

实验3 序列检测器设计

Advanced Electronics System Lab, 202309

一、实验目的

- 1、熟悉并掌握状态机的原理与设计；
- 2、熟悉并掌握信号发生器的原理与设计；
- 3、熟悉并掌握序列检测器的原理与设计；
- 4、熟悉并掌握 Verilog 中模块例化设计方法；
- 5、进一步掌握 Altera FPGA 设计的流程和步骤。

二、工作原理

序列检测器用途很广，在数据通讯、雷达和遥测等领域中常用于检测同步识别标志。序列检测器主要是用一种跳转图来检测一组或多组序列信号。如果收到的序列中包含需要检测的序列，例如“111010011”，则输出标志为“1”，否则输出为“0”。

要检测串行码“111010011”，首先要有被测码，也就是说要有序列产生器。有了序列产生器以后，就可以用序列检测器检测序列产生器的输出。如果序列产生器的序列包含被测序列，则序列检测器输出状态“1”，无被测序列则输出状态“0”或无输出。本实验就是要实现序列产生器和序列检测器，其结构如图1所示。

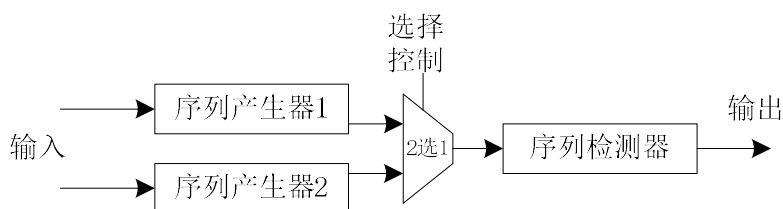


图1 实验结构图

图1包含两个序列产生器、一个选择器和一个序列检测器。两个序列产生器中的一个产生含有“111010011”的序列，另一个不包含。通过选择器选一路序列送入序列检测器，如果输入的序列包含“111010011”，则序列检测器输出状态“1”，表示检测到被测序列，否则序列检测器输出状态“0”，表示没有检测到被测序列。

三、实验步骤

1、编程设计两个序列产生器

序列的产生可以看作是一个有限状态机（参考第5章有限状态机部分），它

有 N 个状态，除节拍时钟和复位信号以外，不需要其他输入，但每个状态都有一个输出，可以相同也可以不同。例如，一个包含“110001”的序列产生器的状态转换图如图 2 所示。

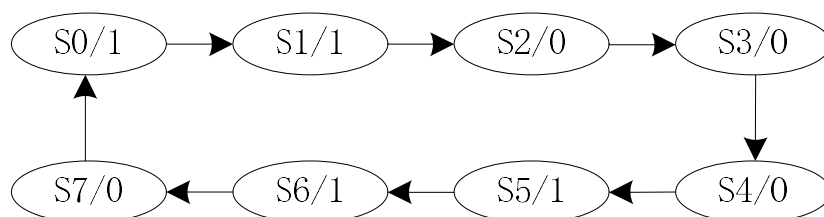


图 2 包含“110001”序列的序列产生器状态转换图

图 2 所示的状态转换图是一个没有输入信号的封闭状态转换图(时钟和复位信号除外)，当其处于其中一个状态时，下一时刻会强制转换到下一个状态。图中的“ S_x/x ”表示当前状态为“ S_x ”，当前输出为“ x ”。那么图 2 所产生的序列就是形如“1100011011000110110001...”的序列，其中包含了“110001”序列。

改写图 2 所示的序列产生转换图，用 Verilog 编程方式设计一个包含“111010011”的序列产生器，再设计一个不包含“111010011”的序列产生器。

2、编程设计一个序列检测器

序列检测器也可以看作是一个状态机，它的输入是一个序列，输出是检测到特定序列与否的标志信号。这里序列检测器要检测“111010011”序列，检测“111010011”序列的状态转换图如图 3 所示。

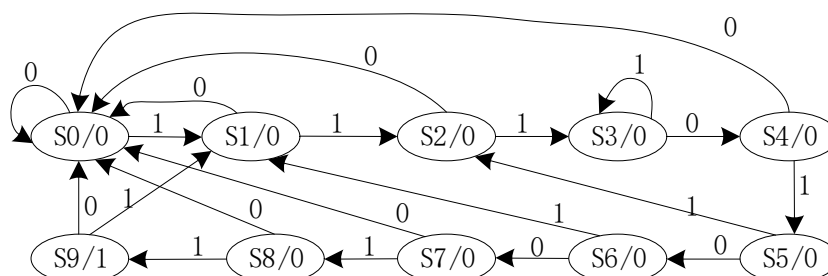


图 3 检测“111010011”序列的状态转换图

根据图 3 所示的状态转换图，用 Verilog 编写检测“111010011”的序列检测器。

3、编写 Test Bench 程序并成功能仿真

针对以上设计的序列产生器、序列检测器，编写 Verilog Test Bench 进行仿真。需要注意的是，在编写包含时钟端口的 Test Bench 程序时，时钟信号要单独产生，其它的输入信号另外产生。比如：

```
initial clk = 0;
always #1 clk = ~clk; //产生时钟信号
```

```

.....
initial begin //产生其他信号
    #0 a = 0;
    #2 a = 1;
    #2 a = 0;
    #1 $stop;
end

```

若时间单位为 10ns，以上语句产生了周期为 20 ns 的时钟信号，先 10ns 低电平，后 10ns 高电平，然后不断地反复。

在仿真结束后，通过“Tools”菜单”Netlist Viewers”子菜单下的“RTL Viewer”命令查看设计电路的 RTL 级的电路结构。

4、为设计工程分配管脚

通过查阅“实验指导书 altera_cyclone5.pdf”文档中的“附件一 FPGA 管脚约束”，分别为设计中的输入和输出选定管脚并添加到 Quartus II 工程。

5、完成硬件验证

分配管脚之后，对设计文件进行全编译，生成配置文件，并将配置文件配置到 FPGA 中。观察 LED 灯，查看结果是否正确，并通过切换输入序列验证序列检测器是否正确。

四、结束

至此，我们完成了 Altera FPGA 设计的基本流程和步骤。如果你结束了今天的实验，请备份实验数据和实验记录，关闭所有打开的软件并关闭计算机；关闭实验箱的电源，并拆除所有相关的连线，恢复到实验之前的整齐样子，别留下任何物品。O(∩_∩)O 谢谢！



