

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：龙永奇

学 院：计算机科学与技术学院

系：本系

专 业：计算机科学与技术

学 号：3220105907

指导教师：董亚波

2023 年 12 月 9 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：____ 龙永奇 ____ 专业：____ 计算机科学与技术 ____ 学号：____ 3220105907 ____

同组学生姓名：____ 贾一多 ____ 指导老师：____ 董亚波 ____

实验地点：____ 东 4-509 ____ 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

内容：

1. 任务 1：原理图方式设计 4 位同步二进制计数器
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

原理：

1. 4 位二进制同步计数器

计数器即实现计数的逻辑电路，其核心是对时钟脉冲个数进行计数，实现测量、控制以及分频功能，其基本构成单位计数单元由各类触发器构成，常见触发器有 SR、D、JK、T 触发器等，并且在实验 10 已经实现并对部分触发器进行了模拟仿真。

计数器可分为同步计数器和异步计数器，本次实验的目标为实现同步加法计数器和同步可逆计数器。对于同步计数器，其时钟脉冲同时作用与各触发器，因此避免了异步计数器面临的逐级延迟问题，提高工作效率的同时降低了错误率，但是会使得计数器脉冲负载随着位数的增加而大大加重。

4 位二进制同步计数器由四个触发器组成，每个触发器的 C 接入计

数脉冲 clk，初始状态为`0000`，每输入一个时钟脉冲就从最底端触发器翻转一次，其他触发器为下降沿反转，真值表如图：

	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 2.1 4 位二进制触发器真值表

可见 DA，DB，DC，DD 为四个触发器，其中 DA 为最底端触发器
 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		$Q_A Q_B$			
		00	01	11	10
$Q_C Q_D$	00	1	1		
	01	1	1		
	11	1	1		
	10	1	1		

图 2.2 DA 与 Q 关系

可以得到公式：

$$D_A = \overline{Q_A}$$

DB 与 Q 的关系如下：

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

图 2.3 DB 与 Q 关系

对应公式为：

$$\begin{aligned}
 D_B &= \overline{Q_A} Q_B + Q_A \overline{Q_B} \\
 &= \overline{Q_A \oplus Q_B}
 \end{aligned}$$

DC 与 Q 的关系如下：

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

图 2.4 DC 与 Q 的关系

对应公式为：

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

DD 与 Q 的关系如下：

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

图 2.5 DD 与 Q 的关系

对应公式为：

$$D_C = \overline{Q_A}Q_D + \overline{Q_B}Q_D + \overline{Q_C}Q_D + Q_AQ_BQ_C\overline{Q_D}$$

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

进位 R_C 的输出函数为：

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

2. 可逆二进制同步计数器

可逆计数器就是在 4 位二进制计数器的基础上增加减法计数器和控制电路两项功能，在本次实验中，通过控制 S 选择正反向计数：

- S = 1，正向计数
- S = 0，反向计数

$$D_A = \overline{Q_A}$$

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

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

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

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

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

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

3. 分频器

通过将 100MHz 的时钟信号 50,000,000 次分频得到 1Hz 秒脉冲信号，最为计数器的输入，其 Verilog 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'b1;
    end else begin
        cnt <= 0;
        clk_1s <= ~clk_1s;
    end
end
endmodule
```

三、实验过程和数据记录

1. 原理图方式设计 4 位同步二进制计数器

(a) 新建工程 MyCounter, Top Level Source 为 HDL

(b) 新建 Schematic 类型源文件 Counter4b

(c) 用原理图方式设计, 输入原理图如下:

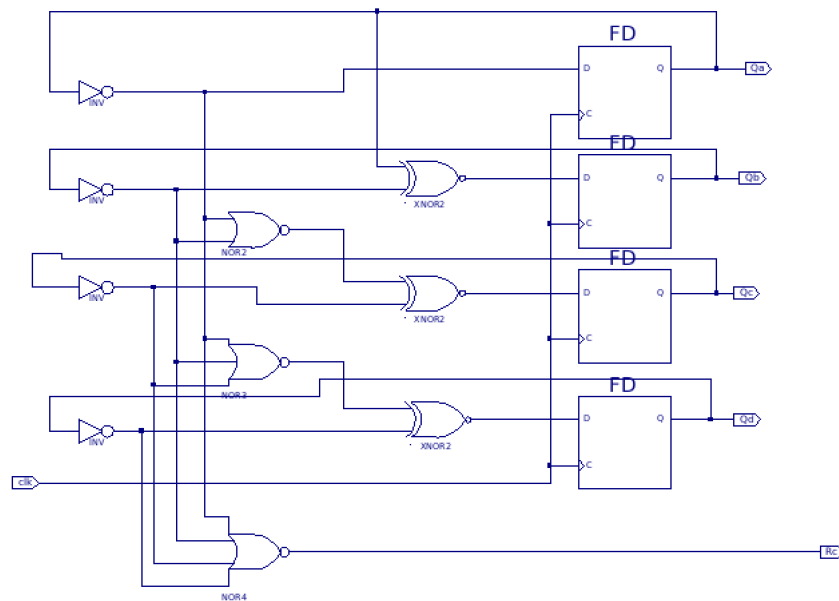


图 3.1 4 位二进制同步计数器

(d) 建立仿真文件 Counter4b_sim.v, 输入以下代码:

```
`timescale 1ns / 1ps
module Counter4b_Counter4b_sch_tb();

// Inputs
reg clk;

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

// Bidirs

// Instantiate the UUT
Counter4b UUT (
    .Qa(Qa),
```

```

        .Qb(Qb),
        .Qc(Qc),
        .Qd(Qd),
        .clk(clk),
        .Rc(Rc)
    );
// Initialize Inputs
    initial forever begin
        clk = 1'b0; #100;
        clk = 1'b1; #100;
    end
endmodule

```

(e) 得到波形图如下：

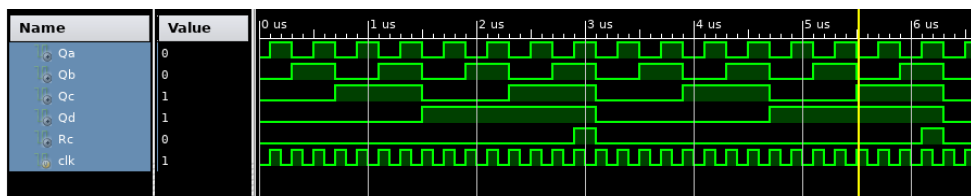


图 3.2 仿真波形图

Qa, Qb, Qc, Qd 随时钟周期从 0, 0, 0, 0 变为 1, 1, 1, 1, 在全为 1 时发生了进位, 此时 Rc 为 1, 结果与预期相符。

2. 时钟行为设计

(a) 新建源文件 clk_1s.v

(b) 输入以下 Verilog 代码：

```

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

```

3. 顶层模块设计

(a) 新建源文件 Top.v, 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 显示于 LED

(b) 由于需要使用数码显像管, 因此需要导入 DispNum 模块

(c) Top 模块代码如下:

```
`timescale 1ns / 1ps
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]
),.Rc(LED));
    DispNum_sch m2(.clk(clk),.HEXS({num[3:0],4'b0000, 4'b0000,
4'b0000}),.RST(1'b0),.AN(AN),.Segment(SEGMENT));
endmodule
```

(d) 新建引脚约束文件 K7.ucf, 代码如下:

```
NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
NET "LED" LOC = W23 | IOSTANDARD = LVCMOS33;

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 "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;
```

(e) 生成 bit 文件, 下载到 SWORD 板上进行验证:

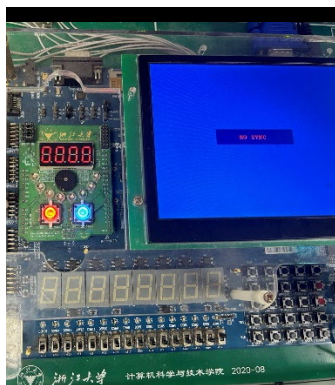


图 3.3 0 $R_c = 0$



图 3.4 2 $R_c = 0$

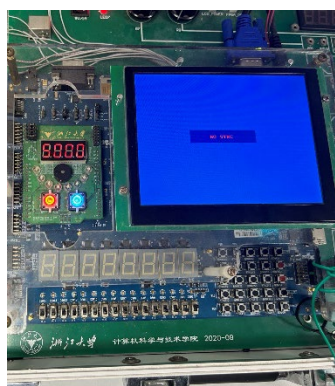


图 3.5 5 $R_c = 0$



图 3.6 F $R_c = 1$

每秒最左侧的数字增加 1，计数到 F 时发生进位，进位灯亮起

4. 以 Verilog 行为描述方式设计 16 位可逆二进制同步计数器

(a) 新建工程 myRevCounter，Top Level Source 为 HDL

(b) 新建源文件 RevCounter.v

(c) 使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
endmodule
```

(d) 新建仿真测试文件 RevCounter_sim.v, 输入以下代码:

```
`timescale 1ns / 1ps
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; #5;
        clk = 1'b1; #5;
    end
    initial begin
        s = 1'b0; #200;
        s = 1'b1; #200;
    end
endmodule
```

(e) 生成波形图如下:

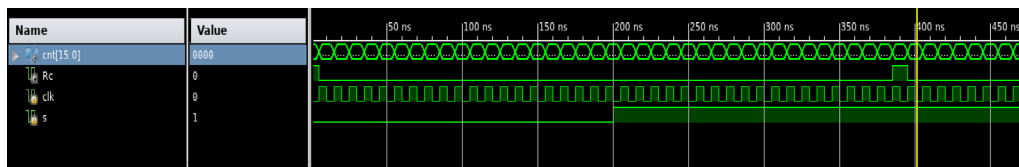


图 3.7 仿真波形

先反向计数 200ns, 开始时有一次进位, 再正向计数, 400ns 处有一次进位。

5. 时钟行为设计

(a) 新建源文件 clk_100ms.sch

(b) 用 Verilog 代码描述功能, 将 clk_1s 中的 50_000_000 改为 50_000_00, 从而使 1s 变为 100ms:

```
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'b1;
    end else begin
        cnt <= 0;
        clk_100ms <= ~clk_100ms;
    end
end
endmodule

```

6. 顶层模块设计

(a)新建源文件 Top.v, 右键设为“Set as Top Module”。Top 输入为 clk (100MHZ), RevCounter 模块输入为 100ms 时钟, 用 sw[0]控制自增/自减 1 (0.1s), 计数结果显示在 4 位数码管上, Rc 状态用 LED 灯来显示。

(b)Top 代码如下:

```

`timescale 1ns / 1ps
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_sch
m2(.clk(clk),.HEXS(num),.RST(1'b0),.AN(AN),.Segment(SEGMENT));
endmodule

```

(c)创建 K7.ucf 文件如下:

```

NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
NET "LED" LOC = W23 | IOSTANDARD = LVCMOS33;
NET "S" LOC = AA10 | IOSTANDARD = LVCMOS15;

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 "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;

```

(d) 下载到 SWOARD 板上验证

➤ 自减



图 3.8 0016 $R_c = 0$ 图 3.9 0000 $R_c = 1$ 图 3.10 FFFd $R_c = 0$

➤ 自增

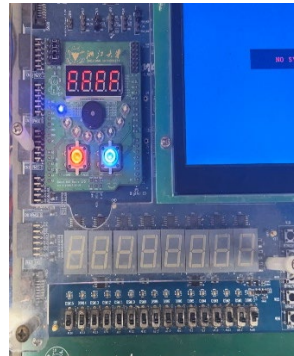


图 3.11 0001 $R_c = 0$ 图 3.12 FFFF $R_c = 1$ 图 3.13 FFF8 $R_c = 0$

自减时 0000 进位灯亮，自增时 FFFF 进位灯亮，符合预期

四、实验结果分析

本次实验通过仿真并使用 SWORD 板进行验证，实验结果符合预期。

在对 RevCounter 模块进行仿真模拟的过程中，发现两次进位的时间总是间隔过段，无法判断电路图是否正常运行，在助教老师的帮助下将 $S = 0$, $S = 1$

的时间延长至 200ms，在反向计数一定时间后在进行正向计数，使得两次进位间隔更大，结果更明显，最终问题得以解决：

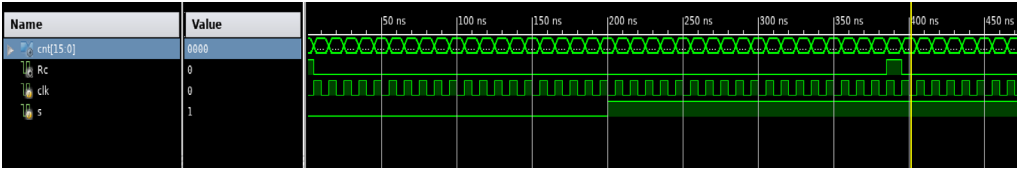


图 4.1

五、讨论与心得

在这次实验中，我们成功地应用触发器，实现了两种类型计数器。同步触发器的电路结构相对复杂，需要使用门电路进行配合，但由于各触发器的状态更新是同步进行的，因此计数工作的速度相对较快。总体而言，这次实验提供了对不同计数器实现的深入了解，同时也突显了同步和异步电路之间的权衡。

但是在抓拍 100ms 时钟的进位过程中，手速不够，多次错过，最后采取了录制视频再截图的方式解决。感谢老师、助教以及本组同学的帮助，本次实验顺利完成！