

组成原理实验课程第五次实验报告

实验名称	存储器的实现实验			班级	张金
学生姓名	冯思程	学号	2112213	指导老师	董前琨
实验地点	实验楼 A306		实验时间	2023.5.22 14: 00	

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

实验任务:

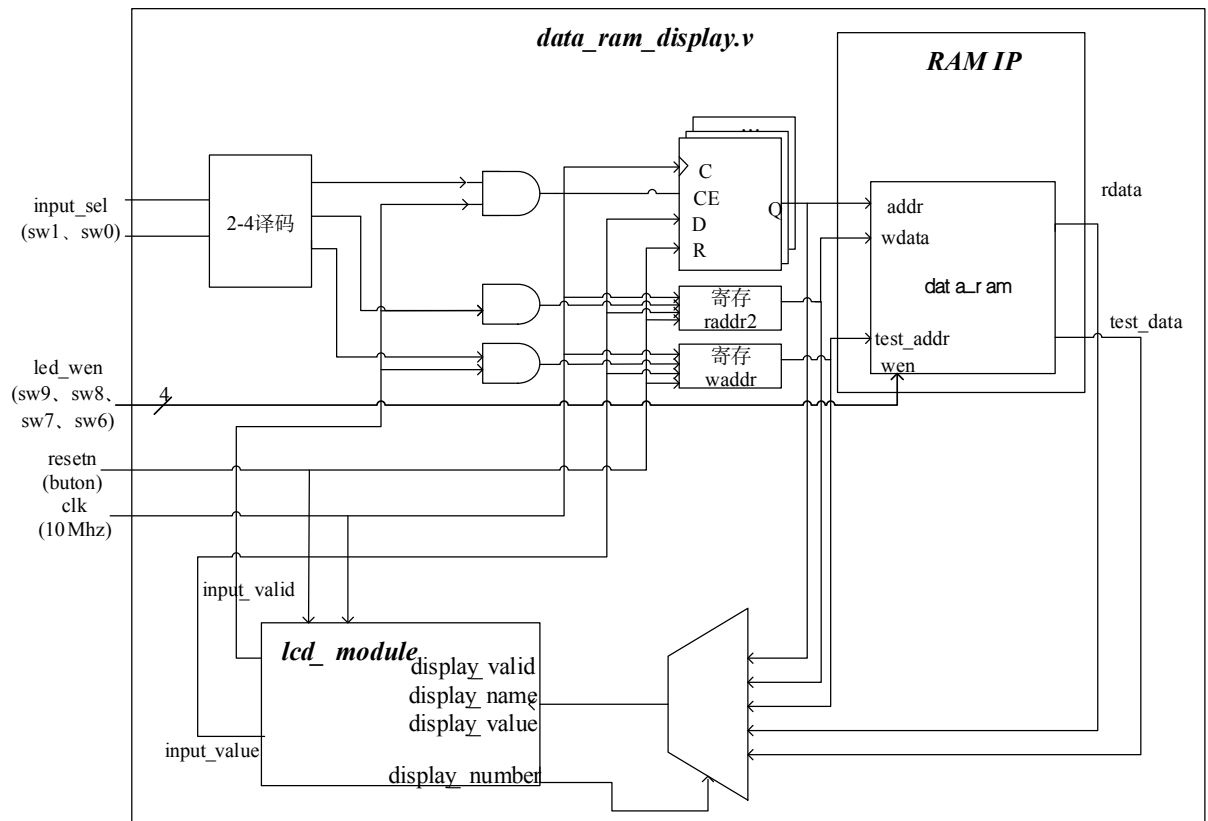
- (1) 学习存储器的设计及原理，如：ROM 读地址索引读取数据过程及时序，RAM 读写时序，同步和异步的区别等。
- (2) 学习计算机中内存地址编址和数据索引方法。
- (3) 自行设计本次实验的方案，画出结构框图，详细标出输入输出端口，确定存储器宽度、深度和写使能位数。
- (4) 学习 ISE 工具中调用库 IP 的方法。
- (5) 本次实验要求调用 xilinx 库 IP 实例化一块 RAM。实例化的 RAM 选择为同步 RAM。本次实验的 RAM 建议设置为两个端口，一个端口用来正常的读写，另一个端口作为调试端口只使用读功能用于观察存储器内部数据。
- (6) 调用 xilinx 库 IP 实例化一块 RAM，并进行仿真，得到正确的波形图。
- (7) 将以上设计作为一个单独的模块，设计一个外围模块去调用该模块，见图 6.1。外围模块中需调用封装好的 LCD 触摸屏模块，显示 RAM 的正常端口的地址、待写入的数据和读出的数据，显示调试端口的地址和读出的数据。并且需要利用触摸功能输入正常端口的地址和写数据，以及调试端口的地址。
- (8) 将编写的代码进行综合布局布线，并下载到实验箱中的 FPGA 板子上进行演示。

针对组成原理第五次的加法器实验，要求:

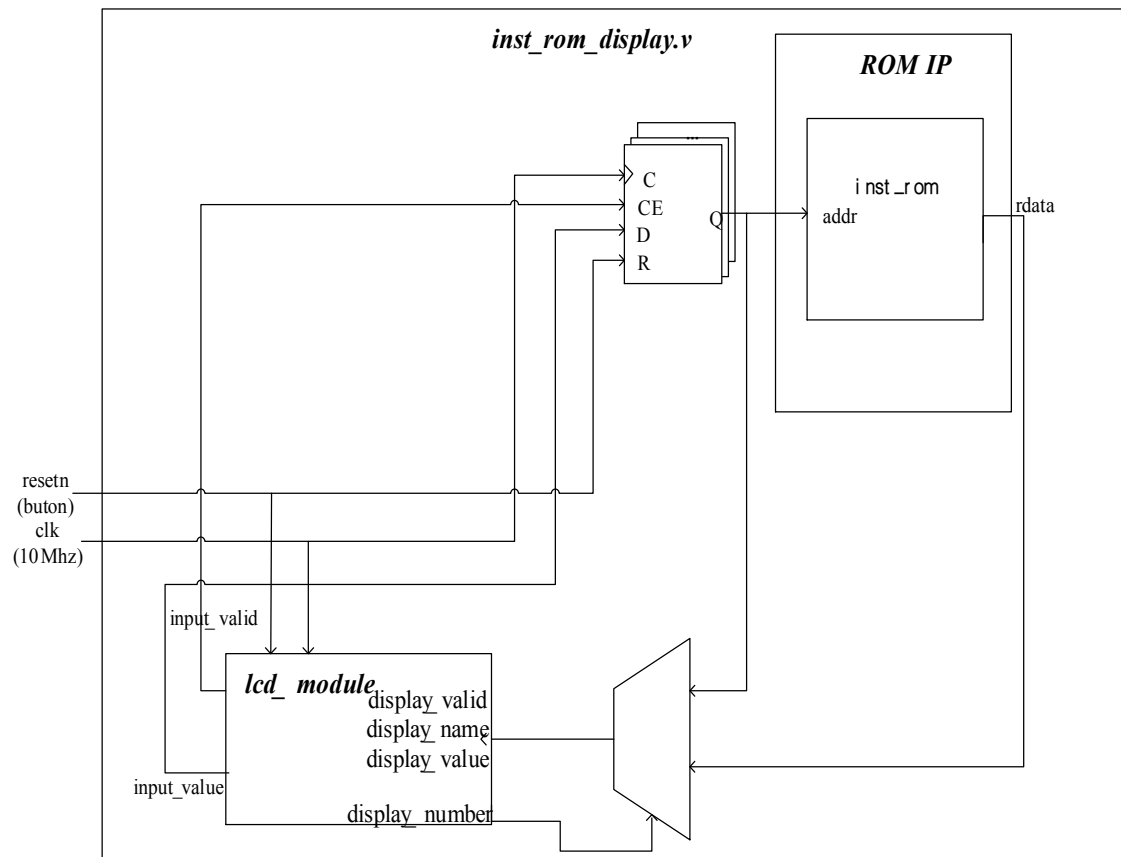
- (1) 建立四个工程分别完成同步、异步的 rom 和 ram 存储器实验，在实验箱报告中每个工程至少用一组上箱照片和介绍性文字总结验证功能。
- (2) 实验原理图使用实验指导书的图 6.13 和 6.14 即可，无需修改。
- (3) 在实验总结中用自己的话总结回答下面几个问题：1) 总结一下 ROM 和 RAM 的区别；2) 分析一下同步存储器和异步存储器的特点，思考说明一下何时需要使用同步存储器，何时需要使用异步存储器；
- (4) 按实验报告模板要求完成实验报告，并提交，注意实验报告中要有介绍分析的内容，针对实验箱照片，要解释图中信息，是否验证成功。

3、实验原理图

data_ram 参考设计的顶层模块框图:



inst_rom 参考设计的顶层模块框图:



这次实验要求没有要求对原来的程序进行修改，所以这里无需对模块图进行修改。

4、实验步骤

(1) 首先是进行同步的 RAM 的复现实验:

首先要对工程进行创建和配置, 步骤如下:

- 1 在 IP Catalog 中找到并选择"Memories and Storage Elements".
- 2 在下一个列表中选择"RAMs & ROMs & BRAM".
- 3 在可用的选项中找到并选择"Block Memory Generator".
- 4 在参数设置中, 选择"True Dual Port RAM"作为 Memory 类型.
- 5 在 Write Enable 部分, 勾选"Byte Write Enable".
- 6 在 Byte Size 选项中选择 8 位.
- 7 在"Basic"选项卡中完成设置.

接下来, 我们将在 Port A 和 Port B 选项卡中进行类似的设置:

- 1 在"Port A options"选项卡中, 将 RAM 宽度设置为 32 位.
- 2 将深度设置为 256.
- 3 在"Enable Port Type"选项中选择"Always Enabled".
- 4 在"Port B options"选项卡中, 进行与步骤 8 和步骤 9 相似的设置.

(2) 然后是同步的 ROM 的复现实验:

步骤与上文类似, 在参数设置时候要选择 Single Port ROM 作为 Memory 类型, ROM 是只读的, 所以这里没有进行测试端口的配置, 然后其他设置与上文相同, 在最后需要加载上 ROM 的初始化文件.

(3) 异步 RAM 的复现实验:

从 sourcecode 导入代码之后烧录到实验箱上进行验证.

(4) 异步 ROM 的复现实验:

同理异步的 RAM.

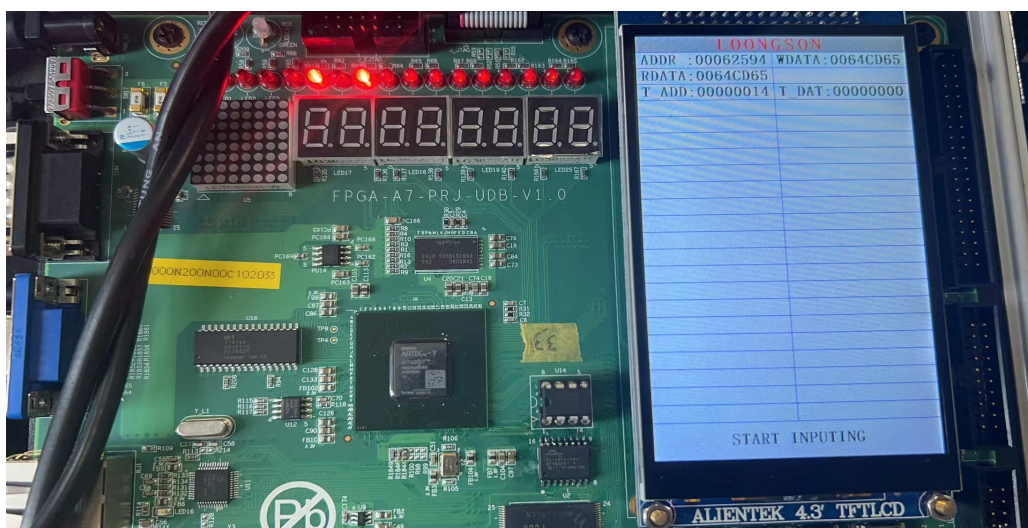
5、实验结果分析

上箱结果验证:

i 同步 RAM: (为了方便观察将 4 个拨码开关全部生效, 即使能信号一直生效)

对触摸屏上的显示进行解释:

ADDR 代表输入的读写地址, WDATA、RDATA 分别为输入的写数据和读出的读数据, T_ADD 代表的是输入 test 的地址 (即测试地址), T_DAT 即为测试地址处读出的数据. 在 ADDR 输入 00062594, 在 WDATA 输入 0064CD65, 发现读取数据全部获得同步, RDATA 就是 0064CD65, 如下图:

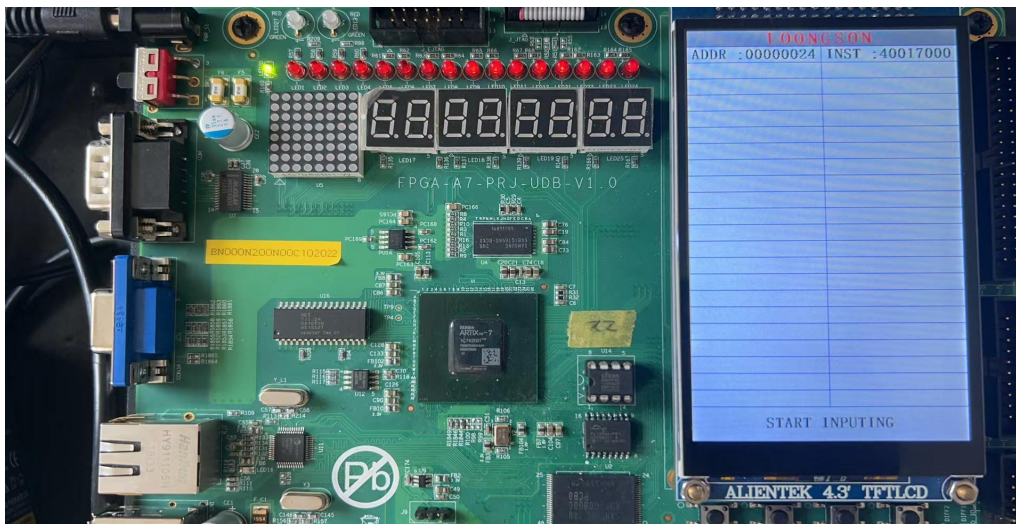


在 T_ADD 输入 00062594, 即刚才输入的数据存储的地址, 会发现 T_DAT 读出

0064CD65, 这里截图后来由于手机内存限制被误删了, 但是功能是正确无误的。根据上箱验证说明成功实现了存储器功能, 通过 test 部分可以访问地址及其数据信息。

ii 同步 ROM:

这里的 ROM 是随机的只读存储器, 我同样将使能信号一直保持有效, 方便验证, 在这里我随机输入一个地址 00000024, 读取到了改地址的数据 40017000, 如下图:



成功验证了功能的正确完整性。

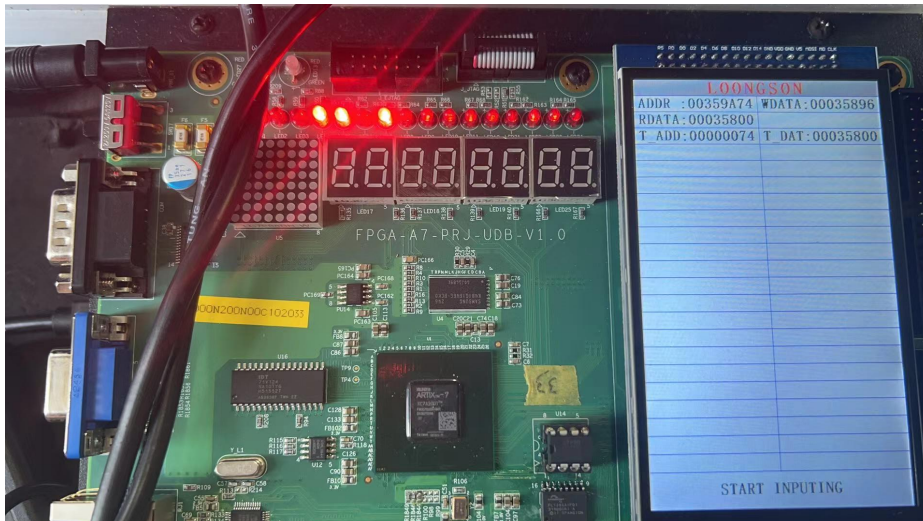
lii 异步 RAM 复现:

同上文对触摸屏上的显示进行解释:

ADDR 代表输入的读写地址, WDATA、RDATA 分别为输入的写数据和读出的读数据, T_ADD 代表的是输入 test 的地址 (即测试地址), T_DAT 即为测试地址处读出的数据。说明:

下图一中读写地址为 000369A8, 写入数据为 00358D66, 将四个拨码开关全部拨下进行同步, 可以看到此时 RDATA 与 WDATA 的数据一致。而在图二中, 只拨下了前三个拨码开关使数据写入, 地址为 00359A74 可以看到写入数据为 00035896, 而读出的数据为 00035800, 可以看到最后两位并没有被写入。



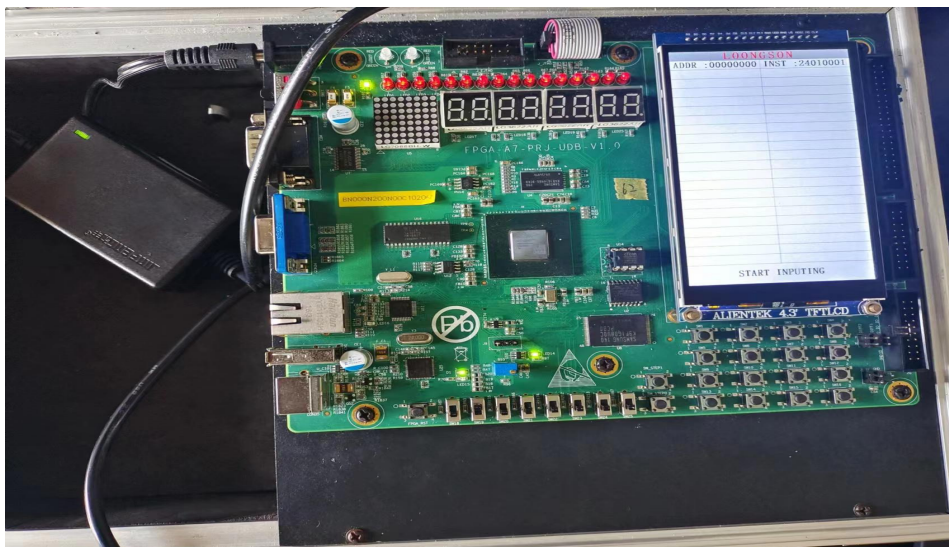


在 T_ADD 输入 000369A8 和 00359A74，即刚才输入的数据存储的地址，会发现 T_DAT 读出是 00358D66 和 00035800，这里截图后来由于手机内存限制被误删了，但是功能是正确无误的。根据上箱验证说明成功实现了存储器功能，通过 test 部分可以访问地址及其数据信息。

Vi 异步 ROM 复现:

也将使能信号一直生效，这里的触摸屏显示和同步 ROM 是完全一样的，它也可以根据输入的地址读取数据，对同步 ROM 的主要区别在于不需要系统时钟来指挥同步读取，所以速度也会较慢。

下图中，输入地址是 00000000，读取到的数据是 24010001:



成功输出，目标功能成功实现。

6、总结感想

首先对实验要求的问题进行解答:

1) ROM 和 RAM 的区别:

答:

ROM: ROM 是一种只能读取数据而无法写入或修改的存储器。在本实验中，他只能读取某个地址中存储的数据。ROM 的内容在电源关闭后也能保持不变。

RAM: RAM 是一种能够读取和写入数据的存储器。在本实验中，它既可以读取某地址的数

据，也可以写入数据到任意地址。**RAM** 的内容在电源关闭后会被清空，所以需要持久保存的数据应当存储在非易失性存储器中。

区别：

1 可读性：

ROM：ROM 是只读存储器，其中存储的数据在制造过程中被编程，并且在正常操作期间不能被修改或擦除。它包含了固化的指令、固件和永久性数据。

RAM：RAM 是可读写存储器，可以在运行时读取和写入数据。它用于临时存储正在使用的程序和数据。

2 数据保存：

ROM：ROM 中的数据在断电或重新启动后仍然保持不变。它可以被用来存储固定的系统指令和设置，以及设备的固件。

RAM：RAM 中的数据是临时存储的，当计算机断电或重新启动时，其中的数据将被清除。

RAM 用于存储正在运行的程序、操作系统和临时数据。

3 访问速度：

ROM：ROM 的访问速度通常比 **RAM** 慢，因为 ROM 中的数据在制造过程中被固化，无需实时读取或写入。

RAM：RAM 的访问速度通常比 ROM 快，因为它需要频繁地读取和写入数据。

4 存储容量：

ROM：ROM 的存储容量通常比较大，可以存储大量的固定数据和指令。

RAM：RAM 的存储容量相对较小，用于临时存储正在运行的程序和数据。

5 可擦写性：

ROM：ROM 中的数据通常无法被擦除或修改。一些特定类型的 ROM (如 EEPROM 和 Flash 存储器) 具有擦写功能，可以在一定条件下修改其中的数据。

RAM：RAM 中的数据可以随时读取、写入和修改。

总结起来，**ROM** 是只读存储器，用于存储固定的指令、固件和永久性数据，而 **RAM** 是可读写存储器，用于临时存储正在运行的程序和数据。**ROM** 的数据在断电后仍然保持不变，而 **RAM** 的数据是临时的，会在断电或重新启动后被清除。

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？

答：

同步存储器和异步存储器是两种不同的存储器类型，它们在数据传输和时序控制方面有着显著的特点。以下是它们的主要特点和适用场景：

同步存储器特点：

1 时钟驱动：同步存储器的读取和写入操作是由外部时钟信号驱动的。数据的读取和写入必须在时钟的上升或下降沿发生。

2 数据传输精确：由于有时钟信号的同步控制，同步存储器的数据传输是精确同步的。数据的读取和写入在特定时钟周期内完成，保证了数据的稳定性和准确性。

3 复杂性高：同步存储器通常需要额外的控制逻辑电路和时序控制电路来保证数据的同步传输，因此设计和实现上更为复杂。

4 延迟相对较大：由于同步存储器需要等待时钟信号的触发，因此其访问延迟相对较大。

异步存储器特点：

1 无需时钟信号：异步存储器的读取和写入操作不依赖外部时钟信号，而是根据存储器本身的内部时序和控制信号来进行。

2 数据传输不精确：由于没有外部时钟信号的同步控制，异步存储器的数据传输可能存在一

定的不确定性和不精确性。读取和写入的时机取决于输入和输出信号的相互匹配。

3 简单性高：异步存储器相对于同步存储器来说，不需要额外的控制逻辑电路和时序控制电路，设计和实现上相对简单。

4 访问速度快：由于无需等待时钟信号的触发，异步存储器的访问延迟相对较低，读写速度更快。

根据这些特点，我们可以考虑何时使用同步存储器和异步存储器：

同步存储器适用于对数据传输精确性要求较高的场景，例如在时序控制严格、需要确保数据稳定性和准确性的系统中，如处理器、图形卡等。同步存储器的设计和实现较复杂，但能够提供可靠的数据同步和时序控制。

异步存储器适用于对数据传输速度较为关注的场景，例如在需要快速读写操作的应用中，如缓存、磁盘驱动器等。由于异步存储器无需等待时钟信号的触发，可以实现更低的访问延迟和更高的数据传输速度。

总结起来，选择同步存储器还是异步存储器要根据具体的应用需求来决定。如果应用对数据传输的精确性和稳定性要求较高，且能够容忍较高的访问延迟和复杂性，则选择同步存储器。如果应用对数据传输速度较为关注，且对传输精确性要求较低且能够容忍一定的不确定性，则选择异步存储器。