# 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: 16进制计数器实验 实验时间: 2022.5.21

班级:计算机科学与技术3班 学号:202105566416 姓名:熊心雨

## 一、实验目的

1）验证同步16进制计数器的功能，掌握层次化设计的方法

2）进一步掌握vivado的设计流程，掌握时序电路的设计仿真和硬件测试方法。

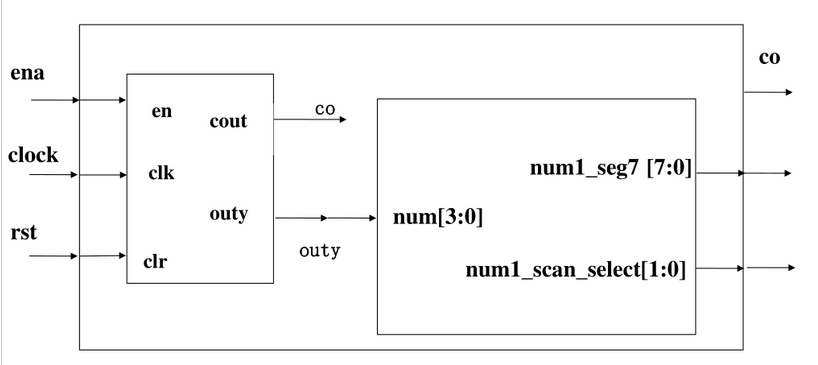
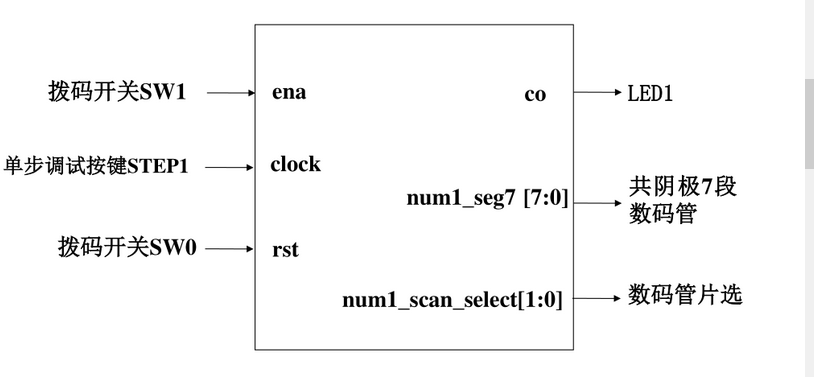
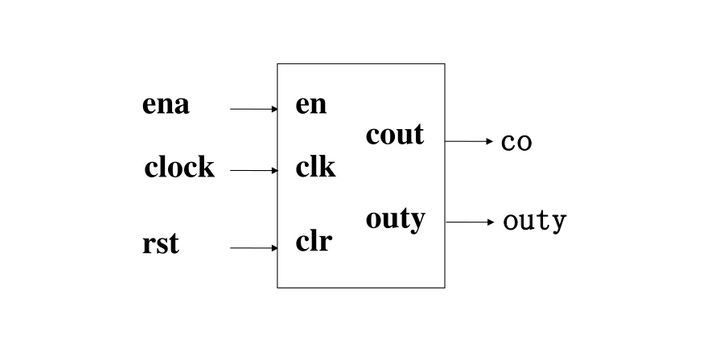
3）具备采用vivadoHDL设计16进制计数器仿真测试代码的能力。

## 二、实验内容和要求

1）16进制计数器的输入和仿真 ：

学习软件平台和设计流程。利用Vivado完成16进制计数器实验的文件编辑输入和仿真测试等步骤，并给出仿真波形。

1. 设计实验方案 ：



1. 在实验系统上进行硬件测试 ：

根据实验方案创建约束文件，绑定引脚，并下载 到试验箱的FPGA板子上进行硬件测试。

4） 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、硬件测试和实验操作步骤；给出源程序代码、仿真波形图、数据记录及实验结果分析。

## 三、实验设备与软件环境

1、装有Xilinx Vivado 的教学计算机一台。

2、LS-CPU-EXB-002 教学系统实验箱一套。

## 四、实验方案或原理

计数器能记忆输入脉冲的个数，主要用于定时、分频、产生节拍脉冲及进行数学运算等。加法加数器每输入一个CP脉冲，加法计数器的计数值加1。十六进制计数即从0000一直计数到1111；当计数到1111时，若再来一个CP脉冲，则回到0000，同时产生进位1。

同步十六进制计数器设计采用if-else语句对计数器的输出分别进行赋值，能实现对输入脉冲的计数，并具有使能和异步清零功能。

## 五、代码设计及实验步骤

1）代码设计

**1.counter.v**

module counter(en,clk,clr,cout,outy);

input en,clk,clr;

output [3:0]outy;

output cout;

reg [3:0]outy;

always @(posedge clk or posedge clr)

begin

if(clr) outy<=4'b0000;

else if(en)

begin

if(outy==4'b1111) outy<=4'b0000;

else outy<=outy+1'b1;

end

end

assign cout=((outy==4'b1111)&en)?1:0;

Endmodule

**2.counter\_decl7s.v**

module counter\_decl7s(ena,clock,rst,co,num1\_scan\_select,num1\_seg7);

input ena,clock,rst;

output [7:0]num1\_seg7;

output co;

output [1:0]num1\_scan\_select;

wire [3:0]outy;

counter counter(ena,clock,rst,co,outy);

decl7s decl7s(outy,num1\_scan\_select,num1\_seg7);

endmodule

**2.counter\_.tb.v**

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

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

set\_property PACKAGE\_PIN AC21 [get\_ports ena]

set\_property PACKAGE\_PIN AD24 [get\_ports rst]

set\_property PACKAGE\_PIN H7 [get\_ports co]

set\_property PACKAGE\_PIN E26 [get\_ports {num1\_scan\_select[1]}]

set\_property PACKAGE\_PIN G25 [get\_ports {num1\_scan\_select[0]}]

set\_property PACKAGE\_PIN C4 [get\_ports {num1\_seg7[7]}]

set\_property PACKAGE\_PIN A2 [get\_ports {num1\_seg7[6]}]

set\_property PACKAGE\_PIN D4 [get\_ports {num1\_seg7[5]}]

set\_property PACKAGE\_PIN E5 [get\_ports {num1\_seg7[4]}]

set\_property PACKAGE\_PIN B4 [get\_ports {num1\_seg7[3]}]

set\_property PACKAGE\_PIN B2 [get\_ports {num1\_seg7[2]}]

set\_property PACKAGE\_PIN E6 [get\_ports {num1\_seg7[1]}]

set\_property PACKAGE\_PIN C3 [get\_ports {num1\_seg7[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_scan\_select[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_scan\_select[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {num1\_seg7[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports clock]

set\_property IOSTANDARD LVCMOS33 [get\_ports ena]

set\_property IOSTANDARD LVCMOS33 [get\_ports rst]

set\_property IOSTANDARD LVCMOS33 [get\_ports co]

1. **decl7s.v**

module decl7s(num,num1\_scan\_select,num1\_seg7);

input [3:0] num;

output [1:0] num1\_scan\_select;

output [7:0] num1\_seg7;

reg [7:0] num1\_seg7;

assign num1\_scan\_select=2'b10;

always@(num)

case(num)

4'b0000 : num1\_seg7<=8'b01111110;//0

4'b0001 : num1\_seg7<=8'b00110000;//1

4'b0010 : num1\_seg7<=8'b01101101;//2

4'b0011 : num1\_seg7<=8'b01111001;//3

4'b0100 : num1\_seg7<=8'b00110011;//4

4'b0101 : num1\_seg7<=8'b01011011;//5

4'b0110 : num1\_seg7<=8'b01011111;//6

4'b0111 : num1\_seg7<=8'b01110000;//7

4'b1000 : num1\_seg7<=8'b01111111;//8

4'b1001 : num1\_seg7<=8'b01111011;//9

4'b1010 : num1\_seg7<=8'b01110111;//A

4'b1011 : num1\_seg7<=8'b00011111;//b

4'b1100 : num1\_seg7<=8'b01001110;//C

4'b1101 : num1\_seg7<=8'b00111101;//d

4'b1110 : num1\_seg7<=8'b01001111;//E

4'b1111 : num1\_seg7<=8'b01000111;//F

default : num1\_seg7<=8'b00000000;//0

endcase

endmodule

1. **counter\_decl7s\_tb.v**

`timescale 1ns/1ns

`include "counter.v"

`include "decl7s.v"

`include "counter\_decl7s.v"

module conuter\_decl7s\_tb;

parameter bit\_width=4;

reg clock,rst,ena;

wire [7:0]num1\_seg7;

wire co;

wire [1:0]num1\_scan\_select;

initial begin

clock<=0;rst<=1;ena<=0;

#0 $display("time\tena\tclock\trst\tco\tnum1\_scan\_select\tnum1\_seg7");

#5 rst<=0;ena<=1;

end

always #1 clock=~clock;

counter\_decl7s cnt\_decl7s(ena,clock,rst,co,num1\_scan\_select,num1\_seg7);

endmodule

2）创建工程 ：

在E盘新建一个文件夹COUNTER并且将设计源代码counter\_decl7s.v、counter.v、decl7s.v以及仿真测试代码counter\_tb.v、counter\_decl7s\_tb.v都放在该文件夹下，后在该文件夹下创建工程。

启动Vivado软件,在Quick Start中点击 Create Project，出现新建工程向导，选择“Next”，输入工程名称counter\_decl7s，选择工程的文件位置位E:/ counter\_decl7s。

3）添加设计模块：

在工程中添加设计文件和仿真测试文件,并指定FPGA器件的型号为“Arix”的“xc7a200fbg676-2”

4）功能仿真：

在左侧的导航栏中的“Run simulation“中选择”Run Behaviorl simulation“进行仿真。

分别对4位二进制加法计数器模块counter.v和计数器实验顶层模块counter\_decl7s.v的功能进行仿真。

5）实验方案设计 ：

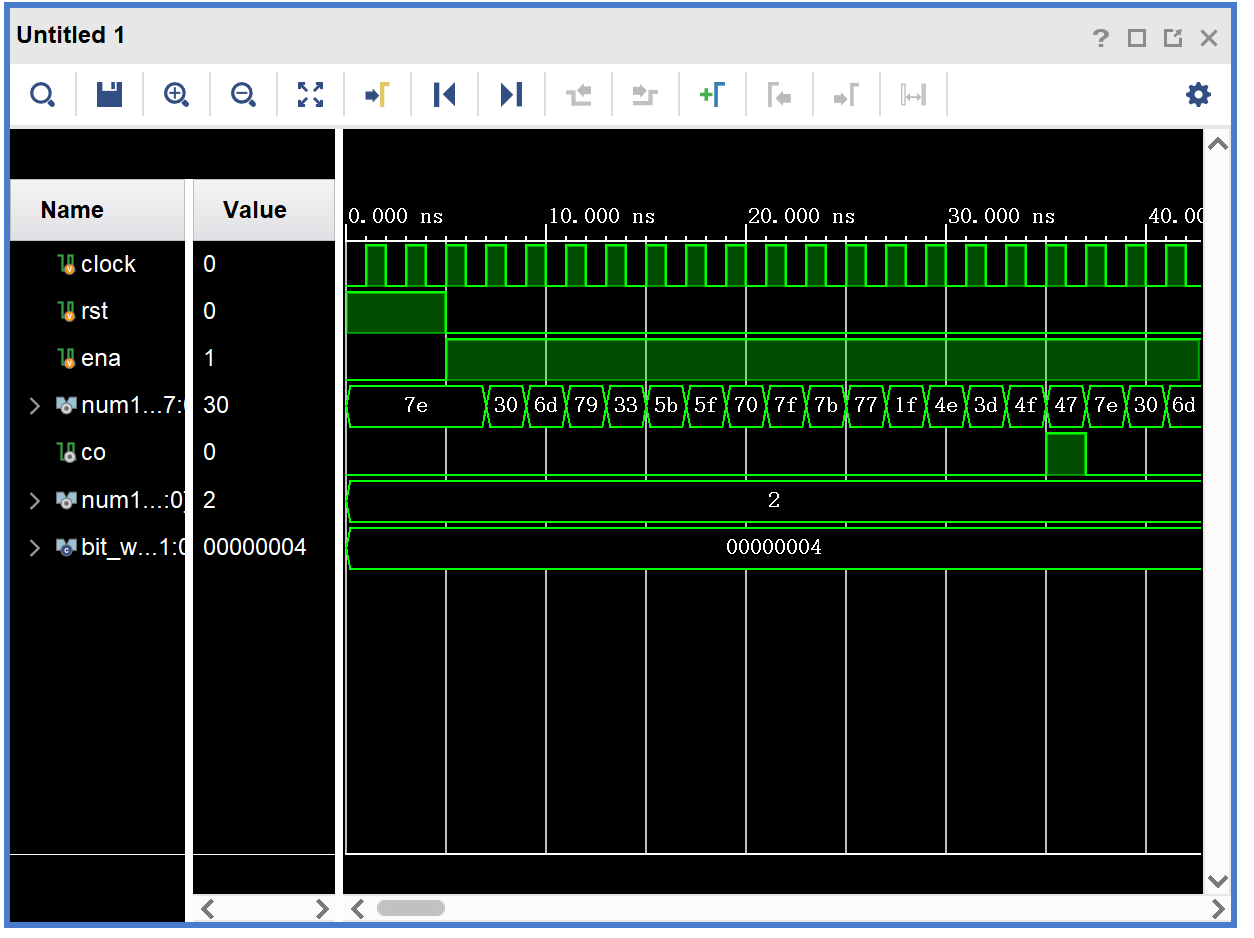
通过拨码开关分别控制使能和清零信号，单步按钮（按下为0，松开为1）作为计数脉冲输入，数码管显示译码输出的计数值，LED灯用于显示进位输出，而num1\_scan\_select则用于选择第几个数码管。

6）创建约束文件

7）生产下载文件

8）下载

## 六、实验结果验证及分析（包括实验现象及原因分析）



## 七、实验过程中出现的问题及处理情况（如排故障的方法等）或可研究与探索的问题与方法。

无

## 八、实验操作评分照片

