Vivado工程3

时序逻辑电路的实现

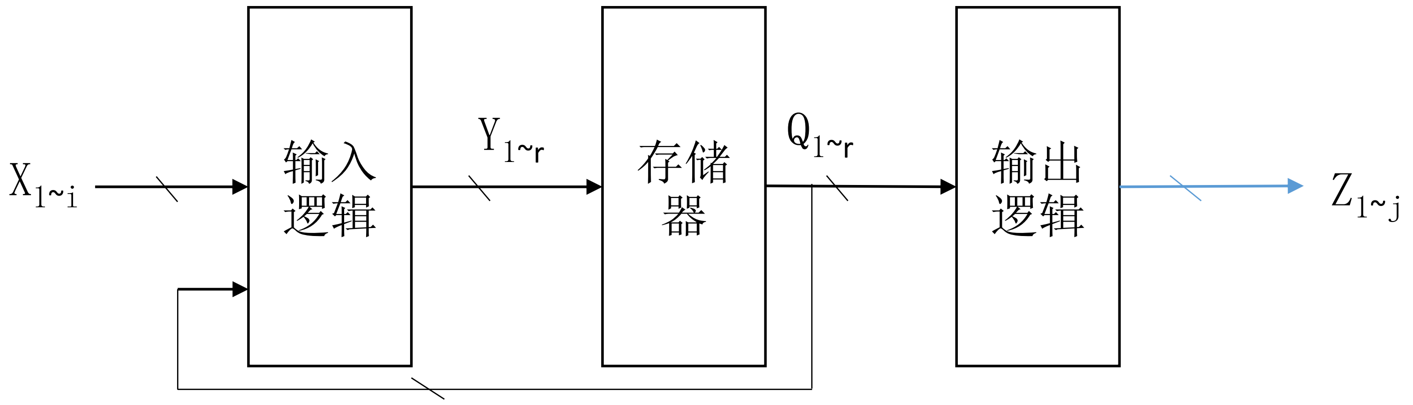
2019.11.15

FPGA芯片：xc7a35tcpg236-1

内容

时序逻辑电路

时序逻辑电路可以通过存储器或寄存器保存状态，并在每次时钟脉冲到来时改变存储元件的状态，称为同步时序电路，如果没有统一的时钟信号，则称为异步时序电路。



时序电路由组合电路和存储电路组成：

（1）输出方程　$Z = F_1(X, Q^n)$

（2）驱动方程　$Y = F_2(X, Q^n)$

（3）次态方程　$Q^{n+1} = F_3(Y, Q^n)$

驱动方程也称为激励方程

Q^n 和Q^{n+1} 分别称为现态和次态，可以简写为Q和Q*

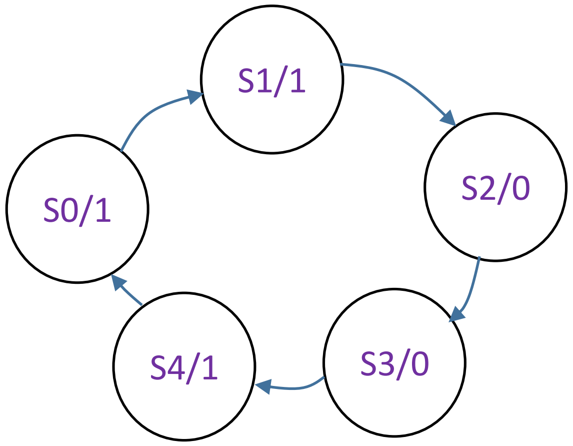
次态方程也称为转移方程

序列发生器

- 功能

  在输入时钟信号的作用下，均匀地输出序列11001。

- 状态图



五个状态需要3位二进制表示：

S0：000
S1：001
S2：010
S3：011
S4：100

- 状态转移表和卡诺图

Q2Q1Q0	Q2*Q1*Q0*	Z
000	001	1
001	010	1
010	011	0
011	100	0
100	000	1

Q2*Q1*Q0*/Z

Q2/Q1Q0	00	01	10	11
0	001/1	010/1	011/0	100/0
1	000/1	010/1	010/0	100/0

利用转移方程和输出方程填入的

- 转移方程和输出方程 (利用四个卡诺图得到的)

Q2* = Q1Q0
Q1* = Q1^Q0 (异或)
Q0* = Q0’Q2’

Z=Q1’

为了自启动，填入未用状态：
101=>010/1, 110=>010/0,111=>100/0

- 使用D触发器保存状态，则可以得到激励方程：

D2 = Q2* = Q1Q0
D1 = Q1* = Q1^Q0
D0 = Q0* = Q0’Q2’

xlfsq.v

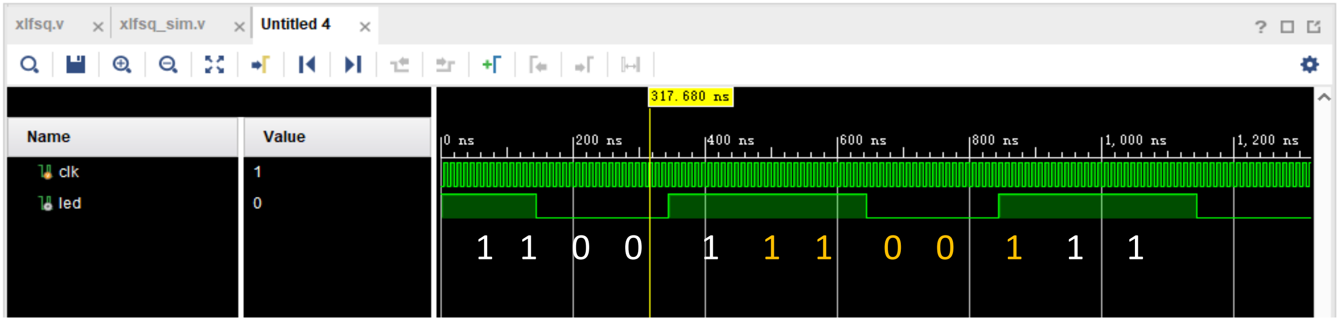
```
`timescale 1ns / 1ps
module xlfsq(
     input clk,
     output led
   );
   reg[31:0] divclk_cnt = 0;
   reg divclk=0;
   reg q0=0;
   reg q1=0;
   reg q2=0;
   always@(posedge clk)
   begin
      if(divclk_cnt==4)   // 4 * 5ns * 2 =  40ns
      begin
         divclk =~ divclk;
         divclk_cnt = 0;
      end
      else
      begin
          divclk_cnt = divclk_cnt+1'b1;
      end
   end;
   assign led=~q1;
   always@(posedge divclk)   // 40ns * 2 = 80ns
   begin
     q2<=q1&q0;
     q1<=q1^q0;
     q0<=~q0&~q2;
   end
endmodule
```

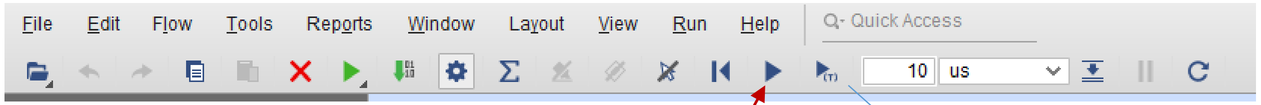
实现时把4改为25000000
25000000 * 10ns * 2 = 500ms

500ms * 2 = 1s

xlfsq_sim.v

```
`timescale 1ns / 1ps

module xlfsq_sim();
   reg clk;
   wire led;
   xlfsq uut(
        clk,led
   );
   initial begin
       clk=0;
   end
   always #5 clk=~clk;
endmodule
```



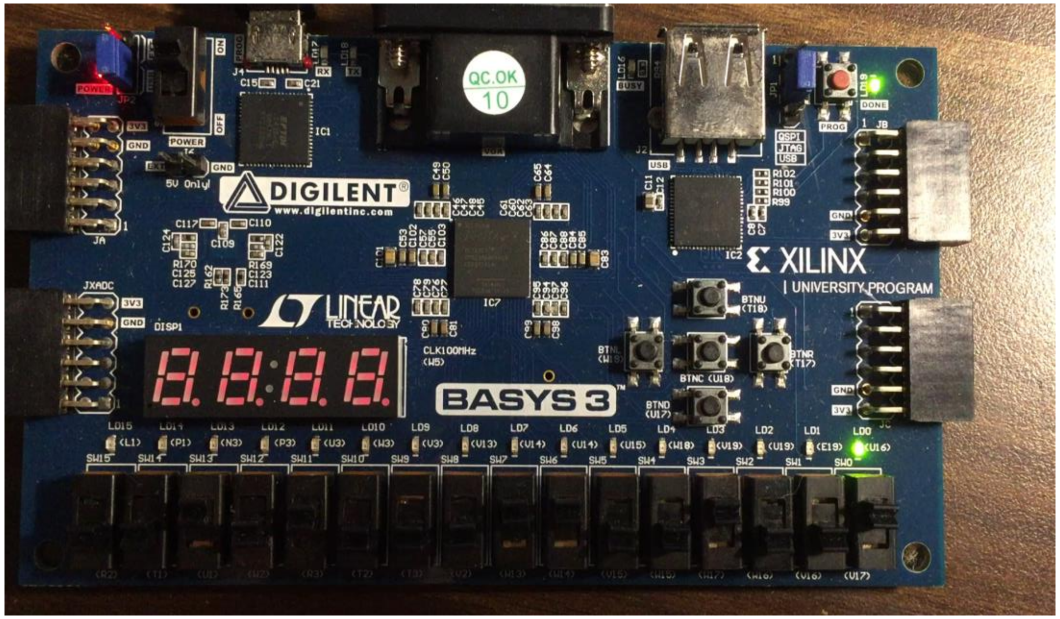


延长仿真时间，原来的默认值为1000ns　每次延长10us

W5为时钟信号引脚，这里定义时钟周期为10ns，0到5ns为低电平，剩余为高电平。

xlfsq.xdc

```
set_property PACKAGE_PIN U16 [get_ports led]
set_property IOSTANDARD LVCMOS33 [get_ports led]
set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} {get_ports clk}
```



FPGA引脚：

clk: W5

led0: U16 led1: E19 led2: U19 led3: V19 led4: W18
led5:U15 led6: U14 led7: V14 led8: V13 led9: V3
led10: W3 led11: U3 led12: P3 led13: N3 led14: P1
led15: L1

sw0: V17 sw1:V16 sw2: W16 sw3: W17 sw4: W15
sw5: V15 sw6: W14 sw7: W13 sw8: V2 sw9: T3
sw10: T2 sw11: R3 sw12: W2 sw13: U1 sw14: T1
sw15: R2

BTNC: U18 BTNL: W19 BTNR: T17 BTNU: T18 BTND: U17
（Center Left Right Up Down）

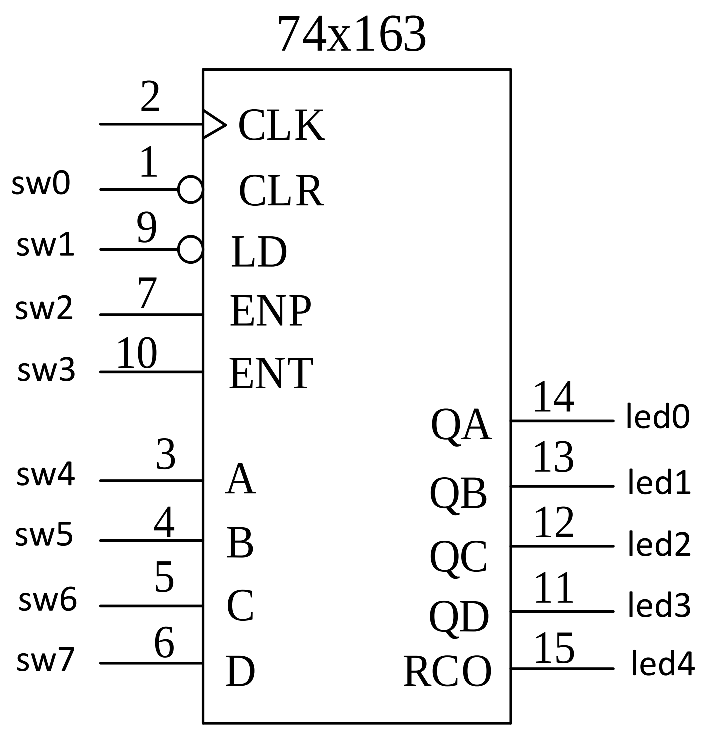
FPGA时钟工作原理

直接采用状态进行编程(xlfsq2.v)：

```
`timescale 1ns / 1ps

module xlfsq2(input clk,output led);
  reg led;
  reg[31:0] divclk_cnt = 0;
  reg divclk=0;
  reg [2:0] state=state_A;
  parameter
state_A=3'b000,state_B=3'b001,state_C=3'b010,state_D=3'b011,state
_E=3'b100;
  always@(posedge clk)
    begin
        if(divclk_cnt==25000000)
        begin
            divclk =~ divclk;
            divclk_cnt = 0;
        end
        else
        begin
             divclk_cnt = divclk_cnt+1'b1;
        end
    end
  always@(posedge divclk)
    begin
        case (state)
        state_A:
        begin
          state<=state_B;  led<=1;
        end
        state_B:
        begin
            state<=state_C;  led<=0;
        end
        state_C:
        begin
             state<=state_D;  led<=0;
        end
        state_D:
        begin
             state<=state_E;  led<=1;
        end
        state_E:
        begin
             state<=state_A; led<=1;
        end
        default:
        begin
             state<=state_A;  led<=1;
        end
        endcase
    end

endmodule
```

注意用下拉菜单把要实现的设计源码设置为Top

同步计数器



CLK	CLR_L	LD_L	ENP	ENT	工作状态
↑	0	×	×	×	同步清零
↑	1	0	×	×	同步置数
×	1	1	0	×	保持
×	1	1	×	0	保持，RCO=0
↑	1	1	1	1	计数

输出：QA~QD，始终为当前计数值
清零：输出QA~QD为0
置数：输出QA~QD等于输入A~D
计数：输出QA~QD不断加1，到1111后又
变为0000，即循环计数。
保持：保持计数值不变

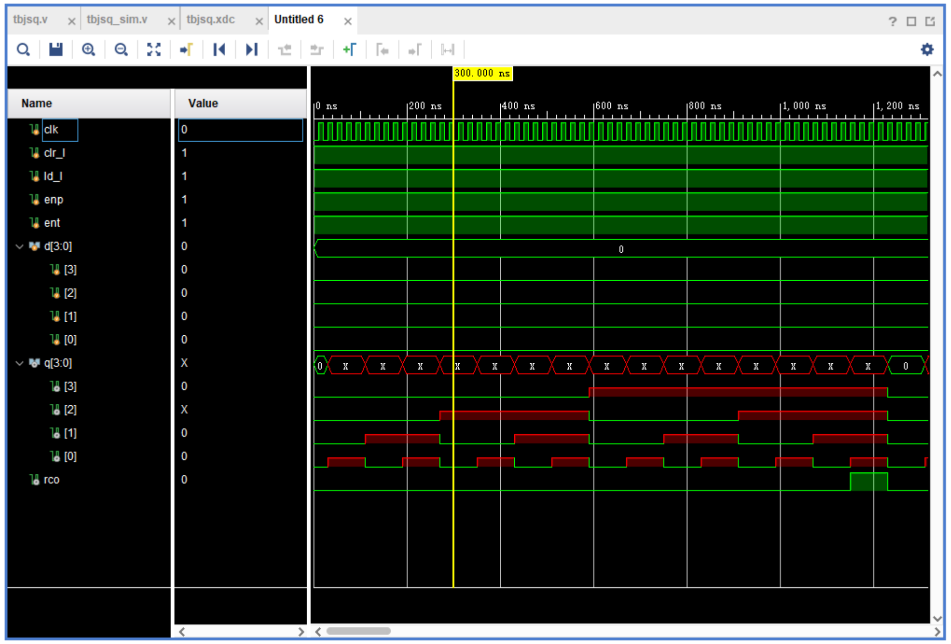
LD-Load:A~D=>QA~QD

tbjsq.v

```
`timescale 1ns / 1ps
module tbjsq(clk,clr_l,ld_l,enp,ent,d,q,rco);
input clk,clr_l,ld_l,enp,ent;
input[3:0] d;
output [3:0] q;
output rco;
reg [3:0] q=0;
reg rco=0;

reg[31:0] divclk_cnt = 0;
reg divclk=0;
always@(posedge clk)
   begin
      if(divclk_cnt==1)   //   1 or 25000000(imp)
      begin
         divclk =~ divclk;
         divclk_cnt = 0;
      end
      else
      begin
          divclk_cnt = divclk_cnt+1'b1;
      end
end;

always @ (posedge divclk) begin
                     if (clr_l==0)  q<=0;
                     else if (ld_l==0) q<=d;
  else if ((ent==1) && (ent==1)) q<=q+1;
  else q<=q;
end
always @ (q or ent) begin
                     if ((ent==1) && (q==15)) rco=1;
                     else rco=0;
end
endmodule
```

tbjsq_sim.v

```
`timescale 1ns / 1ps

module tbjsq_sim;
   reg clk=0;
                        reg clr_l=1;
                        reg ld_l=1;
                        reg enp=1;
                        reg ent=1;
                        reg[3:0] d=0;
                        wire[3:0] q=0;
   wire rco;
   tbjsq uut(clk,clr_l,ld_l,enp,ent,d,q,rco);
   always # 10 clk=~clk;

endmodule
```



Name	Value
clk	0
clr_l	1
ld_l	1
enp	1
ent	1
d[3:0]	0
[3]	0
[2]	0
[1]	0
[0]	0
q[3:0]	X
[3]	0
[2]	X
[1]	0
[0]	0
rco	0

```
set_property PACKAGE_PIN U16 [get_ports q[0]]
set_property IOSTANDARD LVCMOS33 [get_ports q[0]]
set_property PACKAGE_PIN E19 [get_ports q[1]]
set_property IOSTANDARD LVCMOS33 [get_ports q[1]]
set_property PACKAGE_PIN U19 [get_ports q[2]]
set_property IOSTANDARD LVCMOS33 [get_ports q[2]]
set_property PACKAGE_PIN V19 [get_ports q[3]]
set_property IOSTANDARD LVCMOS33 [get_ports q[3]]
set_property PACKAGE_PIN W18 [get_ports rco]
set_property IOSTANDARD LVCMOS33 [get_ports rco]

set_property PACKAGE_PIN V17 [get_ports clr_l]
set_property IOSTANDARD LVCMOS33 [get_ports clr_l]
set_property PACKAGE_PIN V16 [get_ports ld_l]
set_property IOSTANDARD LVCMOS33 [get_ports ld_l]
set_property PACKAGE_PIN W16 [get_ports enp]
set_property IOSTANDARD LVCMOS33 [get_ports enp]
set_property PACKAGE_PIN W17 [get_ports ent]
set_property IOSTANDARD LVCMOS33 [get_ports ent]

set_property PACKAGE_PIN W15 [get_ports d[0]]
set_property IOSTANDARD LVCMOS33 [get_ports d[0]]
set_property PACKAGE_PIN V15 [get_ports d[1]]
set_property IOSTANDARD LVCMOS33 [get_ports d[1]]
set_property PACKAGE_PIN W14 [get_ports d[2]]
set_property IOSTANDARD LVCMOS33 [get_ports d[2]]
set_property PACKAGE_PIN W13 [get_ports d[3]]
set_property IOSTANDARD LVCMOS33 [get_ports d[3]]

set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
{get_ports clk}
```

实现时要修改： if(divclk_cnt== 25000000)

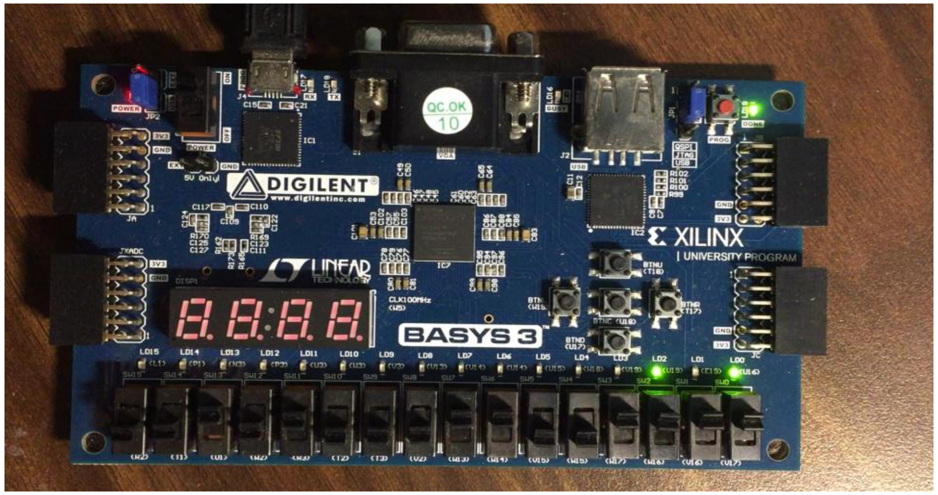
FPGA引脚：

clk: W5

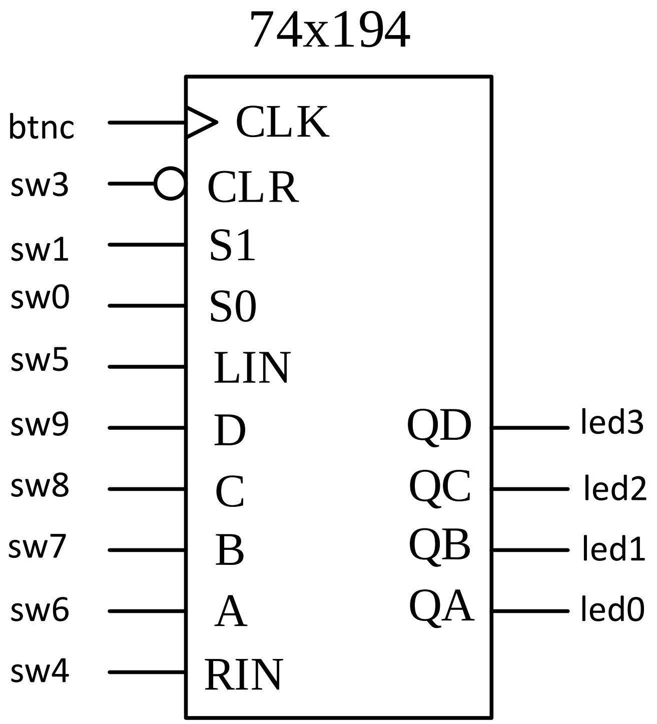
led0: U16 led1: E19 led2: U19 led3: V19 led4: W18
led5:U15 led6: U14 led7: V14 led8: V13 led9: V3
led10: W3 led11: U3 led12: P3 led13: N3 led14: P1
led15: L1

sw0: V17 sw1:V16 sw2: W16 sw3: W17 sw4: W15
sw5: V15 sw6: W14 sw7: W13 sw8: V2 sw9: T3
sw10: T2 sw11: R3 sw12: W2 sw13: U1 sw14: T1
sw15: R2

BTNC: U18 BTNL: W19 BTNR: T17 BTNU: T18 BTND: U17
（Center Left Right Up Down）



移位寄存器



功能	输入	下一状态
	S1 S0	QA* QB* QC* QD*
保持	0 0	QA QB QC QD
右移	0 1	RIN QA QB QC
左移	1 0	QB QC QD LIN
载入	1 1	A B C D

Ywjcq.v

```
`timescale 1ns / 1ps

module ywjcq(clk,clr_l,rin,lin,s,d,q );
input clk,clr_l,rin,lin;
input [1:0] s;
input [3:0] d;
output [3:0] q;
reg [3:0] q;
always @ (posedge clk or negedge clr_l)
            if (clr_l==0) q<=0;
           else case (s)
                     0:q<=q;              //保持
                     1:q<={rin,q[3:1]};  //右移
                     2:q<={q[2:0],lin};  //左移
                     3:q<=d;              //装载
                   default q<=4'bx ;
               endcase
endmodule
```

功能	输入	下一状态
	S1 S0	QA* QB* QC* QD*
保持	0 0	QA QB QC QD
右移	0 1	RIN QA QB QC
左移	1 0	QB QC QD LIN
载入	1 1	A B C D

d: ABCD
q: QAQBQCQD

```
set_property PACKAGE_PIN U16 [get_ports q[0]]
set_property IOSTANDARD LVCMOS33 [get_ports q[0]]
set_property PACKAGE_PIN E19 [get_ports q[1]]
set_property IOSTANDARD LVCMOS33 [get_ports q[1]]
set_property PACKAGE_PIN U19 [get_ports q[2]]
set_property IOSTANDARD LVCMOS33 [get_ports q[2]]
set_property PACKAGE_PIN V19 [get_ports q[3]]
set_property IOSTANDARD LVCMOS33 [get_ports q[3]]

set_property PACKAGE_PIN V17 [get_ports s[0]]
set_property IOSTANDARD LVCMOS33 [get_ports s[0]]
set_property PACKAGE_PIN V16 [get_ports s[1]]
set_property IOSTANDARD LVCMOS33 [get_ports s[1]]

set_property PACKAGE_PIN W17 [get_ports clr_l]
set_property IOSTANDARD LVCMOS33 [get_ports clr_l]
set_property PACKAGE_PIN W15 [get_ports rin]
set_property IOSTANDARD LVCMOS33 [get_ports rin]
set_property PACKAGE_PIN V15 [get_ports lin]
set_property IOSTANDARD LVCMOS33 [get_ports lin]


set_property PACKAGE_PIN W14 [get_ports d[0]]
set_property IOSTANDARD LVCMOS33 [get_ports d[0]]
set_property PACKAGE_PIN W13 [get_ports d[1]]
set_property IOSTANDARD LVCMOS33 [get_ports d[1]]
set_property PACKAGE_PIN V2 [get_ports d[2]]
set_property IOSTANDARD LVCMOS33 [get_ports d[2]]
set_property PACKAGE_PIN T3 [get_ports d[3]]
set_property IOSTANDARD LVCMOS33 [get_ports d[3]]

set_property PACKAGE_PIN U18 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets clk_IBUF]
```

FPGA引脚：

clk: W5

led0: U16	led1: E19	led2: U19	led3: V19	led4: W18
led5:U15	led6: U14	led7: V14	led8: V13	led9: V3
led10: W3	led11: U3	led12: P3	led13: N3	led14: P1
led15: L1				

sw0: V17	sw1:V16	sw2: W16	sw3: W17	sw4: W15
sw5: V15	sw6: W14	sw7: W13	sw8: V2	sw9: T3
sw10: T2	sw11: R3	sw12: W2	sw13: U1	sw14: T1
sw15: R2				

BTNC: U18   BTNL: W19   BTNR: T17   BTNU: T18   BTND: U17
（Center Left Right Up Down）

如果采用按钮或开关作为时钟引脚(需要消除抖动)，而不是用W5，则需要用这个语句进行说明。