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　掌握锁存器与触发器构成的条件和工作原理

1.2　掌握锁存器与触发器的区别

1.3　掌握基本 SR 锁存器、门控 SR 锁存器、D 锁存器、SR 锁存器、D 触发器的基本功能

1.4　掌握基本 SR 锁存器、门控 SR 锁存器、D 锁存器、SR 锁存器存在的时序问题

2　实验任务

2.1　实现基本 SR 锁存器，验证功能和存在的时序问题

2.2　实现门控 SR 锁存器，并验证功能和存在的时序问题

2.3　实现 D 锁存器，并验证功能和存在的时序问题

2.4　实现 SR 主从触发器，并验证功能和存在的时序问题

2.5　实现 D 触发器，并验证功能

二、 实验内容和原理

1　实验内容

1.1　实现基本 SR 锁存器，验证功能和存在的时序问题

1.2　实现门控 SR 锁存器，并验证功能和存在的时序问题

1.3　实现 D 锁存器，并验证功能和存在的时序问题

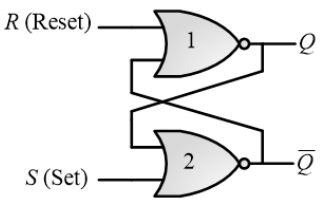
1.4　实现 SR 主从触发器，并验证功能和存在的时序问题

1.5　实现 D 触发器，并验证功能

2　实验原理

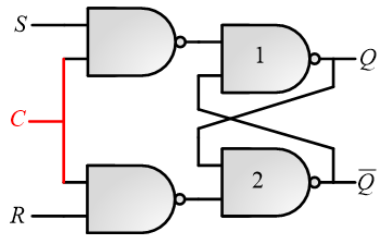
2.1　构成锁存器的充分条件：能长期保持给定的某个稳定状态；有两个稳定状态：0、1；在一定条件下能随时改变逻辑状态，即：置 1 或置 0。最基本的锁存器有：SR 锁存器、D 锁存器 。锁存器有两个稳定状态，又称双稳态电路。

2.2　将两个具有 2 输入端的反向逻辑器件的输出与输入端交叉连起来，另一个输入端作为外部信息输出端，就构成最简单的 SR 锁存器



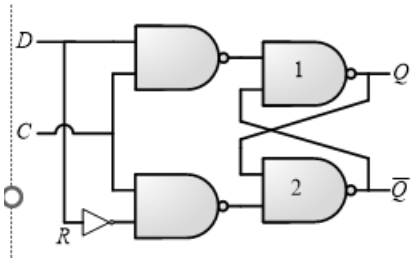
R S	*Q Q*	说明
0 0	*Q Q*	保持
0 1	1 0	置1
1 0	0 1	置0
1 1	0 0	未定义

2.3 门控 SR 锁存器 ：在 SR 锁存器的基础上，增加使能信号 C 控制存储数据的更新。



C R S	Q Q	说明
0 × ×	Q Q	保持
1 0 0	Q Q	保持
1 0 1	1 0	置1
1 1 0	0 1	置0
1 1 1	1 1	未定义

2.4 基本 SR 锁存器缺点：存在不确定状态。解决方法：消除不确定状态。只需 1 个数据输入端 D，输出端 Q 等于输入端 D，采用电平控制 C。

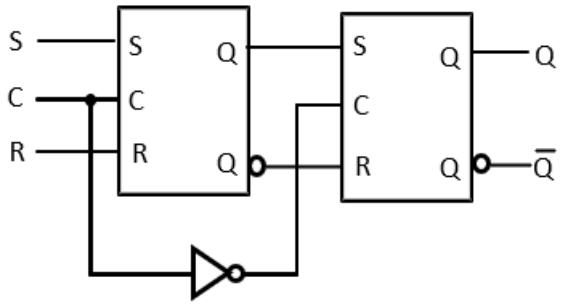


C D	Q Q	说明
0 ×	Q Q	保持
1 0	0 1	置0
1 1	1 0	置1

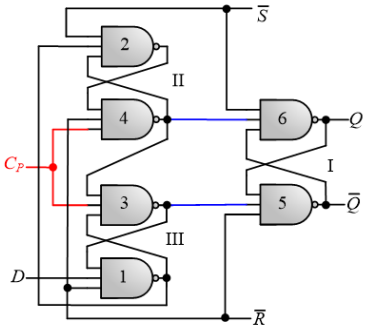
2.5 D 锁存器的缺点：存在空翻现象——如果 D 锁存器直接用在时序电路中作为状态存储元件，当使能控制信号有效时，会导致该元件内部的状态值随时多次改变，而不是保持所需的原始状态值。解决方法：消除空翻现象，使每次触发仅使锁存器的内部状态仅改变一次

2.6 SR 主从触发器：由两个钟控 S-R 锁存器串联构成，第二个锁存器的时钟通过反相器取反。当 C=1 时，输入信号进入第一个锁存器（主锁存器）；当 C=0 时，第二个锁存器（从锁存器）改变输出；从输入到输出的通路被不同的时钟信号值(C

= 1 和 C = 0)所断开。



2.7 正边沿维持阻塞型 D 触发器：避免小脉冲带来的一次性采样问题。



异步控制		上升沿触发			
R	***S***	C_P	***D***	***Q***	***Q***
0	**1**	×	×	**0**	**1**
1	**0**	×	×	**1**	**0**
1	**1**	↑	**0**	**0**	**1**
1	**1**	↑	**1**	**1**	**0**

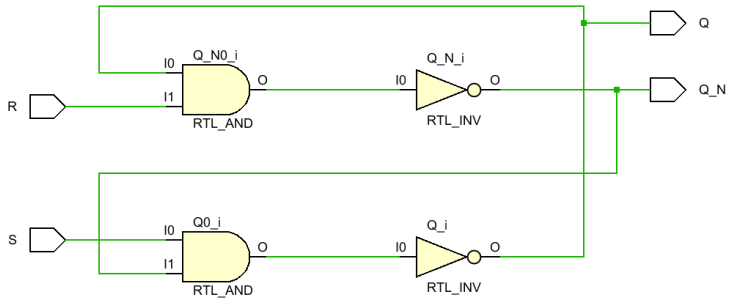
三、 实验过程和数据记录

1 实现并验证功能，并验证功能

1.1 原理图方式设计，并仿真验证功能

```
`timescale 1ns / 1ps
module SR_LATCH(
input wire S,R,
output wire Q,Q_N
    );
    assign Q=~(S&Q_N);
    assign Q_N=~(Q&R);
endmodule
`timescale 1ns / 1ps
module SR_LATCH_test1 ();
```

```
  reg S, R;
  wire Q, Q_N;
  SR_LATCH test1 (
      S,
      R,
      Q,
      Q_N
  );
  initial begin
    R = 1;
    S = 1;
    #50;
    R = 1;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;
    R = 0;
    S = 1;
    #50;
    R = 1;
    S = 1;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;
  end
endmodule
```




```
`timescale 1ns / 1ps
```

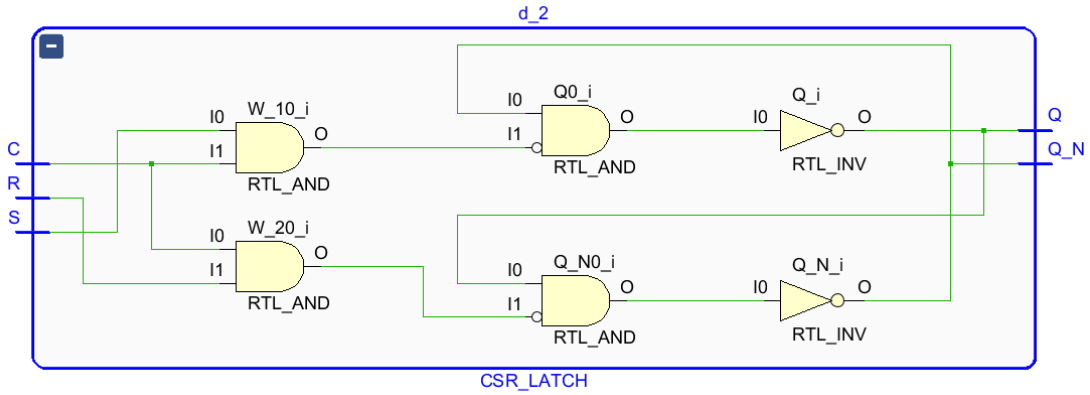
```
module SR_LATCH_test1 ();
  reg S, R;
  wire Q, Q_N;
  SR_LATCH test1 (
      S,
      R,
      Q,
      Q_N
  );
  initial begin
    R = 1;
    S = 1;
    #50;
    R = 1;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;
    R = 0;
    S = 1;
    #50;
    R = 1;
    S = 1;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;
  end
endmodule
```

2　实现门控 SR 锁存器，并验证功能

2.1 用原理图方式实现，并仿真验证功能。

实现

```
`timescale 1ns / 1ps
module CSR_LATCH(
input C,S,R,
output Q,Q_N
    );
    wire W_1,W_2;
    assign W_1=~(S&C);
    assign W_2=~(C&R);
    assign Q=~(Q_N&W_1);
    assign Q_N=~(Q&W_2);
endmodule
```

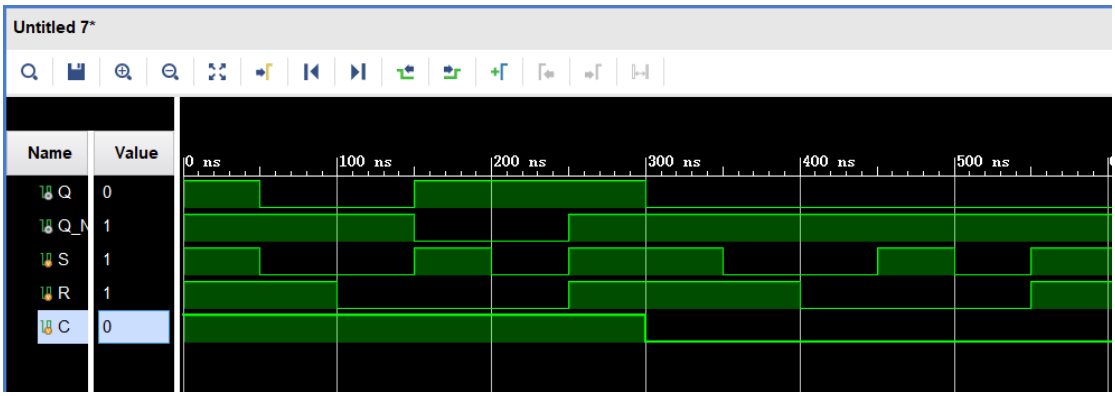


仿真

```
`timescale 1ns / 1ps
module CSR_LATCH_test2 ();
  reg C, S, R;
  wire Q, Q_N;
  CSR_LATCH test2 (
      C,
      S,
      R,
      Q,
      Q_N
  );
  initial begin
```

```
    C = 1;
    R = 1;
    S = 1;
    #50;
    R = 1;
    S = 0;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 0;
    S = 1;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;
    C = 0;
    R = 1;
    S = 1;
    #50;
    R = 1;
    S = 0;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 0;
    S = 1;
    #50;
    R = 0;
    S = 0;
    #50;
    R = 1;
    S = 1;
    #50;

  end
endmodule
```

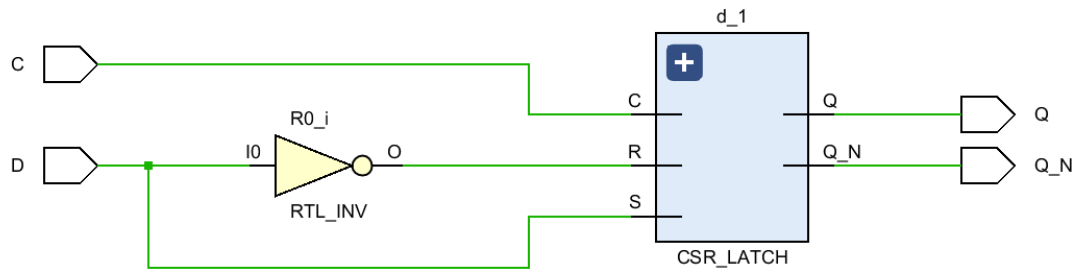


3　实现 D 锁存器，并验证功能和存在的时序问题

3.1 用原理图方式实现，并且仿真验证功能和时序问题

实现

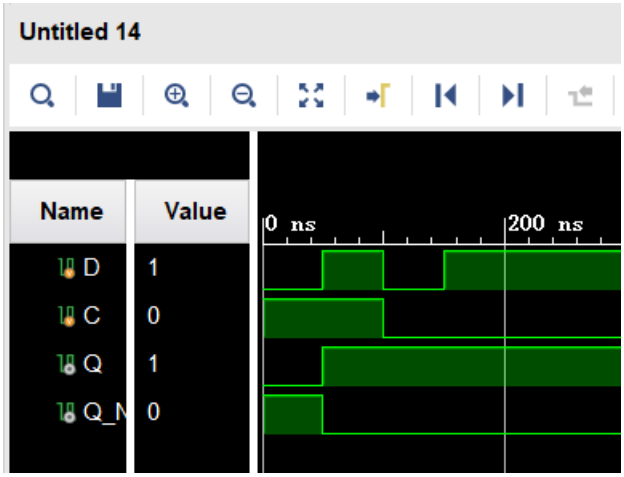
```
`timescale 1ns / 1ps
module D_FLIPFLOP(
input wire D,Cp,S,R,
output wire Q,Q_N
    );
    wire W_1,W_2,W_3,W_4;
    assign W_1=~(~R&D&W_3);
    assign W_2=~(~S&W_1&W_4);
    assign W_3=~(Cp&W_4&W_1);
    assign W_4=~(Cp&~R&W_2);
    assign Q=~(~S&W_4&Q_N);
    assign Q_N=~(Q&~R&W_3);
endmodule
```



仿真

```
`timescale 1ns / 1ps
```

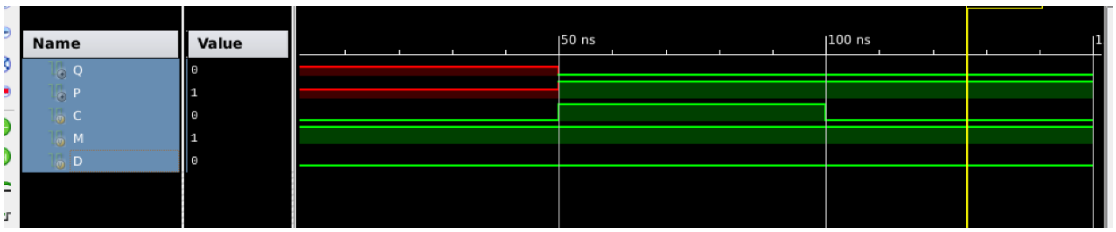
```
module D_LATCH_test3 ();
  reg D, C;
  wire Q, Q_N;
  D_LATCH test3 (
      D,
      C,
      Q,
      Q_N
  );
  initial begin
    D = 0;
    C = 1;
    #50;
    D = 1;
    #50;
    C = 0;
    D = 0;
    #50;
    D = 1;
    #50;
  end
endmodule
```



时序问题（空翻）

```
`timescale 1ns / 1ps
module D_LATCH_test3 ();
  reg  C,sign;
  wire D, Q, Q_N;
  D_LATCH test3 (
```

```
      D,
      C,
      Q,
      Q_N
  );
  assign D=sign|Q_N;
  /*D_LATCH test3 (
      D,
      C,
      Q,
      Q_N
  );
  initial begin
    D = 0;
    C = 1;
    #50;
    D = 1;
    #50;
    C = 0;
    D = 0;
    #50;
    D = 1;
    #50;
  end*/
  initial begin
    C=1;
    sign=1;
    #100;
    sign=0;
  end
endmodule
```



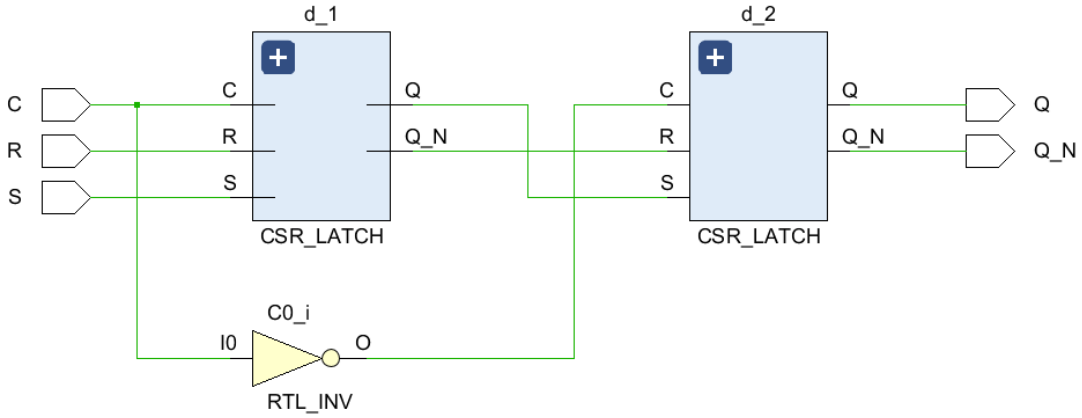
4　实现 SR 主从触发器，并验证功能和存在的时序问题

4.1 原理图实现 SR 主从触发器，并且仿真验证功能和存在的时序问

题

实现

```
`timescale 1ns / 1ps
module MS_FLIPFLOP(
input wire S,R,C,
output Q,Q_N
    );
    wire W_1,W_2;
    CSR_LATCH d_1(C,S,R,W_1,W_2),d_2(~C,W_1,W_2,Q,Q_N);
endmodule
```

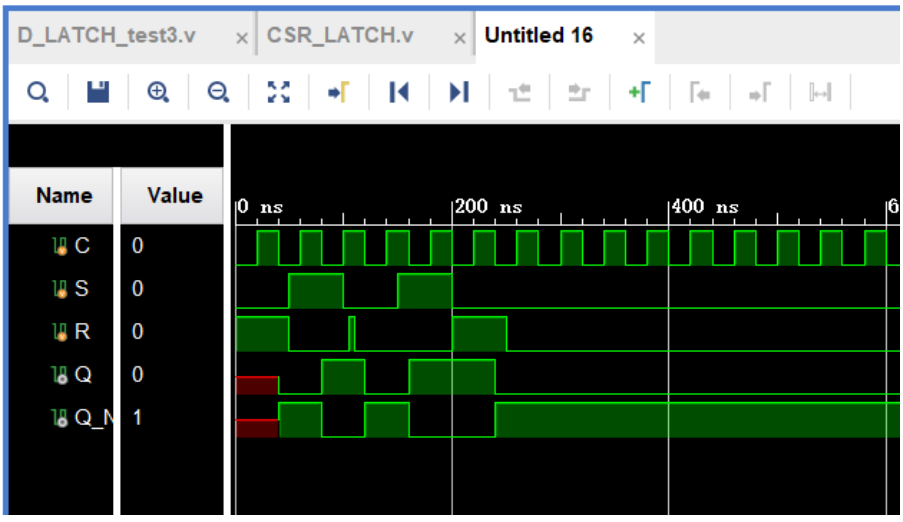


仿真和时序问题（单次采样）

```
`timescale 1ns / 1ps
module MS_FLIPFLOP_test5 ();
  reg C, S, R;
  wire Q, Q_N;
  MS_FLIPFLOP test5 (
      S,
      R,
      C,
      Q,
      Q_N
  );
  initial begin
    R = 1;
    S = 0;
    #50;
    R = 0;
    S = 1;
```

```
    #50 R = 0;
    S = 0;
    #5;
    R = 1;
    S = 0;
    #5;
    R = 0;
    S = 0;
    #40;
    R = 0;
    S = 1;
    #50;
    R = 1;
    S = 0;
    #50;
    R = 0;
    S = 0;
    #50;
  end
  always begin
    C = 0;
    #20;
    C = 1;
    #20;
  end

endmodule
```

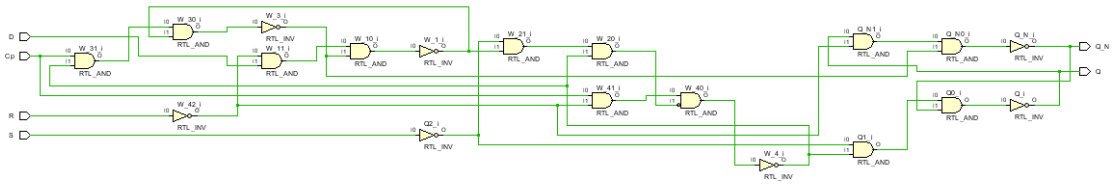


5　实现 D 触发器，并验证功能

5.1 用原理图实现并且仿真来验证功能

实现

```
`timescale 1ns / 1ps
module D_FLIPFLOP(
input wire D,Cp,S,R,
output wire Q,Q_N
    );
    wire W_1,W_2,W_3,W_4;
    assign W_1=~(~R&D&W_3);
    assign W_2=~(~S&W_1&W_4);
    assign W_3=~(Cp&W_4&W_1);
    assign W_4=~(Cp&~R&W_2);
    assign Q=~(~S&W_4&Q_N);
    assign Q_N=~(Q&~R&W_3);
endmodule
```



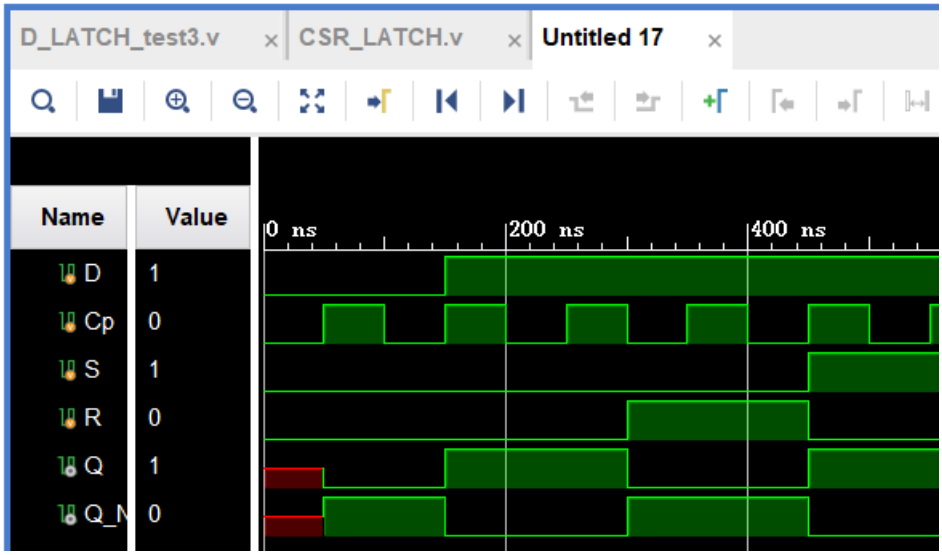
仿真

```
`timescale 1ns / 1ps
module D_FLIPFLOP_test6 ();
  reg D, Cp,S,R;
  wire Q, Q_N;
  D_FLIPFLOP test6 (
      D,
      Cp,
      S,
      R,
      Q,
      Q_N
  );
```

```
  initial begin
    S=0;R=0;
    D = 0;
    #150;
    D = 1;
    #150;
    S=0;R=1;
    #150
    S=1;R=0;
  end

  always begin
    Cp = 0;
    #50;
    Cp = 1;
    #50;
  end

endmodule
```



四、 实验结果分析

实验的仿真波形符合预期。D 锁存器仿真波形后半段空翻因剧烈震荡无仿真线显示。

五、 讨论与心得

采取上边沿触发的 D 触发器抗干扰能力强。