

计算机组成与设计 课程实验报告

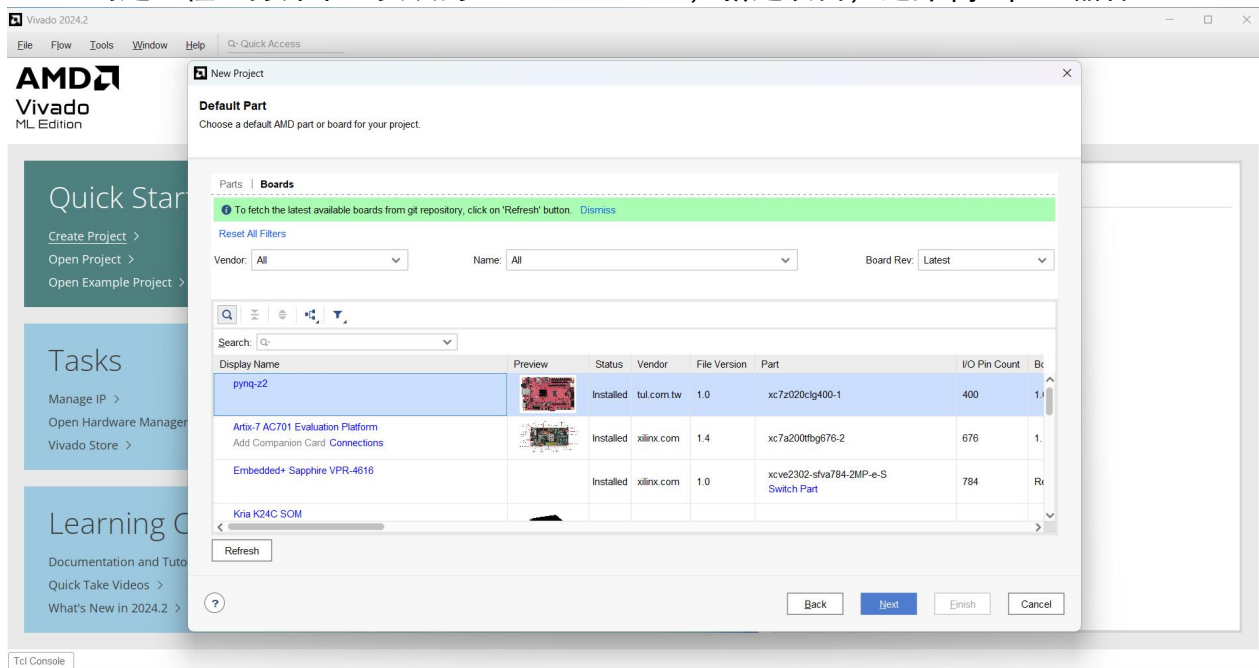
学号：202300130073	姓名：陈玉澍	班级：计科 23 级 1 班
实验题目： 实验 10 ROM 实验		
实验学时：2		实验日期：2025. 4. 29
实验目的： 1、掌握 Vivado 中 ROM 的设置，作为只读存储器 ROM 的工作特性和配置方法； 2、用文本编辑器编辑 coe 文件配置 ROM，学习以 coe 格式文件加载于 ROM 中； 3、在初始化存储器编辑窗口编辑 coe 文件配置 ROM； 4、验证 FPGA 中 ROM 的功能。		
实验软件和硬件环境： 软件环境： Vivado 软件、FPGA 实验平台 硬件环境： 1. 实验室台式机 2. FPGA 服务器，PYNQ-Z2 开发板		
实验原理和方法： <p>.coe 文件格式是 Xilinx 公司开发的，用于描述 FPGA 中存储器初始化数据的文件格式。它通常用于将初始化数据加载到 Xilinx FPGA 的内置存储器中，如 Block RAM (BRAM) 或者只读存储器 (ROM)。</p> <p>.coe 文件格式简单且易于解析，通常由以下几个部分组成： 一个典型的 .coe 文件包含两部分内容：</p> <ol style="list-style-type: none"> Memory Initialization Radix (基数声明)： <ul style="list-style-type: none"> 指定数据和地址的进制表示。 通常有以下几种选择：<i>bin</i> (二进制)，<i>hex</i> (十六进制)，<i>dec</i> (十进制)，<i>oct</i> (八进制)。 <p>例如： <code>memory_initialization_radix=16;</code> 这表示文件中的数据是以十六进制形式表示的。</p> Memory Initialization Vector (数据向量声明)： <ul style="list-style-type: none"> 列出要加载到存储器中的数据值。 数据项之间用逗号分隔，最后一个数据项以分号结尾。 <p>例如： <code>memory_initialization_vector=</code> <code>3A, 1B, 2C, 0D, FF;</code> 这表示要将数据 3A、1B、2C、0D 和 FF 依次加载到存储器的地址 0 到 4 位置。</p> <p>.coe 文件的完整示例 以下是一个完整的 .coe 文件示例：</p>		

```
memory_initialization_radix=16;  
memory_initialization_vector=  
3A, 1B, 2C, 0D, FF, 10, 20, 30;
```

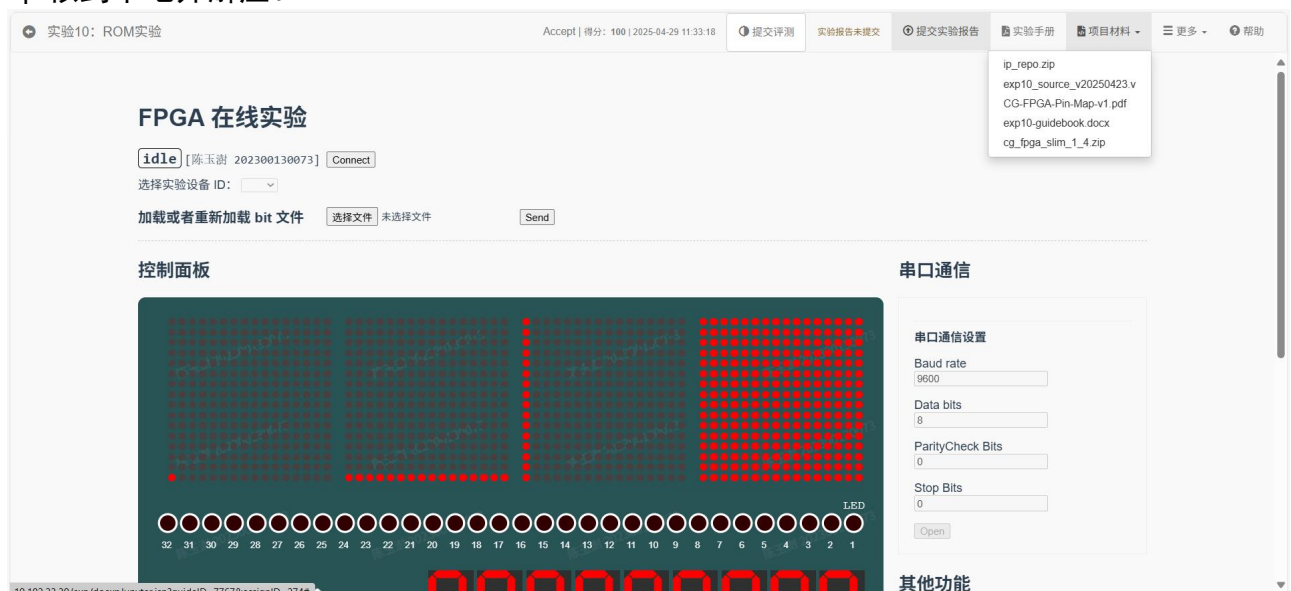
- `memory_initialization_radix=16`: 表示数据以十六进制表示。
- `memory_initialization_vector`: 具体的数据向量, 依次为 3A、1B、2C、0D、FF、10、20、30。

实验步骤:

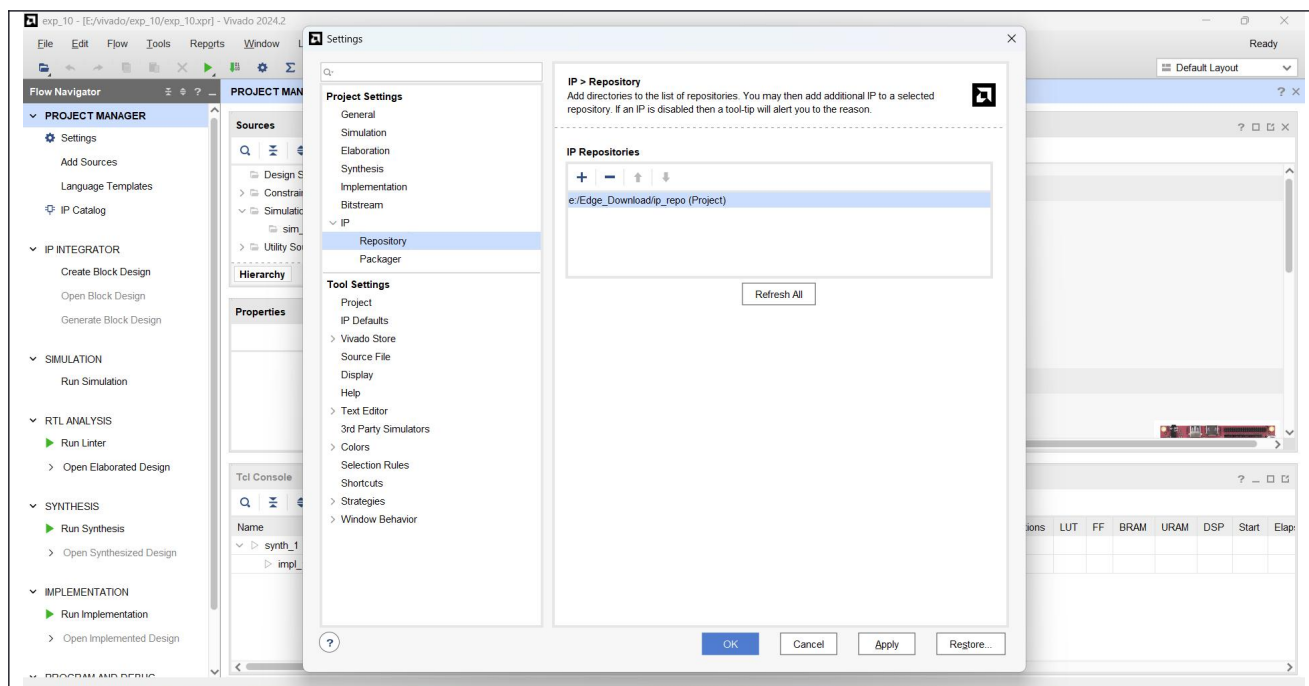
(1) 创建工程: 打开本地安装的 Vivado 2024.2, 新建项目, 选择 pynq-z2 器件。



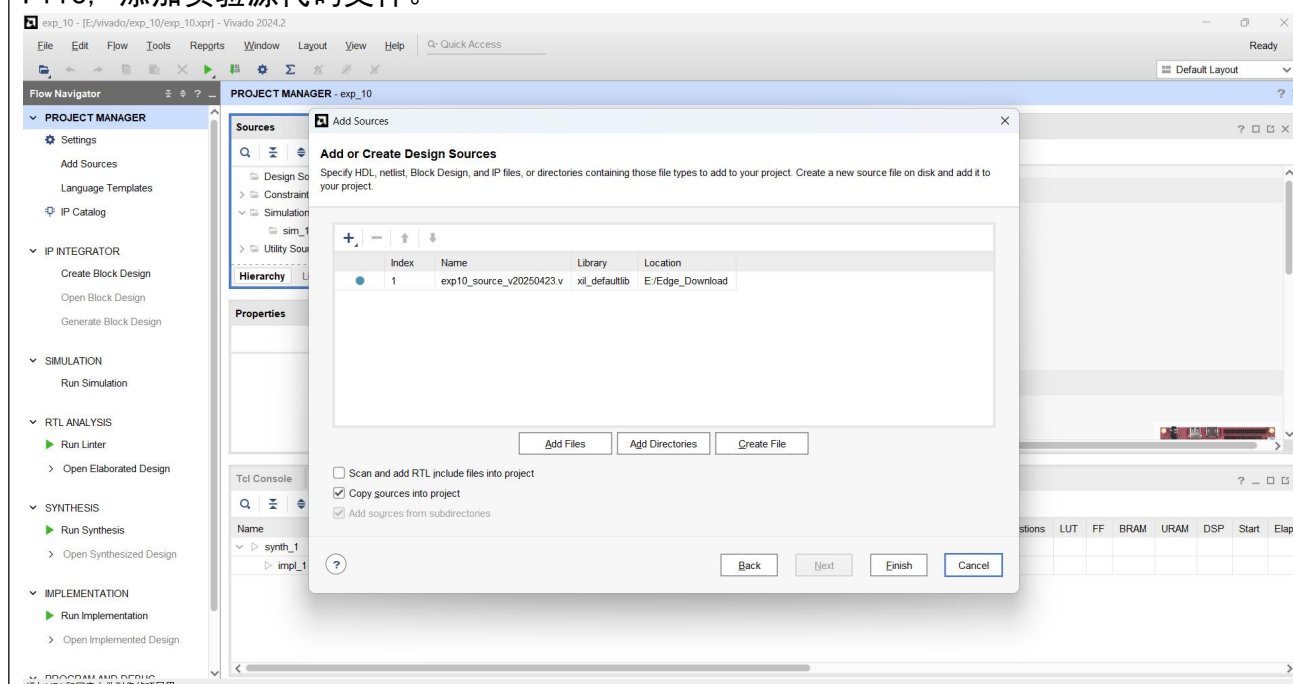
(2) 添加实验环境: 进入 FPGA 在线实验环境, 点击右上角项目材料下载实验源代码和希冀 ip 核到本地并解压。



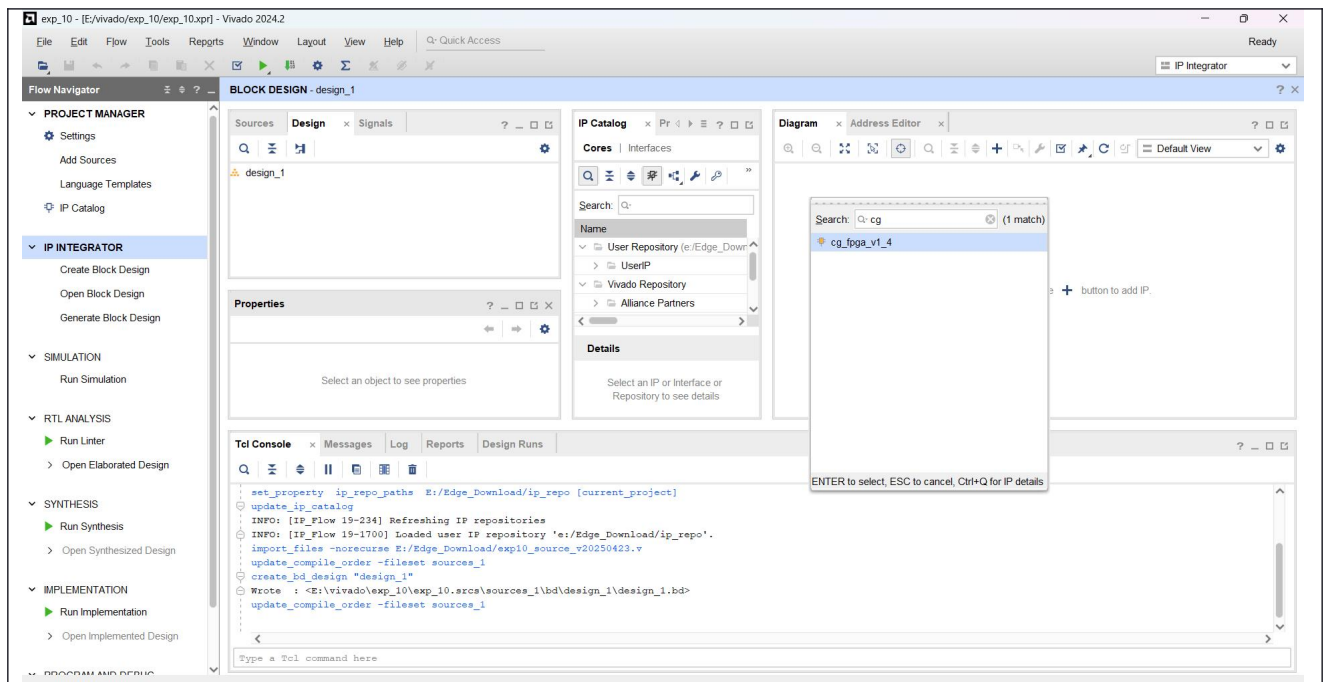
(3) 在 Vivado 项目中, 点击 Settings→IP→Repository, 将上一步解压后的 ip_repo 文件夹的位置添加进 IP 搜索目录。



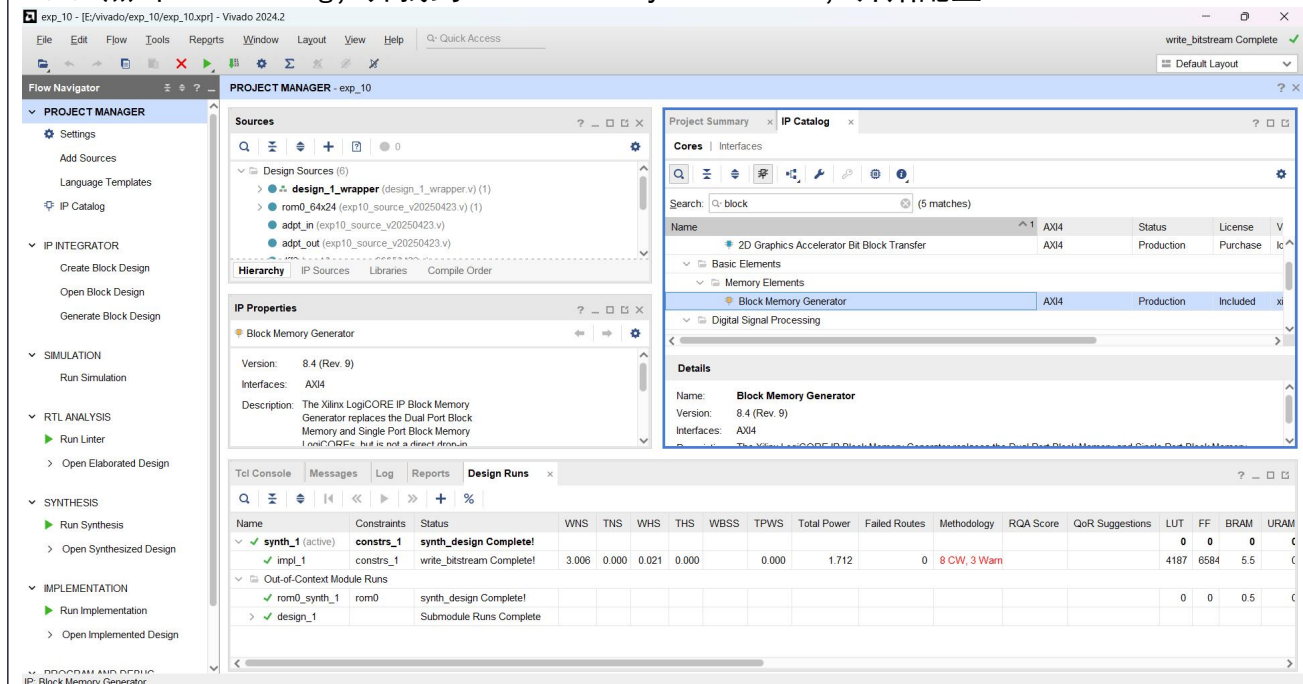
(4) 点击 Sources 窗口中的+, 选择 Add or create design sources → Next → Add File, 添加实验源代码文件。



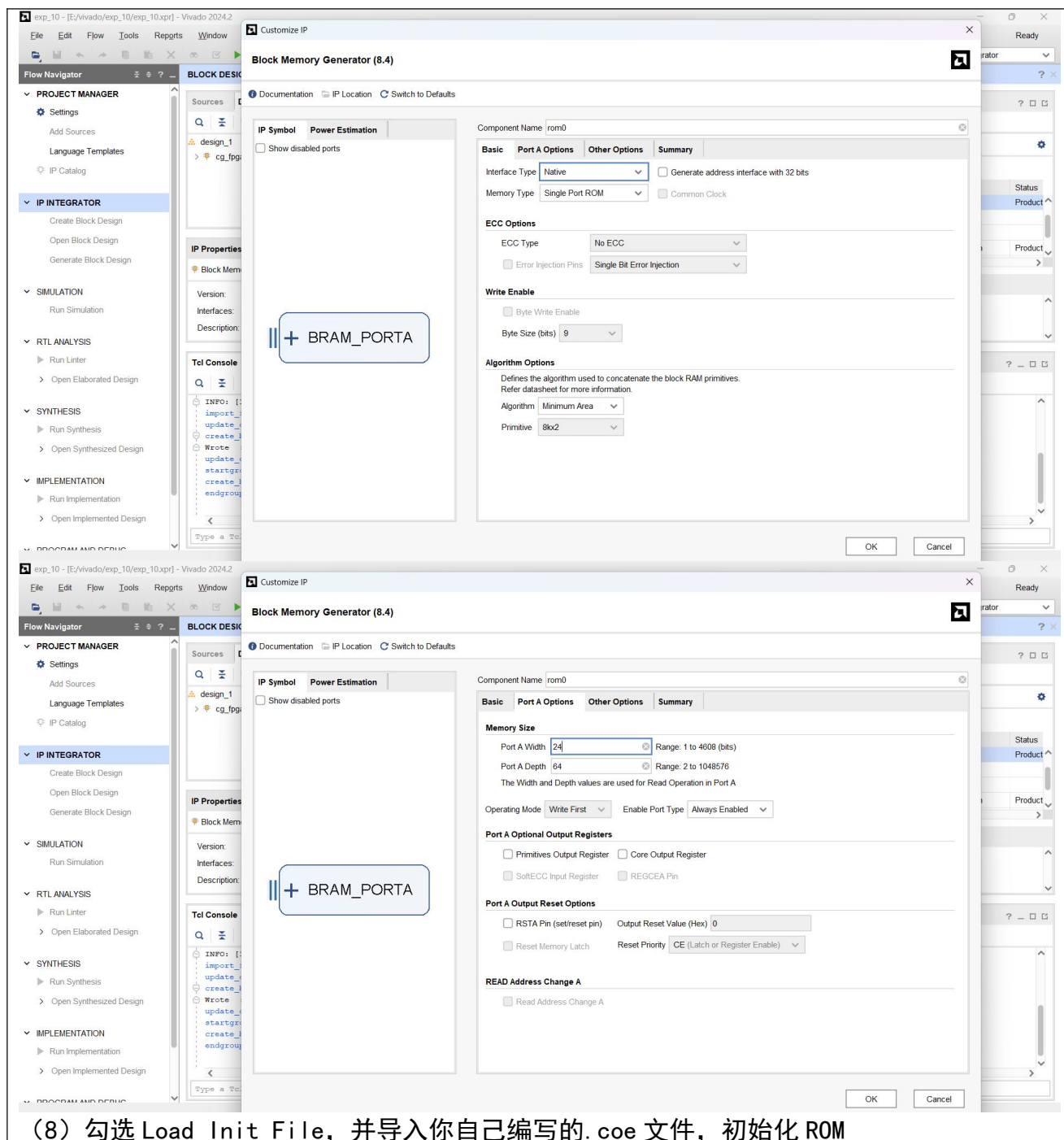
(5) 点击 Create Block Design 创建一个新的顶层设计，随后点击添加 IP 核按钮，添加 cg_fpga IP。



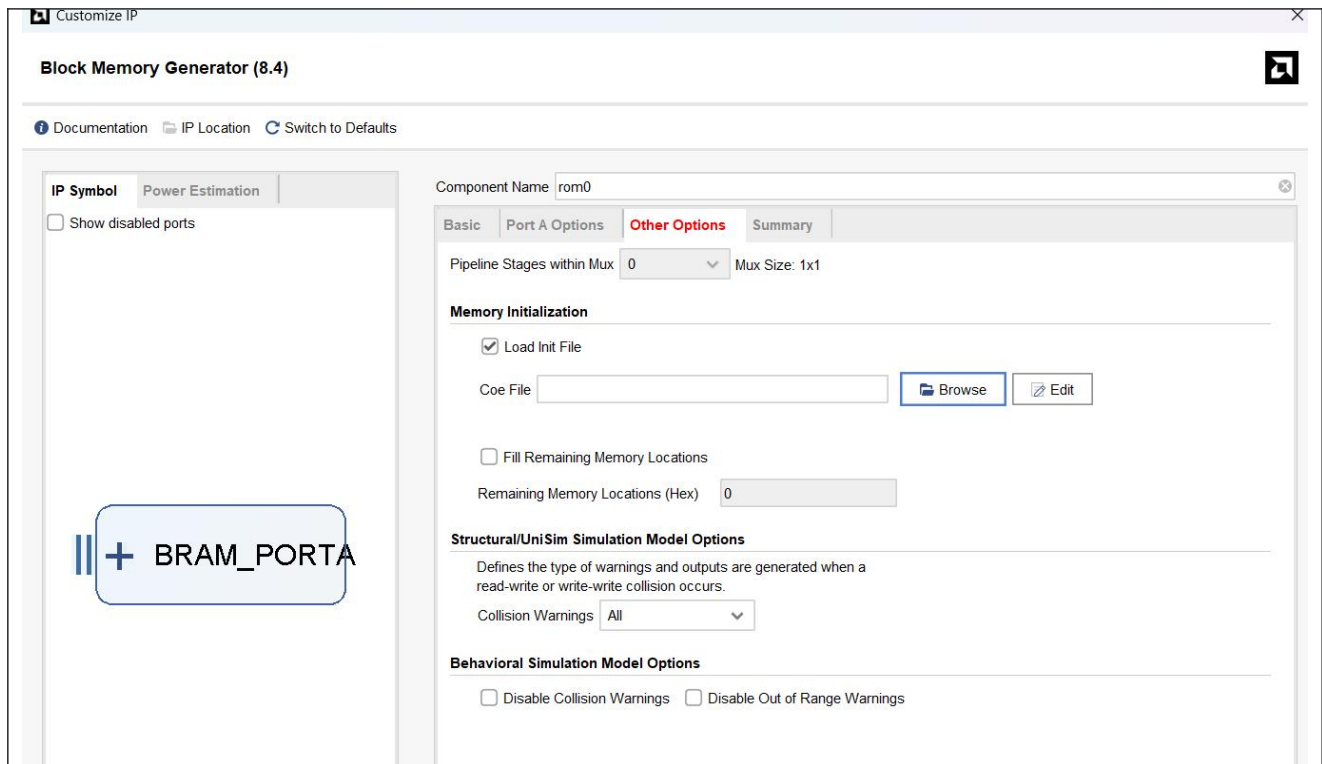
(6) 点击 IP Catalog, 并找到 Block Memory Generator, 开始配置 ROM



(7) 按照如下方法定制单口 ROM, 并命名为 rom0:

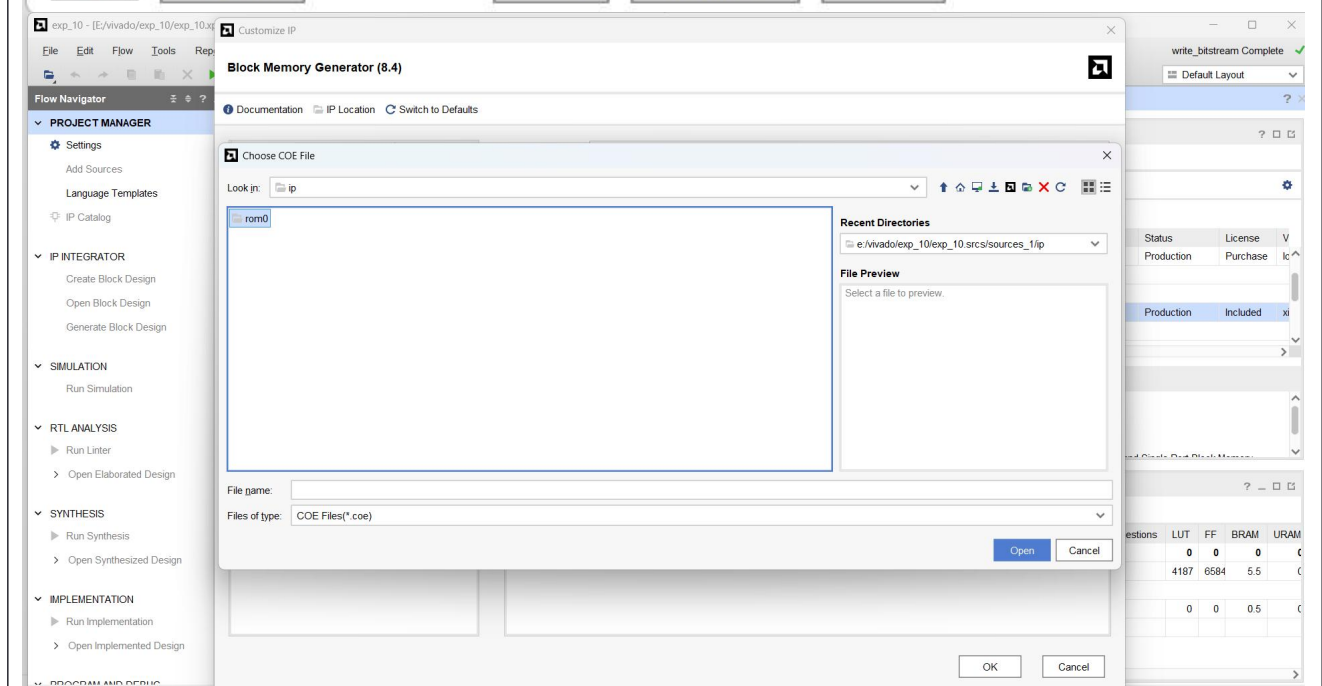
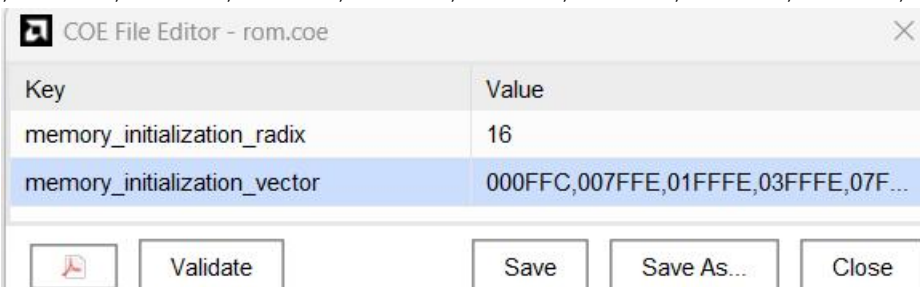


(8) 勾选 Load Init File, 并导入你自己编写的 .coe 文件, 初始化 ROM

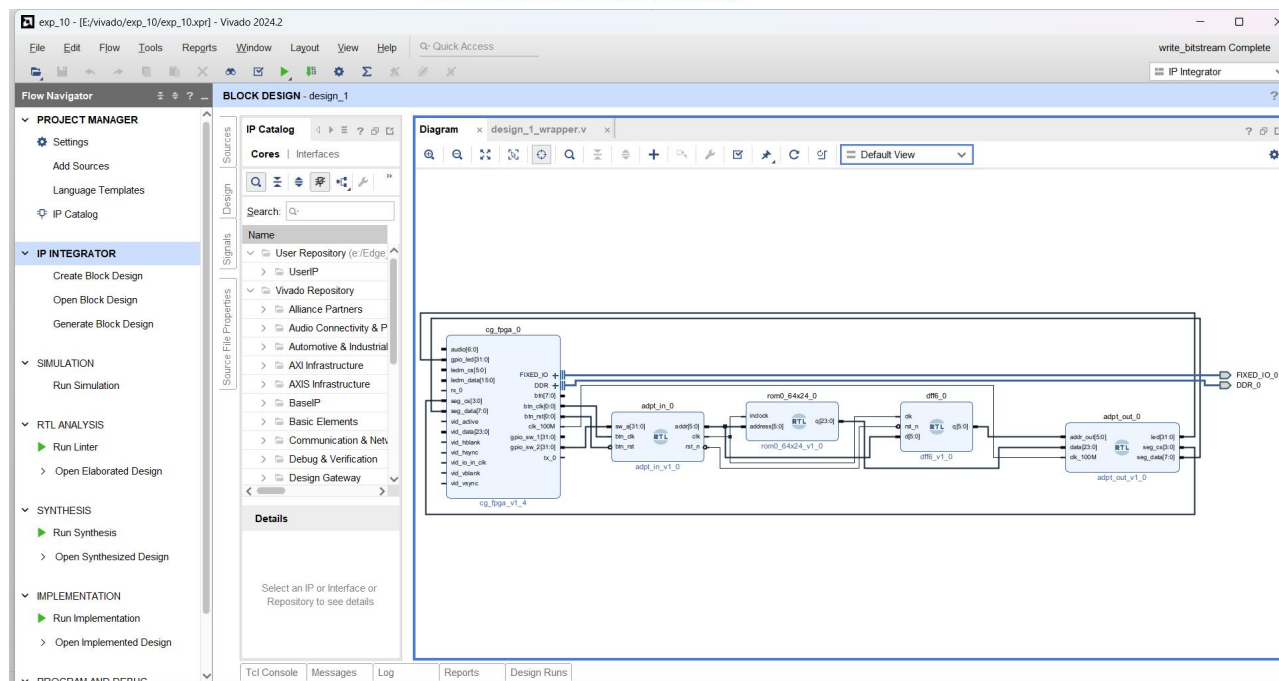


(9) 向 coe 文件中加入如下数据:

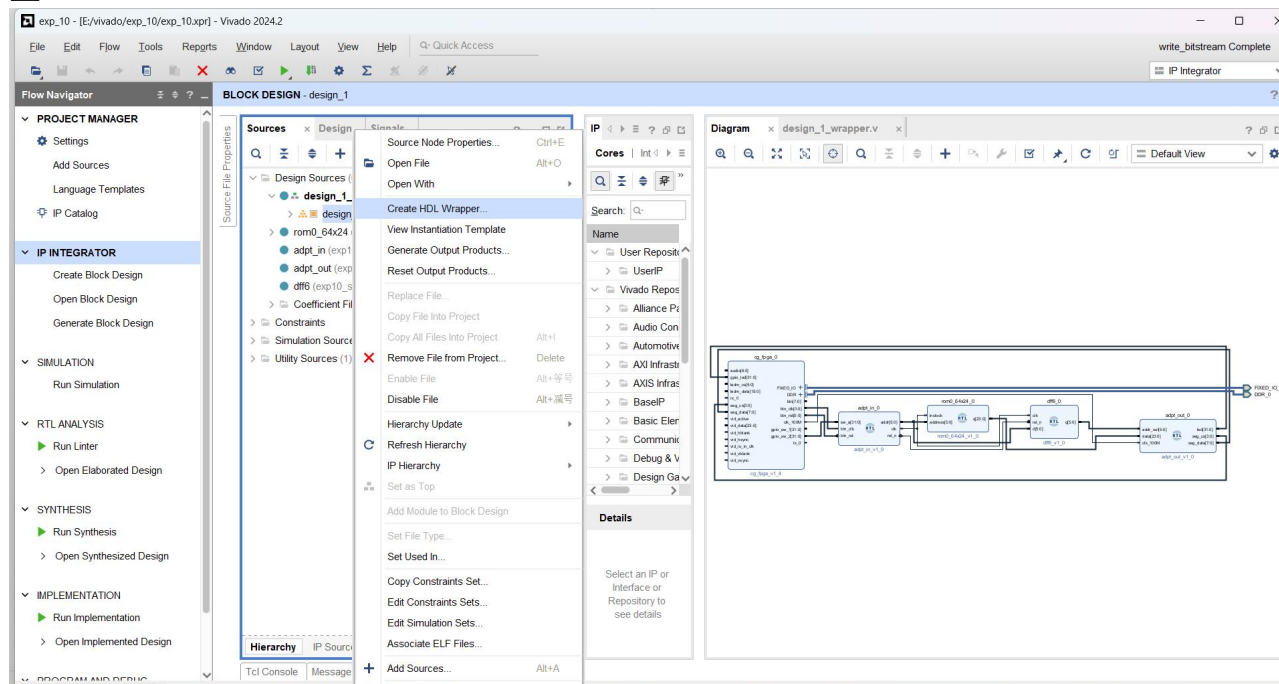
000FFC, 007FFE, 01FFFE, 03FFFE, 07FFFC, 0FFFC, 1FFE38, 3FF000, 3FC001, 7F8000, 7F0000, FF0000, FE0000, FE0000, FE0000, FC0000, FC0000, FE0000, FE0000, FE0000, FE0000, 7F0000, 7F8000, 3FC000, 3FE001, 1FF800, 0FFFF8, 0FFFF8, 07FFFC, 01FFFC, 00FFFC, 003FFC, 0003E0;



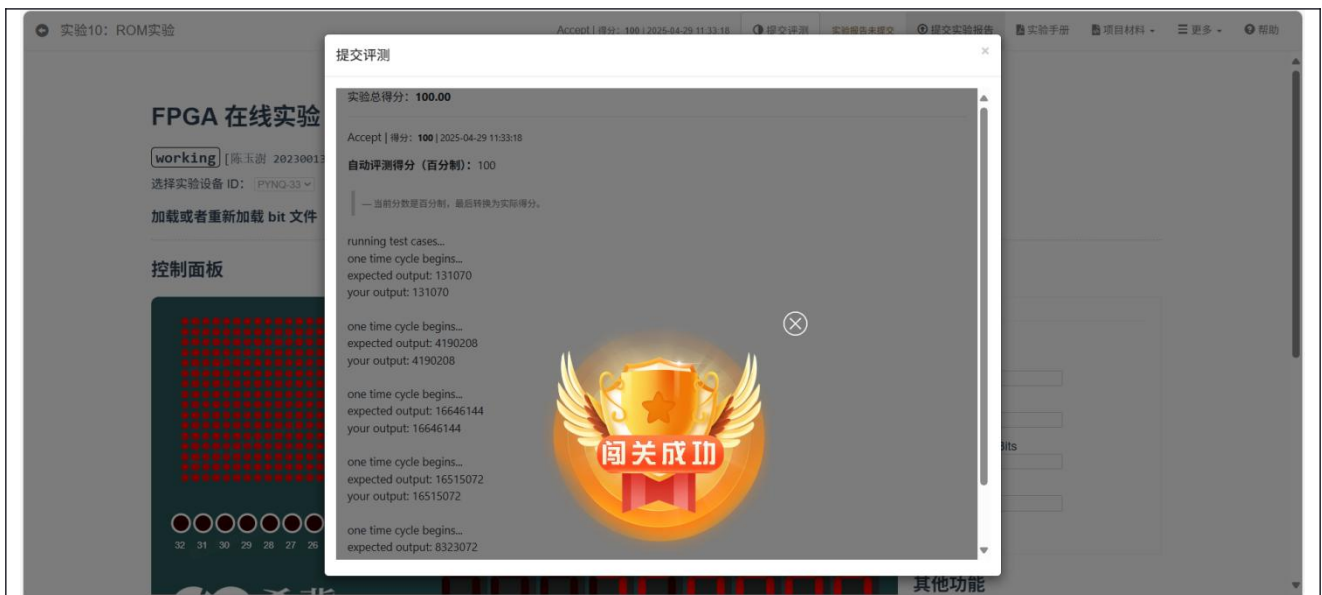
(10) 根据要求完成原理图的输入。每一次上升沿，将地址锁入，数码管 6-1 将显示 ROM 中输出的数据。LED6-1 显示输入的 6 位地址值。



(11) 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。

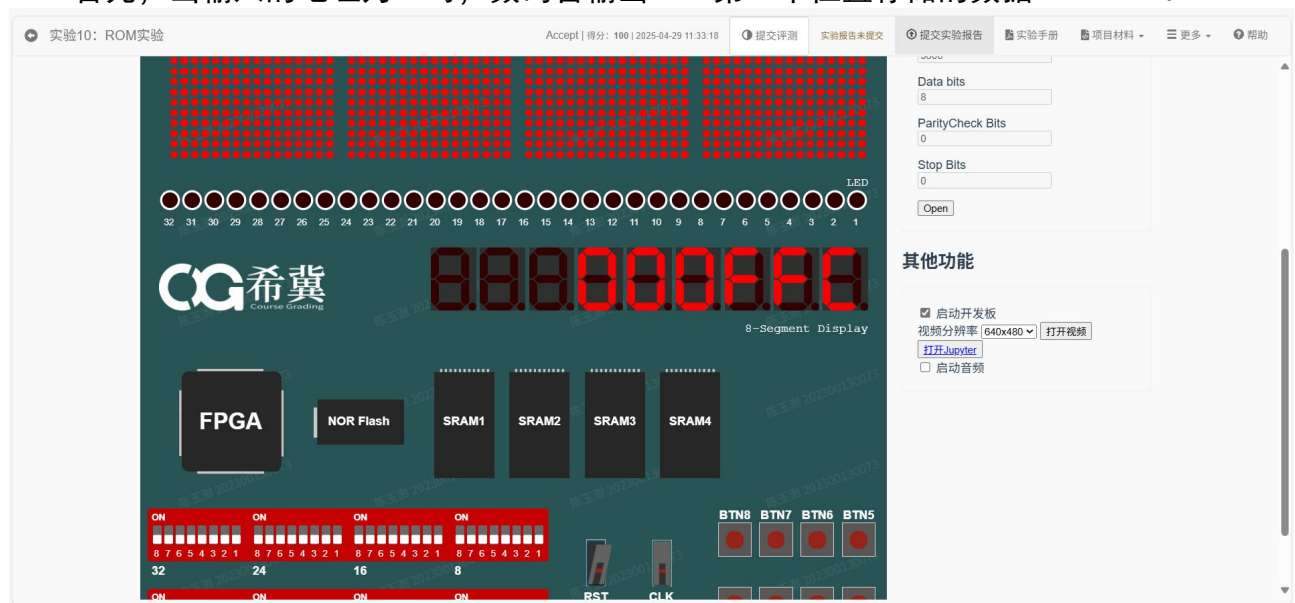


(12) 通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 *.bit 文件，后点击 send，即可将本地 bit 文件烧写至希冀远程 FPGA。

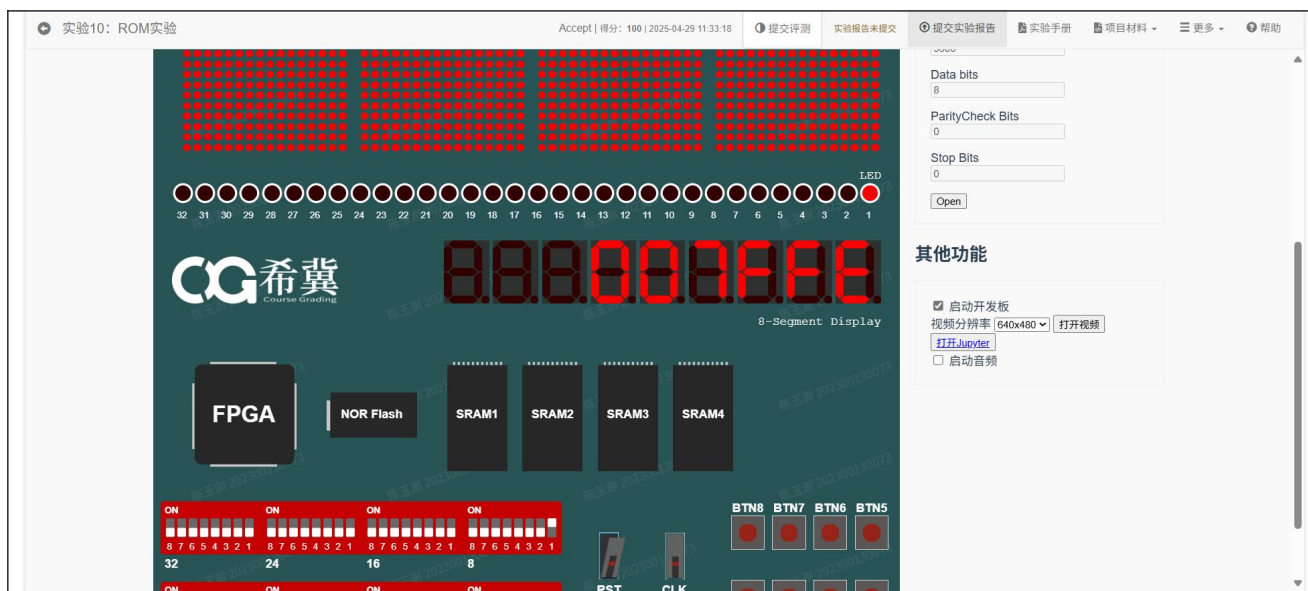


clk 信号的管脚绑定在 btn_clk 上, rst_n 信号的管脚绑定在 btn_rst 上, addr 信号的管脚绑定在上排拨码开关 6-1 上, addr_out 信号的管脚绑定在 LED6-1 上, datat 信号的管脚绑定在数码管 6-1 上。

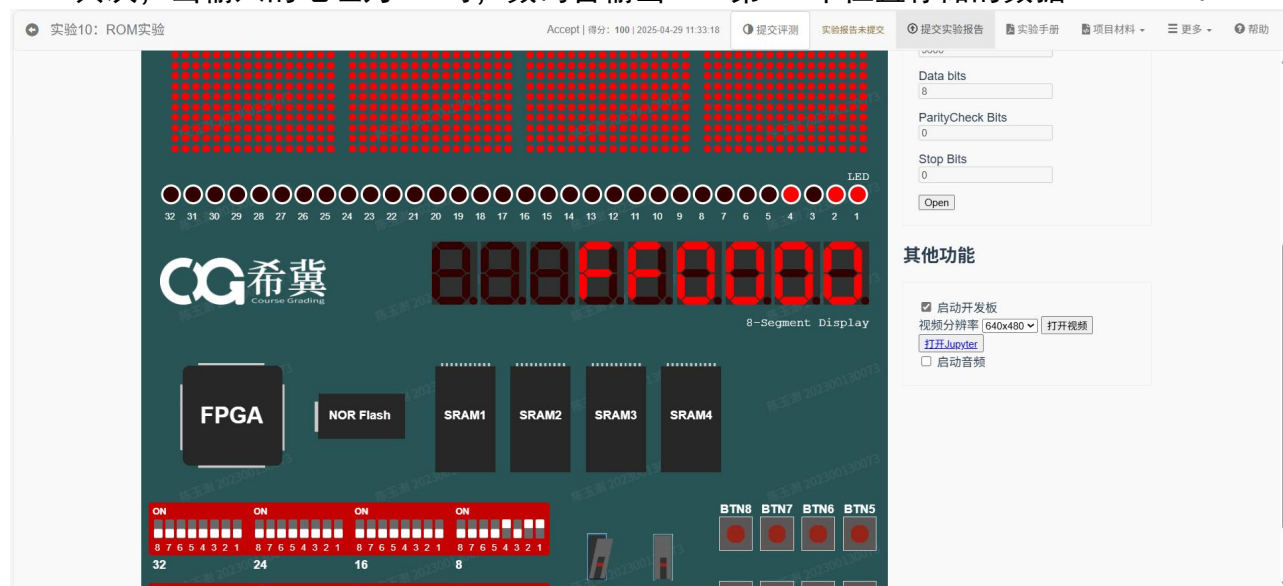
首先, 当输入的地址为 0 时, 数码管输出 ROM 第 0 个位置存储的数据: 000FFC。



其次, 当输入的地址为 1 时, 数码管输出 ROM 第 1 个位置存储的数据: 007FFE。



其次，当输入的地址为 11 时，数码管输出 ROM 第 11 个位置存储的数据：FF0000。



经过验证，发现输入地址得到的数据与当时存入 rom 的数据全部一一对应，符合实验预期。

结论分析与体会：

在实验过程中，成功掌握了 Vivado 中 ROM 的设置流程，熟悉了只读存储器 ROM 的工作特性和配置方法。借助文本编辑器编辑 .coe 文件，并将其加载于 ROM 中，同时也学会了在初始化存储器编辑窗口编辑 .coe 文件配置 ROM。通过定制单口 ROM，设置其接口类型、端口宽度、深度等参数，并导入包含特定数据的 .coe 文件完成初始化。

此次实验不仅加深了对 ROM 原理和工作机制的理解，还提升了在 Vivado 环境下进行 FPGA 开发的实践能力，包括项目创建、IP 核添加、原理图设计、综合与实现以及硬件下载验证等多方面技能，为后续更复杂的数字电路设计和 FPGA 应用开发奠定了坚实基础。

就向 .coe 文件加入数据失败问题处理的：


发现多输入了不该输入的 ‘;’ 和空格都有可能导致输入失败，数据与数据之间应该用 ‘,’ 分隔。

注意：评测时使用的是固定的数据，请向 .coe 文件中加入如下数据：

```
000FFC,007FFE,01FFFE,03FFFE,07FFFC,0FFFFC,1FFE38,3FF000,3FC001,7F8000,7F0000,FF0000,FE0000,FE0000,FE0000,FC0000,FC0000,FE0000,FE0000,FE0000,7F0000,7F8000,3FC000,3FE001,1FF800,0FFFF8,0FFFF8,07FFFC,01FFFC,00FFFC
```

COE File Editor - rom.coe

IO 信号	Key	Value
clk	memory_initialization_radix	16;
rst_n	memory_initialization_vector	000FFC,007FFE,01FFFE,03FFFE,07FFFC
addr		
addr_out		
data	数码管 6-1	

 Validate Save Save As... Close

4、实验任务

1. 查阅该页面的内容，用图形编辑软件