**湘 潭 大 学 实 验 报 告**

课程名称：数字逻辑与数字电路实验名称：序列信号发生器

班 级：22级计算机科学与技术4班 实验日期：2023.5.16

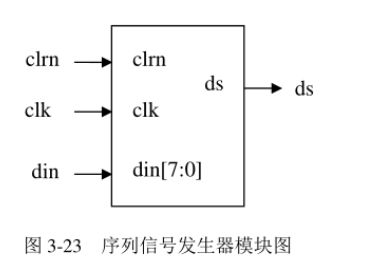
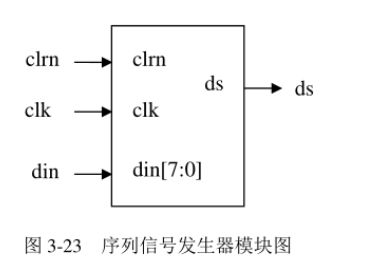
姓 名：张志雄 学 号：202205566912

### 实验目的

1、用状态机实现序列信号发生器的设计，了解一般状态机的设计与应用。  
2、熟悉Vivado，具备使用Verilog HDL编程、仿真并进行硬件测试的能力。

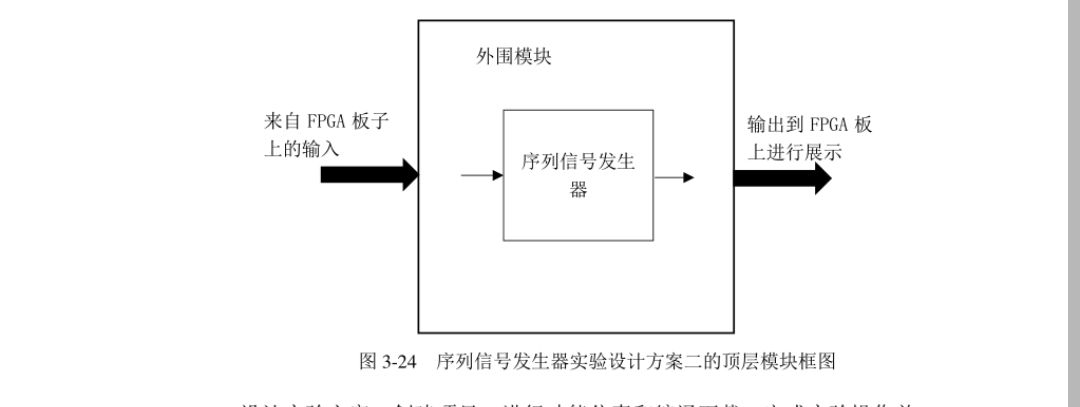
### 实验内容和要求

1. 序列信号发生器设计  
   通过 Verilog HDL 语言编程，实现一个8位序列信号发生器，要求有1个时钟脉冲输入、1个异步清零端、1个8位预置数据输入端和1个串行数据输出端，序列检测器输入、输出端口设计具体要求如下，序列信号发生器模块图如图3-23所示。

  
clrm：异步清零信号，低电平有效：  
clk：时钟脉冲输入，上升沿有效。  
din:8位预置数据输入端   
ds：串行数据输出端  
2、完成序列信号发生器的设计编辑和 din din[7:0]仿真测试等步骤，给出仿真波形，了解串行数据输出端的时序。 图3-23序列信号发生器模块图  
3、在硬件实现中，有不同的设计方案，  
以下例举2个。你也可以自行设计其他方案，不必拘泥于设计示例。  
(1)方案一：用拨码开关控制复位信号；按钮开关控制工作时钟；将8位预置数据由用拨码开关控制复位输入，串行数据输出端接LED灯。将具体的引脚分配填入表3-13。

  
(2)方案二：设计一个外围模块去调用该序列信号发生器模块，如图3-24

所示。外围模块中需调用封装好的LCD触摸屏模块，观察序列信号发生器的输入输出的值等。利用触摸功能输入预置的数据，实时观察输出的串行数据值的变化。



4、设计实验方案，创建项目，进行功能仿真和编译下载，完成实验操作并做好数据记录。  
如果采用实验设计方案一，建议下载后：1用复位按钮控制复位；2用8个拨码开关输入预置数据，例如“11010110”；③按单步调试按钮 8 次，产生 8个时钟上升沿，这时预置输入数据的低位到高位依次显示于LED灯。④设计数据记录表并记录实验结果。  
5、根据实验内容，写出实验报告，包括程序设计、软件编译、仿真分析、硬件测试和详细实验过程；给出程序分析报告、仿真波形图及其分析报告。

### 实验设备和软件环境

实验电脑设备规格：

设备名称：OMEN-15

处理器：AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz

系统类型：64 位操作系统, 基于 x64 的处理器

实验电脑Windows规格

版本：Windows 11

操作系统版本：22621.1413

实验箱设备：

设备名称：Cpu FPGA 教学实验系统V2.0

设备型号：LS-CPU-EXB-001

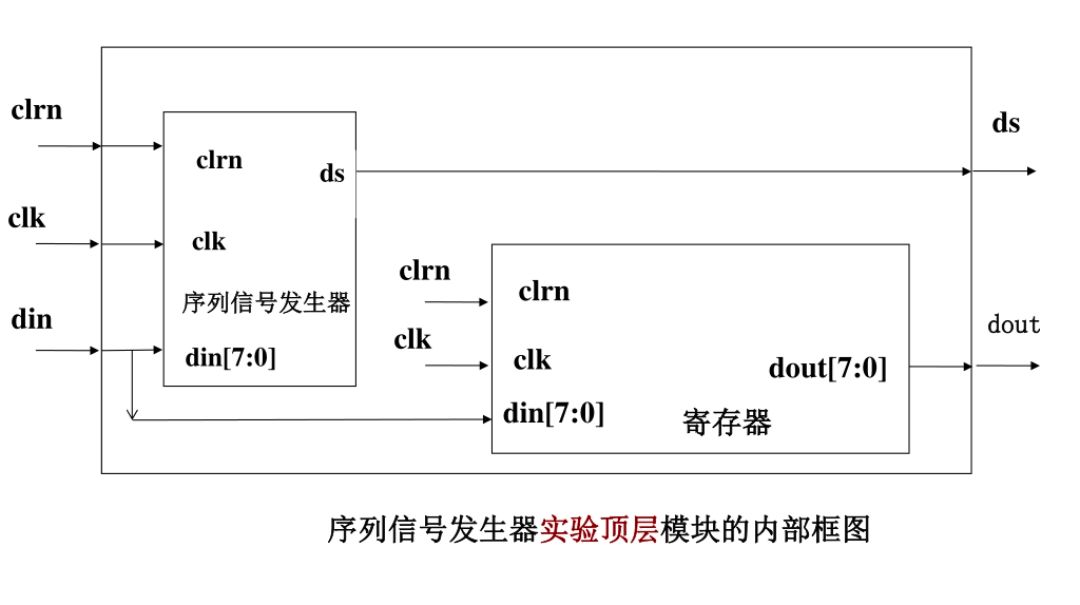
软件环境：

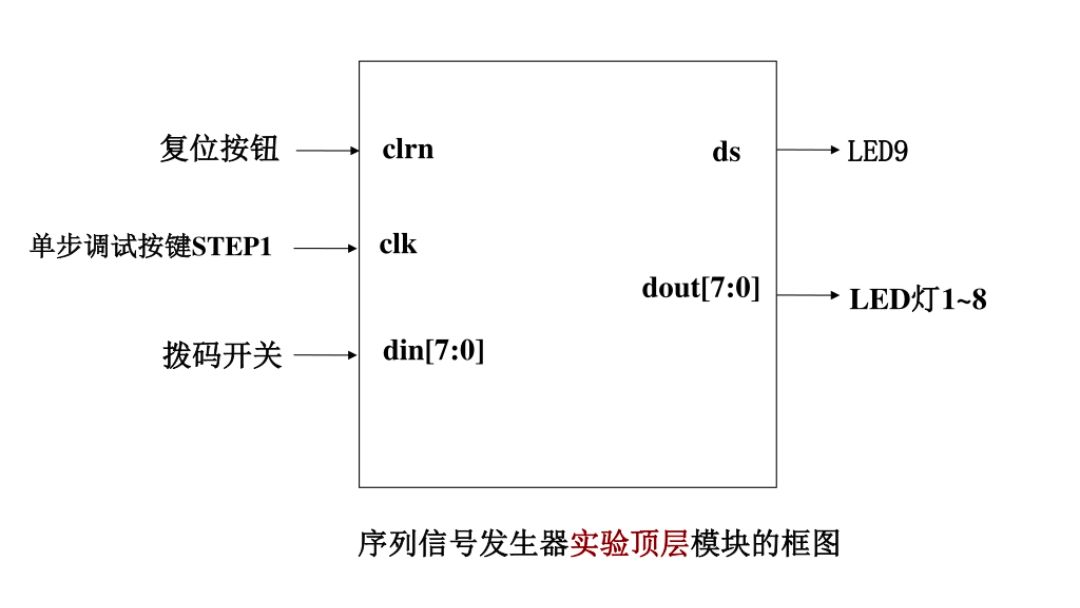
Vivado v2022.2

### 实验方案或原理

序列信号发生器可用于产生一组特定的由二进制码组成的脉冲序列信号。例如在数字信号的传输和数字系统的测试中，需要用到一组特定的串行数字信号‘11010110，就可以利用序列信号发生器产生该序列信号。  
序列信号发生器的设计方法有多种。比如可用计数器和数据选择器组成序列信号发生器，如需产生一个8位的序列信号11010110(时间顺序为自左向右)，则可用一个8进制计数器和一个8选1数据选择器组成。

### 代码设计及测试数据设计





1. sequencer.v（设计逻辑关系电路）

module sequencer(clk,clrn,din,ds,dout);

input clk,clrn;

input [7:0]din;

output [7:0]dout;

output ds;

register rgs(clk,clrn,din,dout);

sequencer\_module seq\_1(clk,clrn,din,ds);

endmodule

1. sequencer\_module.v

module sequencer\_module(clk,clrn,din,ds);

input [7:0]din;

input clk,clrn;

output ds;

wire [2:0]count;

counter cnt(clk,clrn,count);

mux81 mux81\_1(count,din,ds);

endmodule

1. register.v(设计逻辑关系电路)

module register(clk,clrn,din,dout);

input clk,clrn;

input [7:0]din;

output [7:0]dout;

reg [7:0]dout;

always @(posedge clk or negedge clrn)

if(clrn==0)

begin

dout<=8'd0;

end

else begin

dout<=din;

end

endmodule

1. counter.v

module counter(clk,clrn,count);

input clk,clrn;

output [2:0]count;

reg [2:0]count;

always @(posedge clk or negedge clrn)

begin

if(!clrn)

begin

count<=3'd0;

end

else begin

count =count+1;

end

end

endmodule

1. sequencer\_tb.v(模拟电路)

`include"counter.v"

`include"mux81.v"

`include"sequencer\_module.v"

`include"sequencer.v"

`include"register.v"

`timescale 1ns/1ns

module sequencer\_tb;

parameter DATA\_WIDTH=8;

reg [DATA\_WIDTH-1:0] din;

wire [DATA\_WIDTH-1:0] dout;

wire ds;

reg clk,clrn;

initial begin

clk=1;

clrn = 0;

din = 8'b00001110;

#0 $display("time\tclk\tclrn\tdin\tds\tdout");

#4 clrn = 1;

end

always #5 clk=~clk;

always #80 din=din+1;

sequencer seq\_1(.clk(clk),.clrn(clrn),.din(din),.ds(ds),.dout(dout));

endmodule

6、sequencer.xdc(约束文件)

set\_property PACKAGE\_PIN V6 [get\_ports clk]

set\_property CLOCK\_DEDICATED\_ROUTE FALSE [get\_nets clk\_IBUF]

set\_property PACKAGE\_PIN Y3 [get\_ports clrn]

set\_property PACKAGE\_PIN AC21 [get\_ports {din[7]}]

set\_property PACKAGE\_PIN AD24 [get\_ports {din[6]}]

set\_property PACKAGE\_PIN AC22 [get\_ports {din[5]}]

set\_property PACKAGE\_PIN AC23 [get\_ports {din[4]}]

set\_property PACKAGE\_PIN AB6 [get\_ports {din[3]}]

set\_property PACKAGE\_PIN W6 [get\_ports {din[2]}]

set\_property PACKAGE\_PIN AA7 [get\_ports {din[1]}]

set\_property PACKAGE\_PIN Y6 [get\_ports {din[0]}]

set\_property PACKAGE\_PIN J8 [get\_ports ds]

set\_property PACKAGE\_PIN H7 [get\_ports {dout[7]}]

set\_property PACKAGE\_PIN D5 [get\_ports {dout[6]}]

set\_property PACKAGE\_PIN A3 [get\_ports {dout[5]}]

set\_property PACKAGE\_PIN A5 [get\_ports {dout[4]}]

set\_property PACKAGE\_PIN A4 [get\_ports {dout[3]}]

set\_property PACKAGE\_PIN F7 [get\_ports {dout[2]}]

set\_property PACKAGE\_PIN G8 [get\_ports {dout[1]}]

set\_property PACKAGE\_PIN H8 [get\_ports {dout[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clrn]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports ds]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[2]}]

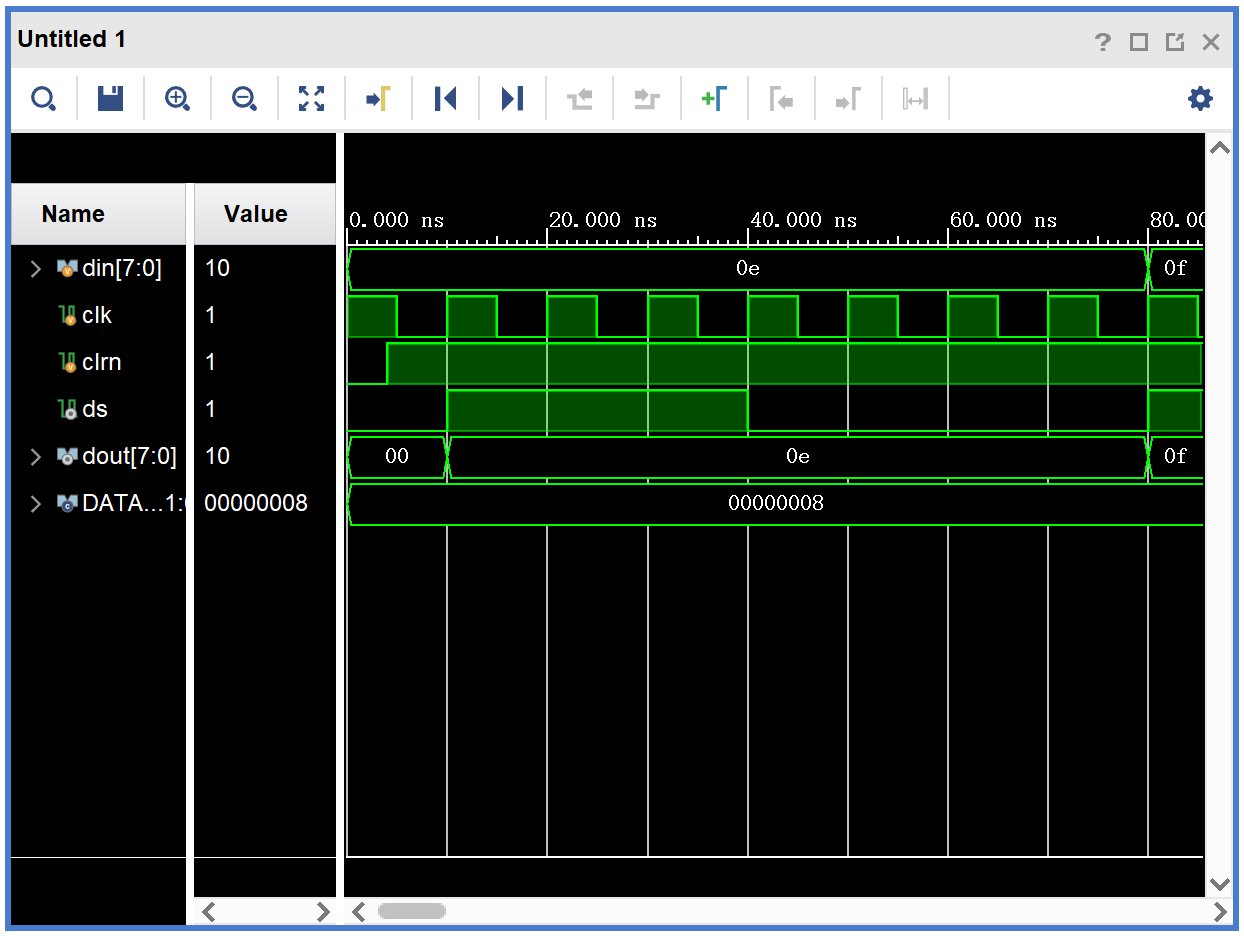
set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {dout[0]}]

### 实验操作步骤

1. 设计逻辑关系电路
2. 设计仿真文件
3. 设计约束文件
4. 下载文件至实验箱测试

### 实验结果验证及分析



### 八、其他

需要注意下拨开关为1，上拨开关为0

灯灭为1，灯亮为0

### 九、实验操作评分照片

