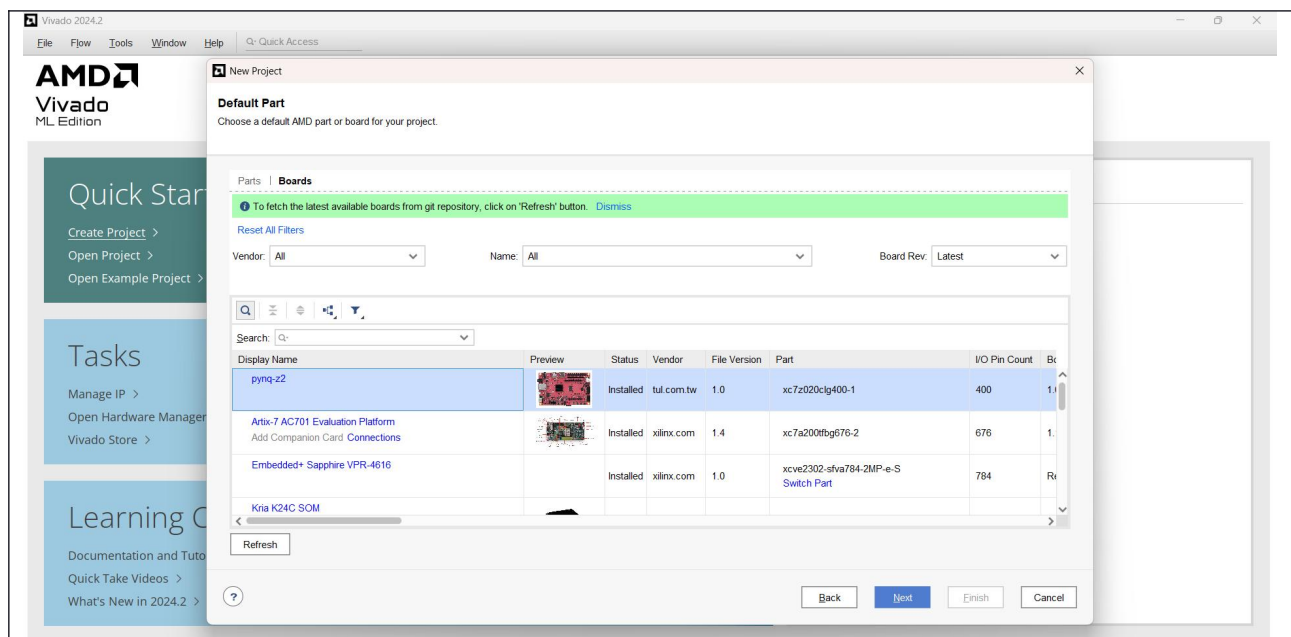


计算机组成与设计\_\_\_\_\_课程实验报告

学号：202300130073	姓名：陈玉澍	班级：计科 23 级 1 班
实验题目： 实验 5 四位补码运算器		
实验学时：2	实验日期：2025. 3. 25	
实验目的： 学会并且设计一个能够实现补码加法、减法、加 1、左移、右移、直接传送等功能的四位补码运算器。		
实验软件和硬件环境： 软件环境： Vivado 软件、FPGA 实验平台 硬件环境： 1. 实验室台式机 2. FPGA 服务器，PYNQ-Z2 开发板		
实验原理和方法： 一、补码运算原理 补码表示：四位二进制补码中，最高位为符号位（0 为正，1 为负），其余位为数值位。正数的补码与原码相同，负数的补码通过对原码取反加 1 得到。 补码加法：直接相加，符号位参与运算，溢出需判断。公式为： $[A]_{补} + [B]_{补} = [A+B]_{补}$ 。 补码减法：转换为加法运算，即 $[A-B]_{补} = [A]_{补} + [-B]_{补}$ ，其中 $[-B]_{补}$ 是 $[B]_{补}$ 的按位取反加 1。 移位操作：左移低位补 0，右移高位补符号位（算术右移），逻辑右移高位补 0。		
实验步骤： (1) 创建工程：打开本地安装的 Vivado 2022.2，新建项目，选择 pynq-z2 器件。		

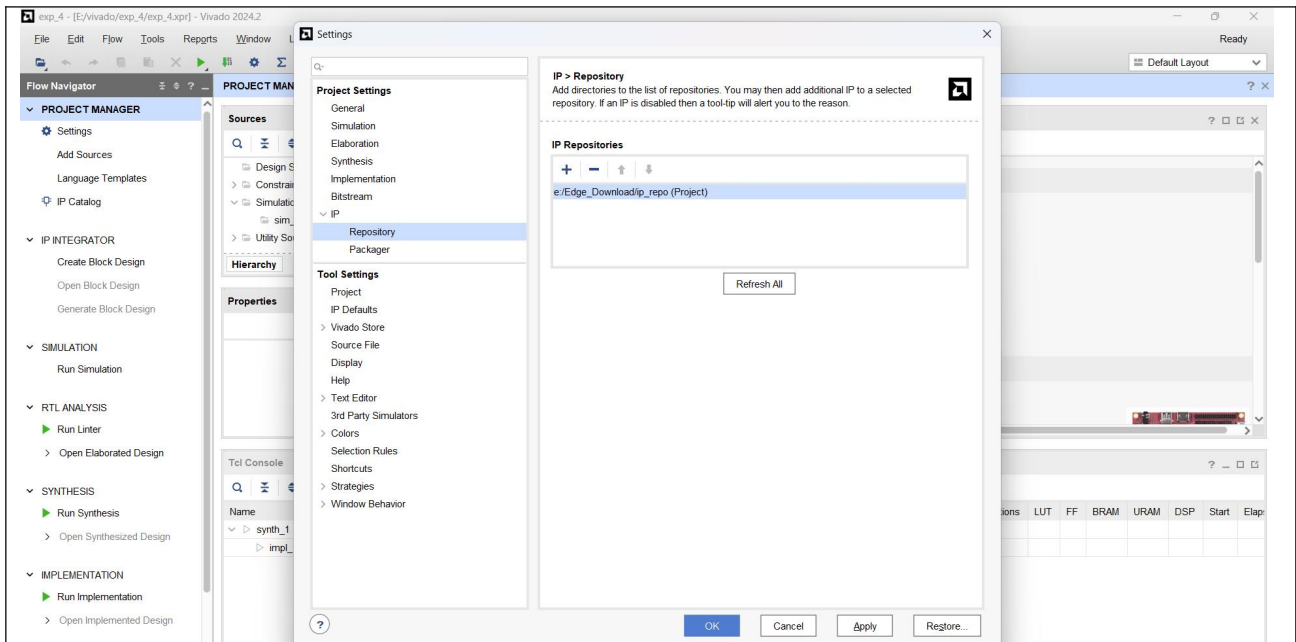


工程向导 会引导您完成设计源文件和目标器件的选择

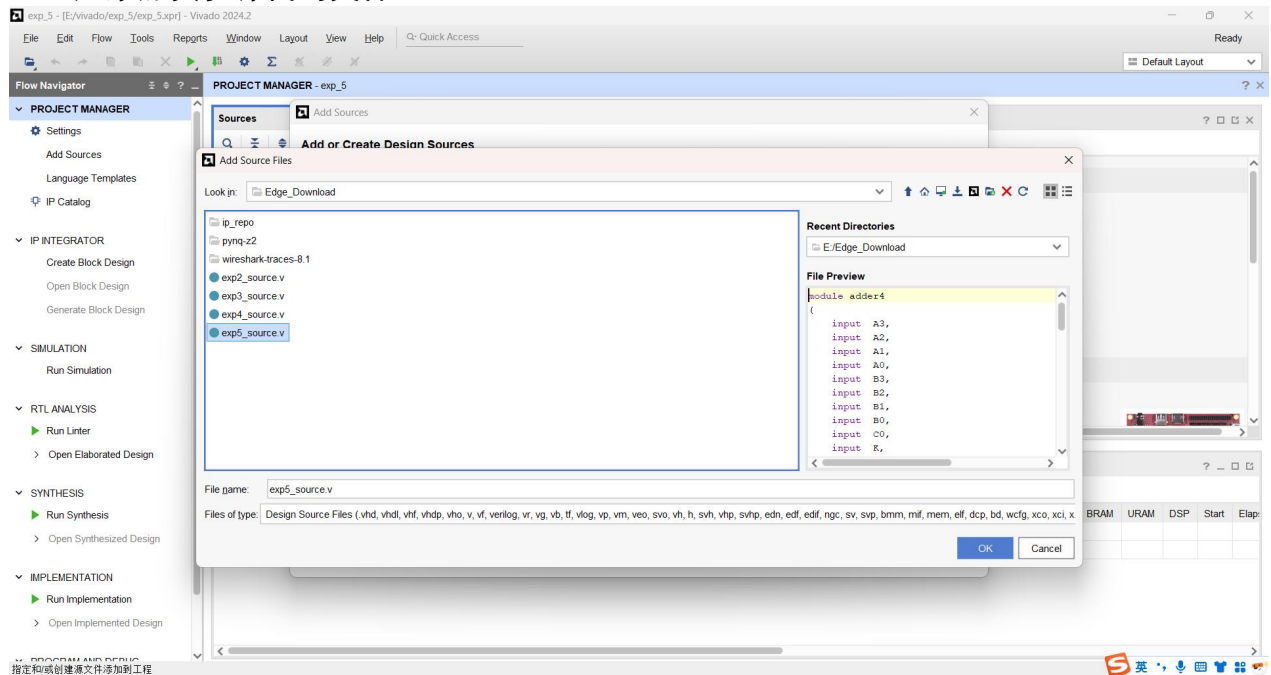
(2) 添加实验环境：进入 FPGA 在线实验环境，点击右上角项目材料下载实验源代码和希冀 ip 核到本地并解压。



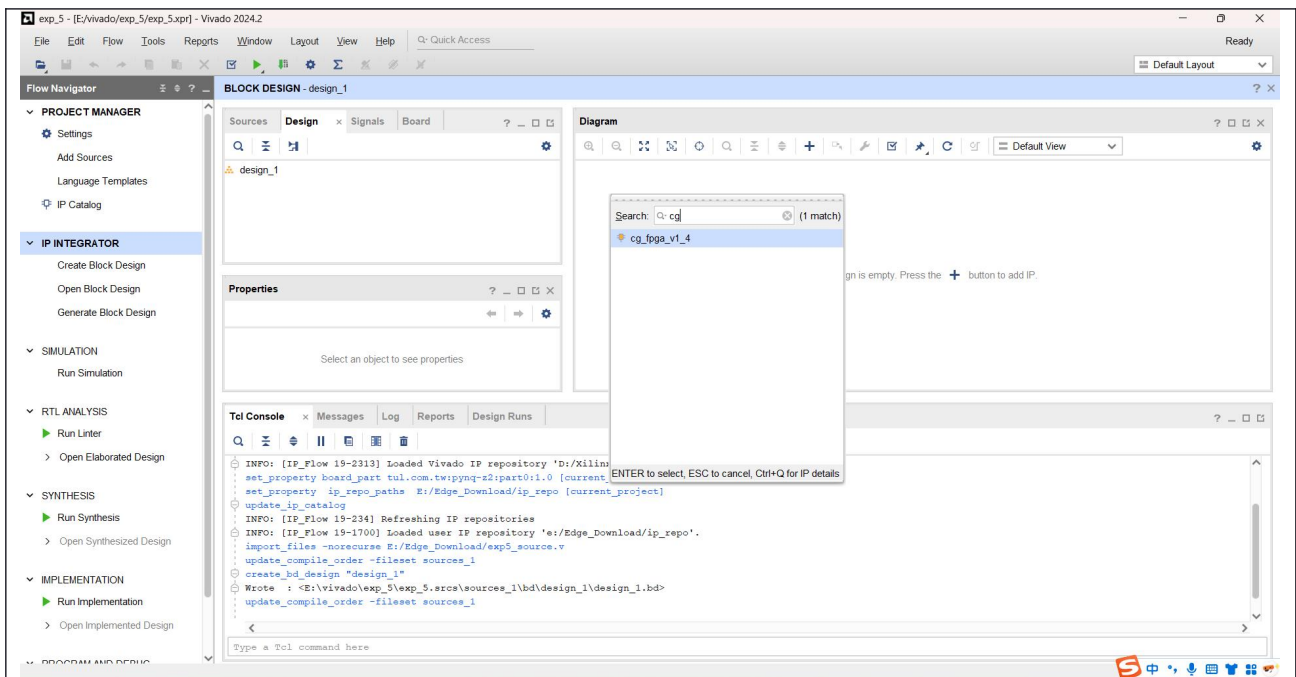
(3) 在 Vivado 项目中，点击 Settings→IP→Repository，将上一步解压后的 ip\_repo 文件夹的位置添加进 IP 搜索目录。



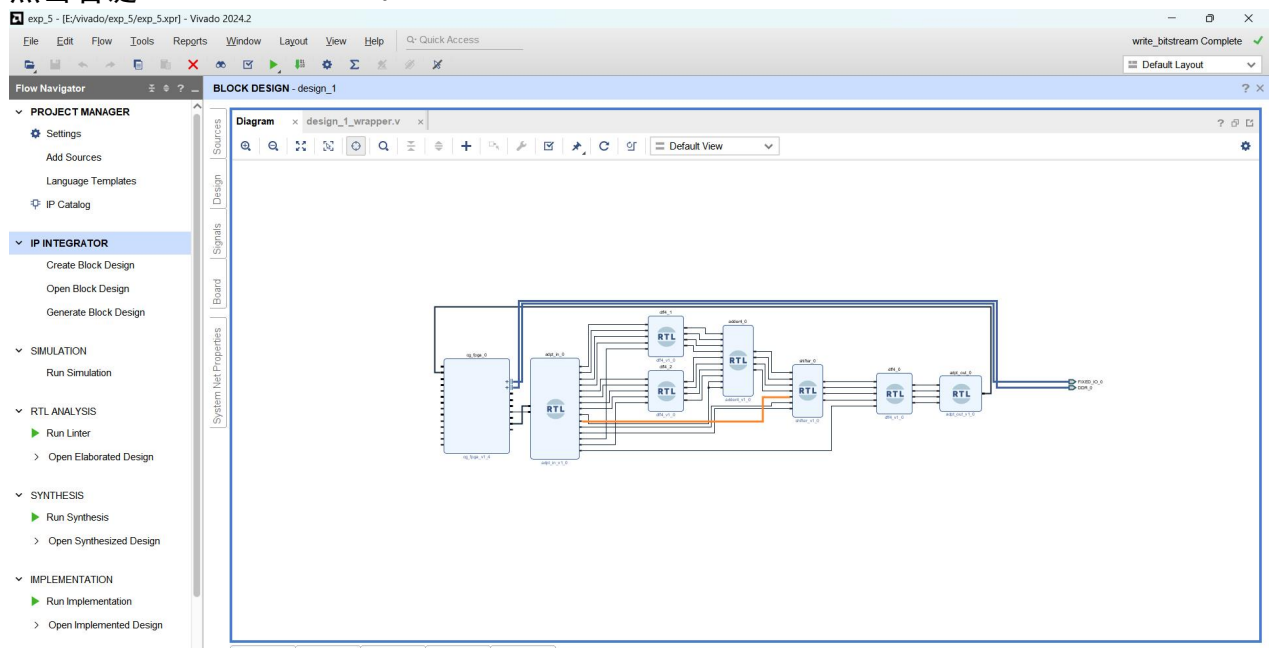
(4) 点击 Sources 窗口中的+, 选择 Add or create design sources → Next → Add File, 添加实验源代码文件。



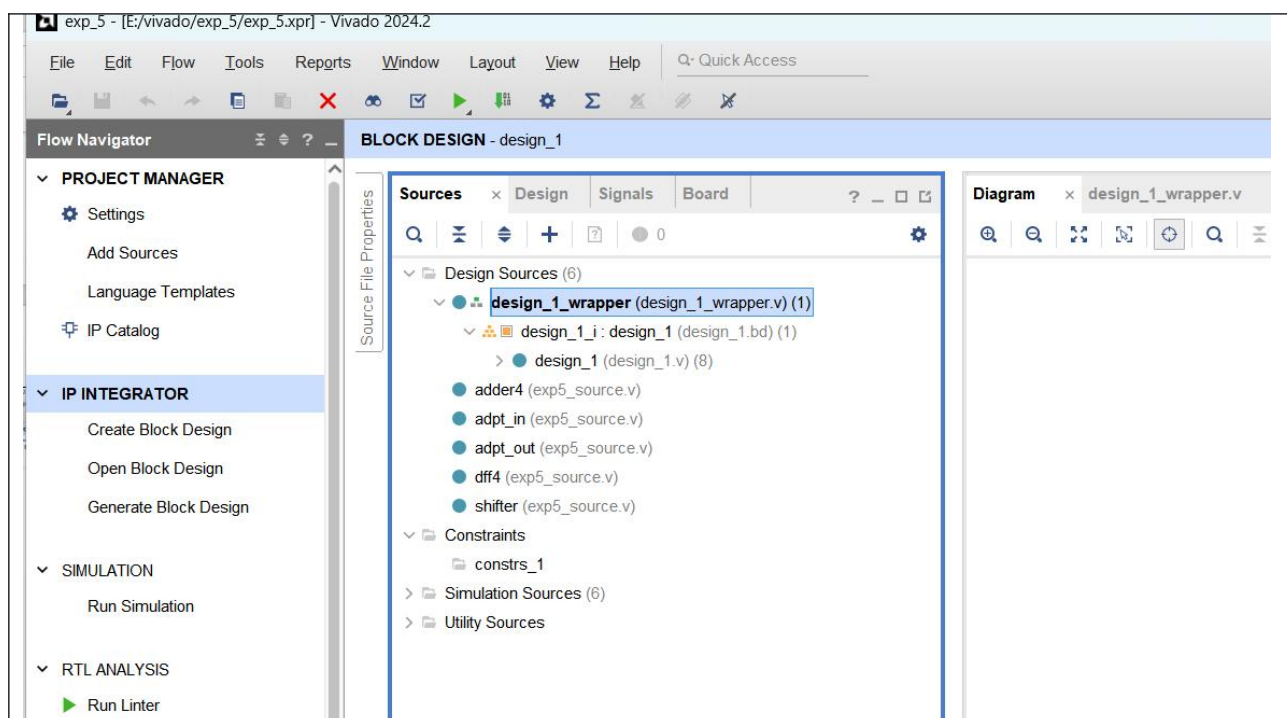
(5) 点击 Create Block Design 创建一个新的顶层设计, 随后点击添加 IP 核按钮, 添加 cg\_fpga IP。



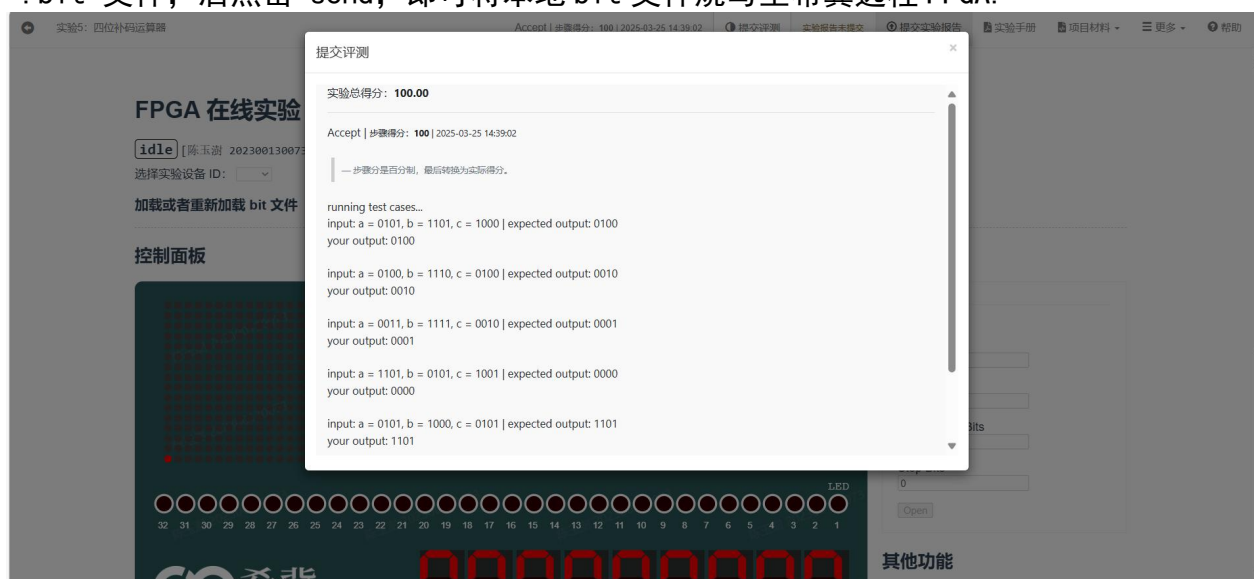
(6) 在 Sources 窗口下的 Design sources 中，根据电路图拖拽相应模块，完成原理图的输入。将输入的模块与 cg\_fpga 如下图连接，并选择 cg\_fpga 模块上的 FIXED\_IO 和 DDR，点击右键→Make External。



(7) 右击 Sources 下顶层设计图标→Create HDL Wrapper，待 Wrapper 正确生成后，点击左下方 Generate Bitstream，开始综合并生成 bit 文件。注意：综合前 wrapper 模块应被设置为顶层（加粗表示），若自动设置错误，需右击 wrapper 图标点击 Set as Top 手动设置。



(8) 通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 \*.bit 文件，后点击 send，即可将本地 bit 文件烧写至希冀远程 FPGA。

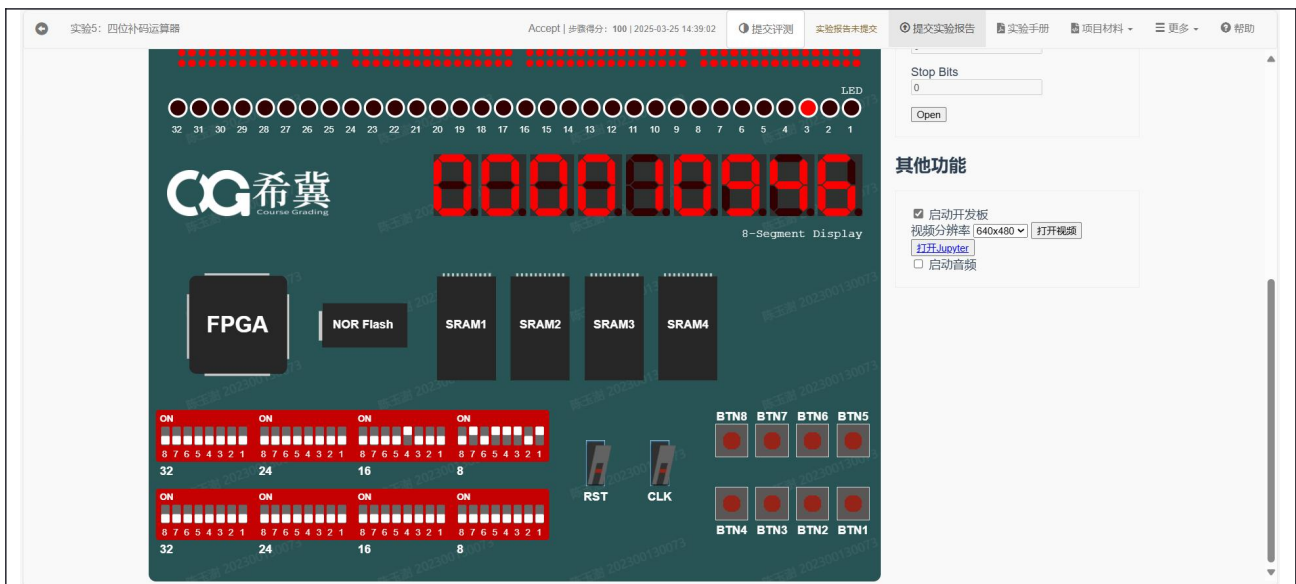


利用开关与指示灯测试：加法、减法、加 1、左移、右移、直传等功能，并分析结果正确性。

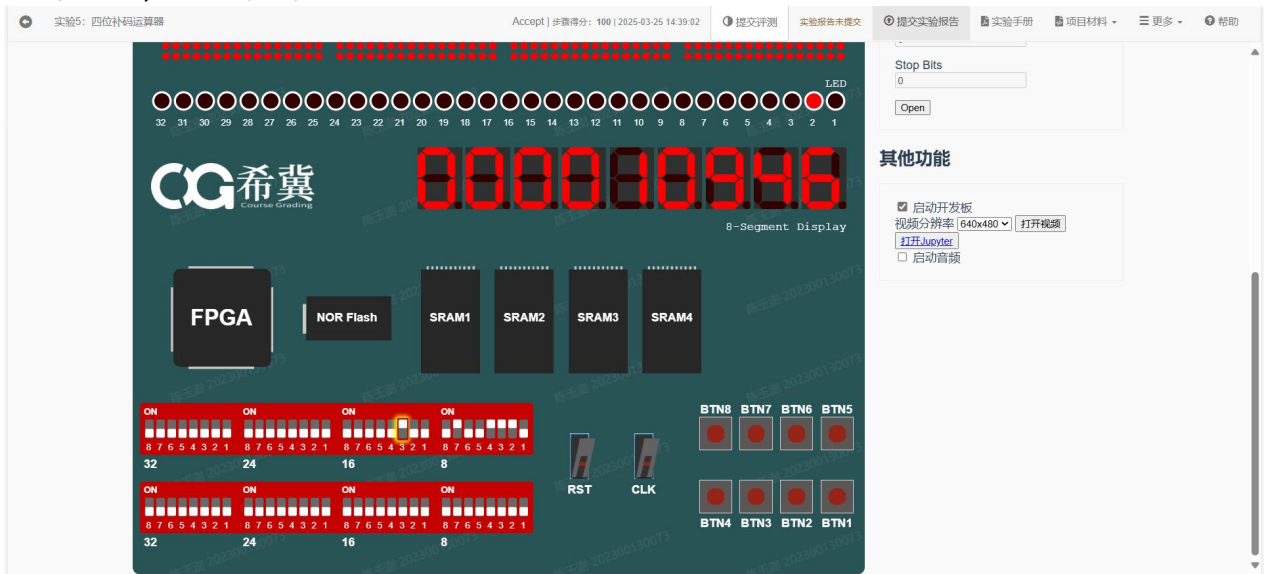
A3-A0 管脚分别绑定在上排拨码开关 4-1，B3-B0 管脚分别绑定在上排拨码开关 8-5，LM, DM, RM, K 管脚绑定在上排拨码开关 12-9 上，Y3-Y0 管脚分别绑定在 LED4-1 上。

首先！当输入为 a=0101，b=1101，c=1000 时，此时，LM=1，a+b= (1) 0010，左移 1 位，得到 0100，LED3 亮灯。

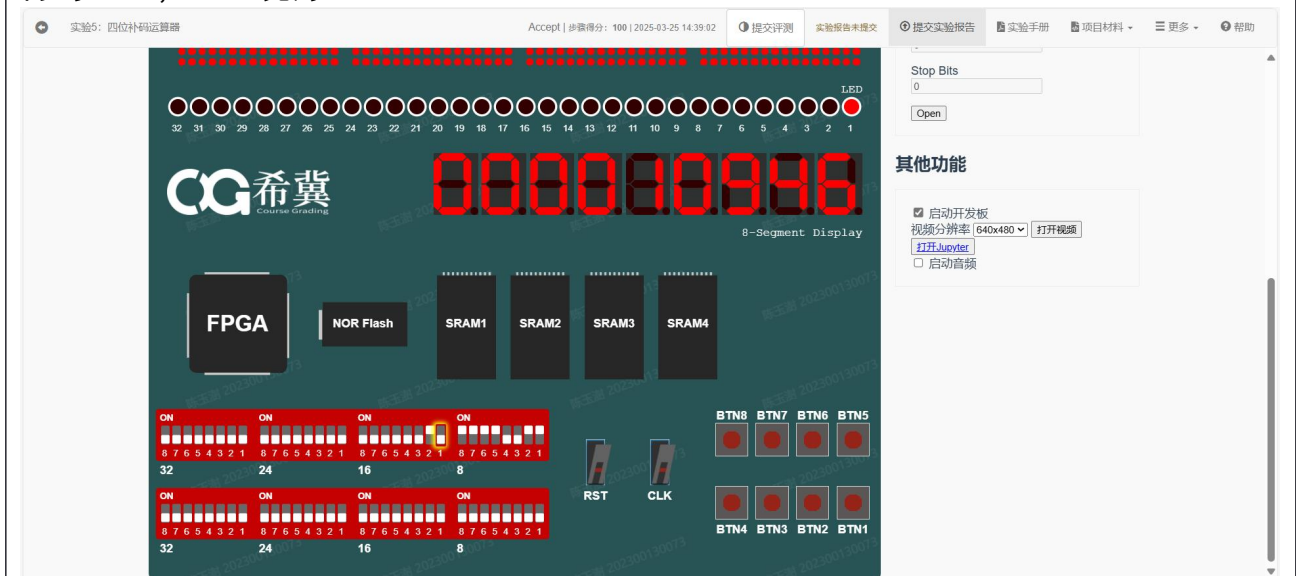




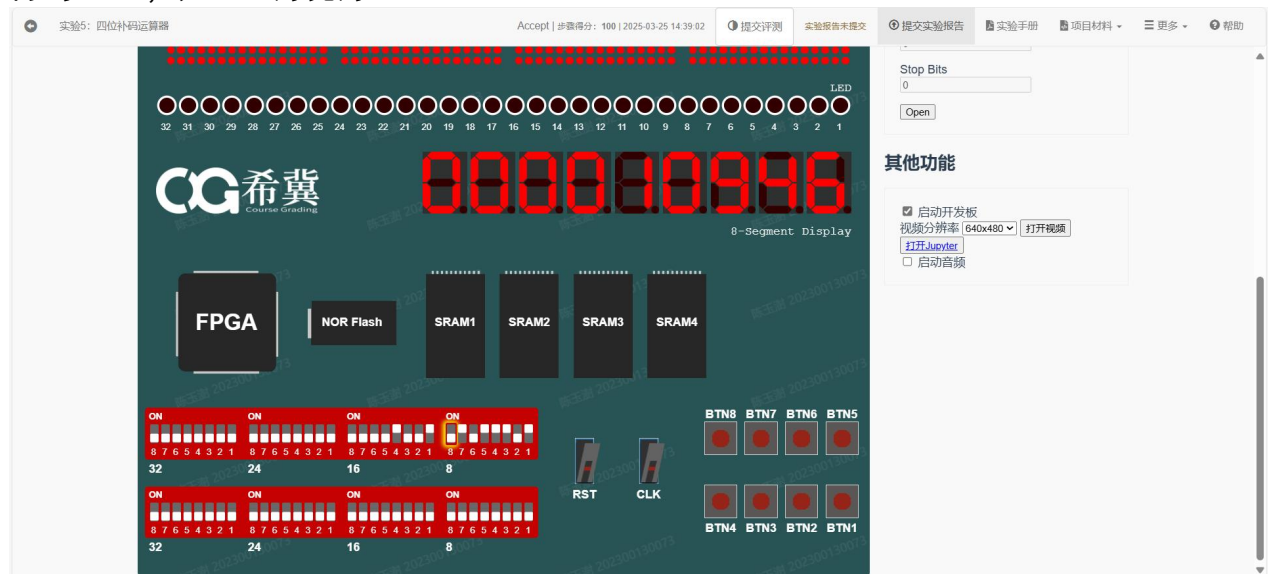
还有，当输入为  $a=0100$ ,  $b=1110$ ,  $c=0100$  时，此时， $DM=1$ ,  $a+b=(1)0010$ ，直接传送，得到 0010，LED2 亮灯。



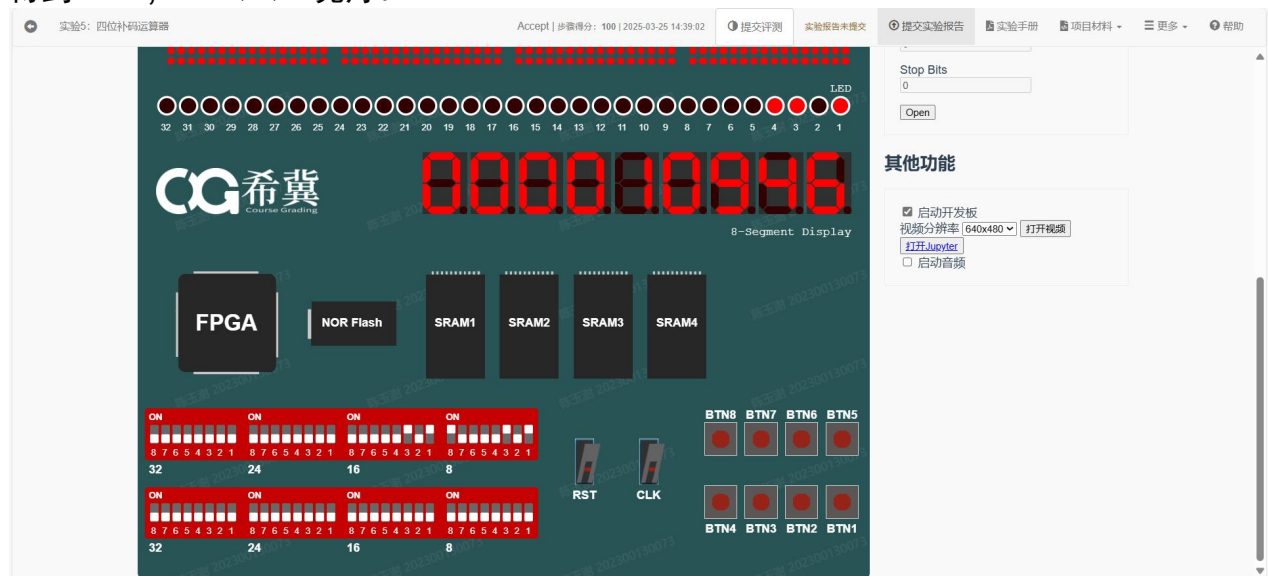
还有，当输入为  $a=0011$ ,  $b=1111$ ,  $c=0010$  时，此时， $RM=1$ ,  $a+b=(1)0010$ ，右移 1 位，得到 0001，LED1 亮灯。



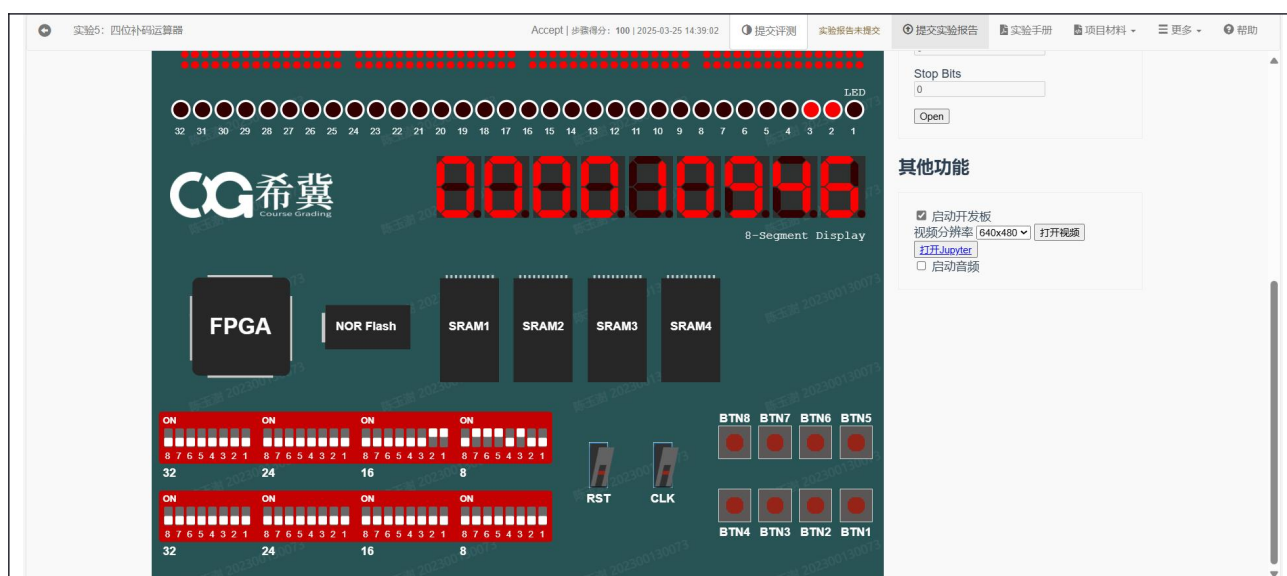
还有，当输入为  $a=1101$ ,  $b=0101$ ,  $c=1001$  时，此时， $LM=1$ ,  $K=1$ ,  $a-b=1000$ ，左移 1 位，得到 0000，无 LED 灯亮灯。



还有，当输入为  $a=0101$ ,  $b=1000$ ,  $c=0101$  时，此时， $DM=1$ ,  $K=1$ ,  $a-b=1101$ ，直接传送，得到 1101，LED4/3/1 亮灯。



还有，当输入为  $a=0100$ ,  $b=0111$ ,  $c=0011$  时，此时， $RM=1$ ,  $K=1$ ,  $a-b=0011$ ，右移 1 位，得到 0110，LED3/2 亮灯。



### 结论分析与体会：

这次的实验让我对于补码运算器有了一个深入的理解。同时深入理解了补码运算与移位操作的硬件实现逻辑。此外，我更加熟练地掌握了 Vivado 中模块连接、IP 核配置及硬件调试的流程。成功设计并验证了四位补码运算器，实现了加减、移位等功能。

就同时触发多个控制信号（如 LM 和 RM）时，输出结果异常问题处理的：

首先通过 Vivado 综合报告定位到控制逻辑中的多路选择器未设置优先级；接着，进行逻辑调整：修改 Verilog 代码，为控制信号设置优先级（LM>DM>RM），确保同一时刻仅执行一种操作；最后重新生成 Bitstream 并烧录，测试所有控制组合，信号冲突消除，功能恢复正常。