|  |  |
| --- | --- |
| logo[1] | 杭州电子科技大学 |

实验报告

Experimental report

|  |  |
| --- | --- |
| **实验课程名称** | **数字系统与处理器** |
| **实验序号** | 5 |
| **实验内容** | 序列检测器设计 |
| **班级** | 18184111 |
| **姓名** | 杨臻 |
| **学号** | 18052032 |
| **指导教师** | 黄继业 |

二○一九 年 十一月 十八日

## 实验目的与要求

实验名称：序列检测器设计

## 实验原理

给定一段二进制序列，与程序内保存的二进制序列进行比对，若不符合则持续输出0，若符合则输出1然后进入下一轮判断

## 实验内容

实验步骤：

1. 建立Vivado工程文件
2. 编写源程序
3. 编写仿真文件
4. 执行仿真
5. 修改仿真文件
6. 执行二次仿真

关键源程序：

主程序：

module schk (input CLK ,DIN, RST, output SOUT);

parameter s0=40,s1=41, s2=42, s3=43, s4=44, s5=45, s6=46, s7=47, s8=48;

reg[8:0] ST,NST;

always @(posedge CLK or posedge RST)

if (RST) ST<=s0; else ST<=NST;

always @(ST or DIN)

begin

case (ST) //DIN保存预置序列

s0 : if (DIN==1'b1) NST<=s1; else NST<=s0; //NST保存当前状态

s1 : if (DIN==1'b1) NST<=s2; else NST<=s0; //NST根据序列是否符合赋值

s2 : if (DIN==1'b1) NST<=s3; else NST<=s0;

s3 : if (DIN==1'b0) NST<=s4; else NST<=s3;

s4 : if (DIN==1'b1) NST<=s5; else NST<=s0;

s5 : if (DIN==1'b0) NST<=s6; else NST<=s2;

s6 : if (DIN==1'b0) NST<=s7; else NST<=s1;

s7 : if (DIN==1'b0) NST<=s8; else NST<=s1;

s8 : if (DIN==1'b0) NST<=s0; else NST<=s1;

default : NST<=s0;

endcase

end

assign SOUT=(ST==s8);

endmodule

仿真程序：

`timescale 10ns / 100ps

module schk\_tb();

reg CLK;

reg DIN;

reg RST;

wire SOUT;

schk s\_tb(

.CLK (CLK),

.DIN (DIN),

.RST (RST),

.SOUT (SOUT)

);

initial

begin

#0

CLK=0;RST=0;DIN=0;

#1

RST=1;

#2

RST=0;

#2

DIN=1;

#2

DIN=0;

#4

DIN=1;

#12

DIN=0;

#4

DIN=1;

#4

DIN=0;

#2

DIN=1; //从这里开始是正确的序列

#12

DIN=0;

#4

DIN=1;

#4

DIN=0;

#10000

$stop;

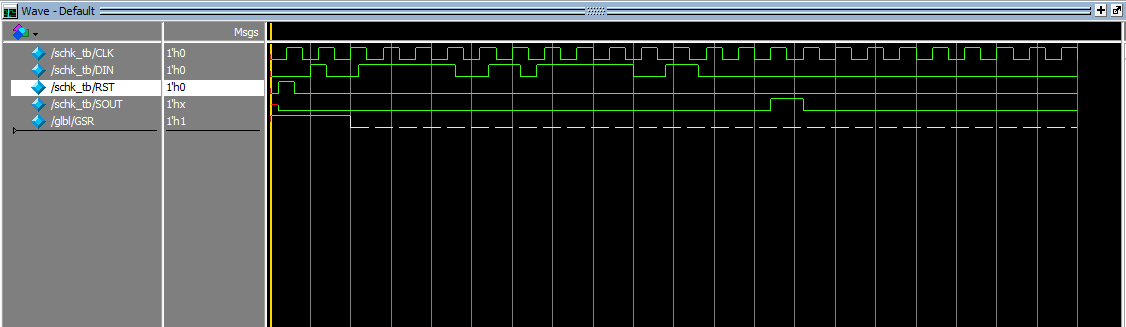
end

always #2 CLK=~CLK;

endmodule

设计框图：

仿真波形：



## 总结与体会

这次实验我自己尝试编写了仿真程序，了解了仿真程序的编写方法以及各项功能的含义。而主程序的代码虽然参考了书本，但对于序列检测器也有了一定的了解。