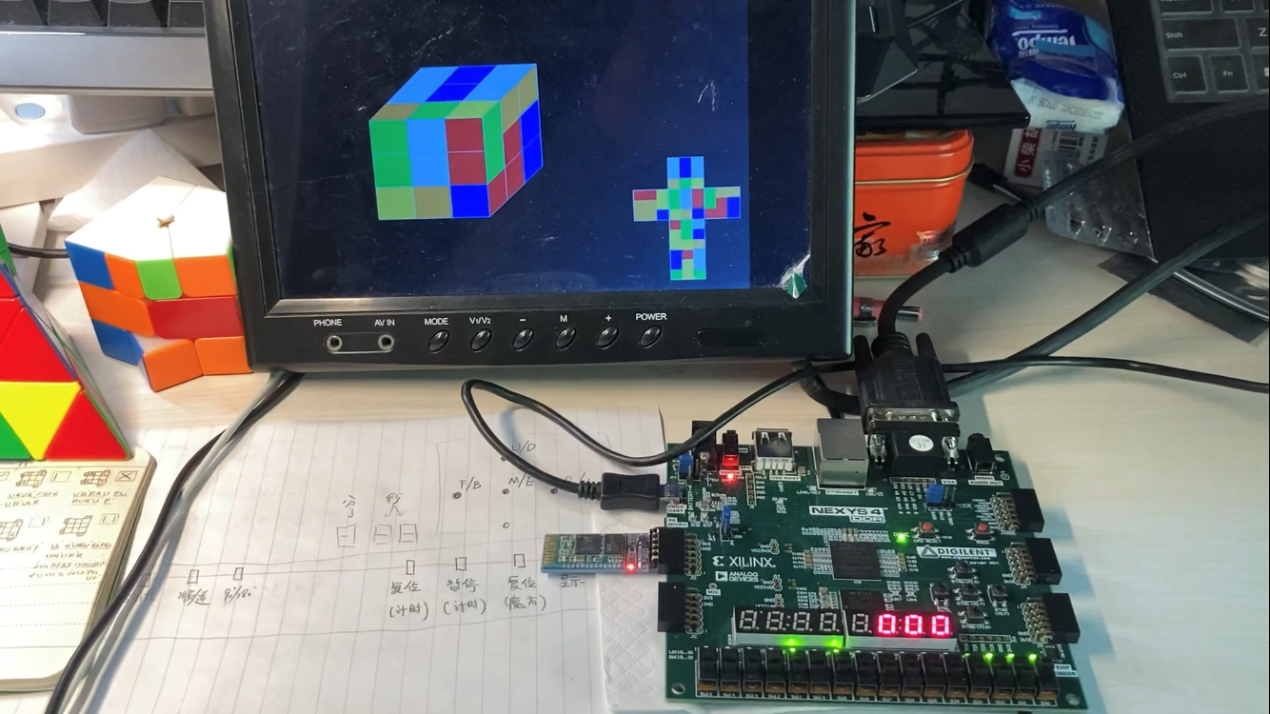
**Fpga虚拟魔方**

1. 目录
2. 背景
   1. 魔方爱好与解魔方程序
   2. 数字逻辑大作业
3. 过程
   1. 学习VGA协议、方块展开图
   2. 构思操作
   3. 虚拟魔方
   4. 计时器与魔方立体图
   5. 学习蓝牙、利用解魔方程序
4. 整体结构
5. 收获与反思
   1. 调试
   2. 构思、封装
6. 参考与链接

注：文中的图、视频与源文件在附件里

**1 背景**

**1.1 魔方爱好与解魔方程序**

中学时我就喜欢上了玩魔方。比如三阶魔方，就学习了许多种解法，比如桥式、ZZ、彳亍、降群等。大一的时候学习了针对计算机求解的降群法。当时当练手，用C++写了一个利用降群法求解魔方的程序。求解过程大致分为四步，每一步将模方状态操作的可能性减少一点，然后利用A\*算法求解。{ U, D, L, R, F, B } -> { U, D, L, R, F2, B2 } -> { U, D, L2, R2, F2, B2 } -> { U2, D2, L2, R2, F2, B2 }。

**1.2 数字逻辑大作业**

这学期数字逻辑可布置了大作业，要用fpga还有一些外设做一个东西。

**2 过程**

**2.1 学习VGA协议、方块展开图**

2021年12月8日，开始着手完成数字逻辑大作业。决定从VGA接口入手。在网上找了许久教程之后，理解了VGA协议的思想，并写了一个测试用的小驱动，在显示屏上显示八个竖着或是横着的色块条（图1.1~1.2），并借此微调摆正四周的黑框。接着，为了测试曲线图形的显示，我试着画了个框（图2.1），成功后觉得挺像魔方展开图，而且展开图都是一个个方块色块，容易画，于是我将六个面共54个色块的颜色各用一个12位寄存器来表示，逐步试着完成了魔方六面展开图的显示。（图2.2~2.4）这时，我也决定了大作业就做一个能打乱复原的虚拟魔方了。

**2.2 构思操作**

因为提供的板子上按钮有限，所以我先暂时停下来，根据我常用的魔方桥式解法，精心设计了一组操作方式。准备抛弃转体，保留（L,R,U,D,F,B,M,E）八种操作，并两两组成一组，用一个拨片实现类似键盘shift切换表里模式的功能，再分配四个按钮触发转动。虚拟魔方还应提供九十度、一百八十度与二百七十度三种角度的转动，所以增加两个拨片以控制转动的角度。（图3）图中的计时功能是其他功能基本完善后添加的，图3为最终版操作示意图。

**2.3 虚拟魔方**

2021年12月10日，完成了虚拟魔方的模块。之前VGA显示模仿展开图只是属于虚拟魔方与用户交互的一部分，理应与虚拟魔方本身分离开来。于是我新建了一个模组，代表虚拟魔方。它取代了原来VGA模块中那54组寄存器，存储着魔方各个色块的信息，接收转动操作的信息以及触发信号，并时刻输出各个色块的颜色信息。这个模块最主要的功能就是对每种旋转操作进行模拟。

为了实现这个模块，我先对魔方54个色块进行了编号，并与模块中的54组寄存器一一对应。根据之前设计的操作数，计算可得共有20种需要处理的信号，每种都要对18个色块进行移位，而且还不是简单的顺序移位。若是手写这些操作，很容易出错，所以我借助C++写了个小脚本，帮助生成了这一部分的代码。具体可看附件中的代码以及注释。

编写这个模块的过程中还有一个小插曲，由于触发信号与移位操作时序冲突，初版模块会出现实际转动与输入信号有依次输入的间隔。（视频1）我增加了一个小小延时器，解决了这个问题，代码中的注释也有解释。

此时虚拟魔方以成雏形，我也尝试使用该虚拟魔方进行复原，还挺好玩，可以在两分钟左右复原，看来我设计的按键布局还算合理。接下来就是完善与用户的交互以及一些小小附加功能了。

**2.4 计时器与魔方立体图**

2021年12月15日，完成了计时器功能与魔方立体图的显示功能。（参考图4）

（复原演示视频https://www.bilibili.com/video/BV1SS4y1D7X3）

**2.5 学习蓝牙、利用解魔方程序**

2021年12月18日，得知作业外设种类还不够，于是临时学习了一下蓝牙的简单串口协议，准备增加利用蓝牙来操控虚拟魔方的项目。还好由于之前的封装还算独立，只需要将操作编码设计好，再加一个小小解码器与之前的虚拟魔方模块整合，很容易就添加好了蓝牙功能。再用C写了个小脚本，在电脑上将输入的字母转动操作码翻译成二进制编码，通过电脑上的串口调试助手发送给fpga板上的蓝牙模块，这个功能就完成了。由于蓝牙可以快速批量处理或发送操作编码，我惊喜地发现一年前写的解魔方小程序也派上用场了。这样打乱、复原或是某些固定的操作公式全都可以通过操作电脑来完成，这个虚拟魔方可玩性大大提升。

至此，这个fpga虚拟魔方也算完成了。

（最终版演示视频https://www.bilibili.com/video/BV1Ea4116776）

**3 整体结构**

这个fpga虚拟魔方，有这几个部分组成：

虚拟魔方核心模块、vga显示模块、蓝牙控制模块、按键控制模块、计时模块。

**4 收获与反思**

**4.1 调试**

由于用到了VGA接口，想要看到显示效果，输出信号就会有很长的周期。IDE自带的波形分析器就基本排不上用场了。所以就只能把程序烧到板子上调试。做蓝牙的时候，总是发现无法正确地对电脑发送的信息进行反应，或者根本对信号没有反应。这时怎么知道是蓝牙模块出了问题，还是其他模块地逻辑出了问题呢？我利用了板子上的led灯，通过一组灯的组合显示出蓝牙模块目前接收到的纯二进制信息，再用另一组灯的组合显示出虚拟魔方模块所接收到的编码信息。这就像C语言编程中的输出调试，就是烧板子和编译比较久。

**4.2 构思、封装**

编写Verilog代码的时候，因为事先进行了适当的封装、分类，所以后来临时加蓝牙模块就相对顺利了。而之前写的那个解魔方的程序，为了能将所需要的信息压进一个四字空间内，用了许多技巧对魔方进行的编码，而且那四步每步都重新进行了编码。导致如今我再看代码，也看得比较困难了，也完全没有了维护的欲望。（当时写的解法，倒数第二步有点小bug，会导致有几率无法得到完全复原的解，最后要手动操作几步。）所以说，在进行有长远计划的项目时，应该先花点时间构思一下，并进行适当的封装，提供相对简洁的接口以便日后维护。

**5 参考与链接**

（数字灯时序从这里学习的<https://blog.csdn.net/qq_45753394/article/details/119517414>）

（蓝牙模块从这里学习的<https://github.com/lllbbbyyy/FPGA-OV2640>）

（复原演示视频<https://www.bilibili.com/video/BV1SS4y1D7X3>）

（最终版演示视频<https://www.bilibili.com/video/BV1Ea4116776>）

（降群法资料1 <http://www.mf8-china.com/forum.php?mod=viewthread&tid=106671>）

（降群法资料2 <https://tieba.baidu.com/p/3002593929>）