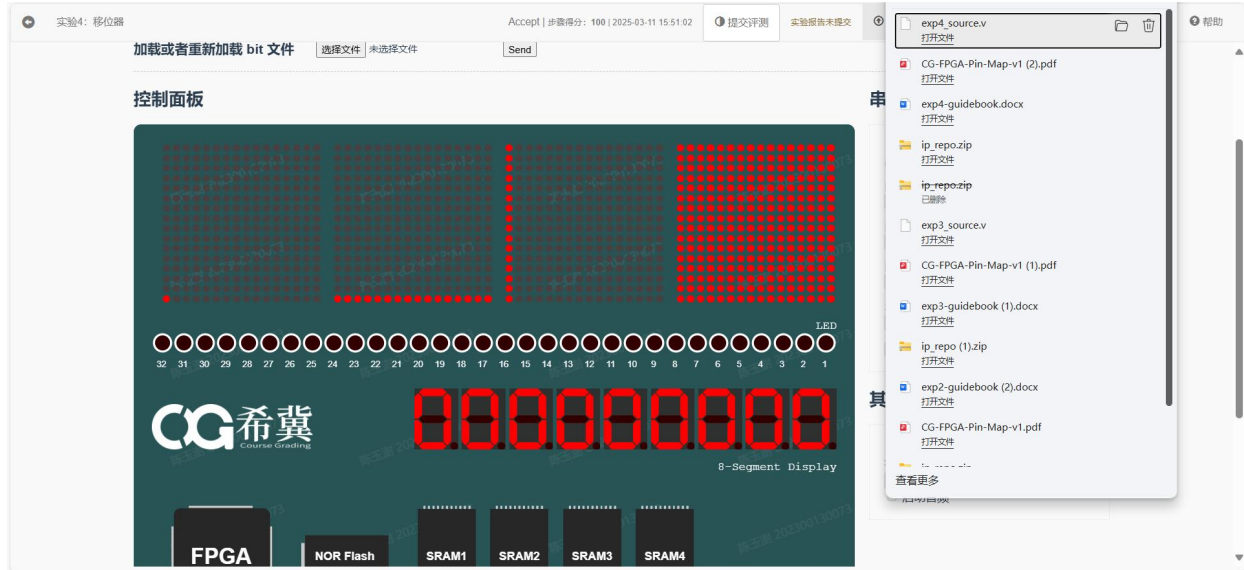
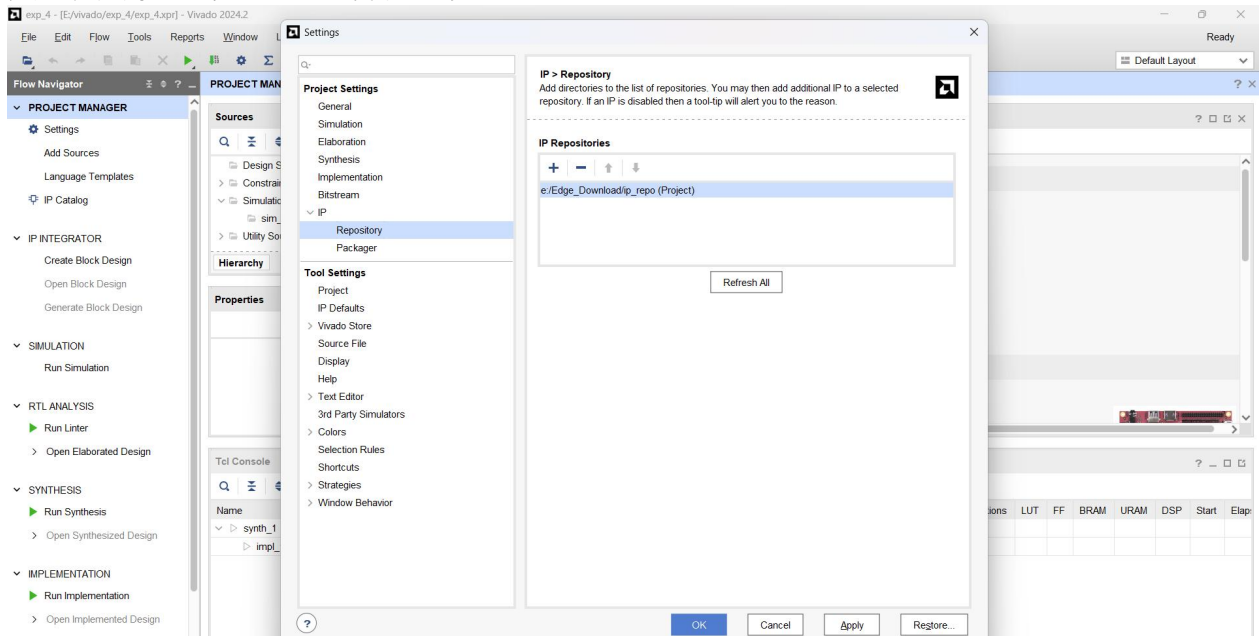




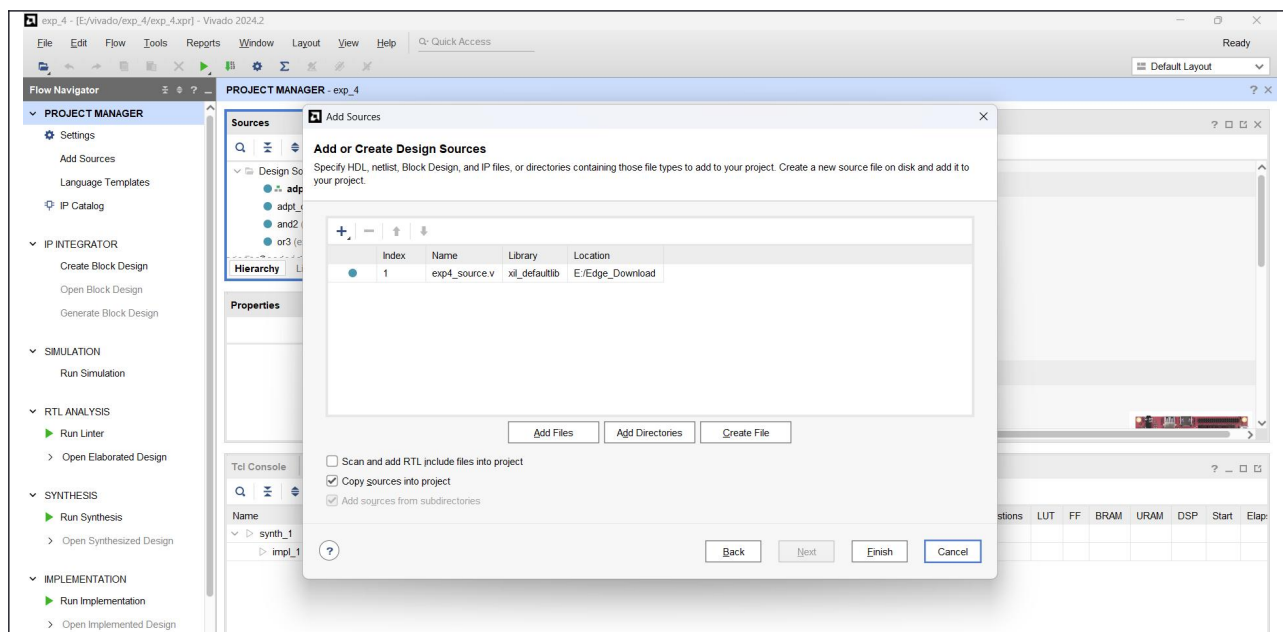
冀 ip 核到本地并解压。



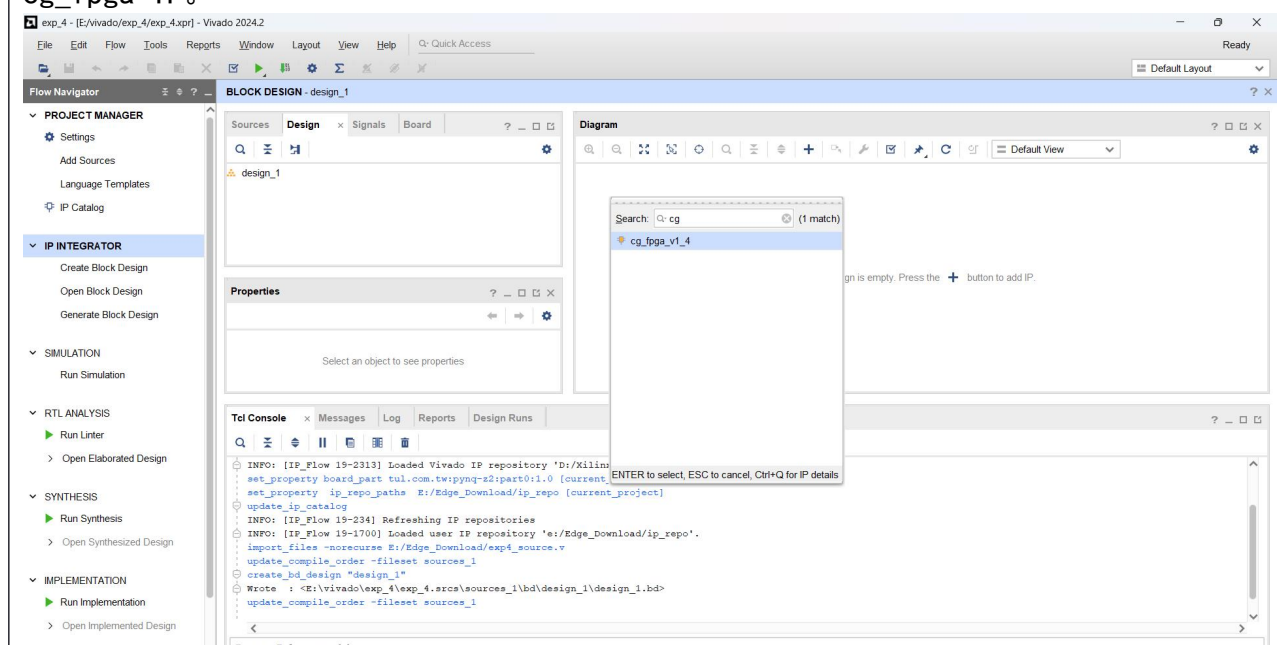
(3) 在 Vivado 项目中，点击 Settings→IP→Repository，将上一步解压后的 ip_repo 文件夹的位置添加进 IP 搜索目录。



(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。

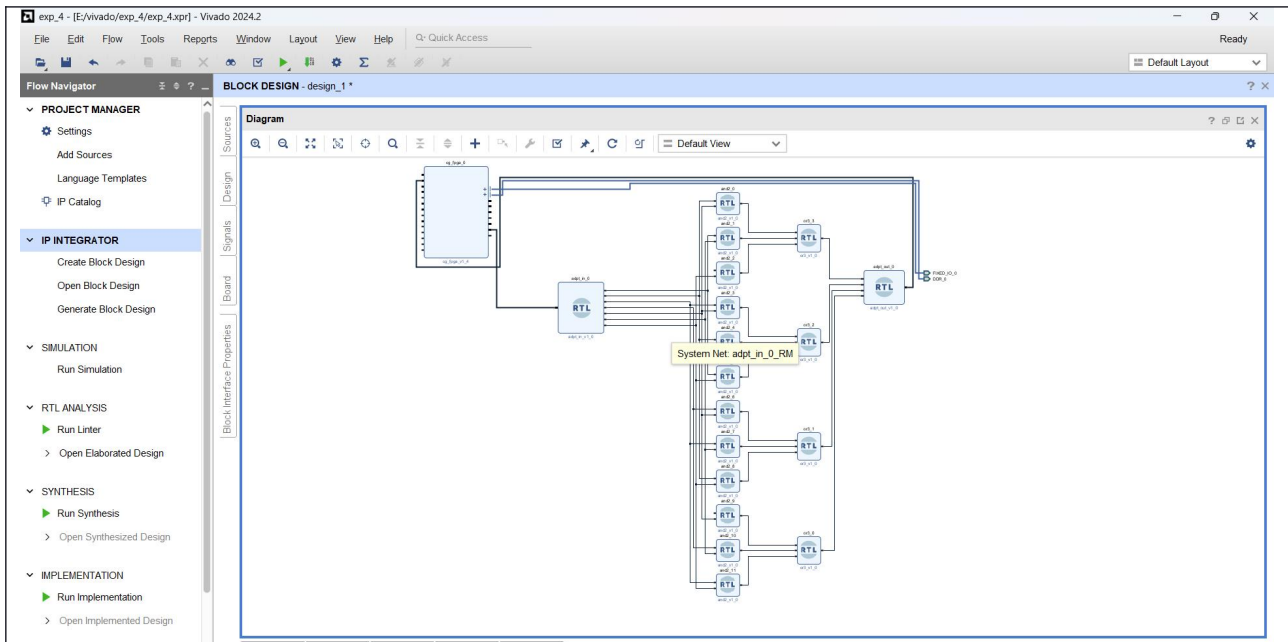


(5) 点击 Create Block Design 创建一个新的顶层设计，随后点击添加 IP 核按钮，添加 cg_fpga IP。

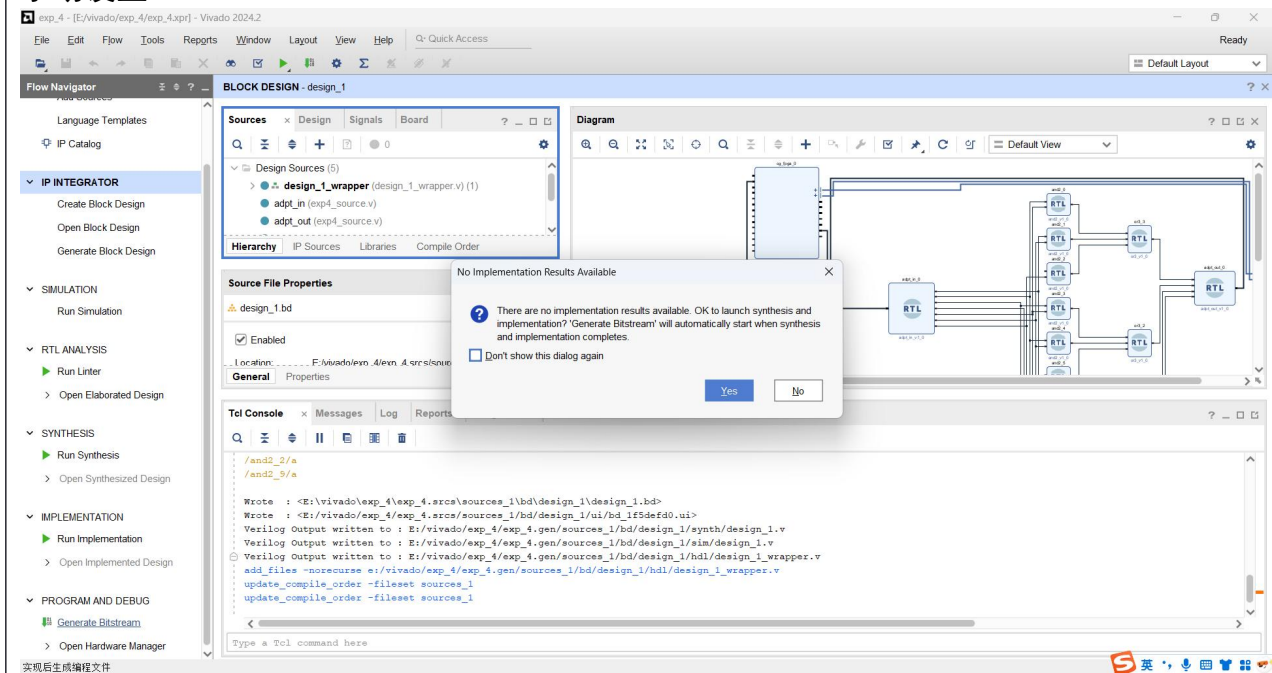


(6) 在 Sources 窗口下的 Design sources 中，根据电路图拖拽相应模块，完成原理图的输入。

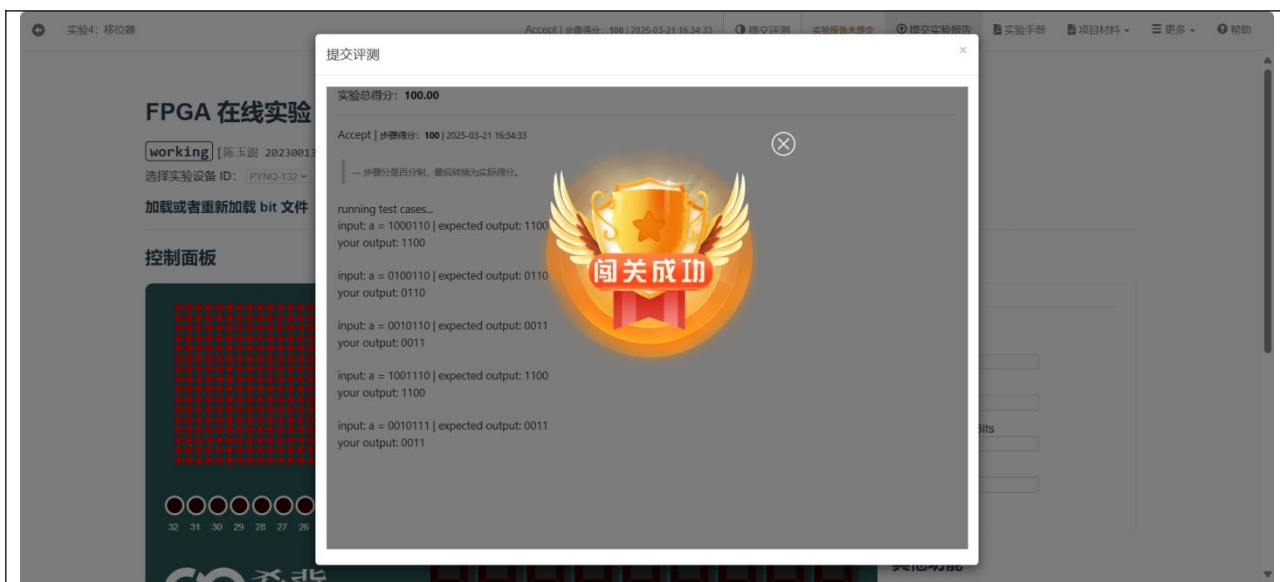
(7) 将输入的模块与 cg_fpga 如下图连接，并选择 cg_fpga 模块上的 FIXED_IO 和 DDR，点击右键→Make External。



(8) 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。



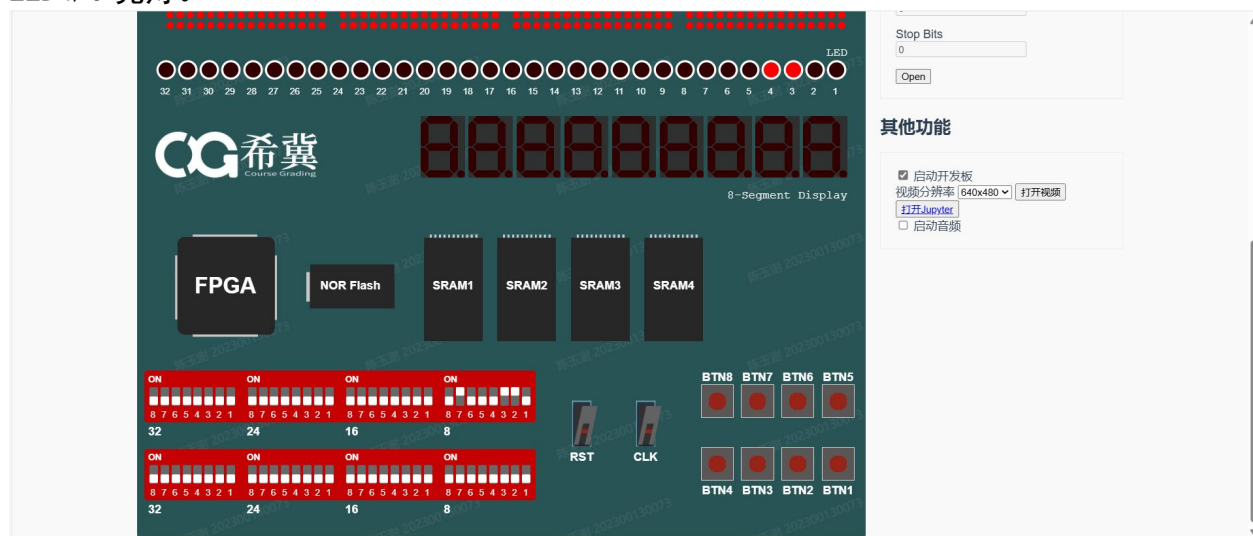
(9) 通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 *.bit 文件，后点击 send，即可将本地 bit 文件烧写至希冀远程 FPGA。



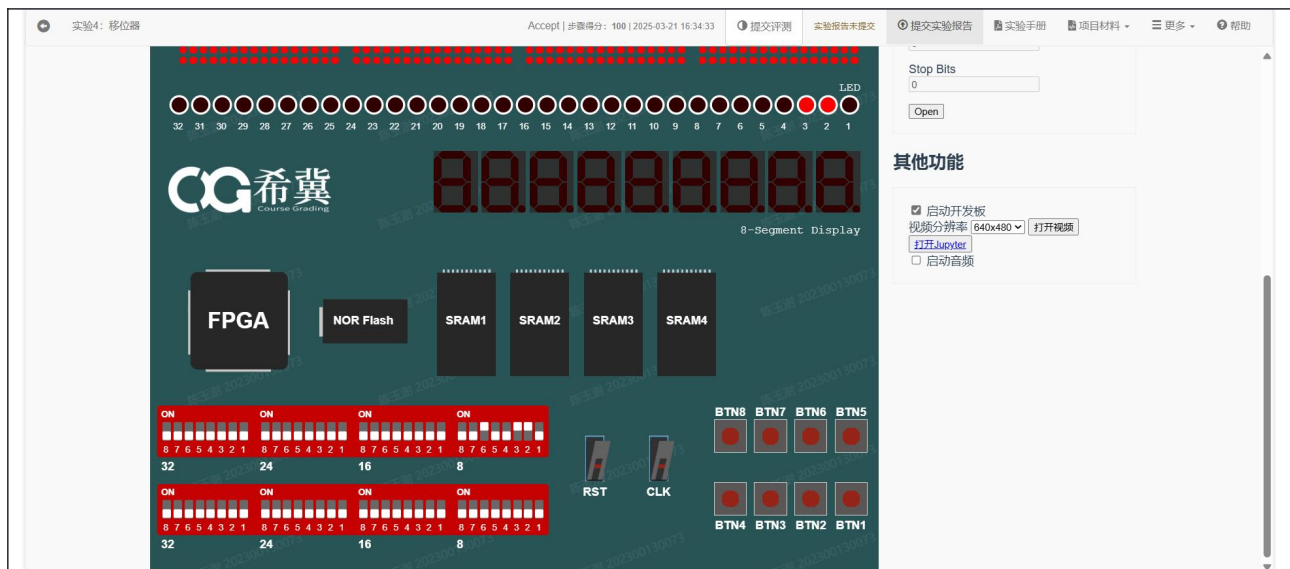
设置 a3-a0 信号为组成任意 4 位数, 在 LM, DM, RM 的作用下分别观察 q3-0 信号的变化, 并分析其正确性。

LM, DM, RM 管脚分别绑定在上排拨码开关 7-5, a3-a0 管脚分别绑定在上排拨码开关 4-1, q3-q0 管脚绑定在 LED4-1 上。

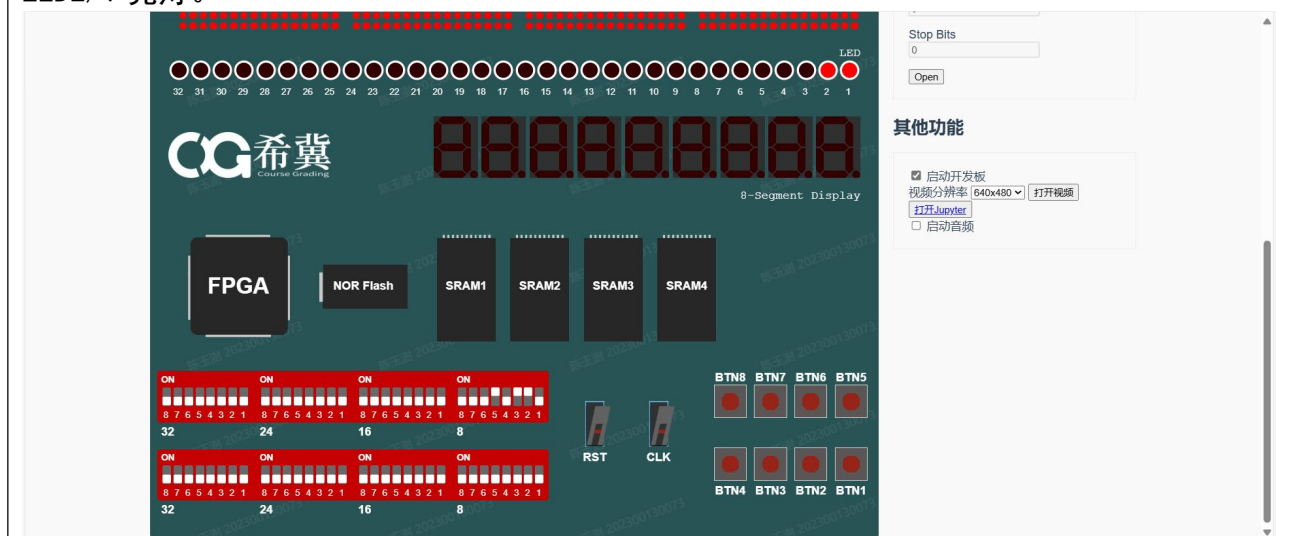
首先! 输入为 1000110 时, 此时, LM=1, a3-a0=0110, 左移 1 位, q3-q0 应得到 1100, LED4/3 亮灯。



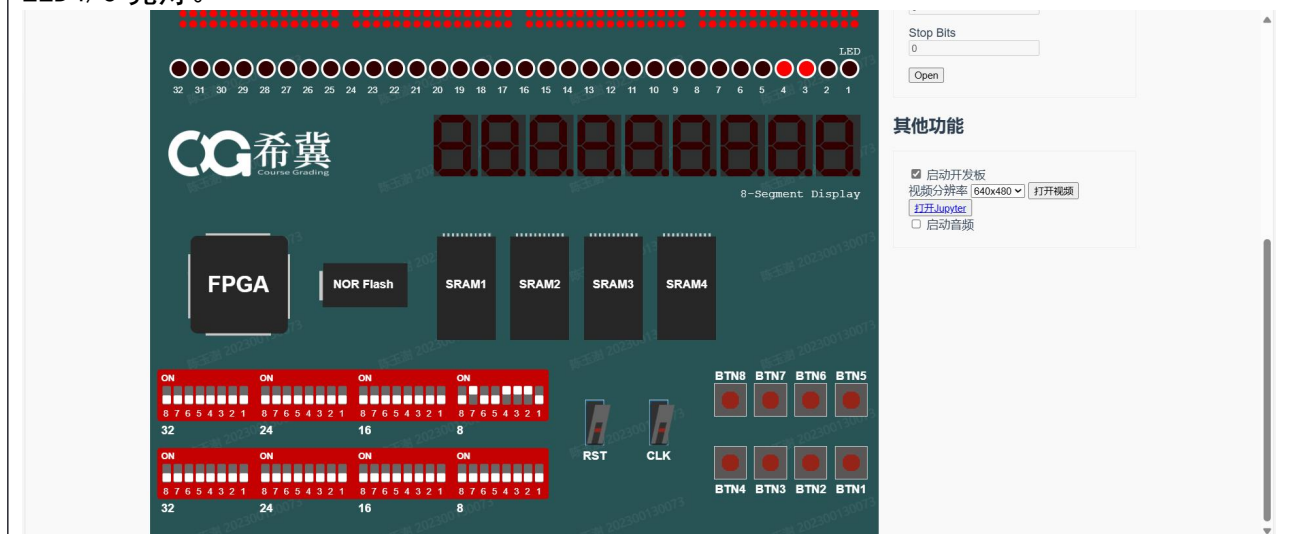
还有, 当输入为 0100110 时, DM=1, a3-a0=0110, 直送, q3-q0 应得到 0110, LED3/2 亮灯。



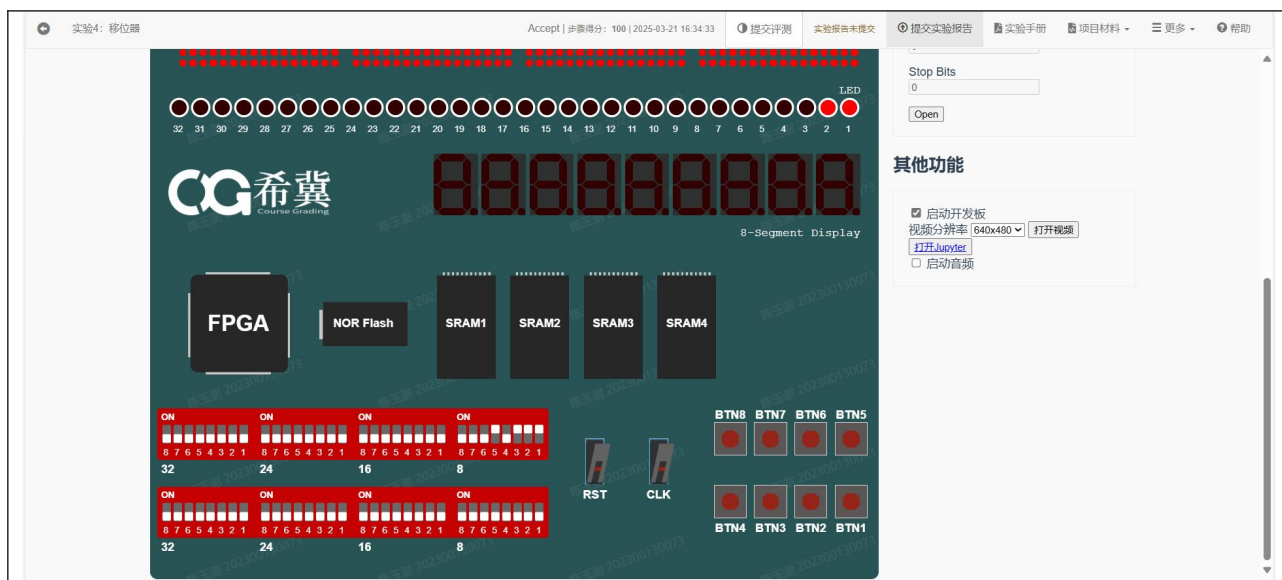
还有，当输入为 0010110 时，此时，RM=1， $a_3-a_0=0110$ ，右移 1 位， q_3-q_0 应得到 0011，LED2/1 亮灯。



还有，当输入为 1001110 时，此时，LM=1， $a_3-a_0=1110$ ，左移 1 位， q_3-q_0 应得到 1100，LED4/3 亮灯。



还有，当输入为 0010111 时，此时，RM=1， $a_3-a_0=0111$ ，右移 1 位， q_3-q_0 应得到 0011，LED2/1 亮灯。



结论分析与体会：

这次的实验让我对于移码有了了一个深入的理解。同时也使我提高了对于 Vivado 软件的应用熟练度，对移码器电路有了更加深入、深刻的认识和掌握，强化了对左移补零、右移空位处理等关键问题的认识。在调试过程中，通过解决译码器信号冲突问题，提升了逻辑分析和硬件调试能力。最终成功完成功能验证，对 FPGA 开发流程和时序约束有了更深刻的实践体会，为后续数字系统设计奠定了基础。

就信号冲突导致移位功能异常的问题处理的：

通过 Vivado 综合报告发现逻辑竞争，手动检查代码发现多路选择器优先级错误。调整控制信号优先级顺序后，重新综合生成 Bitstream，经硬件测试移位功能恢复正常。