

数字电路与数字系统实验

EX08:状态机及键盘输入

191220029 傅小龙
周一 5-6 节班
1830970417@qq.com
2020 年 11 月 16 日

目录

一、实验内容	3
1.1 实验要求	3
1.2 实验工具	3
二、实验过程	3
2.1 模型概述	3
2.2 数字抽象	4
2.3 建立模型	6
2.4 分析/综合	6
2.5 分配引脚	7
2.6 全编译	9
三、实验总结	9
四、附	9
4.1 七段数码管编码器的 Verilog 实现	9

一、实验内容

1.1 实验要求

自行设计状态机，实现单个按键的 ASCII 码显示。

七段数码管低两位显示当前按键的键码，中间两位显示对应的 ASCII 码（转换可以考虑自行设计一个 ROM 并初始化）。只需完成字符和数字键的输入，不需要实现组合键和小键盘。当按键松开时，七段数码管的低四位全灭。

七段数码管的高两位显示按键的总次数。按住不放只算一次按键。只考虑顺序按下和放开的情况，不考虑同时按多个键的情况。

支持 Shift, CTRL 等组合键，在 LED 上显示组合键是否按下的状态指示；

支持 Shift 键与字母/数字键同时按下，相互不冲突；

支持输入大写字符，显示对应的 ASCII 码；

1.2 实验工具

软件环境：

设计、编译、仿真：Quartus Prime Version 17.1.0 Build 590 10/25/2017 SJ Lite Edition

DE10_Standard_SystemBuilder
硬件环境：DE-10 Standard 开发平台

FPGA 芯片：Cyclone V 5CSXFC6D6F31C6

二、实验过程

2.1 模型概述

在 exp08.pdf 表 8 4: 键盘控制器给出的 Verilog 代码的基础上实现对键盘输出的键码及其对应的 ASCII 码的显示，以及能够对按键次数进行统计。

2.2 数字抽象

I) 输入:

数据输入[7:0]datain: 键码输入.
状态输入 shift_state: shift 按键状态.
状态输入 caps_state: capslock 按键状态.
状态输入 ctrl_state: ctrl 按键状态.

II) 输出:

数据输出[7:0]dataout: ASCII 码输出.
下表\图给出了以上输入输出信号在 DE10 平台对应的信号:

	信号名称	DE-10 平台信号
输入	[7:0]datain	[7:0]temp
	shift_state	shift_state
	caps_state	caps_state
	ctrl_state	ctrl_state
输出	[7:0]dataout	[7:0]ascii

*: 关于表项中的 DE-10 平台信号的含义详见 2.3 节中的相关内容.

表 2-1:键盘码转 ASCII 码输入输出信号与 DE10 平台信号对应关系

2.3 建立模型

kbdecoder 模块中, 相应键码的 ASCII 码用类似 ROM 的形式存放, 与键码是一一对应关系. 再根据 shift, capslock 键的状态对 ASCII 码的输出进行赋值.

下面是 kbdecoder 模块的相关代码:

```
1 module kbdecoder(datain, dataout, shift_state, caps_state,
2   ctrl_state);
3   input [7:0]datain;
4   input shift_state; //shift 键是否被按下
5   input caps_state;  //caps 键是否被按下
6   input ctrl_state;  //ctrl 键是否被按下
7   output reg [7:0]dataout;
8
9   reg [7:0] asc [255:0];
10  reg [7:0] ascii_shift [255:0];
11  reg [7:0] ascii_caps [255:0];
12  reg [7:0] temp;
13
14  initial
```

```

15  begin
16      $readmemh(".\\init\\ascii_init.txt", asc, 0, 255);
17      $readmemh(".\\init\\ascii_init_shift.txt", ascii_shift,
18  0, 255);
19      $readmemh(".\\init\\ascii_init_caps.txt", ascii_caps, 0,
20  255);
21
22  end
23
24  always @(*)
25  begin
26      if(shift_state && !caps_state) begin
27          dataout <= ascii_shift[datain];
28      end
29      else if(caps_state && !shift_state) begin
30          dataout <= ascii_caps[datain];
31      end
32      else dataout <= asc[datain];
33  end
34  endmodule
35

```

初始化文件详见项目文件夹的 init/文件夹中的.txt 文件.

对于 shift, capslock 按键的状态设计和 next_data_n 信号的设置在顶层文件中给出如下实现:

```

1  always@(posedge CLOCK_50) begin
2      if(ready == 1 && next_data_n == 1)begin
3          temp <= keycode;
4          next_data_n <= 0;
5          if(keycode == 8'hf0) begin//realse
6              release_flag <= 1;
7              cnt <= cnt + 1;
8          end
9          else if(keycode == 8'h12 || keycode == 8'h59)
10 begin//shift
11     if(release_flag) begin
12         shift_state <= 0;
13         e_out <= 0;
14         release_flag <= 0;
15     end
16     else begin
17         shift_state <= 1;
18         e_out <= 1;
19     end
20 end

```

```

21         else if(keycode == 8'h14) begin //ctrl
22             if(release_flag) begin
23                 ctrl_state <= 0;
24                 e_out <= 0;
25                 release_flag <= 0;
26             end
27             else begin
28                 ctrl_state <= 1;
29                 e_out <= 1;
30             end
31         end
32         else begin
33             if(release_flag) begin
34                 e_out <= 0;
35                 release_flag <= 0;
36                 if(keycode == 8'h58) caps_state <= ~caps_state;
37             else;
38             end
39             else e_out <= 1;
40         end
41     end
42     else begin
43         next_data_n <= 1;
44     end

```

需要注意的是 keyboard 模块给出的键盘码输出并不一定是有效的，只有在 ready 和 next_data_n 信号都为 1 时才有效（Line2）.获取有效的键码输出后，需要将 next_data_n 信号置 0 以准备接受下一个有效信号。

对于松开按键后 HEX 数码管熄灭的实现：通过 reg 型变量 release_flag 记录是否有 f0 键码输出。在接收到 f0 键码的下一个时钟周期将数码管的使能端输入置 0（Line 5-8, 33-37）.松开按键则计数加 1.关于数码管译码器的设计详见附-4.1.

对于 shift, ctrl 按键状态的设置：若 release_flag 为零，则说明是按键按下，将对应的标志信号置 1，否则置 0 并设置数码管的使能信号为 0(Line 9-31).

对于 caps_lock 按键状态的设置：caps_lock 键按下、松开后 caps_lock 按键的状态改变一次（Line 33-38）.

2.4 分析/综合

分析/综合实验成功，如下图所示：

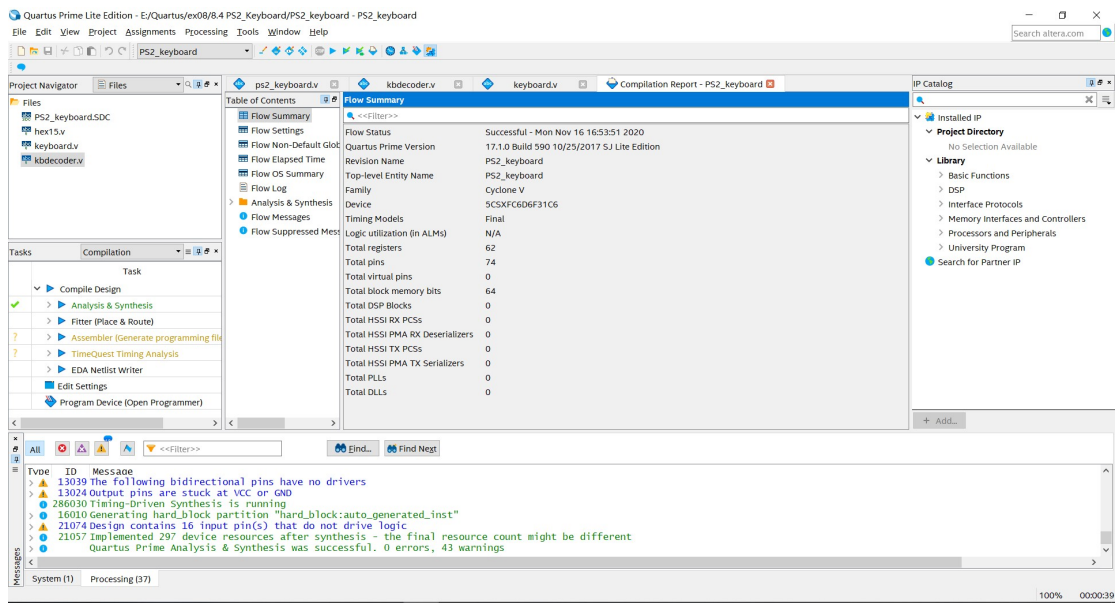


图 2-4-1：分析/综合成功

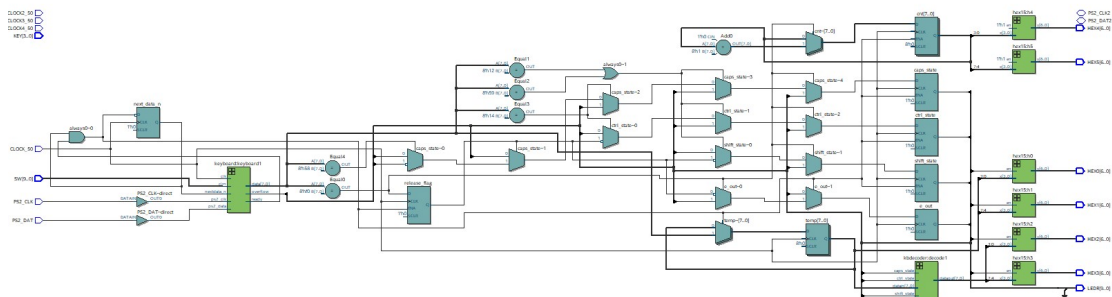


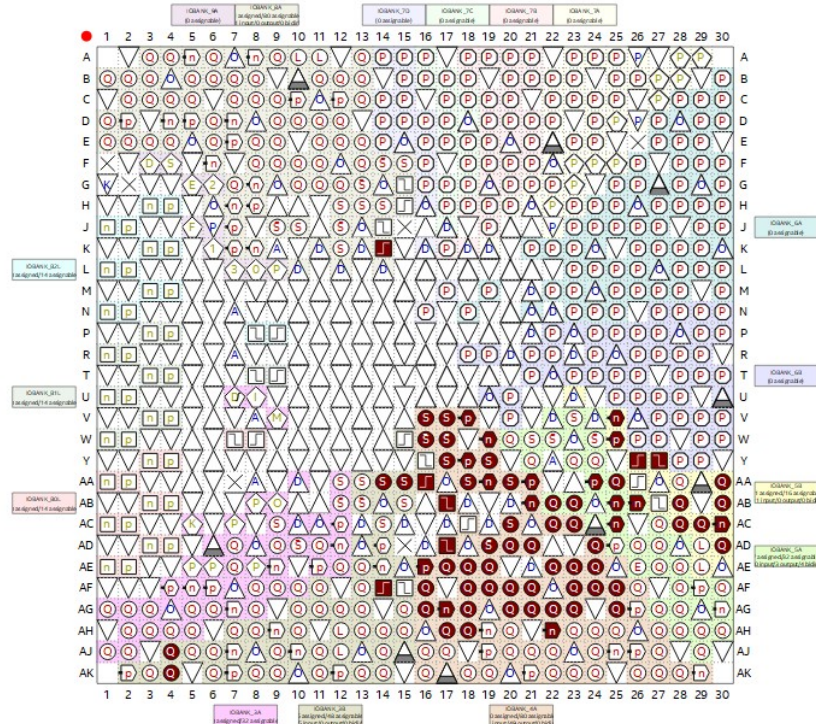
图 2-4-2:RTL 视图

2.5 分配引脚

引脚分配使用 DE10_Standard_SystemBuilder 生成。

Top View - Wire Bond

Cyclone V - 5CSXFC6D6F31C6



Node Name	Direction	Location	I/O Bank	VREF Group	Iter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Analog Setting	GPIO/CTX_CFI	GPIO Pin Termination	Configured Refclk	Common Mode	Iter Slew Rate	Differential Output
CLOCK_50	Input	PIN_AA16	4A	B4A_NO	PIN_AA16	3.3-V LVTTTL		16mA ..auto									
CLOCK_50	Input	PIN_Y26	5B	B5B_NO	PIN_Y26	2.5 V		12mA ..auto									
CLOCK_50	Input	PIN_K14	8A	B8A_NO	PIN_K14	3.3-V LVTTTL		16mA ..auto									
CLOCK_50	Input	PIN_AF14	3B	B3B_NO	PIN_AF14	3.3-V LVTTTL		16mA ..auto									
HEX0[6]	Output	PIN_AH18	4A	B4A_NO	PIN_AH18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[5]	Output	PIN_AG18	4A	B4A_NO	PIN_AG18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[4]	Output	PIN_AH17	4A	B4A_NO	PIN_AH17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[3]	Output	PIN_AG16	4A	B4A_NO	PIN_AG16	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[2]	Output	PIN_AG17	4A	B4A_NO	PIN_AG17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[1]	Output	PIN_V18	4A	B4A_NO	PIN_V18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX0[0]	Output	PIN_W17	4A	B4A_NO	PIN_W17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[6]	Output	PIN_V17	4A	B4A_NO	PIN_V17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[5]	Output	PIN_AE17	4A	B4A_NO	PIN_AE17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[4]	Output	PIN_AE18	4A	B4A_NO	PIN_AE18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[3]	Output	PIN_AD17	4A	B4A_NO	PIN_AD17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[2]	Output	PIN_AE16	4A	B4A_NO	PIN_AE16	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[1]	Output	PIN_V16	4A	B4A_NO	PIN_V16	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX1[0]	Output	PIN_AF16	4A	B4A_NO	PIN_AF16	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[6]	Output	PIN_W16	4A	B4A_NO	PIN_W16	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[5]	Output	PIN_AF18	4A	B4A_NO	PIN_AF18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[4]	Output	PIN_Y18	4A	B4A_NO	PIN_Y18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[3]	Output	PIN_Y17	4A	B4A_NO	PIN_Y17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[2]	Output	PIN_AA18	4A	B4A_NO	PIN_AA18	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[1]	Output	PIN_AH17	4A	B4A_NO	PIN_AH17	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX2[0]	Output	PIN_AA21	4A	B4A_NO	PIN_AA21	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[6]	Output	PIN_AD20	4A	B4A_NO	PIN_AD20	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[5]	Output	PIN_AA19	4A	B4A_NO	PIN_AA19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[4]	Output	PIN_AC20	4A	B4A_NO	PIN_AC20	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[3]	Output	PIN_AA20	4A	B4A_NO	PIN_AA20	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[2]	Output	PIN_AD19	4A	B4A_NO	PIN_AD19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[1]	Output	PIN_W19	4A	B4A_NO	PIN_W19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX3[0]	Output	PIN_Y19	4A	B4A_NO	PIN_Y19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[6]	Output	PIN_AH22	4A	B4A_NO	PIN_AH22	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[5]	Output	PIN_AF23	4A	B4A_NO	PIN_AF23	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[4]	Output	PIN_AG23	4A	B4A_NO	PIN_AG23	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[3]	Output	PIN_AE23	4A	B4A_NO	PIN_AE23	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[2]	Output	PIN_AE22	4A	B4A_NO	PIN_AE22	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[1]	Output	PIN_AG22	4A	B4A_NO	PIN_AG22	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX4[0]	Output	PIN_AD21	4A	B4A_NO	PIN_AD21	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[6]	Output	PIN_AB21	4A	B4A_NO	PIN_AB21	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[5]	Output	PIN_AF19	4A	B4A_NO	PIN_AF19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[4]	Output	PIN_AE19	4A	B4A_NO	PIN_AE19	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[3]	Output	PIN_AG20	4A	B4A_NO	PIN_AG20	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[2]	Output	PIN_AF20	4A	B4A_NO	PIN_AF20	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[1]	Output	PIN_AG21	4A	B4A_NO	PIN_AG21	3.3-V LVTTTL		16mA ..auto	1 (default)								
HEX5[0]	Output	PIN_AF21	4A	B4A_NO	PIN_AF21	3.3-V LVTTTL		16mA ..auto	1 (default)								
KEY[3]	Input	PIN_AA15	3B	B3B_NO	PIN_AA15	3.3-V LVTTTL		16mA ..auto									
KEY[2]	Input	PIN_AA14	3B	B3B_NO	PIN_AA14	3.3-V LVTTTL		16mA ..auto									
KEY[1]	Input	PIN_AK4	3B	B3B_NO	PIN_AK4	3.3-V LVTTTL		16mA ..auto									
KEY[0]	Input	PIN_AJ4	3B	B3B_NO	PIN_AJ4	3.3-V LVTTTL		16mA ..auto									
LED0[9]	Output	PIN_AC22	4A	B4A_NO	PIN_AC22	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[8]	Output	PIN_AB22	5A	B5A_NO	PIN_AB22	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[7]	Output	PIN_AF24	4A	B4A_NO	PIN_AF24	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[6]	Output	PIN_AE24	4A	B4A_NO	PIN_AE24	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[5]	Output	PIN_AF25	4A	B4A_NO	PIN_AF25	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[4]	Output	PIN_AG25	4A	B4A_NO	PIN_AG25	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[3]	Output	PIN_AD24	4A	B4A_NO	PIN_AD24	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[2]	Output	PIN_AC23	4A	B4A_NO	PIN_AC23	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[1]	Output	PIN_AB23	5A	B5A_NO	PIN_AB23	3.3-V LVTTTL		16mA ..auto	1 (default)								
LED0[0]	Output	PIN_AA24	5A	B5A_NO	PIN_AA24	3.3-V LVTTTL		16mA ..auto	1 (default)								
PS2_CLK	Bidir	PIN_AB25	5A	B5A_NO	PIN_AB25	3.3-V LVTTTL		16mA ..auto	1 (default)								
PS2_CLK2	Bidir	PIN_AC25	5A	B5A_NO	PIN_AC25	3.3-V LVTTTL		16mA ..auto	1 (default)								
PS2_DAT	Bidir	PIN_AA25	5A	B5A_NO	PIN_AA25	3.3-V LVTTTL		16mA ..auto	1 (default)								
PS2_DAT2	Bidir	PIN_AB26	5A	B5A_NO	PIN_AB26	3.3-V LVTTTL		16mA ..auto	1 (default)								
SW[9]	Input	PIN_AA30	5B	B5B_NO	PIN_AA30	2.5 V		12mA ..auto									
SW[8]	Input	PIN_AC29	5B	B5B_NO	PIN_AC29	2.5 V		12mA ..auto									
SW[7]	Input	PIN_AD30	5B	B5B_NO	PIN_AD30	2.5 V		12mA ..auto									
SW[6]	Input	PIN_AC28	5B	B5B_NO	PIN_AC28	2.5 V		12mA ..auto									
SW[5]	Input	PIN_V25	5B	B5B_NO	PIN_V25	2.5 V		12mA ..auto									
SW[4]	Input	PIN_W25	5B	B5B_NO	PIN_W25	2.5 V		12mA ..auto									
SW[3]	Input	PIN_AC30	5B	B5B_NO	PIN_AC30	2.5 V		12mA ..auto									
SW[2]	Input	PIN_AB28	5B	B5B_NO	PIN_AB28	2.5 V		12mA ..auto									
SW[1]	Input	PIN_Y27	5B	B5B_NO	PIN_Y27	2.5 V		12mA ..auto									
SW[0]	Input	PIN_AB30	5B	B5B_NO	PIN_AB30	2.5 V		12mA ..auto									

图 2-5 引脚分配图

2.6 全编译

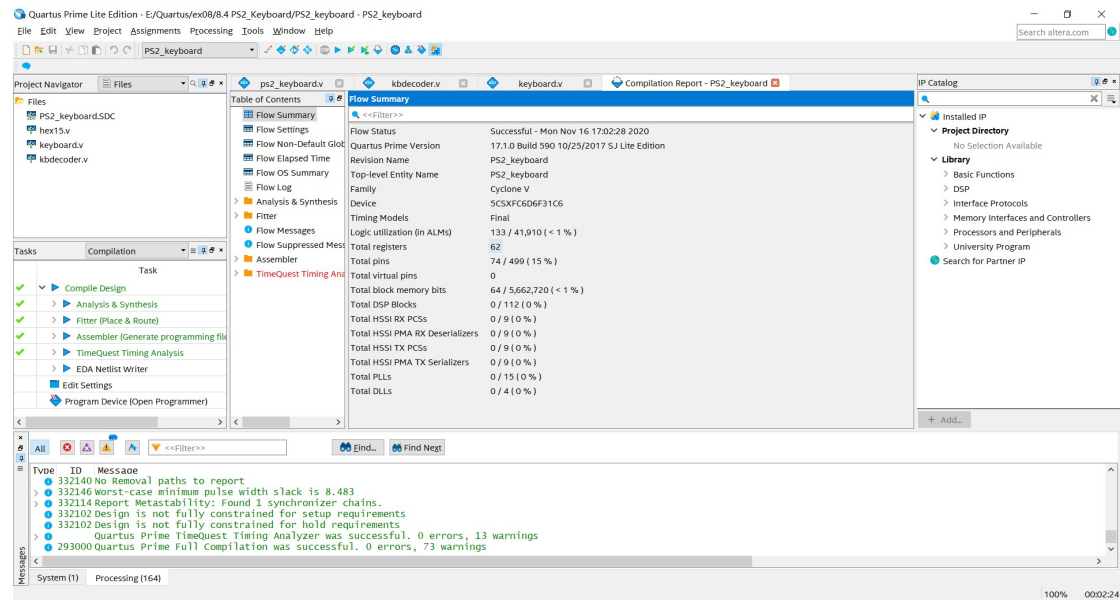


图 2-6 全编译成功

三、实验总结

本次实验主要学习实践了状态机的设计、应用. 还学习了 PS2 键盘的工作方式并在以键盘码接收模块的基础上设计了将键盘码转为 `ascii` 码、按键次数的状态机进行了设计.需要注意的是只有正确理解并运用数据的输入输出指示信号的时序逻辑关系才能使设计的状态机正确工作.

四、附

4.1 七段数码管编码器的 Verilog 实现

```
1
2 module hex15(x, y, en);
3     input [3:0]x;
4     output reg [6:0]y;
5     input en;
6
7     always @(*) begin
8         if(!en) y = 7'b1111111;
9         else case(x)
```

```
9          4'b0000:y = 7'b1000000;
10         4'b0001:y = 7'b1111001;
11         4'b0010:y = 7'b0100100;
12         4'b0011:y = 7'b0110000;
13         4'b0100:y = 7'b0011001;
14         4'b0101:y = 7'b0010010;
15         4'b0110:y = 7'b0000010;
16         4'b0111:y = 7'b1111000;
17         4'b1000:y = 7'b0000000;
18         4'b1001:y = 7'b0010000;
19         4'b1010:y = 7'b0001000;
20         4'b1011:y = 7'b0000011;
21         4'b1100:y = 7'b1000110;
22         4'b1101:y = 7'b0100001;
23         4'b1110:y = 7'b0000110;
24         4'b1111:y = 7'b0001110;
25         default: y = 7'b1111111;
26     endcase
27 end
28 endmodule
```