数字电路与数字系统实验

EX10:音频输出实验

191220029 傅小龙
周一 5-6 节班
1830970417@qq.com
2020 年 11 月 29 日

目录

一、实验内容

1.1 实验要求

将之前实验实现的键盘与本实验的音频输出结合，实现一个简单的键盘电子琴功能。钢琴上的不同音高对应着不同的频率。我们可以根据按下的键的键值，决定播放的正弦的频率，从而实现电子琴的功能。

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

在给出 I2C_Audio_Config, I2S_Audio, Sin_Generator 模块参考代码的基础上，用一个状态机接收键盘传送的键码，根据按下的按键数量对要输出的频率做和声处理，再将要输出的频率传给 Sin_Generator 模块转换成模拟信号. 音量调整通过修改 I2C_Audio_Config 中发送给 WM8731 芯片音量的数据值实现.

2.2 数字抽象

下图给出了各模块间的关系及其作用, 其中 exp10audio 为顶层文件名.

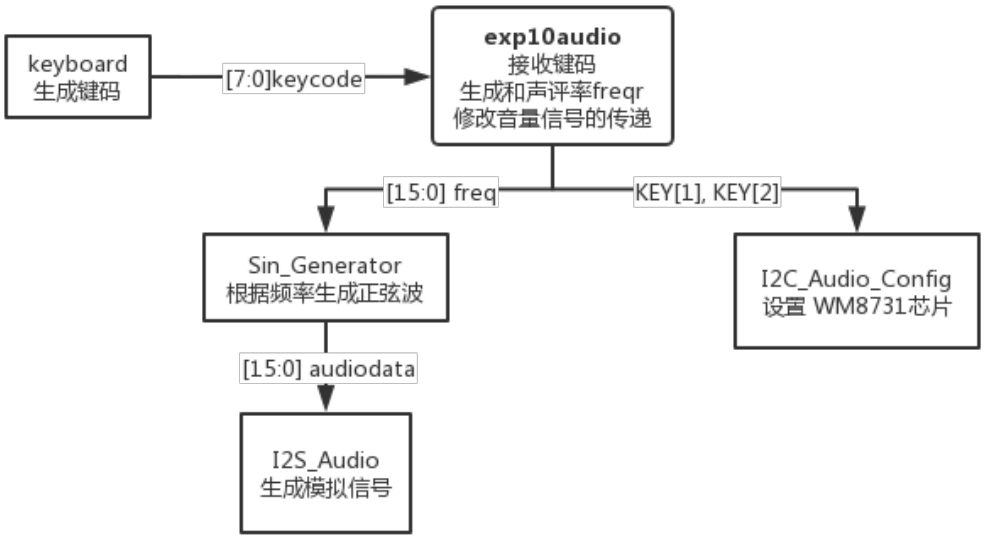


图 2-2-1 模块关系示意图

2.3 建立模型

a) 状态机的设计

该状态机包含 4 个状态 Q0, Q1, Q2, Q3，分别表示有 0 个, 1 个, 2 个, 3 个音的和声，并能够根据键盘模块的 ready 和 next_data_n 信号获取正确的键码以及根据状态数输出相应的频率.

状态机的状态转化条件以及相应的处理方式为：①当获取的键码为通码且与之前的通码不同时，状态数加一并将新获取的通码储存; ②当获取的键码为断码时，状态码减一，并在存储通码的各变量中将松开按键的键码删除.

状态机的输出以当前状态取有效键码对应的频率取平均后参照 ex10.pdf 中计算递增值的方式得到应当传递给 Sin_Generator 模块的频率.(对应下表中 Line66 开始的代码).

下面是上述状态机的相关代码：

```
1    always@(posedge CLOCK_50) begin
2        if(ready == 1 && next_data_n == 1)begin
3        //if(ready == 1)beginf
4            if(temp != keycode) neflag = 1;
5        else;
6            temp <= keycode;
7            next_data_n <= 0;
8            if(keycode == 8'hf0) begin//realse
9                release_flag <= 1;
10               //temp <= keycode;
11               case(state)
```

```
12                  Q0: state <= Q0;
13                  Q2: state <= Q1;
14                  Q3: state <= Q2;
15                  default: state <= Q0;
16              endcase;
17          end
18          else begin
19              if(release_flag) begin
20                  e_out <= 0;
21                  release_flag <= 0;
22                  if(key0 == keycode) begin
23                      key0 = key1;
24                      key1 = key2;
25                      key2 = 0;
26                  end
27                  else if(key1 == keycode) begin
28                      key1 = key2;
29                      key2 = 0;
30                  end
31                  else if(key2 == keycode)begin
32                      key2 = 0;
33                  end
34                  else;
35              end
36              else begin
37                  e_out <= 1;
40                  if(neflag) begin
41                      case(state)
42                          Q0: begin state <= Q1;    key0 =
43  keycode; end
44                          Q1: begin state <= Q2;key1 = key0; key0
45  = keycode; end
46                          Q2: begin state <= Q3;key2 = key1; key1
47  = key0; key0 = keycode; end
48                          Q3: begin state <= Q3;    key2 = key1;
49  key1 = key0; key0 = keycode; end
50                          default: state <= 0;
51                      endcase
52                      neflag = 0;
53                  end
54                  else ;
55              end
56          end
57      end
```

```
58
59      else begin
60          next_data_n <= 1;
61      end
62  end
63
64  assign LEDR[8:7] = state;
65
66  always @(state) begin
67      if(!SW[0])
68          freqr = 16'h0400;
69      else begin
70          case(state)
71              Q0: freqr = 16'h0;
72              Q1: freqr = rom_freq[key0] * a1 / a2;
73              Q2: freqr = (rom_freq[key1] / 2 + rom_freq[key0] /
74  2) * a1 / a2;
75              Q3: freqr = (rom_freq[key2] / 3 + rom_freq[key1] /
76  3 + rom_freq[key0] / 3) * a1 / a2;
77              default:freqr = 16'h0400;
78          endcase
79      end
80  end
```

对于音量的修改则通过操作开发板上的按钮 KEY[1]和 KEY[2]使 I2C_Audio_Config 模块给 WM8731 芯片传送不同的音频设置数据实现.这里共设置了 6 组不同音量的初始化数据，其他参量均一致. 即只有下面这两行代码所示的左右音量不同，故这里不再全部展示.

```
audio_reg1[3]= 7'h02; audio_cmd1[3]=9'h40; //Left Volume
audio_reg1[4]= 7'h03; audio_cmd1[4]=9'h40; //Right Volume
```

在 I2C_Audio_Config 模块内设置 reg 型变量 x 以存储当前使用的初始化数据组的标号. 根据 KEY[1]和 KEY[2]输入改变 x 的值，KEY[1]和 KEY[2]分别对应模块内的 volume_up, volume_down 接口. 在对变量 x 作出修改后，将在下一个 I2C_Audio_Config 模块的 reset_n 重置信号有效时根据 x 的值选择相应的数据组修改 WM8731 芯片中的相应参数.

以下是对 x 变量修改的相关代码：

```
1       always @ (negedge volume_up or negedge volume_down) begin
2           if(!volume_up) begin //set volume++
3               if(reg_x < 4'h6) reg_x <= reg_x + 4'h1;
4               else ;
5           end
6           else if(!volume_down) begin //set volume--
7               if(reg_x > 4'h1) reg_x <= reg_x - 4'h1;
8               else ;
```

```
        end
        else ;
  end
```

以下是对 I2C_Audio_Config 模块中要传送给 WM8731 芯片数据的部分代码的修改：

```
        case(reg_x)
                        4'h1:mi2c_data <= {audio_addr,
  audio_reg1[cmd_count], audio_cmd1[cmd_count]};
                        4'h2:mi2c_data <= {audio_addr,
  audio_reg2[cmd_count], audio_cmd2[cmd_count]};
                        4'h3:mi2c_data <= {audio_addr,
  audio_reg3[cmd_count], audio_cmd3[cmd_count]};
                        4'h4:mi2c_data <= {audio_addr,
  audio_reg4[cmd_count], audio_cmd4[cmd_count]};
                        4'h5:mi2c_data <= {audio_addr,
  audio_reg5[cmd_count], audio_cmd5[cmd_count]};
                        4'h6:mi2c_data <= {audio_addr,
  audio_reg6[cmd_count], audio_cmd6[cmd_count]};
            endcase
```

2.4 分析/综合

分析/综合实验成功，如下图所示：

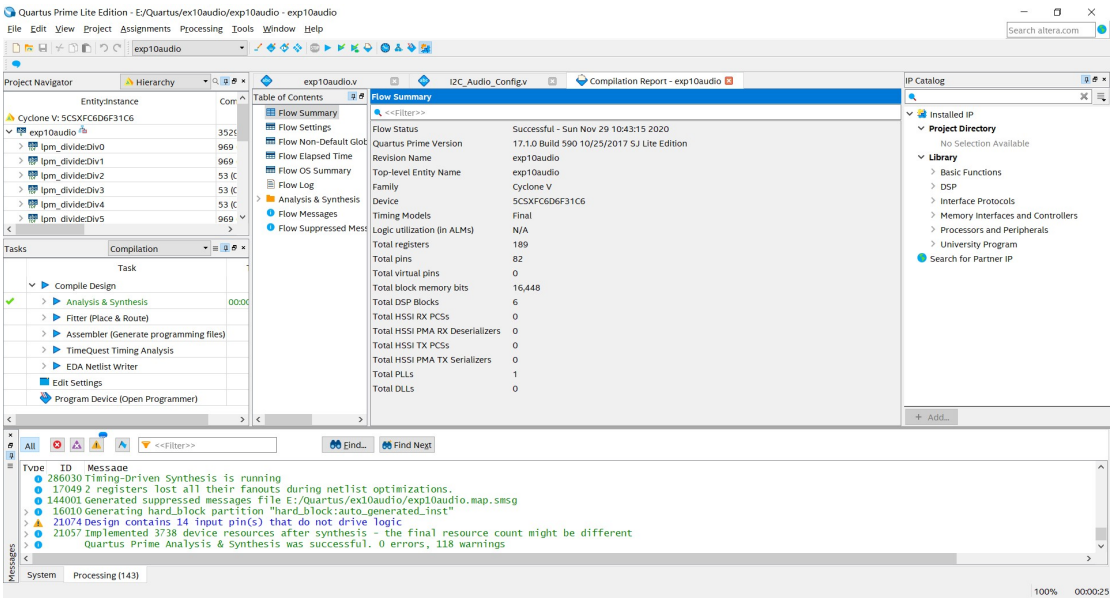


图 2-4-1：分析/综合成功

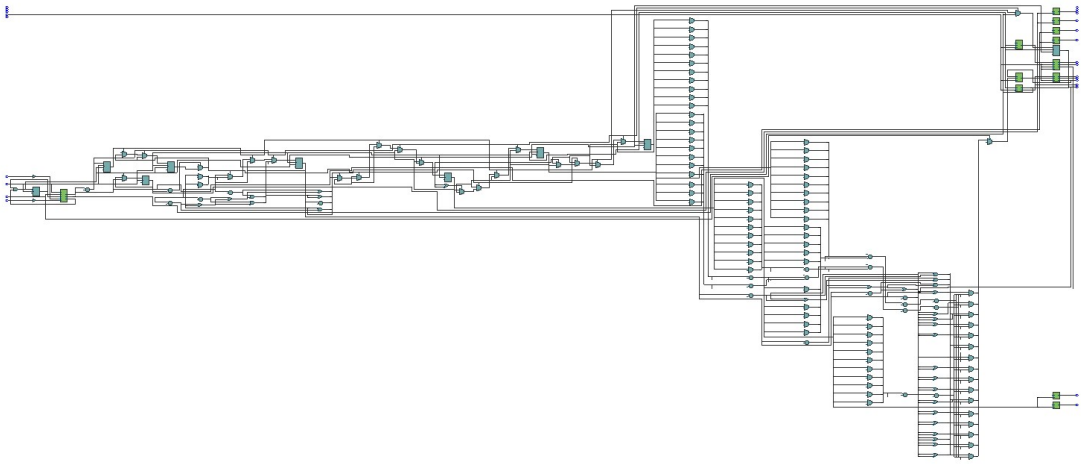
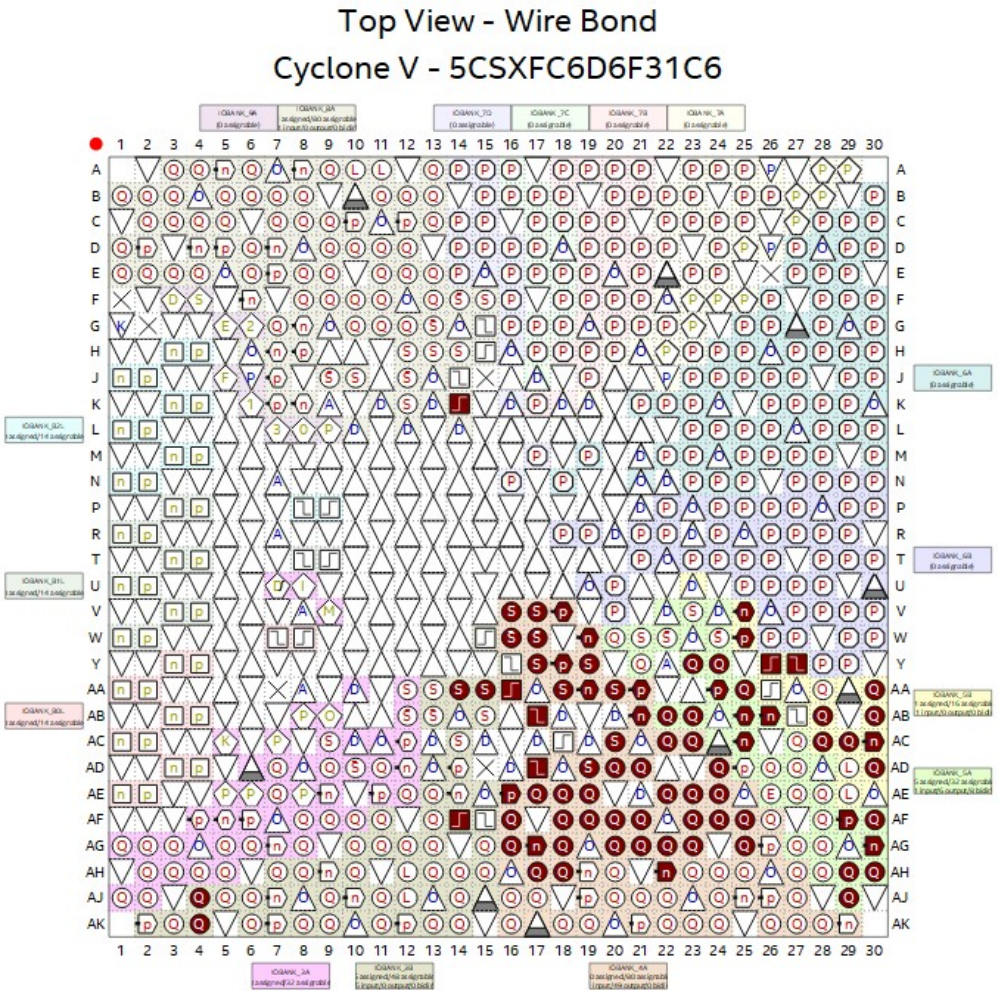


图 2-4-2:RTL 视图

2.5 分配引脚

引脚分配使用 DE10_Standard_SystemBuilder 生成。



AUD_ADCDAT	Input	PIN_AJ29	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)											
AUD_ADCLRCK	Bidir	PIN_AH29	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
AUD_BCLK	Bidir	PIN_AF30	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
AUD_DACDAT	Output	PIN_AF29	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
AUD_DACLRCK	Bidir	PIN_AG30	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
AUD_XCK	Output	PIN_AH30	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
CLOCK2_50	Input	PIN_AA16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)											
CLOCK3_50	Input	PIN_Y26	5B	B5B_N0	2.5 V		12mA ...ault)											
CLOCK4_50	Input	PIN_K14	8A	B8A_N0	3.3-V LVTTL		16mA ...ault)											
CLOCK_50	Input	PIN_AF14	3B	B3B_N0	3.3-V LVTTL		16mA ...ault)											
FPGA_I2C_SCLK	Output	PIN_Y24	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
FPGA_I2C_SDAT	Bidir	PIN_Y23	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[6]	Output	PIN_AH18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[5]	Output	PIN_AG18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[4]	Output	PIN_AH17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[3]	Output	PIN_AG16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[2]	Output	PIN_AG17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[1]	Output	PIN_V18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX0[0]	Output	PIN_W17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[6]	Output	PIN_V17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[5]	Output	PIN_AE17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[4]	Output	PIN_AE18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[3]	Output	PIN_AD17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[2]	Output	PIN_AE16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[1]	Output	PIN_V16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX1[0]	Output	PIN_AF16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[6]	Output	PIN_W16	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										

HEX2[5]	Output	PIN_AF18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[4]	Output	PIN_Y18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[3]	Output	PIN_Y17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[2]	Output	PIN_AA18	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[1]	Output	PIN_AB17	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX2[0]	Output	PIN_AA21	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[6]	Output	PIN_AD20	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[5]	Output	PIN_AA19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[4]	Output	PIN_AC20	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[3]	Output	PIN_AA20	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[2]	Output	PIN_AD19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[1]	Output	PIN_W19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX3[0]	Output	PIN_Y19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[6]	Output	PIN_AH22	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[5]	Output	PIN_AF23	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[4]	Output	PIN_AG23	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[3]	Output	PIN_AE23	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[2]	Output	PIN_AE22	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[1]	Output	PIN_AG22	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX4[0]	Output	PIN_AD21	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[6]	Output	PIN_AB21	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[5]	Output	PIN_AF19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[4]	Output	PIN_AE19	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[3]	Output	PIN_AG20	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[2]	Output	PIN_AF20	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[1]	Output	PIN_AG21	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
HEX5[0]	Output	PIN_AF21	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										

KEY[3]	Input	PIN_AA15	3B	B3B_N0	3.3-V LVTTL		16mA ...ault)											
KEY[2]	Input	PIN_AA14	3B	B3B_N0	3.3-V LVTTL		16mA ...ault)											
KEY[1]	Input	PIN_AK4	3B	B3B_N0	3.3-V LVTTL		16mA ...ault)											
KEY[0]	Input	PIN_AJ4	3B	B3B_N0	3.3-V LVTTL		16mA ...ault)											
LEDR[9]	Output	PIN_AC22	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[8]	Output	PIN_AB22	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[7]	Output	PIN_AF24	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[6]	Output	PIN_AE24	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[5]	Output	PIN_AF25	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[4]	Output	PIN_AG25	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[3]	Output	PIN_AD24	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[2]	Output	PIN_AC23	4A	B4A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[1]	Output	PIN_AB23	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
LEDR[0]	Output	PIN_AA24	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
PS2_CLK	Bidir	PIN_AB25	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
PS2_CLK2	Bidir	PIN_AC25	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
PS2_DAT	Bidir	PIN_AA25	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
PS2_DAT2	Bidir	PIN_AB26	5A	B5A_N0	3.3-V LVTTL		16mA ...ault)	1 (default)										
SW[9]	Input	PIN_AA30	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[8]	Input	PIN_AC29	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[7]	Input	PIN_AD30	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[6]	Input	PIN_AC28	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[5]	Input	PIN_V25	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[4]	Input	PIN_W25	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[3]	Input	PIN_AC30	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[2]	Input	PIN_AB28	5B	B5B_N0	2.5 V		12mA ...ault)											

SW[1]	Input	PIN_Y27	5B	B5B_N0	2.5 V		12mA ...ault)											
SW[0]	Input	PIN_AB30	5B	B5B_N0	2.5 V		12mA ...ault)											

图 2-5 引脚分配图

2.6 全编译

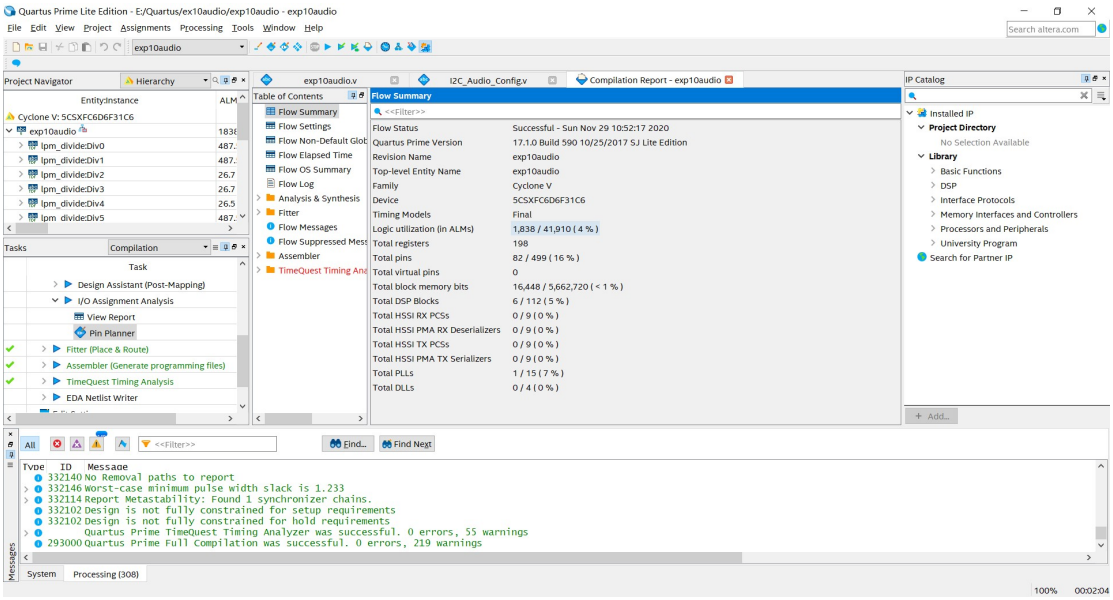


图 2-6 全编译成功

三、实验总结

本次实验主要学习了音频的输出原理，并在给出的参考数字信号到模拟信号的转换和 I2C 音频输出芯片的初始化代码的基础上设计了能够用键盘“弹奏”的电子琴. 实验中复习并运用了之前键盘实验中学习的状态机的设计来实现和声输出，同时注意到设计过程中应注意运算的溢出和精度损失问题. 在设计音量变化功能时进一步深刻了解的 RAM 存储器在读写时应当遵守的时序逻辑.