# 中国科学技术大学计算机学院

# 《数字电路实验》报告


实验题目：计数器

学生姓名：柯志伟

学生学号：PB20061338

完成日期：2021 年 11 月 23 日

计算机实验教学中心制

2020 年 09 月

## 【实验题目】

计数器

## 【实验目的】

- 基础部分
  - 计数器及实现分频器
  - 数码管动态显示
  - 开关输入去抖动
- 选做部分
  - 数码管动态显示时去除先导零
  - 实现按钮按下后并保持时，计数器连续计数

## 【实验环境】

- PC 机
- Vivado 平台
- Nexys4 开发板

## 【实验过程】

- 用 Verilog 分别实现基本模块：计数器、去抖动、同步化、动态显示模块
  代码如下：
  - 计数器

```
module counter #(parameter W = 16,RST_VLU = 0)
       (input clk,rstn,pe,ce
       ,input [W-1:0] d,output reg [W-1:0] q);
          //pe 同步置数使能 ; ce 计数使能 ;
          //clk 时钟 ; rstn 复位
    always @(posedge clk) begin
      if(!rstn) q <= RST_VLU;       //同步复位
      else if(pe) q <= d;
      else if(ce) q <= q+1;         //16位
    end
endmodule
```

  - 去抖动

```
module debounce(input clk,rstn,
       input x,
       output reg y);
       //其实可以例化一个计数器模块
  parameter  Cnt0 = 3'b000, Cnt1 = 3'b001,
  Assign0 = 3'b010,Assign1=3'b011,Default = 3'b111;
  //刚开始进入默认状态，两个计数状态，两个赋值状态
  reg [2:0] CS,NS;
  reg [19:0] Count;
  reg y_reg;
  always @(posedge clk,negedge rstn) begin //异步复位
       if(!rstn)begin
         CS <= Default;
         y_reg <= 0;
       end
       else begin
         CS <= NS;
         y_reg <= y;
       end
    end

always @(posedge clk,negedge rstn) begin
  if(!rstn) Count <=0;
  else begin
  case(CS)
    Default: Count <= 0;
    Cnt0 :begin
    Count <= Count+1;
    end
    Cnt1: Count <= Count+1;
    Assign0: Count <= 0;
    Assign1: Count <= 0;
  endcase
end


end

always @(*) begin
  y = y_reg;
  NS = Default;
  case (CS)
```

```
43      Default: begin
44        y = 0;
45        if(x == 1) NS = Cnt1;
46        else NS = Cnt0;
47        end
48      Cnt0:begin
49        if(Count == 100000) begin
50        NS = Assign0;
51        end
52        else if(x == 0)begin
53        NS = Cnt0;
54        end
55        else begin
56        NS = Default;
57        end
58        end
59      Cnt1:begin
60        if(Count == 100000) begin
61        NS = Assign1;
62        end
63        else if(x == 1)begin
64        NS = Cnt1;
65        end
66        else begin
67        NS = Default;
68        end
69      end
70      Assign0: begin
71        y = 0;
72        if(x == 0)
73        NS = Assign0;
74        else
75        NS = Default;    //只有极个别才在此刻为1
76        end
77      Assign1:begin
78        y = 1;
79        if(x == 1) NS = Assign1;
80        else NS = Default;     //只有极个别才在此刻为1
81      end
82    endcase
83  end
84  endmodule
```

– 同步化

```
module Synchron(input x,input clk,rstn,
      output y);

  reg s1,s2;
  reg s;

  always @(posedge clk) begin
    if(~rstn ) begin
    s1 <= 0;
    s2 <= 0;
    s <= 0;
    end
    else begin
    s1 <= x;
    s2 <= s1;
    s <= s2;
    end
  end
  assign y = (!s)&s2;
endmodule
```

– 数码管动态显示

```
      module dynimage(input [3:0] d0,d1,d2,d3,
      input clk,rstn,
      output reg [3:0] an,
      output reg [6:0] seg);

      wire clkd;          //分频时钟
      reg [3:0] DIn;
      wire [6:0] DOut;

      frequdivision  FreDivClk(clk,~rstn,clkd);

      Decoder7Seg decoder(DIn,DOut);

  parameter s0 = 2'b00, s1 = 2'b01,s2 = 2'b10,s3 = 2'b11;
  reg [1:0] CS,NS;


  always @(posedge clkd,negedge rstn) begin  //异步初始化
    if(~rstn) CS <= s0;
```

```
20      else CS <= NS;
21    end
22
23    always @(*) begin
24      case(CS)
25        s0 : begin
26          DIn = d0;
27          an = 4'b1110;
28          seg = DOut;
29          NS = s1;
30        end
31        s1 : begin
32          DIn = d1;
33          seg = DOut;
34          NS = s2;
35          if((d1==0)&&(d3==0)&&(d2==0)) begin
36          an = 4'b1111;
37          end
38          else an = 4'b1101;
39        end
40        s2 : begin
41          DIn = d2;
42          seg = DOut;
43          NS = s3;
44          if((d3==0)&&(d2==0)) begin
45          an = 4'b1111;
46          end
47          else  an = 4'b1011;
48        end
49        s3 : begin
50          DIn = d3;
51          seg = DOut;
52          NS = s0;
53          if(d3==0) begin
54          an = 4'b1111;
55          end
56          else  an = 4'b0111;
57        end
58      endcase
59    end
60
61
```

```
62    endmodule
63
64
65
66
67
68  module frequdivision #(parameter N = 200000,
69                RST_VLU = 0)(input clk,rst,
70                output reg out);
71    //分频器 N = 100000 ~ 2000000
72    reg [19:0] cnt ;
73
74    always @(posedge clk) begin
75      if(rst)  cnt <= RST_VLU;
76      else if(cnt == (N-1)) cnt <= 0;
77      else cnt <= cnt + 1;
78    end
79
80    always @(posedge clk) begin
81      if(rst) out <= 0;
82      else if(cnt == (N-2)) out <= 1;
83      else out <= 0;
84    end
85  endmodule
86
87  module Decoder7Seg(                //7段译码管
88    input wire [3:0] In,
89    output reg [6:0] Out
90    );
91    always @ (*)
92    begin
93    case(In)
94      4'b0000: Out = 7'b000_0001;
95      4'b0001: Out = 7'b100_1111;
96      4'b0010: Out = 7'b001_0010;
97      4'b0011: Out = 7'b000_0110;
98      4'b0100: Out = 7'b100_1100;
99      4'b0101: Out = 7'b010_0100;
100     4'b0110: Out = 7'b010_0000;
101     4'b0111: Out = 7'b000_1111;
102     4'b1000: Out = 7'b000_0000;
103     4'b1001: Out = 7'b000_0100;
```

```
    4'b1010: Out = 7'b000_1000;  //A
    4'b1011: Out = 7'b110_0000;  //b
    4'b1100: Out = 7'b011_0001;  //C
    4'b1101: Out = 7'b100_0010;  //d
    4'b1110: Out = 7'b011_0000;  //E
    4'b1111: Out = 7'b011_1000;  //F
  endcase
  end
endmodule
```

- 完成数码管动态扫描的仿真、下载、测试
  - 代码如下:

```
    module dynimage(input [3:0] d0,d1,d2,d3,
    input clk,rstn,
    output reg [3:0] an,
    output reg [6:0] seg);

    wire clkd;          //分频时钟
    reg [3:0] DIn;
    wire [6:0] DOut;

    frequdivision  FreDivClk(clk,~rstn,clkd);

    Decoder7Seg decoder(DIn,DOut);

  parameter s0 = 2'b00, s1 = 2'b01,s2 = 2'b10,s3 = 2'b11;
  reg [1:0] CS,NS;


  always @(posedge clkd,negedge rstn) begin  //异步初始化
  if(~rstn) CS <= s0;
  else CS <= NS;
  end

  always @(*) begin
  case(CS)
    s0 : begin
      DIn = d0;
      an = 4'b1110;
      seg = DOut;
      NS = s1;
    end
```

```
31      s1 : begin
32        DIn = d1;
33        seg = DOut;
34        NS = s2;
35        if((d1==0)&&(d3==0)&&(d2==0)) begin
36        an = 4'b1111;
37        end
38        else an = 4'b1101;
39      end
40      s2 : begin
41        DIn = d2;
42        seg = DOut;
43        NS = s3;
44        if((d3==0)&&(d2==0)) begin
45        an = 4'b1111;
46        end
47        else  an = 4'b1011;
48      end
49      s3 : begin
50        DIn = d3;
51        seg = DOut;
52        NS = s0;
53        if(d3==0) begin
54        an = 4'b1111;
55        end
56        else  an = 4'b0111;
57      end
58    endcase
59    end
60
61
62    endmodule
63
64
65    module frequdivision #(parameter N = 200000,
66              RST_VLU = 0)(input clk,rst,
67              output reg out);
68    //分频器 N = 100000 ~ 2000000
69    reg [19:0] cnt ;
70
71      always @(posedge clk) begin
72        if(rst)  cnt <= RST_VLU;
```

```
73        else if(cnt == (N-1)) cnt <= 0;
74        else cnt <= cnt + 1;
75      end
76
77      always @(posedge clk) begin
78        if(rst) out <= 0;
79        else if(cnt == (N-2)) out <= 1;
80        else out <= 0;
81        end
82      endmodule
83
84    module Decoder7Seg(                //7段译码管
85    input wire [3:0] In,
86    output reg [6:0] Out
87    );
88      always @ (*)
89      begin
90      case(In)
91        4'b0000: Out = 7'b000_0001;
92        4'b0001: Out = 7'b100_1111;
93        4'b0010: Out = 7'b001_0010;
94        4'b0011: Out = 7'b000_0110;
95        4'b0100: Out = 7'b100_1100;
96        4'b0101: Out = 7'b010_0100;
97        4'b0110: Out = 7'b010_0000;
98        4'b0111: Out = 7'b000_1111;
99        4'b1000: Out = 7'b000_0000;
100       4'b1001: Out = 7'b000_0100;
101       4'b1010: Out = 7'b000_1000;   //A
102       4'b1011: Out = 7'b110_0000;   //b
103       4'b1100: Out = 7'b011_0001;   //C
104       4'b1101: Out = 7'b100_0010;   //d
105       4'b1110: Out = 7'b011_0000;   //E
106       4'b1111: Out = 7'b011_1000;   //F
107     endcase
108     end
109   endmodule
```

– 仿真与下载测试

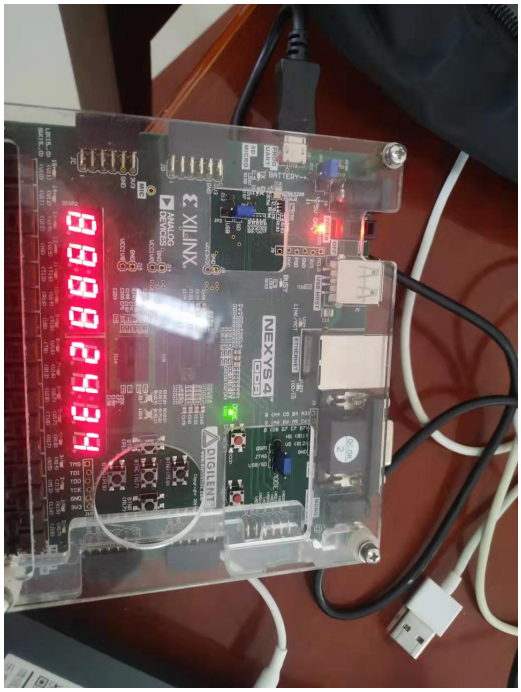

图 1: 数码管动态显示下载测试

– 查看 RTL 分析和综合后电路图

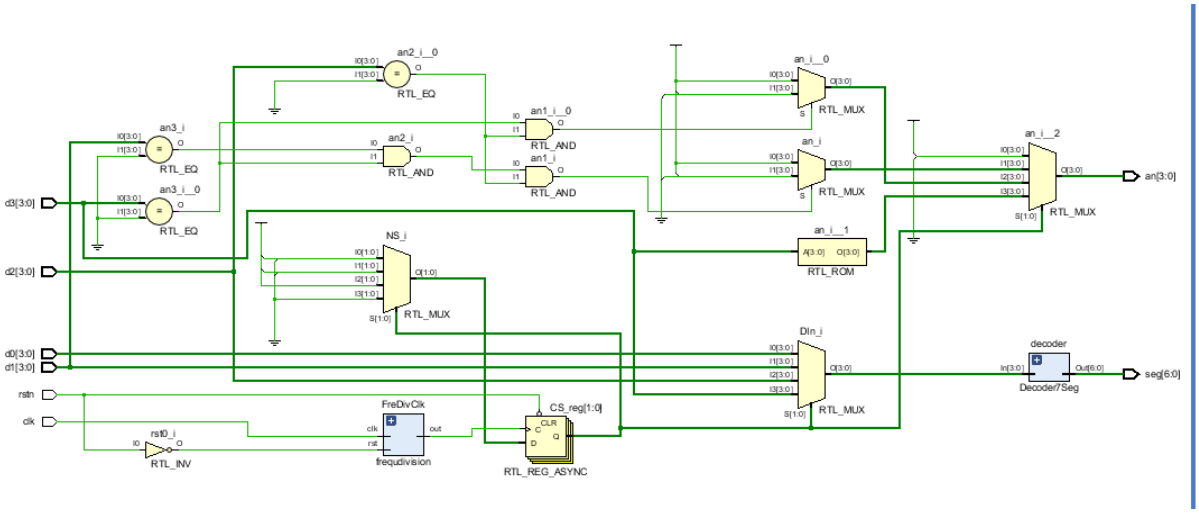


图 2: 数码管动态显示 RTL 分析后电路图

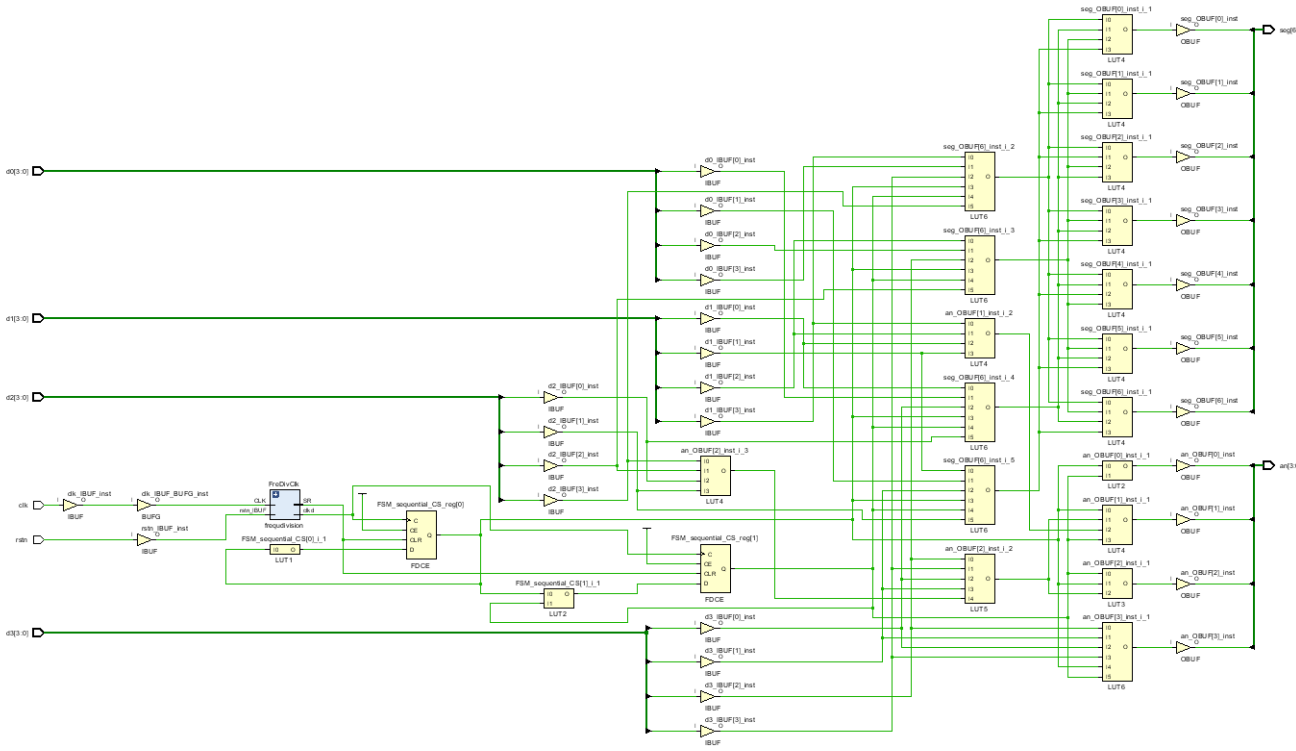

图 3: 数码管动态显示综合后电路图

– 查看综合后电路资源使用情况

Hierarchy

| Name | Slice LUTs (63400) | Slice Registers (126800) | Bonded IOB (210) | BUFGCTRL (32) |
|---|---|---|---|---|
| dynimage | 41 | 23 | 29 | 1 |
| FreDivClk (frequdivision) | 26 | 21 | 0 | 0 |

图 4: 数码管动态显示综合后电路资源使用情况

- 完成开关输入抖动验证的仿真、下载、测试

– 代码如下:

```
1   module ShowBounce(input clk1,clk2,rstn,
2   output [3:0] an,
3   output [6:0] seg);
4   wire [15:0] Out;
5   wire  [3:0] Out0,Out1,Out2,Out3;
6   counter cnt(clk1,rstn,0,1,0,Out);
7   assign Out0 = {Out[3],Out[2],Out[1],Out[0]};
8   assign Out1 = {Out[7],Out[6],Out[5],Out[4]};
9   assign Out2 = {Out[11],Out[10],Out[9],Out[8]};
10  assign Out3 = {Out[15],Out[14],Out[13],Out[12]};
```

```
11      dynimage DynImage(Out0,Out1,Out2,Out3,
12      clk2,rstn,an,seg);
13
14      endmodule
15
16
17      module counter #(parameter W = 16,RST_VLU = 0)
18      (input clk,rstn,pe,ce
19      ,input [W-1:0] d,output reg [W-1:0] q);
20       //pe 同步置数使能 ; ce 计数使能 ; clk 时钟 ; rstn 复位
21      always @(posedge clk,negedge rstn) begin
22          if(!rstn) q <= RST_VLU;       //同步复位
23          else if(pe) q <= d;
24          else if(ce) q <= q+1;         //16位
25      end
26      endmodule
27
28
29      module dynimage(input [3:0] d0,d1,d2,d3,
30                      input clk,rstn,
31                      output reg [3:0] an,
32                      output reg [6:0] seg);
33
34                      wire clkd;          //分频时钟
35                      reg [3:0] DIn;
36                      wire [6:0] DOut;
37              frequdivision  FreDivClk(clk,~rstn,clkd);
38              Decoder7Seg decoder(DIn,DOut);
39
40      parameter s0 = 2'b00, s1 = 2'b01,s2 = 2'b10,s3 = 2'b11;
41      reg [1:0] CS,NS;
42
43
44      always @(posedge clkd,negedge rstn) begin  //异步初始化
45          if(~rstn) CS <= s0;
46          else CS <= NS;
47      end
48
49      always @(*) begin
50            case(CS)
51            s0 : begin
52                 DIn = d0;
```

```
53                an = 4'b1110;
54                seg = DOut;
55                NS = s1;
56            end
57            s1 : begin
58              DIn = d1;
59              seg = DOut;
60              NS = s2;
61              if((d1==0)&&(d3==0)&&(d2==0)) begin
62                an = 4'b1111;
63              end
64              else an = 4'b1101;
65            end
66            s2 : begin
67              DIn = d2;
68              seg = DOut;
69              NS = s3;
70              if((d3==0)&&(d2==0)) begin
71                an = 4'b1111;
72              end
73              else  an = 4'b1011;
74
75            end
76            s3 : begin
77              DIn = d3;
78              seg = DOut;
79              NS = s0;
80               if(d3==0) begin
81                 an = 4'b1111;
82               end
83               else  an = 4'b0111;
84            end
85            endcase
86      end
87
88
89      endmodule
90
91      module frequdivision #(parameter N = 200000,
92      RST_VLU = 0)
93      (input clk,rst, output reg out);
94       //分频器 N = 100000 ~ 2000000
```

```
95      reg [19:0] cnt ;
96      always @(posedge clk) begin
97          if(rst)  cnt <= RST_VLU;
98          else if(cnt == (N-1)) cnt <= 0;
99          else cnt <= cnt + 1;
100     end
101     always @(posedge clk) begin
102          if(rst) out <= 0;
103          else if(cnt == (N-2)) out <= 1;
104          else out <= 0;
105     end
106     endmodule
107
108     module Decoder7Seg(                //7段译码管
109     input wire [3:0] In,
110     output reg [6:0] Out
111          );
112     always @ (*)
113          begin
114          case(In)
115               4'b0000: Out = 7'b000_0001;
116               4'b0001: Out = 7'b100_1111;
117               4'b0010: Out = 7'b001_0010;
118               4'b0011: Out = 7'b000_0110;
119               4'b0100: Out = 7'b100_1100;
120               4'b0101: Out = 7'b010_0100;
121               4'b0110: Out = 7'b010_0000;
122               4'b0111: Out = 7'b000_1111;
123               4'b1000: Out = 7'b000_0000;
124               4'b1001: Out = 7'b000_0100;
125               4'b1010: Out = 7'b000_1000;   //A
126               4'b1011: Out = 7'b110_0000;   //b
127               4'b1100: Out = 7'b011_0001;   //C
128               4'b1101: Out = 7'b100_0010;   //d
129               4'b1110: Out = 7'b011_0000;   //E
130               4'b1111: Out = 7'b011_1000;   //F
131          endcase
132          end
133     endmodule
```

– 仿真与下载测试

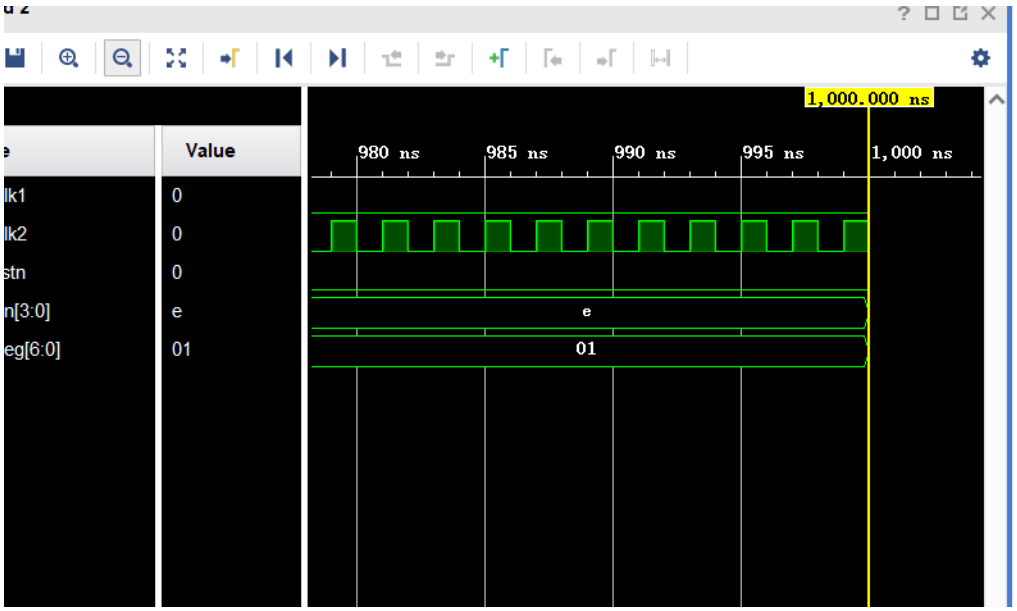


图 5: 开关输入抖动验证仿真

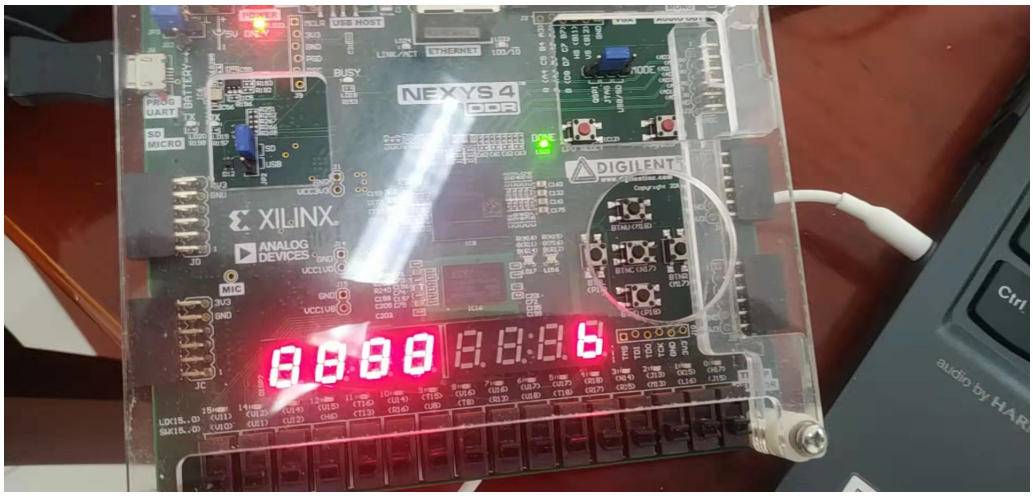


图 6: 开关输入抖动验证下载测试

– 查看 RTL 分析和综合后电路图

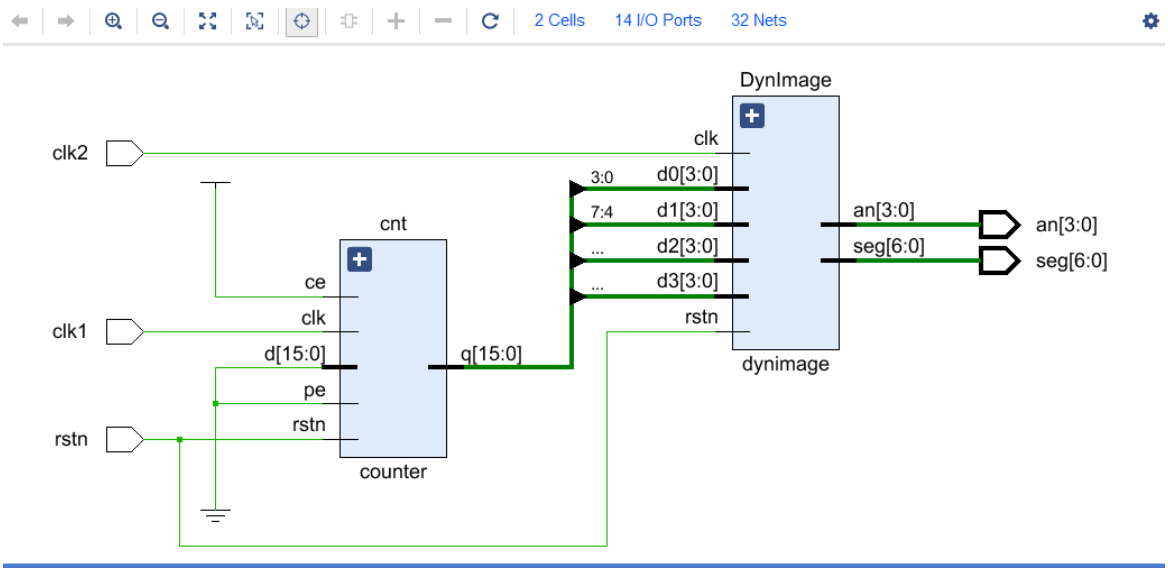


图 7: 开关输入抖动验证 RTL 分析后电路图

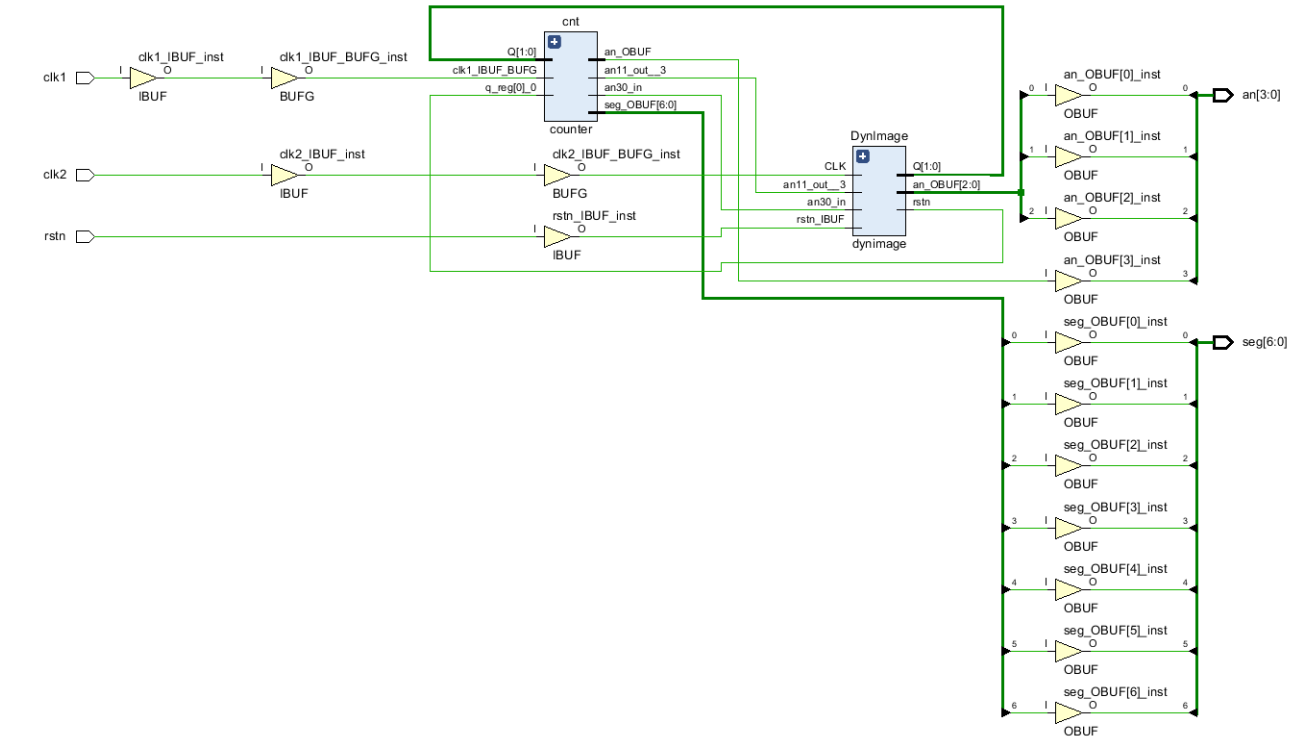


图 8: 开关输入抖动验证综合后电路图

– 查看综合后电路资源使用情况

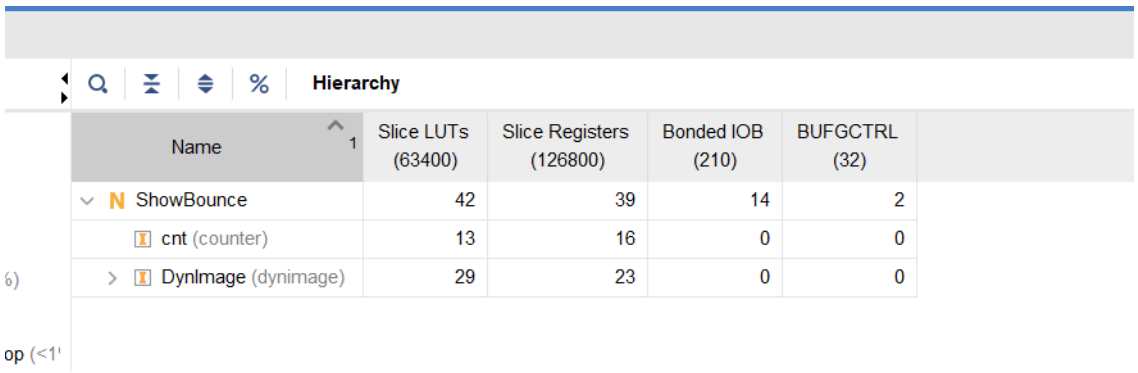


图 9: 开关输入抖动验证综合后电路资源使用情况

- 完成开关输入去抖动后动态显示的仿真、下载、测试 (含有去除先导零与连续输入功能)
  - 代码如下:

```
module dedynimage(input x,rstn,
input clk,
output [3:0] an,
output [6:0] seg);
wire db_x;
wire Syn_x;
wire [15:0] Cnt_Out;
debounceX DB(clk,rstn,x,db_x);
Synchron PS(clk,rstn,db_x,Syn_x);
counter cnt(clk,rstn,0,Syn_x,0,Cnt_Out);
dynimage DIS(clk,rstn,Cnt_Out,an,seg);

endmodule


module debounce(input clk,rstn,
input x,
output reg y); //其实可以例化一个计数器模块
  parameter  Cnt0 = 3'b000, Cnt1 = 3'b001,
  Assign0 = 3'b010,Assign1=3'b011,
  Default = 3'b111;
  //刚开始进入默认状态，两个计数状态，两个赋值状态
  reg [2:0] CS,NS;
  reg [19:0] Count;
  reg y_reg;
  always @(posedge clk,negedge rstn) begin
          //异步复位
    if(!rstn)begin
      CS <= Default;
```

```
30          y_reg <= 0;
31        end
32        else begin
33          CS <= NS;
34          y_reg <= y;
35        end
36      end
37
38      always @(posedge clk,negedge rstn) begin
39        if(!rstn) Count <=0;
40        else begin
41        case(CS)
42          Default: Count <= 0;
43          Cnt0 :begin
44          Count <= Count+1;
45          end
46          Cnt1: Count <= Count+1;
47          Assign0: Count <= 0;
48          Assign1: Count <= 0;
49        endcase
50        end
51      end
52
53      always @(*) begin
54        y = y_reg;
55        NS = Default;
56        case (CS)
57          Default: begin
58            y = 0;
59            if(x == 1) NS = Cnt1;
60            else NS = Cnt0;
61          end
62          Cnt0:begin
63            if(Count == 100000) begin
64              NS = Assign0;
65            end
66            else if(x == 0)begin
67                NS = Cnt0;
68              end
69            else begin
70                NS = Default;
71            end
```

```
72          end
73          Cnt1:begin
74            if(Count == 100000) begin
75              NS = Assign1;
76            end
77            else if(x == 1)begin
78              NS = Cnt1;
79            end
80            else begin
81              NS = Default;
82            end
83          end
84          Assign0: begin
85            y = 0;
86            if(x == 0)
87              NS = Assign0;
88            else
89              NS = Default;  //只有极个别才在此刻为1
90          end
91          Assign1:begin
92            y = 1;
93            if(x == 1) NS = Assign1;
94            else NS = Default; //只有极个别才在此刻为1
95          end
96        endcase
97    end
98  endmodule
99
100 module debounceX(input clk,rstn,
101 input x,output reg y);
102 //其实可以例化一个计数器模块
103   parameter  Cnt0 = 3'b000, Cnt1 = 3'b001,
104   Assign0 = 3'b010,Assign1=3'b011,
105   Cnt1x=4'b0100,Assign0x=3'b110, Default = 3'b111;
106    //刚开始进入默认状态，两个计数状态，两个赋值状态
107   reg [2:0] CS,NS;
108   reg [31:0] Count;
109   reg y_reg;
110   always @(posedge clk,negedge rstn) begin //异步复位
111     if(!rstn)begin
112       CS <= Default;
113       y_reg <= 0;
```

```
114         end
115         else begin
116           CS <= NS;
117           y_reg <= y;
118         end
119       end
120
121       always @(posedge clk,negedge rstn) begin
122       if(!rstn) Count <=0;
123       else begin
124       case(CS)
125         Default: Count <= 0;
126         Cnt0 :begin
127         Count <= Count+1;
128         end
129         Cnt1: Count <= Count+1;
130         Assign0: Count <= 0;
131         Assign1: Count <= 0;
132         Cnt1x: Count <=Count+1;
133         Assign0x: Count <= 0;
134       endcase
135       end
136       end
137
138       always @(*) begin
139         y = y_reg;
140         NS = Default;
141         case (CS)
142           Default: begin
143             y = 0;
144             if(x == 1) NS = Cnt1;
145             else NS = Cnt0;
146           end
147           Cnt0:begin
148             if(Count == 1000000) begin
149             NS = Assign0;
150             end
151             else if(x == 0)begin
152             NS = Cnt0;
153             end
154             else begin
155             NS = Default;
```

```
156             end
157           end
158           Cnt1:begin
159             if(Count == 1000000) begin              //去抖动
160             NS = Assign1;
161             end
162             else if(x == 1)begin
163             NS = Cnt1;
164             end
165             else begin
166             NS = Default;
167             end
168           end
169           Assign0: begin
170             y = 0;
171             if(x == 0)
172             NS = Assign0;
173             else
174             NS = Default;   //只有极个别才在此刻为1
175           end
176           Assign1:begin
177             y = 1;
178             if(x == 1) NS = Cnt1x;
179             else NS = Default;   //只有极个别才在此刻为1
180           end
181           Cnt1x:begin
182             y = 1;
183             if(Count == 2000000000 )
184             NS = Assign0x;
185             else if(x==0)
186             NS = Cnt1x;
187             else NS = Default;
188           end
189           Assign0x:begin //将连续的1信号划分为多个01上升沿
190             y = 0;
191             if(Count == 2000000000) NS = Assign1;
192             else if(x == 1) NS = Assign0x;
193             else NS = Default;
194           end
195         endcase
196       end
197     endmodule
```

```
198
199
200 module Synchron(input clk,rstn,input x,
201 output y);
202   reg s1,s2;
203   reg s;
204
205   always @(posedge clk) begin
206     if(~rstn ) begin
207       s1 <= 0;
208       s2 <= 0;
209       s <= 0;
210     end
211     else begin
212       s1 <= x;
213       s2 <= s1;
214       s <= s2;
215     end
216   end
217   assign y = (!s)&s2;
218 endmodule
219
220
221 module counter #(parameter W = 16,RST_VLU = 0)
222 (input clk,rstn,pe,ce,input [W-1:0] d,
223 output reg [W-1:0] q);
224 //pe 同步置数使能 ; ce 计数使能 ; clk 时钟 ; rstn 复位
225   always @(posedge clk) begin
226     if(!rstn) q <= RST_VLU;      //同步复位
227     else if(pe) q <= d;
228     else if(ce) q <= q+1;        //16位
229   end
230 endmodule
231
232
233
234 module dynimage(
235 input clk,rstn,
236 input [15:0] d,
237 output reg [3:0] an,
238 output reg [6:0] seg);
239
```

```
240     wire clkd;          //分频时钟
241     reg [3:0] DIn;
242     wire [6:0] DOut;
243
244     frequdivision  FreDivClk(clk,~rstn,clkd);
245     Decoder7Seg decoder(DIn,DOut);
246
247     parameter s0 = 2'b00, s1 = 2'b01,s2 = 2'b10,s3 = 2'b11;
248     reg [1:0] CS,NS;
249
250
251     always @(posedge clkd,negedge rstn) begin  //异步初始化
252       if(~rstn) CS <= s0;
253       else CS <= NS;
254     end
255
256     always @(*) begin
257       case(CS)
258         s0 : begin
259           DIn = d[3:0];
260           an = 4'b1110;
261           seg = DOut;
262           NS = s1;
263         end
264         s1 : begin
265           DIn = d[7:4];
266           seg = DOut;
267           NS = s2;
268           if(( d[15:12]==0)&&(d[11:8]==0)&&
269           ( d[7:4]==0)) begin
270           an = 4'b1111;
271           end
272           else an = 4'b1101;
273         end
274         s2 : begin
275           DIn = d[11:8];
276           seg = DOut;
277           NS = s3;
278           if((d[15:12]==0)&&( d[11:8]==0)) begin
279           an = 4'b1111;
280           end
281           else  an = 4'b1011;
```

```
282
283         end
284         s3 : begin
285           DIn = d[15:12];
286           seg = DOut;
287           NS = s0;
288           if(d[15:12]==0) begin
289           an = 4'b1111;
290           end
291           else  an = 4'b0111;
292           end
293       endcase
294     end
295
296
297   endmodule
298
299
300   module frequdivision #(parameter N = 200000,RST_VLU = 0)
301   (input clk,rst, output reg out);
302   //分频器 N = 100000 ~ 2000000
303     reg [19:0] cnt ;
304     always @(posedge clk) begin
305       if(rst)  cnt <= RST_VLU;
306       else if(cnt == (N-1)) cnt <= 0;
307       else cnt <= cnt + 1;
308     end
309     always @(posedge clk) begin
310       if(rst) out <= 0;
311       else if(cnt == (N-2)) out <= 1;
312       else out <= 0;
313       end
314   endmodule
315
316   module Decoder7Seg(                //7段译码管
317     input wire [3:0] In,
318     output reg [6:0] Out
319     );
320     always @ (*)
321     begin
322       case(In)
323         4'b0000: Out = 7'b000_0001;
```

```
            4'b0001: Out = 7'b100_1111;
            4'b0010: Out = 7'b001_0010;
            4'b0011: Out = 7'b000_0110;
            4'b0100: Out = 7'b100_1100;
            4'b0101: Out = 7'b010_0100;
            4'b0110: Out = 7'b010_0000;
            4'b0111: Out = 7'b000_1111;
            4'b1000: Out = 7'b000_0000;
            4'b1001: Out = 7'b000_0100;
            4'b1010: Out = 7'b000_1000;   //A
            4'b1011: Out = 7'b110_0000;   //b
            4'b1100: Out = 7'b011_0001;   //C
            4'b1101: Out = 7'b100_0010;   //d
            4'b1110: Out = 7'b011_0000;   //E
            4'b1111: Out = 7'b011_1000;   //F
        endcase
    end
endmodule
```

– 仿真与下载测试

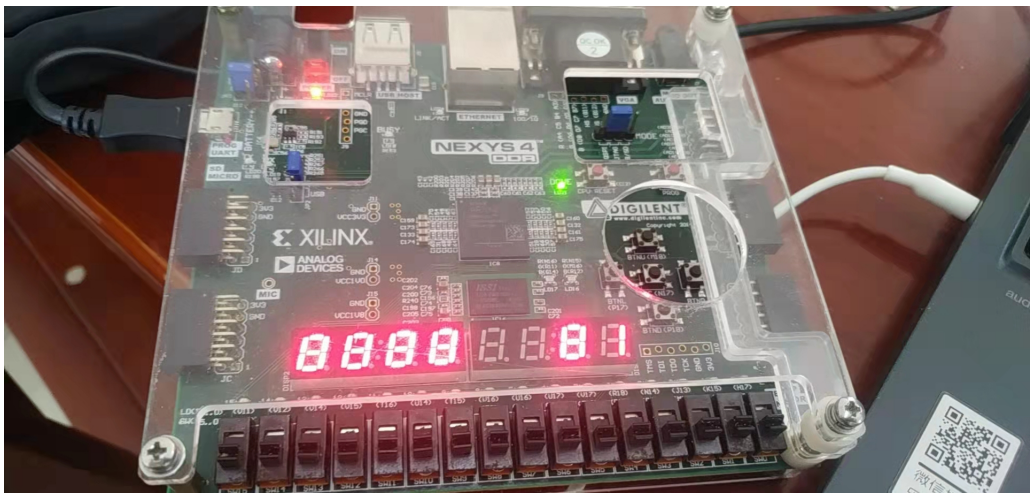


图 10: 开关输入去抖动后动态显示下载测试

– 查看 RTL 分析和综合后电路图

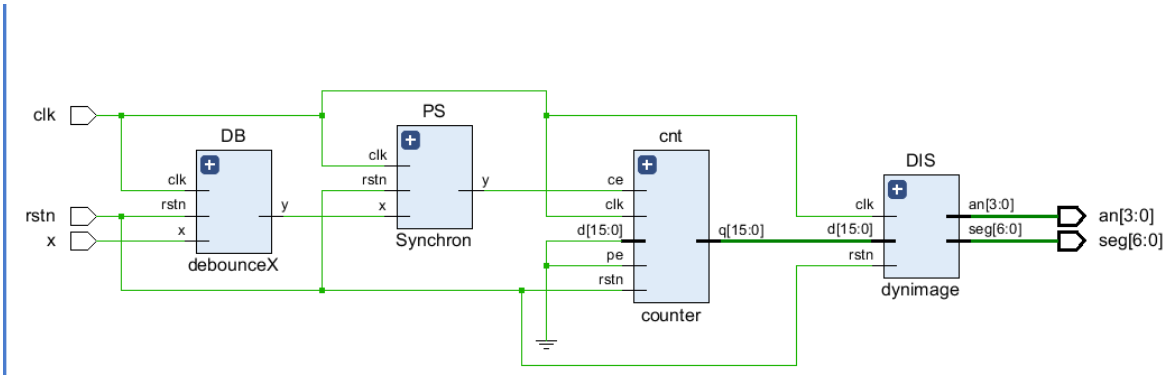


图 11: 开关输入去抖动后动态显示 RTL 分析后电路图

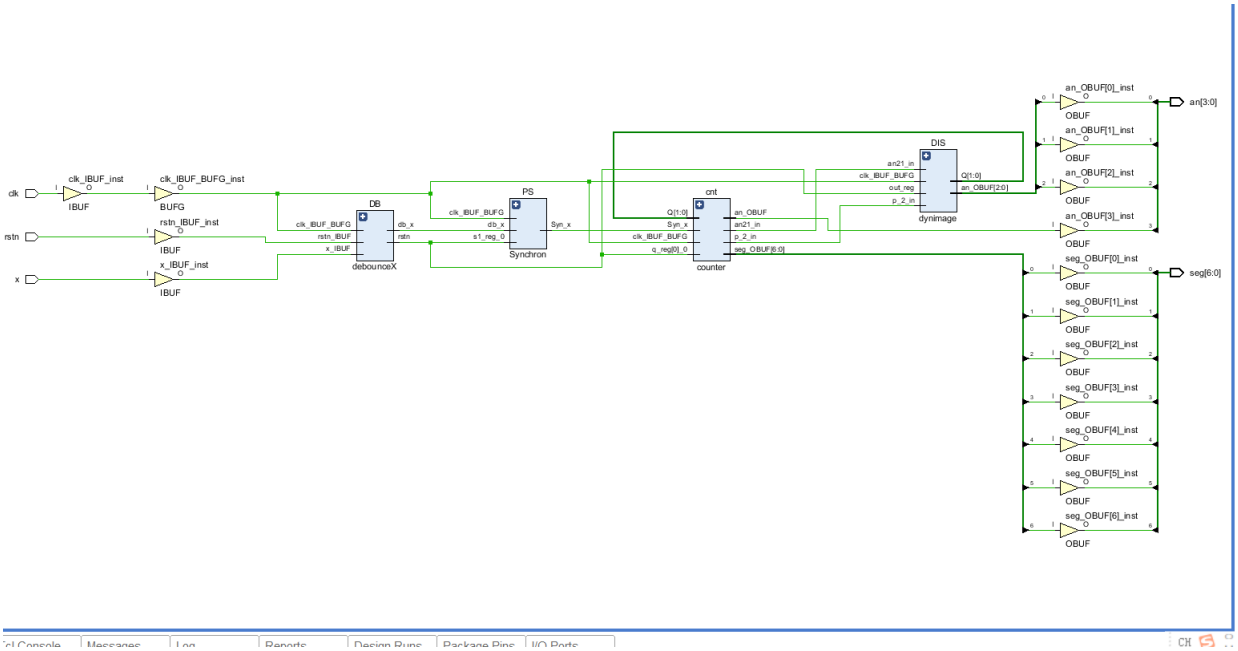

图 12: 开关输入去抖动后动态显示综合后电路图

– 查看综合后电路资源使用情况

Hierarchy

| Name | Slice LUTs (63400) | Slice Registers (126800) | Bonded IOB (210) | BUFGCTRL (32) |
|---|---|---|---|---|
| dedynimage | 68 | 78 | 14 | 1 |
| cnt (counter) | 13 | 16 | 0 | 0 |
| DB (debounceX) | 34 | 36 | 0 | 0 |
| DIS (dynimage) | 20 | 23 | 0 | 0 |
| PS (Synchron) | 1 | 3 | 0 | 0 |

图 13: 开关输入去抖动后动态显示综合后电路资源使用情况

## 【总结与思考】

- 总结:
  - 通过此次实验，熟悉了对输入信号进行去抖动，同步化处理从而使输入更稳定，同时掌握了对时钟进行分频获得一定频率的时钟信号
- 思考:
  - 在此次实验中，发现在对 Verilog 语言描述电路过程中，由于对时序部分，逻辑部分分配不当，出现了一些意想不到的错误，明白应该从生成的实际电路出发去寻找问题所在
  - 在使用 Vivado 过程中，对其中的一些报错信息，警告信息不太理解，需要加强这方面的经验