



南京邮电大学
Nanjing University of Posts and Telecommunications

电工电子实验报告

课程名称: 电工电子实验 (二)
实验项目: 数据选择器的应用
触发器的应用

学 院: 贝尔英才学院
班 级:
学 号:
姓 名:
学 期: 2022-2023 学年第 2 学期

数据选择器的应用

一、实验目的

- 1. 使用 ISE 软件完成组合逻辑设计的输入并仿真
- 2. 掌握 Testbench 中组合逻辑测试文件的写法
- 3. 下载并测试实现的逻辑功能

二、实验仪器

ISE14.7 仿真软件

三、实验内容

用 Verilog HDL 设计一个具有异步置位和异步复位功能的 D 触发器，完成设计模块、Testbench，并保存仿真波形。

四、实验原理

(1) 真值表，如表 1

使能输入	输入			输出
\overline{EN}	A2	A1	A0	Y
1	Φ	Φ	Φ	0
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7

表 1

(2) 设计代码：

```
module select8l(enable,sel,Din,Dout
);
input enable;
input [2:0] sel;
input [7:0] Din;
output Dout;
reg Dout;
always@(enable or sel or Din)
    if(!enable)
        case(sel)
            3'b000 : Dout=Din[0];
            3'b001 : Dout=Din[1];
            3'b010 : Dout=Din[2];
            3'b011 : Dout=Din[3];
            3'b100 : Dout=Din[4];
            3'b101 : Dout=Din[5];
            3'b110 : Dout=Din[6];
            3'b111 : Dout=Din[7];
        endcase
    else
        Dout = 0;
endmodule
```

图 1 8 选 1 数据选择器设计代码

测试代码:

```
module select8ltest;

    // Inputs
    reg enable;
    reg [2:0] sel;
    reg [7:0] Din;

    // Outputs
    wire Dout;

    // Instantiate the Unit Under Test (UUT)
    select8l uut (
        .enable(enable),
        .sel(sel),
        .Din(Din),
        .Dout(Dout)
    );

    initial begin
        // Initialize Inputs
        enable = 0;
        sel = 0;
        Din = 0;

        // Wait 100 ns for global reset to finish
        #100 Din[0] = 0;
        Din[1] = 1;
        Din[2] = 0;
        Din[3] = 1;
        Din[4] = 0;
        Din[5] = 1;
        Din[6] = 0;
        Din[7] = 1;
        #100 sel = 3'b000;
        #100 sel = 3'b001;
        #100 sel = 3'b010;
        #100 sel = 3'b011;
        #100 sel = 3'b100;
        #100 sel = 3'b101;
        #100 sel = 3'b110;
        #100 sel = 3'b111;

        // Add stimulus here

    end
endmodule
```

图 2 8 选 1 数据选择器测试代码

仿真结果：

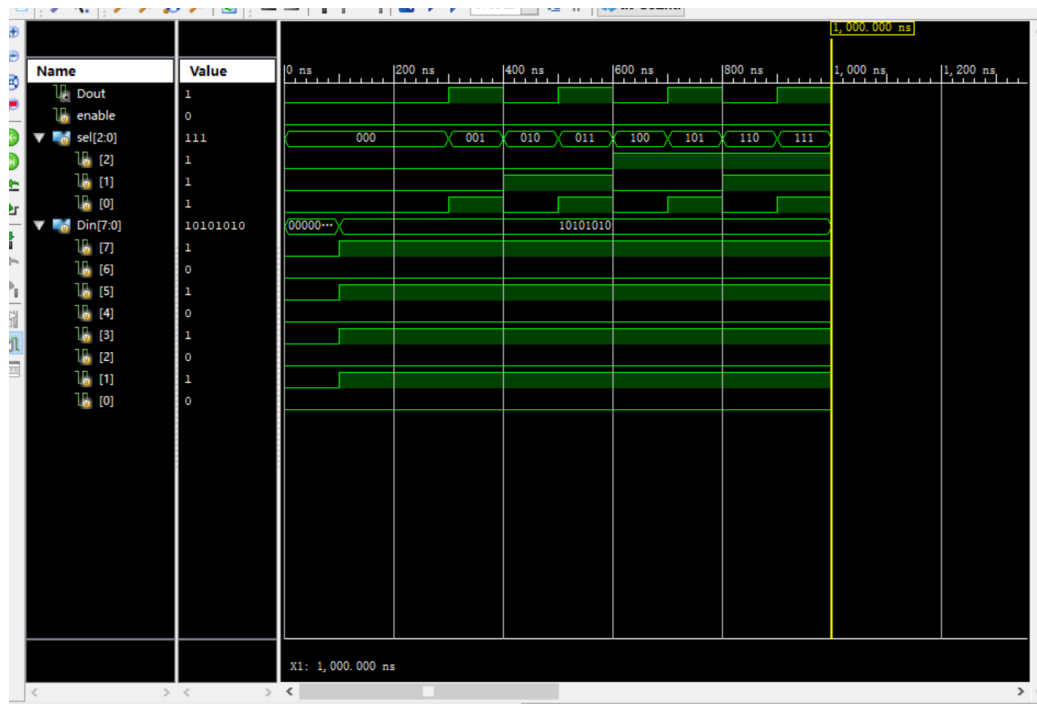


图 3 8 选 1 数据选择器仿真截图

五、实验结果

8 选 1 数据选择器手绘波形图：

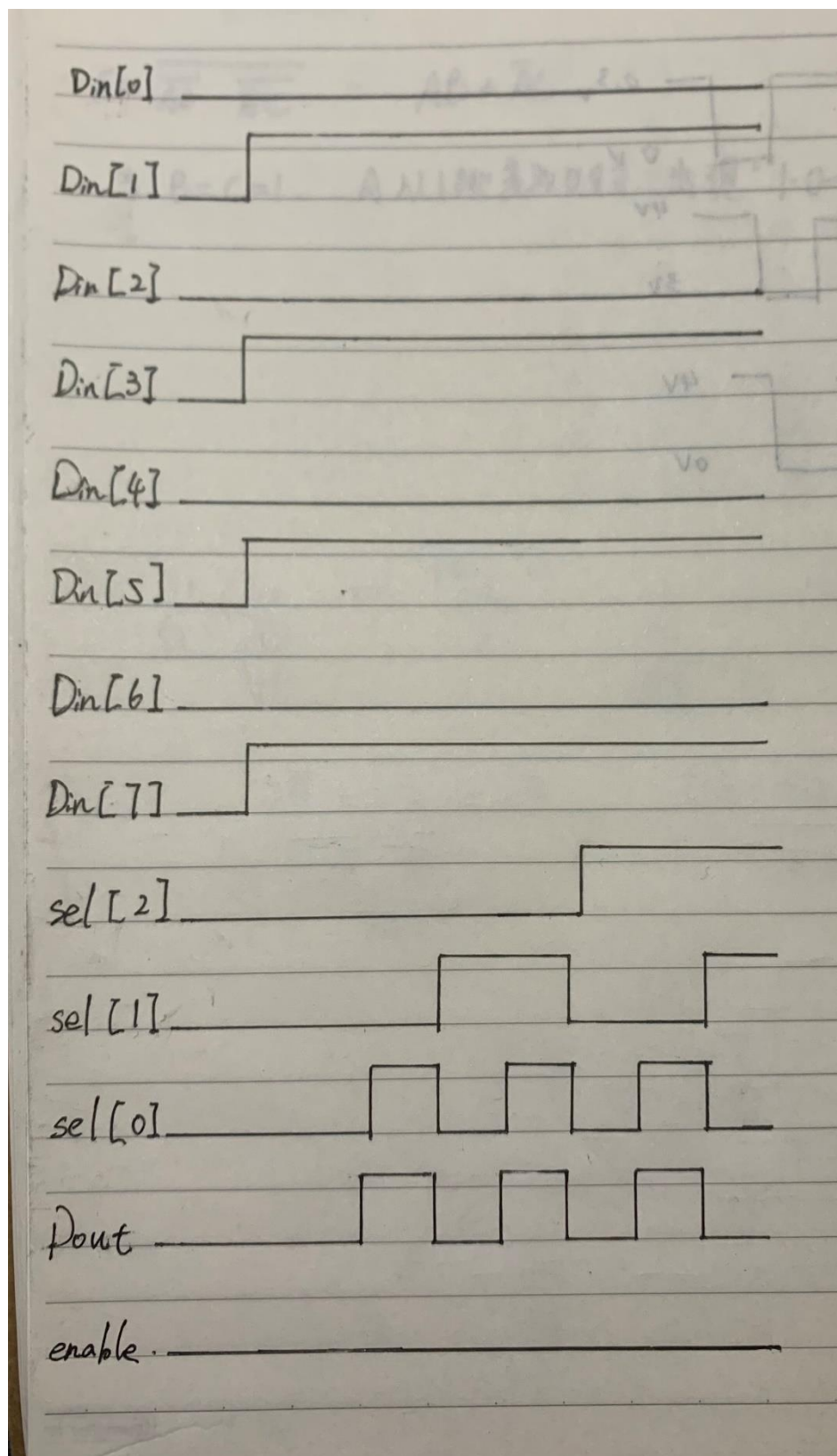


图 4 8 选 1 数据选择器手绘波形

六、实验小结

通过本次实验让我对 8 选 1 数据选择器有了更深层次的认识, 进一步增强了对语言编程实现逻辑功能的认识。

触发器的应用

一、实验目的

1. 使用 ISE 软件完成时序逻辑电路的设计输入并仿真。
2. 掌握 testbench 中时序逻辑测试文件的写法。
3. 下载并测试实现的逻辑功能。

二、实验仪器

ISE14.7 仿真软件

三、实验内容

用 Verilog HDL 设计一个具有异步置位和异步复位功能的 D 触发器，完成设计模块、Testbench，并保存仿真波形。Testbench 能够测试到电路设计要求的所有功能、下载到 FPGA、完成硬件测试并实现电路功能。

四、实验原理

(1) 具有异步置位和异步复位功能 D 触发器的功能表，如表 2：

S_d	R_d	clk	Q	Qn
0	Φ	Φ	1	0
1	0	Φ	0	1
1	1	\uparrow	D	$\sim D$
1	1	0, 1	保持	保持

(2) 设计代码：

```
module DFF(clk,set,clr,Din,Dout
);
input clk,set,clr;
input Din;
output reg Dout;
always @(posedge clk or negedge clr or negedge set)
begin
    if(!clr==1'b1)
        Dout<=1'b0;
    else if(!set==1'b1)
        Dout<=1'b1;
    else
        Dout<=Din;
    end
endmodule
```

图 5 具有异步置位和异步复位功能 D 触发器设计代码

(3) 测试代码

```

module DFFv;

    // Inputs
    reg clk;
    reg set;
    reg clr;
    reg Din;

    // Outputs
    wire Dout;

    // Instantiate the Unit Under Test (UUT)
    DFF uut (
        .clk(clk),
        .set(set),
        .clr(clr),
        .Din(Din),
        .Dout(Dout)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        set = 1;
        clr = 0;
        Din = 1;

        // Wait 100 ns for global reset to finish
        #100 clr=1'b0;
        set=1'b1;
        clk=1'b1;
        Din=1'b1;
        //1
        #100 clr=1'b1;
        set=1'b0;
        clk=1'b0;
        Din=1'b0;
        //2
        //异步复位
        #100 clr=1'b1;
        set=1'b0;
        clk=1'b1;
        Din=1'b0;
        //3
        #100 clr=1'b0;
        set=1'b0;
        clk=1'b0;
        Din=1'b0;
        //4
        //异步置位
        #100 clr=1'b1;
        set=1'b1;
        clk=1'b1;
        Din=1'b0;
        //5
        #100 clr=1'b1;
        set=1'b1;
        clk=1'b0;
        Din=1'b1;
        //6
        #100 clr=1'b1;
        set=1'b1;
        clk=1'b1;
        Din=1'b1;
        //7
    end

```



```

...
#100 clr=1'b1;
      set=1'b1;
      clk=1'b0;
      Din=1'b1;
      //8
      //保持
      // Add stimulus here

end

endmodule

```

图 6 具有异步置位和异步复位功能 D 触发器测试代码

五、实验结果

(1) 仿真截图：

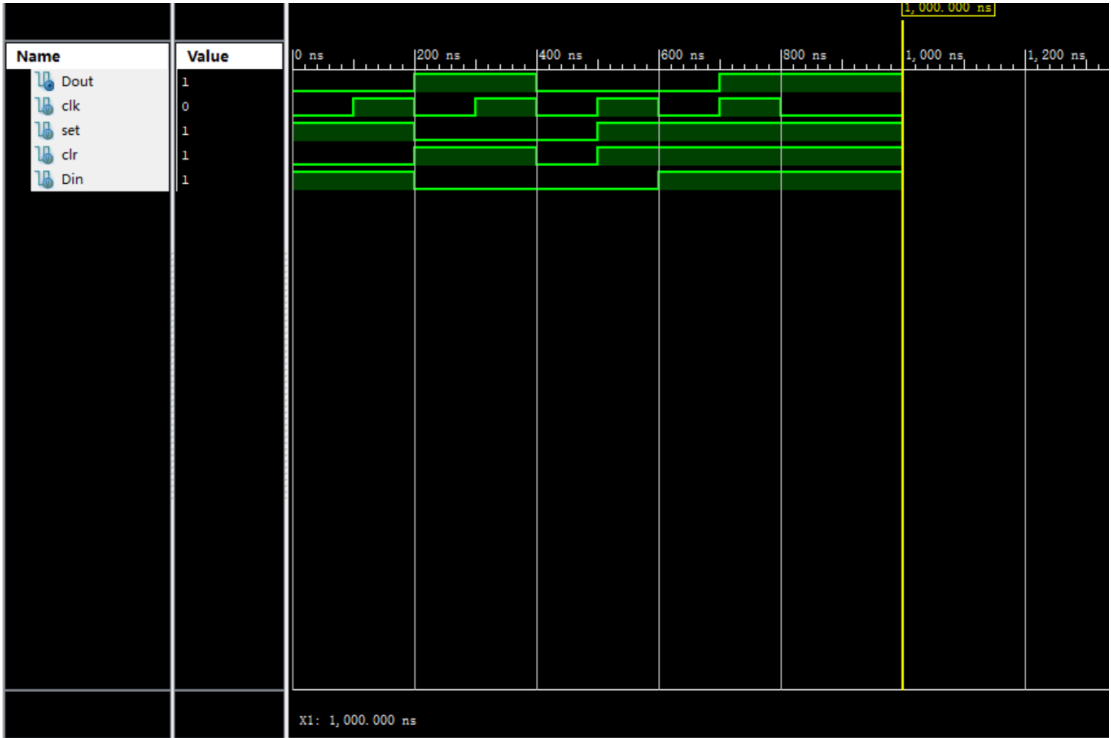


图 7 具有异步置位和异步复位功能 D 触发器仿真截图

(2) 手绘波形图：

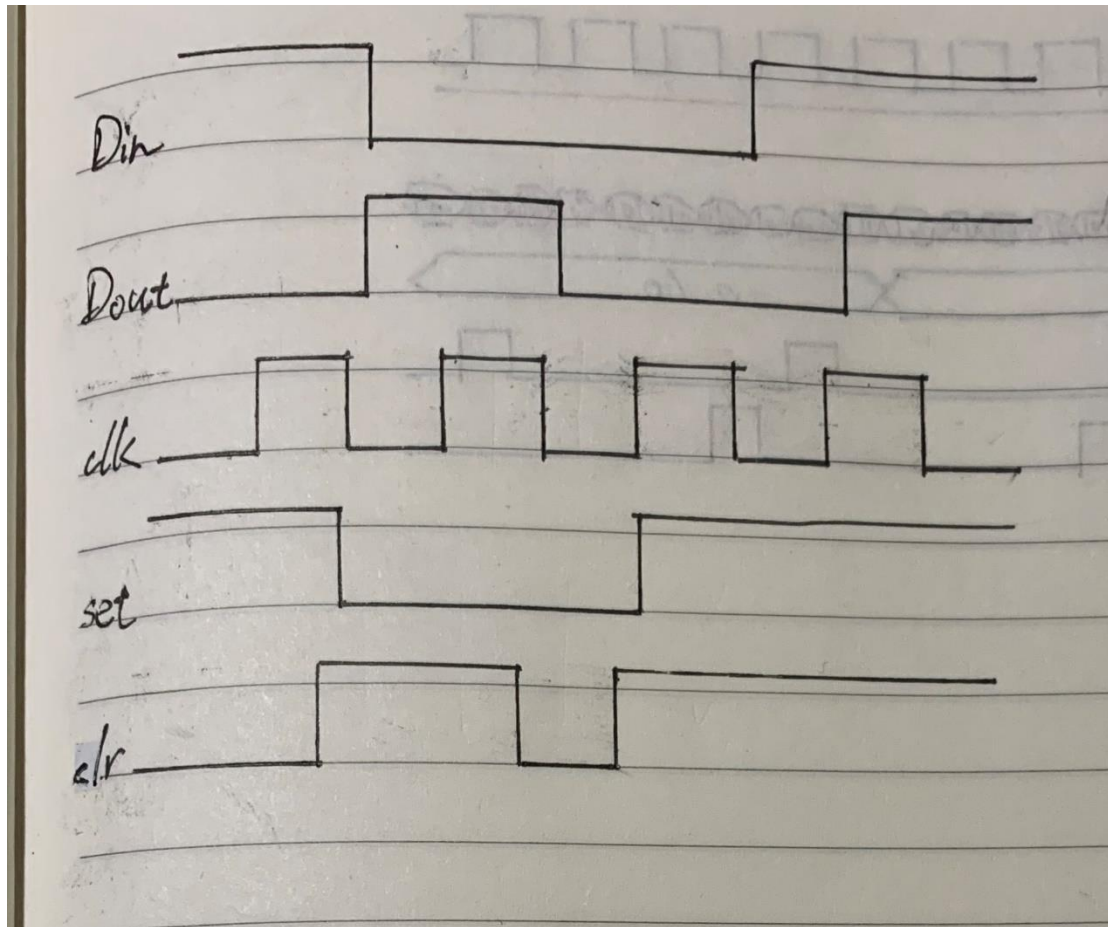


图 8 具有异步置位和异步复位功能 D 触发器手绘波形

六、实验拓展

1. 7bit 可控数字延时器

(1) 设计代码:

```

////////////////////////////////////
module shift7(clk,clr,Din,addr,Dout
);
input clk,clr,Din;
input [2:0] addr;
output Dout;
reg[6:0] reg7;
reg reg1;
assign Dout = reg1;
always@(posedge clk or negedge clr)
begin
    if(!clr)
        reg7 <= 7'b00000000;
    else if(Din)
        reg7 <={reg7[5:0],Din};
    else
        reg7 <=(reg7<<1);
    end
always@( addr or Din or reg7 )
begin
    case( addr )
        3'b000:reg1 <= Din;
        3'b001:reg1 <= reg7[0];
        3'b010:reg1 <= reg7[1];
        3'b011:reg1 <= reg7[2];
        3'b100:reg1 <= reg7[3];
        3'b101:reg1 <= reg7[4];
        3'b110:reg1 <= reg7[5];
        default:reg1 <= Din;
    endcase
end
endmodule

```

图9 7bit 可控数字延时器设计代码

(2) 测试代码:

```

module shift7t;

    // Inputs
    reg clk;
    reg clr;
    reg Din;
    reg [2:0] addr;

    // Outputs
    wire Dout;
    reg[2:0] counter;
    // Instantiate the Unit Under Test (UUT)
    shift7 uut (
        .clk(clk),
        .clr(clr),
        .Din(Din),
        .addr(addr),
        .Dout(Dout)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        clr = 0;
        Din = 0;
        addr = 0;
        counter=3'b000;
        // Wait 100 ns for global reset to finish
        #100;
        clr=1;
        addr=3'b001;
        #400 addr =3'b010;
        #400 addr =3'b011;
        #400 addr =3'b100;
        #400 addr =3'b101;
        #400 addr =3'b110;
        // Add stimulus here

    end

    always@(posedge clk)
    begin
        if(counter<3'b110)
            begin
                counter=counter+1;
                Din=0;
            end
        else
            begin
                counter=0;
                Din=1;
            end
        end
    end
    always #20 clk=~clk;
endmodule

```

图 10 7bit 可控数字延时器测试代码

(3) 仿真波形:

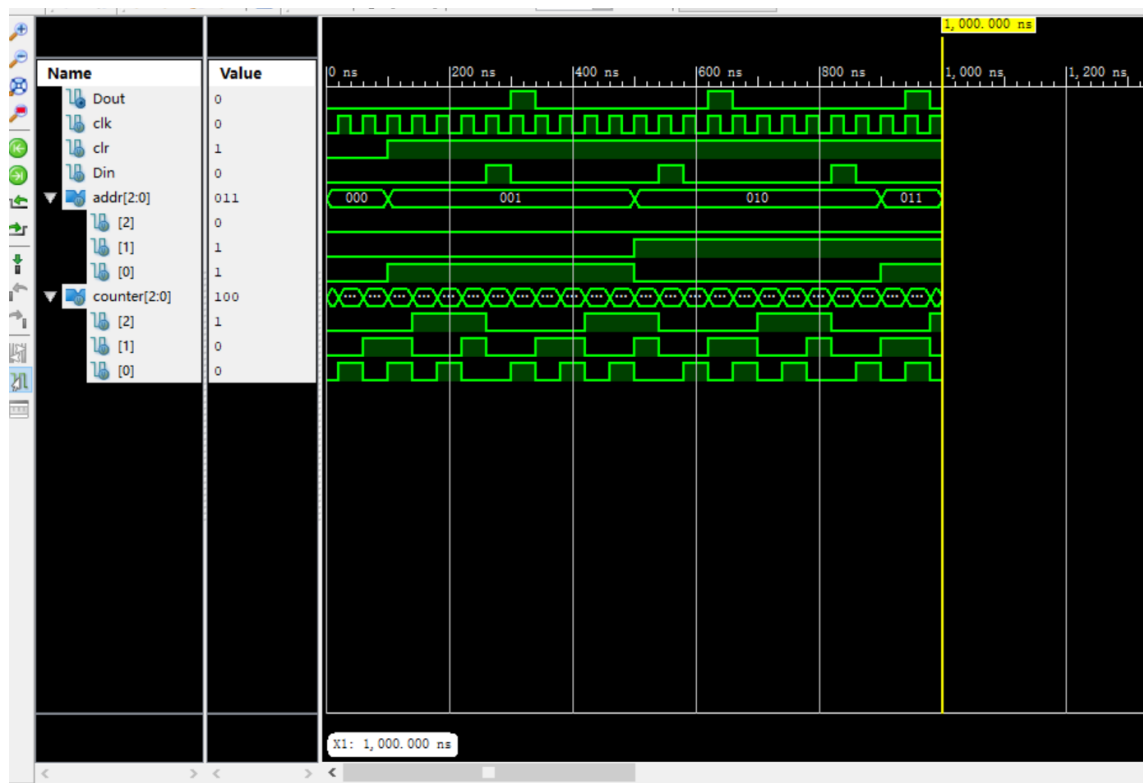


图 11 7bit 可控数字延时器仿真截图

(4) 手绘波形:

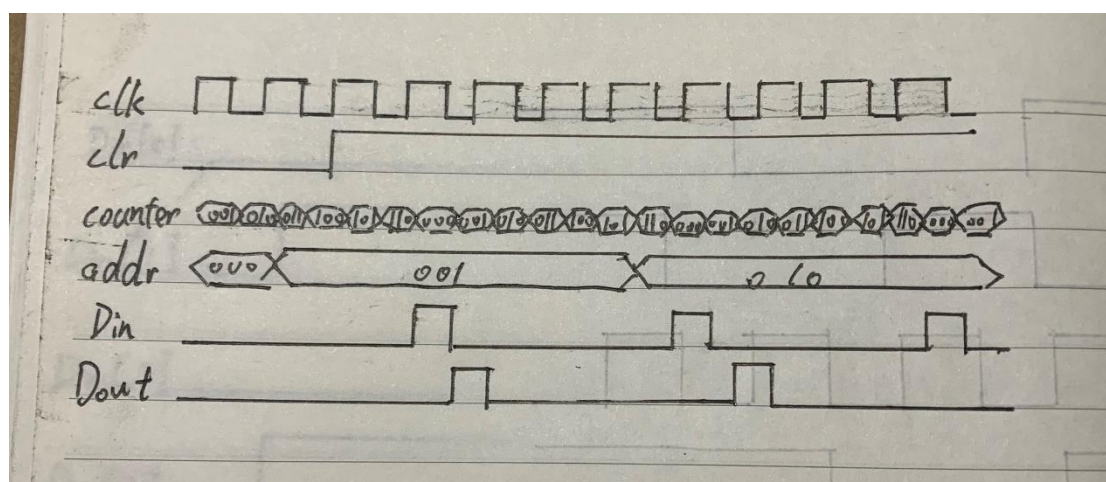


图 12 7bit 可控数字延时器手绘波形

2. 同步复位 DFF

(1) 设计代码:

```

module tongbu(d,clk,q
);
input d,clk;
output q;
reg q;
always@(posedge clk)
q<=d;
endmodule

```

图 13 同步复位 DFF 设计代码

(2) 测试代码:

```

module tongbul;

// Inputs
reg d;
reg clk;

// Outputs
wire q;

// Instantiate the Unit Under Test (UUT)
tongbu uut (
    .d(d),
    .clk(clk),
    .q(q)
);

initial begin
    // Initialize Inputs
    d = 0;
    clk = 0;

    // Wait 100 ns for global reset to finish
    #100
    clk=1;
    d=0;
    #100
    clk=0;
    d=1;
    #100
    clk=1;
    d=1;
    #100
    clk=0;
    d=0;
    #100
    clk=1;
    d=0;

```

```

#100
    clk=0;
    d=1;
#100
    clk=1;
    d=1;
#100
    clk=0;
    d=0;

    // Add stimulus here

end

endmodule

```

图 14 同步复位 DFF 测试代码

(3) 仿真波形:

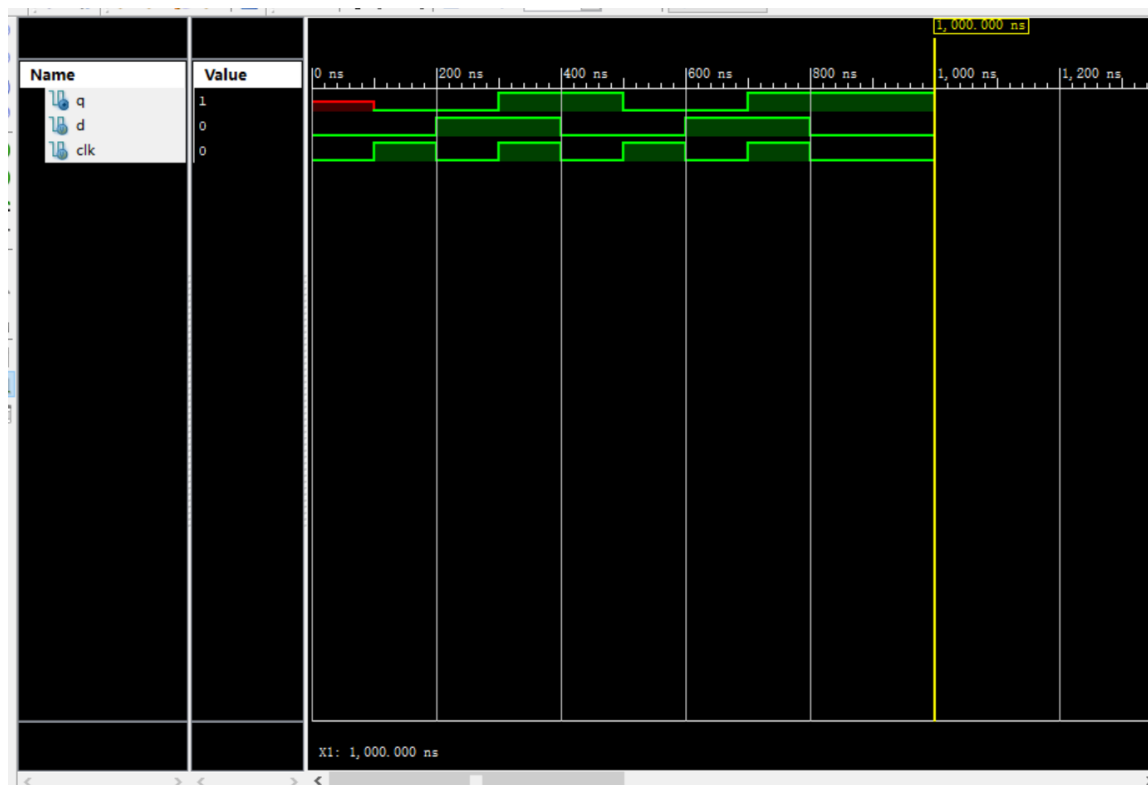


图 15 同步复位 DFF 仿真截图

手绘图:

