|  |  |
| --- | --- |
| logo[1] | 杭州电子科技大学 |

实验报告

Experimental report

|  |  |
| --- | --- |
| **实验课程名称** | **数字系统与处理器** |
| **实验序号** | 8 |
| **实验内容** | 乐曲演奏电路设计 |
| **班级** | 18184111 |
| **姓名** | 杨臻 |
| **学号** | 18052032 |
| **指导教师** | 黄继业 |

二○一九 年 十一月 二十八日

## 实验目的与要求

实验名称 乐曲演奏电路设计

实验目的 学习设计硬件乐曲演奏电路以及相关的控制电路

## 实验原理

硬件乐曲演奏电路顶层模块图。其中由6个子模块构成。与利用微处理器（CPU或MCU）来实现乐曲演奏相比，以纯硬件完成乐曲演奏电路的逻辑要复杂一些。本实验设计项目是乐曲演奏电路的实现。

组成乐曲的每个音符的发音频率值及其持续的时间是乐曲能连续演奏所需的两个基本要素。问题是如何来获取这两个要素所对应的数值以及通过纯硬件的手段来利用这些数值实现所希望乐曲的演奏效果。

## 实验内容

实验步骤：

1. 建立Vivado工程
2. 编写SPKER发声器模块
3. 编写F\_CODE译码电路模块
4. 编写分频器
5. 编写ROM控制模块
6. 编写主程序
7. 编写音乐coe文件
8. 配置IP核
9. 仿真
10. 烧写程序

关键源程序：

Main：

module main(CLK0,RST0,LED,SPK\_KX,H/\*,PM1,CNT\*/);

input CLK0,RST0;

output SPK\_KX;

output [3:0] LED;

output H;

//output PM1;

//output [7:0] CNT;

reg [7:0] CNT8;

wire [3:0] Q;

reg CLK,CLK1M;

wire [10:0] T;

reg [29:0] Q1,Q2;

wire SPK;

reg SPK\_KX;

always @(posedge CLK0 or negedge RST0) //分频器1

begin

if (!RST0)

begin

Q1<=0; CLK<=1'b0;

end

else

begin

if (Q1==0) CLK<=1'b0;

Q1<=Q1+1;

if (Q1==12499999)

begin

Q1<=0;

CLK<=1'b1;

end

end

end

//assign PM1=CLK;

always @(posedge CLK0 or negedge RST0) begin //分频器2

if (!RST0)

begin

Q2<=0;

CLK1M<=1'b0;

end

else

begin

Q2<=Q2+1;

if (Q2==24)

begin

CLK1M=~CLK1M;

Q2<=0;

end

end

end

wire LD;

always @(posedge CLK or negedge RST0) //ROM控制

begin

if (!RST0) CNT8<=8'b00000000;

else

begin

CNT8<=CNT8+1;

if (CNT8==26)

CNT8<=8'b00000000;

end

end

//assign CNT=CNT8;

dist\_mem\_gen\_0 d0( //读取ROM

.a(CNT8),

.clk(CLK),

.spo(Q)

);

F\_CODE FC( //译码器

.INX(Q),

.CODE(LED),

.TO(T),

.H(H)

);

SPKER SP( //蜂鸣器

.CLK(CLK1M),

.RST(RST0),

.TN(T),

.SPKS(SPK)

);

always @(posedge SPK or negedge RST0) //信号发送至SPAKER

begin

if (!RST0) SPK\_KX<=0;

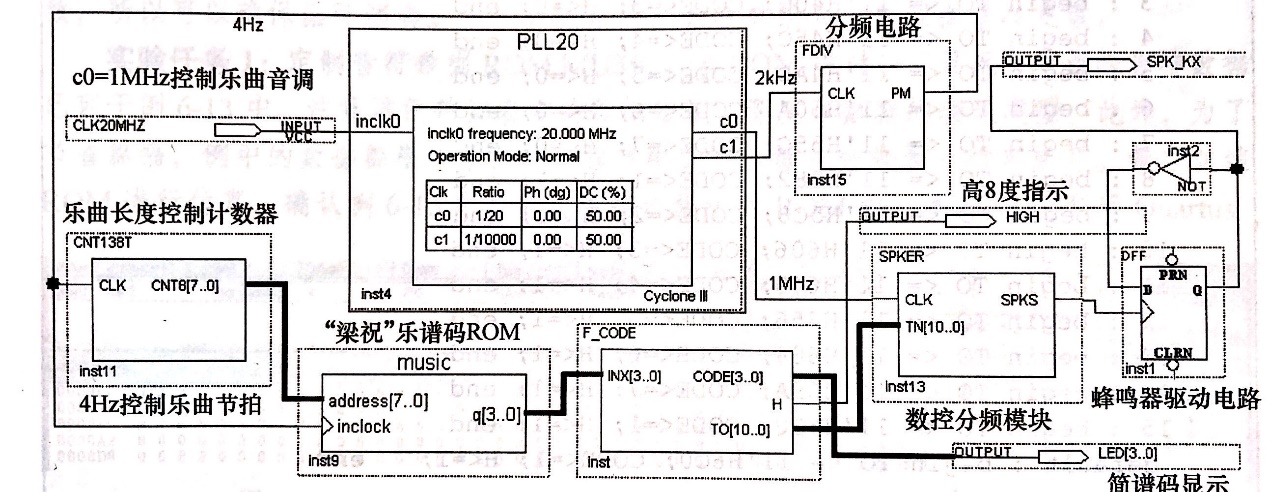
else

SPK\_KX<=~SPK\_KX;

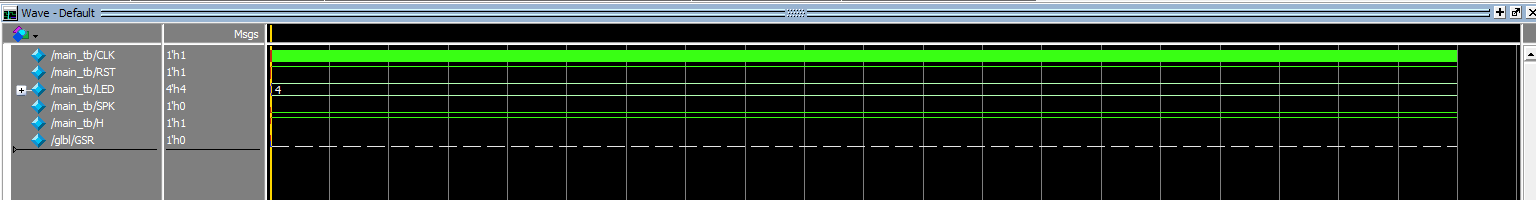
end

endmodule

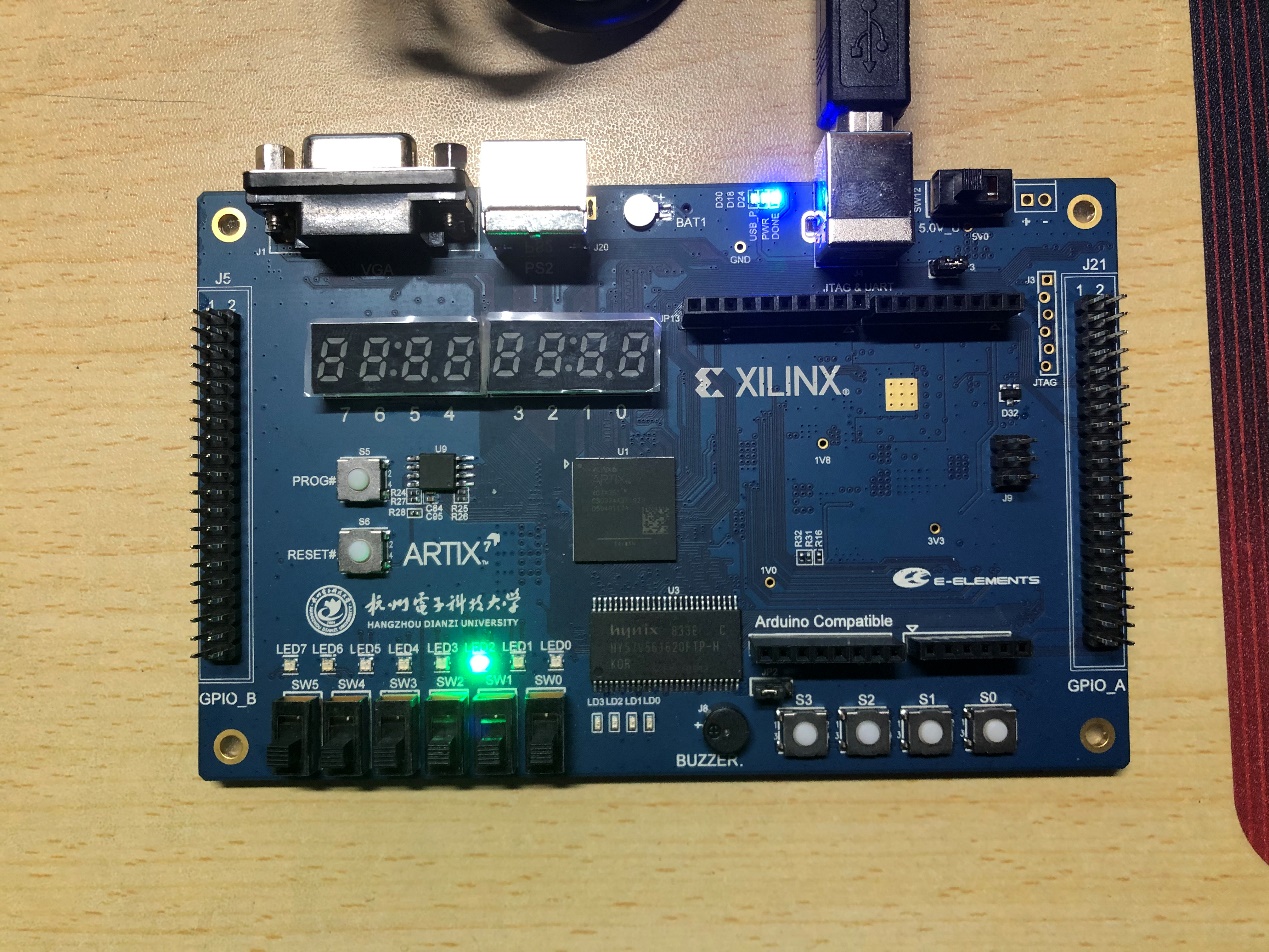
设计框图：



仿真波形：



硬件演示照片：



## 总结与体会

这次实验我成功完成了音乐播放电路的设计，中途遇到了许多没有遇到过的bug，最终还是圆满完成了。