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：秦嘉俊

学　　院：竺可桢学院

系：所在系

专　　业：计算机科学与技术

学　　号：3210106182

指导教师：董亚波

2022 年 11 月 15 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: 锁存器与触发器基本原理

学生姓名： 秦嘉俊 专业： 计算机科学与技术 学号： 3210106182

同组学生姓名： 钟梓航 指导老师： 董亚波

实验地点： 东 4-509 实验日期： 2022 年 11 月 16 日

一、实验目的和要求

1. 掌握锁存器与触发器构成的条件和工作原理
2. 掌握锁存器与触发器的区别
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
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

二、实验内容和原理

2.1 实验设备

- 装有 Xilinx ISE 14.7 的计算机 1 台
- SWORD 开发板 1 套

2.2 内容

1. 实现基本 SR 锁存器，验证功能和存在的时序问题
2. 实现门控 SR 锁存器，并验证功能和存在的时序问题
3. 实现 D 锁存器，并验证功能和存在的时序问题
4. 实现 SR 主从触发器，并验证功能和存在的时序问题
5. 实现 D 触发器，并验证功能

2.3 原理

2.3.1 锁存器

锁存器 (Latch) 是一种对脉冲电平敏感的存储单元电路，它们可以在特定输入脉冲电平作用下改变状态。锁存，就是把信号暂存以维持某种电平状态。构成锁存器的充分条件有：

1. 能长期保持给定的某个稳定状态
2. 有两个稳定状态：0、1
3. 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。

2.3.2 SR 锁存器

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。

SR 锁存器电路是由两个交叉耦合的或非门构成的。我们只需要把单环路存储元件中的反相器简单地替换成或非门就可以得到 SR 锁存器。锁存器有两个输入端 S 和 R，其中 S 用于置位，R 用于复位。当输出 $Q = 1$ 且 $\overline{Q} = 0$ 时，称锁存器处于置位状态 (set state)；当 $Q = 0$ 且 $\overline{Q} = 1$ 时，称锁存器处于复位状态 (reset state)。输出 Q 和 $\overline{Q}$ 通常是互反的，当两个输入同时为 1 时，两个输出都等于 0，这是个未定义状态。

在通常情况下，除非要改变状态，否则锁存器的两个输入信号都将保持为 0,。若输出端 S 变为 1，锁存器进入置位状态。有两种输入条件可以会使电路处在置位状

态。初始条件为 S = 1，R = 0 时，电路进入置位状态。当 S 重置为 0 且使 R = 0，电路保持置位状态不变。当两个输入信号返回到 0 之后，将输入 R 置为 1，电路将进入复位状态。这时，如果将 R 重置为 0，电路将保持复位状态不变。所以，当两个输入同时被置为 0 时，锁存器可能处于置位状态，也可能处于复位状态，这取决于最近哪个输入为 1。

如果将锁存器的两个输入都置为 1，那么两个输出都为 0，这是衣蛾未定义状态，因为它违背了两个输出互反的原则。同时，如果两个输入同时返回 0，会导致输出的下一个状态不确定或者不可预知。所以为了避免上述情况的发生，应该保证两个输入不会同时为 1。

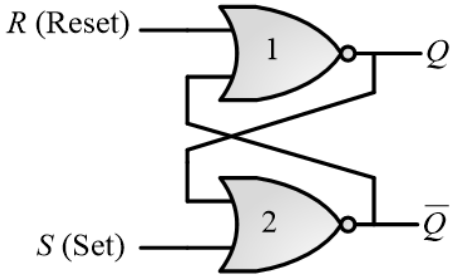


图 1: SR 锁存器

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

图 2: 输入状态表

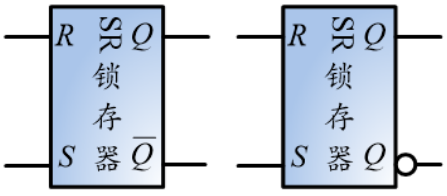


图 3: 逻辑符号

由两个与非门交叉耦合的锁存器叫做 $\overline{SR}$ 锁存器，或者在一些别的资料中叫做 RS 锁存器。

除非要改变锁存器的状态，否则在通常情况下，两个输入信号都为 1。将输入 $\overline{S}$ 置为 0 时，输出 Q 变为 1，锁存器进入置位状态。当输入 $\overline{S}$ 变回 1 时，电路保持置位状态不变。当两个输入信号都为 1 时，将输入信号 $\overline{R}$ 的值变为 0，锁存器的状态会随之改变，此时电路进入复位状态，即两个输入信号都被置为 1 后电路状态仍保持此状态不变。两个输入信号同时等于 0 是未定义状态，应该避免。

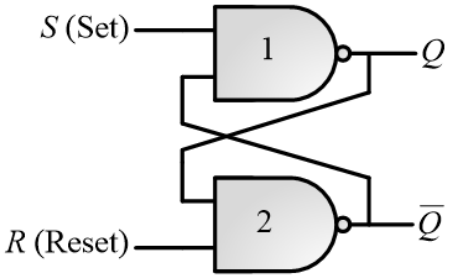


图 4: $\overline{SR}$ 锁存器

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

图 5: 输入状态表

2.3.3 门控 $\overline{SR}$ 锁存器

我们有必要对锁存器的置 1 或清 0 操作采取一定的控制措施。通过增加一个额外的输入控制信号来控制锁存器何时对输入敏感，我们可以改变基本或非门锁存器和与非门锁存器的操作。带有控制输入的 SR 锁存器如图 2.6 所示。它由羁绊与非门锁存器和两个额外的与非门组成。这里输入信号 C 作为另外两个输入信号的使能信号，只要控制这个控制信号 C 保持为 0，与之相连的与非门就一直维持为 1，这是使由两个与非门组成的 $\overline{SR}$ 锁存器保持静止的条件。当控制信号 C 为 1 时，S 和 R 的值才会影响到 $\overline{SR}$ 锁存器的状态。无论 $\overline{SR}$ 处于哪种状态，当 C 变回 0 时，电路就会保持当前状态不变。

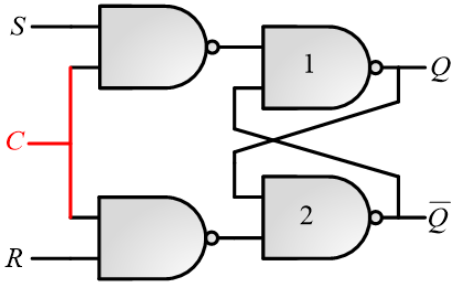


图 6: 门控 $\overline{SR}$ 锁存器

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

图 7: 输入状态表

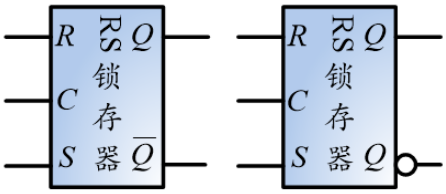


图 8: 逻辑符号

2.3.4 D 锁存器

消除锁存器中不期望存在的未定义状态的一种方法就是确保输入信号 S 和 R 永远不会同时取 1，D 锁存器就是按照这种方法构造的。D 锁存器只有两个输入信号：D(数据信号) 和 C(控制信号)。输入信号 D 的非之际连接到输入端 $\overline{S}$，输入信号 D 加载到输入端 $\bar{r}$。只要控制输入信号为 0，$\overline{SR}$ 锁存器的两个输入信号都处于 1 电平，从而无论 D 为何值电路状态都不会改变。当 C = 1 时，D 被电路采样。如果 D 为 1，那么输出 Q 为 1，电路处于置位状态；如果 D 为 0，那么输出 Q 为 0，电路处于复位状态。

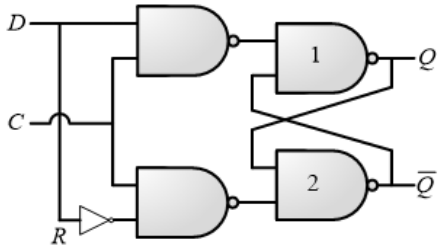


图 9: D 锁存器

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

图 10: 输入状态表

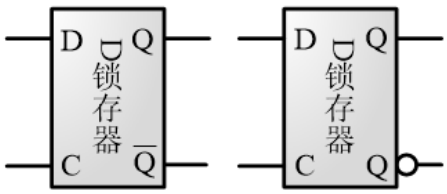


图 11: 逻辑符号

D 锁存器之所以被命名为 D 锁存器是因为它可以在内部存储原件中保存数据。当控制输入信号有效 (为 1) 时，D 锁存器数据输入端的信息被传播到输出端 Q，Q 的值随着数据输入值的变化而变化。控制输入信号失效 (为 0) 后，数据输入端在 C 发生变化时的信息会一直保持在输出端 Q 不变，直到控制信号再次有效为止。

2.3.5 SR 触发器和 D 触发器

触发：外部输入使锁存器状态改变的瞬间状态。

触发器：在锁存器的基础上使每次触发仅使状态改变一次的锁存电路（双稳态）。

也就是说，在实际的数字系统中往往包含大量的存储单元，而且经常要求他们在同一时刻同步动作，为达到这个目的，在每个存储单元电路上引入一个时钟脉冲

（CLK）作为控制信号，只有当 CLK 到来时电路才被“触发”而动作，并根据输入信号改变输出状态。把这种在时钟信号触发时才能动作的存储单元电路称为触发器，以区别没有时钟信号控制的锁存器。常见的触发器有：主从 SR 触发器、D 触发器、JK 触发器、T 触发器。

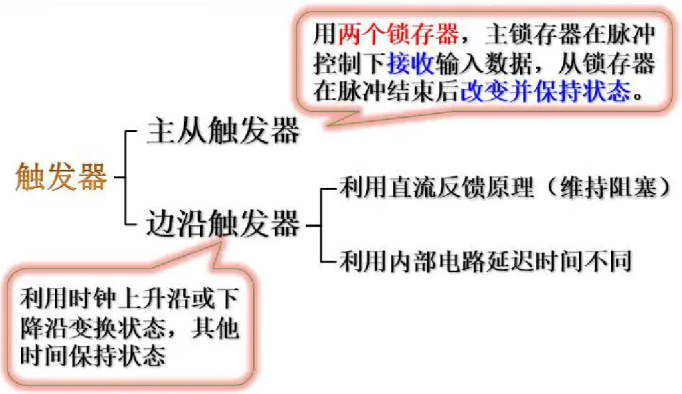


图 12: 触发器的分类

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。

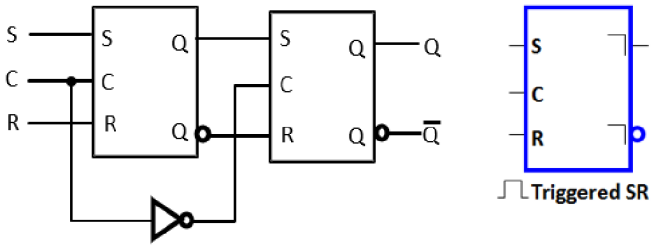


图 13: SR 主从触发器的电路图和逻辑符号图

边沿 D 触发器也称为维持-阻塞边沿 D 触发器。

电路结构：如图下所示，该触发器由 6 个与非门组成，其中的 5、6 构成基本 RS 触发器。

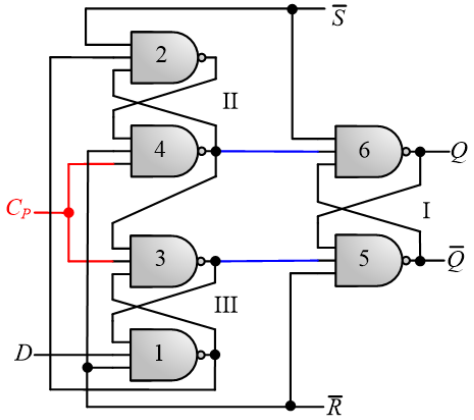


图 14: 正边沿维持阻塞型 D 触发器

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

图 15: 输入状态表

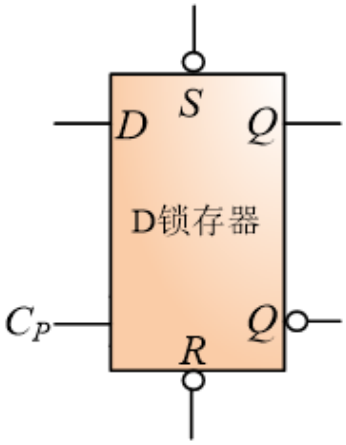


图 16: 逻辑符号

2.3.6 空翻现象

空翻现象: 又称为竞态现象，是数字电路中的一个术语，指在同一个时钟脉冲信号作用区间内，由于时钟脉冲的宽度过大，触发器出现在“0”“1”两逻辑信号中多

次翻转的现象。

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。

三、实验过程和数据记录

3.1 实现基本 SR 锁存器，验证功能和存在的时序问题

1. 新建工程 MyLATCHS

2. 新建源文件 SR_LATCH.sch

3. 用原理图方式设计，用 NAND2 实现
   画出原理图如下：

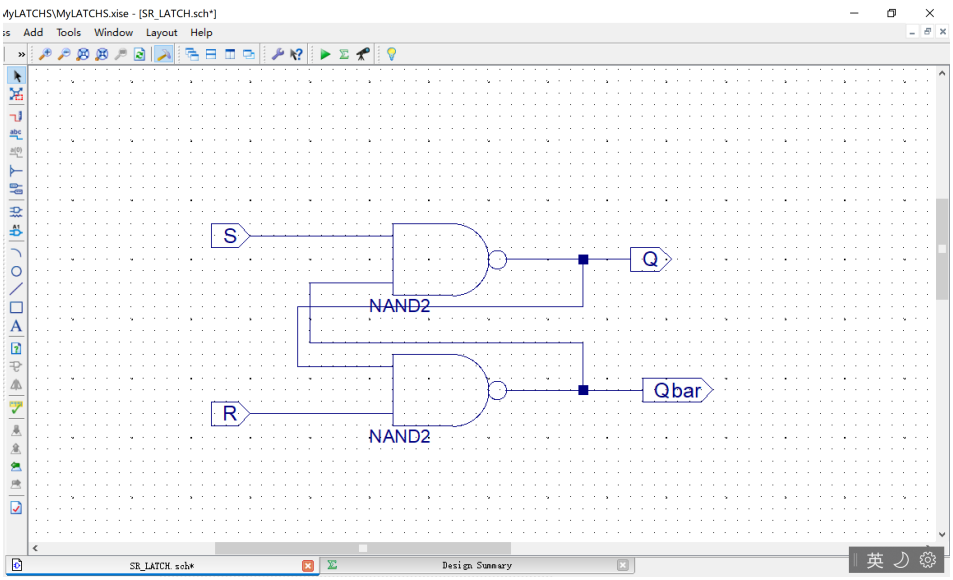


图 17: SR 锁存器电路图绘制

4. 仿真
   建立仿真模拟文件：SR_LATCH_sim.v，里面设置如下代码：

```
1          `timescale 1ns / 1ps
2          module SR_LATCH_SR_LATCH_sch_tb();
3          // Inputs
```

```
4           reg S;
5           reg R;
6
7       // Output
8           wire Q;
9           wire Qbar;
10
11      // Bidirs
12
13      // Instantiate the UUT
14          SR_LATCH UUT (
15               .S(S),
16               .Q(Q),
17               .Qbar(Qbar),
18               .R(R)
19          );
20      // Initialize Inputs
21             initial begin
22              R=1;S=1; #50;
23              R=1;S=0; #50;
24              R=1;S=1; #50;
25              R=0;S=1; #50;
26              R=1;S=1; #50;
27              R=0;S=0; #50;
28              R=1;S=1; #50;
29              end
30      endmodule
```

得到如下波形图：

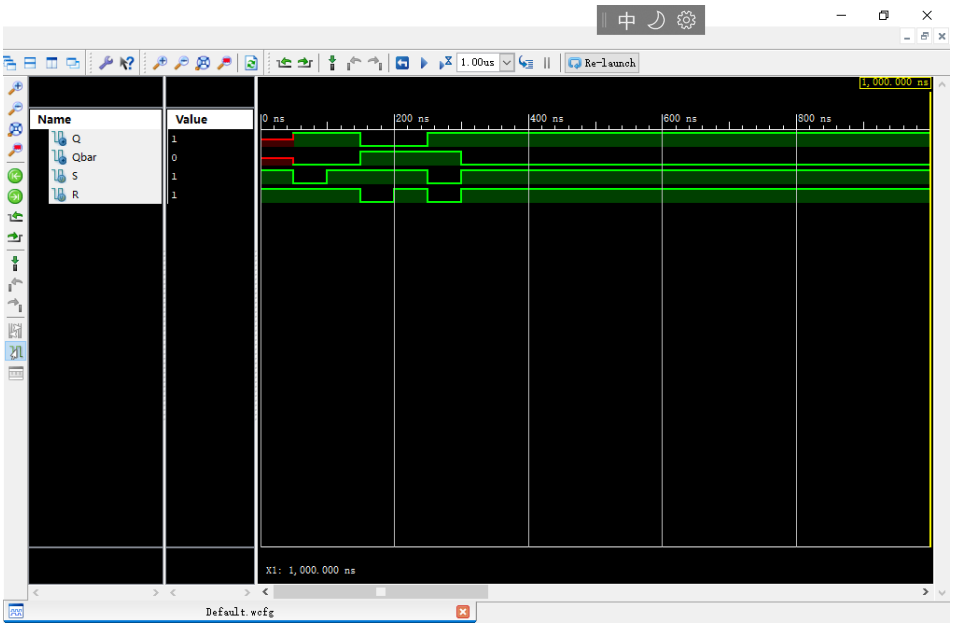


图 18: SR 触发器仿真波形

可以看到，最开始锁存器中的存储值是未知的，当 S=0 R=1 时，我们存入了 1，因此 Q 变为 1；当 S=1 R=0 时，我们存入了 0；当 S R 同时从 0 变为 1 时，Q 和 Q 非的结果是不定的，取决于门的延迟。仿真结果和 SR 锁存器真值表相同，符合预期。

3.2 实现门控 SR 锁存器，并验证功能和存在的时序问题

1. 新建源文件 CSR_LATCH.sch

2. 用原理图方式设计，用 NAND2 实现
   画出原理图如下：

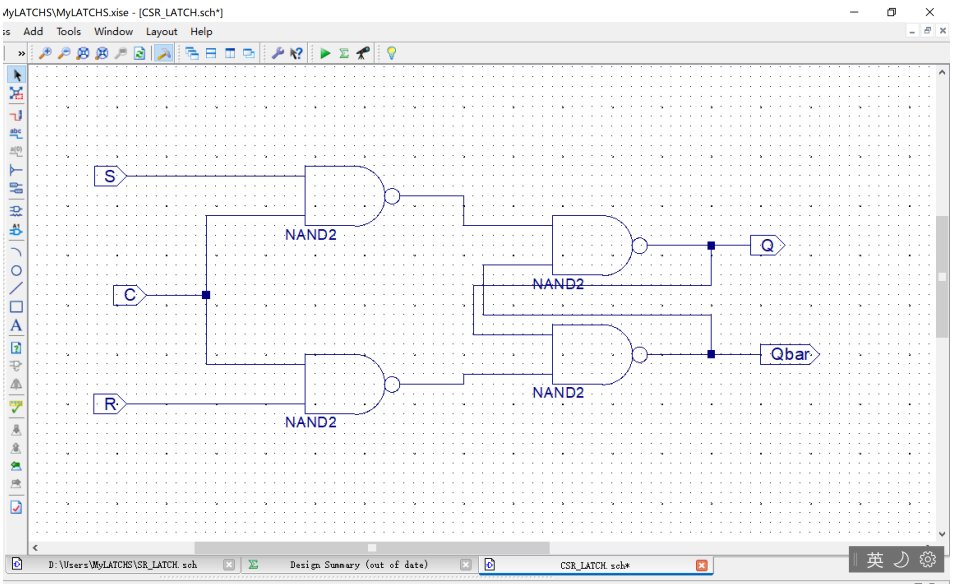


图 19: 门控 SR 锁存器原理图绘制

3. 仿真

新建仿真测试文件 CSR_LATCH_sim.v，里面设置如下代码：

```
1          `timescale 1ns / 1ps
2          module CSR_LATCH_CSR_LATCH_sch_tb();
3
4          // Inputs
5             reg S;
6             reg R;
7             reg C;
8
9          // Output
10            wire Q;
11            wire Qbar;
12
13         // Bidirs
14
15         // Instantiate the UUT
16            CSR_LATCH UUT (
17                 .Q(Q),
18                 .Qbar(Qbar),
19                 .S(S),
20                 .R(R),
```

```
21                  .C(C)
22              );
23          // Initialize Inputs
24              initial begin
25                  C = 0;
26                  S = 0;
27                  R = 0;
28                  C=1;R=1;S=1; #50;
29                  R=1;S=0; #50;
30                  R=0;S=0; #50;
31                  R=0;S=1; #50;
32                  R=0;S=0; #50;
33                  R=0;S=0; #50;
34                  R=0;S=0; #50;
35                  C=0;R=1;S=1; #50;
36                  R=1;S=0; #50;
37                  R=0;S=0; #50;
38                  R=0;S=1; #50;
39                  R=1;S=1; #50;
40                  R=0;S=0; #50;
41               end
42          endmodule
```

得到如下波形图：

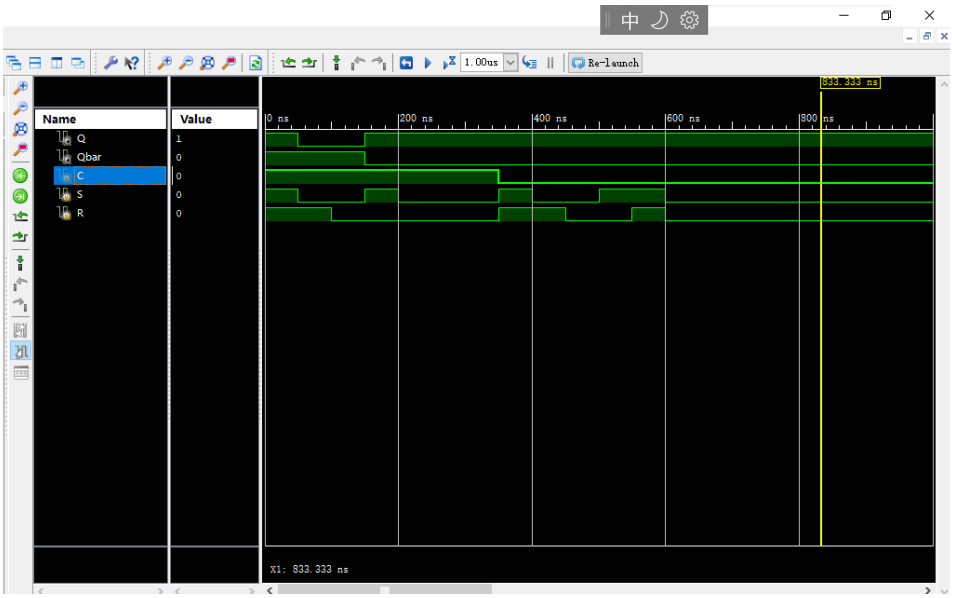


图 20: 门控 SR 锁存器仿真波形

可以看到：C = 1 时，Q 随着 S R 的变化而变化，当 S = 1 时 Q = 1，当 R = 1 时 Q = 0，R = S = 0 时保持当前值（二者都为 1 时状态是未定义）；C = 0 时，Q 的值保持不变。因此仿真结果符合预期。

4. 点击 Create Schematic Symbol，生成自定义符号 CSR_LATCH.sym。

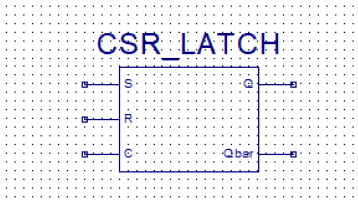


图 21: 逻辑符号图

3.3 实现 D 锁存器，并验证功能和存在的时序问题

3.3.1 D 锁存器的实现

1. 新建源文件 D_LATCH.sch

2. 用原理图方式设计，用 NAND2 实现
   画出原理图如下：

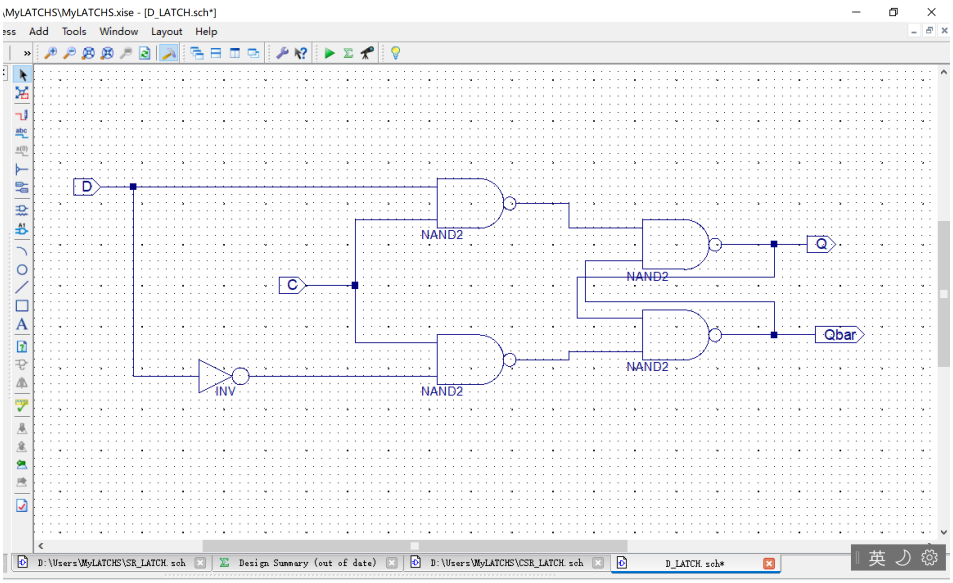


图 22: D_LATCH 电路图绘制

3. 仿真

新建仿真测试文件 D_LATCH_sim.v，里面设置如下代码：

```
1          `timescale 1ns / 1ps
2          module D_LATCH_D_LATCH_sch_tb();
3
4          // Inputs
5             reg D;
6             reg C;
7
8          // Output
9             wire Q;
10            wire Qbar;
11
12         // Bidirs
13
14         // Instantiate the UUT
15            D_LATCH UUT (
16                  .Q(Q),
17                  .Qbar(Qbar),
18                  .D(D),
19                  .C(C)
20            );
```

```
21          // Initialize Inputs
22             initial begin
23                   C=1;D=1; #50;
24                   D=0; #50;
25                   C=0;D=1; #50;
26                   D=0;
27              end
28          endmodule
```

得到如下波形图：

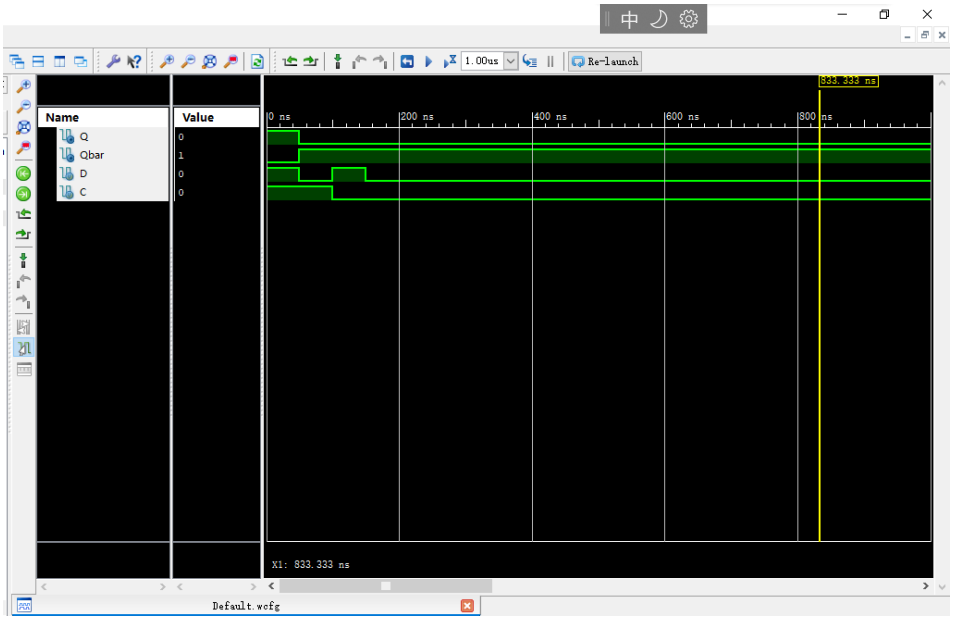


图 23: D_LATCH 仿真波形

可以看到：C = 1 时，Q 随着 D 的变化而变化（Q 为 1 则 D 为 1，反之亦然，此时不存在未定义的状态）；C = 0 时，Q 的值保持不变。因此仿真结果符合预期。

3.3.2 搭建电路验证空翻现象

选中 D_LATCH.sch，点击 Create Schematic Symbol, 生成 D 锁存器的逻辑符号图。新建 Schematic 文件，命名为 D_LATCH_FLIP。绘制用于测试空翻现象的外部电路：

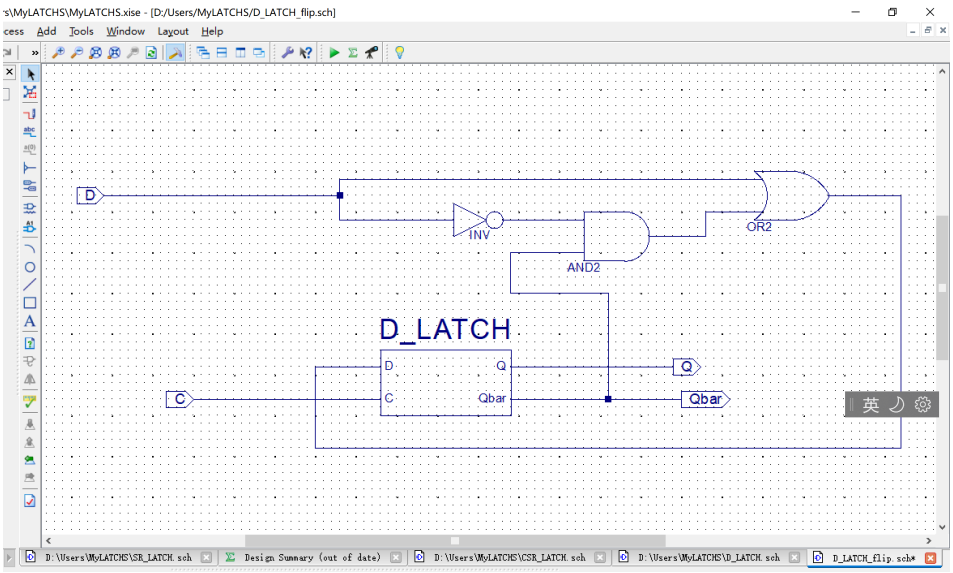


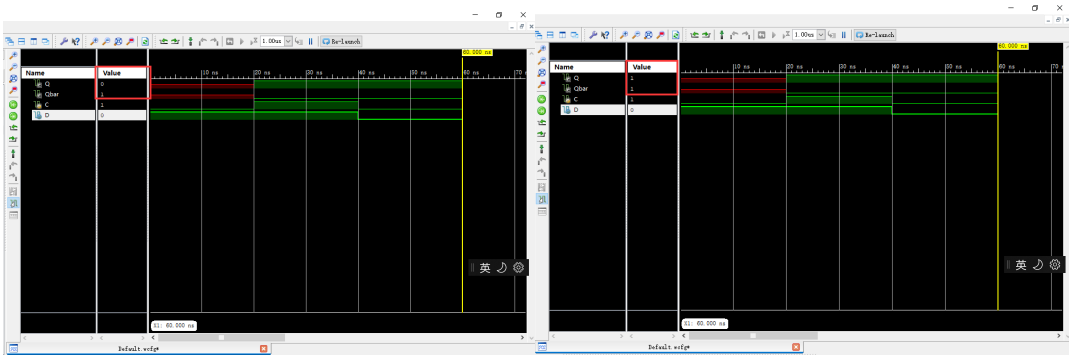
图 24: 验证空翻的电路图绘制

这里的思想是：首先将输入端 D 置为 1，此时 Q 为 1，Qbar 为 0。随后将 D 置为 0，这样相当于 Qbar 和 D 锁存器的输入端连接，理论上会产生一个振荡电路，因为 Qbar 和 D 锁存器输入理论上一直是相反的。

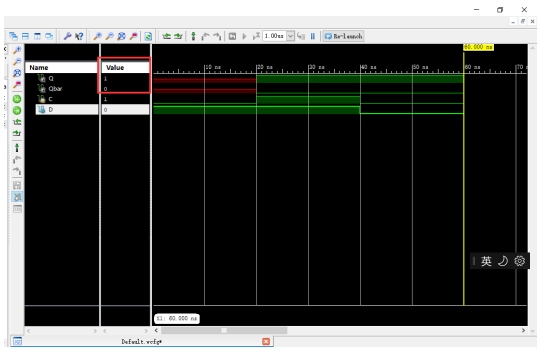
新建仿真测试文件 D_LATCH_FLIP_sim，里面设置如下代码：

```
1     `timescale 1ns / 1ps
2     module D_LATCH_flip_D_LATCH_flip_sch_tb();
3
4     // Inputs
5        reg C;
6        reg D;
7
8     // Output
9        wire Q;
10       wire Qbar;
11
12    // Bidirs
13
14    // Instantiate the UUT
15       D_LATCH_flip UUT (
16            .C(C),
17            .D(D),
18            .Q(Q),
```

```
19              .Qbar(Qbar)
20          );
21      // Initialize Inputs
22          initial begin
23              D = 1;#40;
24              D = 0;
25              end
26              always begin //产生时钟
27              C = 0;#20;
28              C = 1;#20;
29              end
30      endmodule
```



(a) (b)



(c)

图 25: 验证空翻现象

可以发现，我们的波形图在 60ns 后即消失，此时我们的电路就陷入了震荡状态，不停地点击最后时刻会看到各个量的值在变化（如上图所示）。因此我们验证得到了震荡现象。

3.4 实现 SR 主从触发器，并验证功能和存在的时序问题

1. 新建源文件 MS_FLIPFLOP.sch

2. 用原理图方式设计，调用 CSR_LATCH 实现
   画出原理图如下：

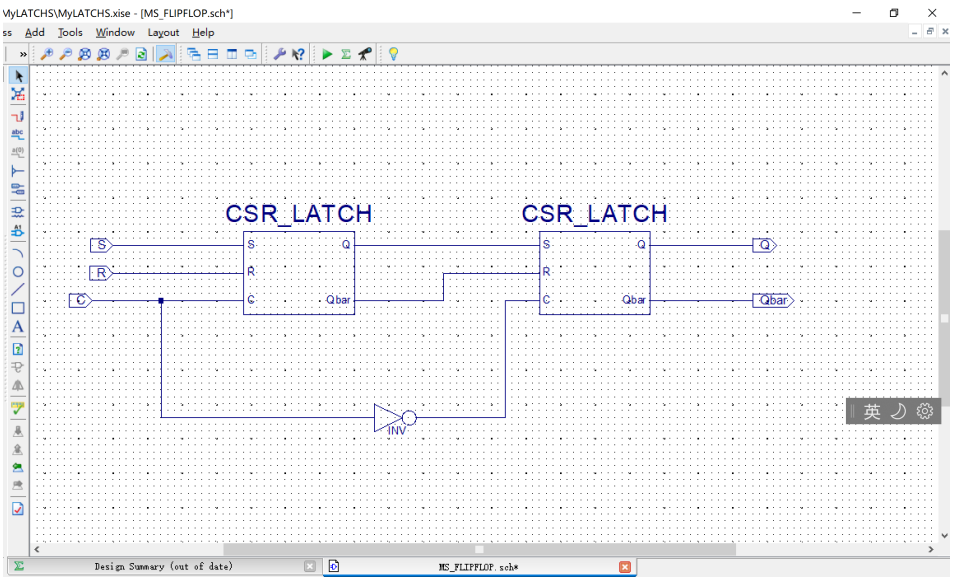


图 26: SR 主从触发器电路图绘制

3. 仿真，仿真波形需要体现一次性采样问题
   新建仿真测试文件 MS_FLIPFLOP_sim.v，里面设置如下代码：

```
1          `timescale 1ns / 1ps
2          module MS_FLIPFLOP_MS_FLIPFLOP_sch_tb();
3
4          // Inputs
5             reg C;
6             reg S;
7             reg R;
8
9          // Output
10            wire Q;
11            wire Qbar;
12
13         // Bidirs
14
```

```
15          // Instantiate the UUT
16             MS_FLIPFLOP UUT (
17                  .Q(Q),
18                  .Qbar(Qbar),
19                  .C(C),
20                  .S(S),
21                  .R(R)
22             );
23          // Initialize Inputs
24
25          initial begin
26              R=0;S=0; #50;
27              R=1;S=0; #50;
28              R=0;S=0; #40;
29              R=0;S=1; #5;
30              R=0;S=0; #5;
31              R=0;S=0; #50;
32              R=1;S=0; #40;
33              R=0;S=0; #50;
34              R=0;S=1; #50;
35              R=0;S=0; #50;
36              R=1;S=1; #50;
37          end
38
39          always begin //产生时钟
40                  C=0;#20;
41                  C=1;#20;
42                  end
43          endmodule
```

得到如下的波形图：

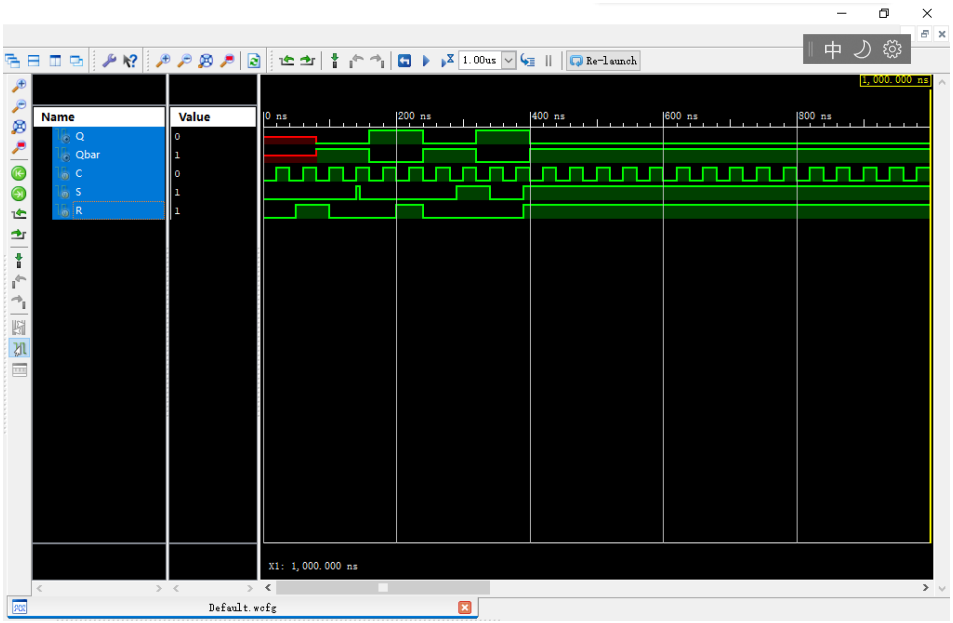


图 27: SR 主从触发器仿真波形图

这个仿真代码同时完成了一次性采样的检验和 SR 主从触发器功能的检验。

首先是一次性采样问题，可以看到在 140ns 时，此时 S = R = 0, 我们给 S 信号施加了一个小脉冲，这时 C 处于上升沿，因此 master 部分存储的值 Q 从 0 变为 1. 同时虽然 S 信号在波动后恢复原样，但存储的值并不会恢复。因此在时钟下降后，slave 部分的值也因此改变，让我们的电路在 160ns 时刻改变了输出。这也正是 SR 主从触发器的一个潜在问题。

随后我们验证 SR 主从触发器的功能，依次尝试 S = 0，R = 1 输出 0，S = 1 R = 0 输出 1，S = 1 R = 1 未定义行为（Q 可能为 1 也可能为 0），经验证可看到结果与预期相符。

3.5 实现 D 触发器，并验证功能

1. 新建源文件 D_FLIPFLOP.sch

2. 用原理图方式设计，调用 NAND3 实现
   画出原理图如下：

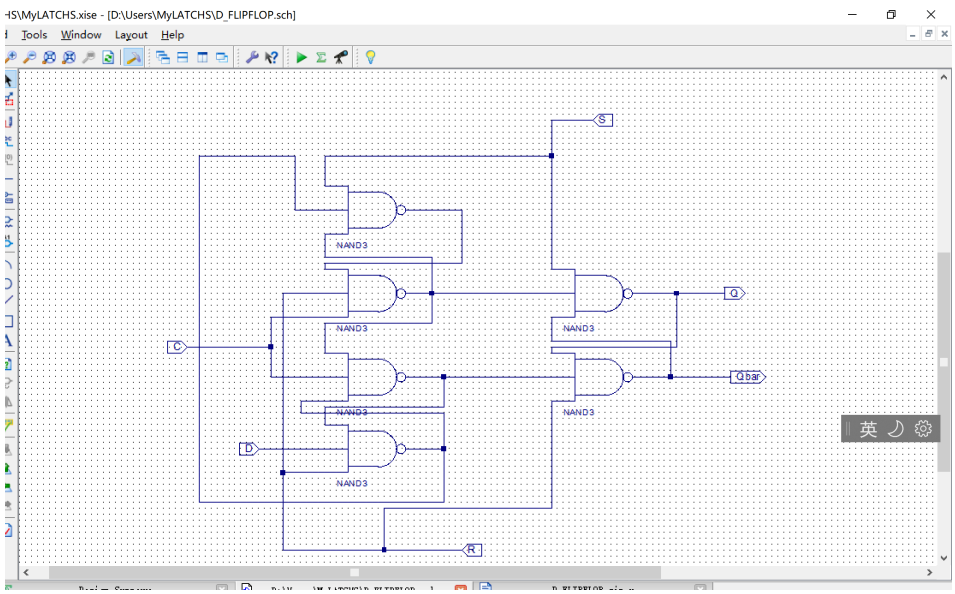


图 28: D 触发器电路图绘制

3. 仿真

新建仿真测试文件 D_FLIPFLOP_sim.v，里面设置如下代码：

```
1          `timescale 1ns / 1ps
2          module D_FLIPFLOP_D_FLIPFLOP_sch_tb();
3
4          // Inputs
5             reg R;
6             reg D;
7             reg C;
8             reg S;
9
10         // Output
11            wire Qbar;
12            wire Q;
13
14         // Bidirs
15
16         // Instantiate the UUT
17            D_FLIPFLOP UUT (
18                 .Qbar(Qbar),
19                 .Q(Q),
20                 .R(R),
```

```
21                  .D(D),
22                  .C(C),
23                  .S(S)
24              );
25          // Initialize Inputs
26               initial begin
27              R = 1;
28              S = 1;
29              D = 0; #150;
30              D = 1; #150;
31              R = 1;
32              S = 1;
33              D = 0; #150;
34              D = 1; #150;
35              R = 0;
36              S = 1;
37              D = 0; #150;
38              D = 1; #150;
39              R = 1;
40              S = 0;
41              D = 0; #150;
42              D = 1; #150;
43              R = 0;
44              S = 1;
45              D = 0; #150;
46              D = 1; #150;
47              end
48              always begin
49              C=0; #50;
50              C=1; #50;
51              end
52          endmodule
```

得到如下仿真波形图

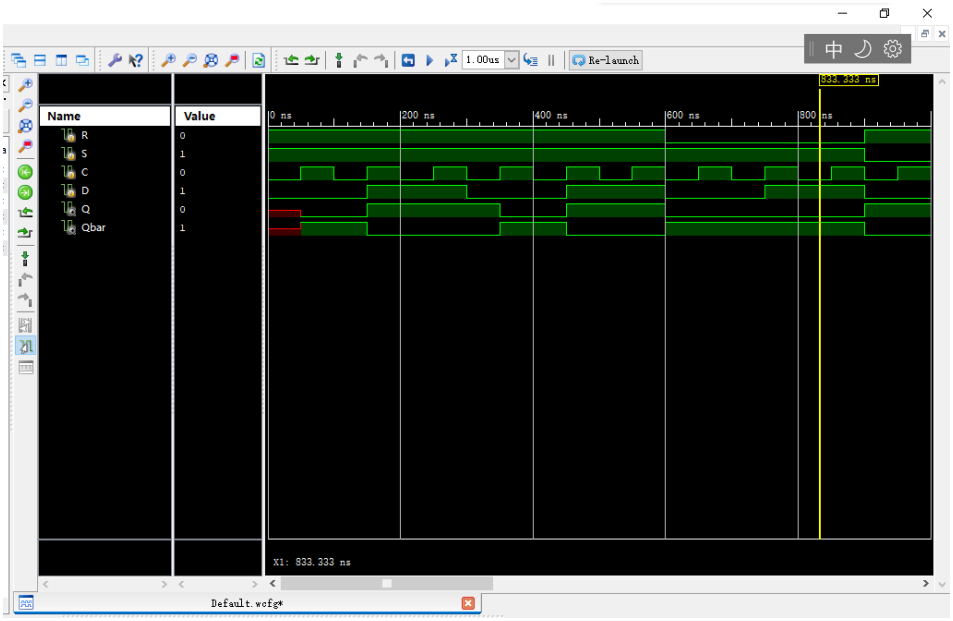


图 29: D 主从触发器仿真波形图

可以看到，当 R = S = 1 时输出 Q 随着输入 D 的变化而变化；在 R = 0，S = 1 时，输出 Q 为 0；R = 1，S = 0 时，输出 Q 为 1（Direct Input，此时输出与 C D 的值无关）和真值表相符，符合预期。

四、实验结果分析

本实验没有使用 SWORD 板，只是进行了仿真。相关结果和 Verilog 代码都已经在前文写出。实验结果基本符合要求：仿真激励波形与真值表都相对应。

五、讨论与心得

在设计一次性采样的时候，最开始我的波形图上并没有我设计的小脉冲，一度以为是进入了某个不为人知的模式。后来不经意地重启电脑，就让小脉冲出现在了我的波形图上。我进一步确信了，重启解决 80% 问题这一哲理！

在验证 D 锁存器的空翻现象时，最开始是直接修改的仿真代码，也就是说隔一段时间把 Qbar 赋值给 D，但是这样的话只能体现出 Qbar 会随着 D 改变，而不能反映出空翻的现象。因此我们需要给 D 锁存器增加外部电路，直接把 Qbar 接给 D，理论上会产生一个振荡电路，而且仿真会看不到波形。

尽信 ppt 不如無 ppt（确信），不要抄 ppt 上的仿真代码，要自己去设计喵。

再接再厉，继续努力！