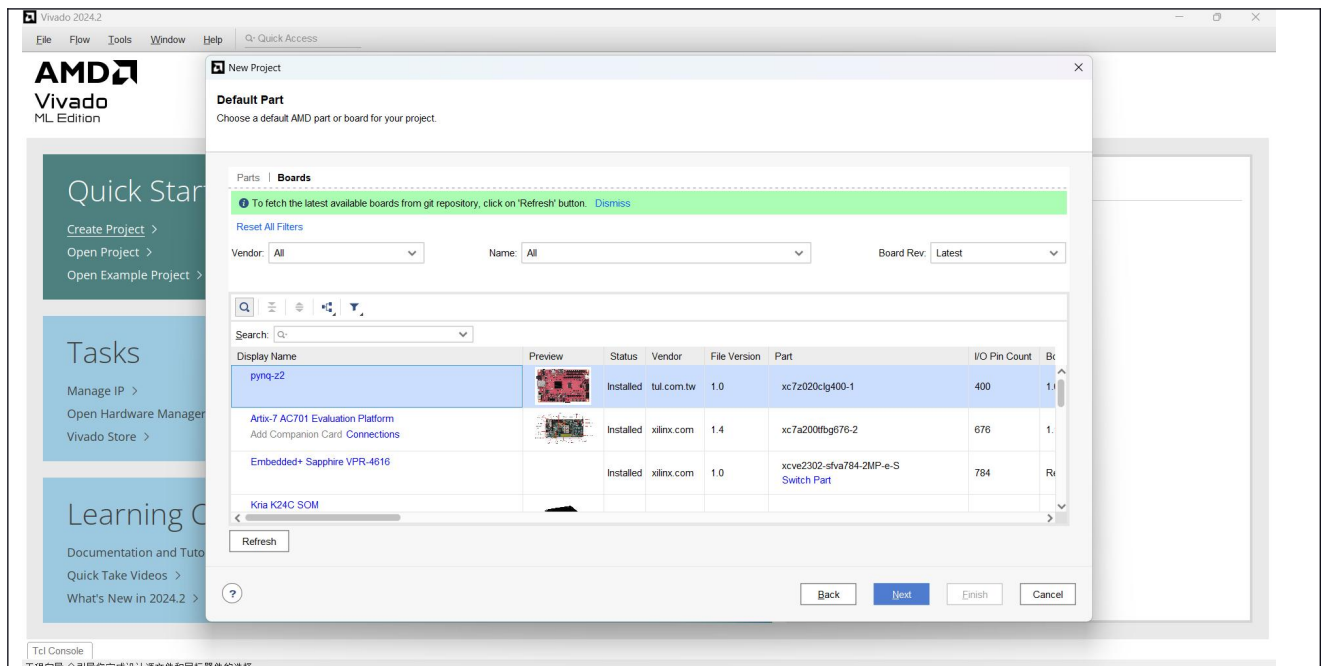
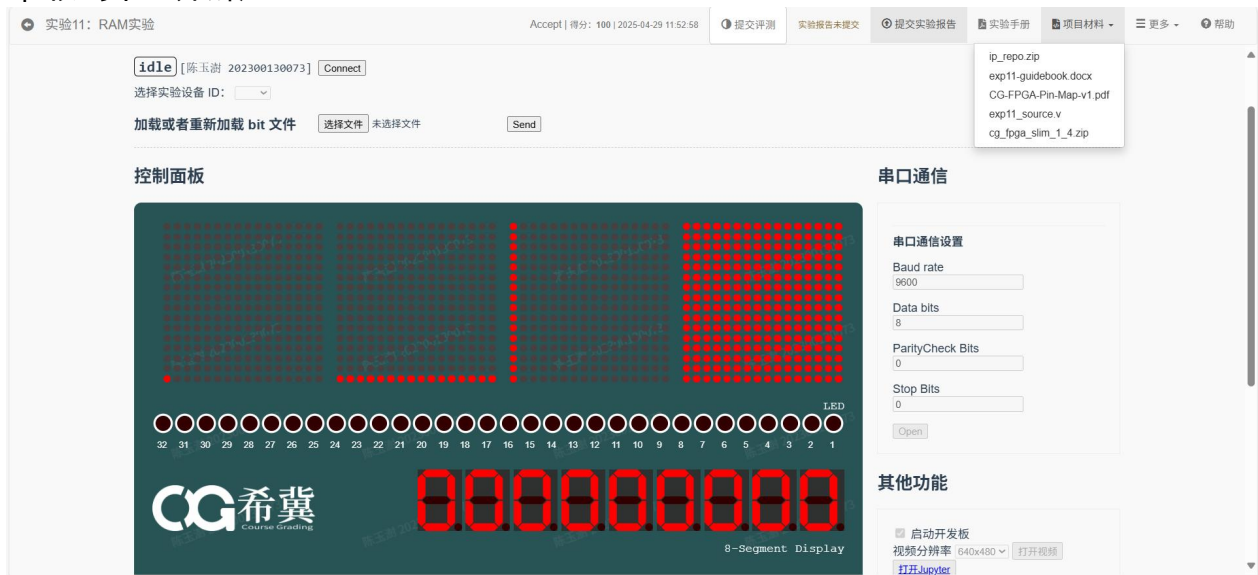


计算机组成与设计 课程实验报告

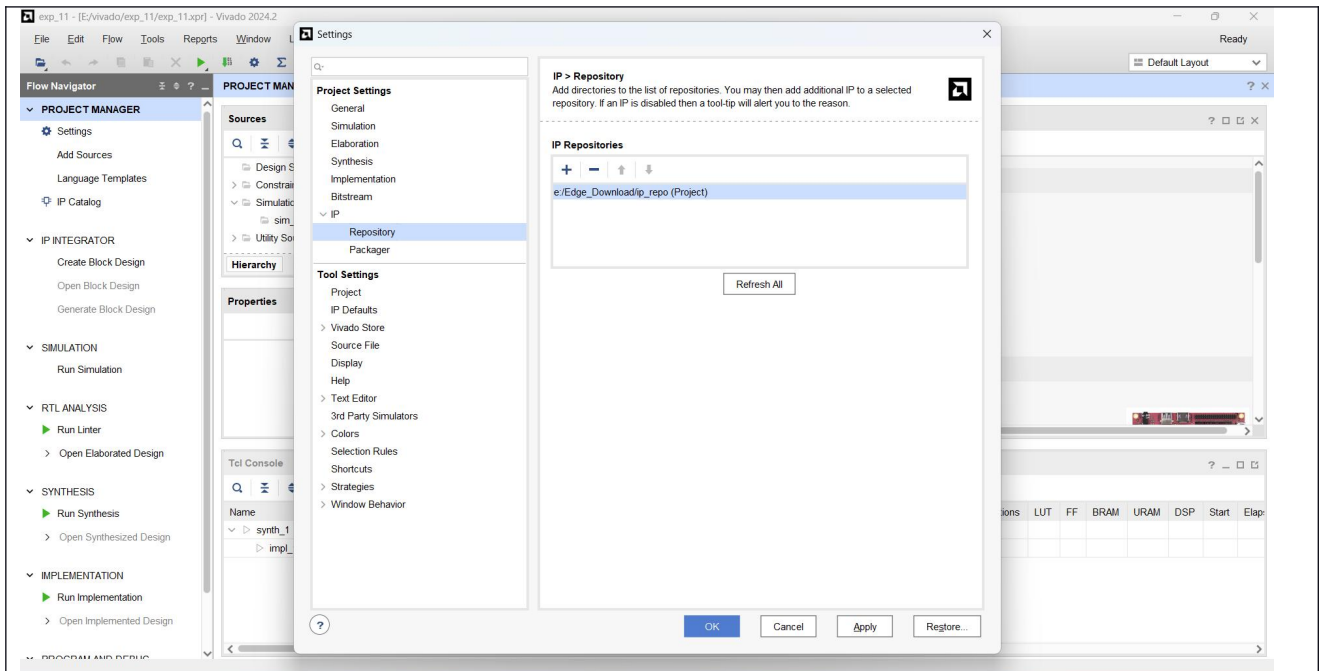
学号：202300130073	姓名：陈玉澍	班级：计科 23 级 1 班
实验题目： 实验 11 RAM 实验		
实验学时：2	实验日期：2025. 4. 29	
实验目的： 1、了解 FPGA 中 RAM 的功能； 2、掌握 block ram 的参数设置和使用方法； 3、掌握 BRAM 作为随机存储器 RAM 的仿真测试方法，工作特性和读写方法。		
实验软件和硬件环境： 软件环境： Vivado 软件、FPGA 实验平台 硬件环境： 1. 实验室台式机 2. FPGA 服务器，PYNQ-Z2 开发板		
实验原理和方法： 在 FPGA 中利用嵌入式阵列块 EAB 可以构成存储器，结构如下图。 1、数据从 ram_dp0 的左边 D[7..0] 输入，从右边 Q[7..0] 输出，R/W 为读/写控制信号端。 2、当输入数据和地址准备好以后，在 inclock 是地址锁存时钟，当信号上升沿到来时，地址被锁存，数据写入存储单元。 3、数据的读出控制是从 A[7..0] 输入存储单元地址，在 CLK 信号上升沿到来时，该单元数据从 Q[7..0] 输出。 4、R/W 是读/写控制端，低电平时进行读操作，高电平时进行写操作；CLK 是读/写时钟脉冲信号；DATA[7..0] 是 RAM_dq0 的 8 位数据输入端；A[7..0] 是 RAM 的读出和写入地址；Q[7..0] 是 RAM_dq0 的 8 位数据输出端。		
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。		



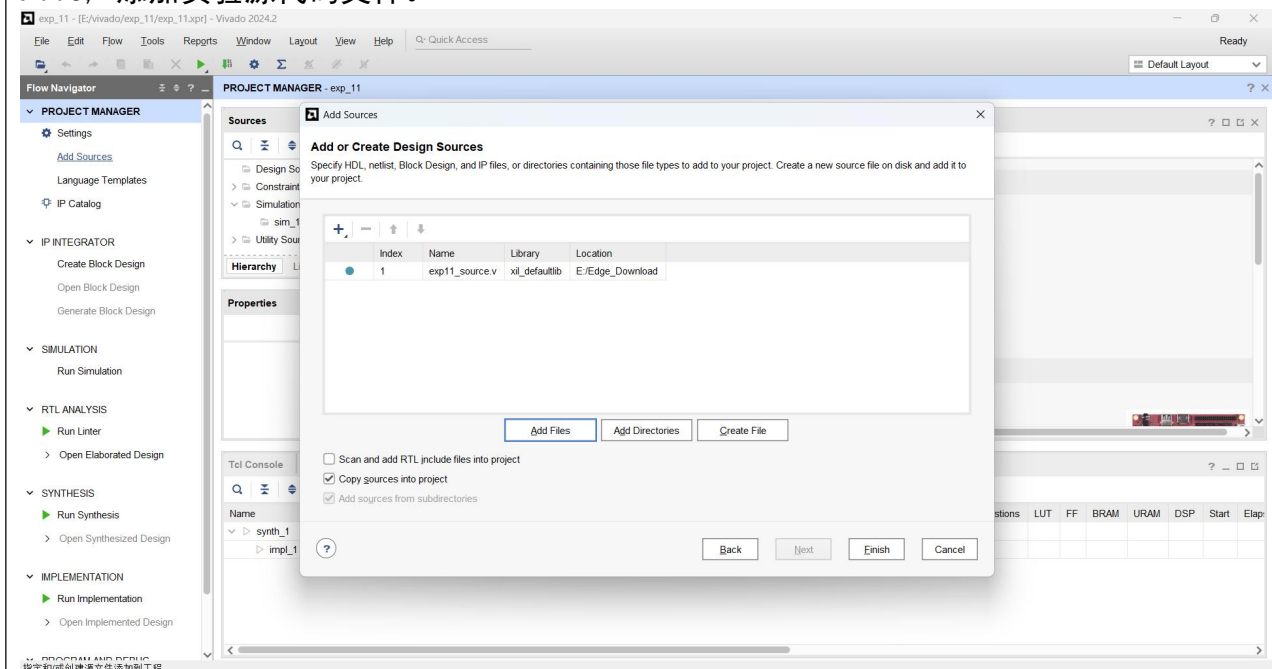
(2) 添加实验环境：进入 FPGA 在线实验环境，点击右上角项目材料下载实验源代码和希冀 ip 核到本地并解压。



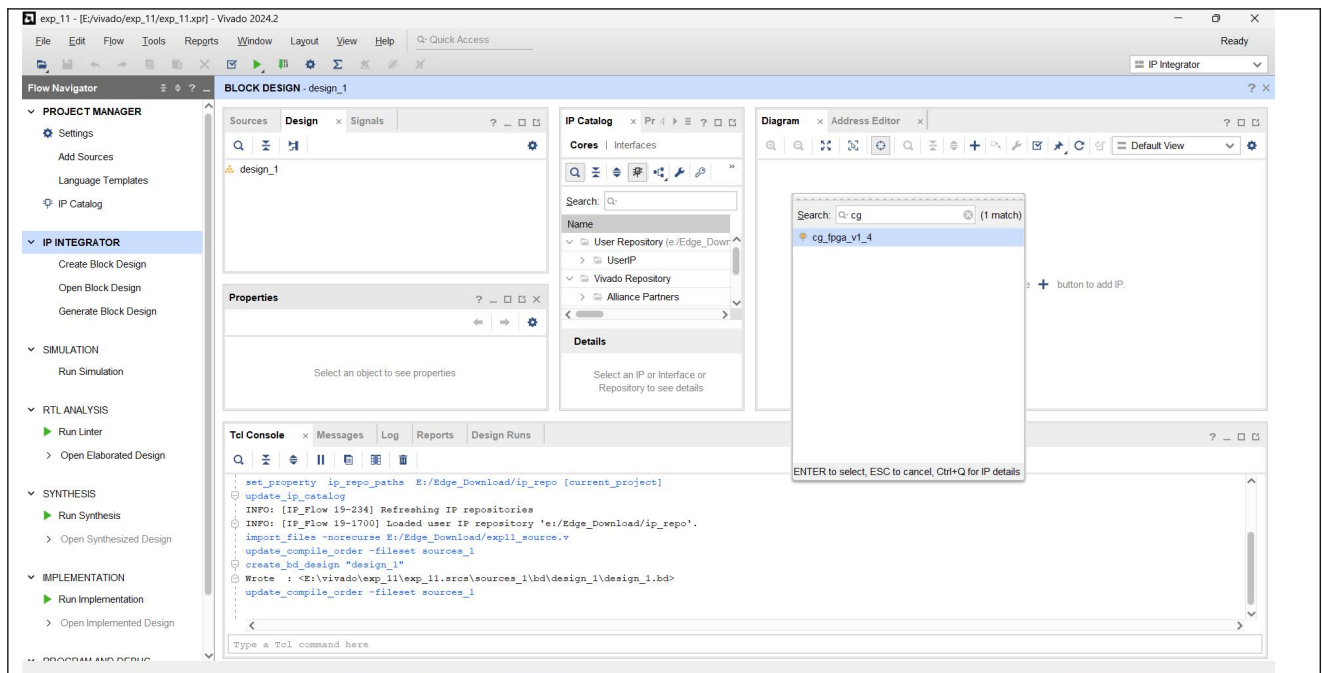
(3) 在 Vivado 项目中，点击 Settings→IP→Repository，将上一步解压后的 ip_repo 文件夹的位置添加进 IP 搜索目录。



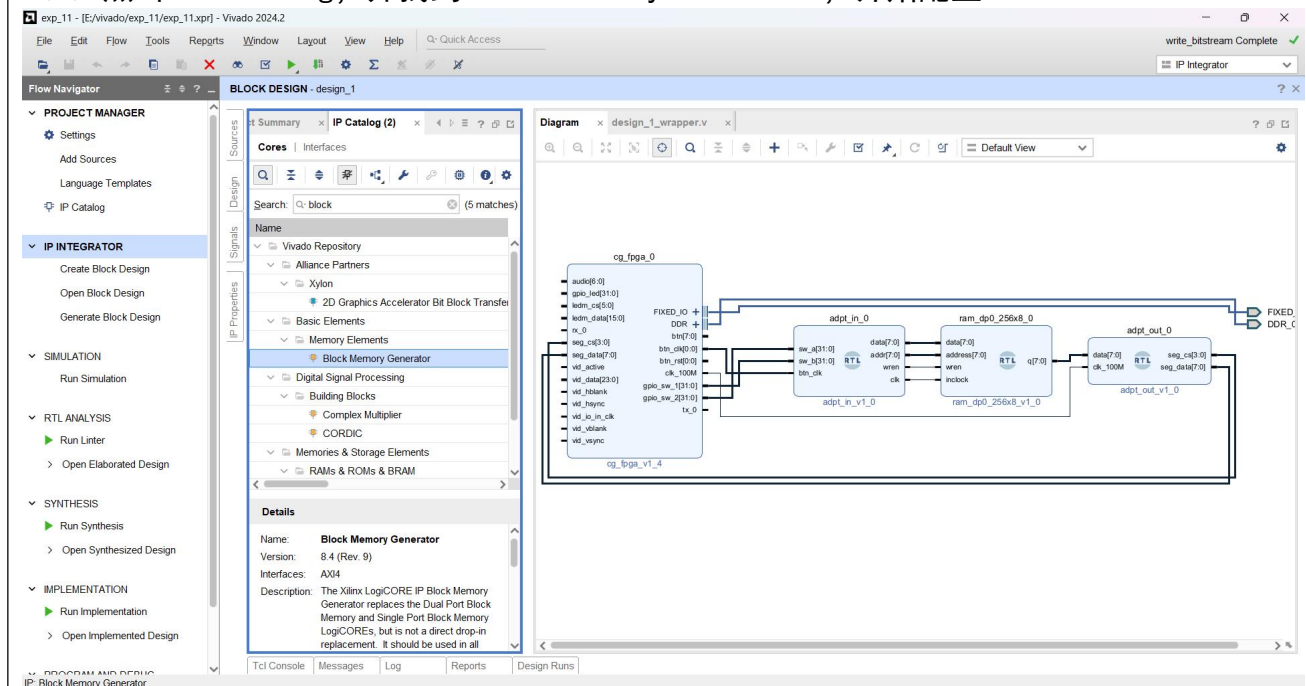
(4) 点击 Sources 窗口中的+, 选择 Add or create design sources → Next → Add File, 添加实验源代码文件。



(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 BRAM



(7) 按照如下方法定制单口 BRAM, 并命名为 ram0:

Customize IP

Block Memory Generator (8.4)

Documentation
IP Location
Switch to Defaults

IP Symbol

Power Estimation

Show disabled ports

|| + BRAM_PORTA

Component Name

ram0

Basic

Port A Options

Other Options

Summary

Interface Type

Native

Generate address interface with 32 bits

Memory Type

Single Port RAM

Common Clock

ECC Options

ECC Type

Error Injection Pins
Single Bit Error Injection

Write Enable

Byte Write Enable

Byte Size (bits)

9

Algorithm Options

Defines the algorithm used to concatenate the block RAM primitives. Refer datasheet for more information.

Algorithm

Minimum Area

Primitive

8kx2

OK

Cancel

Customize IP

Block Memory Generator (8.4)

Documentation
IP Location
Switch to Defaults

IP Symbol

Power Estimation

Show disabled ports

|| + BRAM_PORTA

Component Name

ram0

Basic

Port A Options

Other Options

Summary

Memory Size

Write Width

8

Range: 1 to 4608 (bits)

Read Width

8

Write Depth

256

Range: 2 to 1048576

Read Depth

256

Operating Mode

Write First

Enable Port Type

Always Enabled

Port A Optional Output Registers

Primitives Output Register

Core Output Register

SoftECC Input Register

REGCEA Pin

Port A Output Reset Options

RSTA Pin (set/reset pin)

Output Reset Value (Hex)

0

Reset Memory Latch

Reset Priority

CE (Latch or Register Enable)

READ Address Change A

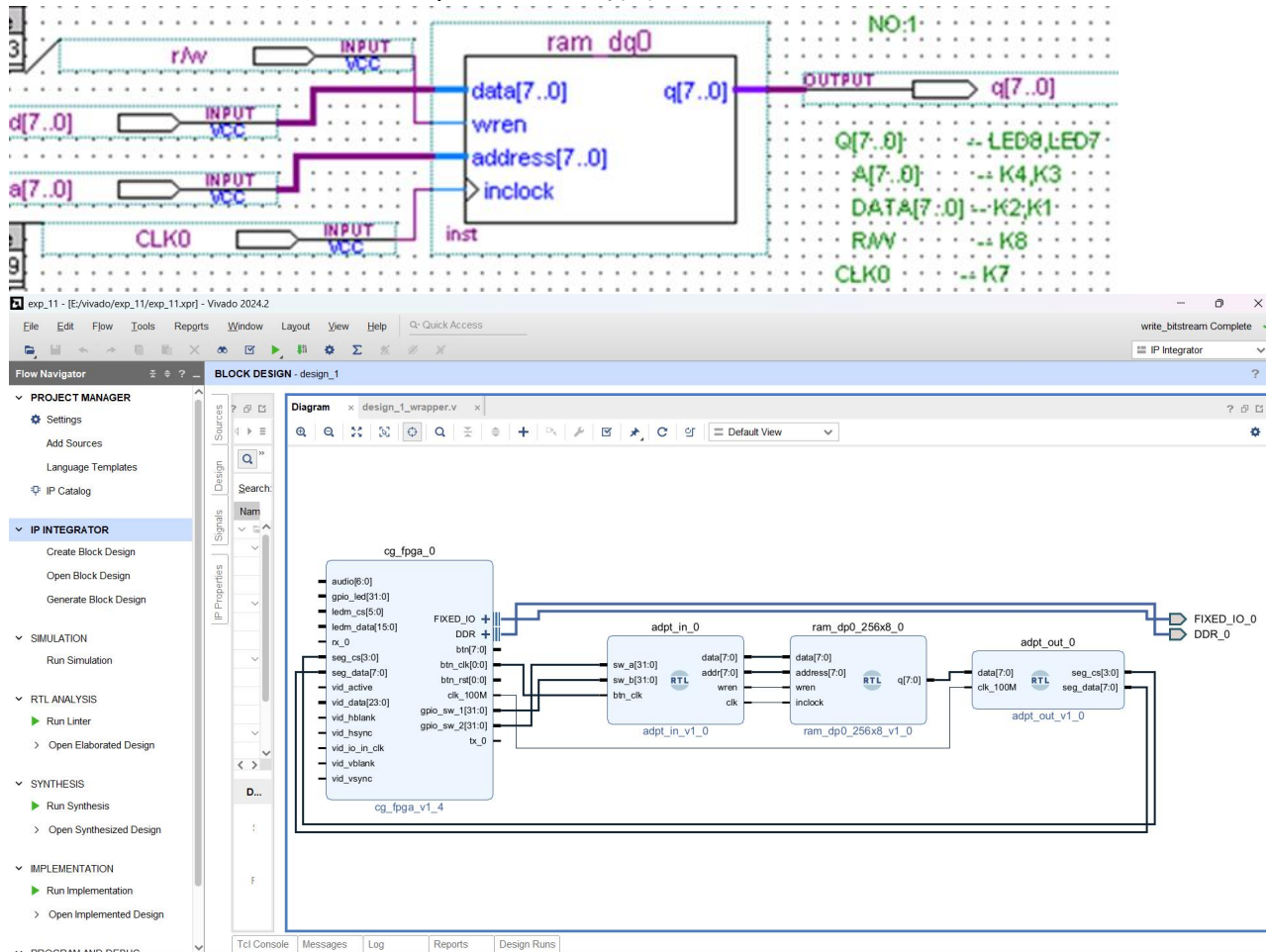
Read Address Change A

OK

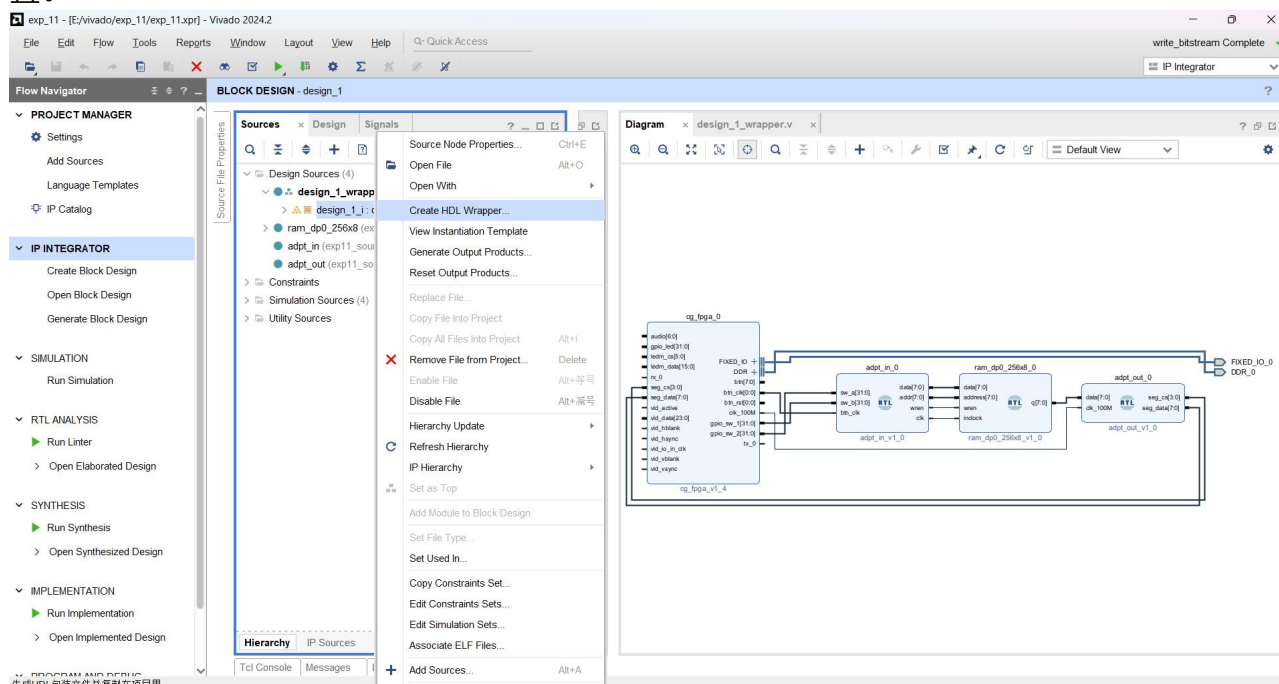
Cancel

(8) 根据要求完成原理图的输入。BRAM 也能使用初始化文件 (*.coe)，请在 Block Memory

Generator 配置窗口的 Other Options 下进行添加。

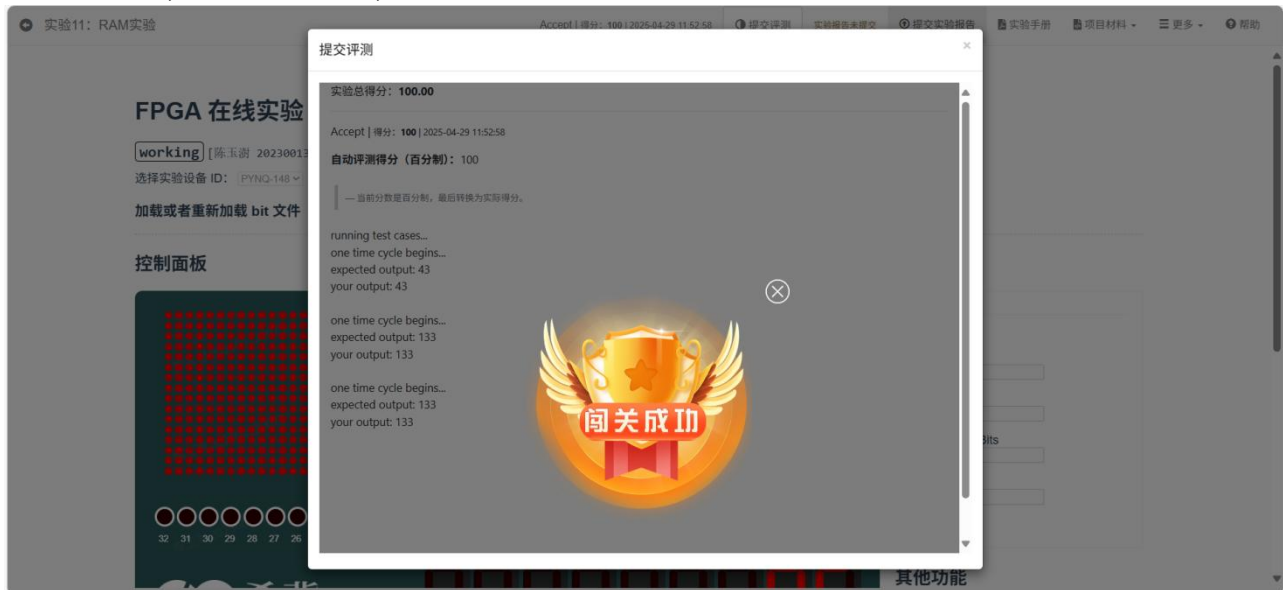


(9) 右击 Sources 下顶层设计图标→Create HDL Wrapper，待 Wrapper 正确生成后，点击左下方 Generate Bitstream，开始综合并生成 bit 文件。注意：综合前 wrapper 模块应被设置为顶层（加粗表示），若自动设置错误，需右击 wrapper 图标点击 Set as Top 手动设置。



(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

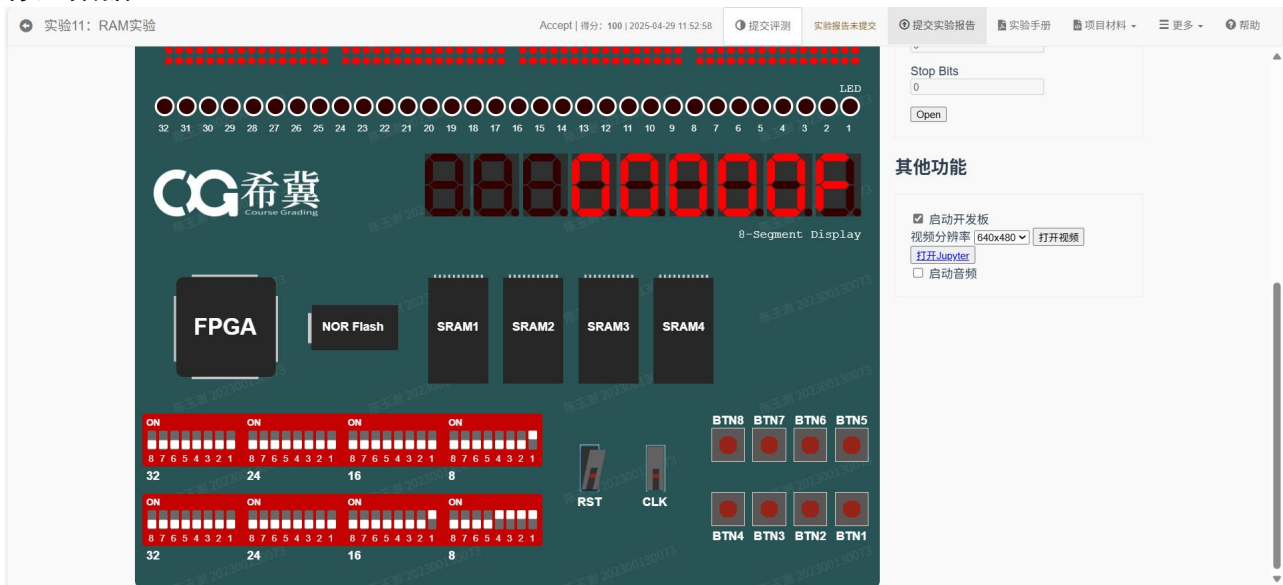
钮，然后在下拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 *.bit 文件，后点击 send，即可将本地 bit 文件烧写至希冀远程 FPGA。



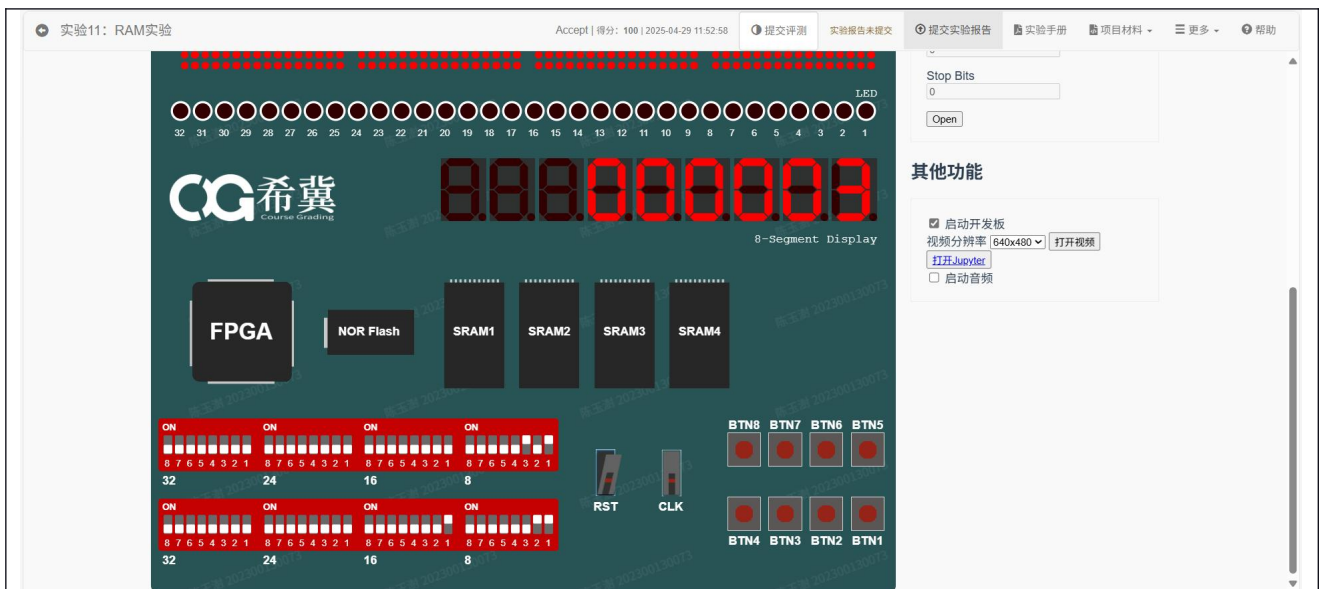
首先控制读出初始化数据，与载入的初始化文件中的数据进行比较，读出的数据将显示在数码管上。然后控制写入一些数据，再读出比较。

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 8-1 上，data_in 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，data_out 信号的管脚绑定在数码管 2-1 上。

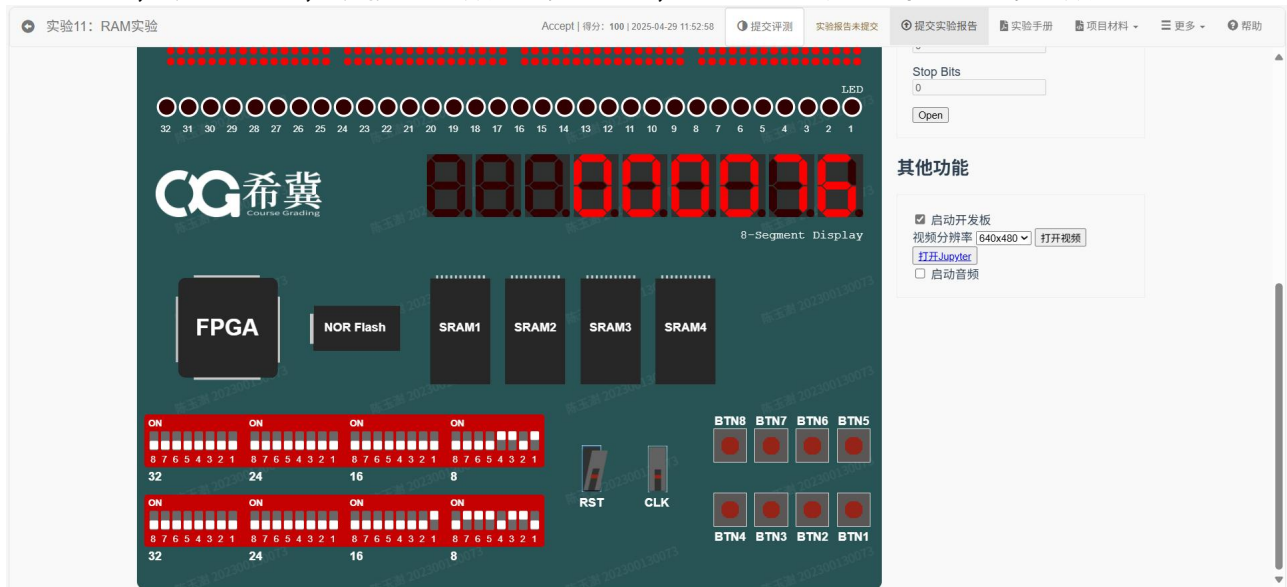
首先，载入初始化的数据，使 wren=1，使能写入数据到 RAM 中，addr=1 的地址位置上存入数据 15。



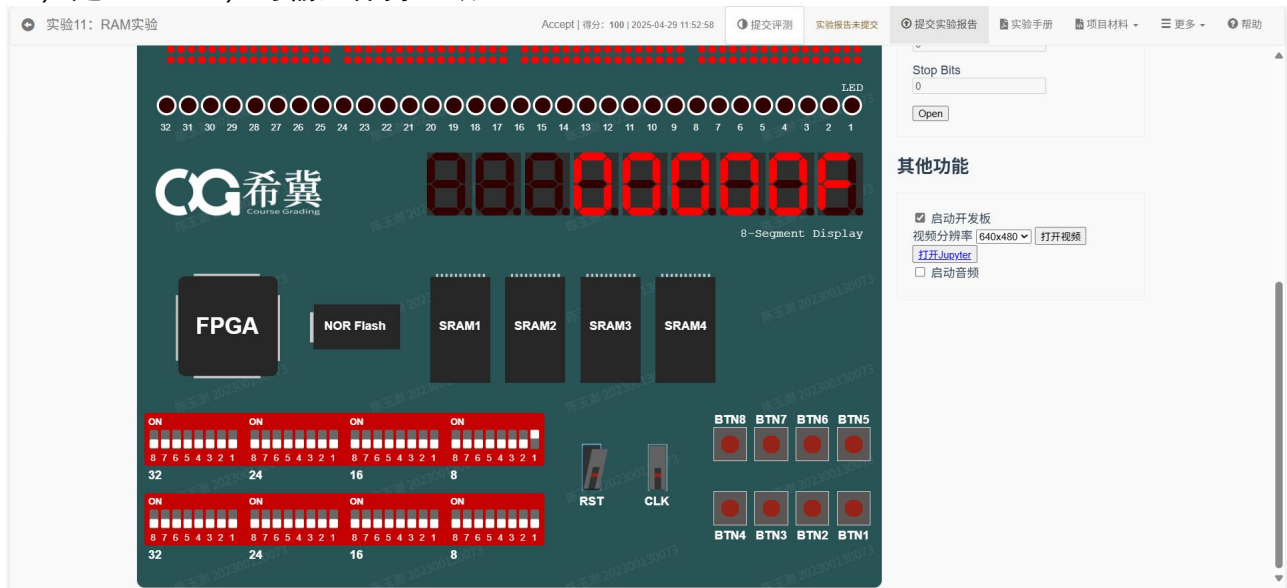
同理，使 wren=1，使能写入数据到 RAM 中，addr=5 的地址位置上存入数据 3。



最后，使 `wren=1`，使能写入数据到 RAM 中，`addr=13` 的地址位置上存入数据 76。

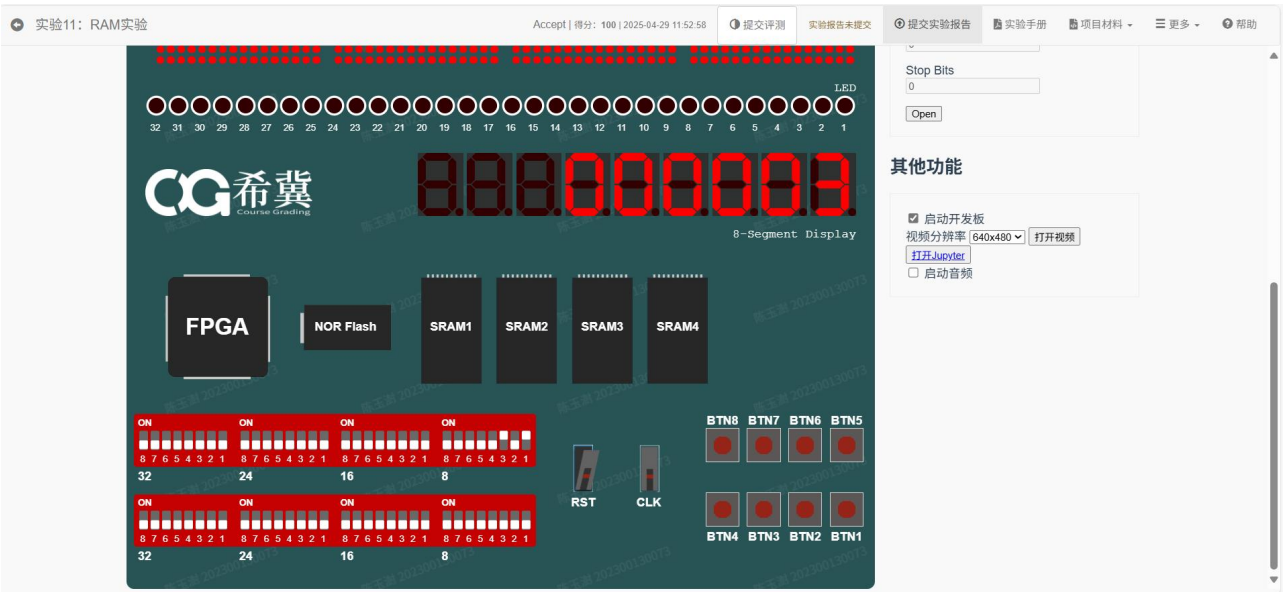


接着，使 `wren=0`，使能读出数据，输入要读的地址 1，读出的数据显示在数码管 2-1 上，是 15 (F)，与输入保持一致。

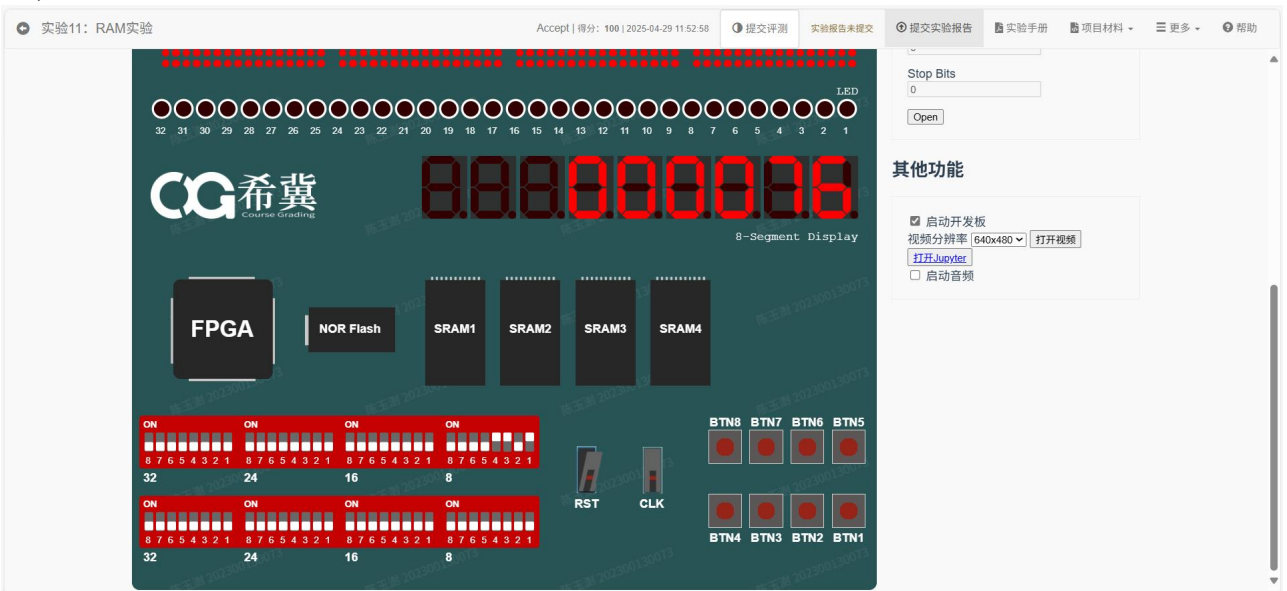


使 `wren=0`，使能读出数据，输入要读的地址 5，读出的数据显示在数码管 2-1 上，是

3, 与输入保持一致。



使 wren=0, 使能读出数据, 输入要读的地址 13, 读出的数据显示在数码管 2-1 上, 是 76, 与输入保持一致。



经过检验, 读出的数据, 与载入的数据进行比较完全一致, 符合实验预期。

结论分析与体会:

本次实验通过 Vivado 完成了 FPGA 中随机存储器 RAM 的配置与功能验证, 掌握了单口 Block RAM (BRAM) 的工作特性及配置方法。

实验中, 利用 Block Memory Generator 定制单口 RAM, 成功设置数据宽度、深度、读写模式等参数, 并通过 .coe 文件完成存储单元初始化。经硬件测试, 在时钟同步下, 读写控制信号 (wren) 有效控制数据写入与读出, 指定地址的输入数据与数码管显示结果一致, 验证了 RAM 动态读写功能的正确性。

通过本次实验, 不仅熟悉了 RAM 在 FPGA 中的实现原理与读写机制, 还提升了 Vivado 环境下的开发技能, 包括项目搭建、IP 核配置、时序设计及硬件调试等, 为后续数字系统中缓存设计及实时数据处理积累了实践经验, 奠定了 FPGA 复杂应用开发的基础。

就...处理的：

本实验进展顺利，没有出现问题。