

浙江大学



计算机逻辑设计基础

Lab 11

Author: 苏煜程

Student ID: 3220105481

Date: 2023 年 12 月 12 日

浙江大学实验报告
课程名称：计算机逻辑设计基础
实验名称：同步时序电路设计
学生姓名：苏煜程 专业：人工智能（图灵班） 学号：3220105481
指导老师：董亚波 实验地点：东 4-509 实验日期：2023 年 11 月 30 日

1 实验目的

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

2 实验任务

1. 原理图方式设计 4 位同步二进制计数器
2. 以 Verilog 行为描述方式设计 16 位可逆二进制同步计数器

3 实验原理

3.1 四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 关系如下表：

	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

其中 D_A, D_B, D_C, D_D 分别代表四个触发器。 D_A 是最低位触发器。根据 D 触发器原理，在 clk 作用下 $Q = D$ ，4 位计数器的 Q 和 D 关系如下表所示：

D_A	$Q_A Q_B$				
$Q_C Q_D$	00	01	11	10	
00	1	1			
01	1	1			
11	1	1			
10	1	1			

得到公式： $D_A = \bar{Q}_A$ 。同理可得

$$D_A = \bar{Q}_A$$

$$D_B = \bar{Q}_A Q_B + Q_A \bar{Q}_B = \overline{\bar{Q}_A \oplus \bar{Q}_B}$$

$$D_C = \bar{Q}_A Q_C + \bar{Q}_B Q_C + Q_A Q_B \bar{Q}_C = \overline{\bar{Q}_A + \bar{Q}_B} \oplus \bar{Q}_C$$

$$D_D = \bar{Q}_A Q_D + \bar{Q}_B Q_D + \bar{Q}_C Q_D + Q_A Q_B Q_C \bar{Q}_D = \overline{\overline{\bar{Q}_A + \bar{Q}_B + \bar{Q}_C}} \oplus \bar{Q}_D$$

进位 R_C 的输出函数：

$$R_C = \overline{\bar{Q}_A + \bar{Q}_B + \bar{Q}_C + \bar{Q}_D}$$

得到电路图：

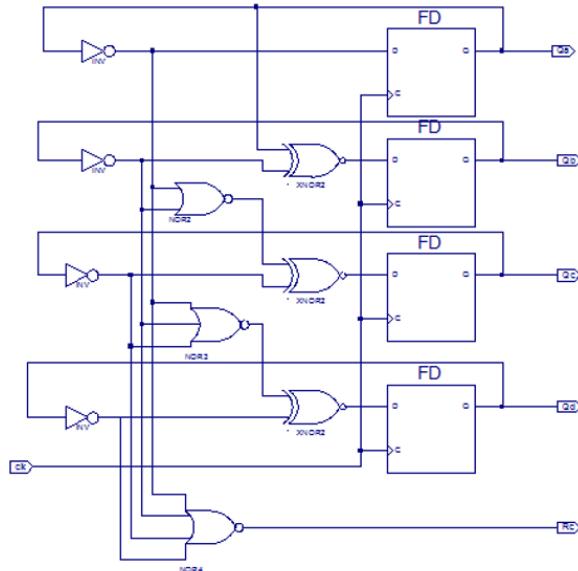


图 1：四位二进制同步计数器电路图

3.2 四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

$$\begin{aligned}D_A &= \bar{Q}_A \\D_B &= \bar{S}\overline{\bar{Q}_A \oplus \bar{Q}_B} + S\overline{Q_A \oplus \bar{Q}_B} = \overline{S \oplus \bar{Q}_A \oplus \bar{Q}_B} \\D_C &= \bar{S}(\overline{\bar{Q}_A \bar{Q}_B} \oplus \bar{Q}_C) + S(\overline{\bar{Q}_A + \bar{Q}_B} \oplus \bar{Q}_C) = \overline{[\bar{S}(Q_A + Q_B) + S(\bar{Q}_A + \bar{Q}_B)] \oplus \bar{Q}_C} \\D_D &= \overline{[\bar{S}(Q_A + Q_B + Q_C) + S(\bar{Q}_A + \bar{Q}_B + \bar{Q}_C)]} \oplus \bar{Q}_D \\R_C &= \bar{S}\bar{Q}_A\bar{Q}_B\bar{Q}_C\bar{Q}_D + S\bar{Q}_AQ_B\bar{Q}_CQ_D\end{aligned}$$

可逆二进制 4 位同步计数器的行为描述：

```
1 module counter_4bit_rev(clk, s, cnt, Rc);
2   input wire clk, s;
3   output reg [3:0] cnt;
4   output wire Rc;
5   initial cnt = 0;
6   assign Rc = (~s & (~|cnt)) | (s & (&cnt));
7   always @ (posedge clk) begin
8     if (s)
9       cnt <= cnt + 1'b1;
10    else
11      cnt <= cnt - 1'b1;
12  end
13 endmodule
```

3.3 分频器

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

```
1 module counter_1s(clk, clk_1s);
2   input wire clk;
3   output reg clk_1s;
4   reg [31:0] cnt;
5   always @ (posedge clk) begin
6     if (cnt < 50_000_000) begin
7       cnt <= cnt + 1'b1;
8     end else begin
9       cnt <= 0;
```

```
10          clk_1s <= ~clk_1s;
11      end
12  end
13 endmodule
```

4 实验内容与步骤

4.1 任务 1：原理图方式设计 4 位同步二进制计数器

1. 新建工程 MyCounter，Top Level Source Type 用 HDL
 2. 新建源文件，类型是 Schematic，文件名称用 Counter4b
 3. 原理图方式进行设计

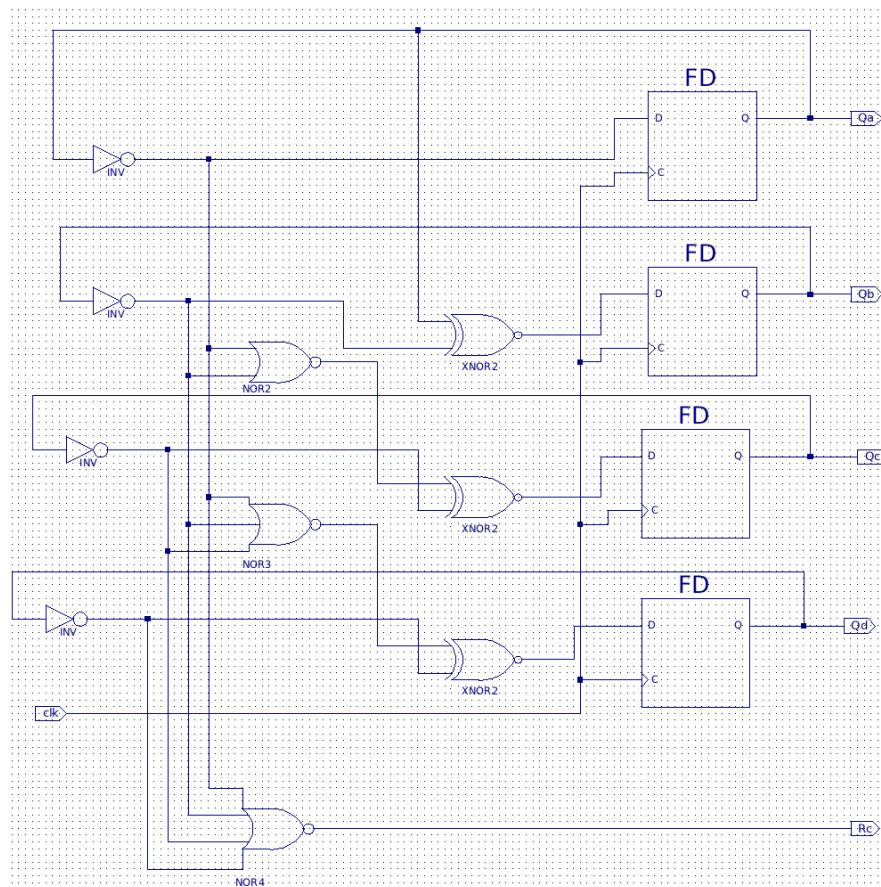


图 2: 四位二进制同步计数器原理图

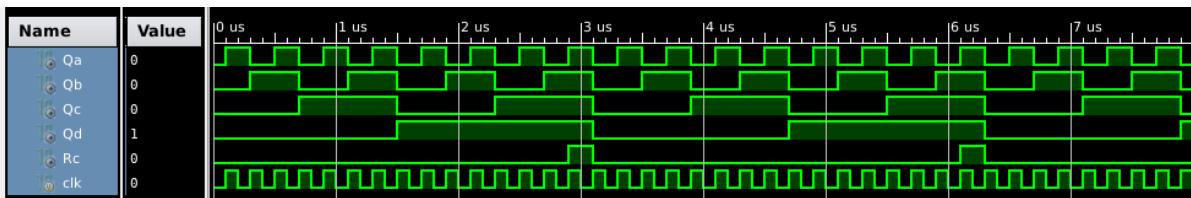
- #### 4. 进行波形仿真

```

initial forever begin
    clk = 1'b0; #100;
    clk = 1'b1; #100;
end

```

(a) 仿真激励代码



(b) 仿真结果

图 3: 四位二进制同步计数器仿真

5. 新建源文件, 用作时钟, 类型是 Verilog, 文件名称用 clk_1s, Verilog 行为描述

```

1 module clk_1s (
2     clk,
3     clk_1s
4 );
5     input wire clk;
6     output reg clk_1s;
7     reg [31:0] cnt;
8     always @ (posedge clk) begin
9         if (cnt < 50_000_000) begin
10            cnt <= cnt + 1'b1;
11        end else begin
12            cnt <= 0;
13            clk_1s <= ~clk_1s;
14        end
15    end
16 endmodule

```

6. 新建源文件, 类型是 Verilog, 文件名称用 Top, 右键设为“Set as Top Module”

7. 输入为 clk (100 MHZ) 时钟, 每秒自增 1, 显示在 1 位数码管上, Rc 显示在 LED 灯上

```

1 module Top(
2     input wire clk,
3     output wire [7:0] Segment,
4     output wire [3:0] AN,
5     output wire Rc
6 );
7

```

```

8   wire clk_1s;
9   wire [3:0] number;
10
11  clk_1s m0(.clk(clk),
12            .clk_1s(clk_1s));
13
14  Counter4b m1(.clk(clk_1s),
15                .Qa(number[0]),
16                .Qb(number[1]),
17                .Qc(number[2]),
18                .Qd(number[3]),
19                .Rc(Rc));
20
21  DispNum m2(.HEXS({4'b0, 4'b0, 4'b0, number[3:0]}),
22              .point(4'b0),
23              .LES(4'b0),
24              .AN(AN),
25              .Segment(Segment),
26              .clk(clk),
27              .RST(1'b0));
28
29  endmodule

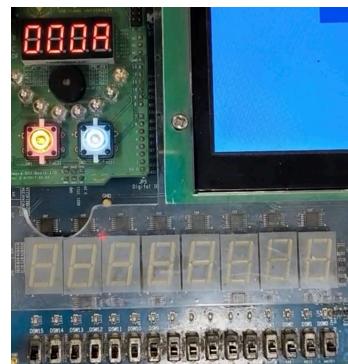
```

8. Generate Programming File 生成 bit 文件

9. 下载到 FPGA 开发板上进行实验。



(a) 9



(b) A

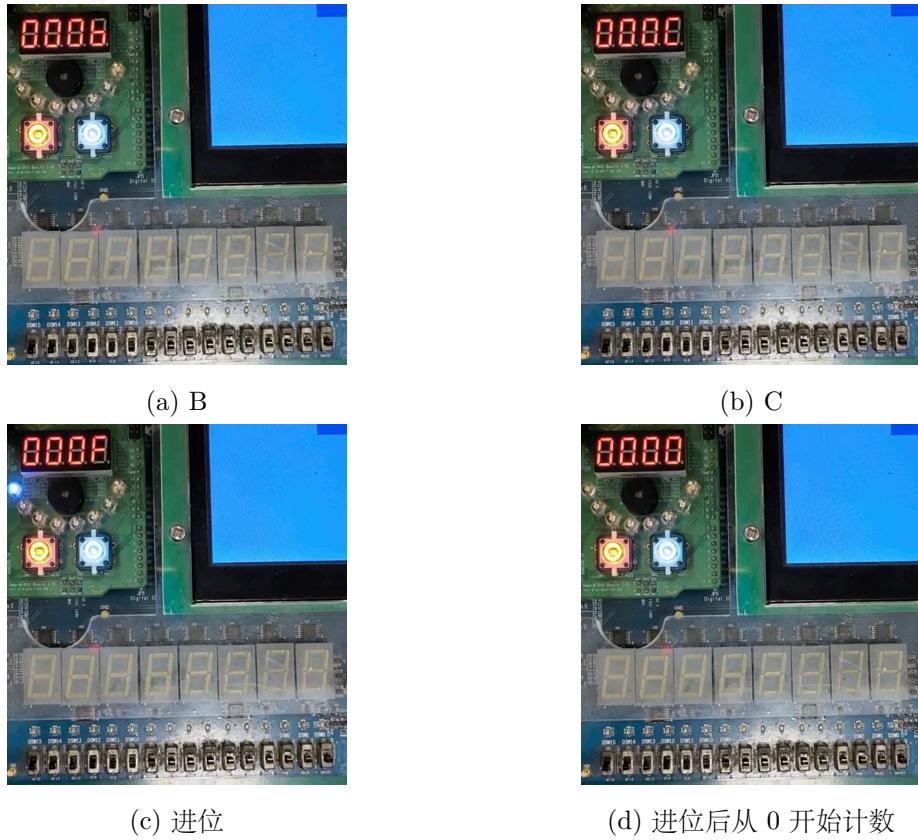


图 5: 四位二进制同步计数器

能够看到，数码管上显示的数字每秒加一，LED 灯上的 R_c 每 16 次加一。符合要求。

4.2 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

1. 新建工程 myRevCounter, Top Level Source Type 用 HDL
2. 新建源文件, 设计 16 位可逆同步二进制计数器, 类型是 Verilog, 文件名称用 RevCounter, 采用行为描述方式进行设计

```

1 module RevCounter(clk, s, cnt, Rc);
2   input wire clk, s;
3   output reg [15:0] cnt;
4   output wire Rc;
5   initial cnt = 0;
6   assign Rc = (~s & (~|cnt)) | (s & (&cnt));
7   always @ (posedge clk) begin
8     if (s) begin
9       cnt <= cnt + 1'b1;
10    end
11    else begin
12      cnt <= cnt - 1'b1;
13    end

```

```

14      end
15  endmodule

```

3. 波形仿真（包含正向计数和反向计数）

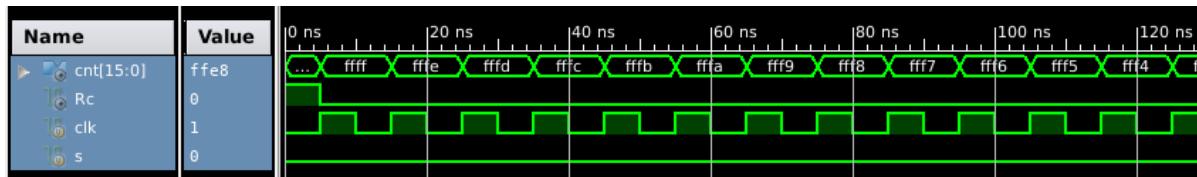
```

initial forever begin
    clk = 0; #5;
    clk = 1; #5;
end

initial begin
    s = 0; #500;
    s = 1; #500;
    s = 0;
end

```

(a) 仿真激励代码



(b) 反向计数



(c) 正向计数

图 6: 16 位可逆二进制同步计数器仿真

4. 新建源文件，设计 100 ms 时钟，类型是 Verilog，文件名称用 clk_100ms，采用 Verilog 行为描述

```

1 module clk_100ms (
2     clk,
3     clk_100ms
4 );
5
6     input wire clk;
7     output reg clk_100ms;
8     reg [31:0] cnt;
9
10    always @ (posedge clk) begin
11        if (cnt < 50_000_00) begin
12            cnt <= cnt + 1'b1;

```

```

13     end else begin
14         cnt <= 0;
15         clk_100ms <= ~clk_100ms;
16     end
17 end
18 endmodule

```

5. 新建源文件, 设计顶层模块, 类型是 Verilog, 文件名称用 Top, 右键设为“Set as Top Module”, Top 模块的输入为 clk (100 MHZ) 时钟, RevCounter 模块的时钟输入为 100 ms 时钟, 用 sw[0] 控制自增/自减 1 (每 0.1 s), 计数结果显示在 4 位数码管上, Rc 状态用 LED 灯来显示

```

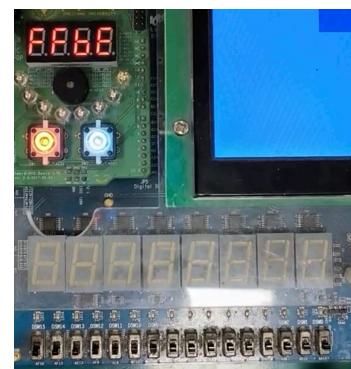
1 module Top(
2     input wire clk,
3     input wire S,
4     output wire [7:0] Segment,
5     output wire [3:0] AN,
6     output wire Rc
7 );
8
9     wire clk_100ms;
10    wire [15:0] number;
11
12    clk_100ms m0(.clk(clk),
13                  .clk_100ms(clk_100ms));
14
15    RevCounter m1(.clk(clk_100ms),
16                  .s(S),
17                  .cnt(number),
18                  .Rc(Rc));
19
20    DispNum m2(.HEXS(number),
21                  .point(4'b0),
22                  .LES(4'b0),
23                  .AN(AN),
24                  .Segment(Segment),
25                  .clk(clk),
26                  .RST(1'b0));
27
28 endmodule

```

7. Generate Programming File 生成 bit 文件
8. 下载到 FPGA 开发板上进行实验。



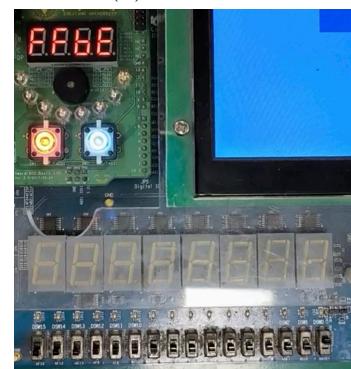
(a) 反向计数



(b) 反向计数



(c) 正向计数



(d) 正向计数



(e) 进位

图 7: 16 位可逆二进制同步计数器

能够看到，数码管上显示的数字每 0.1 s 加一或减一，LED 灯上的 Rc 每次进位时加一。符合要求。

5 实验结果分析

实验结果在前一节已经分析，符合实验要求。

6 讨论与心得

本次实验相对比较简单，任务量不大。这次实验并没有给 top 文件的参考代码，让我加深了解了模块的调用，对之前所写的模块的使用方法有了更深的理解。