

## 徐震 PB20051102 序列检测器设计

### 实验内容

利用有限状态机的原理设计序列检测器和两个序列产生器，中间通过二选一选择器相连，分别产生一个有效序列和一个无效序列，通过切换选择器的选择信号实现被检测序列的切换，检测到有效序列时输出高电平脉冲。

### 设计分析

用有限状态机的原理设计序列检测器和产生器，使用三 `always` 结构和 Moore 状态机模式，

### 源码

```
1  module seq(sel,RESET,CLK,detector_out); //顶层实体
2      input RESET,CLK,sel;
3      output detector_out;
4      wire SEQ_EX,SEQ_IN,DATA_DET,CLK6D;
5
6      SIXFD U0(CLK,CLK6D);
7      SEQ_GENERATE_EXCLUDE U1(CLK6D,RESET,SEQ_EX);
8      SEQ_GENERATE_INCLUDE U2(CLK6D,RESET,SEQ_IN);
9      SELECT U3(sel,SEQ_EX,SEQ_IN,DATA_DET);
10     SEQ_DETECT U4(CLK6D,RESET,DATA_DET,detector_out);
11
12 endmodule
13
```

```
1  module SIXFD(clk,clk6d); //分频器
2      input clk;
3      output clk6d;
4      reg clk6d,clk6d1,clk6d2;
5      reg [499:0] cnt1,cnt2,cnt;
6
7      always@(posedge clk)begin
8          if(cnt1==10'd499)begin cnt1<=10'd000;clk6d1=~clk6d1;end
9          else cnt1<=cnt1+10'd001;
10         end
11
12     always@(posedge clk6d1)begin
13         if(cnt2==10'd499)begin cnt2<=10'd000;clk6d2=~clk6d2;end
14         else cnt2<=cnt2+10'd001;
15         end
16
17     always@(posedge clk6d2)begin
18         if(cnt==10'd009)begin cnt<=10'd000;clk6d=~clk6d;end
19         else cnt<=cnt+10'd001;
20         end
21
22 endmodule
```

```

1 module SEQ_GENERATE_EXCLUDE(CLK,RESET,SEQ_EXCLUDE);//序列产生器, 无效
2     input CLK,RESET;
3     output SEQ_EXCLUDE;
4     reg SEQ_EXCLUDE;
5     reg [2:0]current_st,next_st;
6     parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100,S5=3'b101,S6=3'b110,S7=3'b111;
7
8     always@(posedge CLK or negedge RESET)begin
9         if(!RESET)current_st<=S0;
10        else current_st<=next_st;
11    end
12
13    always@(current_st)begin
14        case(current_st)
15            S0: next_st<=S1;
16            S1: next_st<=S2;
17            S2: next_st<=S3;
18            S3: next_st<=S4;
19            S4: next_st<=S5;
20            S5: next_st<=S6;
21            S6: next_st<=S7;
22            S7: next_st<=S0;
23            default: next_st<=S0;
24        endcase
25    end
26
27    always@(current_st)begin
28        case(current_st)
29            S0: SEQ_EXCLUDE<=1'b1;
30            S1: SEQ_EXCLUDE<=1'b1;
31            S2: SEQ_EXCLUDE<=1'b0;
32            S3: SEQ_EXCLUDE<=1'b0;
33            S4: SEQ_EXCLUDE<=1'b0;
34            S5: SEQ_EXCLUDE<=1'b1;
35            S6: SEQ_EXCLUDE<=1'b1;
36            S7: SEQ_EXCLUDE<=1'b0;
37            default: SEQ_EXCLUDE<=1'b0;
38        endcase
39    end
40
41 endmodule

```

```

1 module SEQ_GENERATE_INCLUDE(CLK,RESET,SEQ_IN);//序列产生器, 有效
2     input CLK,RESET;
3     output SEQ_IN;
4     reg SEQ_IN;
5     reg [3:0]current_st,next_st;
6     parameter S0=4'b0000,S1=4'b0001,S2=4'b0010,S3=4'b0011,S4=4'b0100,S5=4'b0101,S6=4'b0110,S7=4'b0111,S8=4'b1000,S9=4'b1001;
7
8     always@(posedge CLK or negedge RESET)begin
9         if(!RESET)current_st<=S0;
10        else current_st<=next_st;
11    end
12
13    always@(current_st)begin
14        case(current_st)
15            S0: next_st<=S1;
16            S1: next_st<=S2;
17            S2: next_st<=S3;
18            S3: next_st<=S4;
19            S4: next_st<=S5;
20            S5: next_st<=S6;
21            S6: next_st<=S7;
22            S7: next_st<=S8;
23            S8: next_st<=S9;
24            S9: next_st<=S0;
25            default: next_st<=S0;
26        endcase
27    end
28
29    always@(current_st)begin
30        case(current_st)
31            S0: SEQ_IN<=1'b1;
32            S1: SEQ_IN<=1'b1;
33            S2: SEQ_IN<=1'b1;
34            S3: SEQ_IN<=1'b0;
35            S4: SEQ_IN<=1'b1;
36            S5: SEQ_IN<=1'b0;
37            S6: SEQ_IN<=1'b0;
38            S7: SEQ_IN<=1'b1;
39            S8: SEQ_IN<=1'b1;
40            S9: SEQ_IN<=1'b0;
41            default: SEQ_IN<=1'b0;
42        endcase
43    end
44
45 endmodule

```

```

1  module SELECT(sel,SEQ_EX,SEQ_IN,DATA_DET); //选择器
2      input sel,SEQ_EX,SEQ_IN;
3      output DATA_DET;
4      reg DATA_DET;
5
6      always@(sel or SEQ_EX or SEQ_IN)begin
7          case(sel)
8              1'b1: DATA_DET<=SEQ_IN;
9              1'b0: DATA_DET<=SEQ_EX;
10         endcase
11     end
12 endmodule

1  module SEQ_DETECT(CLK,RESET,datainput,detector_out); //检测器
2      input CLK,datainput,RESET;
3      output detector_out;
4      reg detector_out;
5      reg [3:0] current_st,next_st;
6      parameter S0=4'b0000,S1=4'b0001,S2=4'b0010,S3=4'b0011,S4=4'b0100,S5=4'b0101,S6=4'b0110,S7=4'b0111,S8=4'b1000,S9=4'b1001;
7
8      always@(posedge CLK)begin
9          if(!RESET)current_st<=S0;
10         else current_st<=next_st;
11     end
12
13     always@(current_st,datainput)begin
14         case(current_st)
15             S0: begin
16                 if(datainput==1'b1)next_st<=S1;
17                 else next_st<=S0;end
18             S1: begin
19                 if(datainput==1'b1)next_st<=S2;
20                 else next_st<=S0;end
21             S2: begin
22                 if(datainput==1'b1)next_st<=S3;
23                 else next_st<=S0;end
24             S3: begin
25                 if(datainput==1'b0)next_st<=S4;
26                 else next_st<=S3;end
27             S4: begin
28                 if(datainput==1'b1)next_st<=S5;
29                 else next_st<=S0;end
30             S5: begin
31                 if(datainput==1'b0)next_st<=S6;
32                 else next_st<=S2;end
33             S6: begin
34                 if(datainput==1'b0)next_st<=S7;
35
36                 else next_st<=S1;end
37             S7: begin
38                 if(datainput==1'b1)next_st<=S8;
39                 else next_st<=S0;end
40             S8: begin
41                 if(datainput==1'b1)next_st<=S9;
42                 else next_st<=S0;end
43             S9: begin
44                 if(datainput==1'b0)next_st<=S0;
45                 else next_st<=S1;end
46             default: next_st<=S0;
47         endcase
48     end
49
50     always@(current_st)begin
51         if(current_st==S9)detector_out<=1'b1;
52         else detector_out<=1'b0;
53     end
54 endmodule

```

## 仿真

```

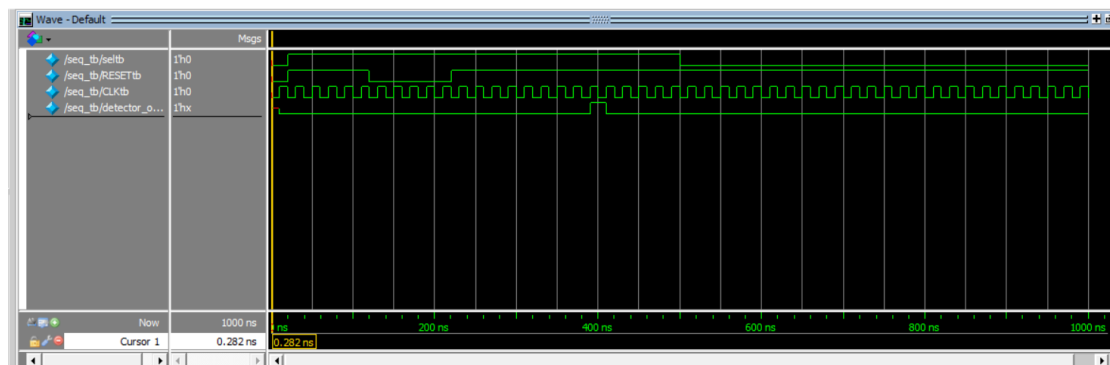
1  module seq(sel,RESET,CLK,detector_out); //顶层实体
2      input RESET,CLK,sel;
3      output detector_out;
4      wire SEQ_EX,SEQ_IN,DATA_DET; //CLK6D;
5
6      //SIXFD U0(CLK,CLK6D);
7      SEQ_GENERATE_EXCLUDE U1(CLK,RESET,SEQ_EX);
8      SEQ_GENERATE_INCLUDE U2(CLK,RESET,SEQ_IN);
9      SELECT U3(sel,SEQ_EX,SEQ_IN,DATA_DET);
10     SEQ_DETECT U4(CLK,RESET,DATA_DET,detector_out);
11
12 endmodule

```

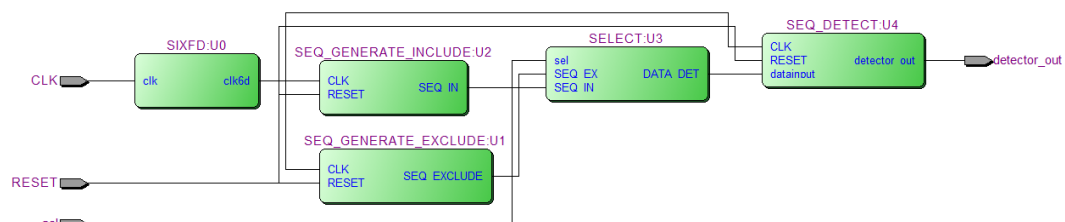
```

1 |`timescale 10ns/1ns
2 |module seq_tb();
3 |    reg seltb,RESETtb,CLKtb;
4 |    wire detector_out_tb;
5 |
6 |    initial begin
7 |        CLKtb<=1'b0;
8 |    end
9 |    always #1 CLKtb = ~CLKtb;
10 |
11 |    initial begin
12 |        RESETtb<=1'b0;
13 |        #2 RESETtb<=1'b1;
14 |        #10 RESETtb<=1'b0;
15 |        #10 RESETtb<=1'b1;
16 |    end
17 |
18 |    initial begin
19 |        seltb<=1'b0;
20 |        #2 seltb<=1'b1;
21 |        #48 seltb<=1'b0;
22 |    end
23 |
24 |    seq U0(seltb,RESETtb,CLKtb,detector_out_tb);
25 |endmodule

```



## 中间结果



Flow Summary	
Flow Status	Successful - Fri Oct 27 12:45:47 2023
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	seq
Top-level Entity Name	seq
Family	Cyclone V
Device	5CEFA2F23C8
Timing Models	Final
Logic utilization (in ALMs)	1,127 / 9,430 ( 12 % )
Total registers	1533
Total pins	4 / 224 ( 2 % )
Total virtual pins	0
Total block memory bits	0 / 1,802,240 ( 0 % )
Total DSP Blocks	0 / 25 ( 0 % )
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total HSSI PMA TX ATT Serializers	0
Total PLLs	0 / 4 ( 0 % )
Total DLLs	0 / 4 ( 0 % )

## FPGA 验证

Pin Planner - D:/FPGA\_EXP/seq/seq - seq

File Edit View Processing Tools Window Help Search altera.com

Report Report not available

Tasks Run Ana Early Pir Early

Top View - Wire Bond  
Cyclone V - 5CEFA2F23C8

Named: \* Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
in CLK	Input	PIN_W16	4A	B4A_N0	PIN_W16	2.5 V (default)
in RESET	Input	PIN_Y11	3B	B3B_N0	PIN_Y11	2.5 V (default)
out detector_out	Output	PIN_D12	7A	B7A_N0	PIN_D12	2.5 V (default)
in sel	Input	PIN_R11	3B	B3B_N0	PIN_R11	2.5 V (default)
<<new node>>						

All Pins

0% 00:00:00

只有当 `sel` 信号和 `reset` 信号都是高电平时才有 `led` 灯闪烁。

## 总结

`Always` 函数的敏感信号要包括检测的输入信号，不是只有时钟与 `reset`！