**数字逻辑与部件设计实验报告**

实验二：ALU

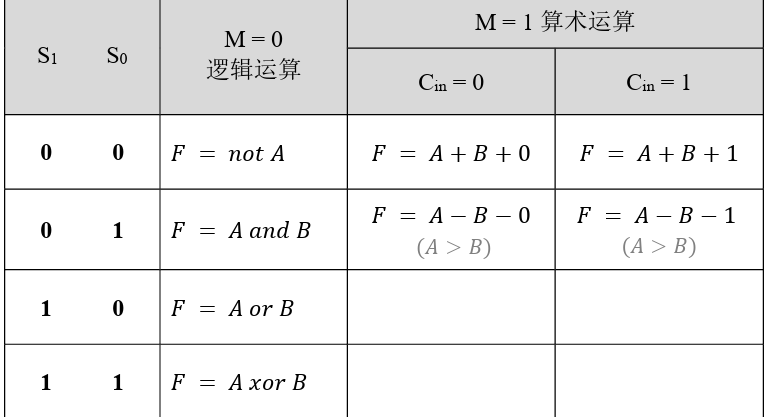


学生姓名： 谢志康

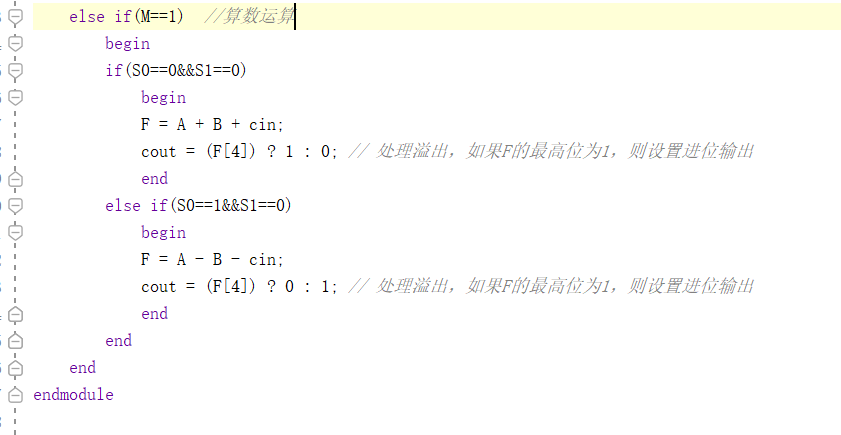
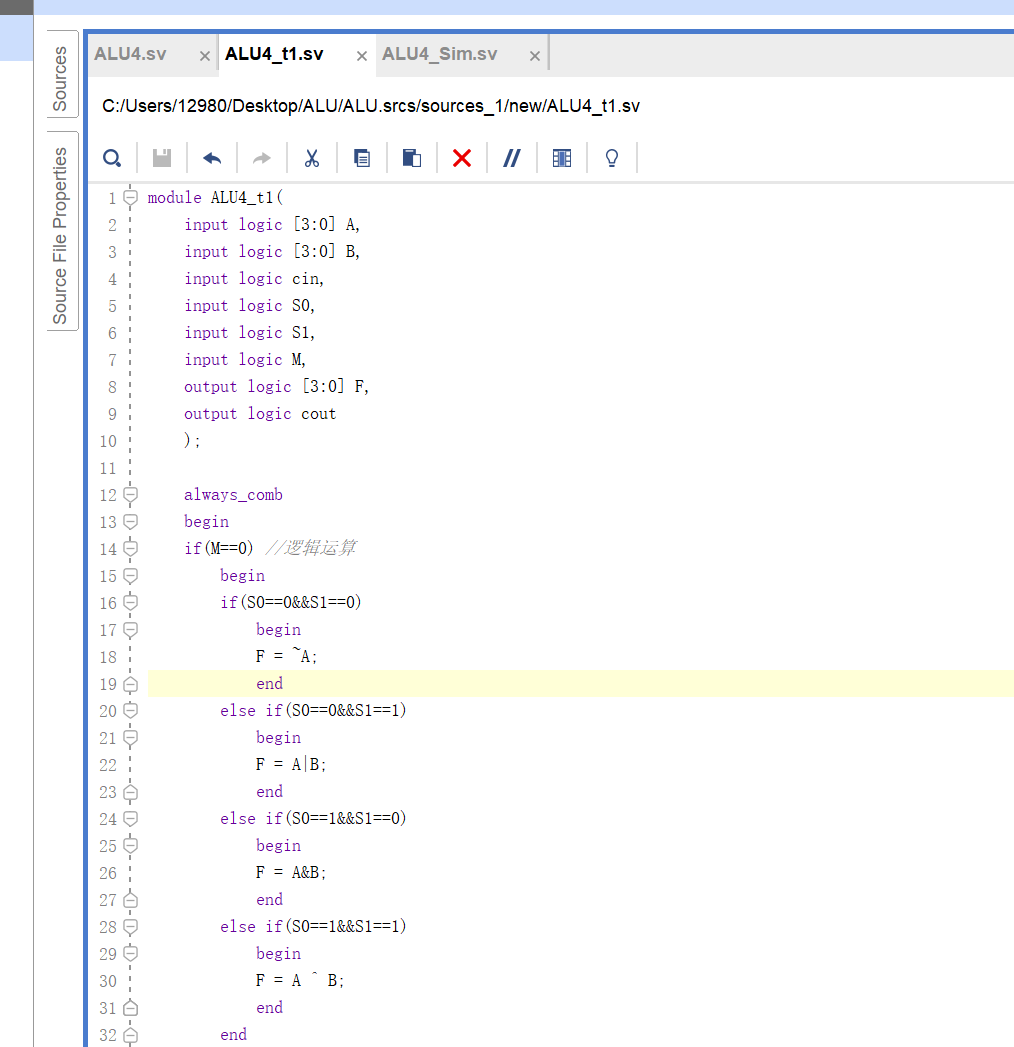
学号：22307110187

日期：2023.9.18

1. 实验内容
2. 设计一个4位算术逻辑单元，输入信号为： 两组4位数据输入信号（A3—A0，B3—B0），一个进位输入信号Cin； 数据输出信号为：4位数据信号（F3—F0）, 一个进位输出信号Cout。 以上数据均为无符号正整数。 功能控制信号有：S1、S0、M。 当M=0时为逻辑运算，M=1时为算术运算。

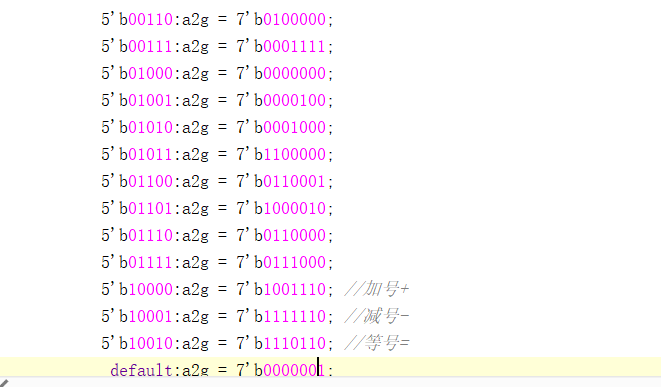


1. 在NEXYS4 开发板上实现上述设计，SW选择可以自己确定。 例如，SW[15]对应：M， SW[14:13]对应：S1和S0， SW[3:0]、SW[7:4]、SW[8]分别对应：A[3:0]、B[3:0]、Cin。 当SW为1时，其上面的LED点亮，否则熄灭。 开发板上的8个七段数码管用于显示十六进制的输入数据和输出数据。
2. 实验方案
3. 首先根据第一题的实验内容构建好ALU4\_t1.sv文件，实现第一题的逻辑运算和算数运算。（详细代码见实验分析介绍）
4. 借助上周数码管实验的思想，构造多层文件，最底层实现灯光功能，再往上控制数码管，再上实现使用ALU控制。
5. 实验分析
6. 第一题：首先构建好ALU4\_t1.sv文件（最底层），主要按照规则实现最底层逻辑。数据输入信号S1，S2，M，cin来控制if-else语句，分别实现对应的要求，代码如下：主要是注意算数运算中由于加减带来的溢出问题，将其储存在进位控制信号cout中。

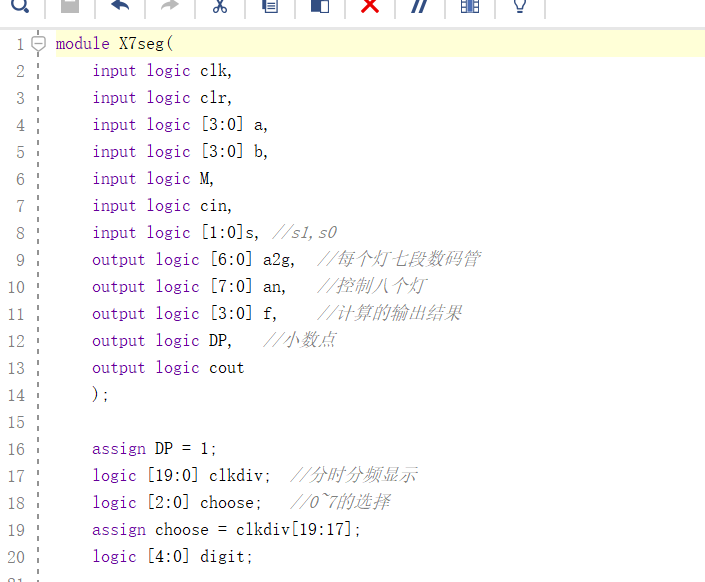


1. 第二题：在第一题和前两周实验的基础上完成。

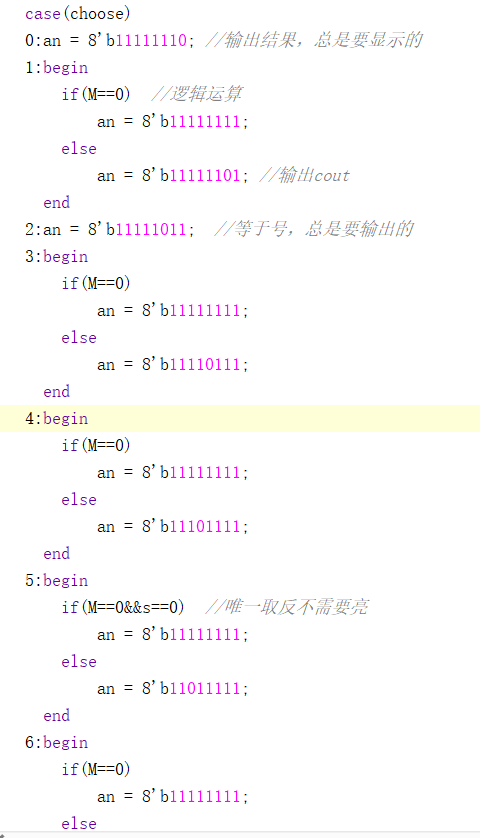
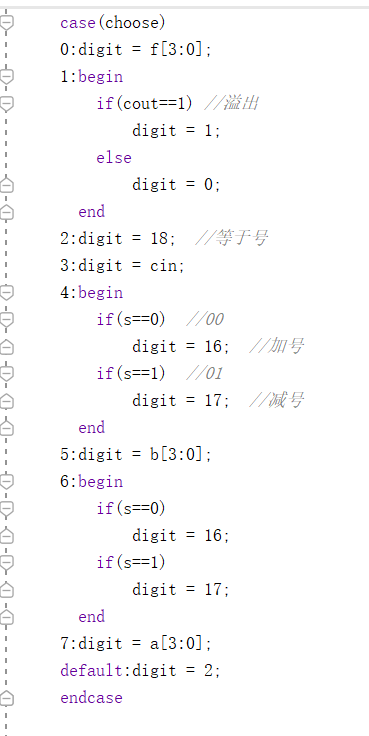
总共有四层文件，这样逻辑很清晰：首先，最底层，定义数码管如何亮，除0~F十六个十六进制整数外，还加了三个符号，+-=。



第二层，主要是做控制数码管的操作，定义了如下变量

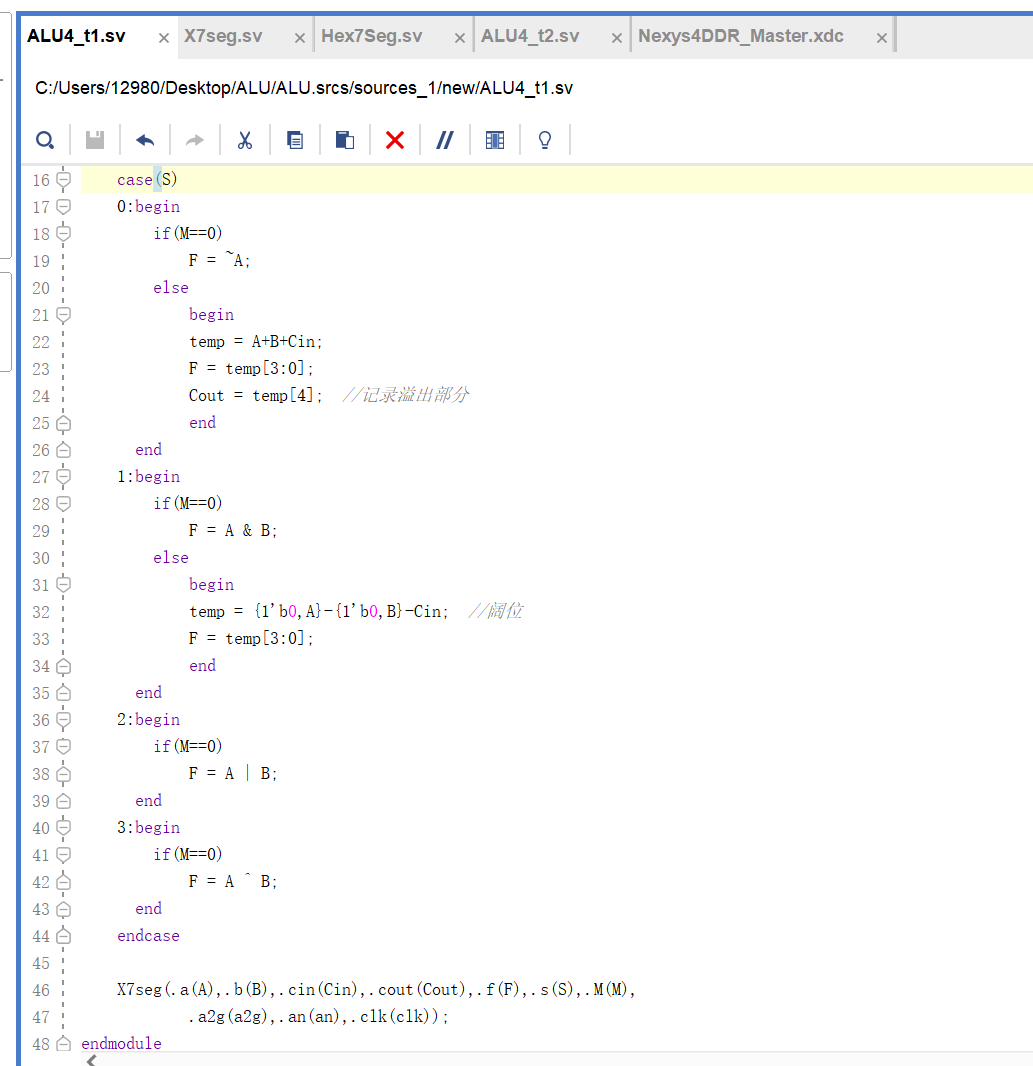


按照题目建议分配A，B，结果F的位置，由choose变量来按照不同的选择进行对应的操作。代码如下：

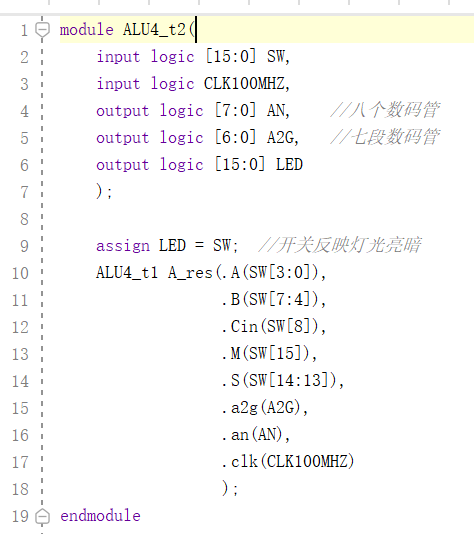


对应第一页的逻辑表实现简易逻辑，有几个位置的数码管必定要亮，譬如等于号，譬如结果f，因此就设那一位为0，其余有时亮有时不亮，参见上面代码。

第三层ALU4\_t1文件，主要用来实现运算。我设了一个temp变量来记录溢出（比f多一位）关键是加减法时的溢出问题，利用中括号这种语句{1’ b0,A}为A的左边补一位，用cout记录溢出。下面为主要实现代码。

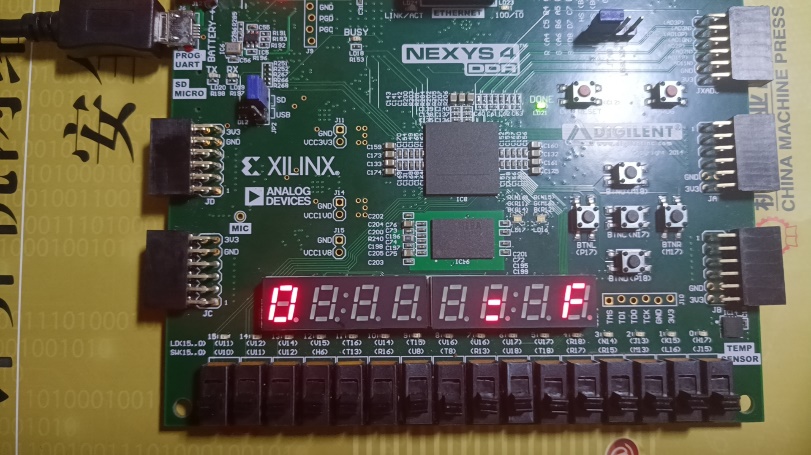


第四层就算是综合了前面三层，连接约束文件，最终实现功能。下面为主要代码：



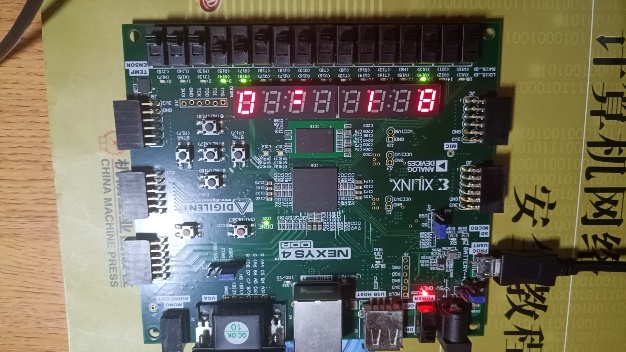
开发板图片和附加解释说明如下：

1.初始：0 （m，s1，s0）为000，表示取反，结果为f（1111）



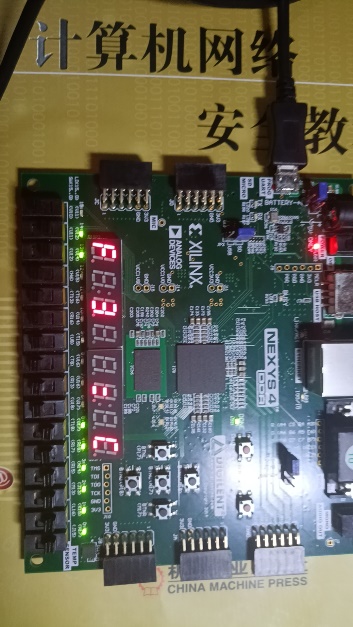
2.逻辑与：001：8&1==0—1000&0001==0000

A&3==2---1010&0011==0010

   
3.逻辑或：010：A|3==B---1010|0011==1011



4.逻辑异或：011：F^3==C---1111^0011==1100



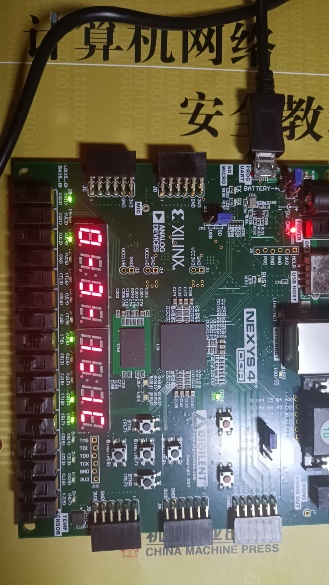
5.算数运算加法：100：F+3+1==19(十六进制13)

SW[8]是cin，此时设置为1，功能正常，进位正常，进位的数字由cout记录下来传给数码管。



6.算数运算减法：101：F-3-0==C

0-3-1==-4 表示为1C即11100，取反码后00011，补码加一为00100即为+4，可知1C就是-4，表示正确。至此，所有逻辑运算和算数运算全部正确实现了。

四、总结与思考

这次实验不仅综合了前两次实验的内容，还增加了许多新知识。通过这次实验，我更加牢固地掌握了数码管的控制和灯光操作，systemverilog语法知识和各层级代码的连接。此外，我学到了利用{}可以实现数字扩位，一般用来记录溢出或符号位。总的来说对我有一定难度，但是还算是比较顺利的做出来了。