

## 第 3 章

## CHAPTER

## 存储设备实验

本章包括三个实验,分别是移位寄存器设计、程序计数器设计和随机存储器设计。

### 3.1 移位寄存器实验

#### 3.1.1 实验题目

左移 4 位寄存器设计与验证。

#### 3.1.2 实验内容

首先设计一位可控缓冲寄存器,然后用其组装可使数据左移 4 位的寄存器。

#### 3.1.3 实验目的与要求

本实验以左移 4 位寄存器为例说明移位寄存器的组成和工作原理。要求理解左移位寄存器的内部构造,并仿照左移位寄存器的设计方法设计出右移位寄存器。

#### 3.1.4 实验步骤

本实验的项目所在的路径为 D:\exam\test4,项目的名称为 MOVES,顶层文件的名称也为 MOVES。

##### 1. 设计一位可控缓冲寄存器 CONS

建立文件 CONS.bdf。设计中需要用到前缘 D 型触发器 dff,它位于 Libraries 栏中的 primitives 目录下的 storage 目录中。设计完成后的效果如图 3-1 所示。图中左下角部分为时钟脉冲输入引脚 Clk 和复位输入引脚 Clr。L 门逻辑电路可以控制触发器数据的输入:如果 L=1,当 Clk 由 0 变到 1 时,那么新数据进入触发器;如果 L=0,当 Clk 由 0 变到 1 时,那么触发器中原有的数据将会被保持。

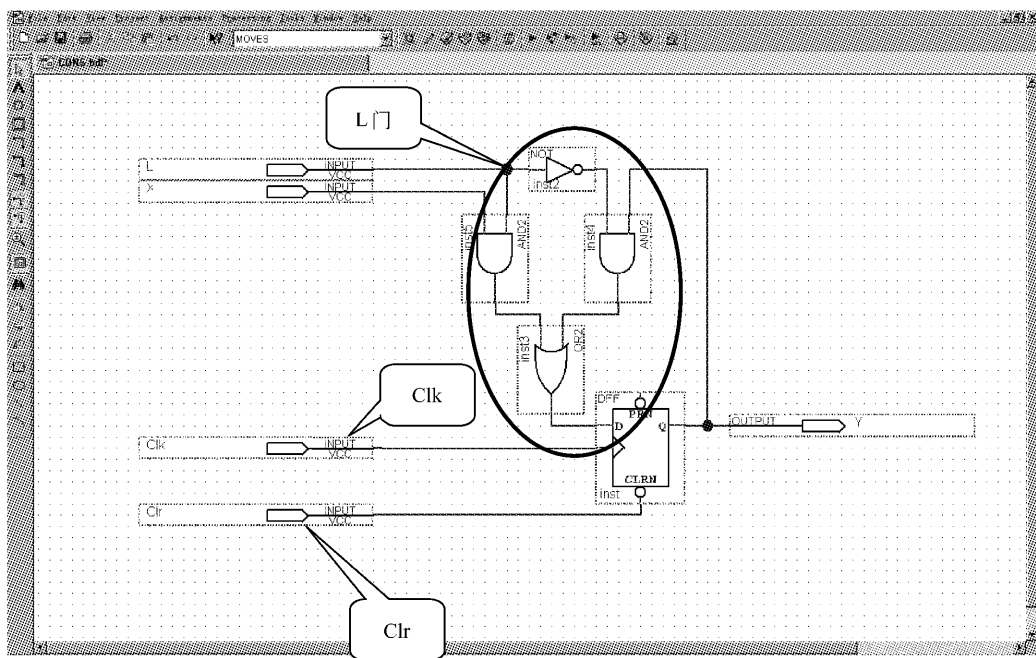


图 3-1 完成后的 CONS

将其封装为模块。

## 2. 设计左移 4 位寄存器 MOVES

建立文件 MOVES.bdf, 设计完成后的效果如图 3-2 所示。图中共用到 8 个模块 CONS,

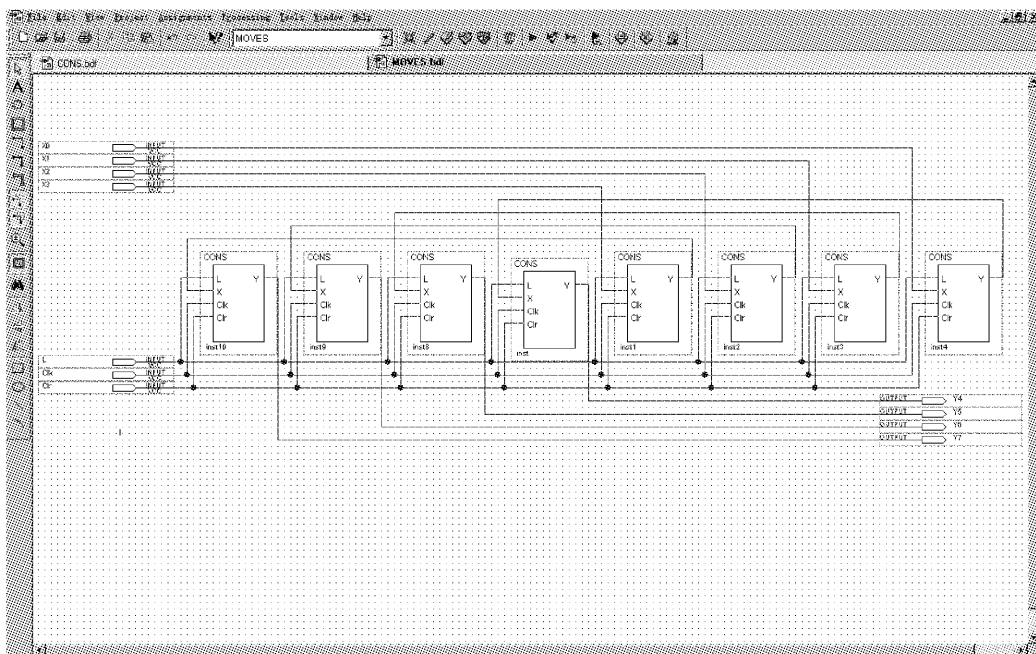


图 3-2 完成后的 MOVES

将低 4 位 CONS 的输入端 X 引出为输入引脚,将高 4 位 CONS 的输出端 Y 引出为输出引脚。

3. 进行功能仿真

编译工作完成后就可以进行功能仿真了。

建立波形文件 MOVES.vwf。Clr 的值在大部分时间内设为 1,仅在第一个时钟脉冲到来之前的一个时间段内为 0。由于触发器 dff 的复位端为后缘触发,所以 Clr 如此赋值便可令触发器 dff 在初始时复位。Clk 的赋值方法是:选中 Clk,然后选择绘图工具栏中的 Overwrite Clock 按钮,系统弹出 Clock 对话框,按照默认的时间设置,单击 OK 按钮即可。L 的值设为 1。Xi(i=0,1,2,3)为数据输入端,赋值为 0110。Yi(i=4,5,6,7)为数据输出端,值尚未确定。完成后的结果如图 3-3 所示。

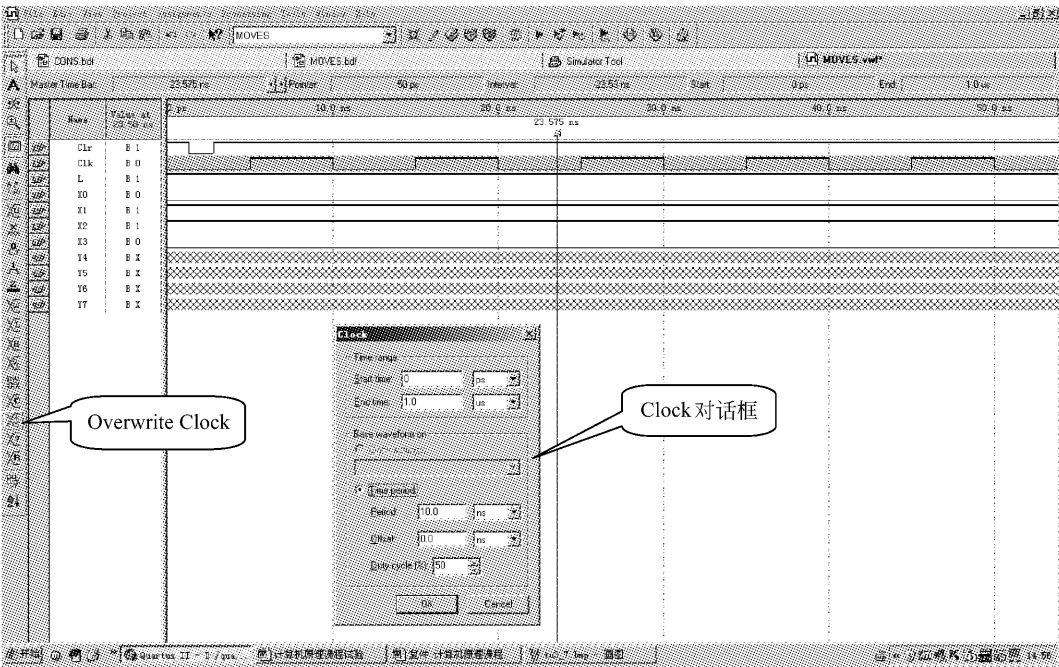


图 3-3 完成后的 MOVES.vwf

波形文件创建完成之后,将其作为功能仿真的对象,查看本实验设计的逻辑电路是否能够实现数据的 4 位左移。功能仿真的结果如图 3-4 所示。从图中可以看出 Yi 的值在第二个时钟脉冲到来时变为 0110,这说明数据被成功地左移了 4 位。

3.1.5 实验结论

将 L 门加在边缘 D 型触发器的前端可以实现数据的长时间保持。本实验设计的逻辑电路可以实现左移 4 位数据的功能,还可以对电路加以改动,实现右移 4 位数据的功能,具体操作方法请同学自行实验。

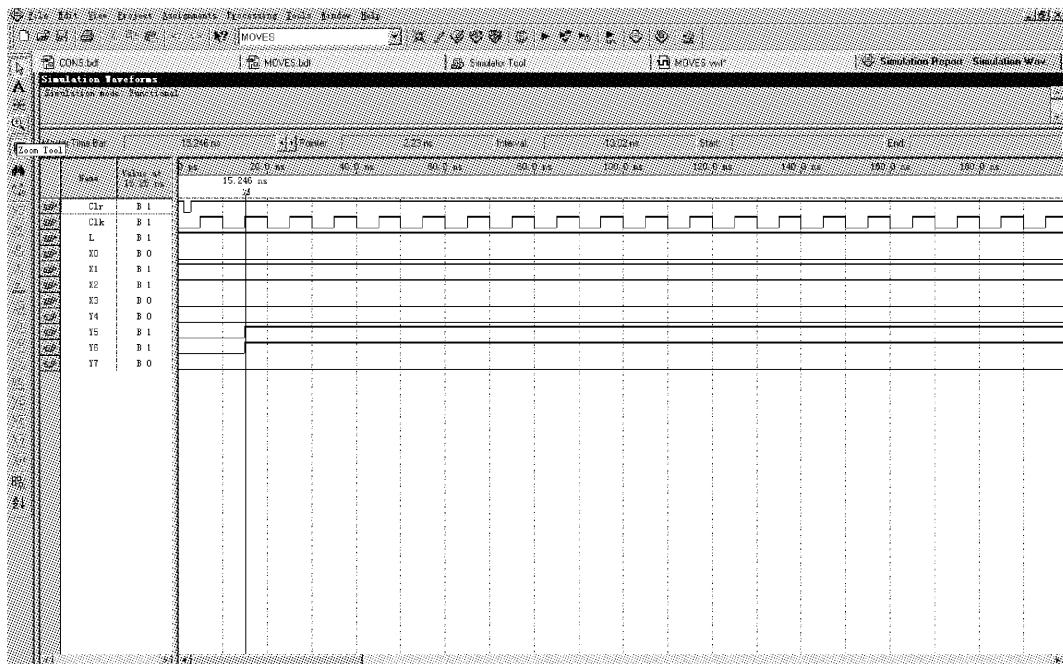


图 3-4 左移 4 位寄存器的功能仿真结果

## 3.2 程序计数器实验

### 3.2.1 实验题目

程序计数器设计与验证。

### 3.2.2 实验内容

设计 4 位程序计数器。

### 3.2.3 实验目的与要求

本实验通过对 4 位程序计数器的逻辑设计体现计数器用可控制的形式加 1 计数的原理。要求理解 JK 触发器的工作原理和程序计数器的计数方法。

### 3.2.4 实验步骤

本实验的项目所在的路径为 D:\exam\test5,项目的名称为 PC\_PRO,顶层文件的名称也为 PC\_PRO。

建立文件 PC\_PRO.bdf。设计中要实现从 0~15 的计数需要 4 个 JK 触发器 jkff,它位于 Libraries 栏中的 primitives 目录下的 storage 目录中。由于要观察程序计数器计数的过程,将 JK 触发器的输出端 Q 引出为输出引脚。设计完成后的效果如图 3-5 和图 3-6

所示。

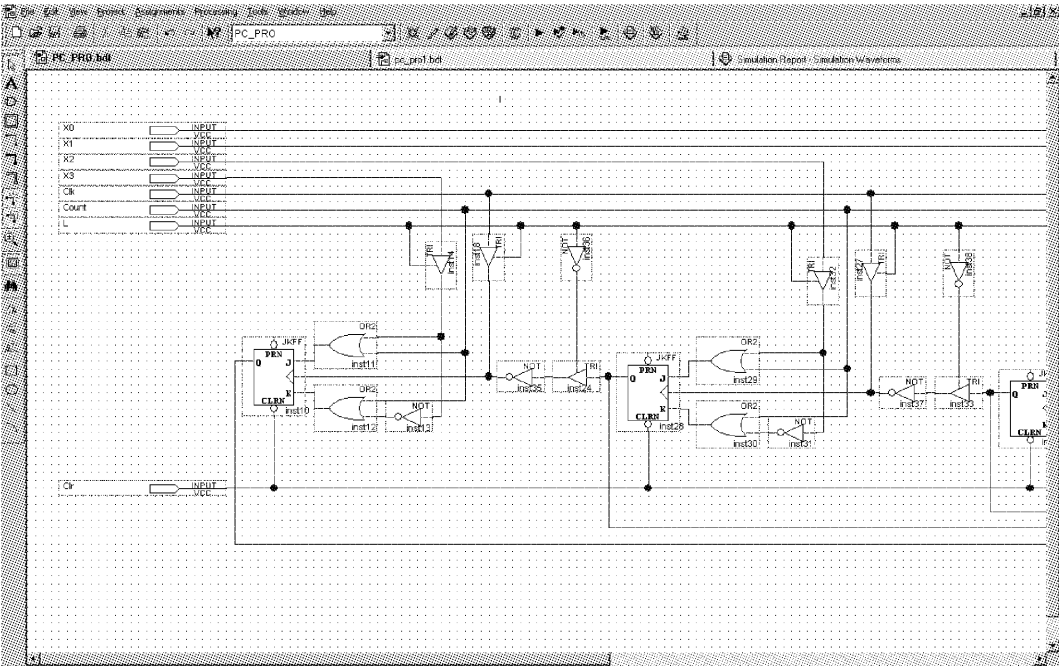


图 3-5 完成后的 PC\_PRO(一)

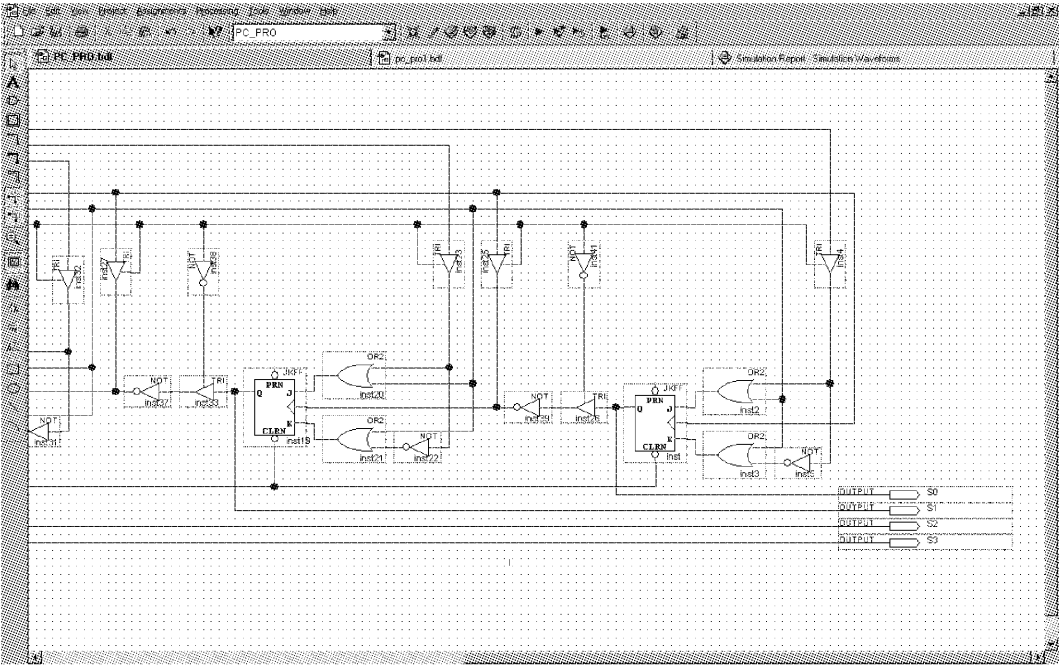


图 3-6 完成后的 PC\_PRO(二)

编译成功之后就可以在功能仿真中观察程序计数器的计数过程了。

建立波形文件 PC\_PRO.vwf。将 Count 赋值为 1, L 赋值为 0, 完成后的效果如图 3-7 所示。程序计数器将从 0 开始计数, 计至 15 后, 再加 1 将重新变为 0, 如此循环。

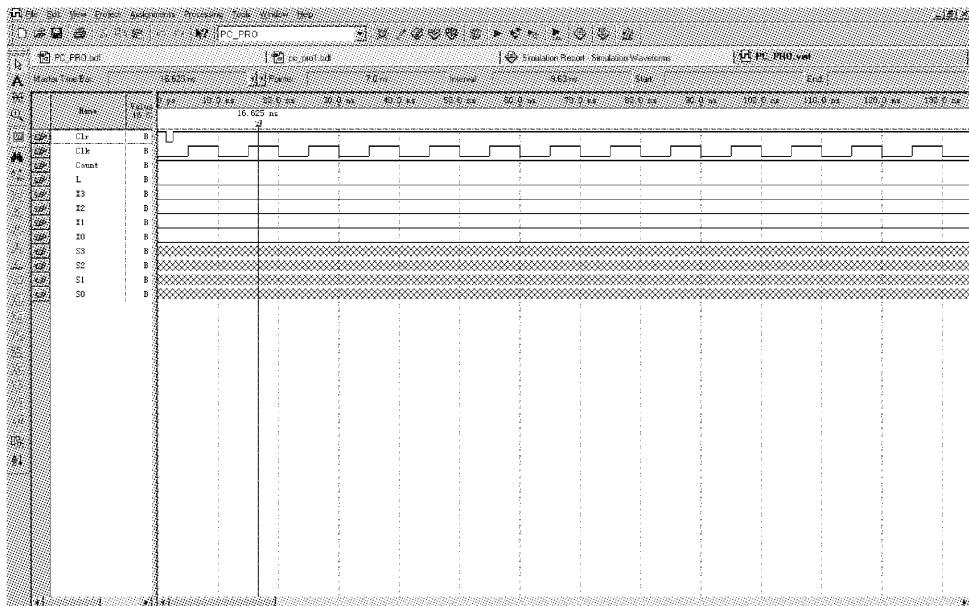


图 3-7 完成后的 PC\_PRO.vwf

波形文件创建完成之后, 对其进行功能仿真, 仿真结果如图 3-8 所示。从图中可以看

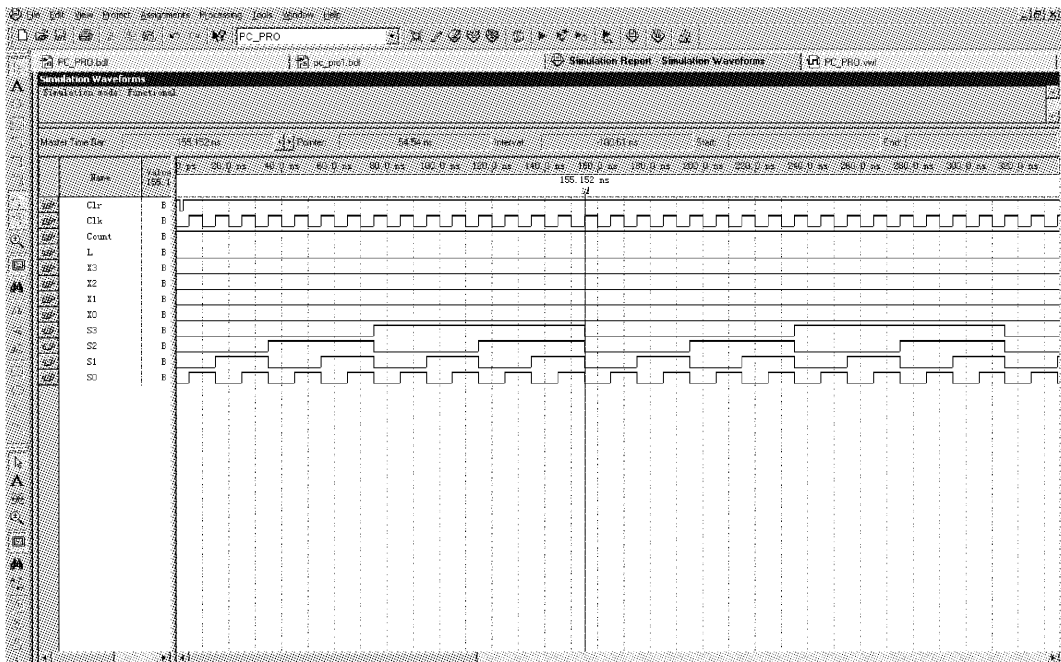


图 3-8 程序计数器的功能仿真结果

出,程序计数器成功地完成了计数过程。

功能仿真只检验逻辑功能,若考虑电路的延时情况则需进行时序仿真。现以本实验设计的程序计数器为例进行时序仿真。在仿真对话框的 Simulation mode 下拉列表中选择 Timing,即时序仿真,如图 3-9 所示。然后单击对话框左下角的 Start 按钮就可以进行时序仿真了。

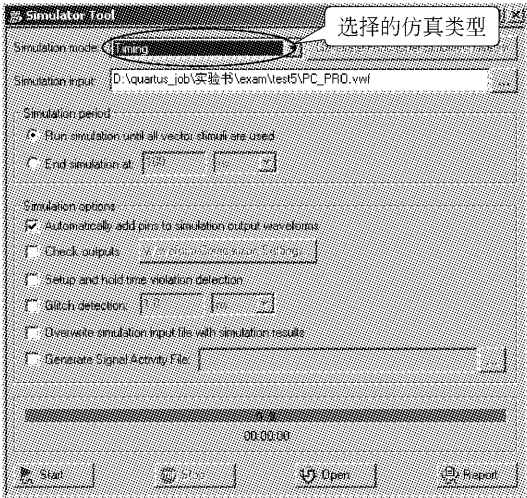


图 3-9 仿真窗口

时序仿真成功后,观察仿真结果,如图 3-10 所示。

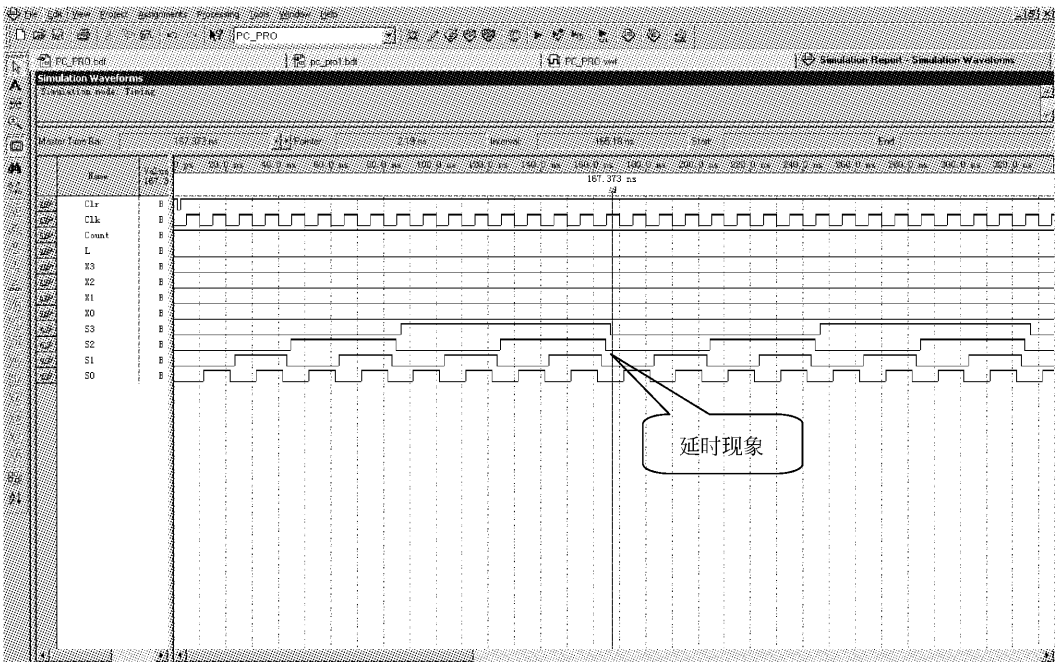


图 3-10 程序计数器的时序仿真结果

### 3.2.5 实验结论

本实验成功实现了程序计数器的计数过程,并且通过时序仿真观察到电路的延时现象。

## 3.3 随机存储器实验

### 3.3.1 实验题目

随机存储器设计与验证。

### 3.3.2 实验内容

首先设计带 E 门的 8 位可控缓冲寄存器,在此基础上设计随机存储单元,最后设计具有 8 个随机存储单元的 8 位随机存储器。

### 3.3.3 实验目的与要求

本实验的目的是使学生理解随机存储器是由大量的可控缓冲寄存器作为存储单元组成的,而存储单元的访问是通过编址的形式进行的,通过对电路的设计实现对数据的读操作和写操作。要求学生熟练掌握随机存储器的组成、工作原理以及使用方法。

### 3.3.4 实验步骤

本实验的项目所在的路径为 D:\exam\test6,项目的名称为 RAM8,顶层文件的名称也为 RAM8。

#### 1. 设计带 E 门的 8 位可控缓冲寄存器 CONSE

建立文件 CONSE.bdf。即将设计的带 E 门的 8 位可控缓冲寄存器可由实验 test4 中设计的一位可控缓冲寄存器改装而成。首先将输出引脚 output 改为既可输入又可输出的双向引脚 bidir,并改名为 Bus[7..0],表示数据位数为 8 位。bidir 位于 Libraries 栏中的 primitives 目录下的 pin 目录中。其次在触发器 dff 的输出端连接一个 E 门,控制是否向总线输出数据,然后将数据输入端连接到总线上。最后将数据线的类型改为 Bus Line,具体操作方法是:选中数据线,单击鼠标右键,选择 Bus Line 即可。操作中注意区分数据线和控制线。

设计完成后的效果如图 3-11 所示。如果 L=1,当 Clk 由 0 变到 1 时,那么数据从总线进入到寄存器;如果 E=1,则数据从寄存器输出到总线上。L 门和 E 门不能同时为 1,因为在同一时间内不能既读数据又写数据,但是可以同时为 0。当它们同时为 0 时,寄存器内的数据将被保持。为了便于在仿真中观察结果,引出输出引脚 out[7..0],并将其封装为模块。

#### 2. 设计随机存储单元 em\_RAM8

随机存储单元是在带 E 门的 8 位可控缓冲寄存器基础上增加 IO 线和 AD 线。IO 线

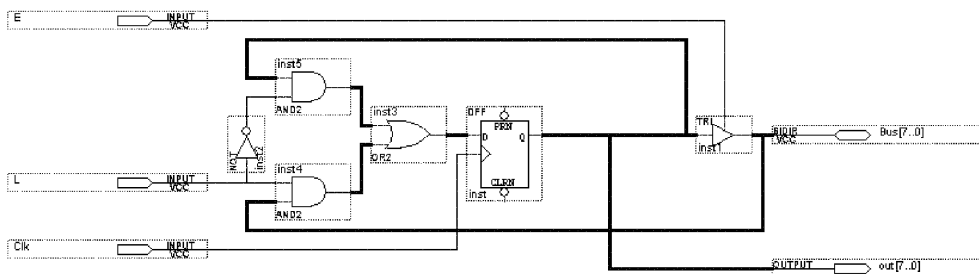


图 3-11 完成后的 CONSE

的作用是控制对存储单元的读操作和写操作,AD 线决定存储单元是否被选通。

建立文件 em\_RAM8. bdf, 利用模块 CONSE 进行设计。完成后的效果如图 3-12 所示。

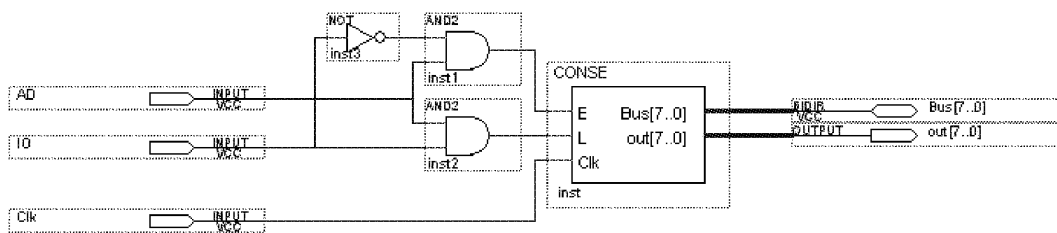


图 3-12 完成后的 em\_RAM8

将其封装为模块。

### 3. 设计具有 8 个存储单元的 8 位随机存储器 RAM8

建立文件 RAM8. bdf。设计中需要用到 8 个模块 em\_RAM8, 将它们的 IO 端和 Clk 端分别连接在一起并引出输入引脚。用  $A_2$ 、 $A_1$ 、 $A_0$  这三条线构建地址译码器, 不同的赋值决定着不同的随机存储单元被选通。每一个 em\_RAM8 的选通线都连接一个 E 门, 将 E 门的控制端连接在一起引出为输入引脚  $M_e$ 。为了向总线灌输数据, 在总线上连接输入引脚 in[7..0]。设计完成后的效果如图 3-13 所示。

### 4. 进行功能仿真

编译成功之后, 建立波形文件 RAM8. vwf, 如图 3-14 所示。图中  $A_2$ 、 $A_1$ 、 $A_0$  的值从 000 递增至 111, in 的值从 00000001 递增至 00001000。可利用波形编辑工具栏中的 Count Value 按钮  进行赋值。令 IO=1,  $M_e$ =1。out0 至 out7 是从 0~7 号存储单元中的数据, 值尚未确定。Bus 的值为 zzzzzzzz, 代表处于高阻状态。

建立波形文件后, 进行功能仿真。仿真后的结果如图 3-15 所示。从图中可以看出, 第一个时钟脉冲到来时, out0 的值由 00000000 变为 00000001。这是由于 IO 和  $M_e$  的值均为 1 且 0 号存储单元被选通, 数据从总线流入 0 号存储单元。此后, 随着时钟的震荡, 数据依次装入 1~7 号存储单元。

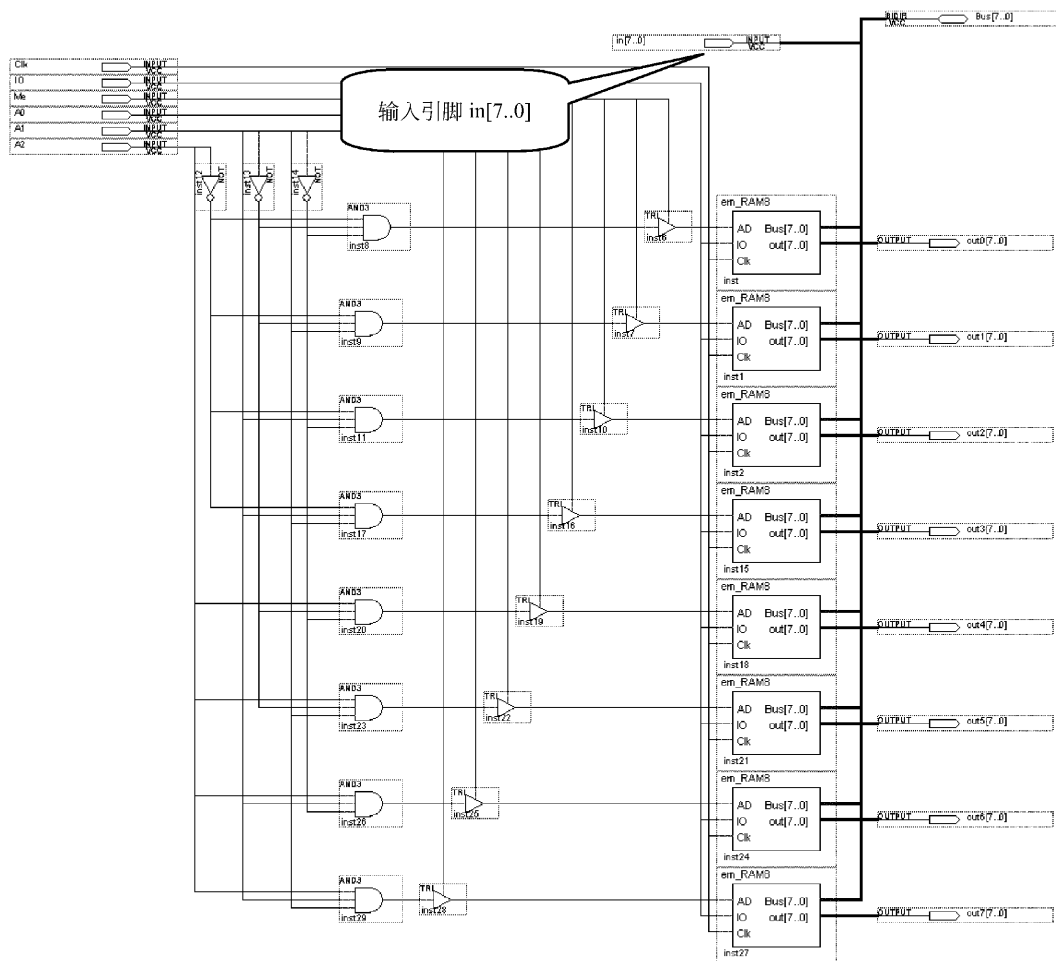


图 3-13 完成后的 RAM8

### 3.3.5 实验结论

本实验向学生展示了随机存储器的工作原理。通过地址译码器可以选择不同的存储单元,如果将地址译码器的位数扩充至 8 位,将会使 RAM 的存储空间扩充至 256B,即具有 256 个存储单元。如果设计多个存储空间为 256B 的随机存储器,再将它们的控制线  $M_0$  用另一个地址译码器来选择,那么将会更大地扩展存储空间。