

实验报告

实验名称：完成一个简单存储器的设计

班级：智能 1602

学号：201608010609

姓名：李鹏飞

实验目标

设计一个简单存储器，实现存储器的功能

实验要求

- 采用 VHDL 或 Verilog 语言
- 自定义存储器的输入和输出
- 实现存储器的存储功能

实验内容

1.存储器简介

存储器是用来存储程序和各种数据信息的记忆部件。存储器可分为主存储器(简称主存或内存)和辅助存储器(简称辅存或外存)两大类。和 CPU 直接交换信息的是主存。

主存的工作方式是按存储单元的地址存放或读取各类信息，统称访问存储器。主存中汇集存储单元的载体称为存储体，存储体中每个单元能够存放一串二进制

码表示的信息，该信息的总位数称为一个存储单元的字长。存储单元的地址与存储在其中的信息是一一对应的，单元地址只有一个，固定不变，而存储在其中的信息是可以更换的。

指示每个单元的二进制编码称为地址码。寻找某个单元时，先要给出它的地址码。暂存这个地址码的寄存器叫存储器地址寄存器(MAR)。为可存放从主存的存储单元内取出的信息或准备存入某存储单元的信息，还要设置一个存储器数据寄存器(MDR)。

2.存储器的特点

① 设置多个存储器并且使他们并行工作。本质：增添瓶颈部件数目，使它们并行工作，从而减缓固定瓶颈。

② 采用多级存储系统，特别是 Cache 技术，这是一种减轻存储器带宽对系统性能影响的最佳结构方案。本质：把瓶颈部件分为多个流水线部件，加大操作时间的重叠、提高速度，从而减缓固定瓶颈。

③ 在微处理机内部设置各种缓冲存储器，以减轻对存储器存取的压力。增加 CPU 中寄存器的数量，也可大大缓解对存储器的压力。本质：缓冲技术，用于减缓暂时性瓶颈。

3.存储器的分类

3.1RAM(Random Access Memory，随机存取存储器)

RAM 的特点是：电脑开机时，操作系统和应用程序的所有正在运行的数据和程序都会放置其中，并且随时可以对存放在里面的数据进行修改和存取。它的工作需要由持续的电力提供，一旦系统断电，存放在里面的所有数据和程序都会自

动清空掉，并且再也无法恢复。

01.DRAM (Dynamic RAM, 动态随机存取存储器)

这是最普通的 RAM，一个电子管与一个电容器组成一个位存储单元，
DRAM 将每个内存位作为一个电荷保存在位存储

单元中，用电容的充放电来做储存动作，但因电容本身有漏电问题，因此
必须每几微秒就要刷新一次，否则数据会丢失。存取时间和放电时间一致，约
为 2~4ms。因为成本比较便宜，通常都用作计算机内的主存储器。

02.SRAM (Static RAM, 静态随机存取存储器)

静态，指的是内存里面的数据可以长驻其中而不需要随时进行存取。每 6
颗电子管组成一个位存储单元，因为没有电容器，因此无须不断充电即可正常
运作，因此它可以比一般的动态随机处理内存处理速度更快更稳定，往往用来
做高速缓存。

3.2ROM(READ Only Memory, 只读存储器)

ROM 是线路最简单半导体电路，通过掩模工艺，一次性制造，在元件正常
工作的情况下，其中的代码与数据将永久保存，并且不能够进行修改。一般应
用于 PC 系统的程序码、主机板上的 BIOS (基本输入/输出系统 Basic
Input/Output System)等。它的读取速度比 RAM 慢很多。

存储器程序框架

考虑到存储器的主要原理就是读取、存储数据。

我们将模拟器的框架设计如下：

```
--定义主体
entity mem is
port(
    --定义接口
    --定义使能端
);
end entity;

--定义结构
architecture mem_behav of mem is
    type memtype is array(natural range<>) of std_logic_vector(7 downto 0);
    --初始化
    signal memdata: memtype(4095 downto 0) := (
        0 => X"04",
        1 => X"00",
        2 => X"00",
        3 => X"00",
        4 => X"08",
        5 => X"00",
        6 => X"00",
        7 => X"00",
        others => X"11"
    );
--主函数
begin
    do_read: process(addrbus, read)
    variable i: integer;
    begin
        i := to_integer(unsigned(addrbus));
        if (read='1') then

            databus <= memdata(i+3) & memdata(i+2) & memdata(i+1) & memdata(i);
        else
            databus <= "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ";
        end if;
    end process do_read;
end;
```

测试

测试平台

模拟器在如下机器上进行了测试

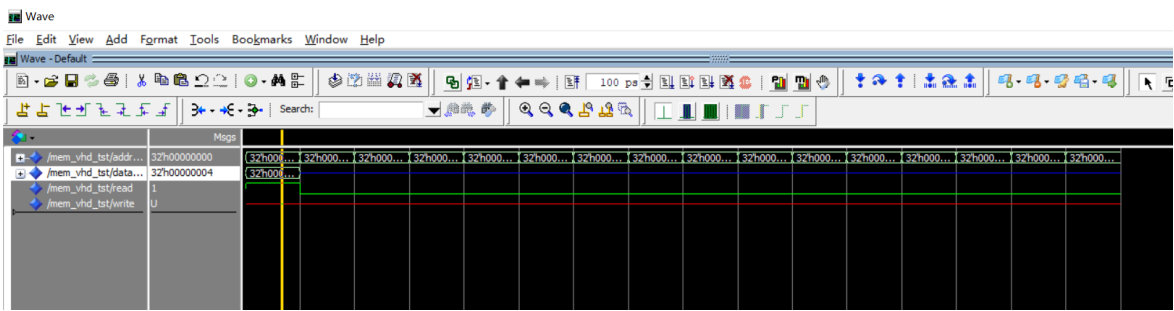
部件	配置	备注
CPU	Core i5-6700U	
内存	DDR4 12GB	
操作系统	Windows10 家庭版	

测试记录

测试环境：Quartus II 16.0

利用 testbench 文件进行测试，文件内容见 mem.vht

全局图：

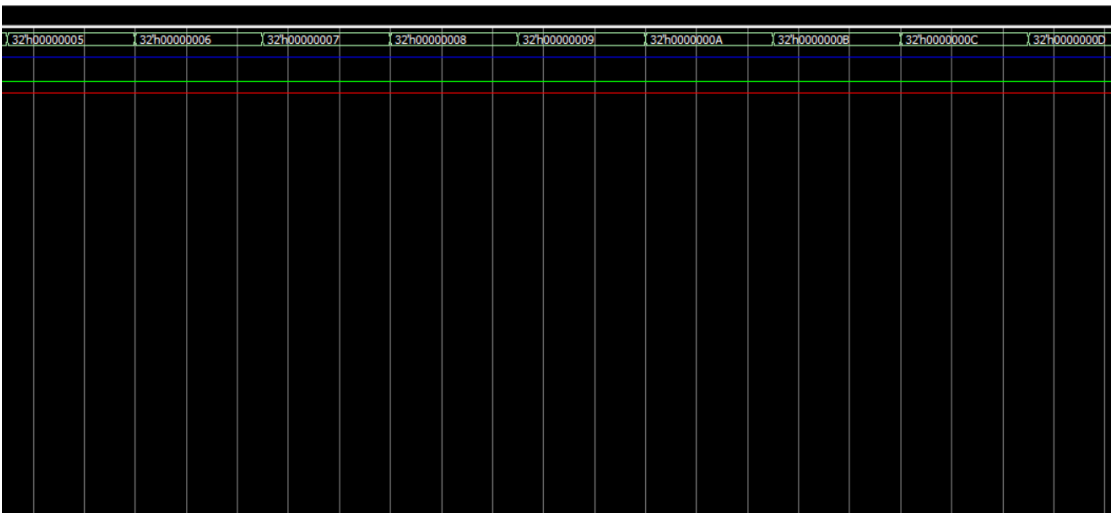


具体波形图：

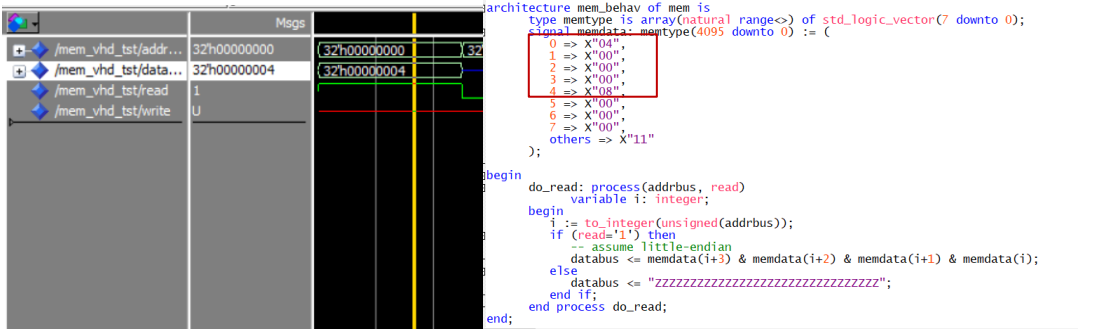


分析和结论

从测试记录来看，存储器实现了 16 个地址的跳转。



并且当 read 为 1 时，我们成功读出提前写入的数据。



根据分析结果，可以认为编写的存储器实现了所要求的功能，完成了实验目标。

实验心得体会

这个实验是做存储器，难度不是很大，主要是第一次用 testbench 文件。一开始的安装 Simulation 工具和调试花了很长时间，最后终于弄明白了整个流程，也跑出了正确的结果。