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：仇国智

学　　院：竺可桢学院

专　　业：计算机科学与技术

学　　号：3220102181

指导教师：董亚波

2023 年 11 月 10 日

# 一、 实验目的和要求

1　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

2　实验任务

2.1　任务 1：原理图方式设计 4 位同步二进制计数器

2.2　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

# 二、 实验内容和原理

1　实验内容

1.1　任务 1：原理图方式设计 4 位同步二进制计数器

1.2　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

2　实验原理

2.1　构 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 |

根据真值表化简得表达式：

$$D_A = \overline{Q_A}$$

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

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

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

2.2 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：

$$D_A = \overline{Q_A}$$

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

$$D_C = \overline{S}[\overline{(\overline{Q_A}\,\overline{Q_B}) \oplus \overline{Q_C}}] + S[\overline{(\overline{Q_A} + \overline{Q_B}) \oplus \overline{Q_C}}] = \overline{[\overline{S\overline{Q_A}\,\overline{Q_B}} + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C}}$$

$$= \overline{[\overline{S}(Q_A + Q_B) + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C}}$$

$$D_D = \overline{S}[\overline{(\overline{Q_A}\,\overline{Q_B}\,\overline{Q_C}) \oplus \overline{Q_D}}] + S[\overline{(\overline{Q_A} + \overline{Q_B} + \overline{Q_C}) \oplus \overline{Q_D}}] = \overline{[\overline{S\overline{Q_A}\,\overline{Q_B}\,\overline{Q_C}} + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D}}$$

$$= \overline{[\overline{S}(Q_A + Q_B + Q_C) + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D}}$$

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

2.3 分频器设计：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。

## 三、 实验过程和数据记录

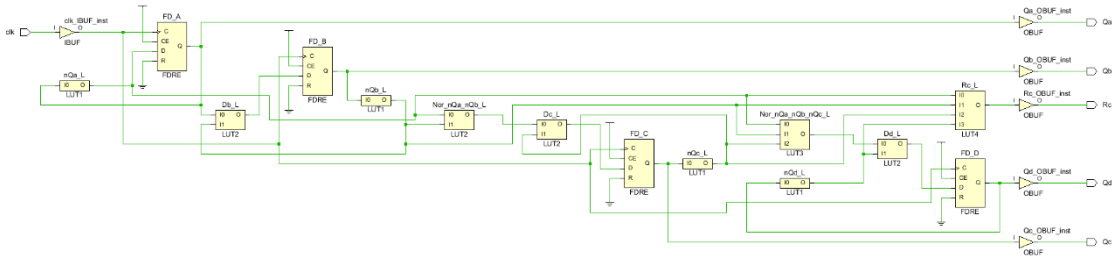
1 实任务 1：原理图方式设计 4 位同步二进制计数器

1.1 新建工程，工程名称用 MyCounter。Top Level Source Type 用 HDL。新建源文件，类型是 Schematic。文件名称用 Counter4b。原理图方式进行设计，进行波形仿真。

设计：

```
`timescale 1ns / 1ps
module counter_4b(clk, Qa, Qb, Qc, Qd, Rc);
input wire clk;
output wire Qa, Qb, Qc, Qd, Rc;
wire Nor_nQa_nQb, Nor_nQa_nQb_nQc;

FD FD_A(.C(clk), .D(Da), .Q(Qa)),
   FD_B(.C(clk), .D(Db), .Q(Qb)),
   FD_C(.C(clk), .D(Dc), .Q(Qc)),
   FD_D(.C(clk), .D(Dd), .Q(Qd));
defparam FD_A.INIT = 1'b0, FD_B.INIT = 1'b0;
defparam FD_C.INIT = 1'b0, FD_D.INIT = 1'b0;

INV nQa_L(.I(Qa), .O(nQa)), nQb_L(.I(Qb), .O(nQb)),
    nQc_L(.I(Qc), .O(nQc)), nQd_L(.I(Qd), .O(nQd));
```

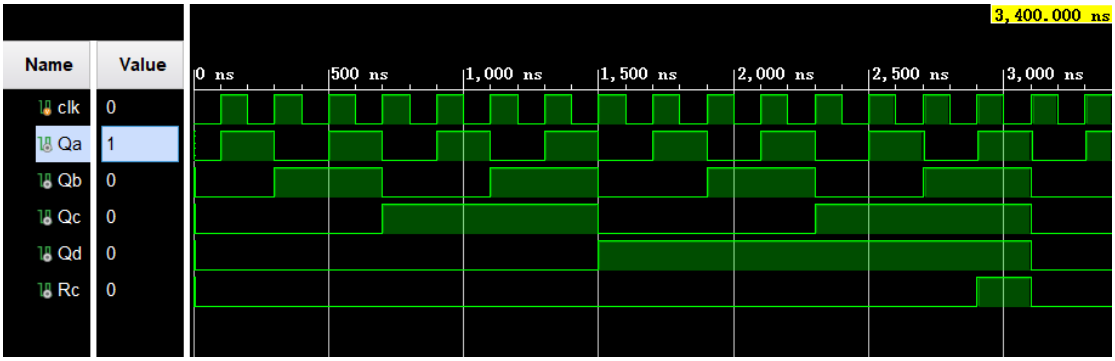
```
assign Da = nQa;

XNOR2 Db_L(.I0(Qa),              .I1(nQb), .O(Db)),
      Dc_L(.I0(Nor_nQa_nQb),     .I1(nQc), .O(Dc)),
      Dd_L(.I0(Nor_nQa_nQb_nQc), .I1(nQd), .O(Dd));
NOR4  Rc_L(.I0(nQa), .I1(nQb), .I2(nQc), .I3(nQd), .O(Rc));
NOR2  Nor_nQa_nQb_L    (.I0(nQa), .I1(nQb), .O(Nor_nQa_nQb));
NOR3  Nor_nQa_nQb_nQc_L(.I0(nQa), .I1(nQb), .I2(nQc), .O(Nor_nQa_nQb_nQc
));
endmodule
```



仿真：

```
`timescale 1ns / 1ps
module counter4b_test1 ();
  wire Qa, Qb, Qc, Qd, Rc;
  reg clk;
  counter_4b test1 (
      clk,
      Qa,
      Qb,
      Qc,
      Qd,
      Rc
  );
  initial
    forever begin
      clk = 1'b0;
      #100;
      clk = 1'b1;
      #100;
    end
```

```
endmodule
```



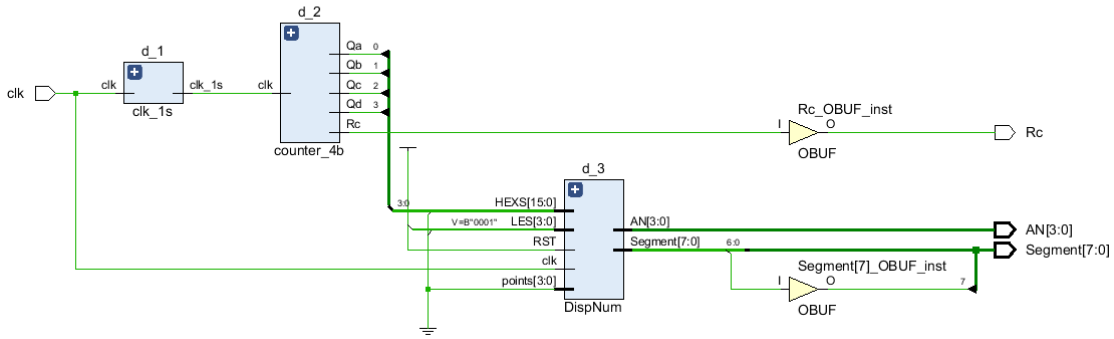

1.2 设计时钟 clk 模块

```
module clk_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'b1;
    end else begin
      cnt <= 0;
      clk_1s <= ~clk_1s;
    end
  end
endmodule
```

1.3 新建源文件，类型是 Verilog，文件名称用 Top。右键设为“Set as Top Module”。输入为 clk（100MHZ）时钟，每秒自增 1，显示在 1 位数码管上，Rc 显示在 LED 灯上

```
module top1 (
    clk,
    Segment,
    AN,
    Rc
);
  input wire clk;
  output wire [7:0] Segment;
```

```
  output wire [3:0] AN;
  output wire Rc;
  wire[15:0] HEXS;
  wire clk_1;
  clk_1s d_1(clk,clk_1);
  counter_4b d_2(clk_1,HEXS[0],HEXS[1],HEXS[2],HEXS[3],Rc);
  DispNum d_3(clk,HEXS,4'b1110,4'b0000,0,AN,Segment);
endmodule
```

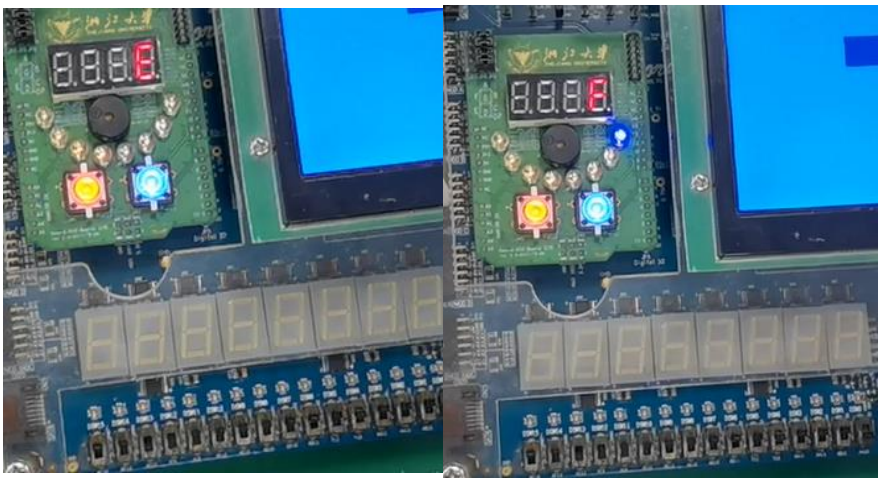



## 1.4 编写引脚约束文件

```
set_property PACKAGE_PIN AB22 [get_ports {Segment[0]}]
set_property PACKAGE_PIN AD24 [get_ports {Segment[1]}]
set_property PACKAGE_PIN AD23 [get_ports {Segment[2]}]
set_property PACKAGE_PIN Y21 [get_ports {Segment[3]}]
set_property PACKAGE_PIN W20 [get_ports {Segment[4]}]
set_property PACKAGE_PIN AC24 [get_ports {Segment[5]}]
set_property PACKAGE_PIN AC23 [get_ports {Segment[6]}]
set_property PACKAGE_PIN AA22 [get_ports {Segment[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[0]}]
```

```
set_property PACKAGE_PIN AD21 [get_ports {AN[0]}]
set_property PACKAGE_PIN AC21 [get_ports {AN[1]}]
set_property PACKAGE_PIN AB21 [get_ports {AN[2]}]
set_property PACKAGE_PIN AC22 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]

set_property PACKAGE_PIN AC18 [get_ports clk]
set_property IOSTANDARD LVCMOS18 [get_ports clk]

set_property PACKAGE_PIN AF24 [get_ports Rc]
set_property IOSTANDARD LVCMOS33 [get_ports Rc]
```

1.5 下载至实验板，检验功能。



# 2　设计 16 位可逆同步二进制计数器

## 2.1 设计计数器模块，并进行仿真

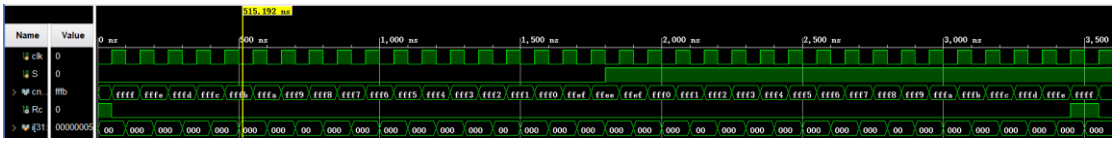
设计：

```
module counter_4b_rev(clk, s, cnt, Rc);
input wire clk, s;
output reg [15:0] cnt;
output wire Rc;
initial cnt = 0;
assign Rc = (~s & (~|cnt)) | (s & (&cnt));
always @ (posedge clk) begin
```

```
    if (s)
        cnt <= cnt + 1'b1;
    else
        cnt <= cnt - 1'b1;
end
endmodule
```

仿真：

```
`timescale 1ns / 1ps
module counter_4b_rev_test1(

    );
    reg clk,S;
    wire [15:0]cnt;
    wire Rc;
    integer i;
    counter_4b_rev test1(clk,S,cnt,Rc);
    initial begin
        S=0;
        for(i=0;i<18;i=i+1)begin
            clk=0;
            #50;
            clk=1;
            #50;
        end
        S=1;
        for(i=0;i<18;i=i+1)begin
            clk=0;
            #50;
            clk=1;
            #50;
        end
    end

endmodule
```



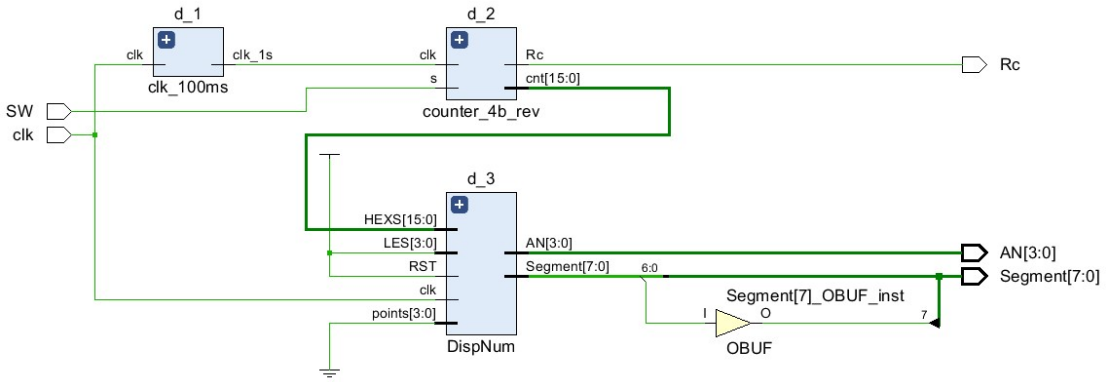

## 2.2 设计时钟模块

```
module clk_100ms (
    clk,
    clk_1s
);
  input wire clk;
  output reg clk_1s;
  reg [31:0] cnt;
  always @(posedge clk) begin
    if (cnt < 50_000_00) begin
      cnt <= cnt + 1'b1;
    end else begin
      cnt <= 0;
      clk_1s <= ~clk_1s;
    end
  end
endmodule
```

## 2.3 设计 top 模块

```
module top2 (
    clk,
    SW,
    Segment,
    AN,
    Rc
);
  input wire clk,SW;
  output wire [7:0] Segment;
  output wire [3:0] AN;
  output wire Rc;
  wire[15:0] HEXS;
  wire clk_1;
  clk_100ms d_1(clk,clk_1);
  counter_4b_rev d_2(clk_1,SW,HEXS[15:0],Rc);
  DispNum d_3(clk,HEXS[15:0],4'b0000,4'b0000,0,AN,Segment);
```

```
endmodule
```




## 2.4 编写引脚约束文件

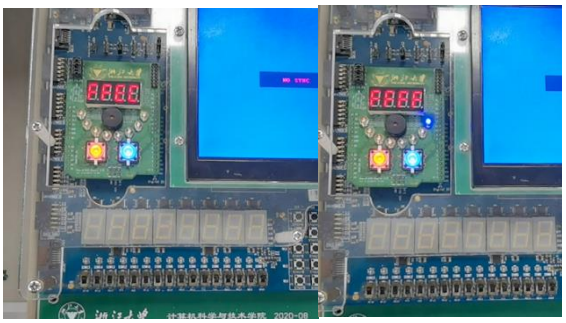
```
set_property PACKAGE_PIN AB22 [get_ports {Segment[0]}]
set_property PACKAGE_PIN AD24 [get_ports {Segment[1]}]
set_property PACKAGE_PIN AD23 [get_ports {Segment[2]}]
set_property PACKAGE_PIN Y21 [get_ports {Segment[3]}]
set_property PACKAGE_PIN W20 [get_ports {Segment[4]}]
set_property PACKAGE_PIN AC24 [get_ports {Segment[5]}]
set_property PACKAGE_PIN AC23 [get_ports {Segment[6]}]
set_property PACKAGE_PIN AA22 [get_ports {Segment[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[0]}]

set_property PACKAGE_PIN AD21 [get_ports {AN[0]}]
set_property PACKAGE_PIN AC21 [get_ports {AN[1]}]
set_property PACKAGE_PIN AB21 [get_ports {AN[2]}]
set_property PACKAGE_PIN AC22 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]

set_property PACKAGE_PIN AC18 [get_ports clk]
```

```
set_property IOSTANDARD LVCMOS18 [get_ports clk]

set_property PACKAGE_PIN AA10 [get_ports SW]
set_property IOSTANDARD LVCMOS15 [get_ports SW]

set_property PACKAGE_PIN AF24 [get_ports Rc]
set_property IOSTANDARD LVCMOS33 [get_ports Rc]
```

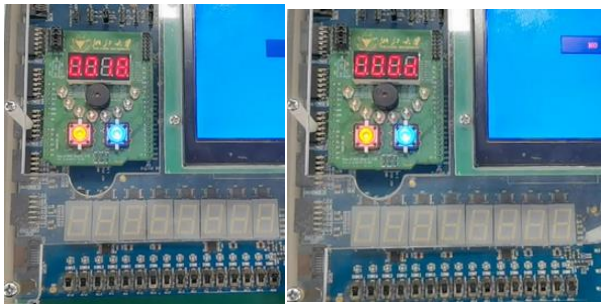
2.5 下载至实验板

实验结果：

数字递增至 FFFF 亮灯：



数字进位递增至 0018，拨动 SW[0]，数字递减至 000d



四、 实验结果分析

仿真波形符合预期，数字随时间每个周期加一。

实际测验结果符合预期，实现了数字递增递减得效果。

五、 实验心得体会

调用先前模块时需注意输入参数位置，和使能是 1 还是 0。