

# 浙江大学

## 本科实验报告

课程名称：数字逻辑设计

姓名：王伟杰

学院：计算机学院

系：软件工程

专业：软件工程

学号：3210106034

指导教师：马德

2022 年 11 月 29 日

# 浙江大学实验报告

课程名称： 数字逻辑设计 实验类型：

实验项目名称： 同步时序电路设计

学生姓名： 王伟杰 专业： 软件工程 学号： 3210106034

同组学生姓名： 王熠、陈苇远 指导老师： 马德

实验地点： 东 4-509 实验日期： 2022 年 11 月 29 日

## 一、实验目的

- 1.1 掌握典型同步时序电路的工作原理和设计方法。
- 1.2 掌握时序电路的激励函数、状态图、状态方程的运用。
- 1.3 掌握用Verilog进行有限状态机的设计、调试、仿真。
- 1.4 掌握用FPGA实现时序电路功能。

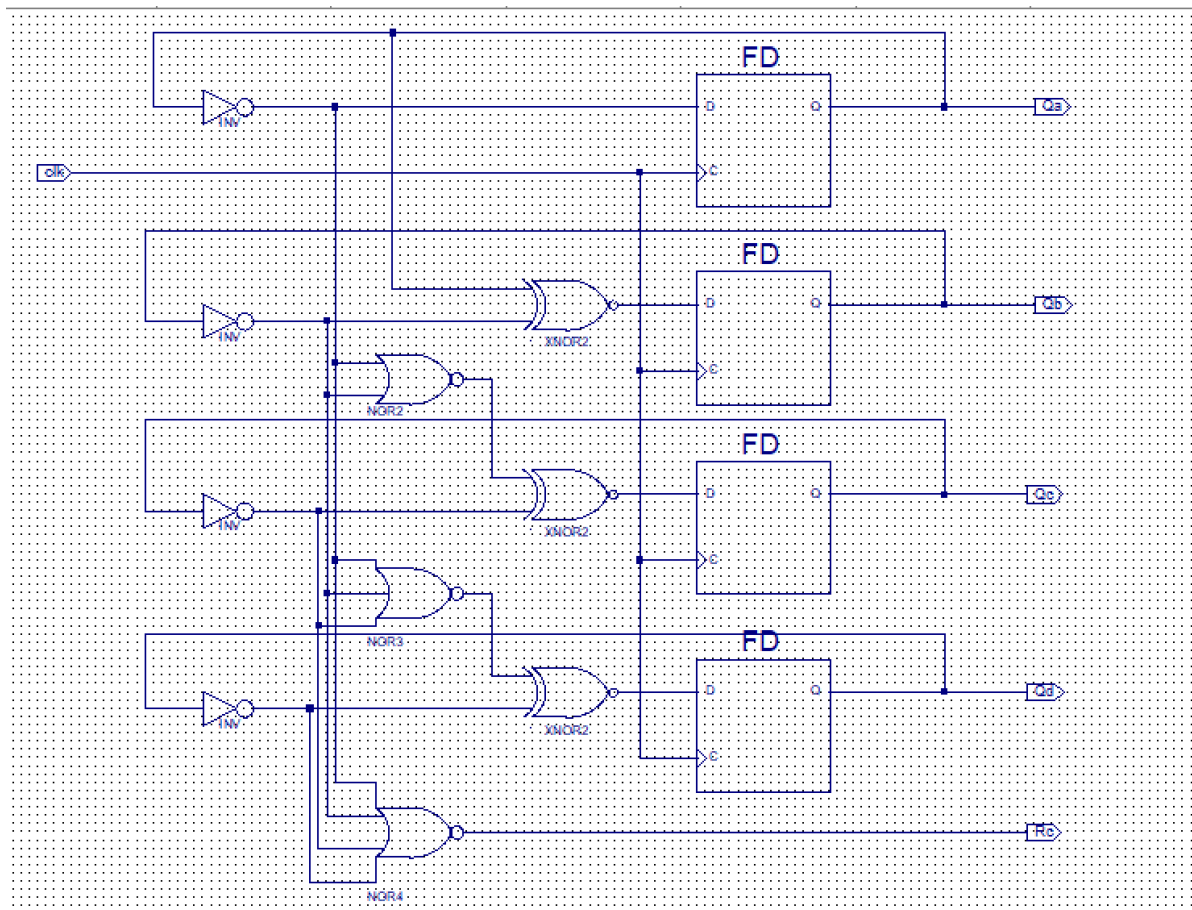
## 二、实验内容

- 2.1 原理图方式设计4位同步二进制计数器。
- 2.2 以Verilog行为描述方式设计16位可逆二进制同步计数器。

## 三、操作方法与实验步骤

### 3.1 设计4位同步二进制计数器

#### 1. 新建schematic源文件，绘制原理图



## 2. 对模块进行仿真

激励代码如下：

```

1  module Counter4b_Counter4b_sch_tb();
2
3  // Inputs
4      reg clk;
5
6  // Output
7      wire Qb;
8      wire Qc;
9      wire Qd;
10     wire Qa;
11     wire Rc;
12
13 // Bidirs
14
15 // Instantiate the UUT
16 Counter4b UUT (
17     .clk(clk),
18     .Qb(Qb),
19     .Qc(Qc),
20     .Qd(Qd),
21     .Qa(Qa),
22     .Rc(Rc)
23 );
24 initial begin
25     fork clk = 0; #50;
26         forever #50 clk <= ~clk;

```

```

27     join
28     end
29
30 endmodule

```

### 3. 新建clk\_1s, 编写代码如下

```

1  module clk_1s(
2      clk,
3      clk_1s
4  );
5
6      input wire clk;
7      output reg clk_1s;
8
9      reg[31:0] cnt;
10
11     always @ (posedge clk) begin
12         if (cnt < 50_000_000) begin
13             cnt <= cnt + 1;
14         end
15         else begin
16             cnt <= 0;
17             clk_1s <= ~clk_1s;
18         end
19     end
20
21 endmodule

```

### 4. 新建top文件, 根据"分频器设计"得到1s时钟, 显示在1位数码管上

```

1  module Top(
2      input wire clk,
3      output wire[7:0] segment,
4      output wire[3:0] an,
5      output wire[7:0] LED
6  );
7
8      wire [3: 0] disp_hexs;
9      wire clk_1s;
10
11     Counter4b m0(.clk(clk_1s), .Qa(disp_hexs[0]), .Qb(disp_hexs[1]),
12     .Qc(disp_hexs[2]), .Qd(disp_hexs[3]), .Rc(LED[0]));
13     clk_1s m1(.clk(clk), .clk_1s(clk_1s));
14     disp_num m2(.clk(clk), .HEXS(disp_hexs), .LES(4'b1110), .points(4'b0),
15     .rst(1'b0), .AN(an), .segment(segment));
16
17     assign LED[7:1]=8'b11111111;
18
19 endmodule

```

## 5. 设置引脚约束、

```
1  NET "clk" LOC = AC18      | IOSTANDARD = LVCMOS18 ;
2
3  NET "segment[0]"         LOC = AB22      | IOSTANDARD = LVCMOS33;
4  NET "segment[1]"         LOC = AD24      | IOSTANDARD = LVCMOS33;
5  NET "segment[2]"         LOC = AD23      | IOSTANDARD = LVCMOS33;
6  NET "segment[3]"         LOC = Y21       | IOSTANDARD = LVCMOS33;
7  NET "segment[4]"         LOC = W20       | IOSTANDARD = LVCMOS33;
8  NET "segment[5]"         LOC = AC24      | IOSTANDARD = LVCMOS33;
9  NET "segment[6]"         LOC = AC23      | IOSTANDARD = LVCMOS33;
10 NET "segment[7]"         LOC = AA22      | IOSTANDARD = LVCMOS33;
11
12 NET "an[0]"               LOC = AD21      | IOSTANDARD = LVCMOS33;
13 NET "an[1]"               LOC = AC21      | IOSTANDARD = LVCMOS33;
14 NET "an[2]"               LOC = AB21      | IOSTANDARD = LVCMOS33;
15 NET "an[3]"               LOC = AC22      | IOSTANDARD = LVCMOS33;
16
17 NET "LED[0]" LOC = AB26      | IOSTANDARD = LVCMOS33;
18 NET "LED[1]" LOC = W24       | IOSTANDARD = LVCMOS33;
19 NET "LED[2]" LOC = W23       | IOSTANDARD = LVCMOS33;
20 NET "LED[3]" LOC = AB25      | IOSTANDARD = LVCMOS33;
21 NET "LED[4]" LOC = AA25      | IOSTANDARD = LVCMOS33;
22 NET "LED[5]" LOC = W21       | IOSTANDARD = LVCMOS33;
23 NET "LED[6]" LOC = V21       | IOSTANDARD = LVCMOS33;
24 NET "LED[7]" LOC = W26       | IOSTANDARD = LVCMOS33;
```

## 3.2 设计16位可逆二进制同步计数器

### 1. 新建工程和Verilog文件RevCounter

```
1  module RevCounter16b(clk, s, cnt, Rc);
2
3      input wire clk, s;
4      output reg [15:0] cnt;
5      output wire Rc;
6
7      initial begin cnt = 0;
8      end
9
10     assign Rc = (~s & (~cnt)) | (s & (&cnt));
11     always @ (posedge clk) begin
12         if (s)
13             cnt <= cnt + 1;
14         else
15             cnt <= cnt - 1;
16     end
17
18 endmodule
```

## 2. 仿真得到激励波形

s = 0时:

```
1 module RevCounter16b_sim;
2     // Inputs
3     reg clk;
4     reg s;
5     // Outputs
6     wire [15:0] cnt;
7     wire Rc;
8     // Instantiate the Unit Under Test (UUT)
9     RevCounter16b uut (
10         .clk(clk),
11         .s(s),
12         .cnt(cnt),
13         .Rc(Rc)
14     );
15     initial begin
16         clk = 0;
17         s = 0;
18     end
19     always begin
20         clk=0;#50;
21         clk=1;#50;
22     end
23 endmodule
```

s = 1时:

```
1 module RevCounter16b_sim;
2     // Inputs
3     reg clk;
4     reg s;
5     // Outputs
6     wire [15:0] cnt;
7     wire Rc;
8     // Instantiate the Unit Under Test (UUT)
9     RevCounter16b uut (
10         .clk(clk),
11         .s(s),
12         .cnt(cnt),
13         .Rc(Rc)
14     );
15     initial begin
16         clk = 0;
17         s = 1;
18     end
19     always begin
20         clk=0;#50;
21         clk=1;#50;
22     end
23 endmodule
```

### 3. 新建Verilog文件top

```
1 module Top(  
2     input wire clk,  
3     input wire sw,  
4     output wire[7:0] segment,  
5     output wire[3:0] an,  
6     output wire[7:0] LED,  
7     output wire seg_clk,  
8     output wire seg_clrn,  
9     output wire seg_sout,  
10    output wire SEG_PEN  
11 );  
12  
13    wire [15:0] disp_hexs;  
14    wire [ 31: 0 ] clk_div;  
15    clkdiv m4( clk, { 1'b0 }, clk_div );  
16    RevCounter16b m0(.clk(clk_100ms), .s(sw), .cnt(disp_hexs), .Rc(LED[0]));  
17    clk_100ms m1(.clk(clk), .clk_100ms(clk_100ms));  
18    disp_num m2(.clk(clk), .HEXS(disp_hexs), .LES(4'b0), .points(4'b0),  
19    .rst(1'b0), .AN(an), .segment(segment));  
20  
21 endmodule
```

其中, clk\_100ms是100ms时钟模块:

```
1 module clk_100ms(  
2     clk,  
3     clk_100ms  
4 );  
5  
6     input wire clk;  
7     output reg clk_100ms;  
8  
9     reg[31:0] cnt;  
10  
11     always @ (posedge clk) begin  
12         if (cnt < 5_000_000) begin  
13             cnt <= cnt + 1;  
14         end  
15         else begin  
16             cnt <= 0;  
17             clk_100ms <= ~clk_100ms;  
18         end  
19     end  
20  
21 endmodule
```

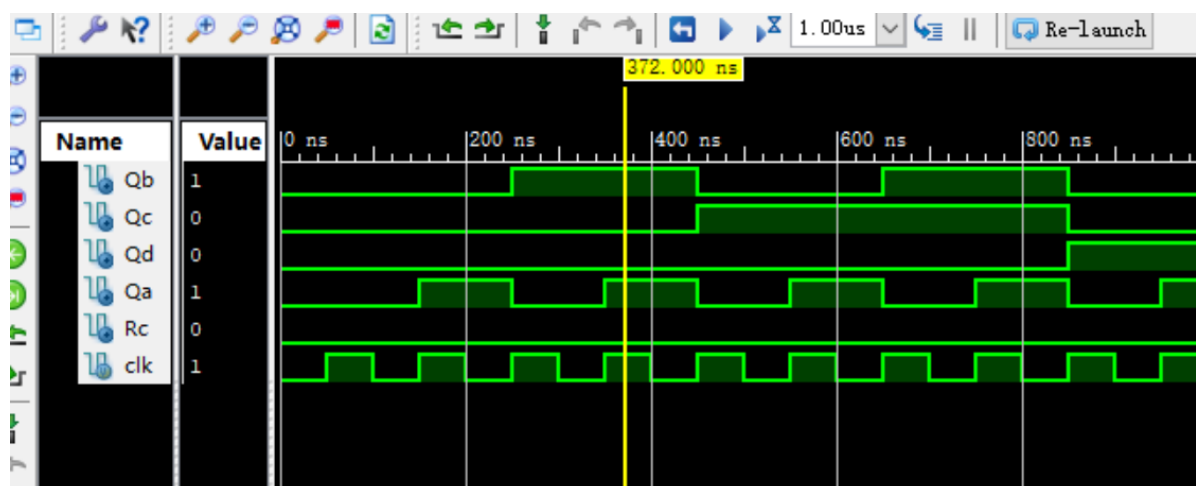
## 4. 设置引脚约束

```
1  NET "clk" LOC = AC18      | IOSTANDARD = LVCMOS18 ;
2
3  NET "sw" loc = AA10       | IOSTANDARD = LVCMOS15;
4
5  NET "segment[0]"         LOC = AB22      | IOSTANDARD = LVCMOS33;
6  NET "segment[1]"         LOC = AD24      | IOSTANDARD = LVCMOS33;
7  NET "segment[2]"         LOC = AD23      | IOSTANDARD = LVCMOS33;
8  NET "segment[3]"         LOC = Y21       | IOSTANDARD = LVCMOS33;
9  NET "segment[4]"         LOC = W20       | IOSTANDARD = LVCMOS33;
10 NET "segment[5]"         LOC = AC24      | IOSTANDARD = LVCMOS33;
11 NET "segment[6]"         LOC = AC23      | IOSTANDARD = LVCMOS33;
12 NET "segment[7]"         LOC = AA22      | IOSTANDARD = LVCMOS33;
13
14 NET "an[0]"              LOC = AD21      | IOSTANDARD = LVCMOS33;
15 NET "an[1]"              LOC = AC21      | IOSTANDARD = LVCMOS33;
16 NET "an[2]"              LOC = AB21      | IOSTANDARD = LVCMOS33;
17 NET "an[3]"              LOC = AC22      | IOSTANDARD = LVCMOS33;
18
19 NET "seg_clk" LOC = M24 | IOSTANDARD = LVCMOS33;
20 NET "seg_clrn" LOC = M20 | IOSTANDARD = LVCMOS33;
21 NET "seg_sout" LOC = L24 | IOSTANDARD = LVCMOS33;
22 NET "SEG_PEN" LOC = R18 | IOSTANDARD = LVCMOS33;
23
24 NET "LED[0]" LOC = AB26 | IOSTANDARD = LVCMOS33;
25 NET "LED[1]" LOC = W24 | IOSTANDARD = LVCMOS33;
26 NET "LED[2]" LOC = W23 | IOSTANDARD = LVCMOS33;
27 NET "LED[3]" LOC = AB25 | IOSTANDARD = LVCMOS33;
28 NET "LED[4]" LOC = AA25 | IOSTANDARD = LVCMOS33;
29 NET "LED[5]" LOC = W21 | IOSTANDARD = LVCMOS33;
30 NET "LED[6]" LOC = V21 | IOSTANDARD = LVCMOS33;
31 NET "LED[7]" LOC = W26 | IOSTANDARD = LVCMOS33;
```

## 四、实验结果与分析

### 4.1 设计4位同步二进制计数器

对四位二进制同步计数器进行仿真，波形结果如下：



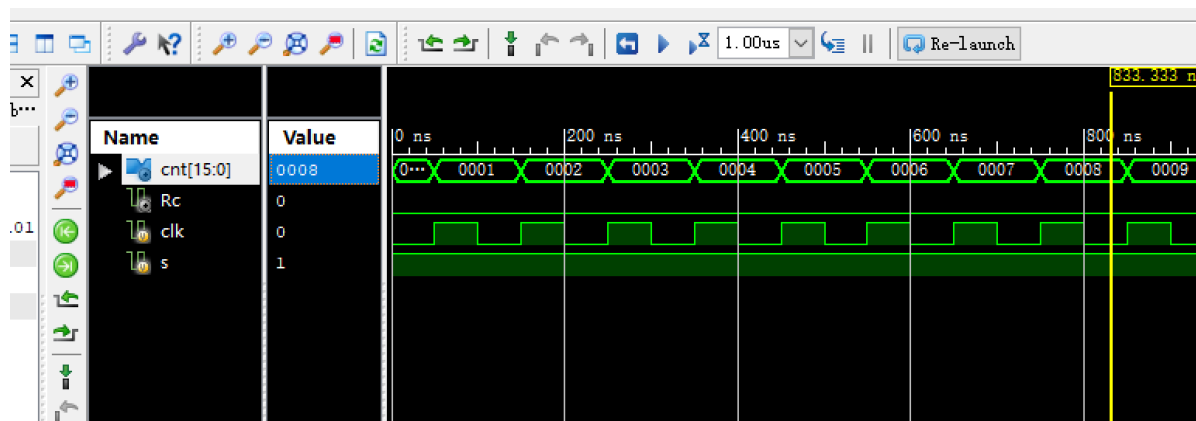
结果符合预期。



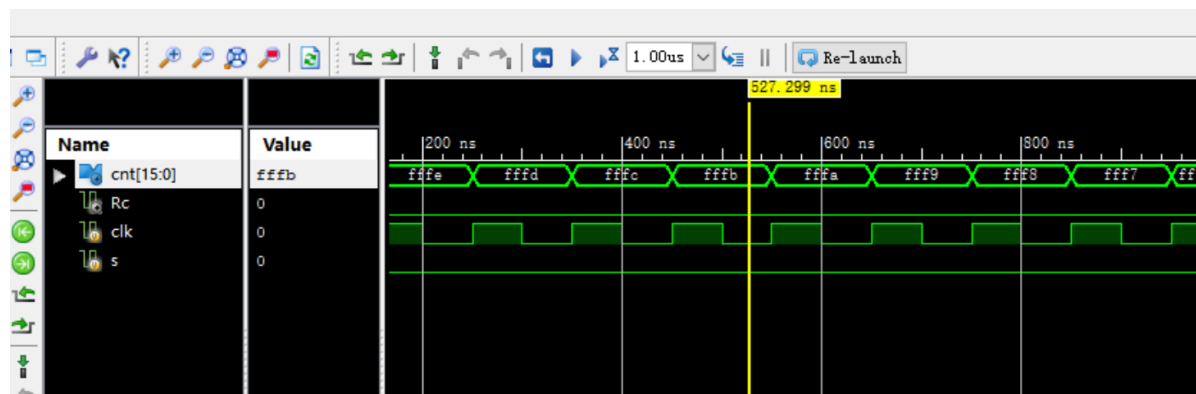
## 4.2 设计16位可逆二进制同步计数器

对16位可逆二进制同步计数器进行仿真：

- 正向计数



- 反向计数



## 五、讨论、心得

经过本次实验，我更好地理解了时钟分频的含义。另外此次实验内容比较丰富，我对 Verilog 的语法有了更好的掌握。

本次实验过程中也发现了自己在理解以及 Verilog 语法方面存在的一些问题，经过搜索和向同学提问的方式增进了理解，改正了错误。