# 实验六 EDA作业二

自35 夏弘宇 2023011004

## 一、实验目的

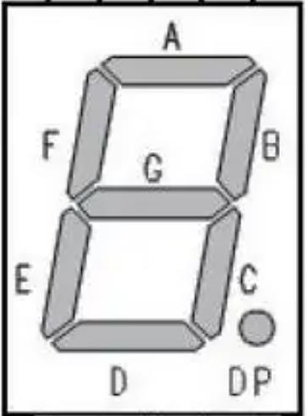
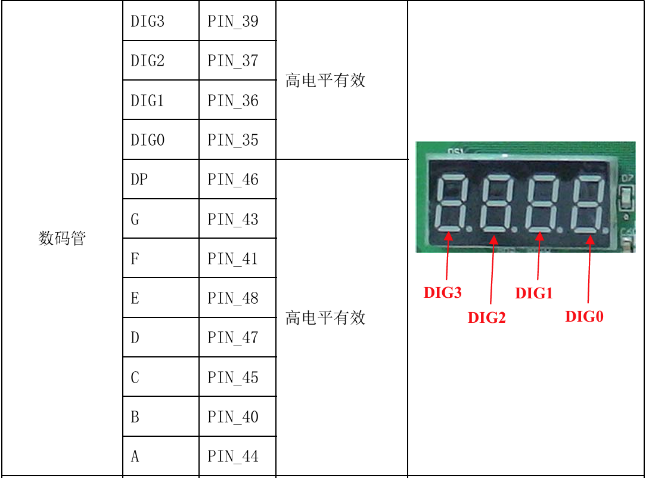
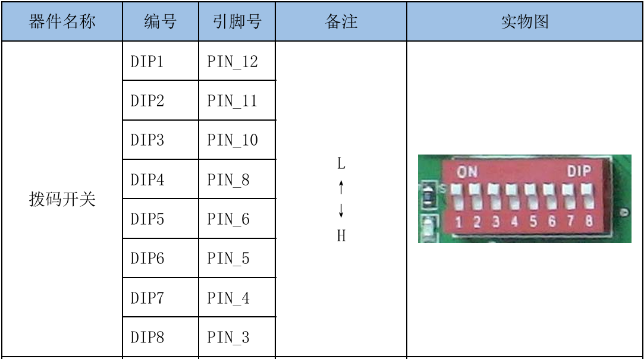
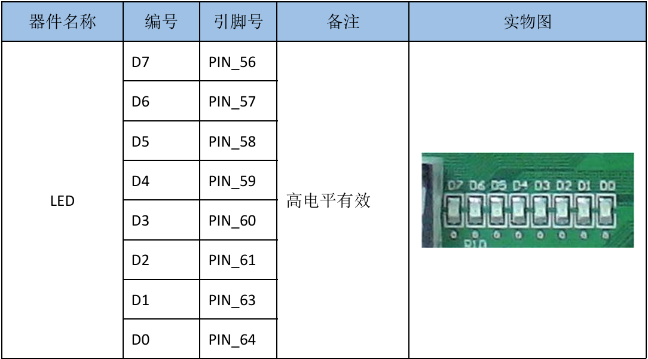
1. 熟练掌握面向FPGA的简单数字系统的设计流程。

2. 学习编写测试文件对设计电路进行仿真验证。

3. 熟悉实验装置——实验板，掌握板上外设的工作原理。

## 二、预习任务

#### 1. 阅读网络学堂中的“FPGA 实验板说明书”了解实验板上的外设资源，并掌握其工作原理。



（1）拨码开关：当拨码开关设置为“1”时，开关断开，输出高电平；反之，当拨码开关设置为“0”时，输出低电平。

（2）发光二极管LED：高电平有效（LED亮）

（3）4位扫描显示数码管：共12根数据线，分为8根段选线、4根位选线，由段选线上信号选择哪一根线段（管）亮，由位选线选择哪一位亮。

当段选信号固定时，数码管上显示的字符内容是固定的。给定特定的位选信号后，对应的数码管会显示出相应的字符。如果按时间轮流控制各个数码管的位选端，数码管将会依次显示字符。在轮流显示的过程中，每个数码管的点亮时间一般为1到2毫秒。由于人眼的视觉暂留现象和发光二极管的余辉效应，尽管各个数码管并不同时点亮，只要扫描速度足够快，肉眼就能看到一组稳定的显示数据。

（4）晶振：提供频率为50MHz的高低电平信号。

#### 2. 根据实验任务中的步骤提示，写出要用到的电路模块及其功能。

1. 二选一数据选择器：
2. 四二选一数据选择器：
3. 4位二进制补码运算器：输入负数时，将输入的原码转为补码，将输出的补码转位原码。
4. 1位二进制全加器：能处理进位的1位二进制全加器。
5. 4位二进制全加器：能处理进位的4位二进制全加器。
6. 位置选择器（位选器）：选择哪一位显示
7. 显示译码器（段选器）：将三位2进制信号转变为显示的数值
8. 分频器：根据需要选择倍数，考虑5、10、16倍，并最终集成为200k。

#### 3. 进入实验室之前，可预先完成电路的设计输入。

## 三、实验任务（含选做）

在可编程逻辑器件上设计一个运算电路，可以实现S＝M+N。M和N为3位二进制数，其中1位是符号位，2位是有效数字。**要求用原理图的输入方式完成**。

用实验板上的拨码开关模拟运算数（原码输入），用发光二极管表示运算数的正负标志；用数码管显示运算数、运算结果（原码）及运算结果的正负标志。

具体内容及步骤如下：

1. 用门电路设计一个1位二进制全加器，将其封装成1位全加器模块。

2. 以1中已封装的1位全加器模块为基础实现一个4位二进制全加器，并仿真检查功能正确与否（仿真工具不限）。

3. 以2中的4位全加器模块为基础实现一个二进制运算器，可以完成运算S＝M+N。**编写测试文件，使用Modelsim仿真验证运算器的功能。**

4. 设计一个数码管的动态扫描显示电路，可以在4位数码管上同时显示4个数字。电路的输入方式不限（即可用原理图或硬件描述语言方式）。

具体内容及步骤：

（1）设计一个分频器，建议将系统时钟由50MHz分频至250Hz。

（2）设计一个电路，使运算器的两个运算数和运算结果同时显示在4位数码管上。



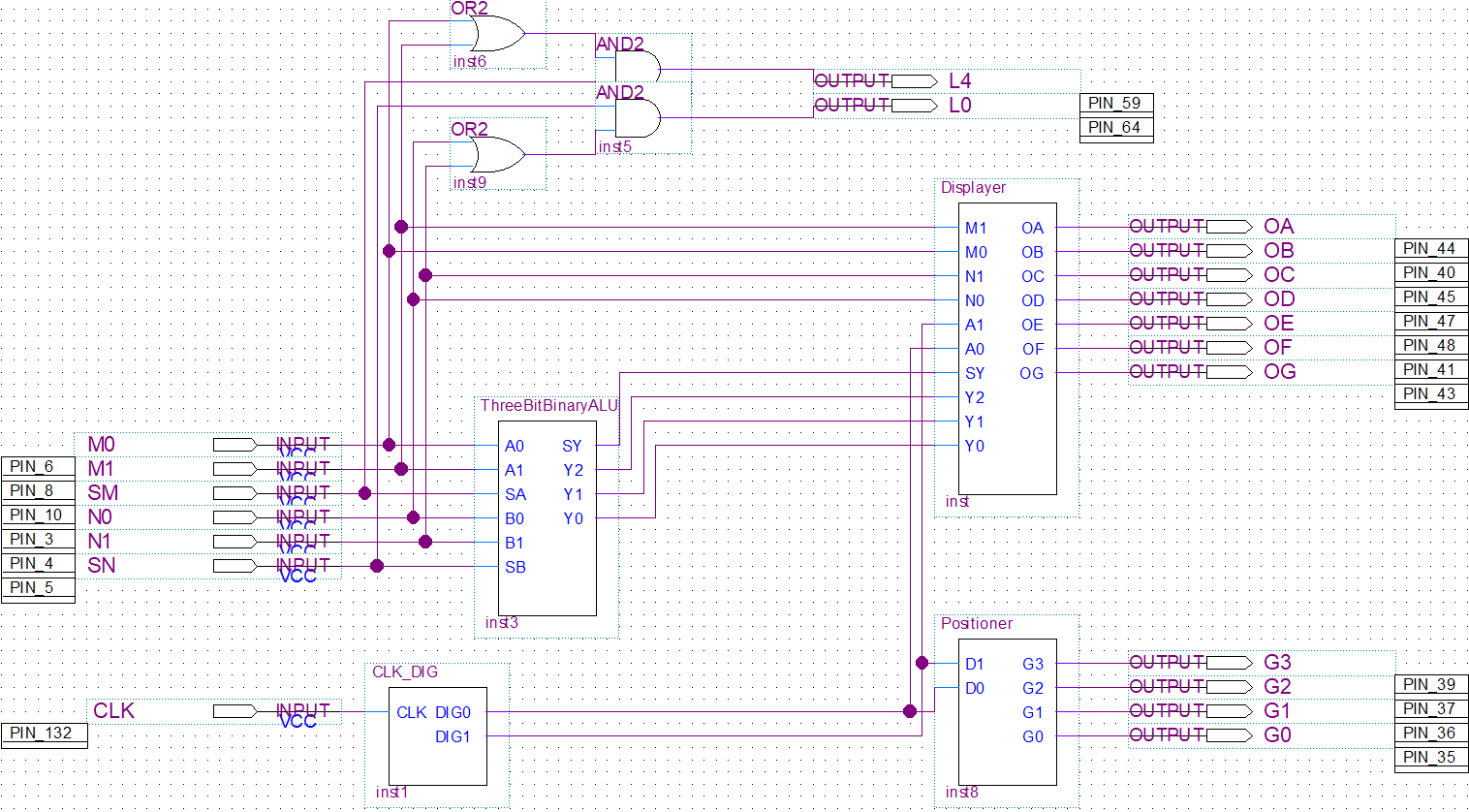
5. 下载到实验板上验证功能。

## 六、实验报告-实验总结

1. 阐述设计思路。

本实验涉及相对复杂逻辑功能的实现，我在设计过程采用模块化的思路。首先来看需要哪些模块：本次实验的核心是运算和显示，对于运算，需要一个**三位二进制运算器**，由于涉及符号，运算器除了需要一个**四位全加器**，还需要**补码运算器**，四位全加器由四个**一位全加器**组成，一位全加器属于简单逻辑，容易实现；对于显示，二极管发光的控制本身就很简单，不必再使用模块，而数码管的控制需要位选与段选两部分，**段选模块**有现成的**7448模块**来负责，但具体输出什么数字与位选信号相关，需要**数据选择器**，由于数字范围0-6，考虑**4位二选一数据选择器**，由4个**一位二选一数据选择器**构成，而位选模块本质上是一个2线-4线译码器，直接实现即可；还有就是位选信号的产生，涉及**时钟信号的分频**。

2. 顶层电路图，并说明其中各模块电路的功能。



1 三位二进制带符号加法器ThreeBitBinaryALU

输入：SA是A的符号，A1和A0分别为A的第1位和第0位；B同理。

输出：计算A+B得到Y，SY为符号，正为0负为1；Y2, Y1和Y0分别为Y的第2,1,0位。

1-1 无输入进位符号的四位全加器FourBitFullAdder

输入：四位无符号二进制数A=A3A2A1A0, B=B3B2B1B0。

输出：四位无符号二进制数S=S3S2S1S0以及进位信号CO。

1-1-1 一位全加器

输入：一位无符号二进制数A,B，进位信号CI

输出：一位无符号二进制数S，进位信号CO

1-2补码运算器Complementor

输入：四位有符号二进制数，S为符号1为负0为正，D=D2D1D0为绝对值。

输出：S为0时，直接输出Y3=S,Y2Y1Y0=D2D1D0；S为1时，输出补码Y3Y2Y1Y0。

2 显示模块（段选）Displayer

输入：M1M0为|M|，N1N0为|N|，SY为Y的符号正为0负为1，Y2Y1Y0为|Y|，A1A0为“位选”信号（实际上应该叫数据选择信号）（00选|M|，01选|N|，10选SY，11选|Y|）。

输出：七位数码管所需的段选信号

2-1 **4位二选一数据选择器Mux4\_21**

输入：A3A2A1A0, B3B2B1B0，选择信号SEL

输出：SEL为1时Y3Y2Y1Y0输出A3A2A1A0，SEL为0时输出B3B2B1B0。

2-1-1 **1位二选一数据选择器Mux4\_21**

输入：A, B，选择信号SEL

输出：SEL为1时Y输出A，SEL为0时输出B。

3 位选模块Positioner：控制哪个数码管显示：00数码管3，01数码管2，10数码管1，11数码管0。

本质上是2线-4线译码器。D1D0输入00，G3G2G1G0输出0001；D1D0输入01，G3G2G1G0输出0010；D1D0输入10，G3G2G1G0输出0100；D1D0输入11，G3G2G1G0输出1000。

4 分频器CLK\_DIG：输入50MHz信号，输出两个62.5Hz的时钟信号，每个周期依次输出00,01,10,11。

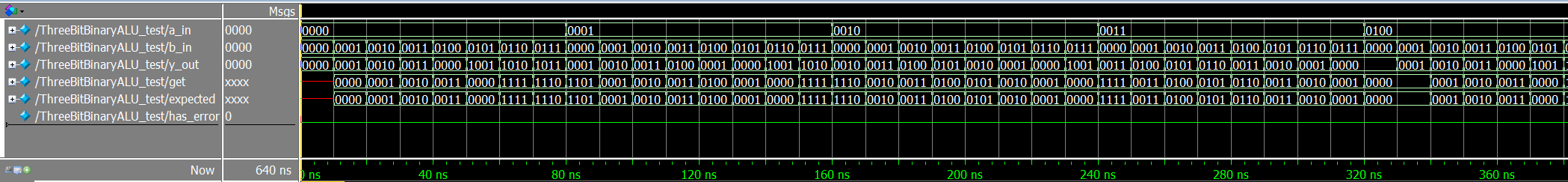
4-1 CLK\_div16：16倍分频器，将输入信号的频率降为1/16

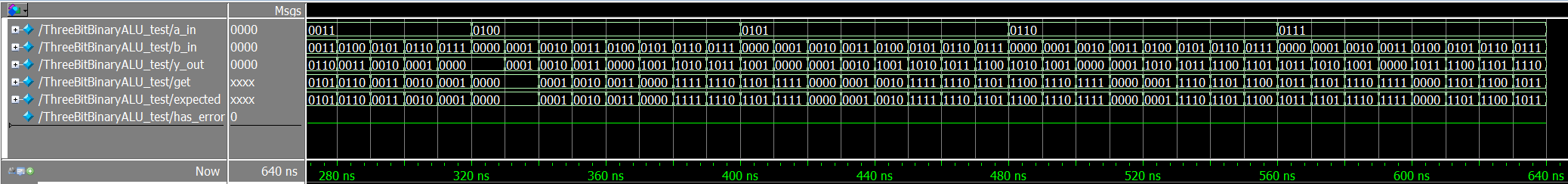
4-2 CLK\_div10：10倍分频器，将输入信号的频率降为1/10

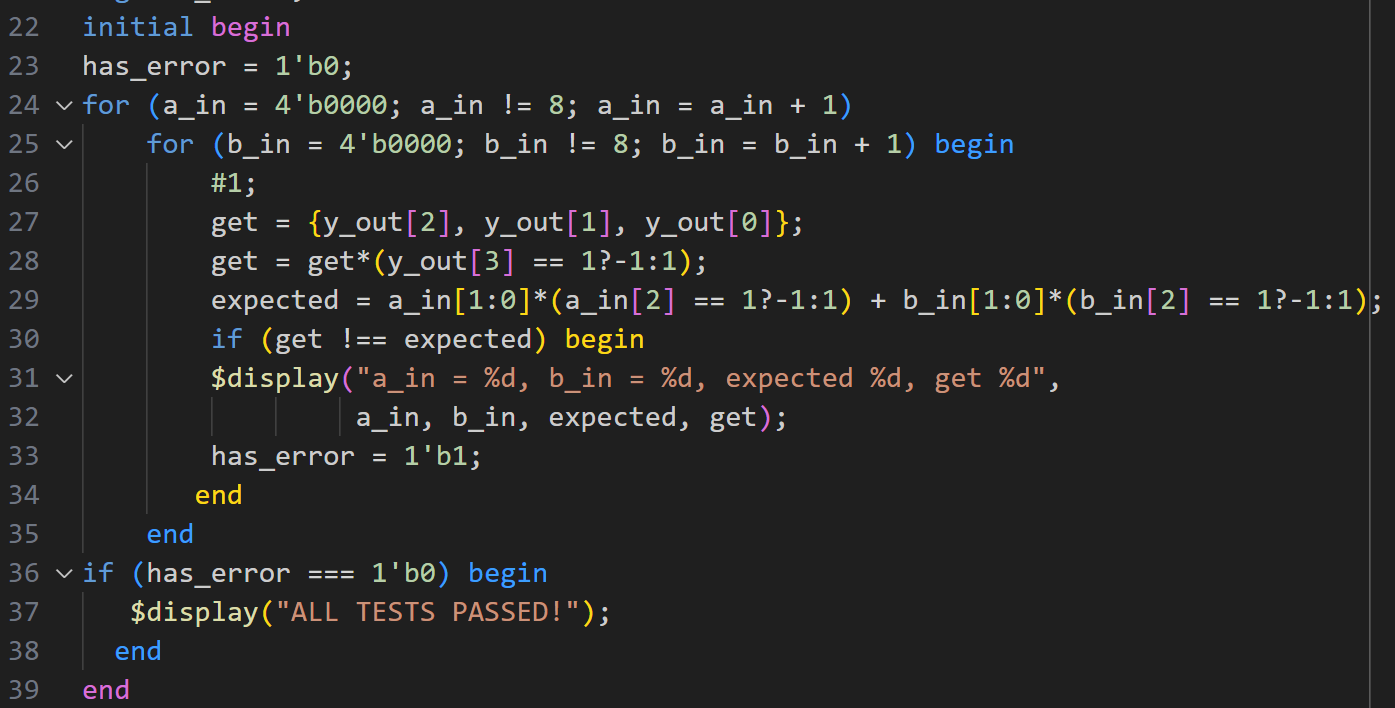
4-3 CLK\_div5：5倍分频器，将输入信号的频率降为1/5

4-4 CLK\_div200k：200k倍分频器，将输入信号的频率降为1/200k

3. 仿真波形图及其分析说明。







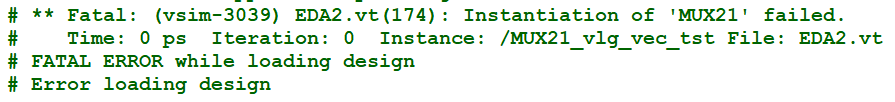
·既然都写testbench了，不妨遍历所有输入情况，由于加法器不是时序逻辑电路，还可以方便地利用程序自动判断正误。采用两重for循环产生输入，每个输入都是3位，第0和1位是数值，第2位是符号，在0-7遍历就可以产生所有输入情况；

·仿真时，窗口输出ALL TESTS PASSED!说明电路逻辑没问题（未判断-0问题），下面简要挑两种情况说明：①输入0100和0100，即输入-0和-0，相加为0，实际输出0000，实现了“无-0”，符合预期；②输入0111和0001，即输入-3和1，相加为-2，实际输出1010，符合预期。

4. 设计和调试中遇到的问题及解决方法。

·四位二进制全加器有八个输入端，仿真测试时，设置8个引脚的输入比较麻烦：采用右键全选，或者双击全选，再设置周期的方式会方便很多。

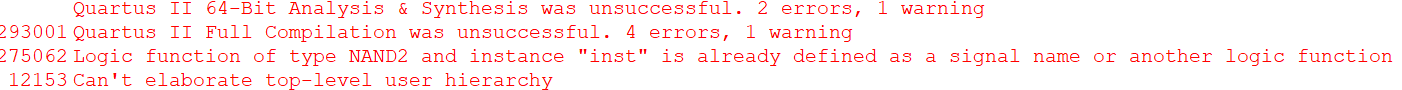
·产生如下报错：MUX21模块的实例化失败。据查是语言设置问题，但本地并没有存在这样的问题，切换仿真工具为自带仿真器问题就解决了。



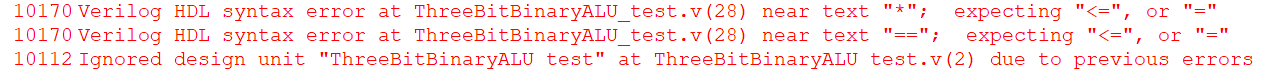
·调整某个模块后，需要在使用该模块的文件内右击点击更新模块，才能生效。

·报错：同一个输入不可以有两个值：发现是两根相交的线莫名其妙多了一个连接点，去除就行

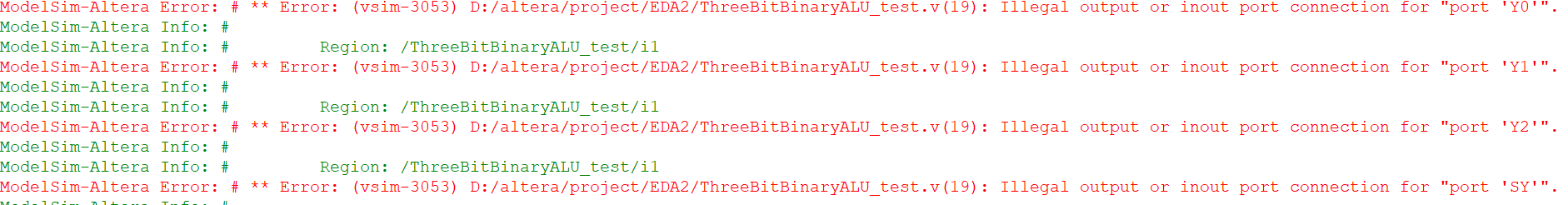
·报错：与非门命名重复，重新命名即可



·测试代码报错：Verilog不能使用\*=，还是得把左侧的变量再抄一遍。



·测试时报错：



·常见易错点：<=的含义跟C++不同，因此for循环中的判断用!=更靠谱。

·仿真波形中，输出产生了高阻态，经查发现在testbench中，输出多开了一位，改成4位后就没有上述现象了：仿真中不像平时编程随便开数组大小，最好做到用多少开多少，方便后续调试。