

组成原理实验课程第 五 次实验报告

实验名称	存储器实现			班级	张金老师
学生姓名	杨冰雪	学号	2110508	指导老师	董前琨
实验地点	实验楼 A306		实验时间	2023.5.22	

1、实验目的

- 1. 了解只读存储器 ROM 和随机存取存储器 RAM 的原理。
- 2. 理解 ROM 读取数据及 RAM 读取、写入数据的过程。
- 3. 理解计算机中存储器地址编址和数据索引方法。
- 4. 理解同步 RAM 和异步 RAM 的区别。
- 5. 掌握调用 xilinx 库 IP 实例化 RAM 的设计方法。
- 6. 熟悉并运用 verilog 语言进行电路设计。
- 7. 为后续设计 cpu 的实验打下基础。

2、实验内容说明

- 1. 掌握存储器的工作原理，明白 ROM 和 RAM，同步和异步的区别；学习并掌握调用 xilinx 库 IP 进行设计的方法。
- 2. 确定存储器的输入输出端口及宽度、深度和写使能设计。
- 3. 画出包含外围模块的整体设计框图，补充完善图。
- 4. 编写 verilog 代码。
- 5. 完成调用存储器模块的外围模块的设计，并编写代码。
- 6. 对代码进行综合布局布线下载到实验箱里 FPGA 板上，进行上板验证；对演示结果进行拍照作为实验报告。
- 7. 实验结束后，需按照规定的格式完成实验报告的撰写。

3、实验原理图

5.3 同步 RAM 的顶层展示模块

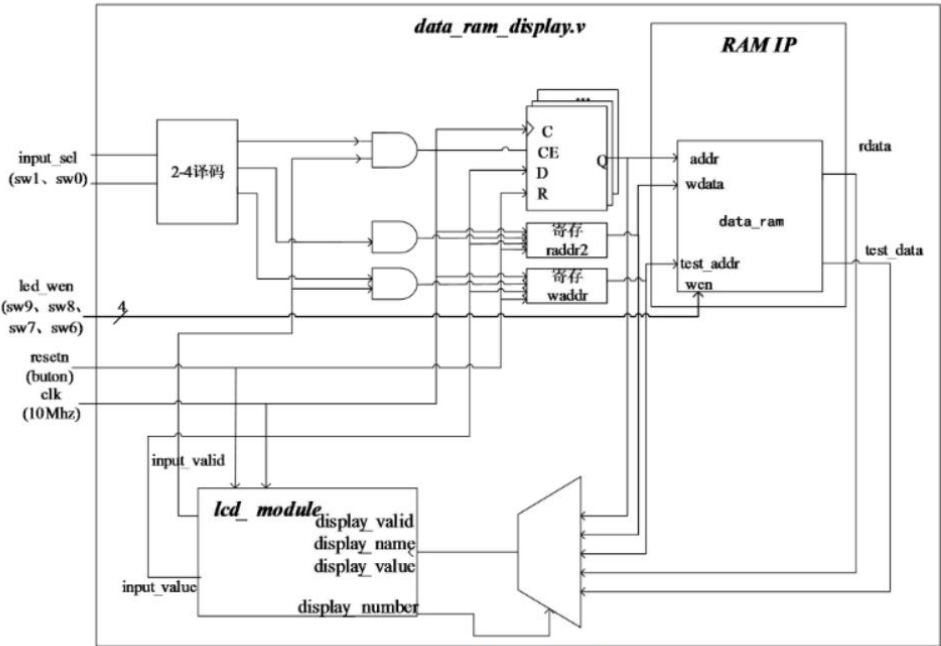


图 6.13 data_ram 参考设计的顶层模块框图

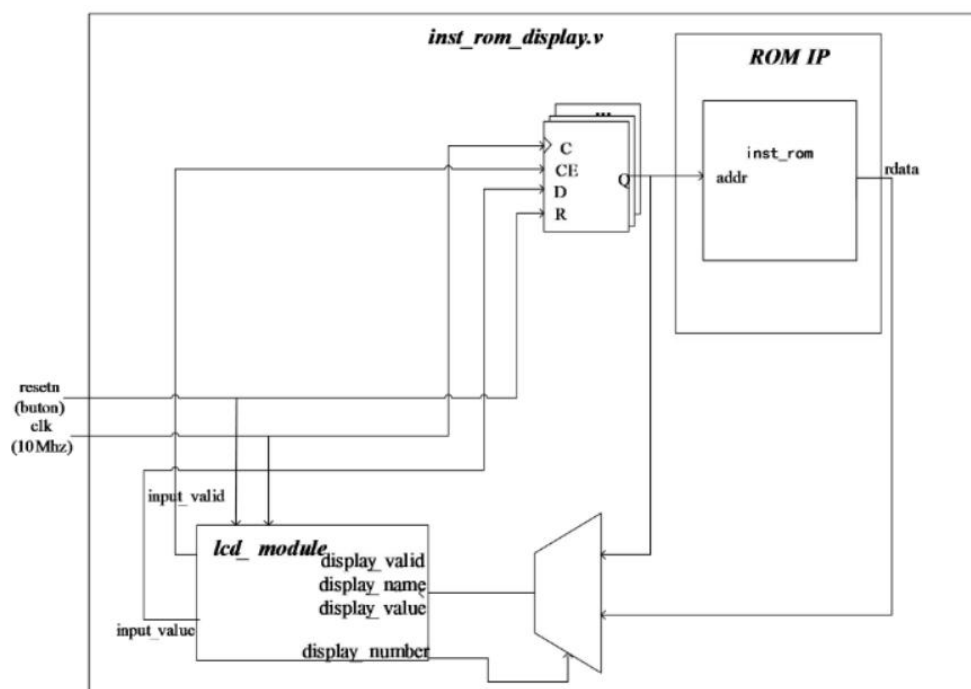


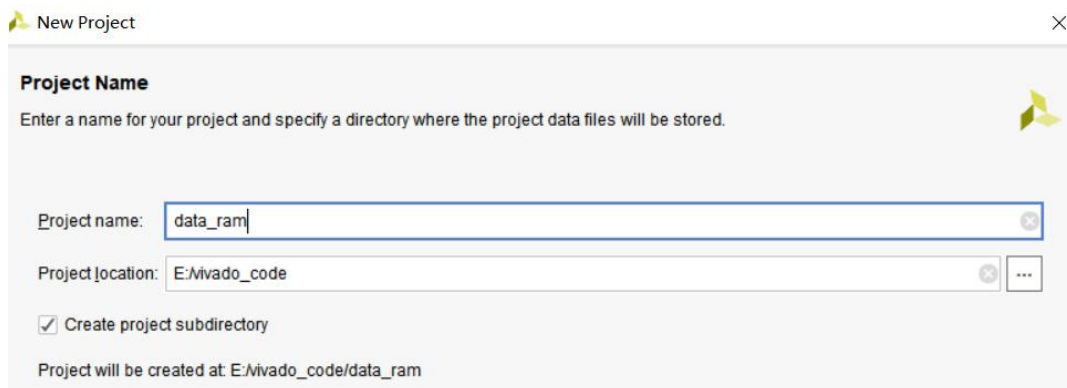
图 6.14 inst_rom 参考设计的顶层模块框图

4、实验步骤

1. 调用 IP 库实例化同步 RAM 和同步 ROM

1) 创建工程

新建工程 data_ram 和 inst_rom

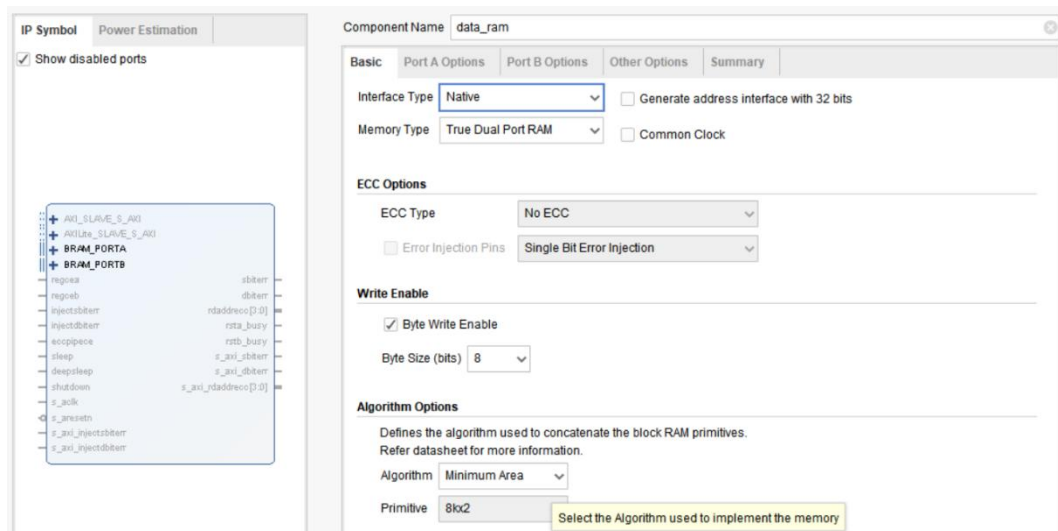


2) 生成 IP 核 ROM

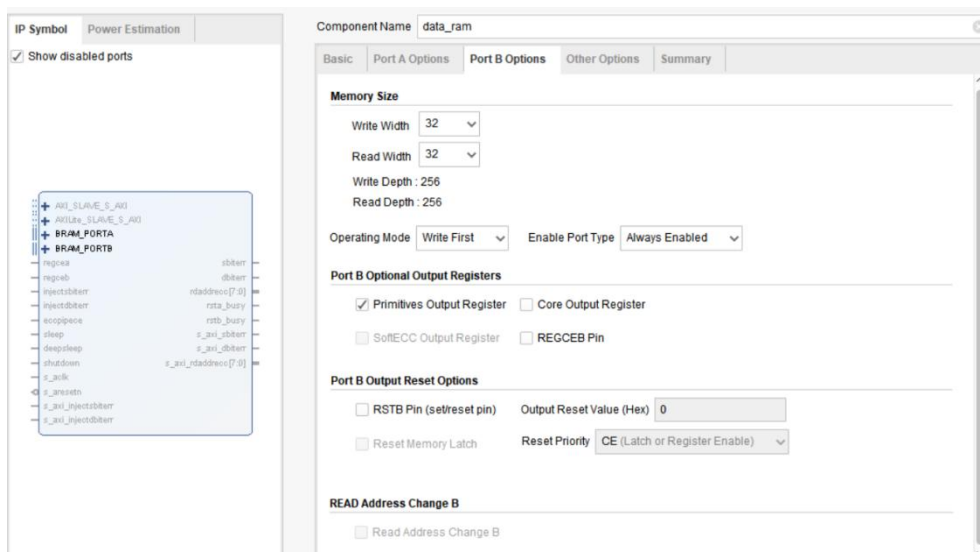
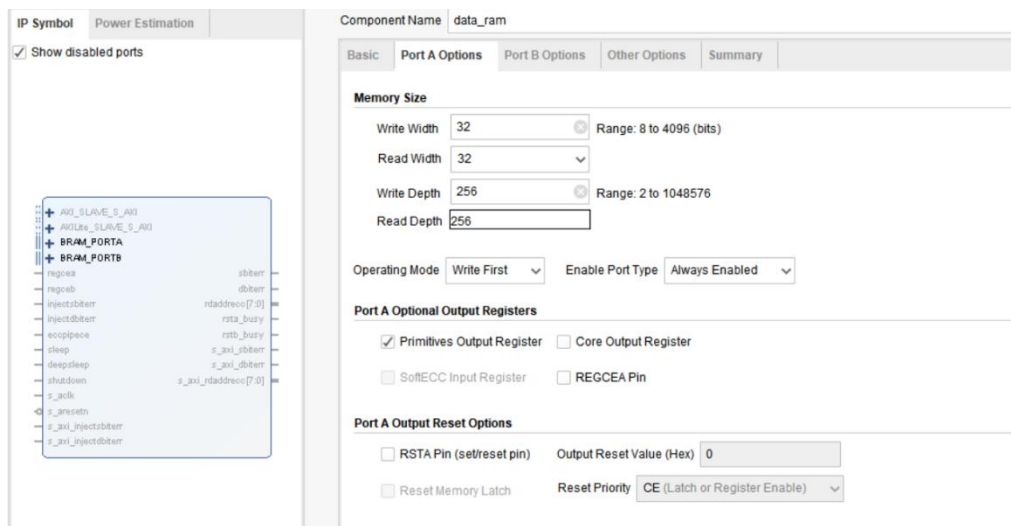
点击“IP Catalog”，在右侧列表中双击选择“Memories and Storage Elements”->“RAMs & ROMs & BRAM”中的“Block Memory Generator”。

Name	AXI4	Status	License	VLNV
Memory Interface Generator (MIG 7 Series)	AXI4	Production	Included	xilinx.com:ip:mig
RAMs & ROMs				
Distributed Memory Generator		Production	Included	xilinx.com:ip:dmg
RAMs & ROMs & BRAM				
Block Memory Generator	AXI4	Production	Included	xilinx.com:ip:blkmem

然后需要依次选择 Memory 的参数，Component Name 输入“data_ram”，选择 Memory 类型为“True Dual Port RAM”，一个端口作为正常的读写端口，一个端口作为调试端口。勾选“Write Enable”下的“Byte Write Enable”，“Byte Size”选 8bits。



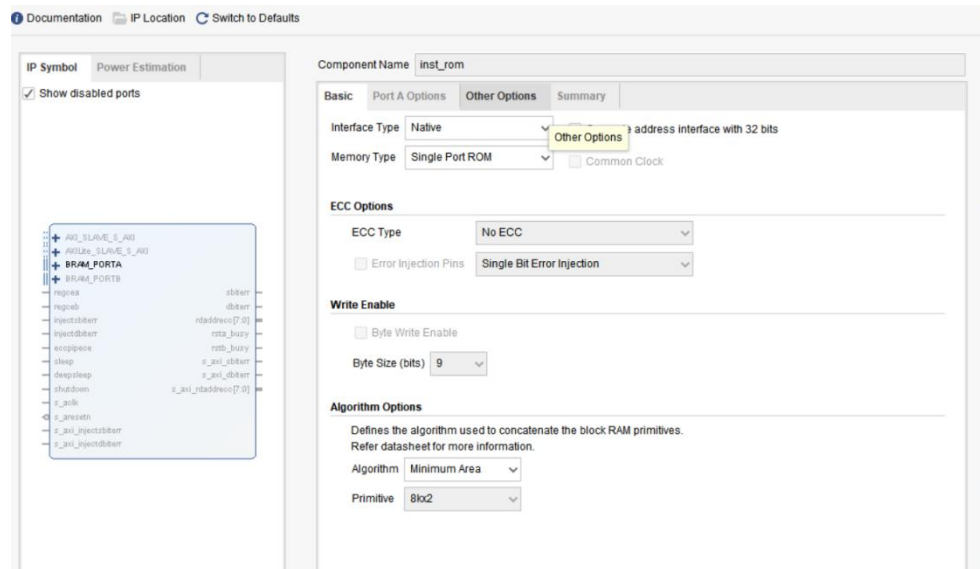
在“Port A options”选项卡下设置，RAM 宽度设置为 32 位，深度为 256，因为后续 CPU 实验是基于 32 位 数据运算的。“Enalbe Port Type”选择“Always Enabled”。在“Port B options”选项卡下进行同样的设置。



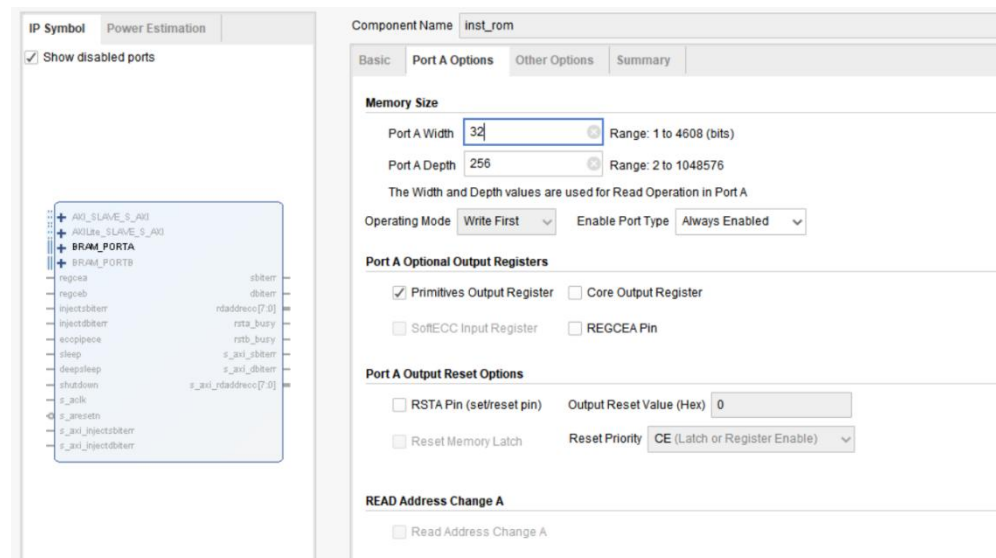
然后点击“Generate”生成 IP 核即可。

3) 生成 IP 核 RAM

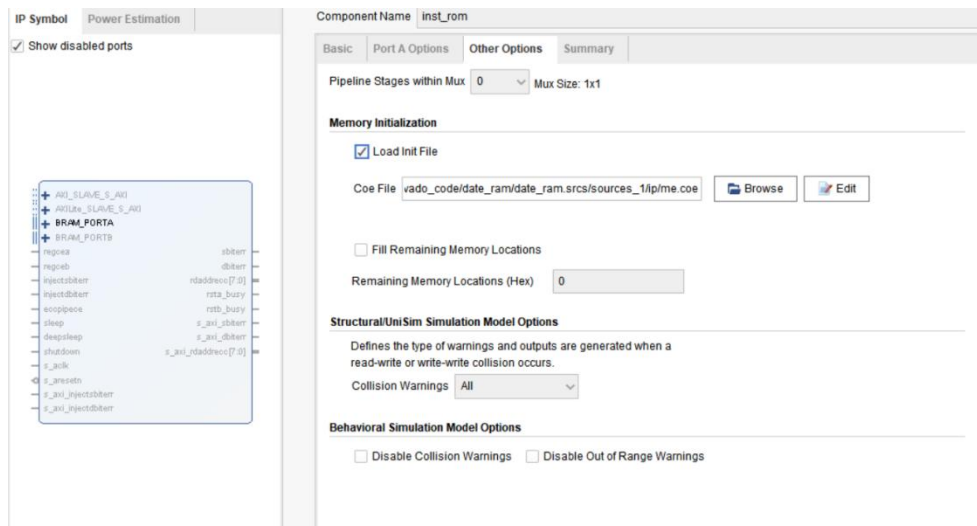
首先新建一个 IP 核，取名为 inst_rom,后续步骤同上面一样，但是 Memory 类型需要选择为“Single Port ROM”。



点击“Port A Options”设置宽度和深度，“Enable Port Type”选择“Always Enabled”，宽度需要设置为 32 位，因为一条指令占用 32 位，深度可以依据自己要执行的指令数设定，此处先设定为 256。



在“other options”中需要勾选“Load Init File”,并选中需要装载的初始化文件(.coe 文件)。



然后点击“Generate”生成 IP 核即可。

4) 添加展示外围模块

添加外围展示模块、lcd 模块和约束文件。

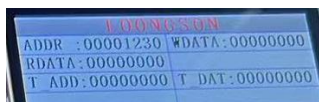
5) 最终上箱验证

5、实验结果分析

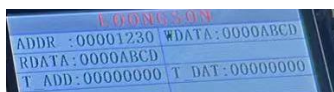
RAM 是而 **RAM** 是 **Random Access Memory** 的缩写，表示 **RAM** 可以随机读写，这可以从以下上箱图片看出，既有读写地址和读写数据，还可以随便输入地址进行测试。

1) 同步 RAM

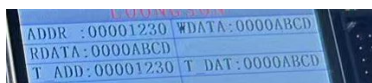
设置读写地址为 0x00001230



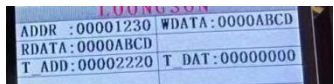
输入写数据为 0x0000ABCD



读数据显示为 0x0000ABCD，与写入数据相同，测试地址设为 0x00001230，读出的测试数据为 0x0000ABCD，与我们写入的数据相同，符合预期结果，正确！

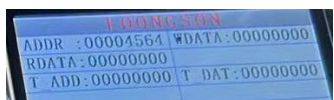


测试地址改为 0x00002220，读出的测试数据为 0x00000000，与预期结果相同，正确！

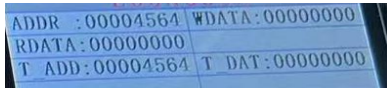


2) 异步 RAM

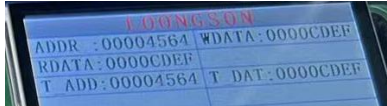
设置读写地址为 0x00004564



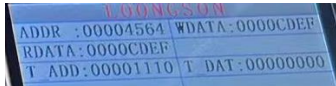
设置测试地址为 0x00004564，这时候写入数据为 0x00000000，读数据和测试数据与写入数据相同。



改变写入数据为 0x0000CDEF，读数据和测试数据都变为 0x0000CDEF。



改变测试地址为 0x00001110，测试地址也变为 0x00000000，结果正确！

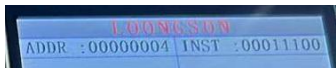


3) 同步 ROM

设置读地址为 0x00000000，读出数据为 0x24010001



设置读地址为 0x00000004，读出数据为 0x00011100



4) 异步 ROM

设置读地址为 0x00000008，读出数据为 0x00000000



ROM 是 Read Only Memory 的意思，也就是说这种存储器只能读，不能写，从一下上箱图片中可以看出只有读地址和读数据，并没有写入的接口。

6、总结感想

1) 总结一下 ROM 和 RAM 的区别

1. ROM 是 Read Only Memory 的意思，也就是说这种存储器只能读，不能写。而 RAM 是 Random Access Memory 的缩写。RAM 则可以随机读写。
2. ROM 和 RAM 都是一种存储技术，但两者原理不同，RAM 为随机存储，掉电不会保存数据，而 ROM 可以在掉电的情况下，依然保存原有的数据。
3. 速度不同，RAM 的速度要远远高于 ROM 的速度。

2) 分析一下同步存储器和异步存储器的特点，思考说明一下何时需要使用同步存储器，何时需要使用异步存储器

同步存储器是在执行完一个进程之后，一直等待系统返回值或消息，这时程序是出于阻塞的，只有接收到返回的值或消息后才往下执行其他的命令。

异步存储器的进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

如果数据存在线程间的共享，或竞态条件，就需要同步。如多个线程同时对同一个变量进行读和写的操作。当应用程序在对象上调用了需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就可以使用异步，提高效率、加快程序的响应。