

浙江大学

本科实验报告

课程名称： 计算机逻辑设计基础

姓 名： 刘晨

学 院： 计算机科学与技术学院

专 业： 图灵班

学 号： 3190104666

指导教师： 董亚波

2020 年 12 月 3 日

浙江大学实验报告

课程名称： 计算机逻辑设计基础

实验类型： 综合

实验项目名称： 同步时序电路设计

学生姓名： 刘晨 专业： 图灵 1901 学号： 3190104666

同组学生姓名： 林初涵 指导老师： 董亚波

实验地点： 东 4-509 实验日期： 2020 年 12 月 3 日

一、实验目的和要求

掌握典型同步时序电路的工作原理和设计方法

掌握时序电路的激励函数、状态图、状态方程的运用

掌握用 Verilog 进行有限状态机的设计、调试、仿真

掌握用 FPGA 实现时序电路功能

二、实验内容和原理

内容：

任务 1：原理图方式设计 4 位同步二进制计数器

任务 2：以 Verilog 行为描述方式设计 16 位可逆二进制同步计数器

原理：

1. 4 位二进制同步计数器

根据 D 触发器原理，在 clk 作用下 $Q = D$ ，4 位计数器的 Q 和 D 关系如右图

激励函数如下表：

$$D_A = \overline{Q_A}$$

$$D_B = \overline{Q_A}Q_B + Q_A\overline{Q_B} = \overline{Q_A} \oplus \overline{Q_B}$$

$$D_C = \overline{Q_A}Q_C + \overline{Q_B}Q_C + Q_AQ_B\overline{Q_C} \\ = (\overline{Q_A} + \overline{Q_B}) \oplus \overline{Q_C}$$

$$D_D = \overline{Q_A}Q_D + \overline{Q_B}Q_D + \overline{Q_C}Q_D + Q_AQ_BQ_C\overline{Q_D} \\ = (\overline{Q_A} + \overline{Q_B} + \overline{Q_C}) \oplus \overline{Q_D}$$

	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0
2	0	1	0	0	1	1	0	0
3	1	1	0	0	0	0	1	0
4	0	0	1	0	1	0	1	0
5	1	0	1	0	0	1	1	0
6	0	1	1	0	1	1	1	0
7	1	1	1	0	0	0	0	1
8	0	0	0	1	1	0	0	1
9	1	0	0	1	0	1	0	1
10	0	1	0	1	1	1	0	1
11	1	1	0	1	0	0	1	1
12	0	0	1	1	1	0	1	1
13	1	0	1	1	0	1	1	1
14	0	1	1	1	1	1	1	1
15	1	1	1	1	0	0	0	0

进位 RC 的输出函数如下：

$$R_C = \overline{\overline{Q_A} + \overline{Q_B} + \overline{Q_C} + \overline{Q_D}}$$

2.4 位可逆二进制同步计数器

可逆二进制同步计数器通过控制端 S 选择正向或者反向计数

S = 1 时，正向计数

S = 0 时，反向计数。各触发器逻辑表达式如下式

$$D_A = \overline{Q_A}$$

$$D_B = \overline{S}(\overline{Q_A} \oplus \overline{Q_B}) + S(\overline{Q_A} \oplus \overline{Q_B}) = \overline{S \oplus \overline{Q_A} \oplus \overline{Q_B}}$$

$$\begin{aligned} D_C &= \overline{S}[(\overline{Q_A} \overline{Q_B}) \oplus \overline{Q_C}] + S[(\overline{Q_A} + \overline{Q_B}) \oplus \overline{Q_C}] = [\overline{S \overline{Q_A} \overline{Q_B}} + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C} \\ &= [\overline{S(\overline{Q_A} + \overline{Q_B})} + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C} \end{aligned}$$

$$\begin{aligned} D_D &= \overline{S}[(\overline{Q_A} \overline{Q_B} \overline{Q_C}) \oplus \overline{Q_D}] + S[(\overline{Q_A} + \overline{Q_B} + \overline{Q_C}) \oplus \overline{Q_D}] = [\overline{S \overline{Q_A} \overline{Q_B} \overline{Q_C}} + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D} \\ &= [\overline{S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})} + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D} \end{aligned}$$

$$R = \overline{S \overline{Q_A} \overline{Q_B} \overline{Q_C} \overline{Q_D}} + S \overline{Q_A} \overline{Q_B} \overline{Q_C} \overline{Q_D} \quad (\text{进位、借位输出})$$

3.分频器

100MHz 信号通过 50,000,000 次分频后，得到 1Hz 的秒脉冲方波，作为计数器的脉冲输入

1

※



是

```

23.  wire XLXN_38;
24.  wire XLXN_39;
25.  wire XLXN_40;
26.  wire Qa_DUMMY;
27.  wire Qb_DUMMY;
28.  wire Qc_DUMMY;
29.  wire Qd_DUMMY;
30.
31.  assign Qa = Qa_DUMMY;
32.  assign Qb = Qb_DUMMY;
33.  assign Qc = Qc_DUMMY;
34.  assign Qd = Qd_DUMMY;
35.  FD #( .INIT(1'b0) ) XLXI_1 (.C(ck),
36.                                .D(XLXN_25),
37.                                .Q(Qa_DUMMY));
38.  FD #( .INIT(1'b0) ) XLXI_2 (.C(ck),
39.                                .D(XLXN_4),
40.                                .Q(Qb_DUMMY));
41.  FD #( .INIT(1'b0) ) XLXI_3 (.C(ck),
42.                                .D(XLXN_3),
43.                                .Q(Qc_DUMMY));
44.  FD #( .INIT(1'b0) ) XLXI_4 (.C(ck),
45.                                .D(XLXN_2),
46.                                .Q(Qd_DUMMY));
47.  INV XLXI_13 (.I(Qa_DUMMY),
48.               .O(XLXN_25));
49.  INV XLXI_14 (.I(Qb_DUMMY),
50.               .O(XLXN_38));
51.  INV XLXI_15 (.I(Qc_DUMMY),
52.               .O(XLXN_39));
53.  INV XLXI_16 (.I(Qd_DUMMY),
54.               .O(XLXN_40));
55.  XNOR2 XLXI_49 (.I0(XLXN_38),
56.                .I1(Qa_DUMMY),
57.                .O(XLXN_4));
58.  XNOR2 XLXI_50 (.I0(XLXN_39),
59.                .I1(XLXN_23),
60.                .O(XLXN_3));
61.  XNOR2 XLXI_51 (.I0(XLXN_40),
62.                .I1(XLXN_24),
63.                .O(XLXN_2));
64.  NOR2 XLXI_52 (.I0(XLXN_38),
65.                .I1(XLXN_25),
66.                .O(XLXN_23));

```

```

67.    NOR3  XLXI_53 (.I0(XLXN_39),
68.                .I1(XLXN_38),
69.                .I2(XLXN_25),
70.                .O(XLXN_24));
71.    NOR4  XLXI_54 (.I0(XLXN_40),
72.                .I1(XLXN_39),
73.                .I2(XLXN_38),
74.                .I3(XLXN_25),
75.                .O(Rc));
76. endmodule

```

对其进行仿真，仿真代码如下：

```

1.  initial forever begin
2.      ck = 1'b0; #20;
3.      ck = 1'b1; #20;
4.  end

```

得到的仿真结果如下：

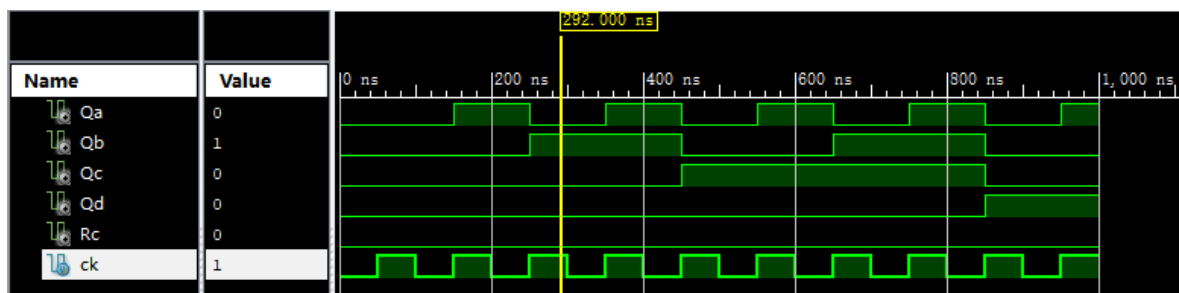


图2 仿真结果

对于 top 模块，top 模块包含以下几部分：

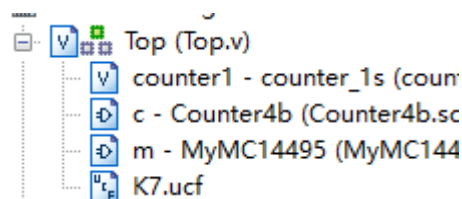


图3 top 模块内容

Counter_1s: 1s 的时钟

Counter4b: 4 位二进制同步计数器

MyMC14495: 把二进制转化为信号

Top.v 的具体 verliog 代码如下：

```

1.  module Top(
2.      input wire clk,

```

```

3.     output wire [7:0] SEGMENT,
4.     output wire rc
5. );
6.     wire clk1;
7.     wire k0,k1,k2,k3;
8.     counter_1s counter1(.clk(clk),.clk_1s(clk1));
9.     Counter4b c(.ck(clk1),.Qa(k0),.Qb(k1),.Qc(k2),.Qd(k3),.Rc(rc));
10.    MyMC14495 m(.D0(k0),.D1(k1),.D2
11.    (k2),.D3(k3),.LE(1'b0),.point(1'b0),
12.    .a(SEGMENT[0]),.b(SEGMENT[1]),.c(SEGMENT[2]),
13.    .d(SEGMENT[3]),.e(SEGMENT[4]),.f(SEGMENT[5]),
14.    .g(SEGMENT[6]),.p(SEGMENT[7]));
15.
16. endmodule

```

counter_1s 的具体 Verilog 代码如下：

```

module counter_1s(clk, clk_1s);
input wire clk;
output reg clk_1s;
reg [31:0] cnt;
always @ (posedge clk) begin
    if (cnt < 50_000_000) begin
        cnt <= cnt + 1;
    end else begin
        cnt <= 0;
        clk_1s <= ~clk_1s;
    end
end
endmodule

```

MyMC14495 原理图在前期实验当中有展示。

UCF 文件如下；

```

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 "Rc" LOC = AF24 | IOSTANDARD = LVCMOS33;

```

在开发板上面的显示结果如下：

在七段数码管上面显示的数字随着时间变换，每秒四位同时增大 1，从 0000,1111 到 ffff 进行循环。当从 ffff 跳到 0000 的时候，指示灯会闪亮一下，不断循环直

至关闭程序。

时间/s	0	1	2	3	4	5	6	15	16
显示	0000	1111	2222	3333	4444	5555	6666	FFFF	0000

任务 2：以 Verilog 行为描述方式设计 16 位可逆二进制同步计数器

16 位可逆二进制同步计数器的 Verilog 代码如下：

```

21 module RevCounter(clk,s,cnt,Rc
22     );
23     input wire clk,s;
24     output reg[15:0] cnt;
25     output wire Rc;
26     initial cnt = 0;
27     assign Rc = (~s & (~|cnt))|(s & (&cnt) );
28     always @ (posedge clk) begin
29         if(s)
30             cnt<=cnt+1;
31         else
32             cnt<=cnt-1;
33     end
34 endmodule

```

对其进行仿真，仿真代码如下：

```

44 initial forever begin
45     s=0;
46     clk = 1'b0; #100;
47     clk = 1'b1; #100;
48 end
49
50 endmodule

```

得到的仿真结果如下：

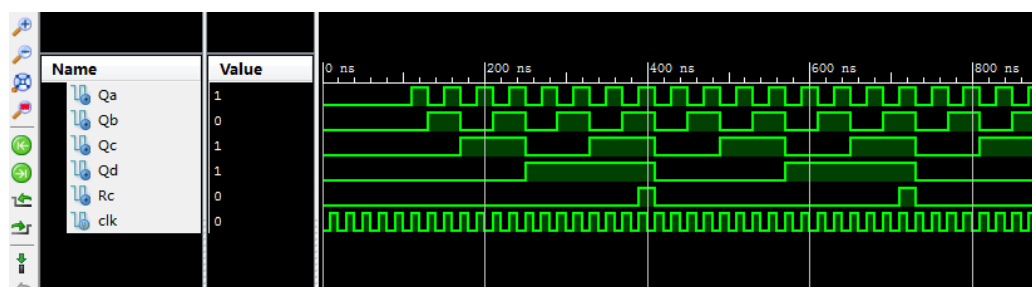


图 4 仿真结果

对于 top 模块，top 模块包含以下几部分：

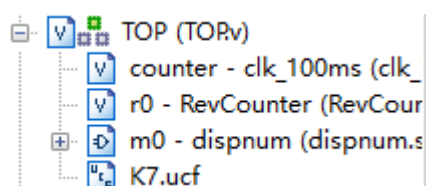


图5 top 模块内容

Clk_100ms: 100ms 的时钟

RevCounter: 16 位可逆二进制同步计数器

dispnum: 把二进制转化为信号

K7.ucf: 把数据映射到实验板上

Top.v 的具体 verliog 代码如下:

```
21 module TOP(  
22     input wire sw,  
23     input wire clk,  
24     output wire [3:0]AN,  
25     output wire [7:0]SEGMENT,  
26     output wire Rc  
27 );  
28     wire clk1;  
29     wire [15:0]cnt;  
30     //counter_1s counter1(.clk(clk),.clk_1s(clk1));  
31     clk_100ms counter(.clk(clk),.clk_1(clk1));  
32     //Counter4b c(.ck(clk1),.Qa(k0),.Qb(k1),.Qc(k2),.Qd(k3),.Rc(rc));  
33     RevCounter r0(.clk(clk1),.s(s),.cnt(cnt[15:0]),.Rc(Rc));  
34     dispnum  
35     m0(.clk(clk1),  
36         .Hexs(cnt[15:0]),  
37         .Les(4'b0)  
38         ,.points(4'b0),.  
39         rst(1'b0),.AN(AN),.Segment(SEGMENT));
```

Clk_100ms 的具体实现如下:

```
21 module clk_100ms(clk,clk_1);  
22     input wire clk;  
23     output reg clk_1;  
24     reg[31:0] cnt;  
25     initial clk_1 =0;  
26     always @ (posedge clk)begin  
27         if (cnt<5_000_000)begin  
28             cnt<=cnt+1;  
29         end else begin  
30             cnt<=0;  
31             clk_1<= ~clk_1;  
32         end  
33     end  
34 endmodule
```

Dispnum 原理图如下:

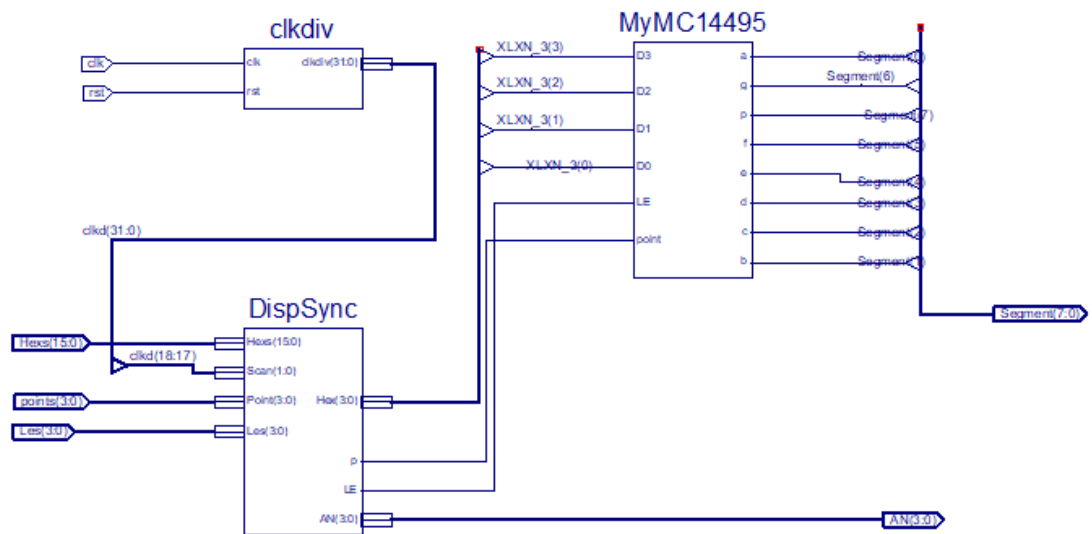


图 6 dispnum 原理图

其中 MyMC14495 原理图在前期实验当中有展示。

UCF 文件如下；

```
NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;

NET "SEGMENT[0]" LOC = AB22 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[1]" LOC = AD24 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[2]" LOC = AD23 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[3]" LOC = Y21 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[4]" LOC = W20 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[5]" LOC = AC24 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[6]" LOC = AC23 | IOSTANDARD = LVCMOS33;
NET "SEGMENT[7]" LOC = AA22 | IOSTANDARD = LVCMOS33;

NET "Rc" LOC = AF24 | IOSTANDARD = LVCMOS33;

NET "AN[0]" LOC = AD21 | IOSTANDARD = LVCMOS33;
NET "AN[1]" LOC = AC21 | IOSTANDARD = LVCMOS33;
NET "AN[2]" LOC = AB21 | IOSTANDARD = LVCMOS33;
NET "AN[3]" LOC = AC22 | IOSTANDARD = LVCMOS33;

NET "s" LOC = AA10 | IOSTANDARD = LVCMOS15;
```

在开发板上面的显示结果如下：

如果 S0 保持关闭状态：

在七段数码管上面显示的数字随着时间变换，每 0.1 秒数字增大 1，从 0000，0001 到 ffff 进行循环。当从 ffff 跳到 0000 的时候，指示灯会闪亮一下，不断循环直至关闭程序。

时间/s	0	0.1	0.2	0.3	6553.4	6553.5	6553.6
显示	0000	0001	0002	0003	FFFE	FFFF	0000

如果 S0 保持打开状态，那么数字将会随着时间变换减小，同样是每 0.1s 数字减小 1. 输出如下图：

时间/s	0	0.1	0.2	0.3	6553.4	6553.5	6553.6
显示	0000	FFFF	FFFE	FFFD	0002	0001	0000

四、实验结果分析

总体和预期实验结果内容一致。

硬件描述代码和原理图所要实现的功能一致。

仿真结果符合预期。

Verliog 代码内容实现了预期功能。

在开发板上的实验结果验证了应有的功能。

五、讨论与心得

在这次实验中，我们学会了同步时序电路设计，了解了同步时序电路设计的方法。我在实验中深刻体会到了实验当中出现的问题和解决的办法，以及协作的重要性。在这次实验当中，我在实验当中遇到了难以调出仿真波形，仿真代码的写作上面出现问题这几个问题，所幸通过老师和同学的帮助，这些问题顺利解决了，在开发板上面的实验结果也验证了这个结果。