

Digital Logic Project 项目文档

一、项目说明

本项目所有源代码以及说明文档均于GitHub中有所保存。

Github URL: <https://github.com/wzxawa/Digital-Logic-Project.git>

二、组织与计划

2.1 团队分工

吴鎏亿：设计了状态机，负责按键消抖模块、顶层模块以及控制模块的一部分，也负责主要的 debug

周子涵：设计了数码管输出的逻辑，控制功放喇叭的输出

王子旭：完成初始状态机模块（状态转换&自清洁模式），负责主要外接键盘bonus实

三人贡献比均为1/3

2.2 开发计划安排和实施情况

开发计划：选择15周进行答辩，在答辩 4 周前开始，整体按照先分工完成模块再整合debug，再冲击完成bonus

具体实施：

第一周：完成分工，分开完成自己部分module书写

第二周：整合各个模块，对状态进行进一步细化

第三周：进行过程中bug修复，完善细节，尝试完成bonus

第四周答辩前：完成外接输出设备bonus，对外接输入设备进行调试。

三、项目细节

3.1 项目简介

本项目是基于EGO1开发版，verilog语言以及vivado开发工具而实现的一个模拟抽油烟机控制电路的数字逻辑项目。该项目主要采用了模块化设计来实现各个功能，在输入端采用了按键消抖模块，同时在内存中设计了常量池模块，在输出部分有八位七段数码管输出模块和功放喇叭输出模块，最重要的控制逻辑全部集中在控制模块中，所有的模块都在一个顶层模块main.v中被调用。本项目的关键在

将右下角的五个按键分别称作 A,S,D,X,W(根据键盘上的位置定义)，最右边的两个拨码开关依次控制 P,light_dip。

- 输出包含：
1. **8位7段数码管**表示当前时间、状态等数据；
 2. **3个LED灯**分别表示“智能提醒”、开机、照明功能；
 3. **功放喇叭**表示状态转换和智能提醒。

关机状态：

在拨码开关P为开时，可以使用短按 X 键直接开机，或者也可以使用手势开关，先按下 A 键，在手势有效时间（默认 5 秒）内按下 D 键实现开机。开机自动进入待机模式，当前时间从 00:00:00 开始计时。

待机模式：

第一次进入待机模式，当前时间、工作时间均为 00:00:00。在油烟机常用模式间的转换中，拨码开关P 均置为关。

在拨码开关P 为关时，W 为菜单键，X 为 查询键。

若要进入工作模式，先按菜单键，后按下 A、S、D、X 分别进入 1档工作、2档工作、飓风模式、自清洁模式。

若要进入查询模式，按下 X 键即可。

工作模式：

在工作模式中，累计工作时间会不断累加，W 键表示返回。7段数码管上会显示当前时间、状态（分别使用 P1,P2,P3 表示）

1档、2档工作状态可自由切换，功放喇叭发出不同声音，可使用 W 键回到待机模式。

飓风模式下，每次开机只能进入一次，自动开始60s倒计时，结束后自动切换至 2 档工作状态。期间按下 W 键会进入强制待机模式，倒计时 60s，不能使用常规按键退出，结束后自动进入待机模式。

如果工作时长超过了智能提醒时间，则对应的警报灯开始闪烁，提醒需要清洁。

自清洁模式：

进入自清洁模式，开始 180s 倒计时，无法使用常规按键退出。自清洁结束回到待机模式，累计工作时间清空。

查询模式：

从待机模式按 查询键进入查询状态，状态栏显示 “FF” ，可按 W 键返回待机模式

分别按下 X、A、S、D 键分别查询 累计工作时间、当前时间、手势开关时间、智能提醒时间。进入具体的查询状态后，可使用 W 键返回查询状态；如果处于可修改的 当前时间、手势开关时间、智能提醒时间 查询状态，可按 X 键进入修改模式。

修改模式：

进入修改模式，7段数码管显示当前数值和状态，“S3 “、” S2”、“S1”分别表示正在修改时、分、秒，按 A 表示加1，按 D 表示减1，按 S 表示确认，进入下一个修改单元。如果已经修改完毕，则按 S 自动回到对应的查询状态。

手势有效时间从实际考虑，能设置的值为 1-59 s。特别的，智能提醒时间、手势有效时间**不能设置为0s**，这样的结果会显示 “EE” ERROR状态，且无法保存，保留原来的数值。

手动自清洁按钮：

在开机状态下，在拨码开关P 为开，按下 W 键，会将累计工作时间清空。

恢复出厂设置：

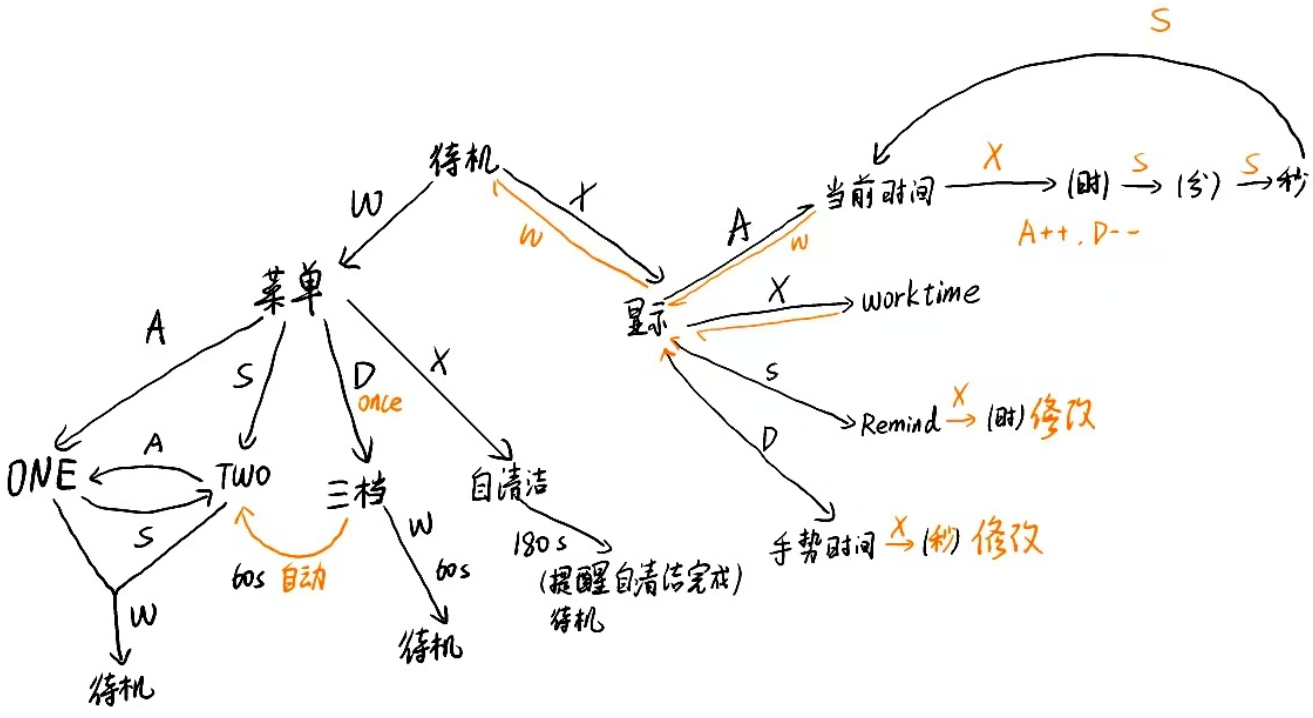
在开机状态下，在拨码开关P 为开，按下 W 键，会强制回到待机状态，当前时间、累计工作时间清空，手势有效时间、智能提醒时间恢复默认设置。

照明模式：

在开机状态，将拨码开关 light_dip 开启，则对应的LED灯亮，不受所有状态影响（除了关机）。

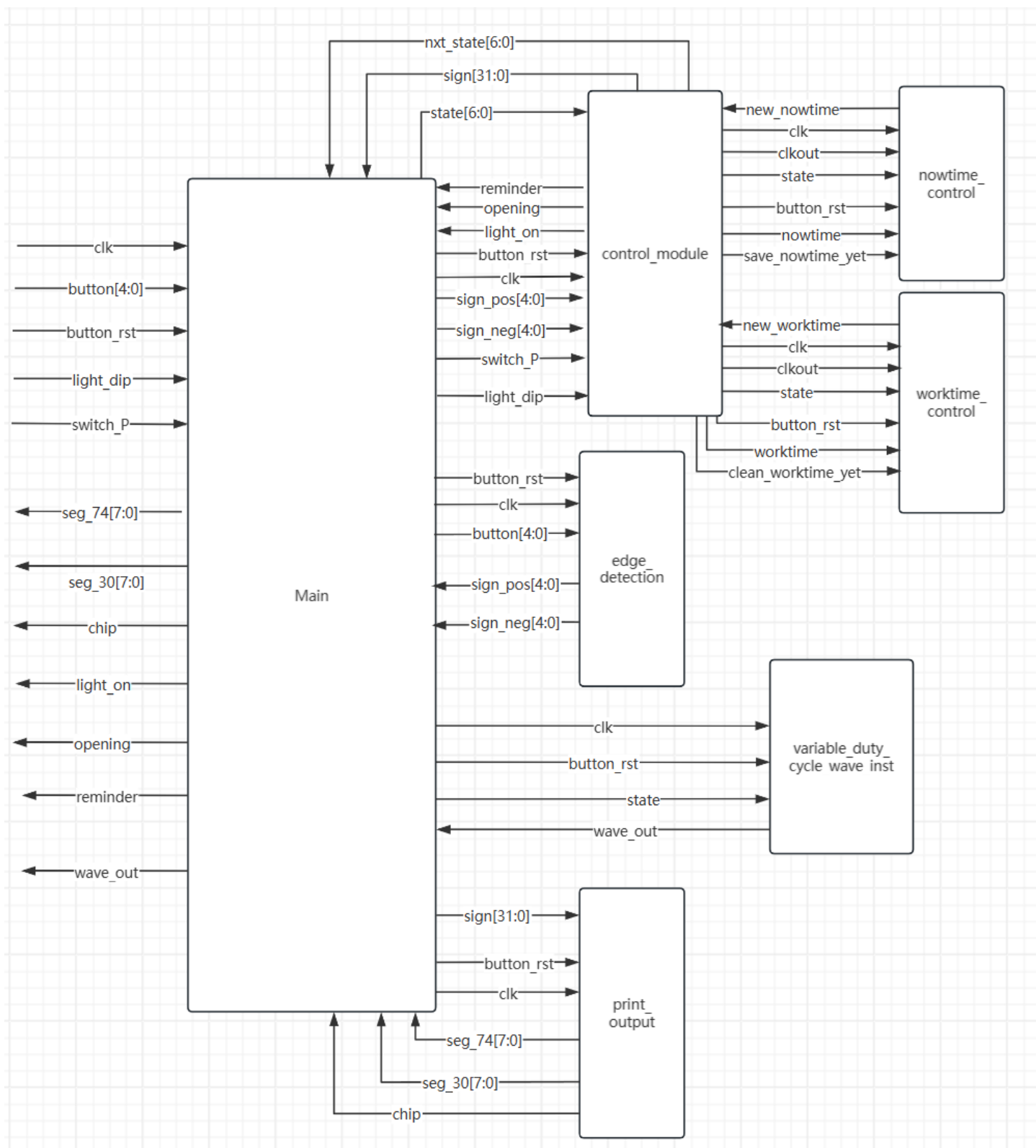
关机方法：

在开机任何状态下，拨码开关P 置为开，长按 X 键 3s 或者在手势有效时间内依次按下 D、A 键均可实现关机。



状态转换逻辑（不包括开关机）

3.4 系统结构说明



系统结构说明

3.5 子模块功能说明

0. 输入、输出端口说明（Main）

- button_A, button_S, button_D, button_W, button_X 分别为右侧5个按键开关
- switch_P: 用于控制状态机的状态转换, 和按钮结合分别表示不同的状态转换

- c. button_rst: 用于复位状态机
- d. clk: 时钟信号
- e. light_dip: 用于控制照明模式
- f. light_on: 表示照明模式是否开启
- g. chip: 控制7段数码管的片选信号
- h. seg_74: 用于控制7段数码管的7-4位数字的段选信号
- i. seg_30: 用于控制7段数码管的3-0位数字的段选信号
- j. opening: 用于控制油烟机开机状态的信号
- k. reminder: 用于控制提醒功能的信号
- l. wave_out: 用于控制功放喇叭的输出

功能描述:Main模块是整个系统的主模块,控制输入信号传入、处理,对状态机和模式进行转换,控制输出信号的输出,实现了整个系统的功能

1.防抖模块(edge_detection)

输入 button_A、button_S,以及DWX,还有clk,button_rst 等按键,输出对应的信号 sign_A,sign_S 等6个信号。

2.控制模块(contaol_module)

接收当前state,并主要根据当前的 state 和用户输入的经过消抖后的信号判断输出的模式和状态的切换,最后联系到显示输出模块,声音模块和更新主模块中 state 的值

3.时间模块(nowtime_control & worktime_control)

传入当前的 state、nowtime (worktime)、clk、clkout 等参数,使用 new_nowtime`、new_worktime 进行修改。将时间模块从控制模块中分离出来,提高系统的模块化程度,使得时间相关的功能更加集中和易于管理。

4.显示输出模块(print_out)

根据 输入的 state,控制 on_off、light 两个灯,以及根据 sign 中的 8 位信号 (32 位二进制),分别对应 7段数码管上的8位。设定了一定的周期使得 8个7段数码管轮流显示数字,利用人眼视觉残留实现视觉上的同时发光。

Reference: [七段数码管26字母对照表\(附带映射表以及映射数组\)_七段数码管显示字母对应表-CSDN博客](#)

5.声音输出模块(variable_duty_cycle_wave_inst)

主要利用了脉冲宽度调制信号。脉冲宽度调制信号是一连串频率固定的脉冲信号,每个脉冲的宽度都可能不同。这种数字信号在通过一个简单的低通滤波器后,被转化为模拟电压信号,电压的大小

跟一定区间内的平均脉冲宽度成正比。这个区间由低通滤波器的3dB截止频率和脉冲频率共同决定。根据输入的 `state` ，在 `state` 切换的时候播放不同频率，不同时长的声音。输出设备为功放喇叭。

6.PARAMETER 文件

存放各种常数，包含状态、时间、初始值等常数，使代码更规范。

3.6 bonus 的实现

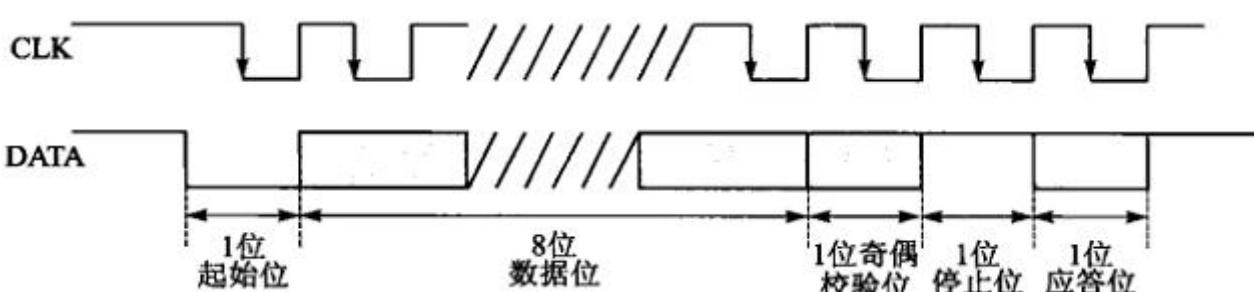
1.手势开关以及手势开关倒计时：

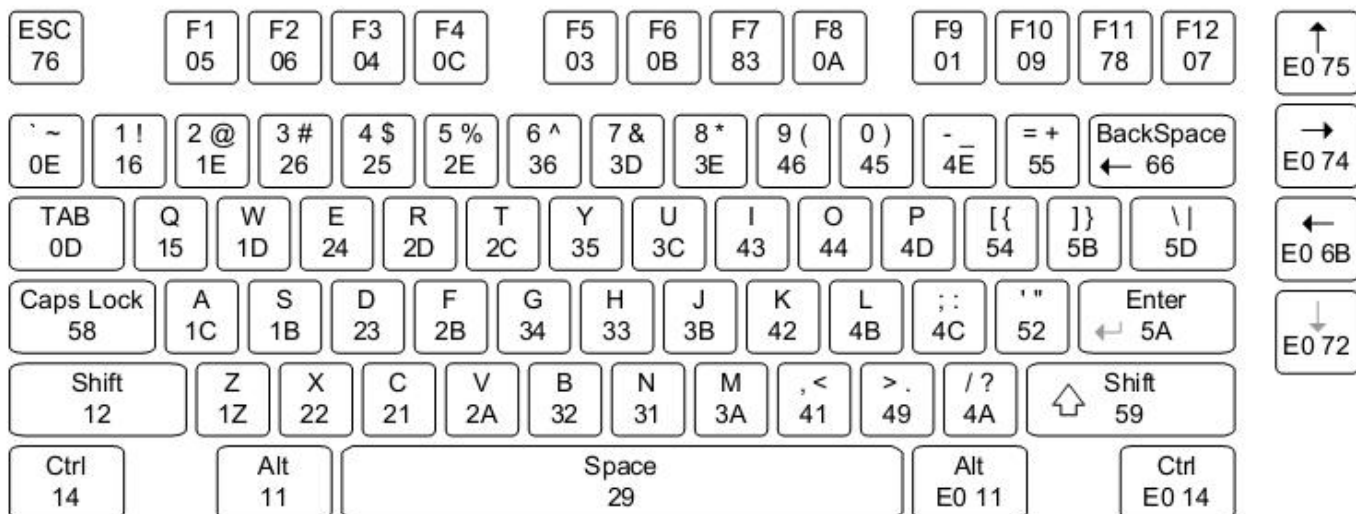
本项目实现了手势开关，且在手势开关的关机等待时间内可以随时退出等待并且该过程不影响系统的正常运行。具体为拨码开关P 为开时，可以使用依次按下 D、A 键进行关机，在有效时间内，7段数码管会显示倒计时，若将拨码开关P 置为关，则会显示油烟机目前的状态，二者互不影响。具体的实现是引入了计时系统，通过对两个信号的先后检测决定状态是否发生改变，总体来说就是信号输入的检测与等待时间的设置最终实现了手势开关这一功能。

2.额外输入设备：

实现ps2_keyboard.v 模块完成，逻辑上成功实现键盘按下模拟lego1开发板上按键开关按下，但接入顶层设计进行测试时发现：由于ps2协议信号较老，目前很多键盘不支持，同时ps2协议传输不稳定，与键盘规格有关。具体模块内容可见github-src-to complete部分。

模块内部逻辑顺序为：

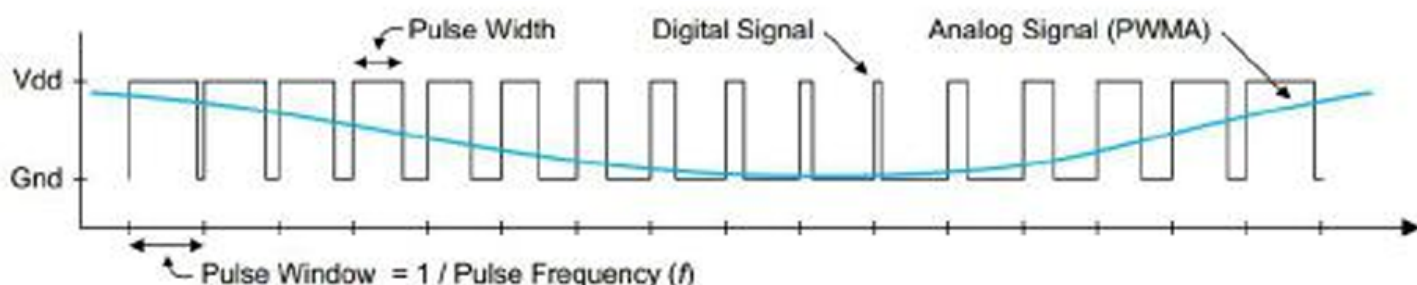
- 0.输入为系统时钟、系统rst、键盘时钟、键盘信号传输data
 - 1.对于键盘时钟信号进行消抖
 - 2.在键盘时钟下降沿捕获输入信号data
 - 3.连续读取11位数据，其中前8位为数据位
- 
- 4.检测是否为“f0”（松开按键前信号），做出区分
- 5.使用状态机将键盘读取8位数据位信号根据键盘扫描码转化按键信号



6.输出为模拟按键信号

3.额外输出设备：

本项目实现了功放喇叭不同频率，不同时长的输出，声音在模式切换时会输出，以及在智能提醒状态会持续输出。具体的实现流程是使用了脉冲宽度调制：脉冲宽度调制信号是一连串频率固定的脉冲信号，每个脉冲的宽度都可能不同。这种数字信号在通过一个简单的低通滤波器后，被转化为模拟电压信号，电压的大小跟一定区间内的平均脉冲宽度成正比。这个区间由低通滤波器的3dB 截止频率和脉冲频率共同决定。例如，脉冲为高电平的时间占有效脉冲周期的 10%的话，滤波电路产生的模拟电压值就是Vdd电压的十分之一。



通过不断尝试，最后设置了100个Pulse Windows，步长为0.05的小周期长度变化幅度以实现了声音的最佳采样频率，达到了一个合适的强度，有通过将传入的系统时钟进行变频已实现不同的声音频率，最后传入state参数，通过检测state的变化以及变化后state的值播放不同的声音。总体来说，流程就是通过传入的时钟频率结合脉冲宽度的调制生成不同形态的数字信号，最后再根据系统状态实现输出。

四、项目总结

4.1 项目遇到的问题 and 解决

1.项目初期，我们对整体模块化缺乏清晰的思路。为了解决这个问题，我们采取了绘制流程图、细分功能模块、大量编写代码以及使用testbench进行测试的方法。通过这些步骤，我们逐渐理清了思路

并解决了问题；

2.multi-driver是一个很严重的critical warning。我们通过重新设计信号连接和逻辑，确保了信号的单一驱动，从而解决了这一问题。

4.2 本项目的提升空间：

- 1.能够再使用其他键盘进行测试，彻底完成bonus外接输入设备部分；
- 2.能够对于模块进一步细化，有助于提升代码可读性和维护能力；
- 3.我们计划对声音效果进行优化，使其更加规范和悦耳。

4.3 团队合作、开发及测试工作的总结

本次项目对于我们每一个队员来说都是很有意义的，首先我们深刻的认识到了verilog语言与其它语言的不同以及它自身运行的一些特性（如multi-drive），也认识到了开发板上的不同接口与不同功能，体会到了如何通过编程语言和模块化设计达成一个电路逻辑以及其中的常见的隔阂。同时我们也尝试很多途径去debug，体会到了找问题在真实工程中的艰难。最后也是感谢每位组员高效，成效地完成了各自的分工和一起解决项目难题，以实现这次Project的提前答辩与较好的展示效果。

4.4 基于 EGO1 的项目构想

- a. **Bouns plus**：使用**蓝牙**或者**VGA**进行bonus输入以及输出。
- b. **增加一个使用密码开机的功能**：通过按键和拨码开关，输入密码，和默认密码（预设密码）验证后，7段数码管上显示结果，同时有LED灯指示。
- c. **智能交通信号灯控制系统**：设计一个模拟真实交通信号灯的控制系统，包括红、黄、绿灯的控制逻辑，以及车辆计数和行人过街按钮，每种情况都可以设置状态进行切换。时钟功能可以用来控制信号灯的切换周期。
- d. **病房呼叫系统**：设计一个病房呼叫系统，当病人紧急呼叫时，系统产生声音提示，并显示病人的编号，还可以查看和修改不同病人的优先级和倒计时。
- e. **背单词机**：可以利用7段数码管设置单词（数字,或者 A,E,F,H 等可以表示的），设置不同的学习模式，实现翻页、多次回答正确后删除、学习时间提醒等功能。