数电实验3：触发器计数器

# 一、实验目的

1.熟悉D，JK，T触发器的逻辑功能。

2.基于Multisim软件，使用D触发器实现移位寄存器。

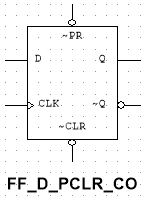
3.了解阻塞赋值与非阻塞赋值的区别。

4.学习如何使用Verilog实现移位寄存器并在FPGA上验证，学习如何使用Vivado软件调试Verilog程序。

5.用Verilog语言实现带清零功能的4位二进制计数器

# 二、实验原理

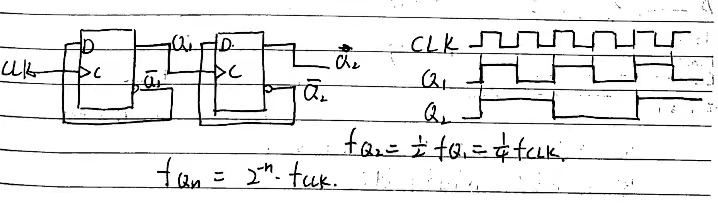
**1. 移位寄存器**



下为D触发器的真值表：

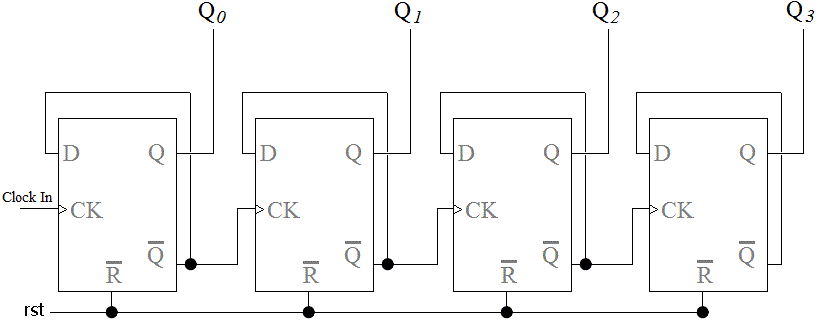
|  |  |  |  |
| --- | --- | --- | --- |
| 输入D | 输入时钟 | 输出Q | 输出~Q |
| 0 | 上升沿 | 0 | 1 |
| 0 | 其他 | 0 | 1 |
| 1 | 上升沿 | 1 | 0 |
| 1 | 其他 | 0 | 1 |

利用D触发器分频：

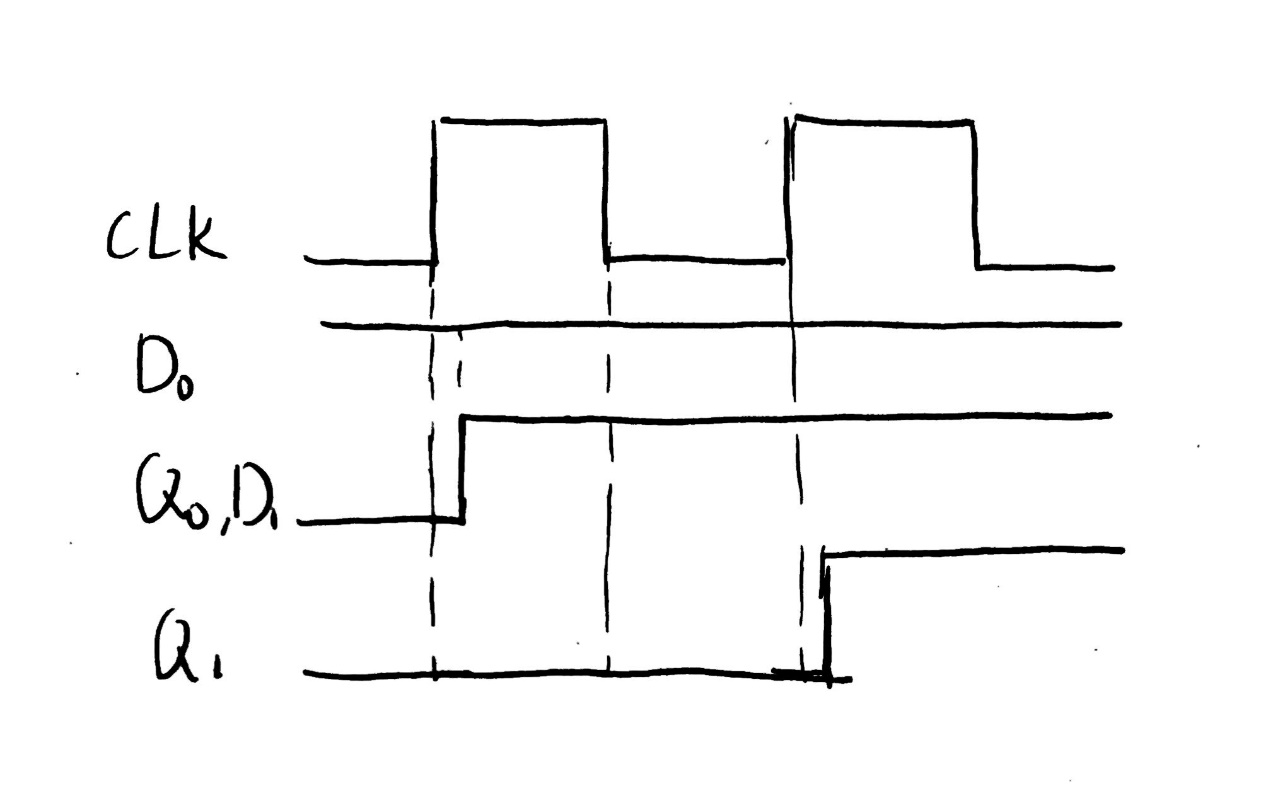


原理如图，第n级的Q输出频率就是。

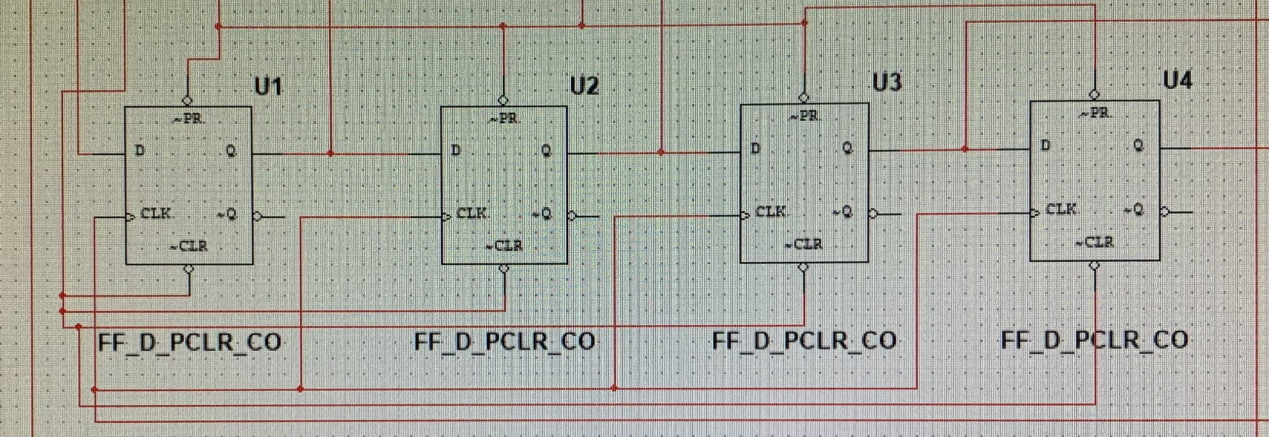
利用D触发器实现2进制计数：

原理与分频类似，但将下一级的CLK输出换为上一级的~Q（为了错开上升沿），使计数器不会同时加1。

利用D触发器实现位移寄存器：



当所有D触发器绑定同一个时钟时，由于触发器内部固有延迟，会存在一个，利用这个迟滞可以使下一级的Q在下一个时钟周期变为高电平，电路如图：



**2.阻塞赋值与非阻塞赋值**

1.阻塞（blocking）赋值方式（如 b=a）：

①赋值语句立即执行，执行完毕后才执行下一条语句（即为阻塞的含义，依次顺序执行）；

②左侧值在赋值语句执行完后立即改变。

2.非阻塞（Non\_blocking）赋值方式（如 b<=a）：

①语句执行到此时，先计算“<=”右侧值，但不立即赋值给左侧；

②always块结束后才完成此次赋值操作；

③这是时序逻辑模块最常用的赋值方法。

1. **基于Verilog的移位寄存器**

在实际的电路设计开发当中，人们希望在烧制FPGA之前，先对编好的Verilog程序进行仿真，分析时序逻辑关系是否符合预期。本次实验可利用Verilog程序进行移位寄存器仿真，观察时序逻辑关系。

1. **基于Verilog的计数器**

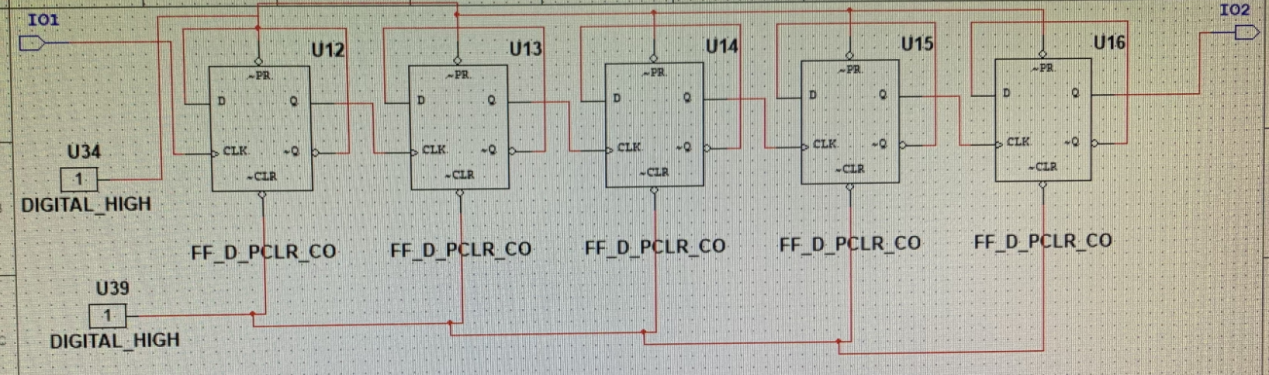
利用D锁存器的时序逻辑关系，可实现基于Verilog的计数器。

# 三、实验内容

**1. 移位寄存器**

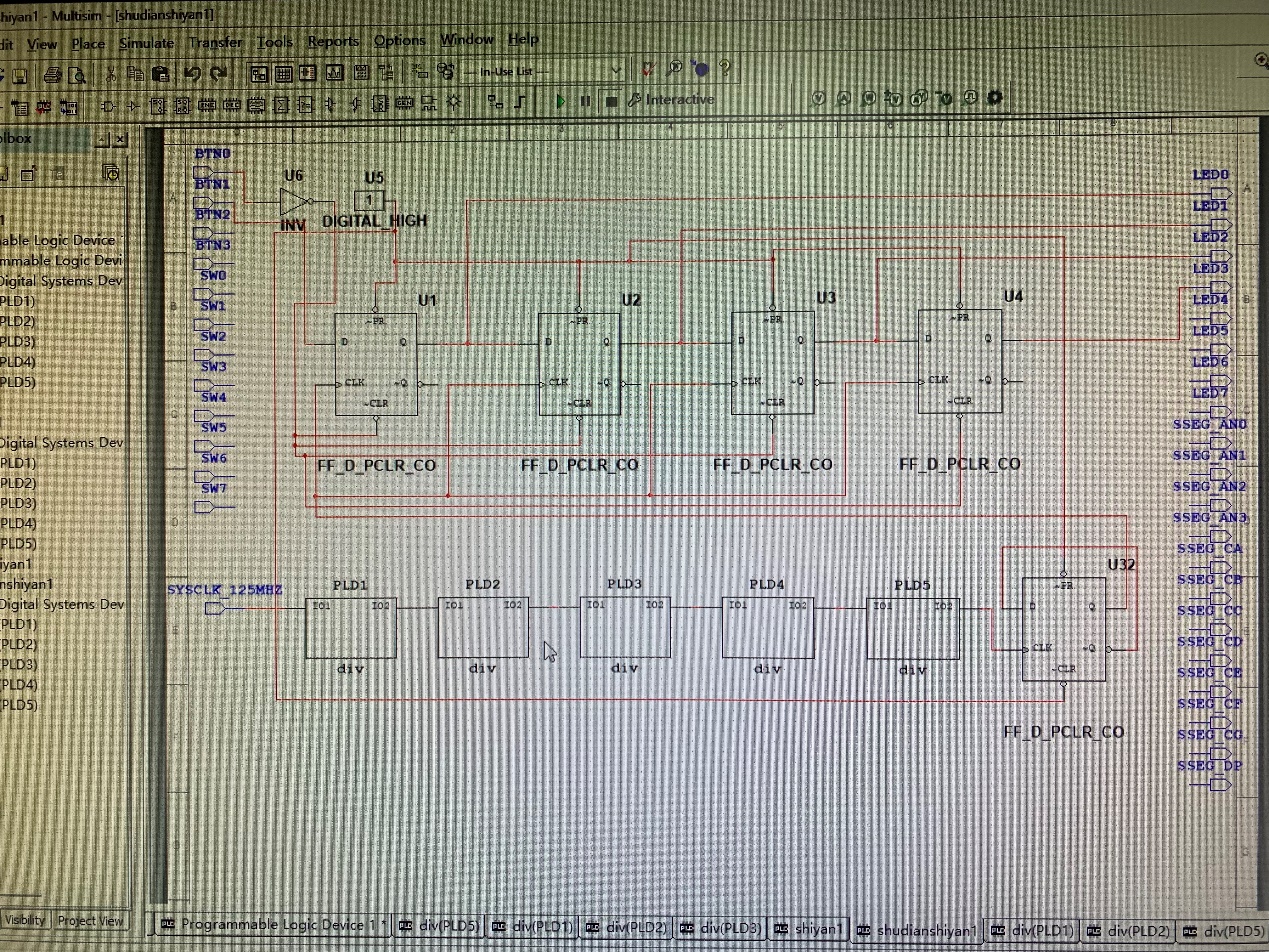
电路图如下：

分频电路：



如此的子电路，一共用了5个加1个DFF，使频率降为，约为1.8Hz。

实验一总电路：



**2.阻塞赋值与非阻塞赋值**

(1) 打开Vivado软件，新建项目，加入CANVAS中提供的两个文件blocking.v和freq\_div.v和blocking.xdc。编译并导出到DSDB运行。

(2) 打开Vivado软件，新建项目，加入CANVAS中提供的两个文件nonblocking.v和freq\_div.v和nonblocking.xdc。编译并导出到DSDB运行。

(3) 基于阻塞赋值与非阻塞赋值的区别，分析以上两个实验的结果为何不同。

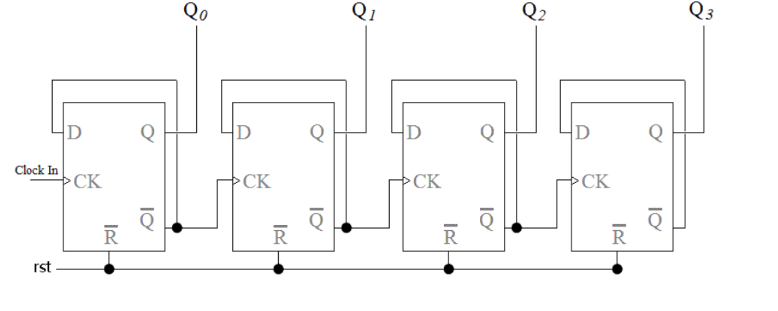
**3.基于Verilog的移位寄存器**

打开Vivado软件，新建项目，加入CANVAS中提供的两个文件shifter4.v和shifter4\_tb.v。按照PPT中“Verilog仿真时序分析演示”介绍的方法，用shifter4\_tb.v对shifter4.v的时序功能进行调试，分析并记录时序图，确认结果符合移位寄存器的功能。

**4..基于Verilog的计数器**

(1) 新建Vivado项目，加入Canvas下载的counter1.v和counter1.xdc文件。阅读counter1.v和counter1.xdc，阅读并理解程序，编译并导出到FPGA运行，根据实验结果验证与自己理解的功能是否一致。

(2) 基于以上的counter1程序，按电路图设计4位计数器，电路图如下：



# 四、实验结果及分析

## 实验一

见文件：实验一：位移寄存器 > 实验一结果

结果分析：

如果长按Button 1，则会看到LED0~LED3依次点亮，直到Button 1抬起，LED0~LED3依次熄灭。按Button 0可以清零所有LED灯。

## 实验二

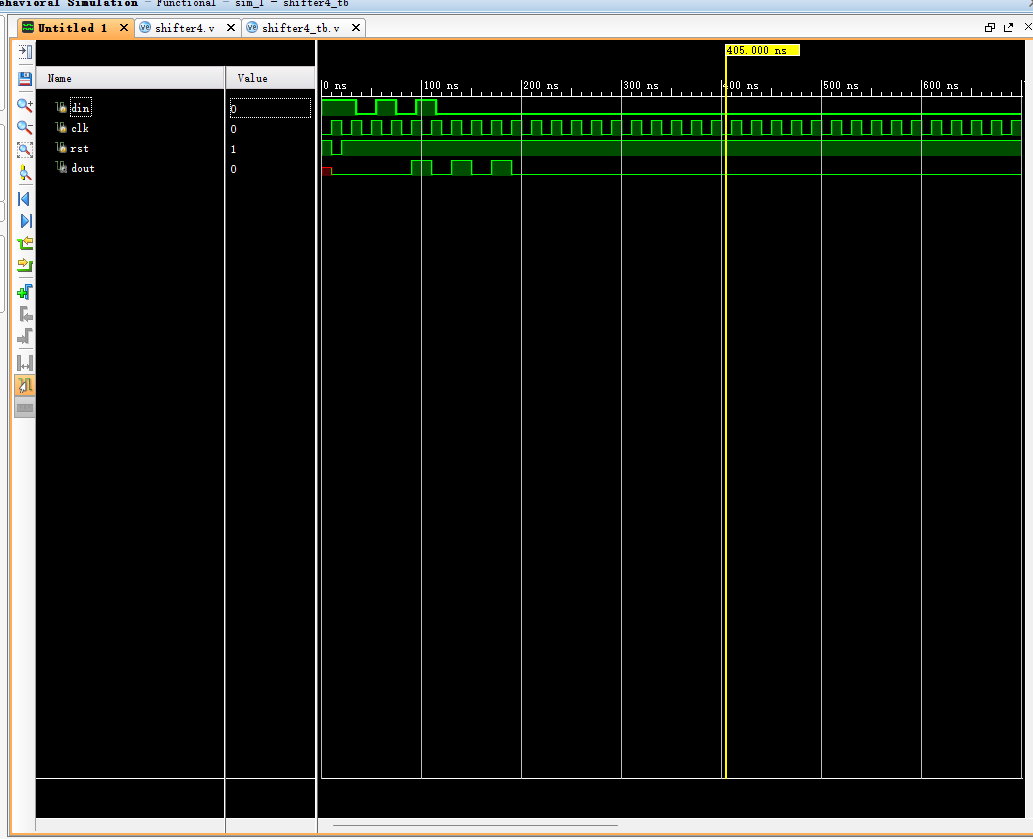
见文件：实验二：阻塞赋值和非阻塞赋值 > 阻塞赋值和非阻塞赋值

结果分析：

阻塞赋值在always块一次循环内就完成负值，而非阻塞赋值在always快完成一次循环后才赋值。

故在运行阻塞赋值程序时，所有LED灯会一起亮一起熄灭，而在运行非阻塞赋值程序时，LED灯会在间隔一个时钟周期（1s）依次亮起依次熄灭。

## 实验三



结果分析：

din，clk，ret按程序赋给了初值，dout初值随机，随后按设定程序运行，时钟信号以20ns为周期，res在10ns时变为0，并调用程序给t赋值，再等10ns变回1。程序成功实现了移位寄存器的功能。

## 实验四

原始的程序结果见：实验四：基于Verilog的计数器 > 原始代码

最终结果见：实验四：基于Verilog的计数器 > 最终程序

最终程序文件见：实验四：基于Verilog的计数器 > counter1.v和counter1.xdc

程序注释：

module top (input clk,

input rst,

output out,

output out1,

output out2,

output out3

);

integer num = 0;

reg clkdiv = 0;

always @(posedge clk) begin

if (num < 62500000) begin//将频率分频为1Hz

num = num + 1;

end

else begin

num = 0;

clkdiv = ~clkdiv;

end

end

counter counter\_inst(.clk(clkdiv),.rst(rst),.out(out),.out1(out1),.out2(out2),.out3(out3));

endmodule

module counter (input clk,

input rst,

output out,

output out1,

output out2,

output out3

);//计数器主程序

wire q;

wire qn;

dff dff(.d(qn),

.clk(clk),

.rst(rst),

.q(q),

.qn(qn)

);

dff dff1(.d(qn1),

.clk(qn),

.rst(rst),

.q(q1),

.qn(qn1)

);

dff dff2(.d(qn2),

.clk(qn1),

.rst(rst),

.q(q2),

.qn(qn2)

);

dff dff3(.d(qn3),

.clk(qn2),

.rst(rst),

.q(q3),

.qn(qn3)

);//搭建5个锁存器，并将他们按电路图搭建

assign out = q;

assign out1 = q1;

assign out2 = q2;

assign out3 = q3;

endmodule

module dff (input d,

input clk,

input rst,

output reg q,

output qn

);//实现D锁存器功能，q为q引脚，rst为reset引脚，clk为时钟信号引脚qn为q

非引脚

always @ (posedge clk or negedge rst)

if (!rst)

q = 0;

else

q = d;

assign qn = ~q;

endmodule

# 五、遇到的问题

在实验一中遇到一下问题：

1. D锁存器的接口不能悬空，否则会导致编译失败。在老师提醒下将相关引脚接对应电平。
2. 必须用系统的125MHz作为最初的时钟信号，否则会导致编译失败。在老师提醒下改正。

# 六、反思与心得

这次数电实验让我收获颇丰。我不仅深入理解了D，JK，T触发器的逻辑功能。还利用Multisim软件，使用D触发器实现了移位寄存器。同时了解阻塞赋值与非阻塞赋值的区别。学习了使用Verilog实现移位寄存器和计数器。我相信这些经验将有助于我未来的学习和工作。同时非常感想老师的指导，辛苦老师熬夜为我们整理实验器材。这也在提醒我们整理实验器材同样是实验过程的重要环节，我们应该认真对待。