

# 电子系统设计实验

## ——第二次实验

信息科学技术学院

姓名：胡睿 PB17061124

# 实 验 报 告

评分：

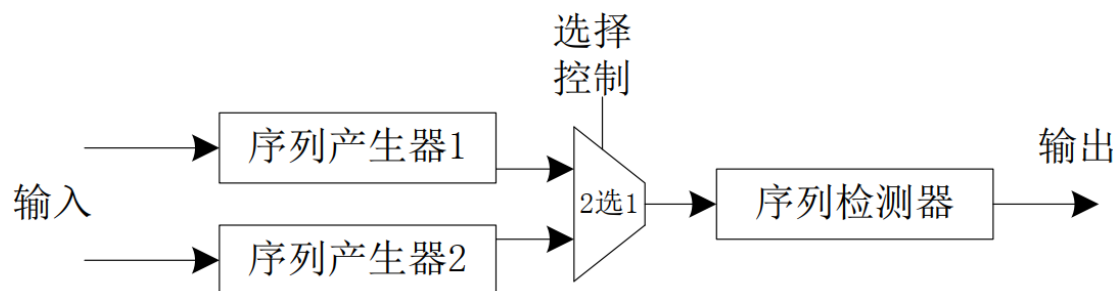
信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

## 【实验题目】 序列检测器

### 【实验目的】

- 1、熟悉并掌握信号发生器的原理与设计；
- 2、熟悉并掌握状态机的原理与设计；
- 3、熟悉并掌握序列检测器的原理与设计；
- 4、熟悉并掌握 VHDL 中元件及其例化；
- 5、进一步掌握 Altera FPGA 设计的流程和步骤。

### 【实验内容】



序列检测器用途很广，在数据通讯、雷达和遥测等领域中常用于检测同步识别标志。

它主要是用一种跳转图来检测一组或多组序列信号。如检测收到一组串行码“111010011”后，输出标志为“1”，否则输出为“0”。

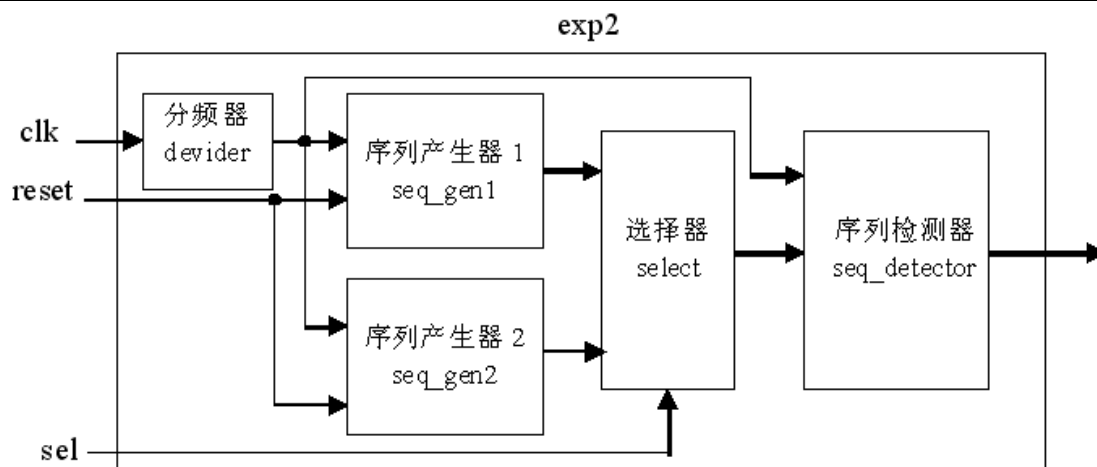
要检测串行码“111010011”，首先要有被测码，也就是说要有序列产生器。然后就是序列检测器，检测序列产生器的输出，有检测到被测序列输出状态“1”，无被测序列则输出状态“0”或无输出。本实验就是要实现含有序列产生器的序列检测器。

### 【设计流程】

# 实 验 报 告

评分：

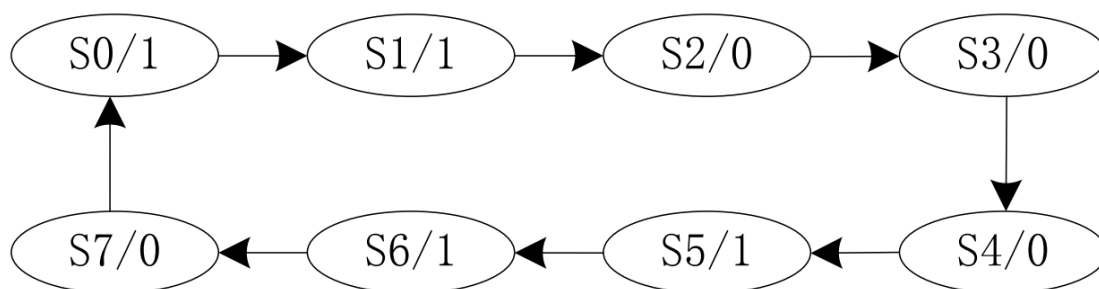
信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_



我们使用如上图所示的结构框图进行设计，在图中给出的实验结构图中包含一个分频器、两个序列产生器、一个选择器和一个序列检测器。其中两个序列产生器中一个产生含有“111010011”序列，一个不包含。用 VHDL 代码完成所有的实验模块，并采用 testbench 进行时序仿真，最后对功能及硬件验证结果进行分析。

## 1、序列产生器+选择器

定义信号 sel 起到选择器的作用，选择器选 1 时序列为 111010011，选择器选 0 时序列为 000101100。定义 8 个状态分别为 S0,S1,S2,S3,S4,S5,S6,S7,S8。



Crst 代表当前状态 Nxst 代表下一状态。当 sel 信号选择 0 或 1 时 S0,S1,S2,S3,S4,S5,S6,S7,S8 分别对应不同的状态，当 reset 信号为 0 时将 S0 状态输出，之后 dataout 信号可以在每个 clk 上升沿将之后的状态输出，这样就可以得到

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

一个序列产生器和选择器。

```
1. ---序列产生器 & 序列选择器 @ 20201029 by hurui
2.
3. library ieee;
4. use ieee.std_logic_1164.all;
5. use ieee.std_logic_unsigned.all;
6.
7. entity seq_generator_hr is --序列产生器
8.     port(
9.         clk: in std_logic;
10.        reset: in std_logic;
11.        sel: in std_logic; --选择器
12.        dataout: out std_logic
13.    );
14. end seq_generator_hr;
15.
16. architecture fsm of seq_generator_hr is
17.     type states is (S0,S1,S2,S3,S4,S5,S6,S7,S8);
18.     signal Crst,Nxst: states;
19.
20. begin
21.
22.     P1:process(clk,reset)
23.     begin
24.         if(reset='0') then
25.             Crst <=S0; --当前状态
26.         elsif(clk'event and clk='1') then
27.             Crst <=Nxst; --下一状态
28.         end if;
29.     end process P1;
30.
31.     P2:process(Crst,sel) --选择器选 1 时序列为 111010011
32.     begin --选择器选 0 时序列为 000101100
33.         case Crst is
34.             when S0=>
35.                 if(sel='1') then
36.                     dataout<='1'; --选择器选 1 时序列为 1
37.                 else
38.                     dataout<='0';
39.                 end if;
```

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
40.      Nxst <=S1;
41.      when S1=>
42.          if(sel='1') then
43.              dataout<='1';    --选择器选 1 时序列为 1
44.          else
45.              dataout<='0';
46.          end if;
47.      Nxst <=S2;
48.      when S2=>
49.          if(sel='1') then
50.              dataout<='1';    --选择器选 1 时序列为 1
51.          else
52.              dataout<='0';
53.          end if;
54.      Nxst <=S3;
55.      when S3=>
56.          if(sel='1') then
57.              dataout<='0';    --选择器选 1 时序列为 0
58.          else
59.              dataout<='1';
60.          end if;
61.      Nxst <=S4;
62.      when S4=>
63.          if(sel='1') then
64.              dataout<='1';    --选择器选 1 时序列为 1
65.          else
66.              dataout<='0';
67.          end if;
68.      Nxst <=S5;
69.      when S5=>
70.          if(sel='1') then
71.              dataout<='0';    --选择器选 1 时序列为 0
72.          else
73.              dataout<='1';
74.          end if;
75.      Nxst <=S6;
76.      when S6=>
77.          if(sel='1') then
78.              dataout<='0';    --选择器选 1 时序列为 0
79.          else
80.              dataout<='1';
81.          end if;
```

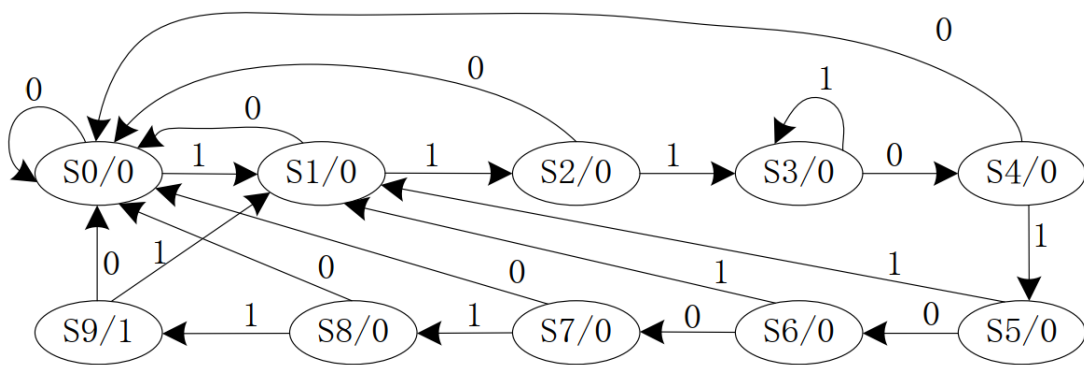
# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
82.          Nxst <=S7;
83.          when S7=>
84.              if(sel='1') then
85.                  dataout<='1';    --选择器选 1 时序列为 1
86.              else
87.                  dataout<='0';
88.              end if;
89.          Nxst <=S8;
90.          when S8=>
91.              if(sel='1') then
92.                  dataout<='1';    --选择器选 1 时序列为 1
93.              else
94.                  dataout<='0';
95.              end if;
96.          Nxst <=S0;
97.      end case;
98.  end process P2;
99. end fsm;
```

## 2、序列检测器



首先在实体中定义输入端口 clk、reset 和 datainput 以及输出端口 detector\_out。

Crst 代表当前状态 Nxst 代表下一状态。当 reset 信号为 0 时将 S0 状态读入当前状态，之后 datainput 信号若与待检测序列相同则依次进入下一状态继续和给定序列对应的状态进行比较，若不同，则返回 S0 状态，在当前状态为 S9 时，证明前九个状态均和给定序列相同，此时输出端口 detector\_out 输出一个高电平。这样就可以得到一个序列检测器。

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
1. ---序列检测器 @ 20201029 by hurui
2.
3. library ieee;
4. use ieee.std_logic_1164.all;
5. use ieee.std_logic_unsigned.all;
6.
7. entity seq_detector_hr is          --序列检测器
8.     port(
9.         clk: in std_logic;
10.        reset: in std_logic;
11.        datainput: in std_logic;
12.        detector_out: out std_logic
13.    );
14. end seq_detector_hr;
15.
16. architecture FSM of seq_detector_hr is
17.     type states is (S0,S1,S2,S3,S4,S5,S6,S7,S8,S9);
18.     signal Crst, Nxst: states;
19.
20. begin
21.
22.     P1:process(clk,reset)
23.     begin
24.         if(reset='0') then
25.             Crst <=S0;
26.         elsif(clk'event and clk='1') then
27.             Crst <=Nxst;
28.         end if;
29.     end process P1;
30.
31.     P2:process(Crst,datainput)
32.     begin
33.         case Crst is
34.             when S0=>
35.                 if(datainput='1') then
36.                     Nxst <=S1;
37.                 else
38.                     Nxst <=S0;
39.                 end if;
40.             when S1=>
41.                 if(datainput='1') then
```

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
42.          Nxst <=S2;
43.      else
44.          Nxst <=S0;
45.      end if;
46.      when S2=>
47.          if(datainput='1') then
48.              Nxst <=S3;
49.          else
50.              Nxst <=S0;
51.          end if;
52.      when S3=>
53.          if(datainput='0') then
54.              Nxst <=S4;
55.          else
56.              Nxst <=S3;
57.          end if;
58.      when S4=>
59.          if(datainput='1') then
60.              Nxst <=S5;
61.          else
62.              Nxst <=S0;
63.          end if;
64.      when S5=>
65.          if(datainput='0') then
66.              Nxst <=S6;
67.          else
68.              Nxst <=S1;
69.          end if;
70.      when S6=>
71.          if(datainput='0') then
72.              Nxst <=S7;
73.          else
74.              Nxst <=S1;
75.          end if;
76.      when S7=>
77.          if(datainput='1') then
78.              Nxst <=S8;
79.          else
80.              Nxst <=S0;
81.          end if;
82.      when S8=>
83.          if(datainput='1') then
```



# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
84.          Nxst <=S9;
85.          else
86.          Nxst <=S0;
87.          end if;
88.          when S9=>
89.          if(datainput='1') then
90.          Nxst <=S1;
91.          else
92.          Nxst <=S0;
93.          end if;
94.          end case;
95.      end process P2;
96.
97.      P3:process(Crst)
98.      begin
99.          case Crst is
100.             when S0 => detector_out<='0';
101.             when S1 => detector_out<='0';
102.             when S2 => detector_out<='0';
103.             when S3 => detector_out<='0';
104.             when S4 => detector_out<='0';
105.             when S5 => detector_out<='0';
106.             when S6 => detector_out<='0';
107.             when S7 => detector_out<='0';
108.             when S8 => detector_out<='0';
109.             when S9 => detector_out<='1';
110.          end case;
111.      end process P3;
112.
113. end FSM;
```

## 3、分频器

对较高频率的时钟进行分频操作得到较低频率的时钟信号。定义 div\_sel 信号便于在调试时选择对时钟信号的不同分频使得现象更加直观。

```
1. ---分频器 @ 20201029 by hurui
2.
3. library ieee;
4. use ieee.std_logic_1164.all;
```

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
5. use ieee.std_logic_unsigned.all;
6.
7. entity divider_sel_hr IS
8.     port(
9.         clk_in: in std_logic;
10.        reset: in std_logic;
11.        div_sel: in std_logic_vector(1 downto 0);
12.        clk_out: out std_logic
13.    );
14. end divider_sel_hr;
15.
16. architecture div of divider_sel_hr is
17.     signal cnt: std_logic_vector(31 downto 0);
18. begin
19.     process(clk_in, reset)
20.     begin
21.         if(reset='0') then
22.             cnt<=X"00000000";
23.         elsif(clk_in'event and clk_in='1') then
24.             cnt<=cnt+1;
25.         end if;
26.     end process;
27.     process(div_sel, cnt)
28.     begin
29.         case div_sel is
30.             when "00"=>
31.                 clk_out <=cnt(20);
32.             when "01"=>
33.                 clk_out <=cnt(21);
34.             when "10"=>
35.                 clk_out <=cnt(22);
36.             when "11"=>
37.                 clk_out <=cnt(23);
38.             when others=>
39.                 clk_out <='0';
40.         end case;
41.     end process;
42. end div;
```

## 4、顶层模块

将以上所有模块相连接定义为一个实体。

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
1. ---顶层模块 @ 20201029 by hurui
2.
3. library ieee;
4. use ieee.std_logic_1164.all;
5. use ieee.std_logic_unsigned.all;
6.
7. entity seq_detec_hr is                                --顶层模块
8.     port(
9.         clk: in std_logic;
10.        reset: in std_logic;
11.        sel: in std_logic;
12.        div_sel: in std_logic_vector(1 downto 0);
13.        detector_out: out std_logic
14.    );
15. end seq_detec_hr;
16.
17. architecture whole of seq_detec_hr is
18.
19.     component seq_detector_hr is                        --序列检测器
20.         port(
21.             clk: in std_logic;
22.             reset: in std_logic;
23.             datainput: in std_logic;
24.             detector_out: out std_logic
25.         );
26.     end component;
27.
28.     component seq_generator_hr is                       --序列产生器 & 序列选择器
29.         port(
30.             clk: in std_logic;
31.             reset: in std_logic;
32.             sel: in std_logic;
33.             dataout: out std_logic
34.         );
35.     end component;
36.
37.     component divider_sel_hr is                         --分频器
38.         port(
39.             clk_in: in std_logic;
40.             reset: in std_logic;
41.             div_sel: in std_logic_vector(1 downto 0);
```

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
42.         clk_out:out std_logic
43.     );
44. end component;
45.
46.     signal data_out: std_logic;
47.     signal clk_out: std_logic;
48.
49. begin
50.     detect:seq_detector_hr port map(clk_out,reset,data_out,detector_out);
51.     gen: seq_generator_hr port map(clk_out,reset,sel,data_out);
52.     div: divider_sel_hr port map(clk,reset,div_sel,clk_out);
53. end whole;
```

## 5、序列检测器 testbench

reset 信号取 0 间隔 10ns 后取 1 使序列产生器开始工作，将 sel 信号分别取 0 和 1 可以看到输入不同序列时序列检测器输出的波形。

```
1. ---序列检测器 testbench @ 20201029 by hurui
2.
3. library ieee;
4. use ieee.std_logic_1164.all;
5. use ieee.std_logic_arith.all;
6. use ieee.std_logic_unsigned.all;
7.
8. entity seq_detec_hr_tb is
9. end seq_detec_hr_tb;
10.
11. architecture arch_tb of seq_detec_hr_tb is
12.     component seq_detec_hr is
13.         port(
14.             clk: in std_logic;
15.             reset: in std_logic;
16.             sel: in std_logic;
17.             div_sel: in std_logic_vector(1 downto 0);
18.             detector_out: out std_logic
19.         );
20.     end component;
21.     signal clk: std_logic:='0';
22.     signal reset: std_logic:='1';
```

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

```
23. signal sel: std_logic:='0';
24. signal div_sel: std_logic_vector(1 downto 0):="00";
25. signal detector_out: std_logic;
26. begin
27. ins: seq_detec_hr port map(clk,reset,sel,div_sel,detector_out);
28. process
29. begin
30.     wait for 10ns;
31.     clk <='1';
32.     wait for 10ns;
33.     clk <='0';
34. end process;
35. process
36. begin
37.
38.     reset <='0';
39.     sel<='1';
40.     wait for 10ns;
41.     reset <='1';
42.     wait for 1000ns;
43.
44.     reset <='0';
45.     sel<='0';
46.     wait for 10ns;
47.     reset <='1';
48.     wait for 1000ns;
49. end process;
50. end arch_tb;
```

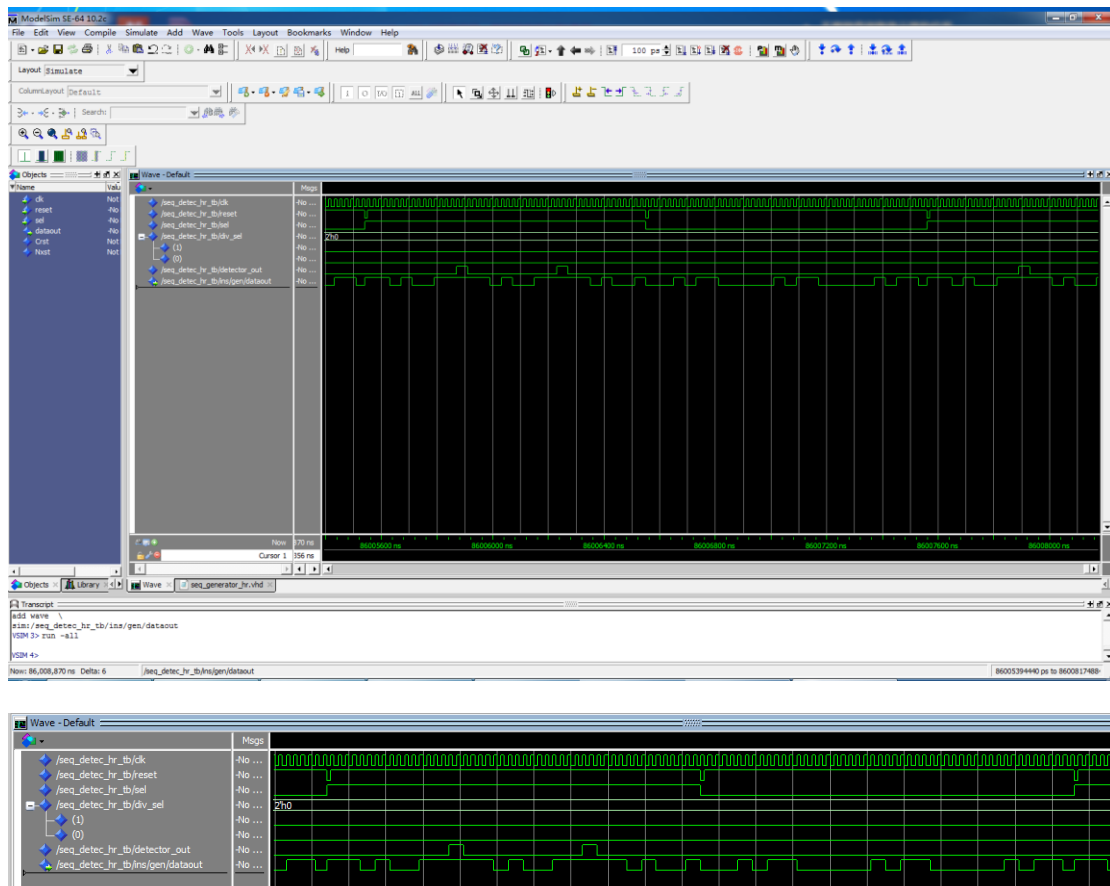
## 【实验结果】

启动 ModelSim 完成设计功能仿真将结束仿真时间设定为设定的 2000 ns, 等待 ModelSim 运行完 2000ns 可以看到 Modelsim 的基本仿真界面及仿真结果, 如下图所示:

# 实 验 报 告

评分：

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_



可以看到波形从上至下依次为：clk 时钟信号、reset 重置信号、sel 选择器信号、div\_sel 分频倍数选择信号、detector\_out 输出检测信号、dataout 信号产生器产生经过选择器选择的待检测的信号。

可以看到选择器 sel 信号为高电平时 dataout 输出信号为 111010011，选择器 sel 信号为低电平时 dataout 输出信号为 000101100。并且在每个 111010011 即待检测的信号结束时 detector\_out 输出检测信号均出现一个高电平，符合设计要求。

将 reset 重置信号、sel 选择器信号、div\_sel 分频倍数选择信号分别配置为四个 DIP 开关的管脚上，将 detector\_out 输出检测信号配置到 LED 灯管脚上，将编译后的文件烧写入 FPGA 开发板中，现象为：当 sel 选择器信号取 0 时 LED 灯无现象，当 reset 重置信号取 0 时 LED 灯无现象，当 sel 取 1 且 reset 取 1 时 LED

# 实 验 报 告

评分:

信 院 系 17 级 姓名 胡 睿 日期 2020-10-29 NO. \_\_\_\_\_

---

灯规律闪烁，说明检测在一定的时间周期内检测到了给定的信号，通过调节 div\_sel 分频倍数选择信号可以看到在不同的分频倍数下 LED 闪烁频率变化，这是由于时钟频率改变导致的。

## 【实验总结】

- 1、在输入 vhdl 代码时一定要注意中英文字符的使用，如果不小心输入了中文字符错误很难发现；
- 2、在调整代码过后一定要 Start Compilation 实现 Analysis & Synthesis、Fitter 及 Assembler 等完整的全编译过程并成功后再进行烧写，否则烧写的依然是原来的代码。
- 3、在编写 tb 文件时一定要注意区分实体和测试实体，否则很容易出现变量重复定义的报错，此时可以检查是否在使用实体名称时将两者混淆。

## 【实验收获】

- 1、熟悉了在 Quart II 中建立工程、编写和调试 vhdl 代码的方法，进一步掌握 Altera FPGA 设计的流程和步骤。
- 2、熟悉使用 modelsim 仿真工具并分析时序图。
- 3、熟悉了为设计工程添加约束、管脚分配的方法。
- 4、熟悉并掌握信号发生器、状态机的原理与设计；
- 5、熟悉并掌握序列检测器的原理与设计；
- 6、熟悉并掌握 VHDL 中元件及其例化；