

# 目录

1 摘要介绍.....	1
2 程序打开方法.....	2
3. 原理讲解.....	4

## 1 摘要介绍

这是我参加学校的首届大学生电子设计竞赛时编写的程序，当时我不具备足够的专业知识去做其他题目，甚至不会选购供其他题目使用的模块。幸好给出的六个题目的 F 题：自拟题目，给了我发挥空间，让拖延成症的我能及时把心里的想法做出来。它实现的功能类似于图 1.1 的老式数字钟。



图1.1 老式数字钟

我试图构建一个仅由数字电路实现的电子手表，这种电子手表很过时以至于零零后可能都没见过，这种数字钟仅由两个按键查看和调节，①多次按下 A 键可以分别切换至调节月、日、时、分的界面，此时按 B 键可以对相应值加一调节；②多次按下 B 键可以分别切换至查看月日和调节秒值的界面，调节秒值页面按 A 键会使秒针清零。

我最初的想法可能来自于《数字电子技术基础（第五版）》（高等教育出版社）第 323 页的例 6.4.3，这是一个利用米利型有限状态机来实现售货饮料机的数字电路。本程序也是使用有限状态机来控制数字钟的各个界面，同时由于 FPGA 实现时各个模块的并行性，在其他界面时并不会影响数字钟时间前进和进位进程，各个界面是互不影响的。

虽然这个数字钟很 LOW，但老师却给了我一个不错的名次，我很惭愧因为有些人做的比我好却没有我名次高。

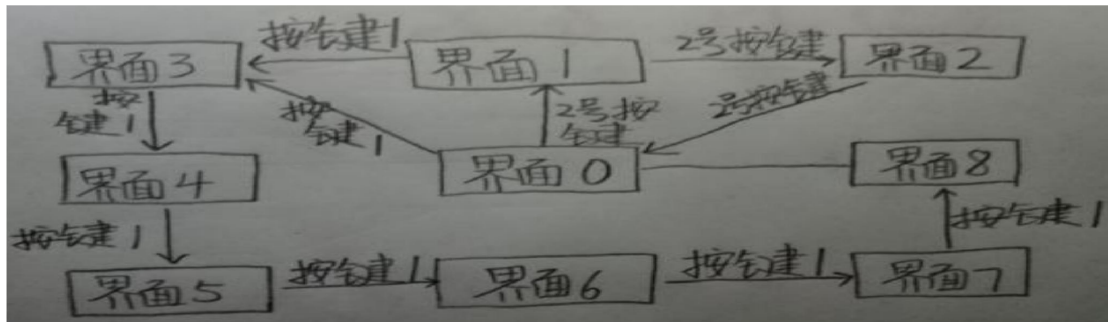


图1.2 数字钟状态转换图与操作方法

## 2 程序打开方法

我知道这个作品对于读者来说很简单，但是由于内容比较乱且 1000 多行代码分布在十多个文件之中，所以程序打开方法还是应该介绍一下的。

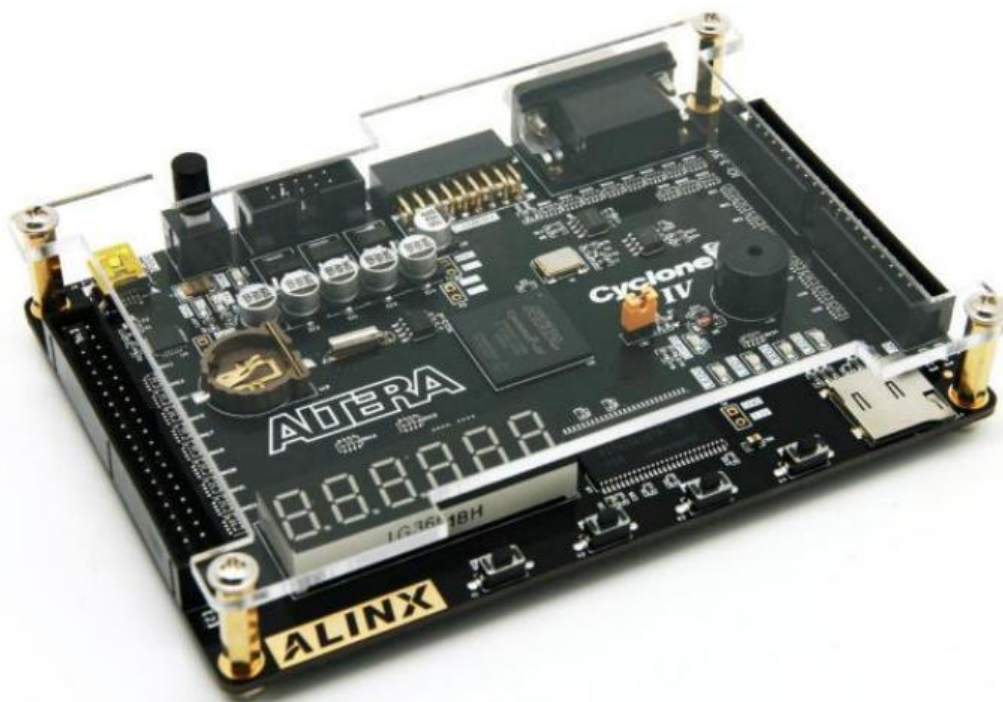
首先要确保你安装的 QuartusII 软件是 12.1 版本或以上，因为 QuartusII 是向下兼容的，高版本可以打开低版本设计的工程，而低版本打开高版本设计时却可能出现错误。其次是安装的芯片包中要有 Altera 公司 Cyclone IV E 系列的 EP4CE6F17C8，否则会出现无法综合的问题。最后请使用正版付费 Quartus 软件或使用它的免费破解版。

The screenshot shows the GitHub repository page for 'lu1198373615 / myDigitalClock'. The repository description is 'A work design and burn by myself when I was a junior at college for a competition of our school.' The page shows 2 commits, 1 branch, and 0 releases. A red box highlights the 'Clone or download' button, which is labeled '2'. Below the button, the 'Clone with SSH' option is selected, showing the URL 'git@github.com:lu1198373615/myDigitalClock'. Below the repository page, a terminal window shows the command to clone the repository:

```
luxy@user MINGW64 ~
$ cd f:

luxy@user MINGW64 /f
$ git clone git@github.com:lu1198373615/myDigitalClock
Cloning into 'myDigitalClock'...
remote: Enumerating objects: 61, done.
remote: Counting objects: 100% (61/61), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 61 (delta 32), reused 58 (delta 32), pack-reused 0
Receiving objects: 100% (61/61), 28.82 KiB | 355.00 KiB/s, done.
Resolving deltas: 100% (32/32), done.
```

确保你的 QuartusII 是可用的后就可以在 Git Bash 的合适路径下输入 `git clone git@github.com:lu1198373615/myDigitalClock.git` 将本工程连同 git 一起复制在所在目录。双击目录里的 myDigitalClock.qpf 文件就可以使用 QuartusII 软件打开本设计啦。顶层实体（原文是 Toplever Entity）是原理图，其他的都是 VHDL 语言文件。



如果不出意外，它是可以成功编译并综合的，之后应该下载在上图的 FPGA 开发板中，这是在 ALINX 淘宝店买的。链接：

[https://detail.tmall.com/item.htm?id=595363916425&ali\\_refid=a3\\_430673\\_1006:1226490189:N:7UlarNbj5V51gqen+Tan6raSb5ybJ2g8:c3ffa8b4f56ac08390bbab394e904be8&ali\\_trackid=1\\_c3ffa8b4f56ac08390bbab394e904be8&spm=a2e15.8261149.07626516002.1](https://detail.tmall.com/item.htm?id=595363916425&ali_refid=a3_430673_1006:1226490189:N:7UlarNbj5V51gqen+Tan6raSb5ybJ2g8:c3ffa8b4f56ac08390bbab394e904be8&ali_trackid=1_c3ffa8b4f56ac08390bbab394e904be8&spm=a2e15.8261149.07626516002.1)

下载后就可以试验程序啦！

界面 0 显示：最左端数码管不显示，左起第二第三位为时针，第四位 G 管闪烁，第五第六位为分针。此时按下 key1 进入界面 3，按下 key2 进入界面 1。

界面 1 显示：最左端数码管不显示，左起第二第三位为月，第四位 G 管亮，第五第六位显示日。此时按下 key1 进入界面 3，按下 key2 进入界面 2。

界面 2 显示：左起第一第二第三位不显示，第四位 G 管闪烁，第五第六位显示秒。此时按下 key1 给秒清零，按下 key2 进入界面 0。

界面 3 显示：左起第一位 G 管闪烁，第二第三显示月，第四第五第六位不显示。此时按下 key1 进入界面 4，按下 key2 给月加一。

界面 4 显示：左起第一位 B 管闪烁，第二第三第四位不显示，第五第六位显示日。此时若按下 key1 进入界面 5，按下 key2 给日加一。

界面 5 显示：左起第一位 A 管闪烁，第二第三位显示时，第四第五第六位不显示。此时若按下 key1 进入界面 6，按下 key2 给时加一。

界面 6 显示：左起第一位 F 管闪烁，第二第三第四位不显示，第五第六位显示分。此时若按下 key1 进入界面 7，按下 key2 给分加一。

界面 7 显示：左起第一位 A,B,F,G 管闪烁，左起第二第三位显示闹钟的时，第四第五第六位不显示，此时按下 key1 进入界面 8，按下 key2 给闹钟的时加一。

界面 8 显示：左起第一位 A,B,F,G 管闪烁，左起第二第三位第四位不显示，第五第六位显示闹钟的分，此时按下 key1 进入界面 0，按下 key2 给闹钟的分加一。

### 3.原理讲解

CPU\_4bit 模块只会在按键信号的影响下改变它输出的 4bit 的信号值，data\_base 模块除了给秒值每秒自加一外，还在且仅在特定 4bit 信号和按键控制下对月、日、时、分进行调节。这个特定 4bit 信号是调节某界面的选通信号，例如选通界面 3 的 4bit 信号是 0011，那么 data\_base 模块仅在 4bit 信号是 0011 时按动 key2 给月加一；选通界面 4 的 4bit 信号是 0100，那么 data\_base 模块仅在 4bit 信号是 0100 时按动 key2 给日加一。

这样做的好处是使我们对时间信号值的改变完全可视，而且在进行某种操作时，秒值自加一和进位完全可以不受影响的进行。

本设计纯粹由 FPGA 实现数字电路完成，没有高级语言复杂的进程、线程理论，但是做到了并行化。

本设计一定有不足和令您不满意的地方，希望读者 Fork 这个仓库并给出修改建议，我会虚心接受。