



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

计算机组成原理 实验三报告

班 级_____9191062301
学生姓名_____孙傲歆
学 号_____919106840333
任课教师_____王晓峰

2021 年 11 月

实验三 存储器实验

1. 实验目的:

1. 了解随机存取存储器 RAM 的原理。
2. 理解 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. 熟悉并运用 verilog 语言进行电路设计。
5. 为后续设计 cpu 的实验打下基础。

2. 实验设备:

1. 装有 Xilinx Vivado 的计算机一台
2. LS-CPU-EXB-002 教学系统实验箱一套

3. 实验内容:

1. 学习存储器的设计及原理和计算机中内存地址编址和数据索引方法。
2. 搭建一个异步的 RAM，能够正确地将数据写入指定地址并读出。
3. 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它，外围模块中需要调用封装好的触摸屏模块。
4. 向地址为 50H-55H 的六个单元送 70H-75H 六个数，验收时任意指定一个地址，输出其存储的数据。
5. 进行综合布局布线，下载到实验箱的 FPGA 板上进行演示。

4. 实验思路:

本实验要求实现随机存储器 RAM，实际上在 verilog 代码层面，我们可以使用数组来当做存储器使用，首先在存储器模块中分配一些数据单元，并将数据清零，然后在实验指定单元中存入相应的数据。然后在具体的 display 模块，我们可以通过控制相关的开关、按钮，来实现读、写信号的控制。也就是说读信号有效时我们将输入的数据存储到对应地址的 ram 数组中，写信号有效时我们将对应地址的 ram 数组的数据读出并显示。这便实现了一个简单的异步 RAM。

5. 实验代码:

由于代码量较大，所以这里只展示关键部分的代码和引脚部分代码。

1) dcram.v

```
module dcram(  
    input clk,
```

```

    input we,
    input [7:0] addr,
    input [31:0] datain,
    output [31:0] dataout
);

reg [31:0] ram[0:127];
assign dataout = ram[addr];
always @(posedge clk) begin
    if(we) ram[addr] <= datain;
end

integer i;
initial begin
    for (i = 0; i < 128; i = i+1)
        ram[i] = 0;
    ram[8'h50] = 32'h70;
    ram[8'h51] = 32'h71;
    ram[8'h52] = 32'h72;
    ram[8'h53] = 32'h73;
    ram[8'h54] = 32'h74;
    ram[8'h55] = 32'h75;
end
endmodule

```

2) dcram_display.v

```

//触摸板输入
always @(posedge clk) begin
    if (!resetsn) begin
        addr <= 8'd0;
    end
    else if (input_valid && !input_sel) begin
        addr <= input_value;
    end
end

always @(posedge clk) begin
    if (!resetsn) begin
        datain <= 32'd0;
    end
    else if (input_valid && input_sel) begin
        datain <= input_value;
    end
end
end

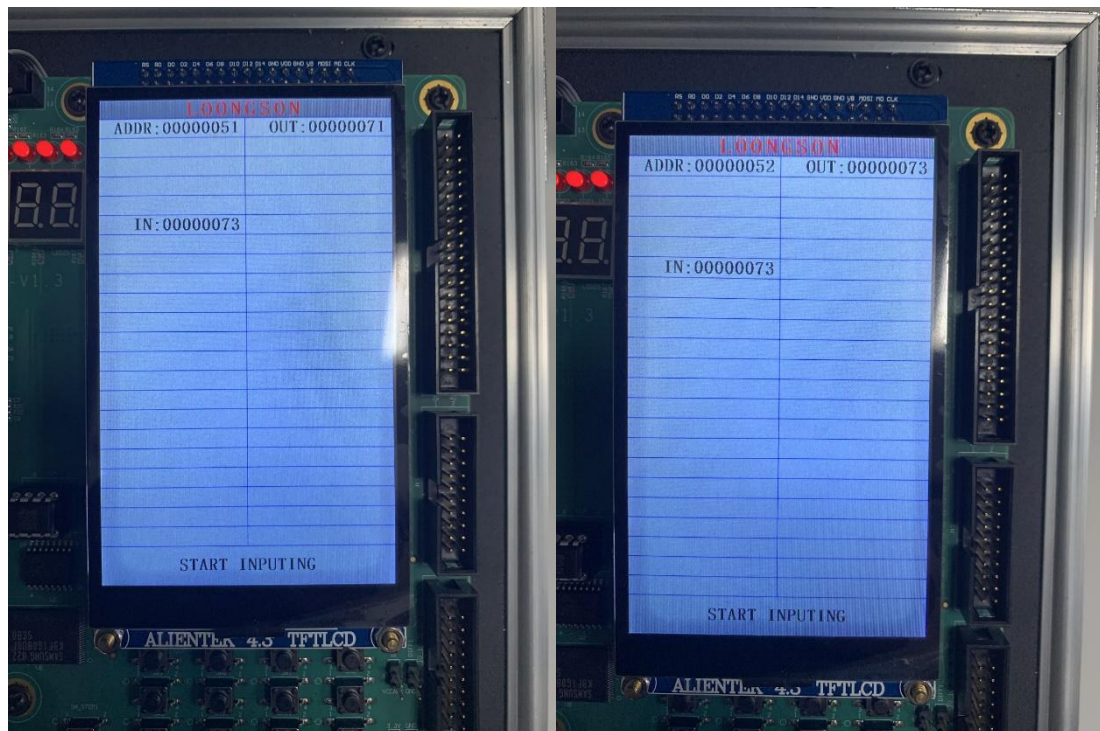
```

```

//触摸板输出
always @(posedge clk)
begin
    case(display_number)
        6'd1 :
        begin
            display_valid <= 1'b1;
            display_name  <= "ADDR";
            display_value <= addr;
        end
        6'd11 :
        begin
            display_valid <= 1'b1;
            display_name  <= "IN";
            display_value <= datain;
        end
        6'd2 :
        begin
            display_valid <= 1'b1;
            display_name  <= "OUT";
            display_value <= dataout;
        end
        default :
        begin
            display_valid <= 1'b0;
            display_name  <= 40'd0;
            display_value <= 32'd0;
        end
    endcase
end

```

6. 结果分析:



如上图所示，是该实验的上板验证结果，使用触摸屏对数据进行读写操作。IN 是期望输入的数据，当写信号有效时，将会将 IN 的数据写入 ADDR 对应单元当中去；而当读信号有效时，IN 输入无论如何输入都不会改变 ADDR 单元的数据，但相应的，会在 OUT 栏中输出 ADDR 单元中的数据。上图就表示 51H 单元中存储了数据 71H，而 52H 单元中存储了 73H 数据（数据已经过了修改）。然而根据上述代码我们可以看出，实际上一共分配了 128 个数组空间，相当于 128 个 ram，地址从 0 到 128 编号，然而这里我们仅使用了 50H-55H 这 6 个单元。

7. 心得体会:

通过这次实验让我了解了随机存储器是如何存储和读出数据的，此次实验较为简单，只要搞懂了 RAM 的实现逻辑，在代码层面其实很好实现。

然而我也知道实际上的 RAM 远比我们实现的要复杂得多，计算机中真正的 RAM 由存储矩阵、地址译码器、读/写控制器、输入/输出、片选控制等多个部分组成。而不是像我们实现的 RAM 那样只是简单的存入数组，然后读取。

总之，通过这次实验让我对数据通路及信号传送等知识有了更深的理解，并且也明白随机存储器的组成结构及其原理，经过前 3 次实验，CPU 中的运算器以及存储器两大模块已经完成，为下面单周期 CPU 实验奠定了一定的基础。