数电实验4：Verilog与FPGA开发及串口通讯

# 实验目的

1. 通过实验熟练掌握Verilog语言在FPGA中的程序设计方法。
2. 通过一个完整的UART串口通信FPGA项目，进一步熟悉Verilog语言，为将来的课程项目设计打下基础。

# 实验原理

## Verilog与FPGA开发

1. 将FPGA开发板自带时钟（125MHz）存在29位数组clk\_cnt中；
2. 取clk\_cnt的最高四位即达到降频目的；
3. 利用降频后的时钟信号进行一系列的时序逻辑操作。

## 串口通讯

1. UART串口通信需要两根信号线来实现，一根用于串口发送，另外一根负责串口接收；
2. UART属于异步通信，不要求通信双方在同一时钟控制下同步传输数据，发送端可以在任意时刻发送数据；
3. 因为是异步，需要在UART发送和接收设备上指定相同的传输速率，以及空闲位、起始位、校验位、停止位，收发双方需遵循相同的通信协议，按ppt所给图片设置好参数；
4. 进行串口通讯。

# 实验内容、结果与分析

## Verilog与FPGA开发

### 四位拨码开关控制显示数字0000~FFFF

#### **①**程序理解

module seg7ment\_sub( **//定义了一个数码管**

input wire [3:0] num,

output reg [6:0] a\_to\_g

);

always @(\*)

case(num) **//以下赋给了当num为不同值时每个数码显示的数字**

0: a\_to\_g=7'b1000000;

1: a\_to\_g=7'b1111001;

2: a\_to\_g=7'b0100100;

3: a\_to\_g=7'b0110000;

4: a\_to\_g=7'b0011001;

5: a\_to\_g=7'b0010010;

6: a\_to\_g=7'b0000010;

7: a\_to\_g=7'b1111000;

8: a\_to\_g=7'b0000000;

9: a\_to\_g=7'b0010000;

'hA: a\_to\_g=7'b0001000;

'hB: a\_to\_g=7'b0000011;

'hC: a\_to\_g=7'b1000110;

'hD: a\_to\_g=7'b0100001;

'hE: a\_to\_g=7'b0000110;

'hF: a\_to\_g=7'b0001110;

default: a\_to\_g=7'b1000000;

endcase

endmodule

module seg7ment\_top ( **//定义了控制4个数码管的类**

input wire [3:0] sw,

output wire [6:0] a\_to\_g,

output wire [3:0] an

);

assign an=4'b0000;

seg7ment\_sub A1(.num(sw),.a\_to\_g(a\_to\_g)); **//实例化**

endmodule

#### ②管脚定义文件

#Seven segment display，**将an用作使能，a\_to\_g用作控制每个段的亮灭**

set\_property -dict { PACKAGE\_PIN K19 IOSTANDARD LVCMOS33 } [get\_ports { an[0] }];

set\_property -dict { PACKAGE\_PIN H20 IOSTANDARD LVCMOS33 } [get\_ports { an[1] }];

set\_property -dict { PACKAGE\_PIN J18 IOSTANDARD LVCMOS33 } [get\_ports { an[2] }];

set\_property -dict { PACKAGE\_PIN J20 IOSTANDARD LVCMOS33 } [get\_ports { an[3] }];

set\_property -dict { PACKAGE\_PIN H19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[0] }];

set\_property -dict { PACKAGE\_PIN H17 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[1] }];

set\_property -dict { PACKAGE\_PIN K18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[2] }];

set\_property -dict { PACKAGE\_PIN K21 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[3] }];

set\_property -dict { PACKAGE\_PIN M20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[4] }];

set\_property -dict { PACKAGE\_PIN H18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[5] }];

set\_property -dict { PACKAGE\_PIN L19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[6] }];

set\_property -dict { PACKAGE\_PIN K20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[7] }];

#Switch，**将sw绑定为switch作为输入**

set\_property -dict { PACKAGE\_PIN T6 IOSTANDARD LVCMOS33 } [get\_ports { sw[0] }];

set\_property -dict { PACKAGE\_PIN U5 IOSTANDARD LVCMOS33 } [get\_ports { sw[1] }];

set\_property -dict { PACKAGE\_PIN T4 IOSTANDARD LVCMOS33 } [get\_ports { sw[2] }];

set\_property -dict { PACKAGE\_PIN V4 IOSTANDARD LVCMOS33 } [get\_ports { sw[3] }];

#### ③实验结果

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> project\_1 -> lab1rst.MP4

### 拨码开关控制数字 0~F 循环显示

#### ①程序理解

// module clk\_sw\_7seg\_sub模块与实验一一致

module clk\_sw\_7seg\_top (  **//定义了控制4个数码管的类**

input clk,

input clr,

input[3:0]sw,

output[3:0]an,

output[6:0]a\_to\_g

);

reg[28:0] clk\_cnt; //clock count  **//将时钟存储在29bit的变量中**

reg [3:0] NUM; //number printed on the segment LEDs

always @(posedge clk or posedge clr) //deal the clock and clear events

begin

if (clr)

clk\_cnt = 0; //if clear button pressed,clear the clock count

else

begin

clk\_cnt = clk\_cnt + 1; //if clock flip, count clock

end

end

always @(\*)

NUM = clk\_cnt[28:25]; **//取时钟信号的前4位，达到降频目的**

clk\_sw\_7seg\_sub A1(.NUM(NUM),.sw(sw),.an(an),.a\_to\_g(a\_to\_g)); //call the sub display module

endmodule

#### ②管脚定义文件

#system clock - 125 MHz，**绑定时钟到clk输入**

set\_property -dict { PACKAGE\_PIN L18 IOSTANDARD LVCMOS33 } [get\_ports { clk }];

#BTN，**绑定BTN0为clr输入**

set\_property -dict { PACKAGE\_PIN U12 IOSTANDARD LVCMOS33 } [get\_ports { clr }];

#Seven segment display

set\_property -dict { PACKAGE\_PIN K19 IOSTANDARD LVCMOS33 } [get\_ports { an[0] }];

set\_property -dict { PACKAGE\_PIN H20 IOSTANDARD LVCMOS33 } [get\_ports { an[1] }];

set\_property -dict { PACKAGE\_PIN J18 IOSTANDARD LVCMOS33 } [get\_ports { an[2] }];

set\_property -dict { PACKAGE\_PIN J20 IOSTANDARD LVCMOS33 } [get\_ports { an[3] }];

set\_property -dict { PACKAGE\_PIN H19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[0] }];

set\_property -dict { PACKAGE\_PIN H17 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[1] }];

set\_property -dict { PACKAGE\_PIN K18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[2] }];

set\_property -dict { PACKAGE\_PIN K21 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[3] }];

set\_property -dict { PACKAGE\_PIN M20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[4] }];

set\_property -dict { PACKAGE\_PIN H18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[5] }];

set\_property -dict { PACKAGE\_PIN L19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[6] }];

set\_property -dict { PACKAGE\_PIN K20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[7] }];

#Switch

set\_property -dict { PACKAGE\_PIN T6 IOSTANDARD LVCMOS33 } [get\_ports { sw[0] }];

set\_property -dict { PACKAGE\_PIN U5 IOSTANDARD LVCMOS33 } [get\_ports { sw[1] }];

set\_property -dict { PACKAGE\_PIN T4 IOSTANDARD LVCMOS33 } [get\_ports { sw[2] }];

set\_property -dict { PACKAGE\_PIN V4 IOSTANDARD LVCMOS33 } [get\_ports { sw[3] }];

#### ③实验结果

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> project\_2 ->>lab2rst.MP4

**问题：计算数码管显示数字的变化频率是多少？**

**解答：clk\_cnt的第0位是125MHz，所以第25位是125MHz/为3.725Hz，即为数字变化频率。**

### 二位数码管循环显示数字00~FF

#### ①程序理解

**原程序：**

assign s = clk\_cnt[15]; ***//取时钟的倍频为交替显示的频率，即3814Hz***

***//由于频率很快，人眼看不出交替，达到全显示的效果***

assign an[0] = s; ***//第四盏灯使能***

assign an[1] = ~s; ***//第三盏灯使能，二者交替，故为bar(s)***

assign an[3:2] = 2'b11;  ***//前两盏灯一直熄灭***

always @(\*)

case (s) //take the higher four bits as number for printing

1:NUM = clk\_cnt[32:29]; ***//此时十位亮，取时钟的(前两位)b为(十位)h***

0:NUM = clk\_cnt[28:25]; ***//此时个位亮***

endcase

clk\_7segff\_sub A1(.NUM(NUM),.a\_to\_g(a\_to\_g)); //call the sub display module

endmodule

**修改后程序：**

仅改动这两句：

assign an[0] = ~s;

assign an[1] = s;

相当于将个位十位对调。

#### ②管脚定义文件

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

#system clock - 125 MHz

set\_property -dict { PACKAGE\_PIN L18 IOSTANDARD LVCMOS33 } [get\_ports { clk }];

#BTN，**绑定BTN0为clr输入**

set\_property -dict { PACKAGE\_PIN U12 IOSTANDARD LVCMOS33 } [get\_ports { clr }];

set\_property -dict { PACKAGE\_PIN V12 IOSTANDARD LVCMOS33 } [get\_ports { BTN1 }];

set\_property -dict { PACKAGE\_PIN U7 IOSTANDARD LVCMOS33 } [get\_ports { BTN2 }];

set\_property -dict { PACKAGE\_PIN Y6 IOSTANDARD LVCMOS33 } [get\_ports { BTN3 }];

#Seven segment display，**绑定数码管**

set\_property -dict { PACKAGE\_PIN K19 IOSTANDARD LVCMOS33 } [get\_ports { an[0] }];

set\_property -dict { PACKAGE\_PIN H20 IOSTANDARD LVCMOS33 } [get\_ports { an[1] }];

set\_property -dict { PACKAGE\_PIN J18 IOSTANDARD LVCMOS33 } [get\_ports { an[2] }];

set\_property -dict { PACKAGE\_PIN J20 IOSTANDARD LVCMOS33 } [get\_ports { an[3] }];

set\_property -dict { PACKAGE\_PIN H19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[0] }];

set\_property -dict { PACKAGE\_PIN H17 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[1] }];

set\_property -dict { PACKAGE\_PIN K18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[2] }];

set\_property -dict { PACKAGE\_PIN K21 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[3] }];

set\_property -dict { PACKAGE\_PIN M20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[4] }];

set\_property -dict { PACKAGE\_PIN H18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[5] }];

set\_property -dict { PACKAGE\_PIN L19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[6] }];

set\_property -dict { PACKAGE\_PIN K20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[7] }];

#Switch，**绑定sw输入**

set\_property -dict { PACKAGE\_PIN T6 IOSTANDARD LVCMOS33 } [get\_ports { sw[0] }];

set\_property -dict { PACKAGE\_PIN U5 IOSTANDARD LVCMOS33 } [get\_ports { sw[1] }];

set\_property -dict { PACKAGE\_PIN T4 IOSTANDARD LVCMOS33 } [get\_ports { sw[2] }];

set\_property -dict { PACKAGE\_PIN V4 IOSTANDARD LVCMOS33 } [get\_ports { sw[3] }];

set\_property -dict { PACKAGE\_PIN W8 IOSTANDARD LVCMOS33 } [get\_ports { SW4 }];

set\_property -dict { PACKAGE\_PIN U9 IOSTANDARD LVCMOS33 } [get\_ports { SW5 }];

set\_property -dict { PACKAGE\_PIN W10 IOSTANDARD LVCMOS33 } [get\_ports { SW6 }];

set\_property -dict { PACKAGE\_PIN V9 IOSTANDARD LVCMOS33 } [get\_ports { SW7 }];

#### ③实验结果

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> project\_3 -> lab3rst1.MP4

数电实验4：Verilog与FPGA开发及串口通讯 -> project\_3 -> lab3rst2.MP4

### 二位数码管循环显示数字00~99

#### ①程序理解

module B7\_to\_D2( input [6:0] B\_7, ***//输入7位二进制***

output [7:0] D\_2 ***//输出以8位二进制表示的2位十进制(00-99)***

);

reg [3:0] c;

reg [7:0] q;

assign D\_2 = {c,q[3:0]}; ***//将前四位赋值c，后四位赋值q***

always @(\*)

begin

c = B\_7[6:0]/10; ***//c的值为{[(B\_7的十进制)的十位]的二进制}存储***

q = (B\_7[6:0]%10); ***//与c类似，只不过是个位***

end

endmodule

module clk\_7seg99\_top(input clk,

input clr,

input pause,

output [3:0] an,

output [6:0] a\_to\_g

);

reg [30:0] clk\_cnt;

reg [3:0] NUM;

reg [16:0] s;

reg p\_flag;

wire [7:0] temp;

wire [7:0] temp\_1;

always @(posedge clk or posedge clr)  ***//当按下清0或在时钟的上升沿***

begin

if (clr)  ***//清零部分，使时钟重新计时***

clk\_cnt = 0;

else if (!p\_flag) ***//没有暂停时***

begin

if (clk\_cnt[30:24] < 99) ***//取时钟最高7位，当其小于循环周期99时***

clk\_cnt = clk\_cnt + 1; ***//时钟前进***

else ***//若到了99***

clk\_cnt = 0; ***//则清零重新计时***

end

end

always @(posedge pause)  ***//当按下暂停时***

p\_flag = ~p\_flag;  ***//反转暂停信号，使时钟clk\_cnt暂停前进***

always @(posedge clk)

begin

if (clk)

begin

s = s + 1;  ***//s接入时钟计时***

if (s[16])  ***//s满了就清零***

s = 0;

end

end

assign temp\_1 = clk\_cnt[30:24]; ***//取时钟的最高7位，赋值给B\_7***

assign an[0] = s[15]; ***//使能信号，s[15]的频率为时钟的，即3814Hz***

assign an[1] = ~s[15];  ***//与实验三类似，使这两盏灯交替显示***

assign an[3:2] = 2'b11; ***//前两盏灯一直熄灭***

always @(\*)

case (s[15])

1:NUM = temp[7:4];  ***//第3盏灯亮，输入信号为D\_2的前四位，即十位***

0:NUM = temp[3:0];  ***//与上类似，第四盏灯输入个位***

endcase

B7\_to\_D2 C1(temp\_1, temp);

clk\_7seg99\_sub A1(.NUM(NUM),.a\_to\_g(a\_to\_g));

endmodule

#### ②管脚定义文件

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

#system clock - 125 MHz

set\_property -dict { PACKAGE\_PIN L18 IOSTANDARD LVCMOS33 } [get\_ports { clk }];

#BTN

set\_property -dict { PACKAGE\_PIN U12 IOSTANDARD LVCMOS33 } [get\_ports { clr }];

set\_property -dict { PACKAGE\_PIN V12 IOSTANDARD LVCMOS33 } [get\_ports { pause }];

#Seven segment display

set\_property -dict { PACKAGE\_PIN K19 IOSTANDARD LVCMOS33 } [get\_ports { an[0] }];

set\_property -dict { PACKAGE\_PIN H20 IOSTANDARD LVCMOS33 } [get\_ports { an[1] }];

set\_property -dict { PACKAGE\_PIN J18 IOSTANDARD LVCMOS33 } [get\_ports { an[2] }];

set\_property -dict { PACKAGE\_PIN J20 IOSTANDARD LVCMOS33 } [get\_ports { an[3] }];

set\_property -dict { PACKAGE\_PIN H19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[0] }];

set\_property -dict { PACKAGE\_PIN H17 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[1] }];

set\_property -dict { PACKAGE\_PIN K18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[2] }];

set\_property -dict { PACKAGE\_PIN K21 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[3] }];

set\_property -dict { PACKAGE\_PIN M20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[4] }];

set\_property -dict { PACKAGE\_PIN H18 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[5] }];

set\_property -dict { PACKAGE\_PIN L19 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[6] }];

set\_property -dict { PACKAGE\_PIN K20 IOSTANDARD LVCMOS33 } [get\_ports { a\_to\_g[7] }];

#Switch

set\_property -dict { PACKAGE\_PIN T6 IOSTANDARD LVCMOS33 } [get\_ports { sw[0] }];

set\_property -dict { PACKAGE\_PIN U5 IOSTANDARD LVCMOS33 } [get\_ports { sw[1] }];

set\_property -dict { PACKAGE\_PIN T4 IOSTANDARD LVCMOS33 } [get\_ports { sw[2] }];

set\_property -dict { PACKAGE\_PIN V4 IOSTANDARD LVCMOS33 } [get\_ports { sw[3] }];

#### ③实验结果

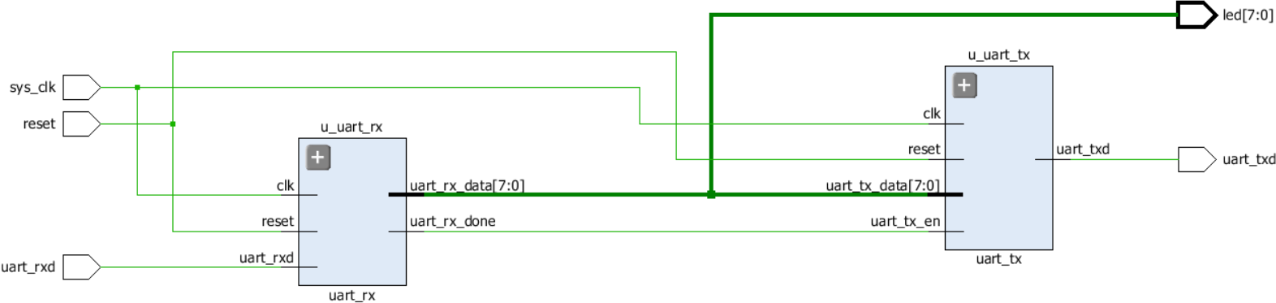
见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> project\_4 -> lab4rst.mp4

## 串口通讯

#### ①程序设计

本实验项目自顶向下分为几个模块：uart\_loopback.v（顶层模块）、uart\_rx.v（子模块）、uart\_tx.v（子模块），以及端口约束文件uart.xdc。程序框图如下：



修改思路：

将特定输入信号作为标志，并新增一个作为LCD开关的输出值，当接收到特定字符时，会闭合数字开关，使LCD显示，同样也可关闭。

#### ②实验结果

单次发送结果：

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> uart -> uart单次发送.png

多次发送：

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> uart -> uart多次发送.mp4

## 附加实验：BONUS

#### ①实验内容说明

1) 触摸屏功能展示

用Vivado打开项目touchscreen目录中的touchscreen.xpr，已经编译好（无需重新编译，可少占用时间），直接上传到FPGA。

功能说明：

SW0=0触摸位置显示功能关闭，无论触摸与否，四个七段数码管全部显示0。

SW0=1触摸位置显示功能开启。

SW0=1时，如果触碰触摸屏，LED0亮。

SW0=1时，如果没有触碰触摸屏，LED0暗。

SW0=1，SW1=0时，LED1灭，手指划过触摸屏，七段数码管显示手指所处横坐标。

SW0=1，SW1=1时，LED1亮，手指划过触摸屏，七段数码管显示手指所处纵坐标。

(2) OLED液晶屏功能展示

用Vivado打开项目oled目录中的oled.xpr，直接上传到FPGA。

功能说明：

在OLED液晶屏上显示64个ASCII字符。

BTN0重置。

#### ②实验结果

触摸屏功能展示实验结果：

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> bonus -> LCDtorchScreen.MOV

OLED液晶屏功能展示：

见文件：

数电实验4：Verilog与FPGA开发及串口通讯 -> bonus ->OLEDscreen.MOV

# 遇到的问题

### 在二位数码管循环显示数字00~FF实验无法理解s值取取时钟的倍频的含义。在老师的帮助下理解实际上数码管以此频率时刻进行跳动，只是由于肉眼无法分辨。

1. 在BONUS实验中遇到文件可以编译但无法上传的问题。将代码文件单独作为一个新的文件，避免无法找到文件的错误。

# 五、反思与心得

在进行数字电子技术实验，并运用Verilog语言与FPGA进行开发的过程中，我深刻体验到了数字电路设计的魅力和实用性。通过这次实验，我获得了丰富的经验，提升了我的硬件设计和编程技能，实现了数码管16进制交替显示等内容。串口通信更是实际生活中常用到的技术，通过学会如何与外部设备进行通信，我不仅实现了数字的输入输出，还为日后更复杂的系统集成打下了坚实基础。