**数字逻辑与部件设计实验报告**

**实验02：【七段数码管】**

**姓名：刘丹桐 学号：21307090080 日期：2023.9.13**

**一：实验内容：**

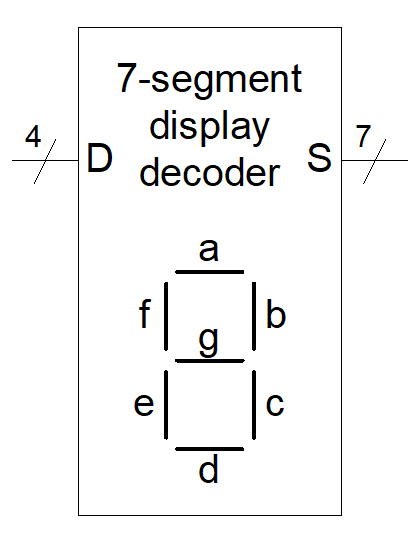
1．设计驱动右侧4个七段数码管，显示固定数据2023。

2．设计驱动8个七段数码管，实现左侧4个七段数码管显示自己的学号最后4位；右侧4个七段数码管显示内容可以根据16个拨码开关(SW)进行相应的16进制数据显示。

**二：实验方案：**

**实验1：**

首先依据课件写出七段数码管的16进制译码器，该开发板上七段数码管的接线方式为共阳极，点亮时引脚输出为低电平（0）。由于每个七段数码管可以显示一位16进制数字，故需要输入4位digit。输出为7位A2G，分别对应七段数码管如图所示的七个位置。



注意到多个LED同时点亮时其值为相同数字，为了使其能够显示4个不同的数字，利用了人眼的视觉暂留现象和时钟分频的方法，轮流点亮八个七段数码管。实现方法是利用20位的变量clkdiv，取最高两位作为亮起的数码管的序号。如此一来，每过2^18个时钟周期，下一个七段数码管亮起，实现分时复用。

该模块的输入有x（16位二进制数，作为七段数码管显示的内容），clk（时钟），clr（时钟复位）；

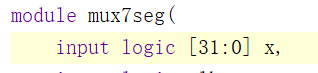
输出有dp（小数点，在本实验中不亮），an（4位，控制4个数码管的使能），a2g。

**实验2：**

在实验1的基础上稍作修改，使分时复用模块支持32位x输入，其中高16位为学号（直接在模块内赋值），低十六位为16个拨码开关SW所表示的十六进制数。

译码器部分代码不变，分时复用模块的主要改动如下：

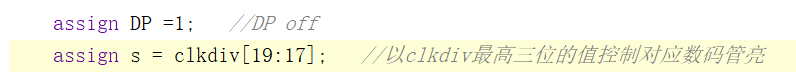
1.将输入的x变为32位：



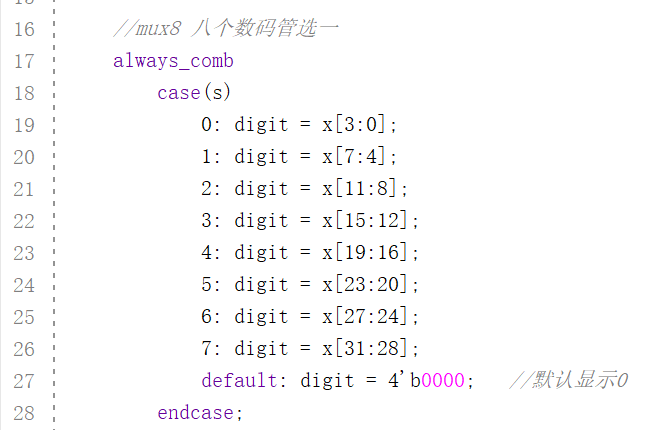
2.将选择特定数码管亮起的s改为3位使其能够表示0~7的八位数字：



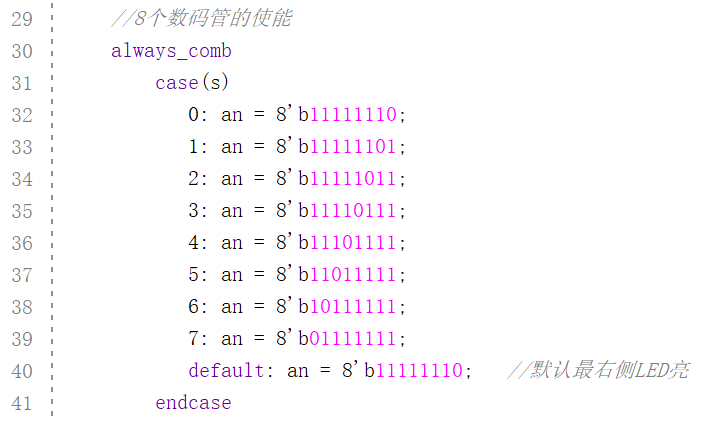
3.将s对应的值改为clkdiv最高三位：



4.将数码管能够显示的数字扩展为8位：



5.将使能扩展至8位：

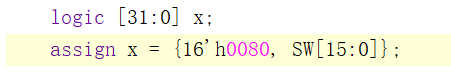


**顶层模块的改动如下：**

1.将输入的SW改为十六位：

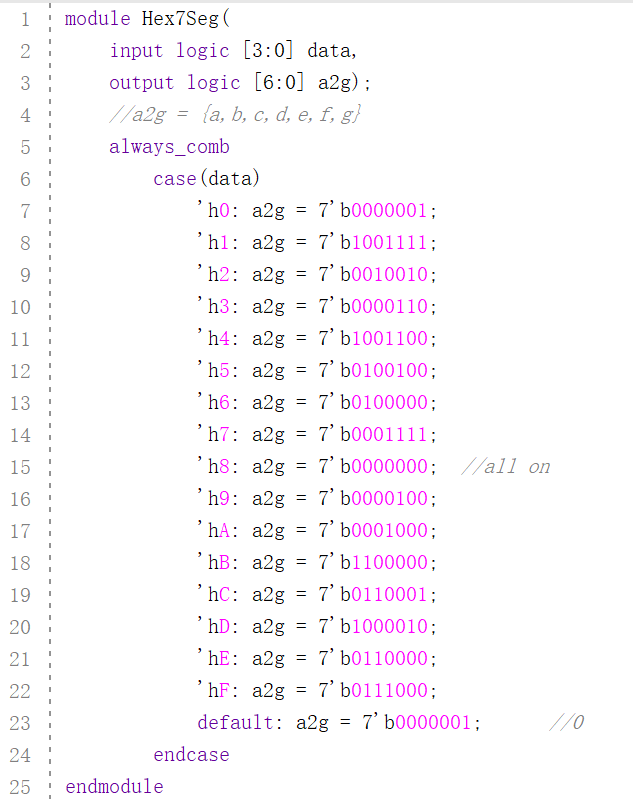


2.给x赋值：

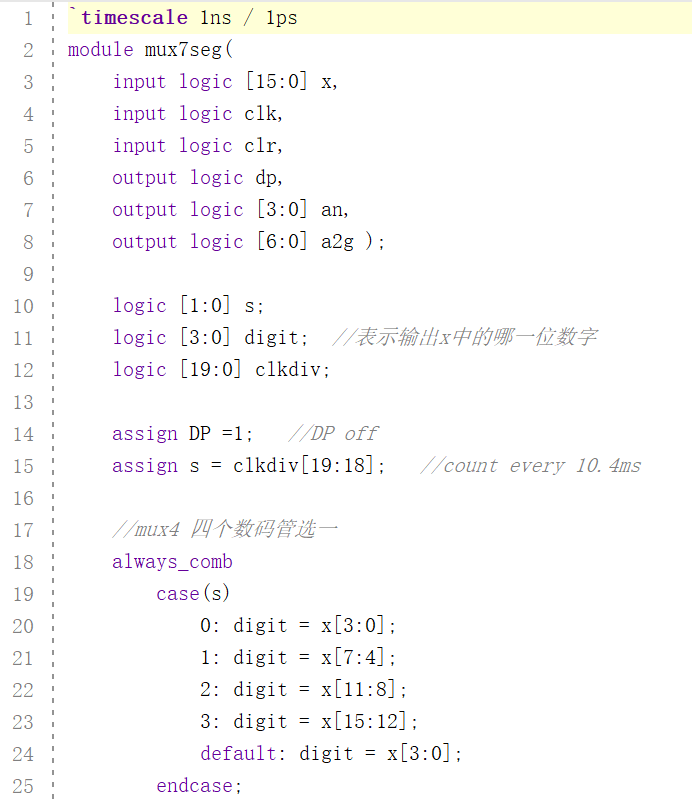


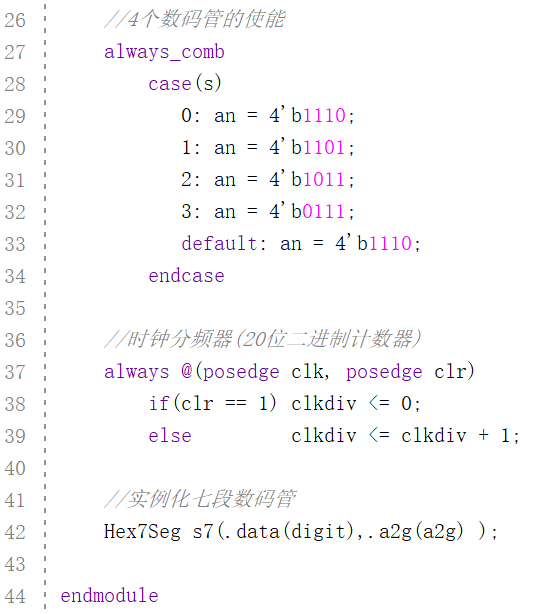
**三、关键代码：**

1.七段数码管十六进制译码器(Hex7Seg.sv)：

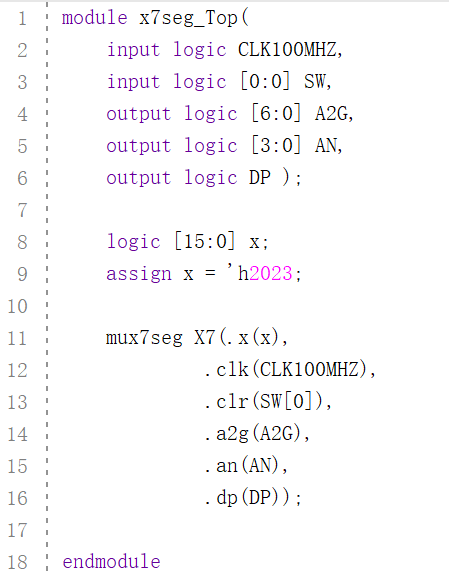


2.分时复用模块(mux7seg.sv)：



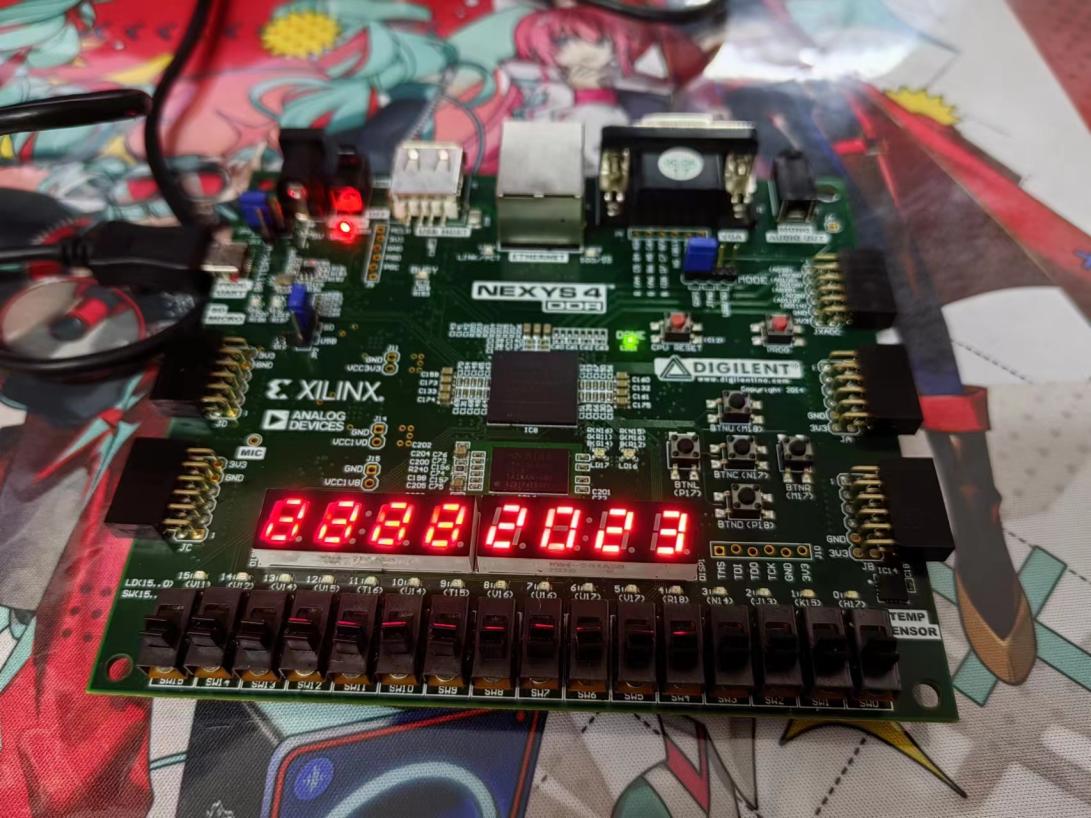


3.顶层模块(x7seg\_Top.sv)：



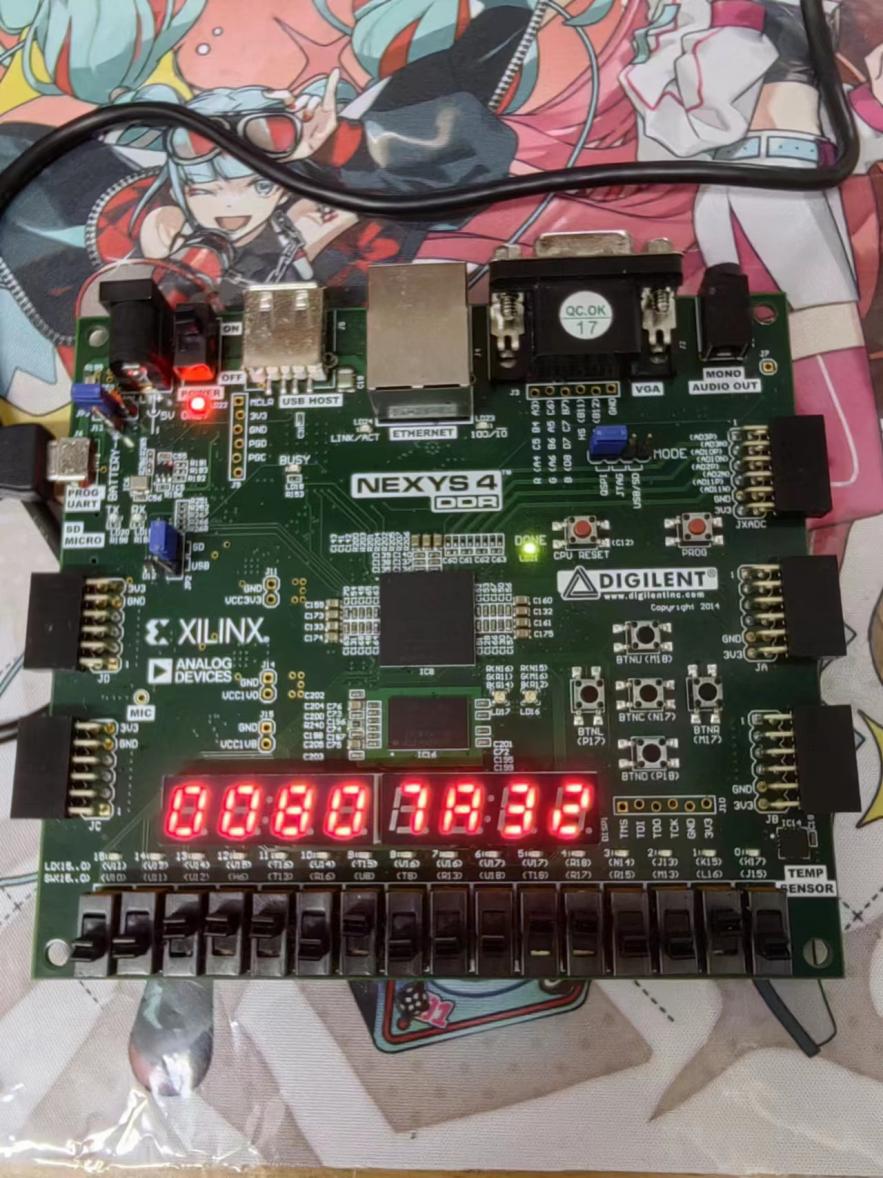
（以上均为实验一的代码）

**四、实验现象：**



（实验一：右侧显示2023）

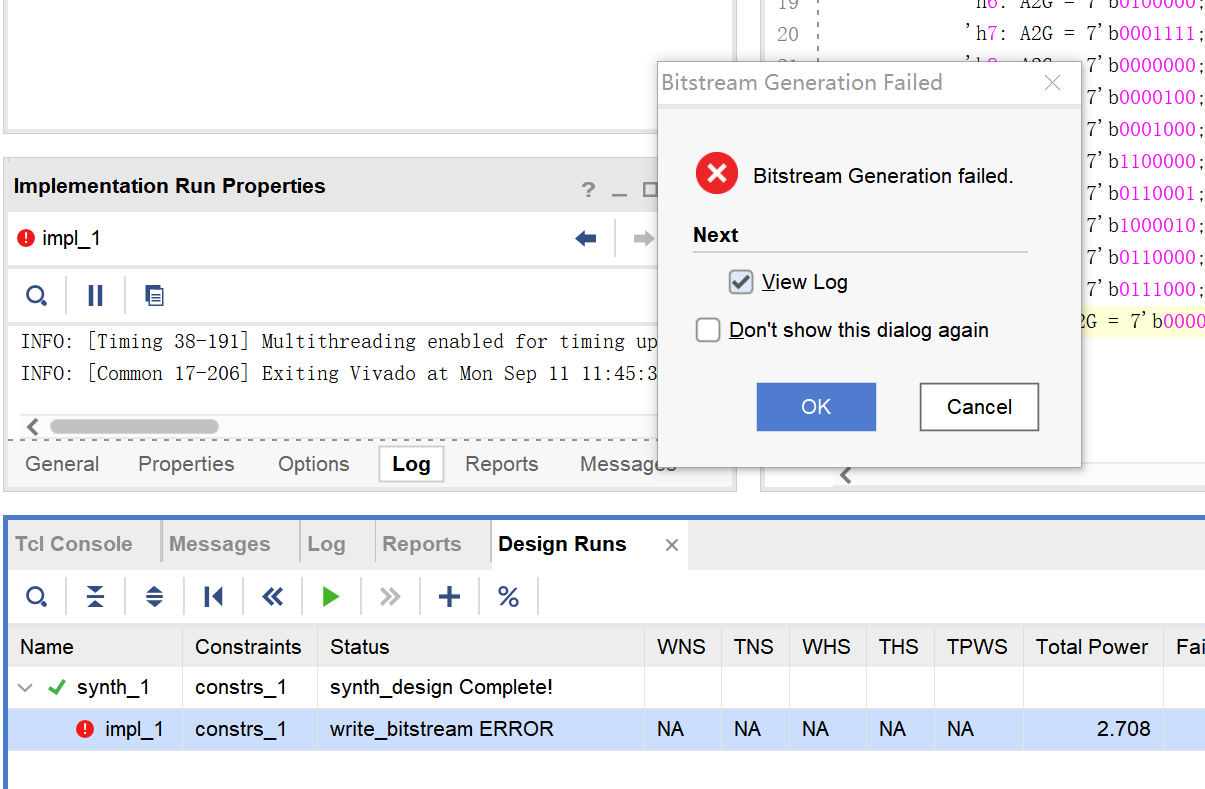


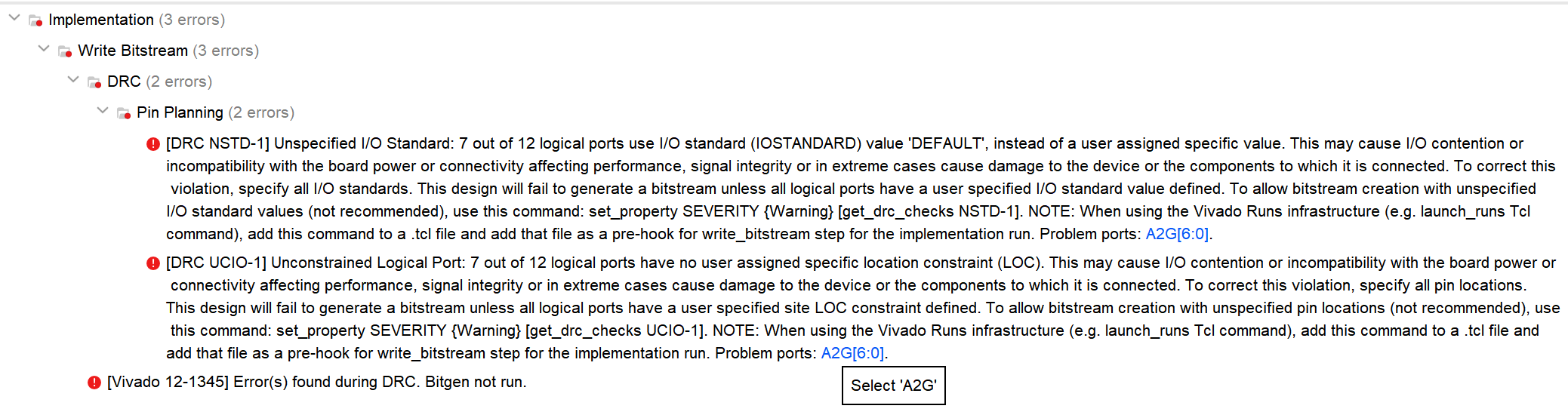


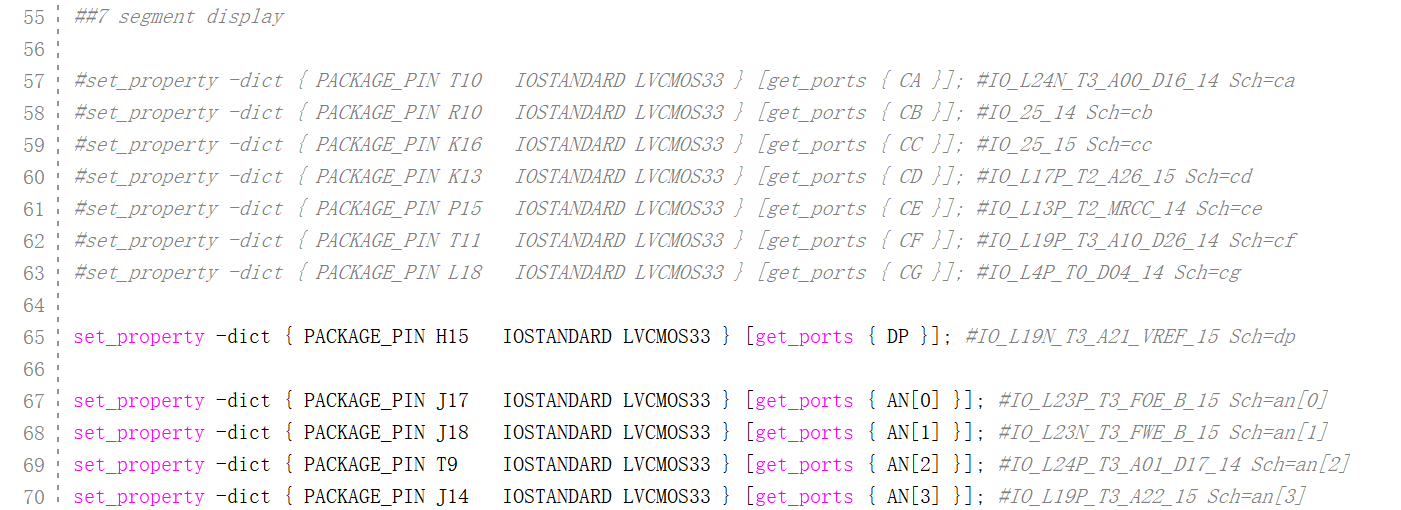
（实验2：左侧显示学号（0080），右侧显示拨码开关表示的数字）

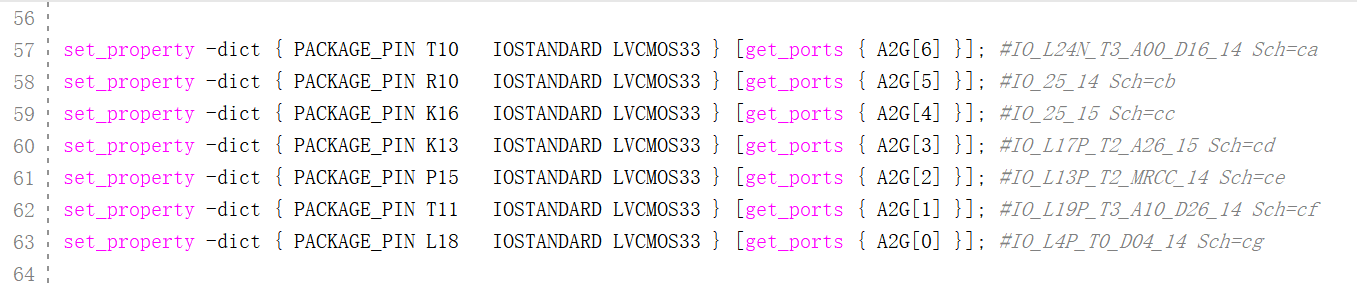
**五、遇到的问题和解决方案：**

1.运行课件上的示例代码时遇到了报错，看了报错信息发现是引脚（A2G）没有解注释和改名导致的，改过约束文件后解决了这一问题。

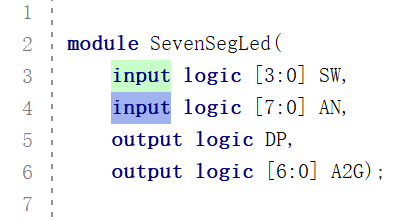








2.烧录bit文件后发现板子上的七段数码管不亮，检查了一遍代码发现把使能写成了输入端口，改为output后解决了这一问题。



**六、总结与收获：**

本次实验相较于第一次实验难度有了巨大的飞跃，在各个环节都可能出现问题，最终导致无法正常运行。但是在修改示例代码和debug的过程中我对七段数码管和system verilog的语法和逻辑都有了更加深刻的理解。此外我还解答了同组组员的疑惑，帮组员分析可能出现的错误，帮他解决了问题。总之本次实验的成功使我对顺利完成下一次实验充满信心！><