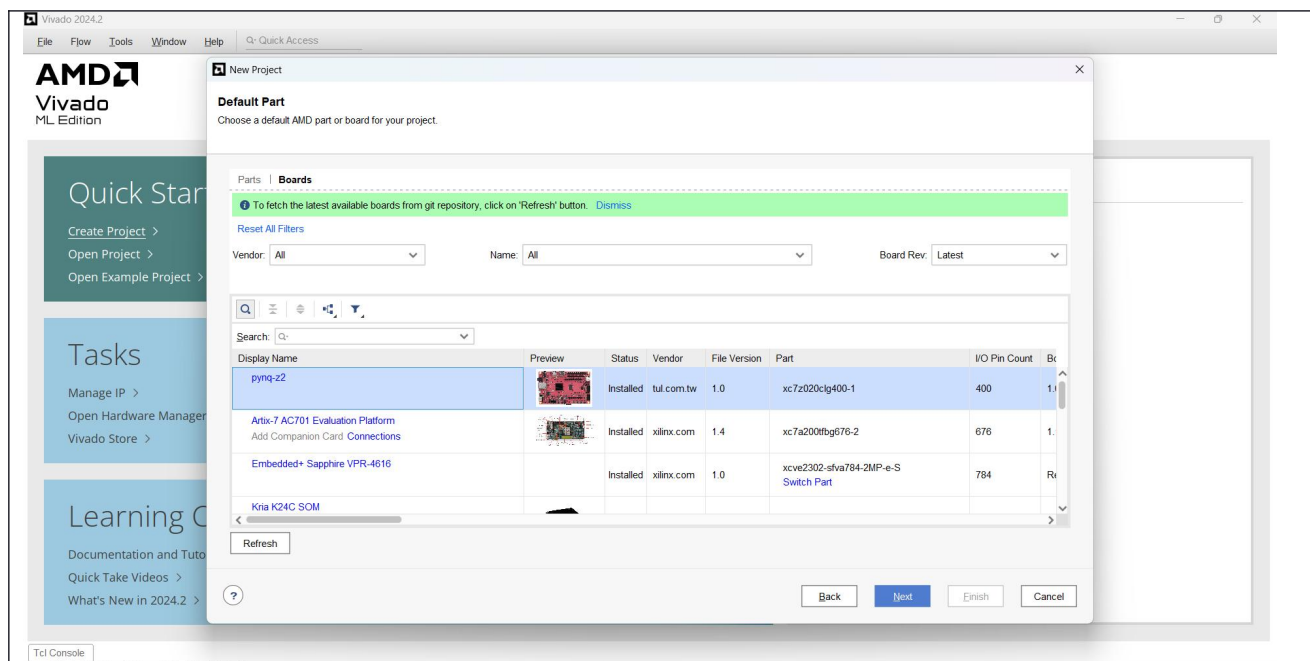


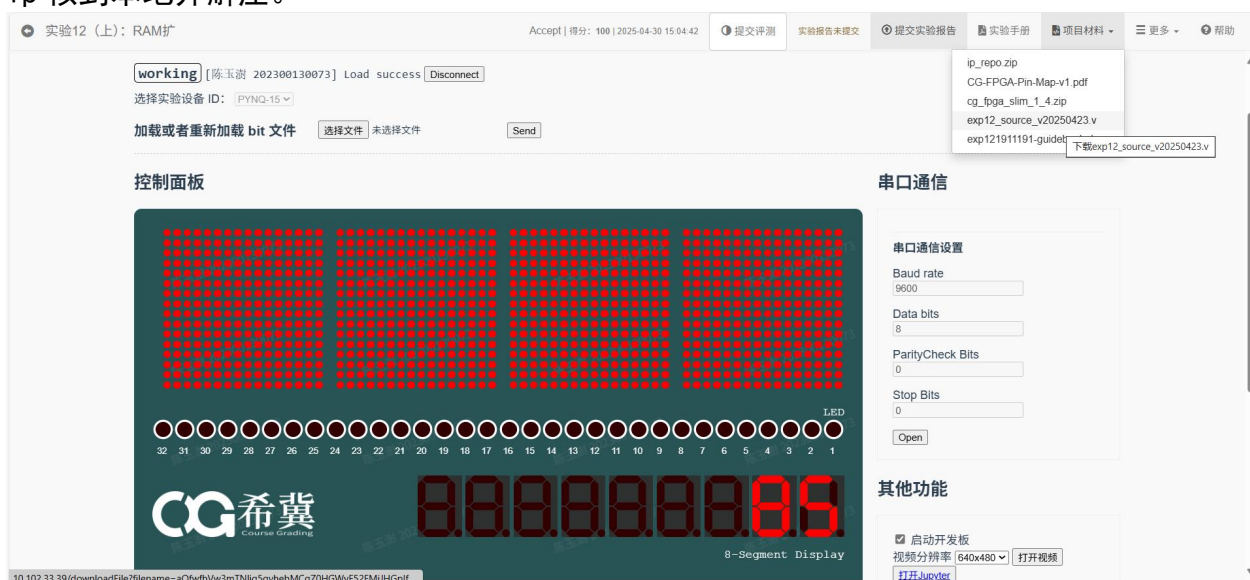
计算机组成与设计

课程实验报告

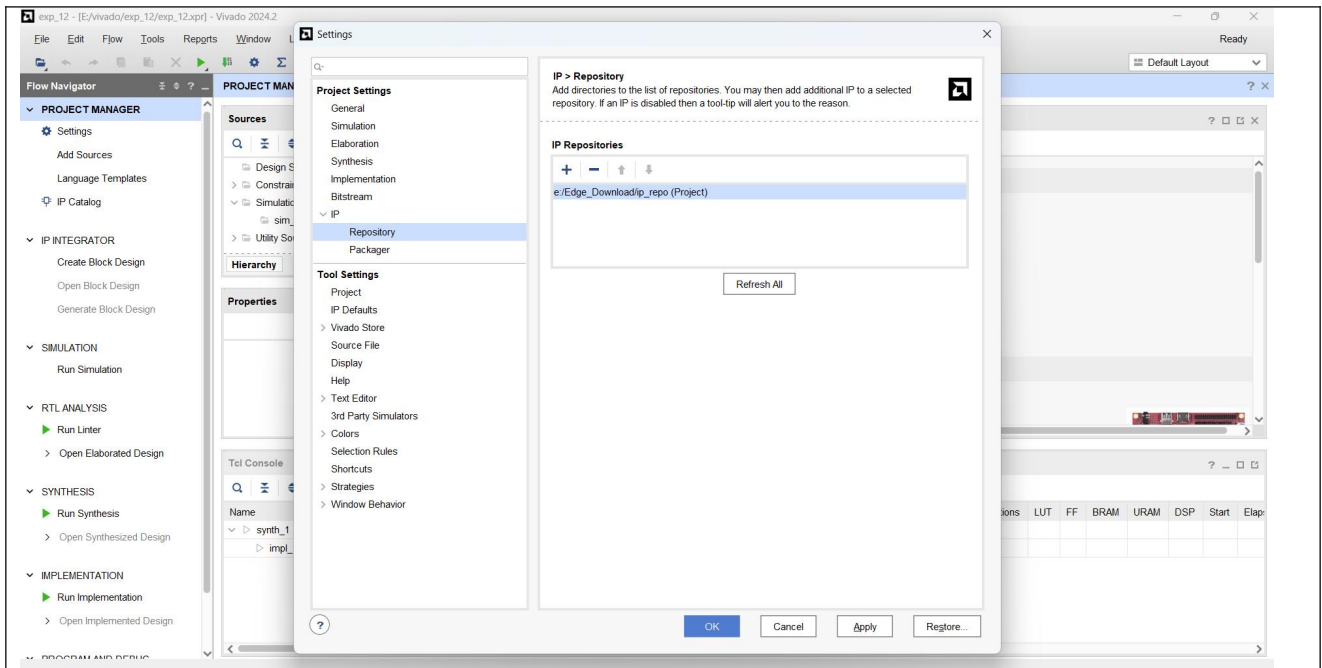
学号：202300130073	姓名：陈玉澍	班级：计科 23 级 1 班
实验题目： 实验 12 RAM 扩展实验		
实验学时：2	实验日期：2025. 4. 30	
实验目的： 1. 了解半导体静态随机读写存储器 RAM 的工作原理及其使用方法。 2. 掌握半导体存储器的字、位扩展技术。		
实验软件和硬件环境： 软件环境： Vivado 软件、FPGA 实验平台 硬件环境： 1. 实验室台式机 2. FPGA 服务器，PYNQ-Z2 开发板		
实验原理和方法： 本次实验设计的 RAM 基于 Xilinx Block Memory Generator IP 核，实现了一个单口同步的 RAM： 1、基本结构与配置 （1）存储容量：1K×4（1024 个存储单元，每个单元 4 位数据），地址线宽度为 10 位（2^10=1024），数据线宽度为 4 位。 （2）接口信号： 1）时钟信号（clk）：同步控制读写操作，数据在时钟边沿有效。 2）地址线（address [9:0]）：选择存储单元，范围 0-1023。 3）数据线（data [3:0]）：写入时输入数据，读取时输出数据。 4）写使能信号（wren）：高电平时允许写入，低电平时禁止写入（此时执行读操作）。 2. 读写操作机制 （1）写操作：当写使能信号（wren=1）有效时，在时钟（clk）上升沿，数据（data [3:0]）被写入地址线（address [9:0]）指定的存储单元。写入后，该单元数据更新为输入值。 （2）读操作：当写使能信号（wren=0）无效时，地址线（address [9:0]）指定单元的数据在时钟上升沿输出至数据线（q [3:0]），读取操作不改变存储单元内容。		
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。		



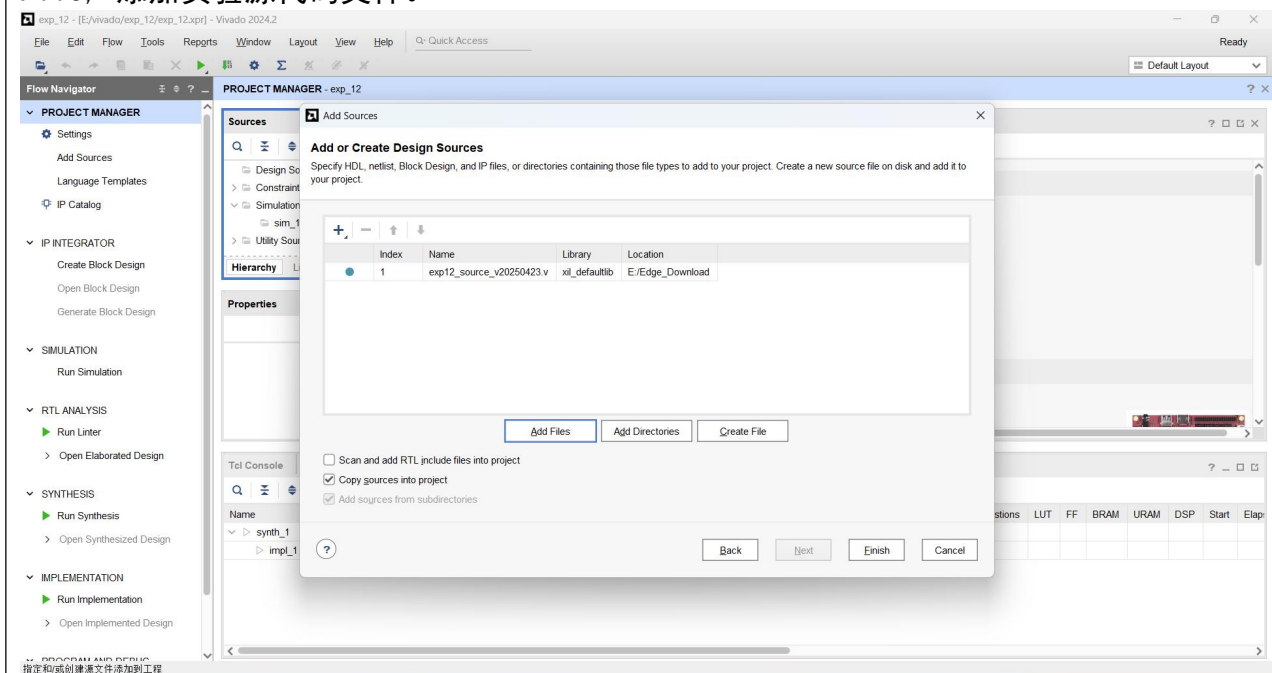
(2) 添加实验环境：进入 FPGA 在线实验环境，点击右上角项目材料下载实验源代码和希冀 ip 核到本地并解压。



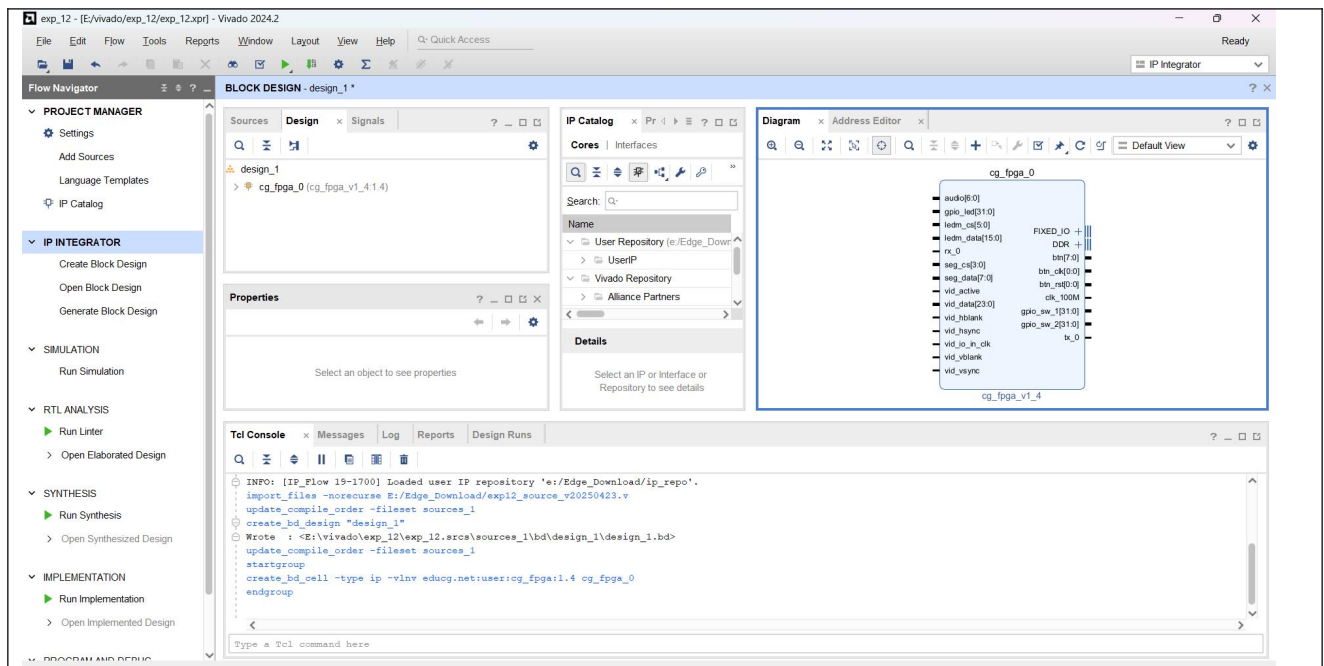
(3) 在 Vivado 项目中，点击 Settings→IP→Repository，将上一步解压后的 ip_repo 文件夹的位置添加进 IP 搜索目录。



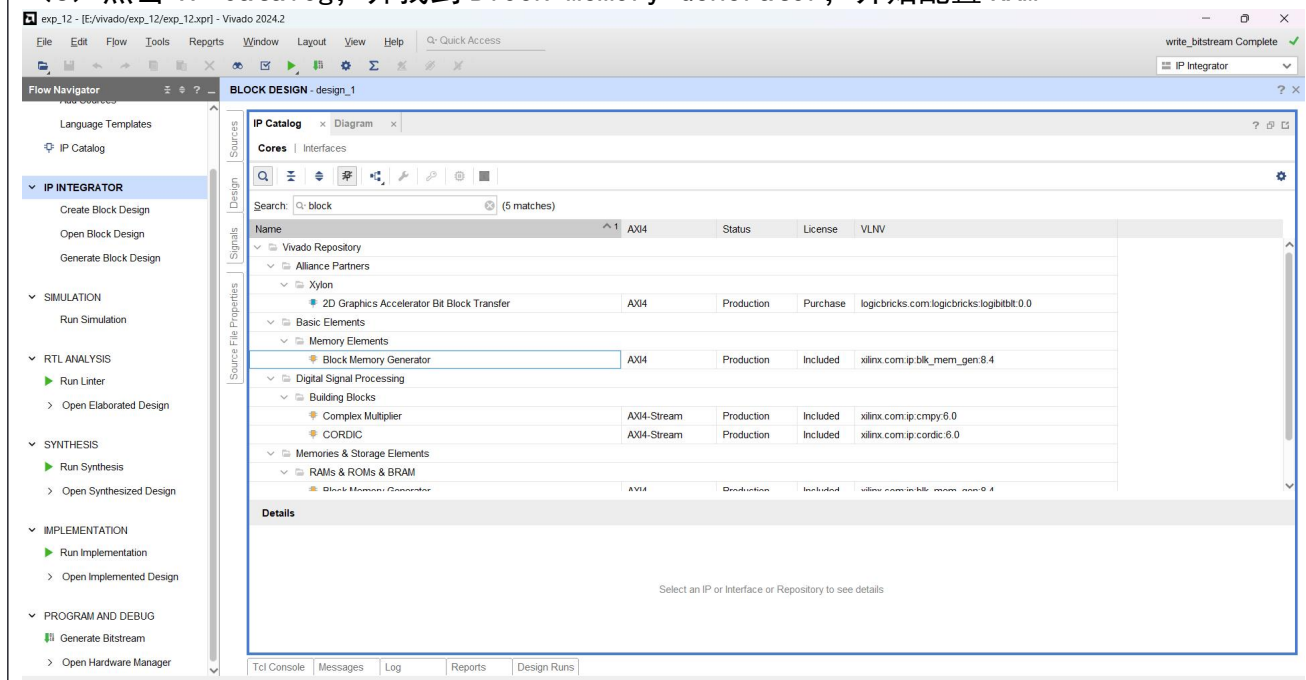
(4) 点击 Sources 窗口中的+, 选择 Add or create design sources → Next → Add File, 添加实验源代码文件。



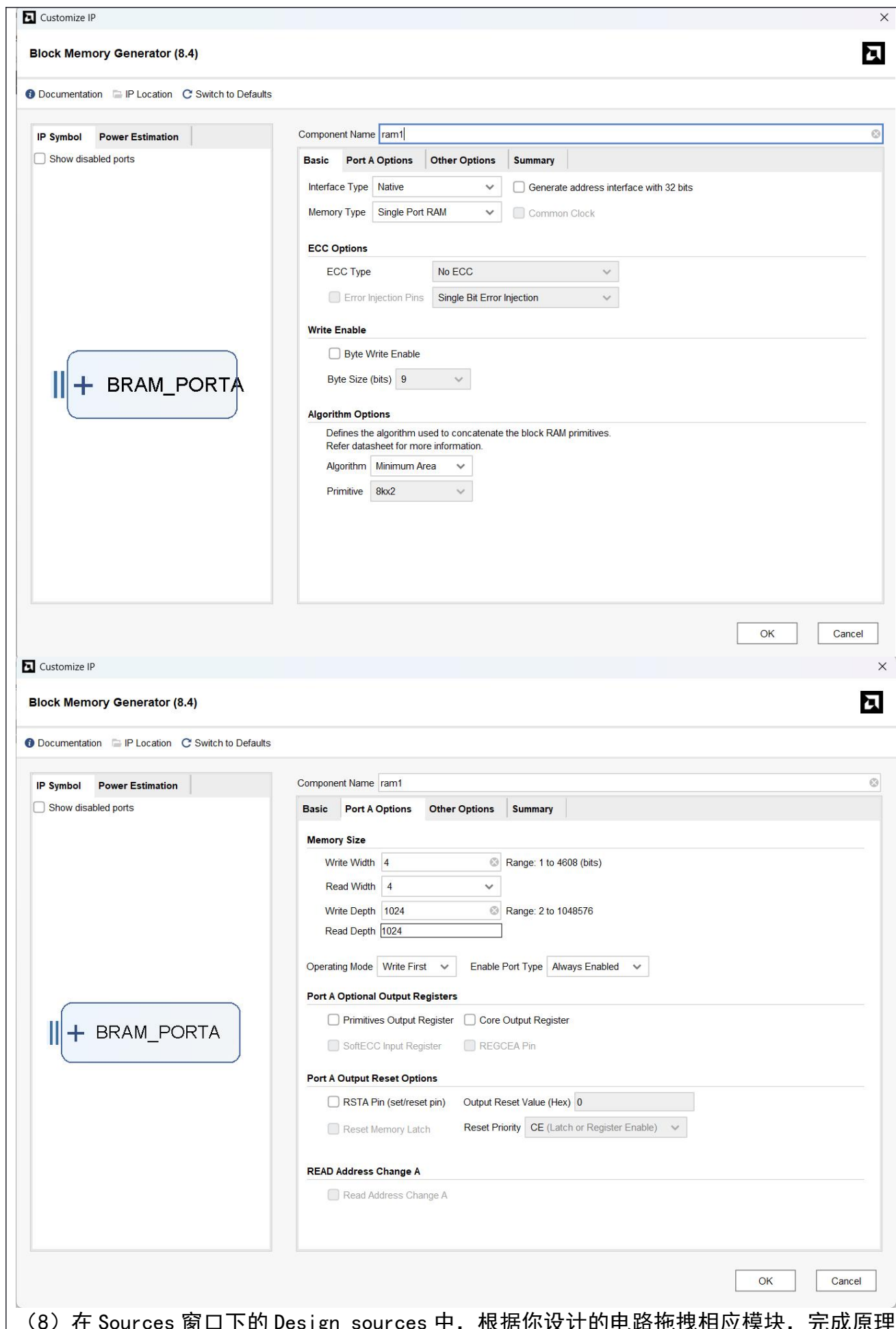
(5) 点击 Create Block Design 创建一个新的顶层设计，随后点击添加 IP 核按钮，添加 cg_fpga IP.



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

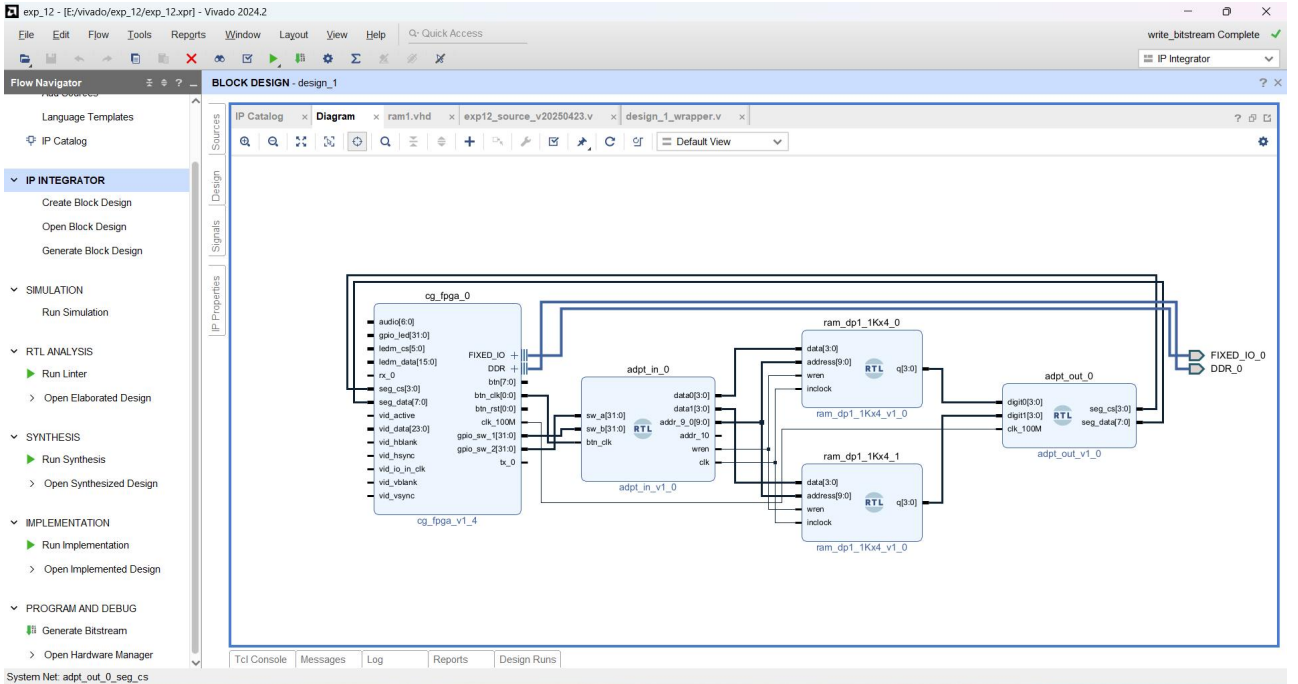


(7) 按照如下方法定制单口 RAM (1K*4), 并命名为 ram1:

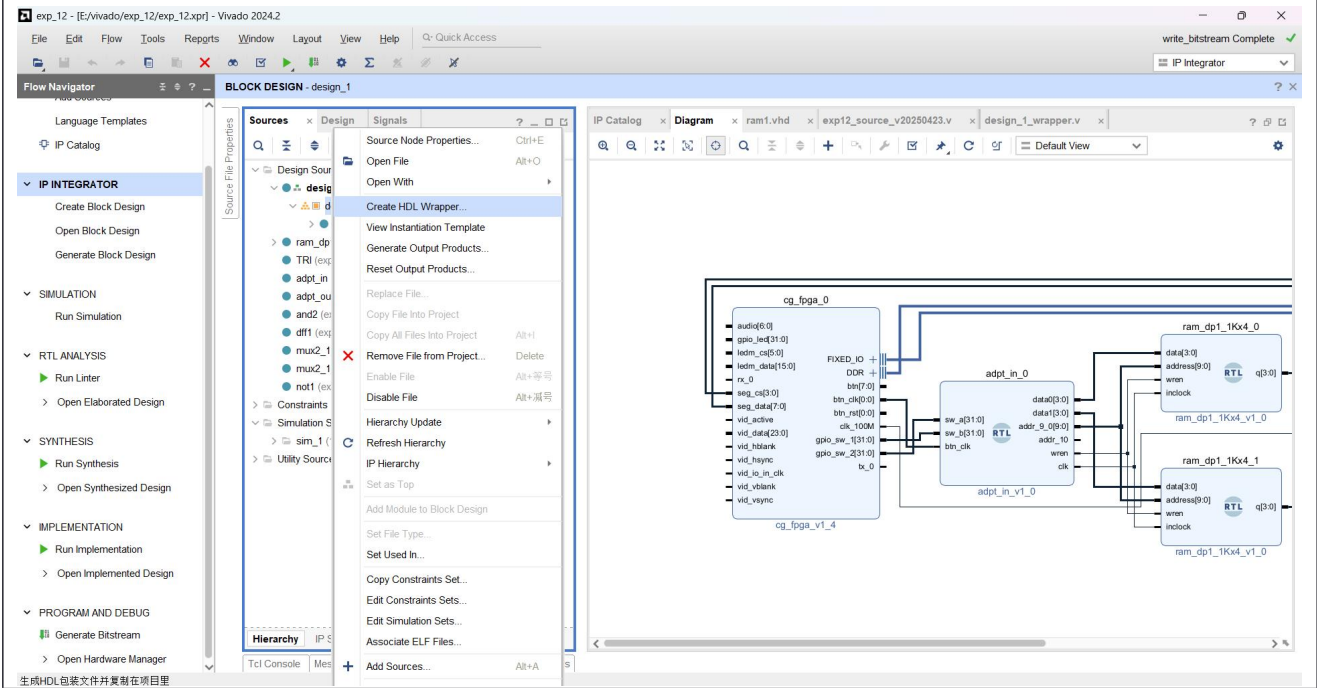


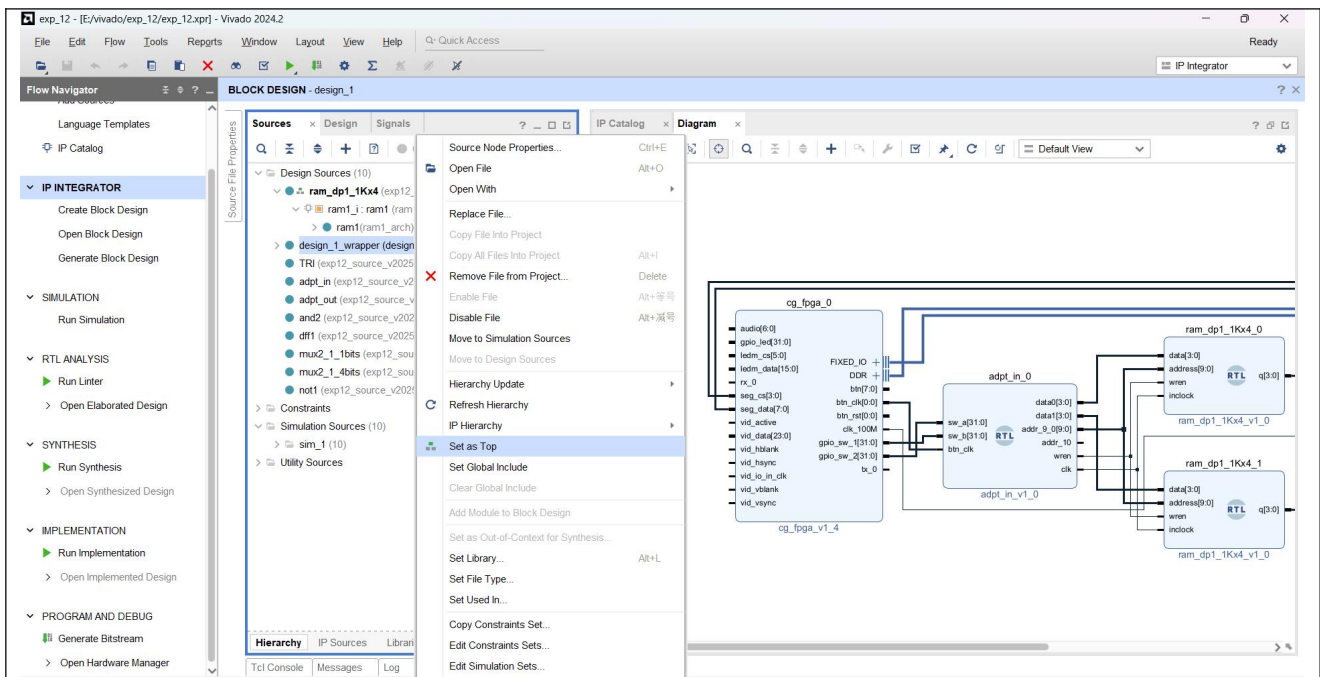
(8) 在 Sources 窗口下的 Design sources 中，根据你设计的电路拖拽相应模块，完成原理

图的输入。

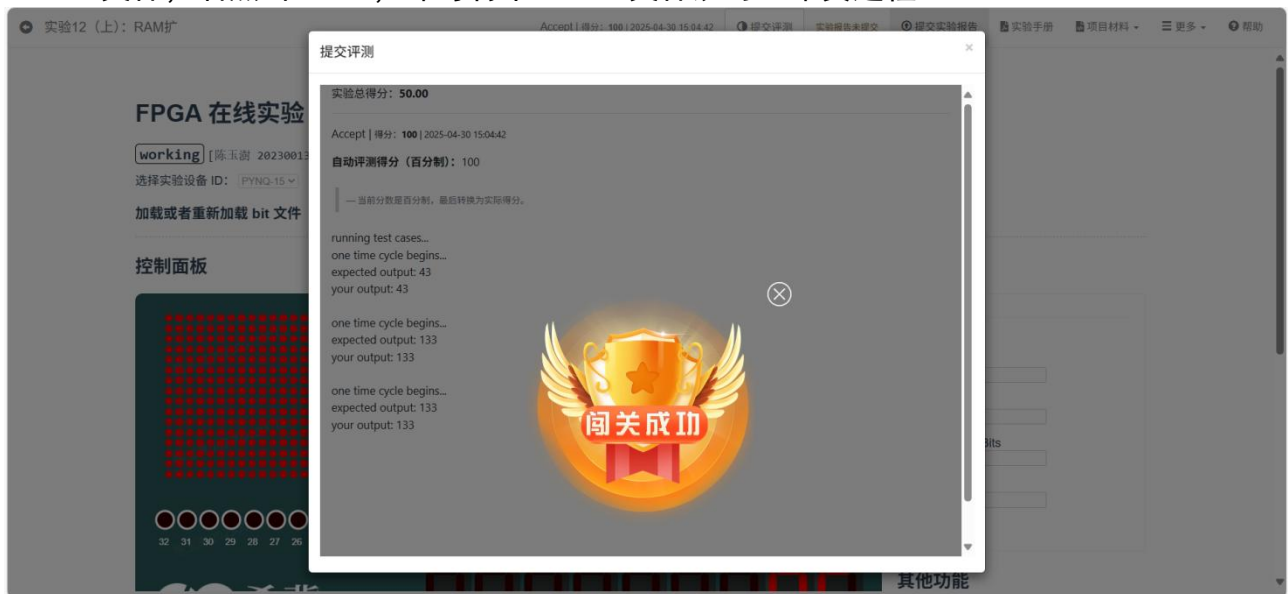


(9) 右击 Sources 下顶层设计图标→Create HDL Wrapper, 待 Wrapper 正确生成后, 点击左下方 Generate Bitstream, 开始综合并生成 bit 文件。注意: 综合前 wrapper 模块应被设置为顶层 (加粗表示), 若自动设置错误, 需右击 wrapper 图标点击 Set as Top 手动设置。





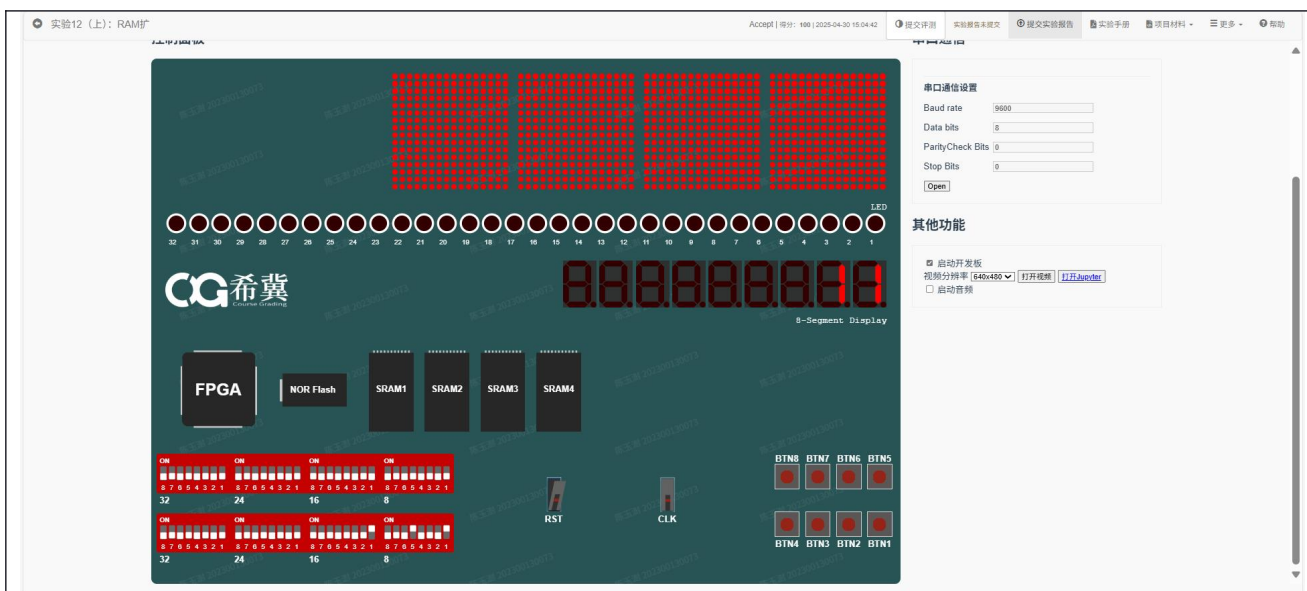
(10) 通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 *.bit 文件，后点击 send，即可将本地 bit 文件烧写至希冀远程 FPGA。



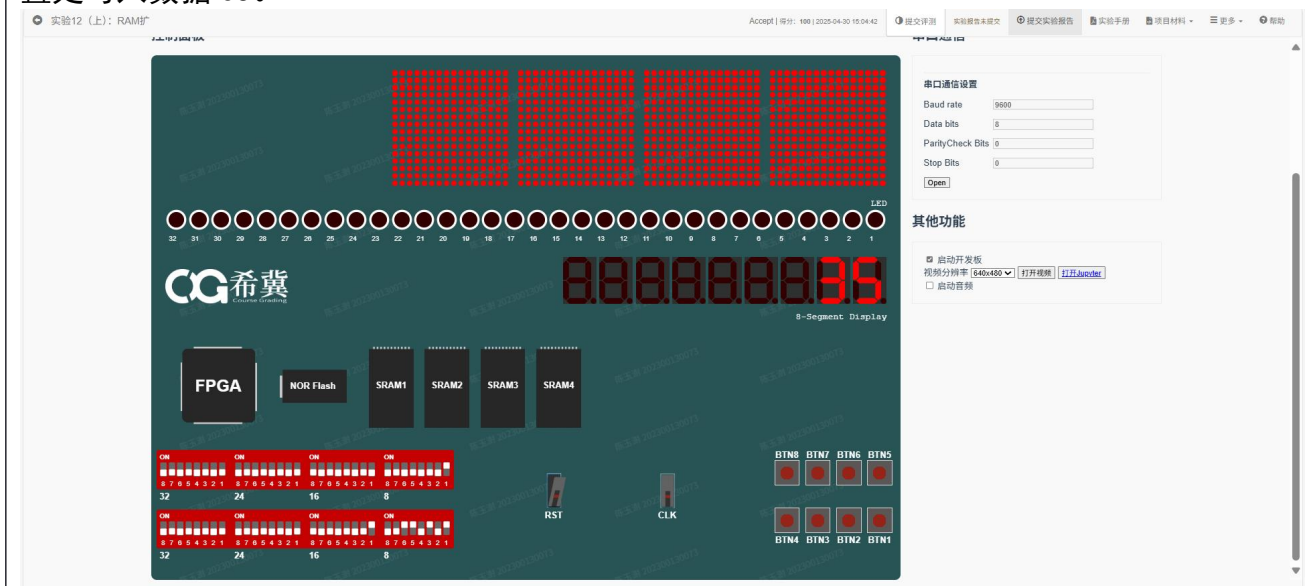
clk 信号的管脚绑定在 btn_clk 上，addr_9_0 信号的管脚绑定在上排拨码开关 10-1 上，addr_10 信号的管脚绑定在上排拨码开关 11 上，data1(4bits)，data0(4bits) 信号的管脚绑定在下排拨码开关 8-1 上，wren 信号的管脚绑定在下排拨码开关 9 上，digit1，digit0 信号的管脚绑定在数码管 2-1 上。

首先，先向半导体静态随机读写存储器 RAM 写入数据。置下派拨码开关 9 为 1（即 wren=1），此时状态为写数据。

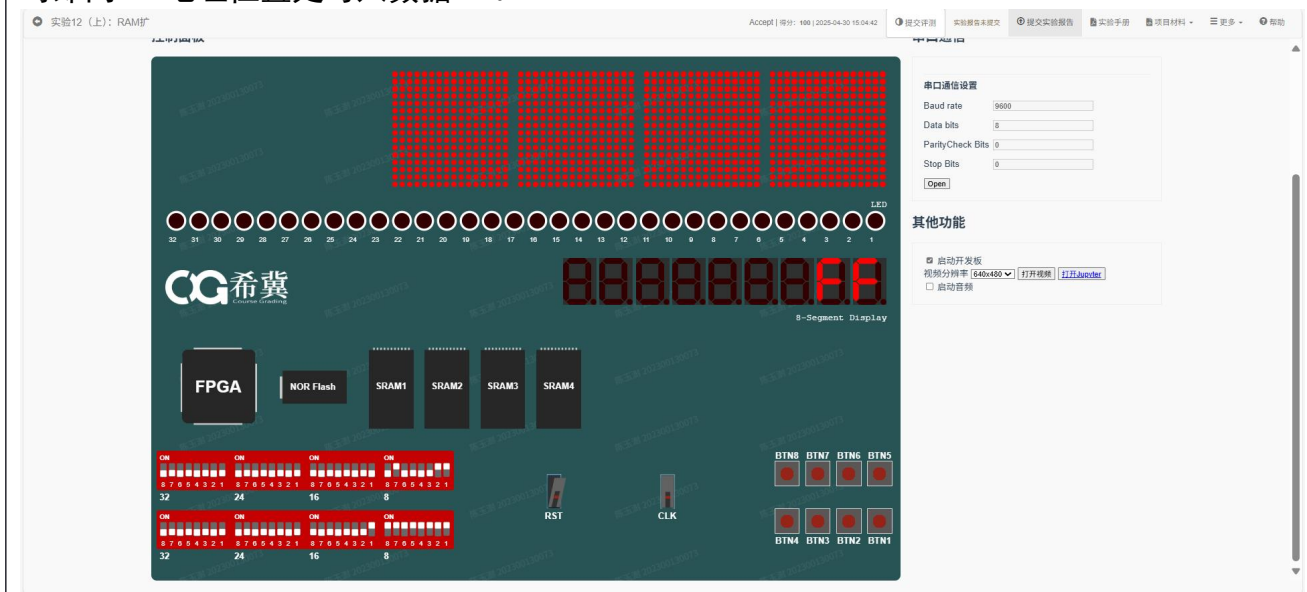
置地址 addr10-1 为全 0，data1(4bits)=1，data0(4bits)=1，此时即向 0 地址位置处写入数据 11。



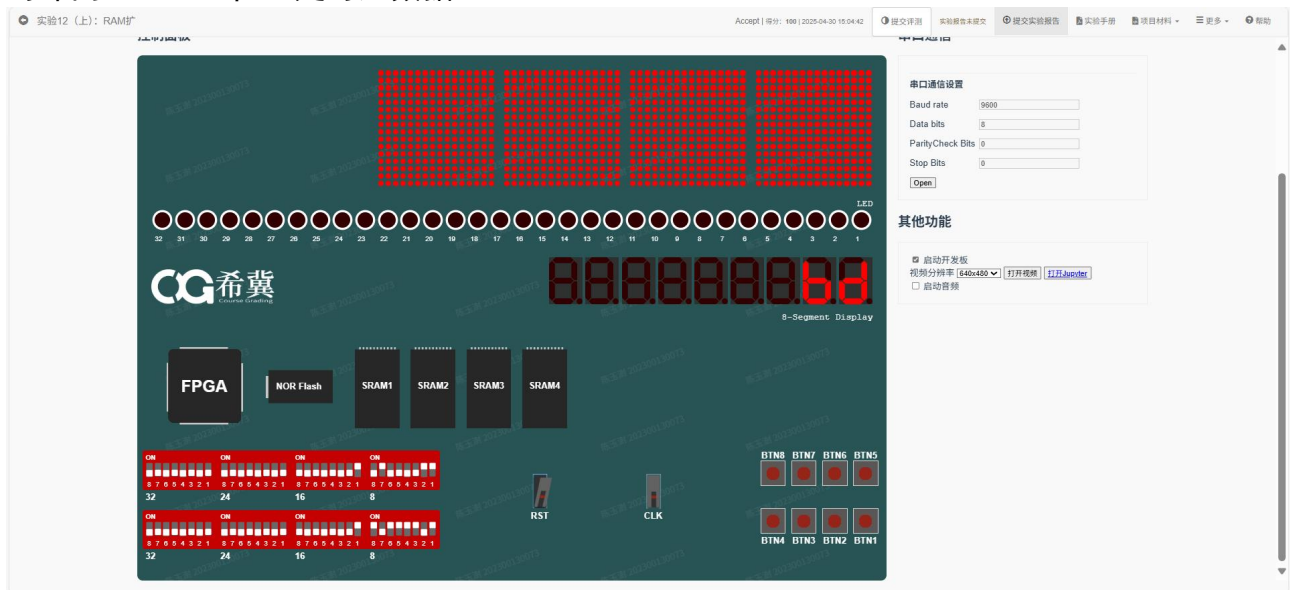
置地址 addr10-1 为 00...01, data1(4bits)=3, data0(4bits)=5, 此时即向 1 地址位置处写入数据 35。



置地址 addr10-1 为 001000011, data1(4bits)=F (15), data0(4bits)=F (15), 此时即向 67 地址位置处写入数据 FF。

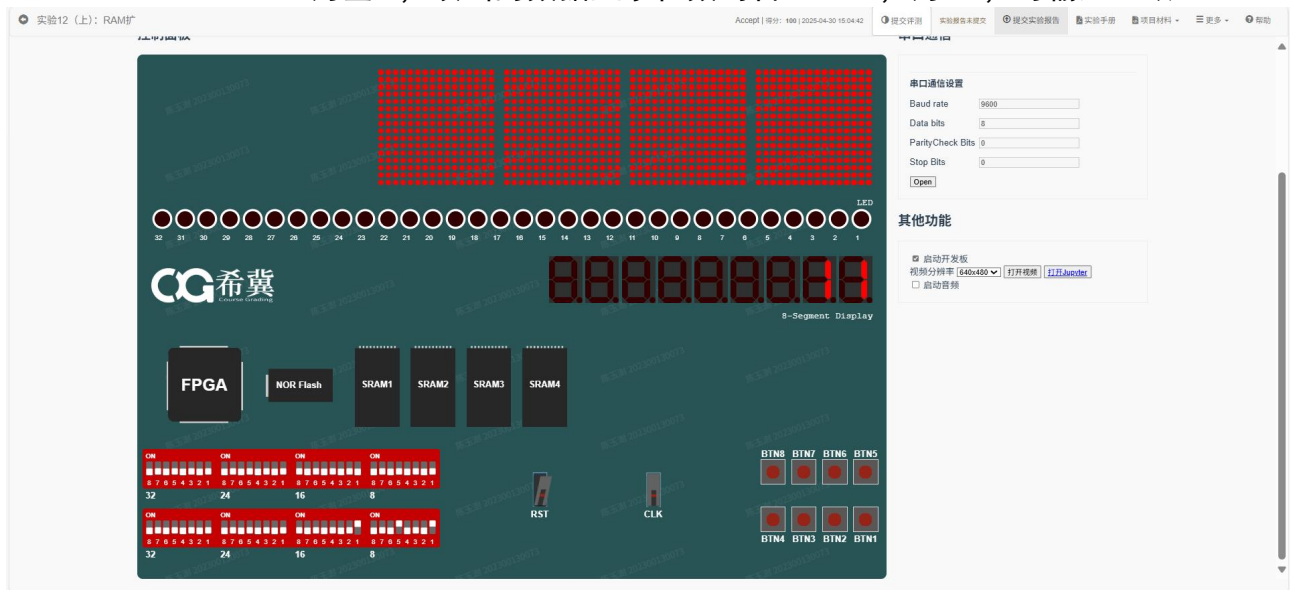


置地址 addr_{10-1} 为 101000011, $\text{data}_1(4\text{bits})=\text{b}(11)$, $\text{data}_0(4\text{bits})=\text{d}(13)$, 此时即向 579 地址位置处写入数据 bd。

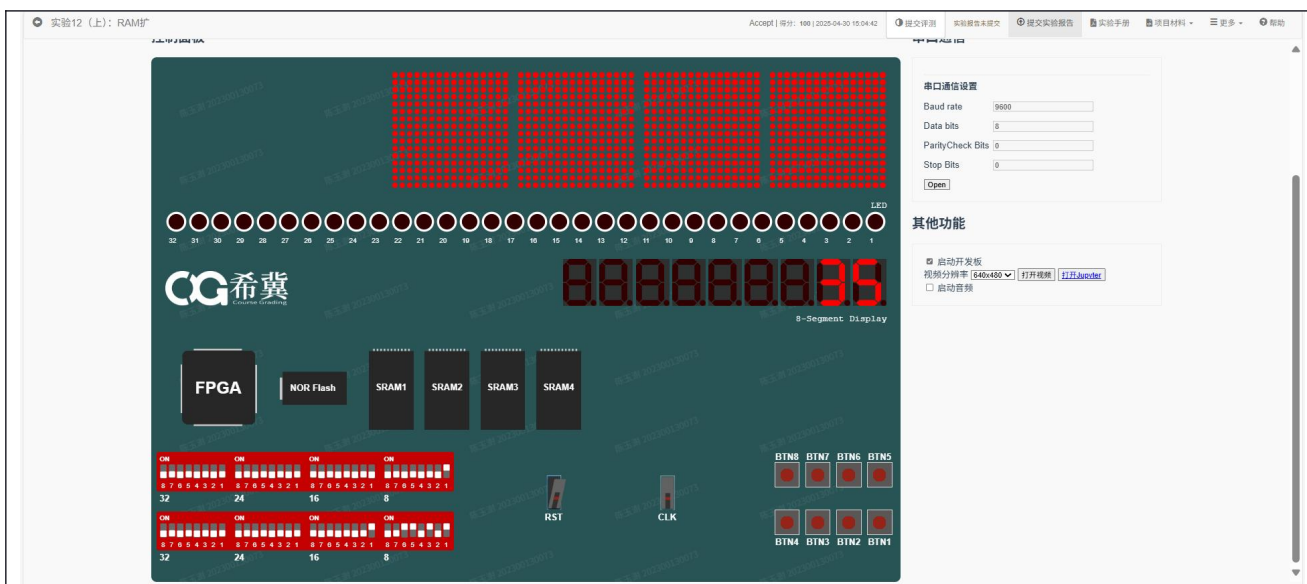


接着, 从半导体静态随机读写存储器 RAM 读出数据。置下派拨码开关 9 为 0 (即 $\text{wren}=0$), 此时状态为读数据。

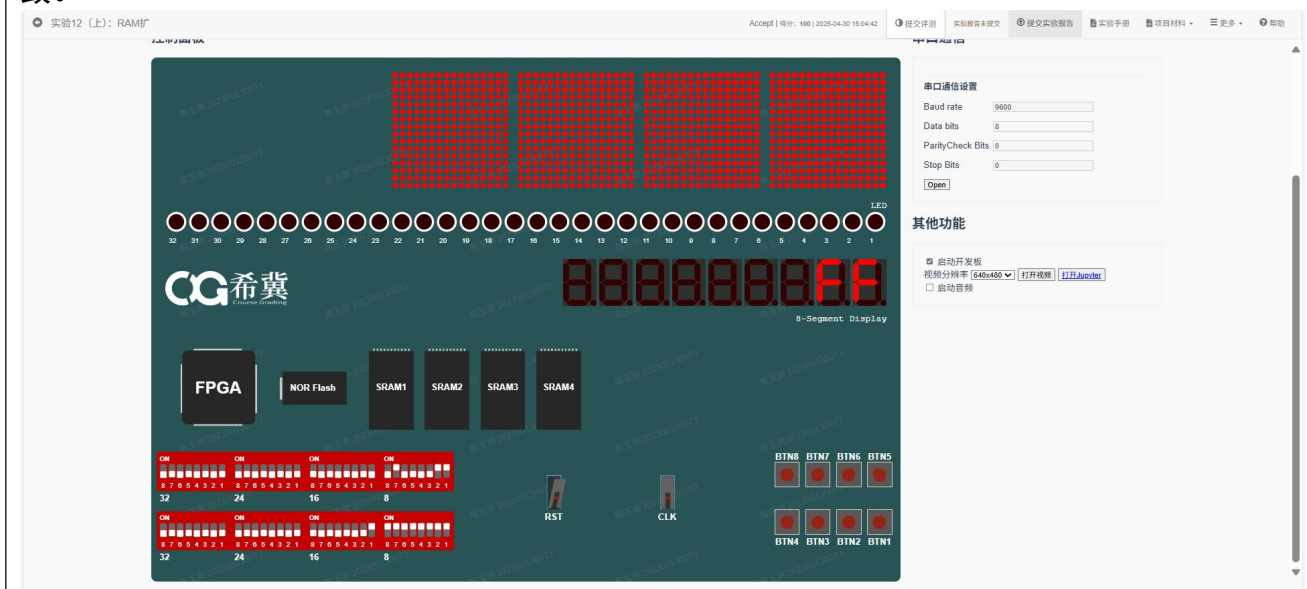
置地址 addr_{10-1} 为全 0, 读出的数据显示在数码管 2-1 上, 为 11, 与输入一致。



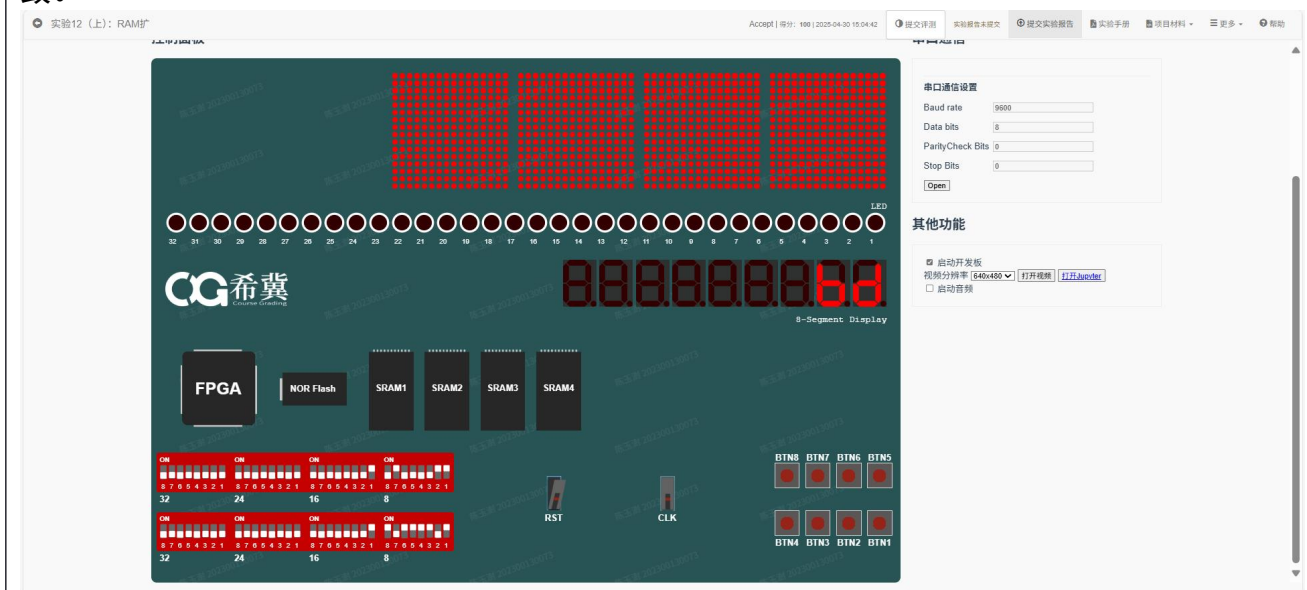
置地址 addr_{10-1} 为 00...01, 读出的数据显示在数码管 2-1 上, 为 35, 与输入一致。



置地址 addr10-1 为 001000011，读出的数据显示在数码管 2-1 上，为 FF，与输入一致。



置地址 addr10-1 为 101000011，读出的数据显示在数码管 2-1 上，为 bd，与输入一致。



经过实验检验发现，设计的半导体静态随机读写存储器 RAM 读写结果符合实验预期。

结论分析与体会：

本次实验通过 Vivado 软件成功设计并验证了 $1K \times 4$ 的半导体静态随机读写存储器 RAM，实现了对 RAM 工作原理的理解及字位扩展技术的应用。经硬件验证，RAM 读写功能正常：写入数据时，通过地址线、数据线和写使能信号 `wren` 控制，可将指定数据存入对应地址单元；读取数据时，禁用写使能，通过地址线选中单元，数据正确输出至数码管显示。实验结果表明，设计的 RAM 满足预期功能，有效掌握了 FPGA 中 RAM 的配置与调试方法，加深了对数字电路存储模块工作机制的理解。

就设计 RAM 时，发现配置 `ram` 后在 Design sources 并没有显示元件的问题处理的：

将元件名称改为 `ram1` 后，发现可以正常显示。这是因为 Vivado 工程中可能已存在同名元件，导致新配置的 RAM 元件无法正常显示。修改名称为 `ram1` 后，避免了这种名称冲突，从而能够正常显示。比如之前创建过未被正确删除或识别的 `ram` 元件，再次使用相同名称时就会出现这个问题。