合环境，为后续的电路设计、验证等工作打好基础。

### 二、 实验要求

- 1、学会使用 ISE 的基本功能；
- 2、按照实验指导书说明，完成一个计数器电路的综合、下载等工作。

### 三、 实验原理

FPGA 设计流程：

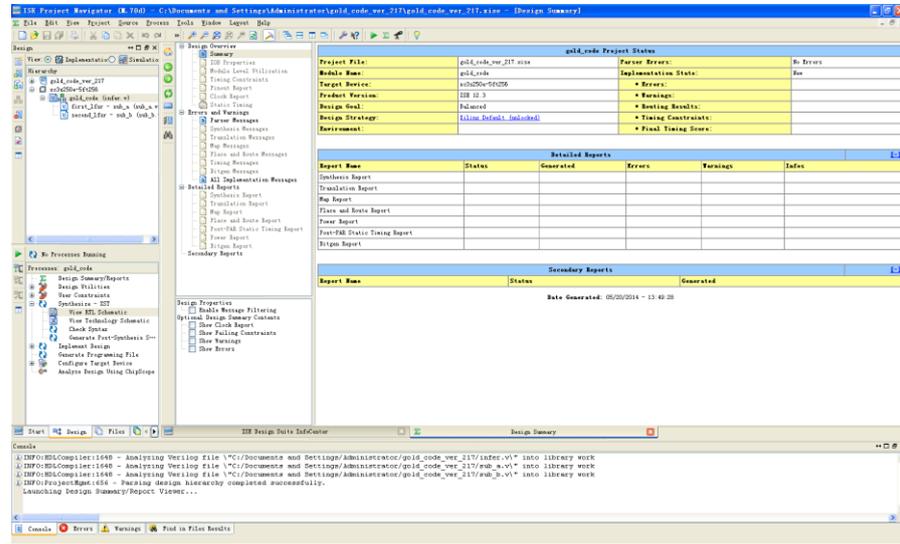


### 四、 实验步骤

## 1、操作步骤:

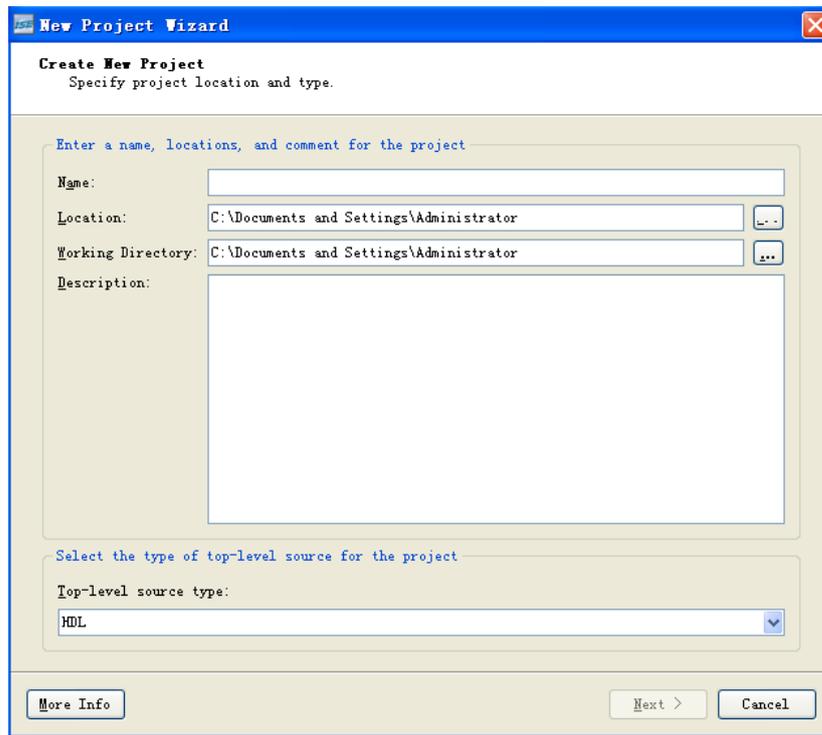
ISE 综合工具同时支持 VHDL 和 verilog 两种硬件描述语言的综合。它们的仿真步骤基本一致。在此只给出 verilog 的仿真步骤。

### 1) 打开 ISE 工具，显示如下界面:

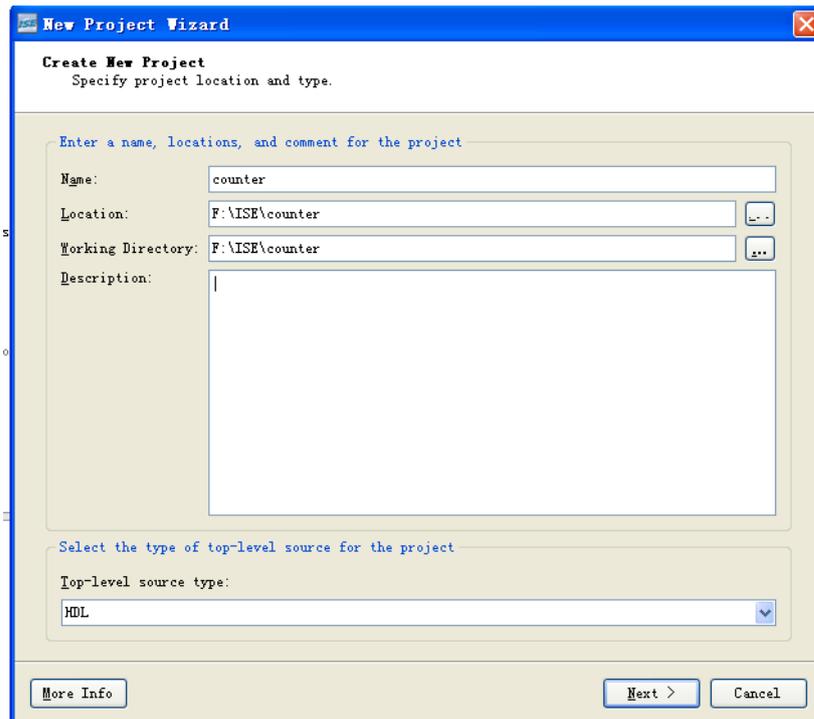


### 2) 创建工程

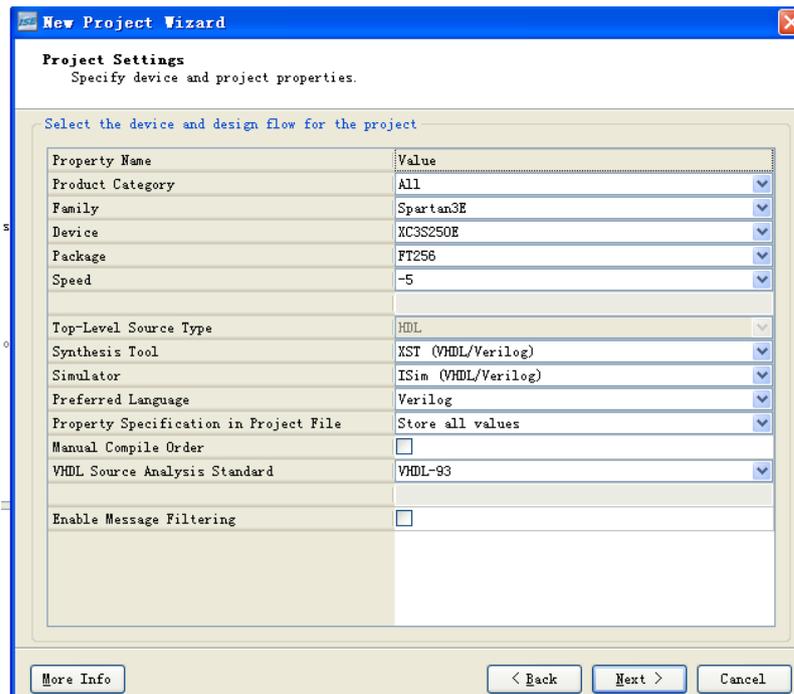
点击 File->new->project, 弹出下面窗口:



选择工程路径，点击 Next。弹出下面窗口:



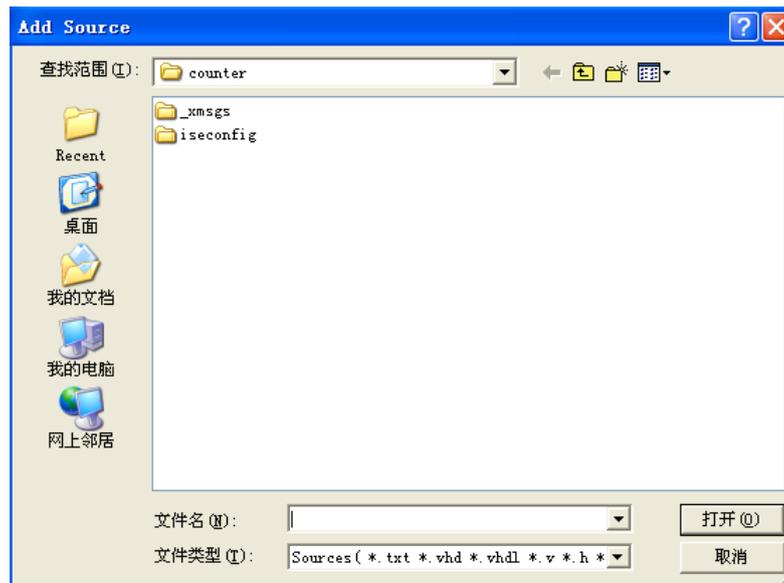
点击 Next。弹出下面窗口：



选择 FPGA 芯片类型，点击 Next。弹出下面窗口：

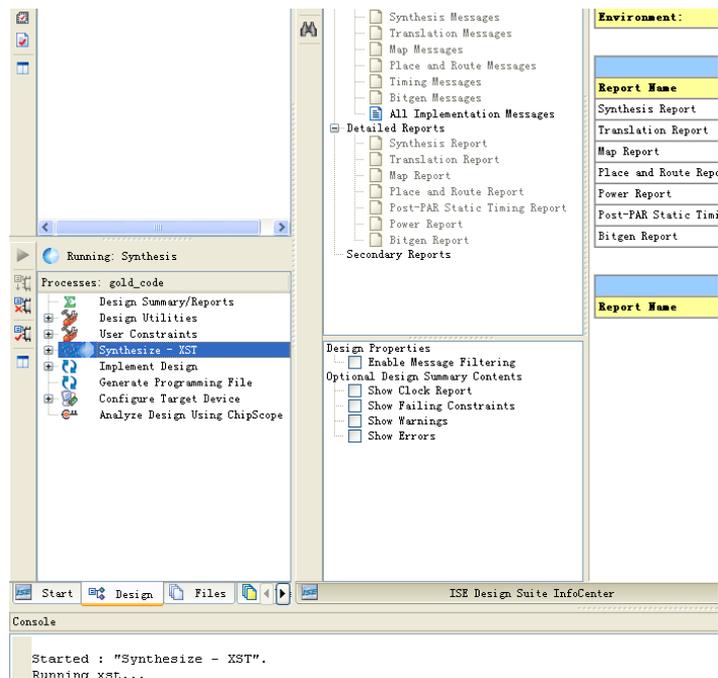
### 3) 添加设计

点击 Add Source 添加设计和约束文件。



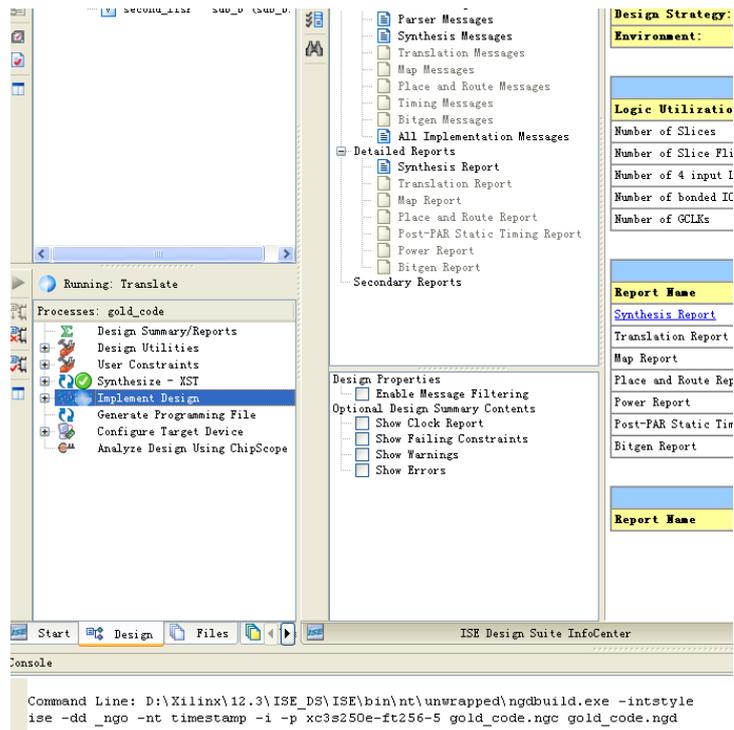
#### 4) 设计综合

点击 Synthesis-XST, 弹出下面窗口:



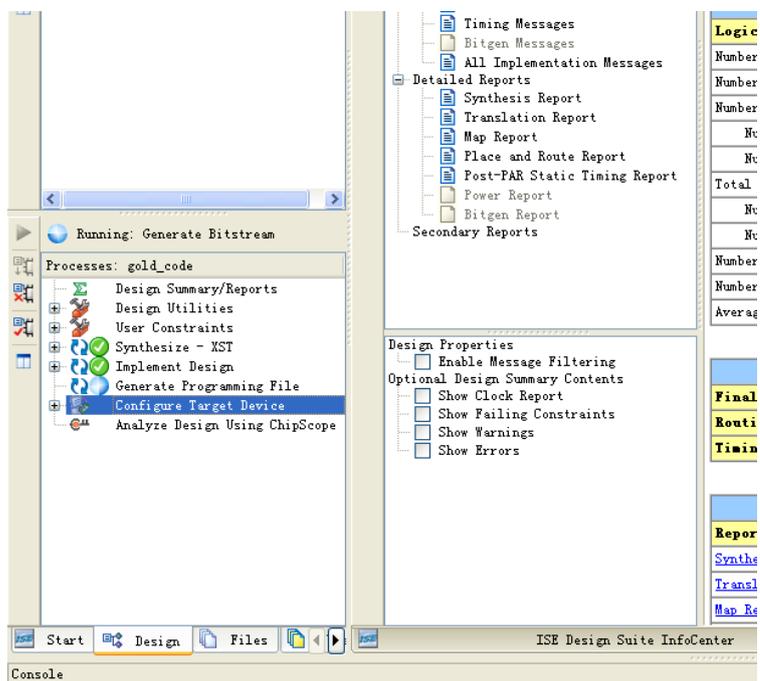
#### 5) 设计实现

点击 Implement Design, 弹出下面窗口:



## 6) 设计下载

点击 Configure Target Device, 弹出下面窗口:



连接 FPGA 开发板, 并下载比特文件。

## 五、 撰写报告

按附录要求撰写总结报告。

## 实验四 约翰逊计数器

### 一、实验目的

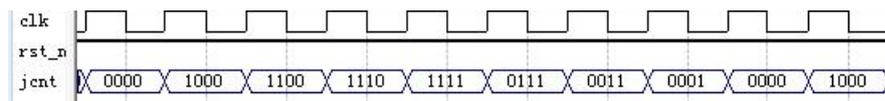
- 1、掌握时序电路描述的方法；
- 2、掌握时序电路仿真程序编写技巧；
- 3、理解约翰逊计数器的工作原理。

### 二、实验要求

- 1、设计可综合的约翰逊计数器；
- 2、编写电路仿真程序，要求测试输入的所有组合情况；
- 3、观察仿真结果。

### 三、实验原理

约翰逊(Johnson)计数器又称扭环计数器,是一种用  $n$  位触发器来表示  $2n$  个状态的计数器。它与环形计数器不同,后者用  $n$  位触发器仅可表示  $n$  个状态。 $n$  位二进制计数器( $n$  为触发器的个数)有  $2^n$  个状态。若以四位二进制计数器为例,它可表示 16 个状态。其接口时序如下图所示:



### 四、实验内容及步骤

- 1、根据约翰逊计数器的工作原理,设计并绘制其结构图;
- 2、根据原理图,使用 verilog HDL 语言,进行行为级描述;
- 3、编写电路的仿真激励,要求仿真所有计数情况,观察仿真结果。

### 五、撰写报告

按附录要求撰写总结报告。

## 实验五 序列检测器设计

### 一、实验目的

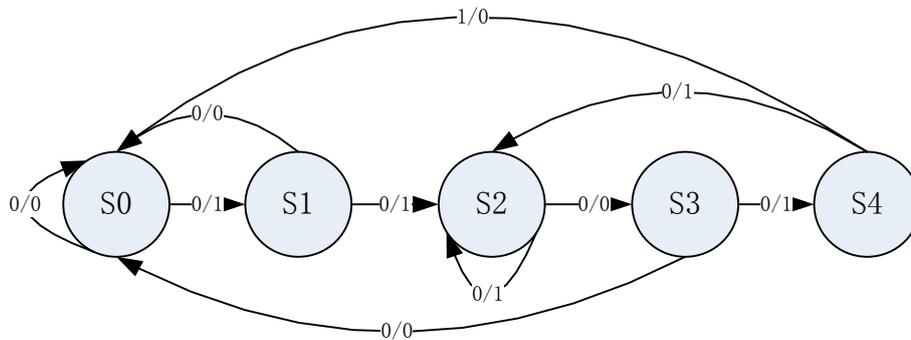
- 1、掌握时序电路描述的方法；
- 2、掌握时序电路仿真程序编写技巧。

### 二、实验要求

- 1、设计可综合的序列检测器；
- 2、编写电路仿真程序，要求测试输入的所有组合情况；
- 3、观察仿真结果。

### 三、实验原理

本设计采用状态机的设计方法，完成“1101”序列检测的功能。其状态转移图如下所示：



### 四、实验内容及步骤

1. 根据序列检测器的工作原理，设计并绘制状态转移图；
2. 根据状态转移图，使用 verilog HDL 语言，进行行为级描述；
3. 编写电路的仿真激励，要求给出检测序列的所有组合情况，观察仿真结果。

### 五、撰写报告

按附录要求撰写总结报告

## 实验六 任务与函数实验

### 一 实验目的

- 1、掌握任务和函数的使用方法；
- 2、理解定义任务所需的条件，学会任务的声明和调用。
- 3、理解定义函数所需的条件，学会函数的声明和调用。

### 二 实验要求

- 1、用函数设计 8 位 ALU 功能模型；
- 2、用任务设计一个可以对 16 位数进行按位与、或和异或操作的功能模型；
- 3、观察仿真结果。

### 三 实验原理

task 和 function 说明语句分别用来定义任务和函数。利用任务和函数可以把一个很大的程序模块分解成许多较小的任务和函数便于理解和调试。输入、输出和总线信号的值可以传入、传出任务和函数。任务和函数往往还是大的程序模块中在不同地点多次用到的相同的程序段。

任务和函数不同点主要有如下四条：

- 1) 函数只能与主模块共用同一个仿真时间单位，而任务可以定义自己的仿真时间单位。
- 2) 函数不能启动任务，而任务能启动其它任务和函数。
- 3) 函数至少要有一个输入变量，而任务可以没有或有多个任何类型的变量。
- 4) 函数返回一个值，而任务则不返回值。

函数的目的是通过返回一个值来响应输入信号的值。任务却能支持多种目的，能计算多个结果值，这些结果值只能通过被调用的任务的输出或总线端口送出。Verilog HDL 模块使用函数时是把它当作表达式中的操作符，这个操作的结果值就是这个函数的返回值。

### 四 实验内容及步骤

1. 根据任务的工作原理，设计 16 位数据位操作的功能单元；
2. 根据函数的工作原理，设计 8 位 ALU 功能模型；
3. 编写电路的仿真激励，要求给出所有组合情况，观察仿真结果。

### 五 撰写报告

按附录要求撰写总结报告

## 实验七：总线功能模型实验

### 一 实验目的

- 1、掌握总线功能模型的方法思想；
- 2、掌握总线功能模型的程序编写技巧。

### 二 实验要求

- 1、用总线功能模型的方法设计一个 CPU 接口模型；
- 2、观察仿真结果。

### 三 实验原理

Verilog 模块之间的连接是通过模块端口进行的。为了给组成设计的各个模块定义端口，我们必须对期望的硬件设计有一个详细的认识。不幸的是，在设计的前期，我们很难把握设计的细节。而且，一旦模块的端口定义完成后，我们也很难改变端口的配置。另外，一个设计中的许多模块往往具有相同的端口定义，在 Verilog 中，我们必须在每个模块中进行相同的定义，这为我们增加了无谓的工作量。Verilog 提供了一个新的、高层抽象的模块连接，这个连接被称为接口（Interface）。接口在模块中就像一个单一的端口一样使用。在最简单的形式下，一个接口可以认为是一组线网。通过使用接口，我们在进行一个设计的时候可以不需要首先建立各个模块间的互连。随着设计的深入，各个设计细节也会变得越来越清晰，而接口内的信号也会很容易地表示出来。当接口发生变化时，这些变化也会在使用该接口的所有模块中反映出来，而无需更改每一个模块。

### 四 实验内容及步骤

1. 根据总线功能模型原理，设计基于英特尔 CPU 的总线接口；
2. 编写电路的仿真激励，要求给出所有组合情况，观察仿真结果。

### 五 撰写报告

按附录要求撰写总结报告

## 开放性实验

### 一 实验目的

通过以小组的形式，完成一个较大的数字集成电路设计、仿真和验证。  
提高学生的设计能力和团队合作意识。

### 二 实验要求

二到四个学生一组，完成一个较大的数字集成电路设计、仿真和验证工作。要求在 FPGA 开发板上实现该设计，完成总体设计方案、详细设计方案、验证方案和设计总结报告的编写工作。

### 三 实验题目如下：

基于 FPGA 的蓝牙时钟设计  
基于 spartan\_3e 开发板的打地鼠游戏  
智能电梯  
自动售货机的设计  
基于 FPGA 的篮球计分器  
出租车计价器  
电子密码锁设计  
基于串行通信的语音传输  
交通灯控制逻辑电路设计  
基于遥控和自控的寻迹小车  
音乐播放器  
基于 fpga 的公交报站系统  
基于 FPGA 的贪吃蛇设计  
基于 FPGA 的机械精密控制系统  
基于 FPGA 的小游戏——爱消除  
多功能数字钟设计  
带密码锁的拔河比赛  
基于 FPGA 的 DDS 设计