**实验二：序列检测器**

丁文浩 无43 2014011079

**一、实验目的**

1.掌握有限状态机的实现原理和方法；

2.掌握序列检测的方法

**二、设计方案**

**1.使用有限状态机实现序列检测器**

（1）原理

有限状态机（ Finite State Machine, FSM）是逻辑电路设计中经常要遇到的，在数字电路中， 经常需要通过建立有限状态机的方式来进行时序数字逻辑的设计。在复杂数字系统设计中，有限状态机主要通过硬件描述语言实现，硬件描述语言能够清晰的描述状态转移过程和输入输出变量关系，使得时序逻辑设计大大简化，进而极大降低系统设计复杂度，提高系统模块化程度。

有限状态机从本质上讲是由寄存器和组合逻辑构成的时序电路，各个状态之间的转移总是在时钟的触发下进行的。可以通过建立原始状态表和状态化简来设计电路。

本次实验的内容是在连续输入的串行数据流中检测特定序列“ 101011”，一旦检测到一个“ 101011”就输出一个宽度为1个时钟周期的高电平脉冲。

首先确认由于该序列的长度为6，所以在加上初始状态后需要记录的状态数为6 + 1 = 7（init、stateA、stateB、stateC、stateD、stateE、stateF）。通过分析需要检测的序列“101011”发现，在以下几种情况下该序列可能发生重复使用的情况：

a.最后一个1作为下一个序列的开头

b.最后一位为0时可以作为新的序列的第二位开始使用

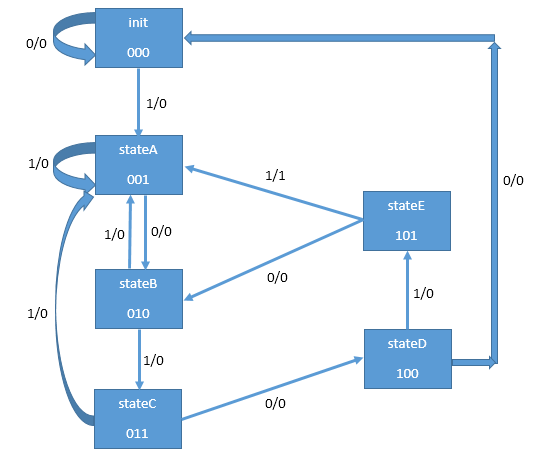
c.每一次应该为0的时候如果输入1都可以作为开头第一个使用

由此画出状态转移图，并且发现状态F和状态A由于次态和输出是一样的，故可以合并。

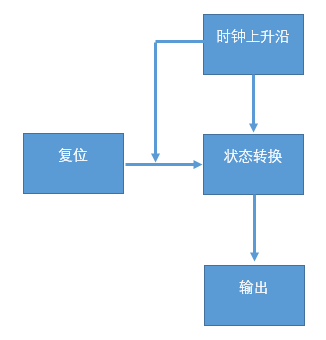
对这6个状态使用3Bit进行编码，并且用3个LED作为指示输出。

如果检测到指定序列，则指定的LED将会在下一个时钟的上升沿点亮。

（2）化简后的状态转移图

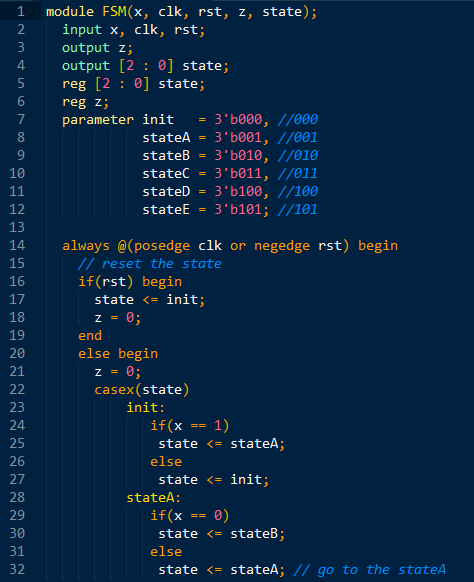


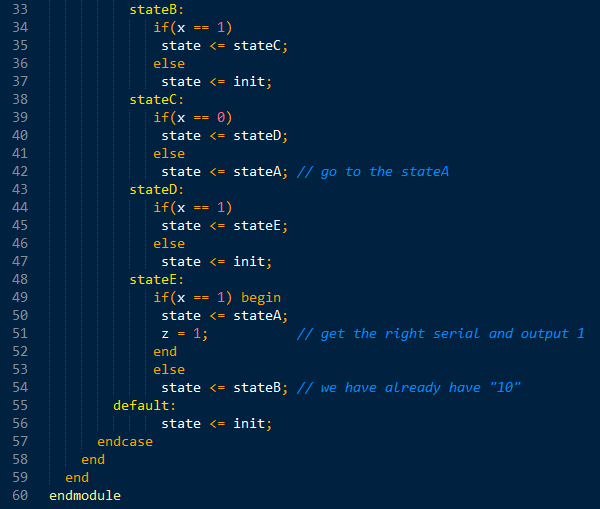
流程图：



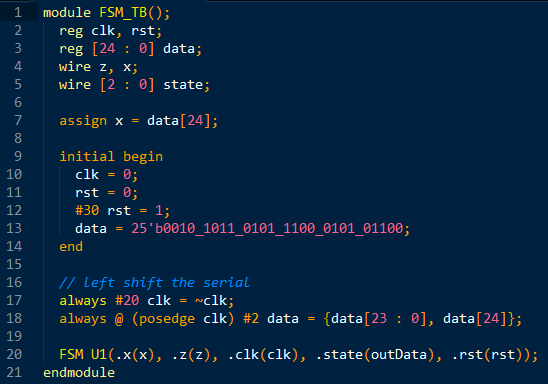
（3）关键代码

有限状态机模块代码：

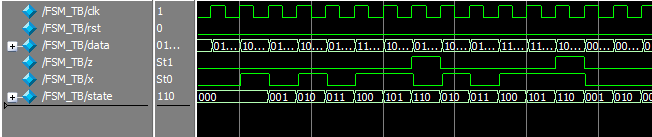


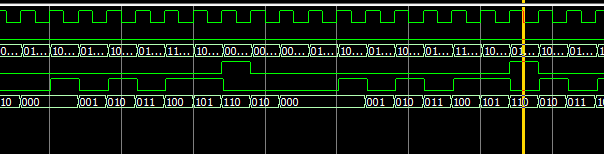


testbench代码：



（4）仿真波形





由上述仿真波形可以看出在出现“101011”这个序列之后下一个时钟上升沿z就会输出高电平，并且在红框表示的区域可以看到两个重叠的序列可以同时检测出来。

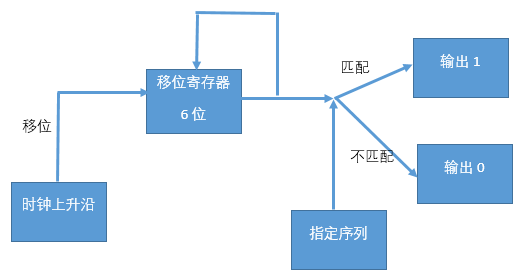
**2.移位寄存器实现序列检测器**

（1）原理

由于该序列有6个bit，所以需要6个寄存器同时输出6个数据来判断当前的数据是否是符合标准的，在下一次时钟上升沿的时候进行移位，把所有的数据向右移动，并且添加一个新的数据继续进行判断。

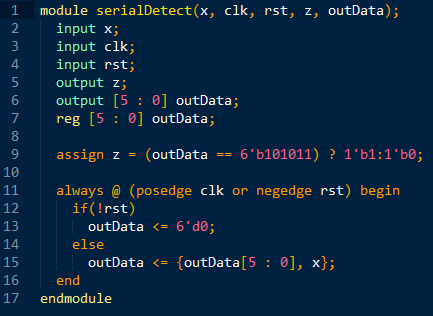
通过6个LED来指示现在的数据输出。

（2）流程图

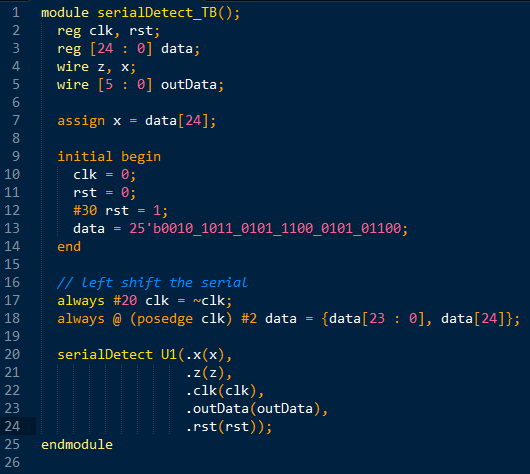


（3）关键代码

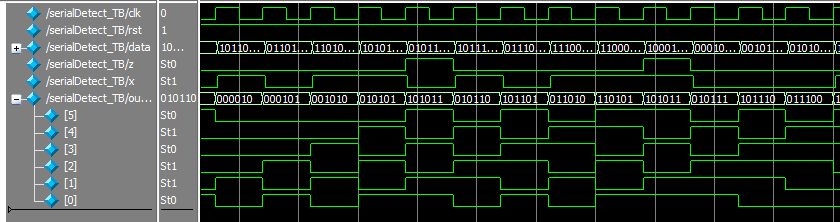
移位寄存器模块代码：



testbench代码



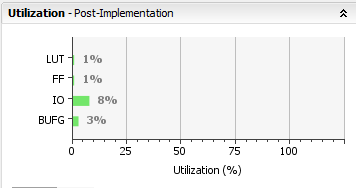
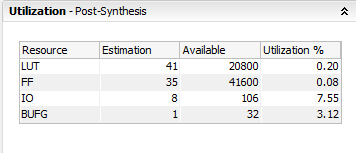
（4）仿真波形



由上述仿真波形可以看出在出现“101011”这个序列之后下一个时钟上升沿z就会输出高电平，并且在红框表示的区域可以看到两个重叠的序列可以同时检测出来。

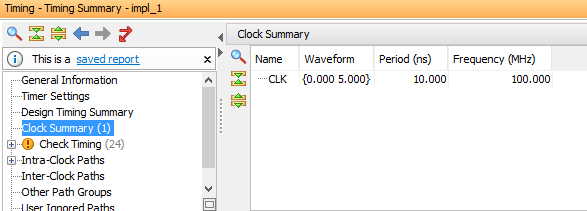
**三、综合情况（有限状态机方案）**

1.面积报告

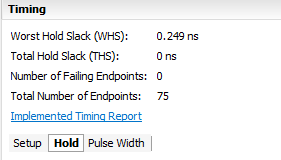
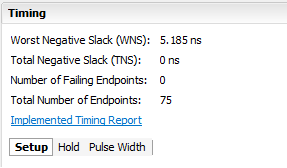
 

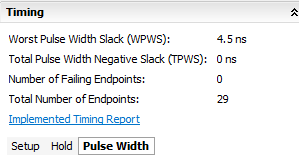
2.时序性能

（1）时钟分析



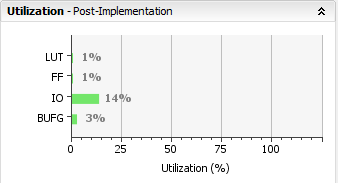
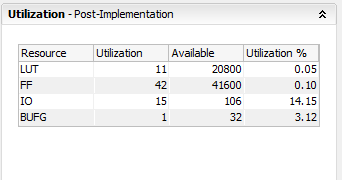
（2）建立时间与保持时间和脉宽分析





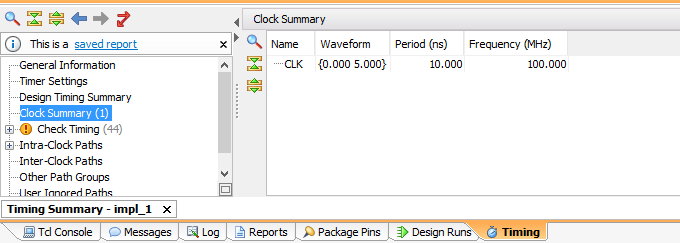
**四、综合情况（移位寄存器方案）**

1.面积报告

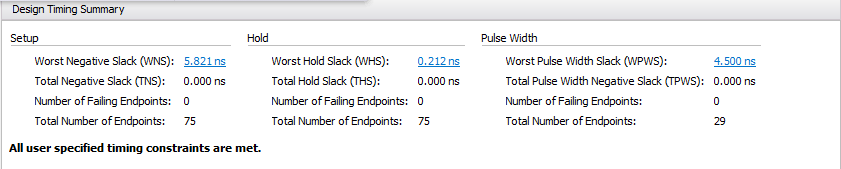


2.时序性能

（1）时钟分析

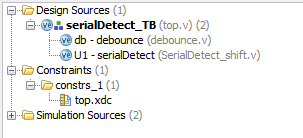


（2）建立时间与保持时间和脉宽分析



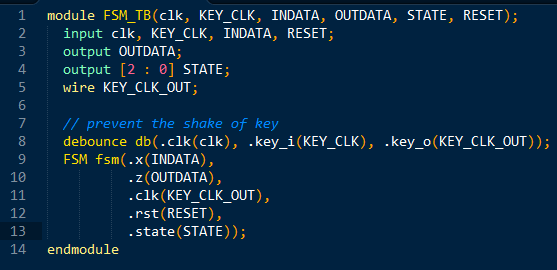
**五、实验总结**

1.代码清单

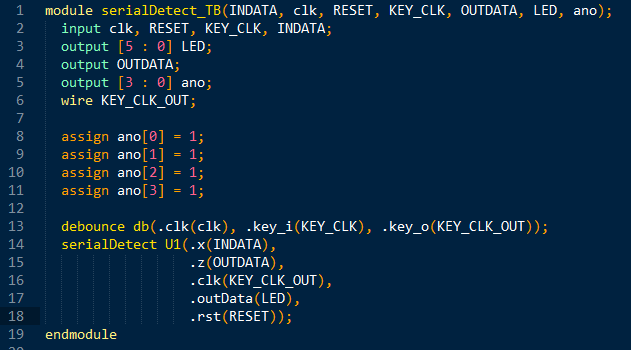


左图为有限状态机的文件列表，右图为移位寄存器的文件列表

有限状态机的top文件



移位寄存器的top文件



2.现场验收

现场验收了移位寄存器的效果，并且助教查看了有限状态机的状态转换图。

3.实验总结与体会

（1）一开始对于复位的电平没有看清楚，因为第一个实验是使用的开关作为复位的触发，但是第二个实验是使用按键作为复位，所以应该是高电平有效。一开始的时候直接复制的上一次的代码，造成了一直在复位没有输出的效果。

（2）有限状态机一开始使用了7个状态，后来发现最后一个状态与第二个状态的次态和输出是相同的，故可以合并成一个状态，最终使用了6个状态。

（3）通过本次实验，深入了解了有限状态机的构建过程。