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 11 月 28 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： 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 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 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

2.2 原理

2.2.1 锁存器

锁存器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：

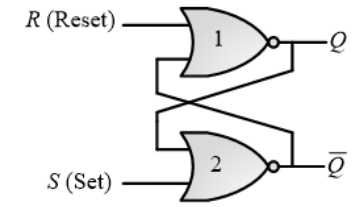
1. 能长期保持给定的某个稳定状态

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

2.2.2 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



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

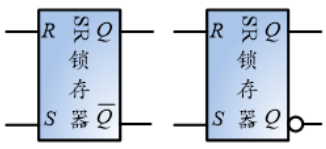
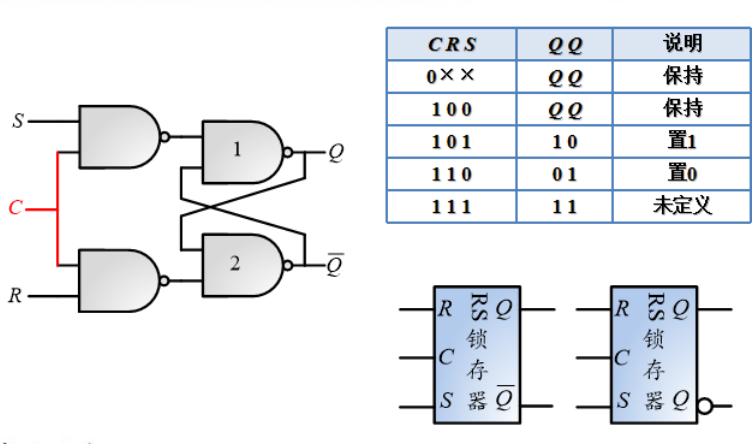


图 1: SR 锁存器

2.2.3 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 $\overline{SR}$ 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。

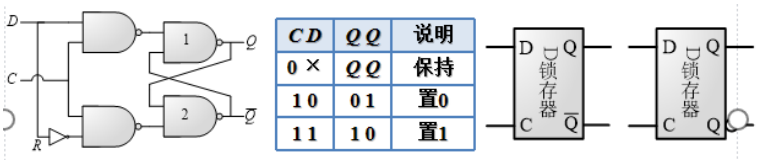


CRS	QQ	说明
0××	QQ	保持
100	QQ	保持
101	10	置1
110	01	置0
111	11	未定义

图 2: 门控 $\overline{SR}$ 锁存器

2.2.4 D 锁存器

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。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 S，输入信号 D 加载到输入端 r。只要控制输入信号为 0，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.



CD	QQ	说明
0×	QQ	保持
10	01	置0
11	10	置1

图 3: D 锁存器

但是 D 锁存器存在空翻现象：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

2.2.5 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）。

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。

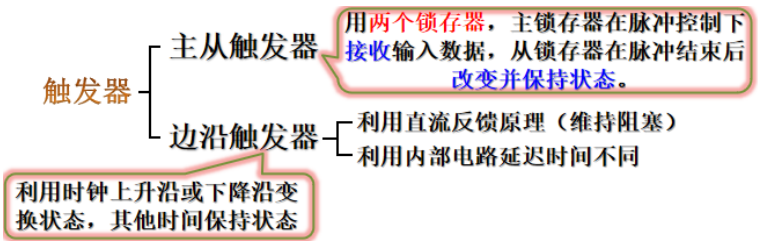


图 4: 触发器的分类

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

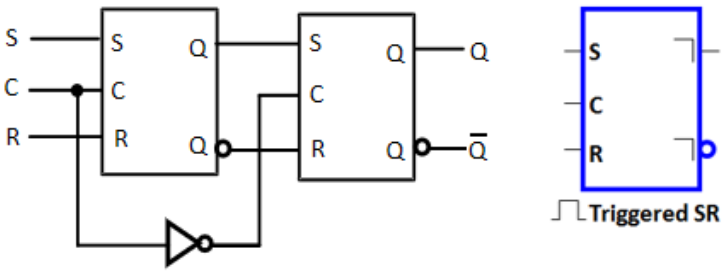
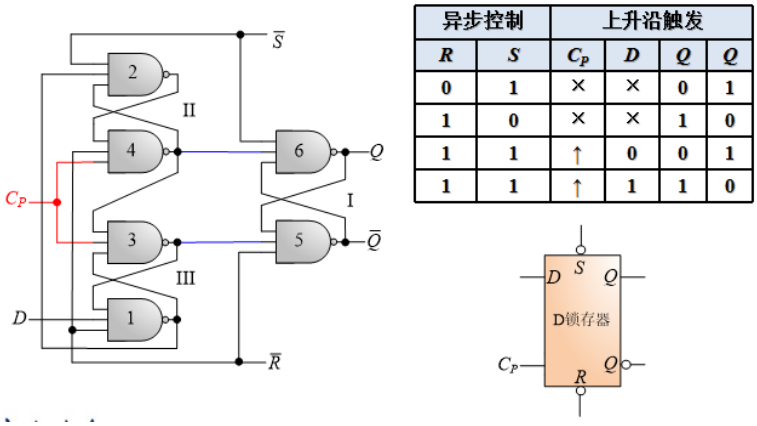


图 5: SR 主从触发器

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 6: D 触发器

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

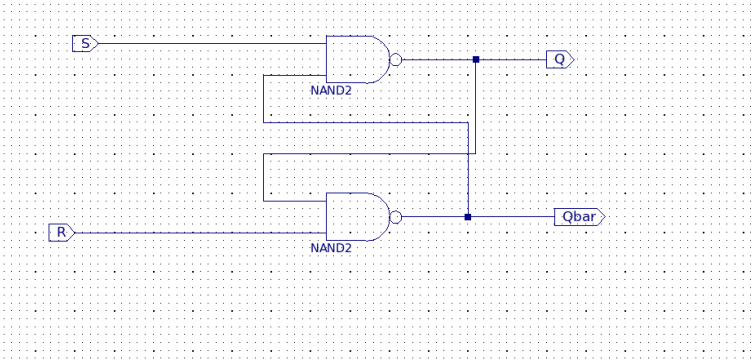


图 7: $\overline{SR}$ 锁存器电路图

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

```
module SR_LATCH_SR_LATCH_sch_tb();

// Inputs
   reg S;
   reg R;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   SR_LATCH UUT (
                .S(S),
                .R(R),
                .Q(Q),
                .Qbar(Qbar)
   );
// Initialize Inputs
       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
```

得到如下波形图

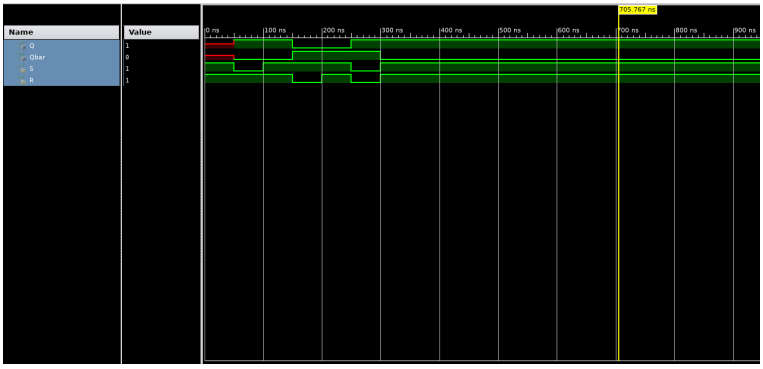


图 8: $\overline{SR}$ 锁存器仿真波形

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

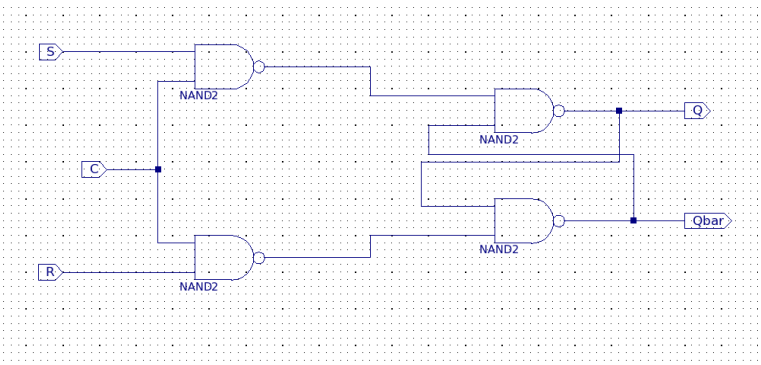


图 9: 门控 SR 锁存器原理图

3. 仿真。新建仿真测试文件 CSR_LATCH_sim.v，里面设置如下代码：

```
module CSR_LATCH_CSR_LATCH_sch_tb();

// Inputs
```

```
    reg C;
    reg S;
    reg R;

// Output
    wire Q;
    wire Qbar;

// Bidirs

// Instantiate the UUT
    CSR_LATCH UUT (
                    .C(C),
                    .S(S),
                    .R(R),
                    .Q(Q),
                    .Qbar(Qbar)
    );
// Initialize Inputs
        initial begin
                    C = 0;
                    S = 0;
                    R = 0;
                    C=0;R=1;S=1;#50;
                    C=1;R=0;S=1;#50;
                    R=0;S=0;#50;
                    R=1;S=0;#50;
                    R=0;S=0;#50;
                    R=1;S=1;#50;
                    R=0;S=0;#50;
                    C=0;R=1;S=1;#50;
                    R=1;S=0;#50;
                    R=0;S=0;#50;
                    R=0;S=1;#50;
                    R=1;S=1;#50;
```

```
R=0;S=0;#50;
end

endmodule
```

得到如下仿真波形：

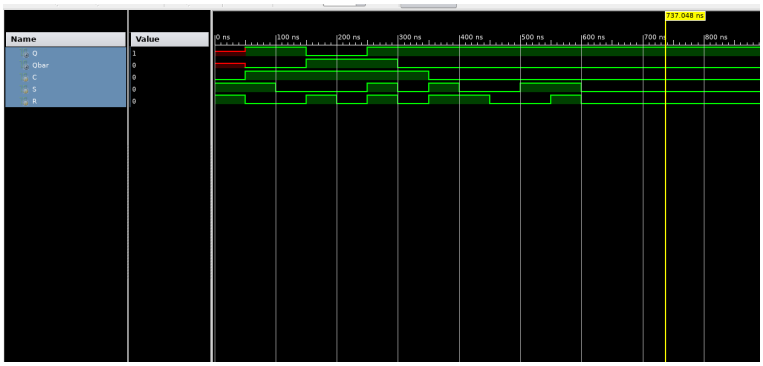


图 10: CSR 锁存器波形

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 ,S=0 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. 生成逻辑符号图

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

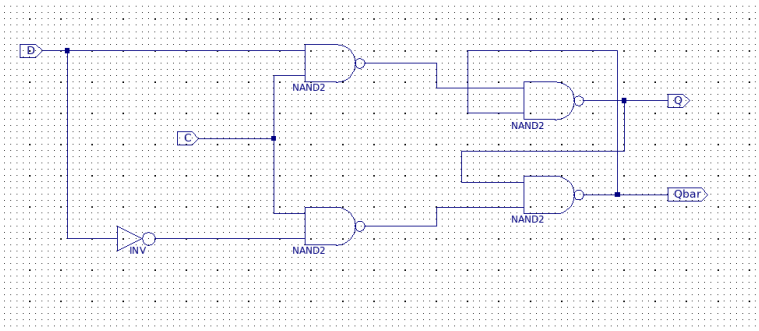


图 11: D 锁存器原理图

3. 仿真：新建仿真测试文件 D_LATCH_sim.v，里面设置如下代码：

```
‘timescale 1ns / 1ps

module D_LATCH_D_LATCH_sch_tb();

// Inputs
    reg C;
    reg D;

// Output
    wire Q;
    wire Qbar;

// Bidirs

// Instantiate the UUT
   D_LATCH UUT (
                .Q(Q),
                .C(C),
                .D(D),
                .Qbar(Qbar)
    );
// Initialize Inputs
        initial begin
                C=1;D=1;#50;
                D=0;#50;
```

```
C=0;D=1;#50;
D=0;
end
endmodule
```

得到如下波形图

图 12: D 锁存器波形图

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。

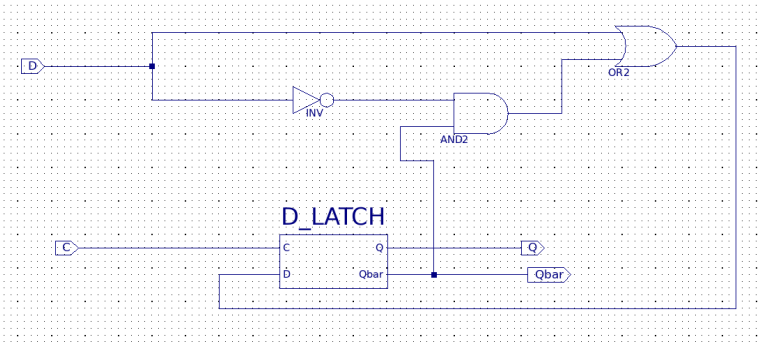


图 13: 验证空翻的电路图

思路就是首先将 D 置为 1，此时 Q 为 1，Qbar 为 0, 根据电路图可知逻辑关系 $D_{input} = D + (\overline{D}Qbar) = D + Qbar$, 又根据 D 锁存器中 $Qbar = \overline{D}$, 可知当我们把 D 置为 0 时，会产生振荡电路。

为了验证正确性，输入如下仿真代码：

```
‘timescale 1ns / 1ps

module D_LATCH_FLIP_D_LATCH_FLIP_sch_tb();

// Inputs
   reg C;
   reg D;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   D_LATCH_FLIP UUT (
                .C(C),
                .Q(Q),
                .Qbar(Qbar),
                .D(D)
   );
// Initialize Inputs
       initial begin
                D=1;#40;
                D=0;
                end
                always begin
                C=0;#20;
                C=1;#20;
                end
endmodule
```

得到仿真波形如下：

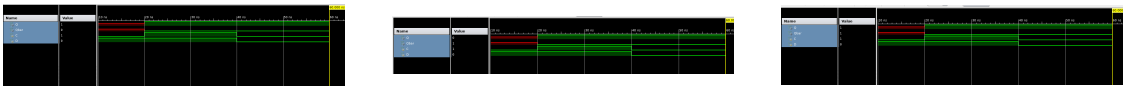


图 14: 仿真波形

可以发现，波形图在 60ns 后即消失，此时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

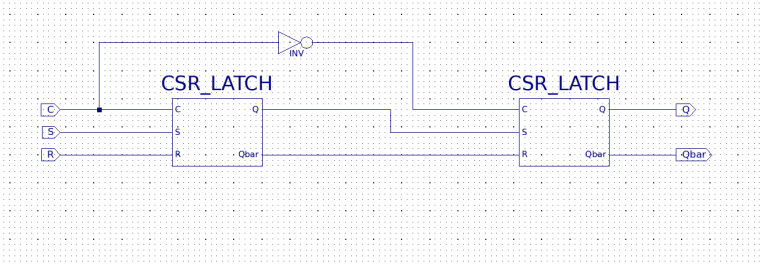


图 15: SR 主从触发器的电路图

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：

```
‘timescale 1ns / 1ps

module MS_FLIPFLOP_MS_FLIPFLOP_sch_tb();

// Inputs
    reg S;
    reg R;
    reg C;

// Output
    wire Q;
```

```
    wire Qbar;

// Bidirs

// Instantiate the UUT
    MS_FLIPFLOP UUT (
                    .Q(Q),
                    .Qbar(Qbar),
                    .S(S),
                    .R(R),
                    .C(C)
    );
// Initialize Inputs
        initial begin
                    R=0;S=0;#50;
                    R=1;S=0;#50;
                    R=0;S=0;#40;
                    R=0;S=1;#5;
                    R=0;S=0;#5;
                    R=0;S=0;#50;
                    R=1;S=0;#40;
                    R=0;S=0;#50;
                    R=0;S=1;#50;
                    R=0;S=0;#50;
                    R=1;S=1;#50;
                    R=0;S=0;#50;
                    end
                    always begin
                            C=0;#20;
                            C=1;#20;
                            end
endmodule
```

得到如下仿真波形

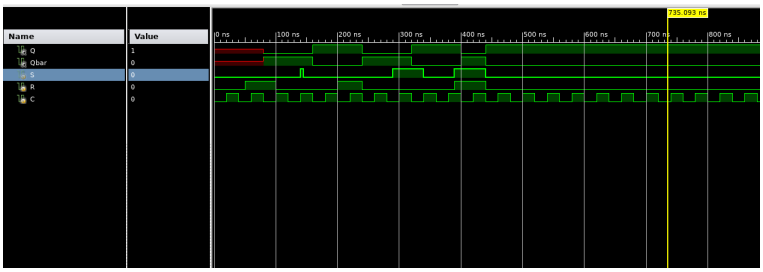


图 16: SR 主从触发器的波形

首先是一次性采样问题，可以看到在 40ns 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，因此主锁存器部分存储的值 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，并验证功能和存在的时序问题

1. 新建源文件 D_FLIPFLOP.sch
2. 用原理图方式设计，调用 NAND3 实现

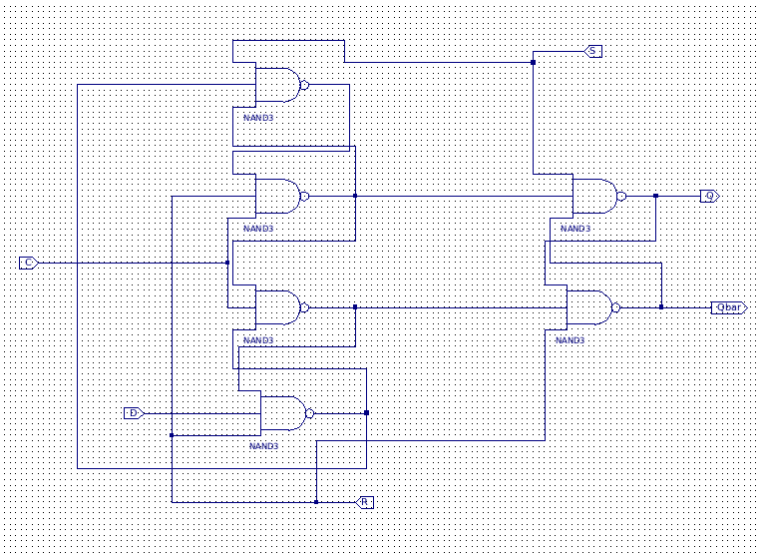


图 17: D 主从触发器原理图

3. 仿真，新建仿真测试文件 D_FLIPFLOP_sim.v，里面设置如下代码：

```
‘timescale 1ns / 1ps

module D_FLIPFLOP_D_FLIPFLOP_sch_tb();

// Inputs
   reg S;
   reg C;
   reg R;
   reg D;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   D_FLIPFLOP UUT (
                  .S(S),
                  .Q(Q),
```

```
                    .Qbar(Qbar),
                    .C(C),
                    .R(R),
                    .D(D)
    );
// Initialize Inputs
        initial begin
                    R=1;
                    S=1;
                    D=0;#150;
                    D=1;#150;
                    R=1;
                    S=1;
                    D=0;#150;
                    D=1;#150;
                    R=0;
                    S=1;
                    D=0;#150;
                    D=1;#150;
                    R=1;
                    S=0;
                    D=0;#150;
                    D=1;#150;
                    R=0;
                    S=1;
                    D=0;#150;
                    D=1;#150;
                    end
                    always begin
                            C=0;#50;
                            C=1;#50;
                            end
endmodule
```

得到仿真波形图

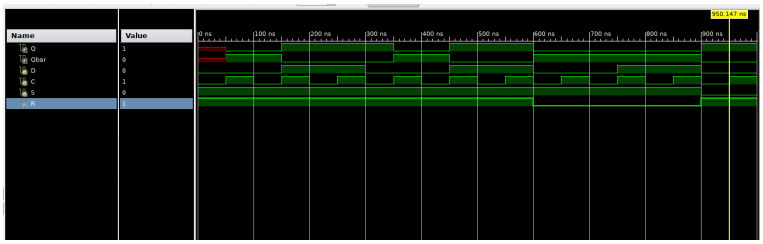


图 18: D 触发器波形图

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

这次实验没有使用 SWORD 板，而是通过仿真来更好理解锁存器和触发器的原理。说实话，得知没有用 SWORD 板后我先是震惊然后有一丝兴奋（ ˘ ）。但是由于没有上板，验收过程要求也更高了，要对老师或助教一一解释仿真波形的过程及原因。但正因为如此，在准备验收之前我把锁存器触发器的原理以及一次性采样的原因过了一遍，最后很顺利通过验收，不得不说我对这部分的知识理解更加深刻了。总而言之，继续努力！