

# 简单存储器的设计与实现

班级：智能 1602 学号：201608010627 姓名：任小禹

## 一、实验目标

完成简单只读存储器的设计

## 二、实验要求

采用 VHDL 或 Verilog 语言编写

## 三、实验内容

### 1、设计内容

我设计的存储器拥有 8 位地址线，32 位数据线的只读存储器。其按字节编址，存储空间 256\*8 位。一般情况下，读出数据时一次读出四个字节（一个字）。

### 2、设计思路

编写存储器的 vhdl 文件，存储器使用 mif 文件初始化，最后使用波形仿真测试结果。

### 3、程序框架

#### vhdl 文件

##### (1) 实体声明

```
entity rxymem is
generic( data_width: natural:=32;
         addr_width: natural:=8);
port(
    clk:in std_logic;
    addrbus: in std_logic_vector((addr_width-1) downto 0);
    databus: out std_logic_vector((data_width-1) downto 0);
    we: in std_logic
);
end entity;
```

##### (2) 内存数组、地址范围、初始化文件声明

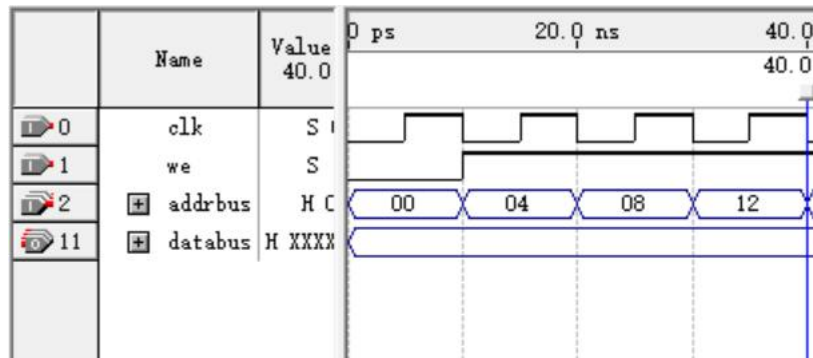
```
architecture mem_behav of rxymem is
    type memtype is array(natural range<>) of std_logic_vector(7 downto 0);--address by byte
    signal memdata: memtype((2**addr_width-1) downto 0);
    attribute ram_init_file : string;
    attribute ram_init_file of memdata: signal is "rxymem.mif";
```

我们这里按字节编址。

##### (3) 读操作进程



## 2、测试输入

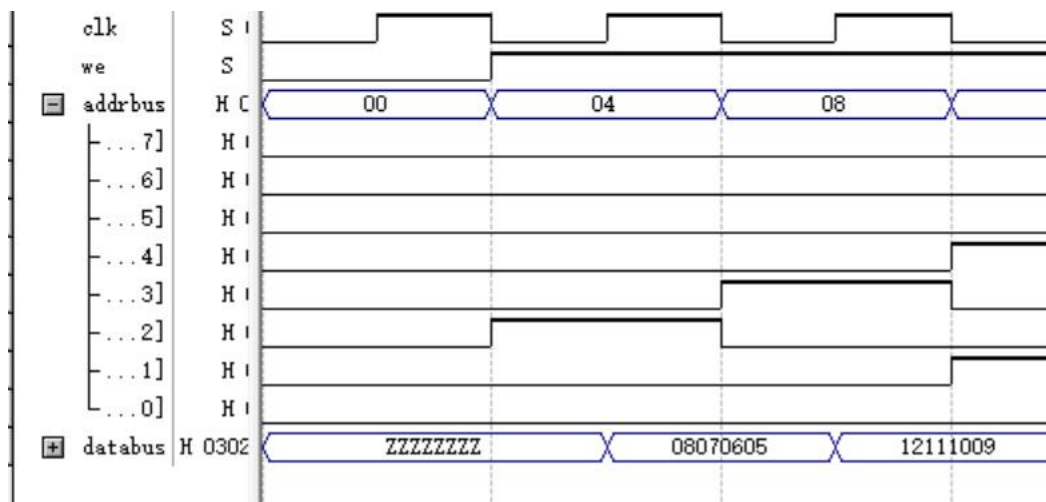


clk 信号 周期 10ns

读信号 10ns 时开始为 1

addebus 信号 每 10ns 地址单元增加 0x4，每次读四字节的数据。

## 3、测试记录



我们可以看到，刚开始 databus 为高阻态，当读信号为 1 时，才开始读出数据。时钟上升沿触发时，从 addrbus 所指地址单元开始读出 4 个字节，我们默认是小端机器，所以高字节保存在内存的高地址中。

## 五、分析与结论

从测试记录来看，简单存储器实现了读操作，根据我们在存储器所存数据，而且为小端，小端高字节保存在内存的高地址中：

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	01	02	03	04	05	06	07	08
8	09	10	11	12	13	14	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0

最后得到的数据 0x08070605, 0x12111009 正确。根据分析结果, 可以认为所设计的简单存储器实现了所要求的功能, 完成了实验目标。

## 六、遇到的问题

1、初始化存储器 mif 文件不能和存储器设计 vhdl 文件联系起来:

查阅相关资料发现 Quartus II 9.0 vhdl 里面有 ram\_init\_file VHDL Synthesis Attribute 的用法, 只需要在存储器信号后面加上 attribute 语句, 声明字符串类型和初始化文件即可。

```
type mem_t is array(0 to 255) of unsigned(7 downto 0);
signal ram : mem_t;
attribute ram_init_file : string;
attribute ram_init_file of ram :
signal is "my_init_file.mif";
```

## 七、收获与体会

存储器的设计也还算是比较简单, 之前设计的时候, 都是直接将 vhdl 文件生成 symbol 文件后就可以使用 mif 文件进行激励了, 直接在 vhdl 进行初始化的用法还没有用过, 还好后来在 Quartus II 的官方文件里找到了, 问题也就迎刃而解了。这次存储器设计最大的收获是存储器数组的声明与实现, 以及相关读操作进程的实现, 锻炼了编写 VHDL 语言的能力, 获益匪浅, 设计出来之后, 也是很有成就感。