

燕山大学  
EDA 课程设计报告

题目： 三十四、游戏机

姓名： 田博松

班级： 电子科学与技术 21-2 班

学号： 202111040246

成绩：

## 一、设计要求

用三位数码管显示 0-7 之间的数码，按下按钮，三个数码管循环显示，抬起按钮，显示停止，当显示内容相同时为赢。

1. 三个数码管循环显示的速度不同；
2. 停止时的延迟时间也要不同；
3. \*如果赢了游戏时，要有数码管或 LED 的花样显示或声音提示；
4. 实现作弊功能。当按下作弊按钮时，最后总会显示的三个数字总会相同；
5. 七段数码管 DIG4-DIG7 当游戏进行时显示 GAME 字样，当游戏结束时显示 OVER 字样；
6. 游戏结束时，可以按住某个按钮使三位数码管重新转动起来。

## 二、设计过程及内容

### 1. 模块划分

整体设计分别由分频模块、延时模块、数字驱动模块、比较模块、作弊模块、扫描模块、提示模块和重置模块组成，下面是这几个模块的详细介绍。

#### (1) 分频模块

通过两个分频模块，最终得到 5HZ、2.5HZ、1HZ 三个频率。

其中第一个分频模块由老师给出，得到 1KHZ、10HZ、1HZ 三个后续用到的模块。其中 1KHZ 的频率为后续的数码管扫描电路提供扫描频率。10HZ 可以由第二个分频模块分频得到 2.5HZ 和 5HZ 两个频率。这样便得到三个数码管分别的驱动频率 10HZ、5HZ、1HZ，达到三个数码管循环显示速度不同，停止时延迟时间也不同。

其中，第一个分频模块由老师用 VHDL 语言给出，第二个分频模块由 74161 十六进制计数器组成，由此计数器可以得到 1/2 分频、1/4 分频、1/8 分频、1/16 分频。

#### (2) 延时模块

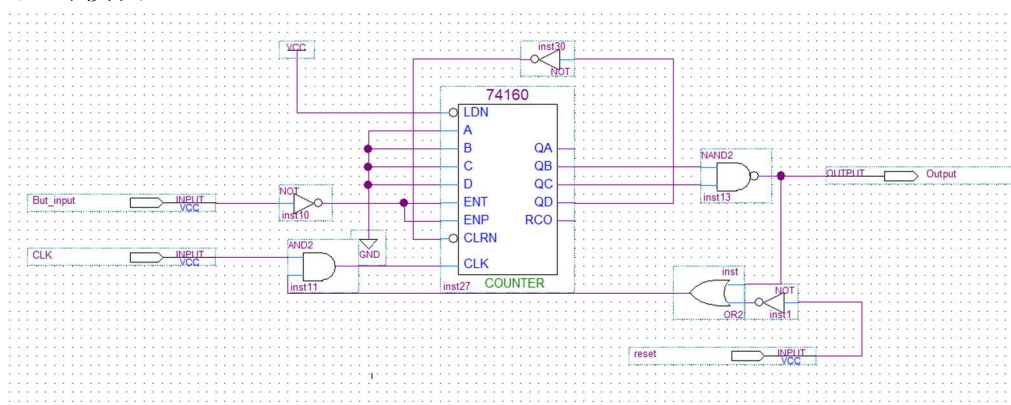


图 1 延时模块

当 But\_input 为 1 时，即按钮未按下时，74160 十进制计数器并未开始工作，此时 QB、QC 默认输出为 0，经过与非门在 Output 处输出为 1，驱动后续八进制计数器工作；当 But\_input 为 0 时，即按钮按下时，74169 的 ENT 和 ENP 开始置 1，此时 74160 作为延时模块开始工作，

刚开始工作时 74160 的输出为 0000，当 74160 的输出由 0101 变为 0110 时，对后续八进制计数器的驱动开始停止，由于三个延时模块的 CLK 输入频率不同，所以对驱动八进制计数器的停止时间，即延时时间，便不同。

### （3） 数字驱动模块

通过 74160 十进制计数器设计成八进制计数器，但是 CLK 输入由一个与门控制，即 CLK 输入和延时模块的输入，当延时模块未工作时输出 1，此时八进制计数器正常工作，当延时模块工作时输出 0，此时三个八进制计数器由于上一级的延时模块不同，CLK 被逐渐置 0，停止工作，此时便实现三个数码管循环显示速度不同，停止时延迟时间也不同。

### （4） 比较模块

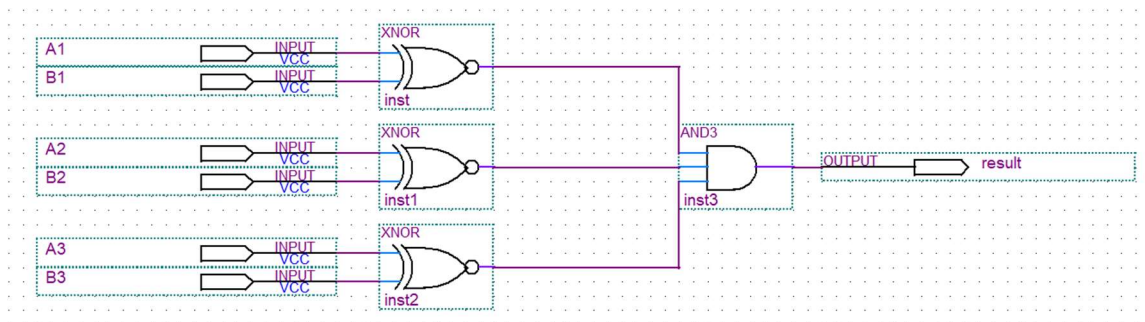


图 2 比较模块

由于需要比较的数字为 0-7，故二进制最多为 3 位，所以在比较模块中加入三个同或门，在三个同或门后面加入一个与门，即三个数字相同时才会输出为 1。

### （5） 作弊模块

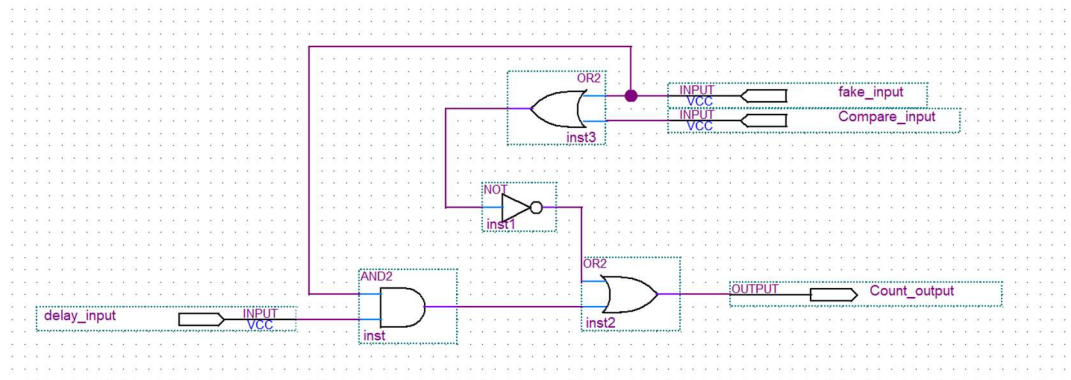


图 3 作弊模块

此作弊模块有三个输入，一个输出。分别是作弊按钮输入、比较输入、延时模块输入，作弊输出。当作弊按钮未按下时，fake\_input 为 1，即 inst 的某个输入为 1，inst2 的某个输入为 0，此时 Count\_output 输出与 delay\_input 一致，即此时作弊模块未生效。当作弊按钮按下时，此时 fake\_input 输入 0，在 AND2 中阻断了延时模块的输入，此时 Count\_output 仅由 Compare\_input 决定，当比较器发现三位数字中十位与百位数字不同时，便会继续驱动数字驱动模块，使其继续工作，当十位或者百位的驱动模块输出的数字与个位相同时，作弊模块才会停止驱动数字驱动模块。

### (6) 扫描模块

老师给出的扫描电路仅能输出 0-9，经过对模块内 VHDL 语言的修改。在输入 0-9 时还是正常发输出数字信息，当输入为 10-35 时，即可扫描输出 A-Z 共 26 个英文字母。

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity binary_output is
6  port (
7      input : in std_logic; -- 输入信号
8      output1 : out std_logic_vector(5 downto 0); -- 输出1
9      output2 : out std_logic_vector(5 downto 0); -- 输出2
10     output3 : out std_logic_vector(5 downto 0); -- 输出3
11     output4 : out std_logic_vector(5 downto 0) -- 输出4
12 );
13 end binary_output;
14
15 architecture behav of binary_output is
16 begin
17     process (input)
18     begin
19         if input = '0' then -- 如果输入为1
20             output1 <= "010000"; -- 输出1为16
21             output2 <= "001010"; -- 输出2为10
22             output3 <= "010110"; -- 输出3为22
23             output4 <= "001110"; -- 输出4为14
24         else -- 如果输入为0或其他值
25             output1 <= "011000"; -- 输出1为24
26             output2 <= "011111"; -- 输出2为31
27             output3 <= "001110"; -- 输出3为14
28             output4 <= "011011"; -- 输出4为27
29         end if;
30     end process;
31 end behav;
```

图 4 GMAE\_OVER 输出模块

通过与扫描模块的配合，当按下或抬起游戏按钮时，通过给七段数码管的 DIN4-DIN7 输入不同的数字，经由扫描电路在七段数码管的后四位输出 GAME 或者 OVER 字样。

### (7) 提示模块

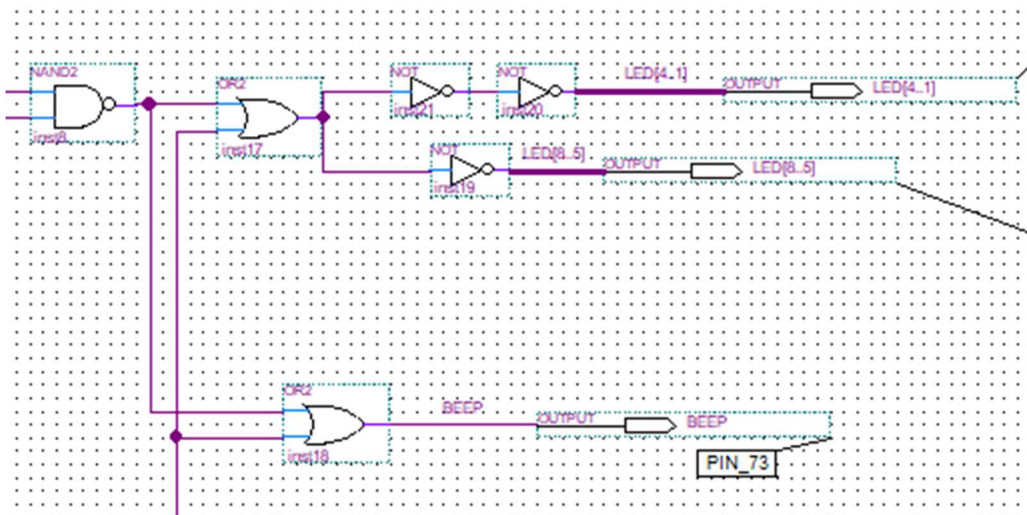


图 5 提示模块

蜂鸣器和 LED 由 1HZ 时钟和比较模块的输出共同控制，当游戏机三个数字相同时，蜂鸣器与 LED 便会以 1HZ 的频率变换状态。

### (8) 重置模块

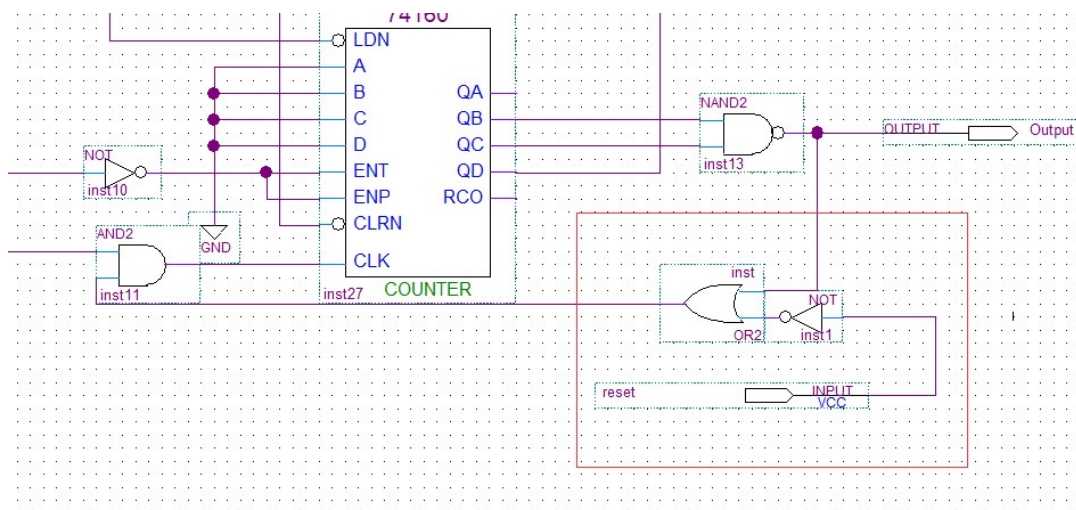


图 6 重置模块

将重置按钮至于延时模块中，当延时模块正常工作时，重置按钮并不会影响延时模块，当延时完毕后，QD-QA 的输出将停在 0110，当外部 RESET 按钮按下时，将带动延时模块的再次工作，适时停止，此时再次驱动数字驱动模块继续工作

## 2. 主要模块设计与仿真

### (1) 延时模块/重置模块

延时模块的设计说明已在模块划分中进行了详细介绍，这里仅对其进行相应的仿真。由于重置按钮设计在了延时模块其中，这里捎带对其进行仿真。

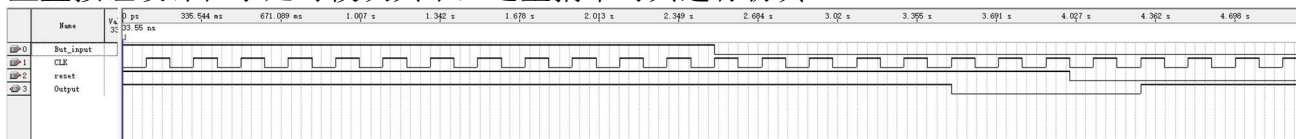


图 7 延时模块/重置模块仿真

由仿真图可得知，在 Reset 按钮未按下时，即 Reset 为 1 时，But\_input 按下。即 But\_input 为 0，此时 Output 经过一段时间置 0，此时将不会驱动后面的数字驱动模块，故七段数码管将逐个停止；当 Reset 按钮按下时，即 Reset 为 0 时，此时 Output 重新置 1，将继续驱动七段数码管转动。

### (2) 作弊模块

由于前面已经在设计部分对作弊模块做了详细的介绍，这里不再赘述，仅对作弊模块作一定仿真。

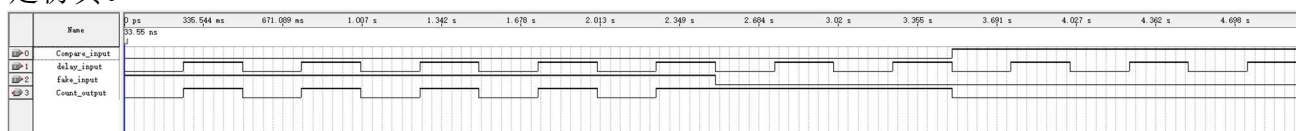


图 8 作弊模块仿真

当 Compare\_input 为 0 时，此时输出数字不一致，当 fake\_input 未按下时，由仿真图可知，Count\_output 与 dealt\_input 数据一致，此时作弊电路不对延时模块和数字驱动模块之间的信号传输造成影响；当 fake\_input 按下时，作弊模块阻断其之间的通信，使 Count\_input 输出为



1, 继续驱动数字驱动模块转动, 直到 Compare\_input 输入为 1 时, 此时数码管两个位置上的数字相同, 停止驱动数字驱动数码管, 最终显示三个数字相同的数字, 实现作弊效果。

### (3) 数字驱动模块

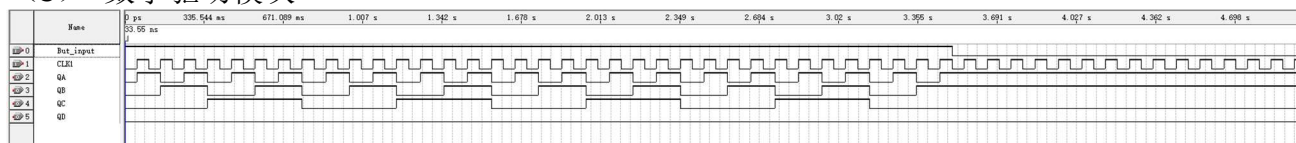


图 9 数字驱动模块仿真

当 But\_input 为 1 时, CLK 时钟可以正常驱动这个八进制的数字驱动模块, 当 But\_input 为 0 时, 数字驱动模块停止, 且停止时输出的二进制数字随机。

### 3. 总图设计与仿真

总体设计分别由分频模块、延时模块、数字驱动模块、比较模块、作弊模块、扫描模块、提示模块和重置模块组成。

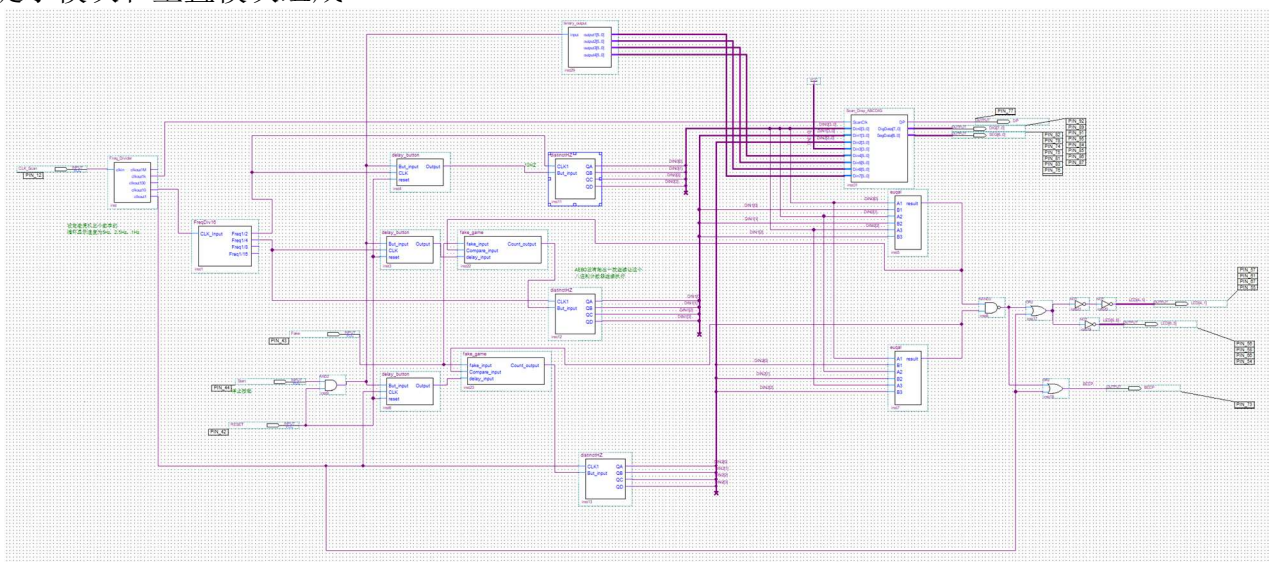


图 10 设计总图

开发板上的晶振提供 50MHZ 频率, 经分频模块后得到扫描模块使用的 1KHZ 频率及三个数码管分别使用的 5HZ、2.5HZ、1HZ 三个频率。当未按下 S4 时, 三个数码管由三个不同频率驱动的数字驱动模块管理; 当按下 S4 时, 通过延时模块作用三个数码管逐渐停止。通过比较模块, 当三个数字相同时便会触发提示模块使其发出提示。而当按下 S5 时, 作弊模块会阻断延时模块对数字驱动模块的驱动, 转而由自己本身结合比较模块的输出进行判断去驱动数字驱动模块。当按下 S6 时, 重置按钮将打破延时模块此时延时条件, 使延时模块重新驱动数字驱动模块, 使三个数码管重新转动起来。

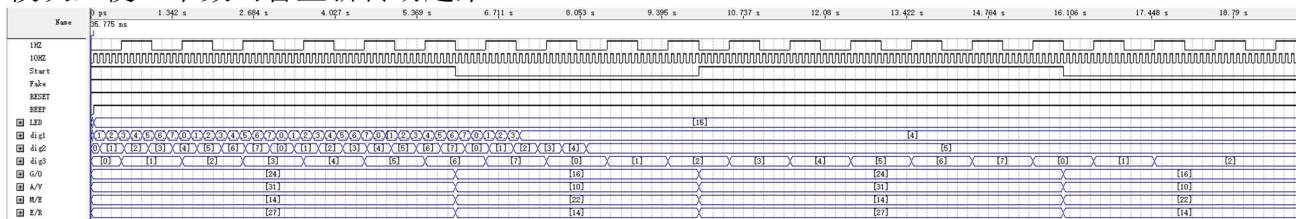


图 11 总图仿真 (仅有 S4 按钮在参与)

由仿真图可得，当按下按钮 S4 时，三个数码管最终停在了 254。且在按下时，DIN7-DIN4 输出 14、22、10、16，未按下时，输出 27、14、31、24，经由扫描模块扫描以后，输出 GAME 和 OVER 字样。

图 11 总图仿真（仅有 S4 按钮和 S5 按钮在参与）

由仿真图可知，当按下 S4 和 S5 时，十位和百位的数码管逐渐向个位上的数码管看齐。最后三个数码管显示的数字为 444，实现作弊功能。

### 三、设计结论

在设计过程当中，对于延时模块的设计是本设计中最值得思考的一个问题。要如何设计，才能使三个数码管的停止时间不同，最后我考虑到由于三个数字驱动模块所被驱动的频率的不同，所以延时时间的不同也要依赖这三个不同的频率，所以最后我想到使用 **74160** 十进制计数器去设计延时模块。

其次在进行总体仿真时，运行时间过长以至于不能运行出仿真结果。经由老师指导，是由于系统的晶振频率过高，相对于我的设计需要的频率形成数量级上差别，以至于长时间运行不能得出结果。当把分频模块去掉后，仿真时仅添加自己设计中需要的频率，便能在短时间内得到仿真结果。

在本次 EDA 课程设计中，我完成了一款游戏机的设计。在这个过程中，我学到了很多知识和技能，也遇到了一些困难和问题，但最终我成功地实现了我的设计目标，并且获得了很大的成就感和满足感。

感谢老师的指导和帮助，他们给了我很多有用的建议和资源，让我能够顺利地完成我的设计。他们不仅教会了我如何使用 EDA 工具，如 VHDL、Quartus II，还教会了我如何进行调试和测试，以及如何优化我的设计性能和效率。

我认为 EDA 课程设计是一门非常有价值和有意义的课程，它不仅让我掌握了 EDA 的理论和实践，还让我培养了分析问题、解决问题、创新思维等方面的能力。我觉得这些能力对于我的未来学习和工作都非常重要和有用。我建议老师可以在课程中增加一些更多样化和更具挑战性的设计题目，让学生可以有更多的选择和更大的空间来展示他们的才华和潜力。

#### 四、成绩单（可选）

设计过程与平时 表现（最多 10%）	验收答辩 （最多 90%）	报告撰写 (最多 10%)	总成绩	备注

（注：不够可加页）