

# 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:	李瀚轩
学院:	竺可桢学院
系:	所在系
专业:	计算机科学与技术
学号:	3220106039
指导教师:	董亚波

2023 年 12 月 5 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: 李瀚轩 专业: 计算机科学与技术 学号: 3220106039

同组学生姓名: 郑涵文 指导老师: 董亚波

实验地点: 东四 509 实验日期: 2023 年 11 月 30 日

## 一、实验目的和要求

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

## 二、实验内容和原理

### 2.1 实验内容

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

## 2.2 实验原理

### 2.2.1 4 位二进制同步计数器

1. 计数器在数字系统中主要是对脉冲的个数进行计数，以实现测量、计数和控制的功能，同时兼有分频功能。
2. 同步计数器指的是被测量累计值，其特点是大大提高了计数器工作频率。
3. 四位二进制同步计数器由四个触发器组成，各位触发器的时钟脉冲输入端 C 接同一个计数脉冲 Clk。设初状态从 0000 开始，每输入一个计数脉冲 Clk，最低触发端就会翻转一次，而其它触发器仅在  $C = 1$  的条件下，在 Clk 前沿翻转一次。真值表如下：

	$Q_A$	$Q_B$	$Q_C$	$Q_D$	$D_A$	$D_B$	$D_C$	$D_D$
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

图 1: 四位二进制同步计数器真值表

其中  $D_A, D_B, D_C, D_D$  分别代表四个触发器。 $D_A$  是最低端触发器。根据原理我

们可以得到激励函数和  $R_C$  进位输出函数。

## 2.2.2 可逆二进制同步计数器

四位二进制同步可逆计数器是在前面四位二进制同步计数器的基础上增加控制电路和减计数器构成的。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， $S=0$  则反向计数

$$\begin{aligned}
 D_A &= \overline{Q_A} \\
 D_B &= \overline{S}(\overline{Q_A} \oplus \overline{Q_B}) + S(\overline{Q_A} \oplus \overline{Q_B}) = \overline{S} \oplus \overline{Q_A} \oplus \overline{Q_B} \\
 D_C &= \overline{S}[(\overline{Q_A} \overline{Q_B}) \oplus \overline{Q_C}] + S[(\overline{Q_A} + \overline{Q_B}) \oplus \overline{Q_C}] = [\overline{S} \overline{Q_A} \overline{Q_B} + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C} \\
 &= [\overline{S}(\overline{Q_A} + \overline{Q_B}) + S(\overline{Q_A} + \overline{Q_B})] \oplus \overline{Q_C} \\
 D_D &= \overline{S}[(\overline{Q_A} \overline{Q_B} \overline{Q_C}) \oplus \overline{Q_D}] + S[(\overline{Q_A} + \overline{Q_B} + \overline{Q_C}) \oplus \overline{Q_D}] = [\overline{S} \overline{Q_A} \overline{Q_B} \overline{Q_C} + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D} \\
 &= [\overline{S}(\overline{Q_A} + \overline{Q_B} + \overline{Q_C}) + S(\overline{Q_A} + \overline{Q_B} + \overline{Q_C})] \oplus \overline{Q_D} \\
 R_C &= \overline{S} \overline{Q_A} \overline{Q_B} \overline{Q_C} \overline{Q_D} + S \overline{Q_A} \overline{Q_B} \overline{Q_C} \overline{Q_D} \quad (\text{进位、借位输出})
 \end{aligned}$$

图 2: 各触发器逻辑表达式

## 2.2.3 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、实验过程与数据记录

## 3.1 原理图方式设计 4 位同步二进制计数器

### 3.1.1 原理图绘制

1. 新建工程
2. 新建源文件 Counter4b.sch
3. 原理图方式进行设计，电路图如下：

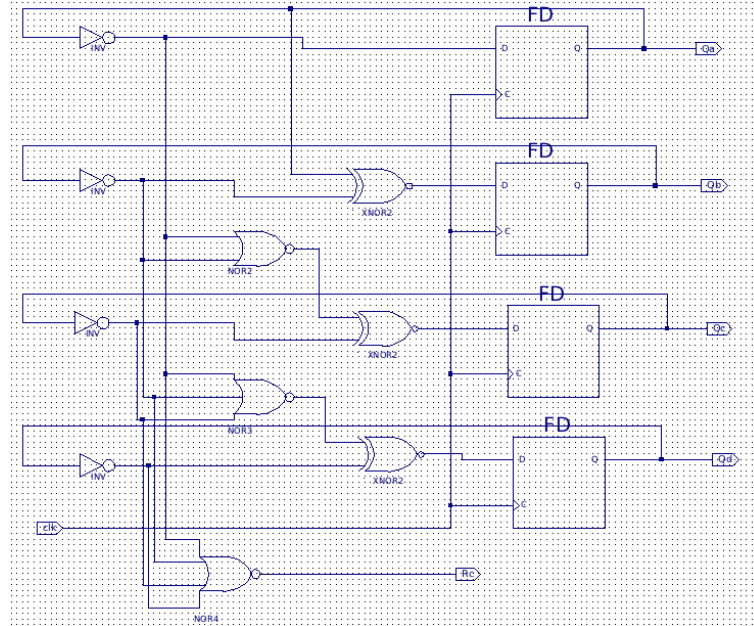


图 3: 原理图

4. 仿真新建仿真测试文件，输入如下仿真代码：

```
module Counter4b_Counter4b_sch_tb ();

// Inputs
reg clk;

// Output
wire Qb;
wire Qc;
wire Qd;
wire Qa;
wire Rc;

// Bidirs

// Instantiate the UUT
Counter4b UUT (
    .Qb(Qb),
    .Qc(Qc),
    .Qd(Qd),
```

```

        .Qa(Qa) ,
        .Rc(Rc) ,
        .clk( clk)

    );
// Initialize Inputs
    initial forever begin
        clk=1'b0;#100;
        clk=1'b1;#100;
    end
endmodule

```

得到如下仿真波形图（由于波形过长，就截取部分）

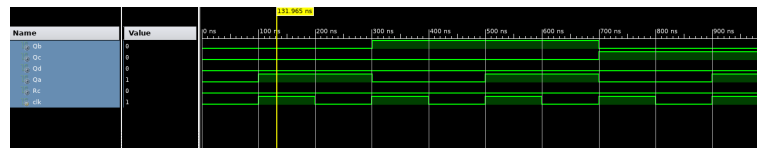


图 4: 仿真波形图

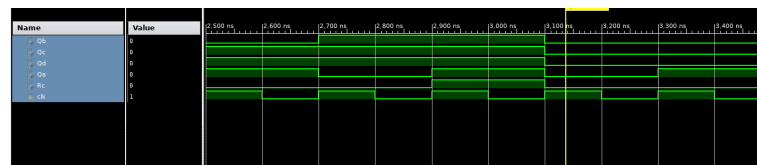


图 5: 仿真波形图 (含进位)

仿真波形中， $(Q_a, Q_b, Q_c, Q_d)$  从  $(0,0,0,0)$  依次变化到  $(1,1,1,1)$ . 在最后一进位之前  $R_c$  为 1，即产生进位。结果符合预期。

### 3.1.2 时钟行为设计

新建源文件用作时钟，代码如下：

```

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;
        end
        else begin
            cnt <= 0;
            clk_1s <= ~clk_1s;
        end
    end

endmodule

```

### 3.1.3 顶层模块设计

1. 新建 verilog 类型源文件，设置为 Top Module，预计输入 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
2. 导入 DispNum 模块之后，在 top.v 里进行 verilog 行为描述，代码如下：

```

module top(
    input wire clk,
    output wire [7:0] Segment,
    output wire [3:0] AN,
    output wire led
);

    wire clk_1s;
    wire [3:0] num;
    clk_1s m0(.clk(clk), .clk_1s(clk_1s));
    Counter4b m1(.clk(clk_1s), .Qa(num[0]), .Qb(num[1]), .Qc(num[2]), .Qd(num[3]));
    DispNum m2(.clk(clk), .HEXS({num[3:0], 4'b0000, 4'b0000, 4'b0000}));

endmodule

```

3. 新建引脚约束文件 k7.ucf，代码如下

```

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 "led" LOC=W23 | IOSTANDARD=LVCMOS33;
NET "AN[3]" LOC=AC22 | IOSTANDARD=LVCMOS33;
NET "AN[2]" LOC=AB21 | IOSTANDARD=LVCMOS33;
NET "AN[1]" LOC=AC21 | IOSTANDARD=LVCMOS33;
NET "AN[0]" LOC=AD21 | IOSTANDARD=LVCMOS33;

```

#### 4. 生成.bit 文件，下载到 SWORD 板上验证

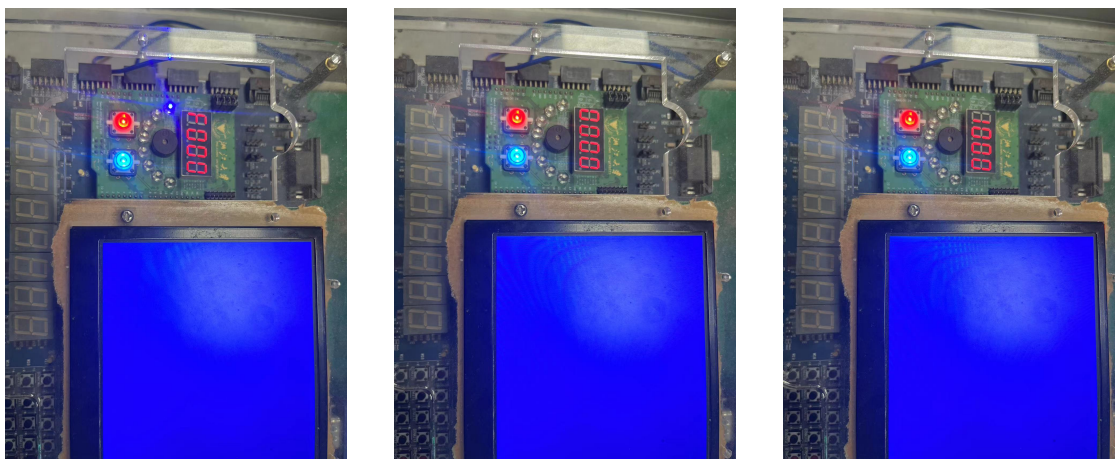


图 6: 4 位同步二进制计数器上板结果

计数器最左边的数字每秒加一，在示数为 F 时灯会亮表示进位，符合预期。这里只列出三张图片，上板结果正确性已在验收时得到验证。

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

### 3.2.1 16 位可逆计数器

#### 1. 新建工程



2. 新建 myRevCounter.v 源文件，采用行为描述方式进行设计代码如下：

```
module RevCounter(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
```

3. 波形仿真 (包含正向计数和反向计数)，仿真代码如下：

```
module RevCounter_sim;

// Inputs
reg clk;
reg s;

// Outputs
wire [15:0] cnt;
wire Rc;

// Instantiate the Unit Under Test (UUT)
RevCounter uut (
    .clk(clk),
    .s(s),
    .cnt(cnt),
    .Rc(Rc)
);
```

```

initial forever begin
    clk = 1'b0;#10;
    clk = 1'b1;#10;
end
initial begin
    s=1'b0;#80;
    s=1'b1;#80;
    s=1'b0;
end

endmodule

```

得到如下波形图

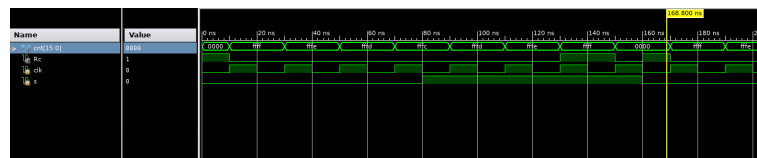


图 7: 仿真波形图

可以看到，最开始  $S=0$  时计数器反向计数， $S=1$  后变为正向计数，在计数器达到 fff 时  $R_c = 1$  代表进位。波形图符合预期。

### 3.2.2 时钟行为设计

新建源文件，设计 100ms 时钟，类型 Verilog。将 clk\_1s 中的 50\_000\_000 改为 50\_000\_00 即可，时钟由 1s 变为原来的十分之一。代码如下：

```

module clk_100ms( clk ,clk_100ms );

    input wire clk ;
    output reg clk_100ms ;
    reg [31:0] cnt ;
    always @ (posedge clk) begin
        if (cnt<50_000_00) begin
            cnt<=cnt+1;
        end else begin

```

```

        cnt <= 0;
        clk_100ms <= ~clk_100ms;
    end
end
endmodule

```

### 3.2.3 顶层模块设计

1. 新建源文件，设计顶层模块，右键设为 top module top 模块的输入为 clk(100MZ) 时钟，RevCounter 模块的时钟输入为 100ms 时钟，用 sw[0] 控制自增/自减 1，计数结果显示在 4 位数码管上，Rc 状态用 LED 灯显示
2. 将 DispNum add source，写入如下代码：

```

    module top(
    input wire clk,
    input wire S,
    output wire [7:0] Segment,
    output wire [3:0] AN,
    output wire led
    );

        wire clk_100ms;
        wire [15:0] num;
        clk_100ms m0(.clk(clk), .clk_100ms(clk_100ms));
        RevCounter m1(.clk(clk_100ms), .s(S), .cnt(num), .Rc(led));

        DispNum m2(.clk(clk), .HEXS(num), .RST(1'b0), .AN(AN), .Segment
endmodule

```

3. 创建引脚约束文件，代码如下：

```

NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
NET "Segment[0]" LOC = AB22 | IOSTANDARD = LVCMOS33;#a
NET "Segment[1]" LOC = AD24 | IOSTANDARD = LVCMOS33;#b
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;#g
NET "Segment[7]" LOC = AA22 | IOSTANDARD = LVCMOS33;#point
NET "led" LOC = W23 | IOSTANDARD = LVCMOS33;
NET "AN[3]" LOC = AC22 | IOSTANDARD = LVCMOS33;
NET "AN[2]" LOC = AB21 | IOSTANDARD = LVCMOS33;
NET "AN[1]" LOC = AC21 | IOSTANDARD = LVCMOS33;
NET "AN[0]" LOC = AD21 | IOSTANDARD = LVCMOS33;
NET "S" LOC = AA10 | IOSTANDARD = LVCMOS15;

```

#### 4. 下载到 SWORD 板上验证

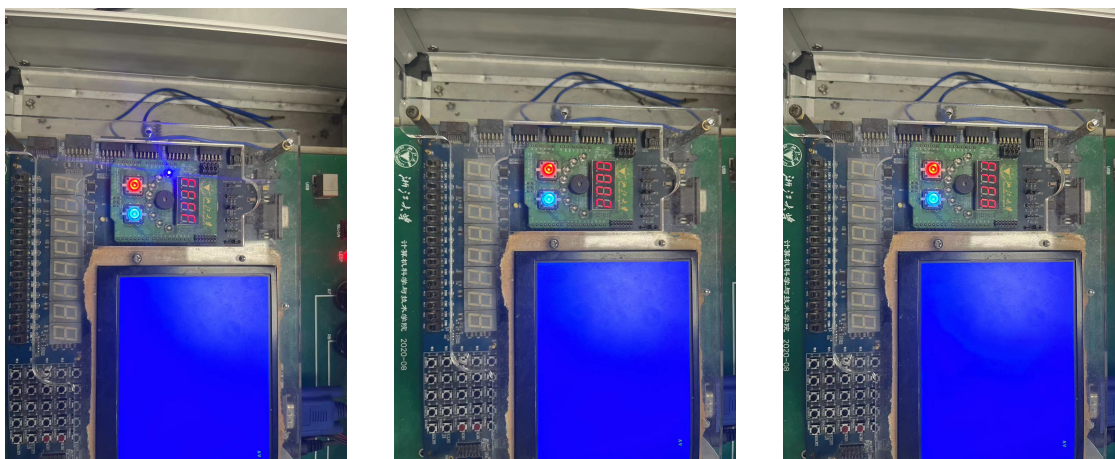


图 8: 4 位同步二进制计数器上板结果

在 Program 后示数开始自减，开关为 1 时自增。上图展现了进位前后的三个状态。递增时 FFFF 时进位灯亮，递减时 0000 进位灯亮。由于图片无法显示自增自减的过程，但是上板结果正确性在验收时得到验证，结果符合预期。

## 四、实验结果分析

相关结果都已经在前文写出。实验结果基本符合要求：仿真激励波形与真值表都相对应；下载到 sword 板上后，结果都与真值表相符。仿真结果和 Verilog 代码分析在前文已经给出。

其中与往常略有不同的是在第二个实验的仿真波形中可以调整数字显示的形式，从二进制变为十六进制，方便观察。

## 五、讨论与心得

这次实验完成了两种计数器的实现，让我更好理解了触发器在电路中的应用与作用。这次实验行为描述设计的部分增多了，我对 verilog 的熟练程度也加深了。在第一个实验生成 .bit 文件时，报了 ERROR。在同学的帮助下，我学会了通过查看 RTL Schematic 来检查我的模块是否正确连接，而不是一直盯着报错看。除此之外，一切顺利，也较早通过了验收。总而言之，继续努力！