

数字电路与数字系统实验

实验七 存储器

姓名： 你猜

学号： 你猜

班级： 你猜

邮箱： 你猜

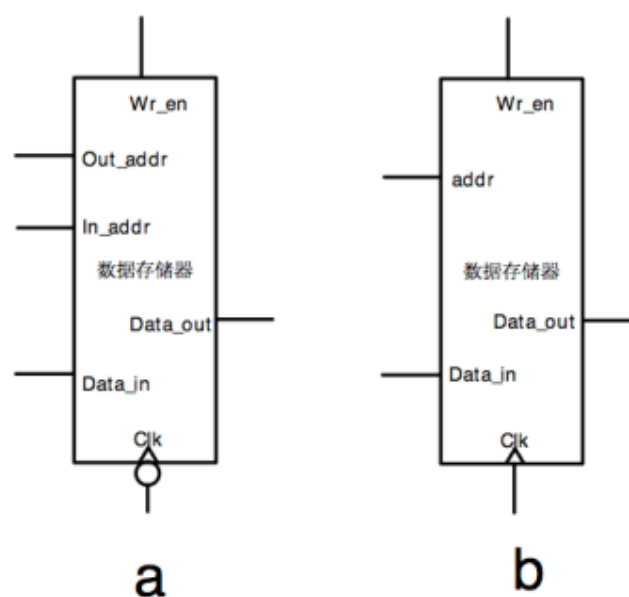
实验时间： 你猜

一、实验目的

学习存储器的原理，通过 verilog 语言设计、实现两个大小为 16×8 的存储器，均可以进行读写，其中 ram2port 用 IP 核生成。

二、实验原理

存储器是一组存储单元，用于在计算机中存储二进制的数据，如图 7-1 所示。存储器的端口包括输入端、输出端和控制端口。输入端口包括：读/写地址端口、数据输入端口等；输出端口一般指的是数据输出端口；控制端口包括时钟端和读/写控制端口。存储器的工作过程如下：



写数据：在时钟（clk）有效沿（上升或下降沿），如果写使能（Wr_en，也可以没有使能端）有效，则读取输入总线（Data_in）上的数据，将其存储到输入地址线（In_addr）所指的存储单元中。

读数据：存储器的输出可以受时钟和使能端的控制，也可以不受时钟和使能端的控制。如果输出受时钟的控制，则在时钟有效沿，将输出地址所指示的单元中的数据，输出到输出总线上（Data_out）；如果不受时钟的控制，则只要输出地址有效，就立即将此地址所指的单元中的数据送到输出总线上。

三、实验环境

Quartus 18.1、FPGA 开发板

四、实验过程

设计思路：

参考实验手册 7.2 的内容，可以很轻松的写出 ram1；而 ram2port 是用 IP 核生成，按照实验手册的步骤，可以很顺利的完成。

设计代码：

ram.v:

```
1 module ram(clk, we, inaddr, inaddr1, indata, hex0, hex1, outdata2);
2   input clk;
3   input we;
4   input [3:0] inaddr;
5   input [3:0] inaddr1;
6   wire [3:0] addr;
7   wire [3:0] addr1;
8   wire [7:0] outdata1;
9   input [7:0] indata;
10  output [6:0] hex0, hex1;
11  output [7:0] outdata2;
12  assign addr = inaddr;
13  assign addr1 = inaddr1;
14  ram1 R1(.clk(clk), .we(we), .inaddr(inaddr), .outaddr(addr), .indata(indata), .outdata1(outdata1));
15  ram2port R2(.clock(clk), .data(indata), .wren(we), .wraddress(inaddr1), .rdaddress(addr), .outdata2(outdata2));
16  show S(.outdata(outdata1), .HEX0(hex0), .HEX1(hex1));
17 endmodule
18
```

ram1.v:

```
1 module ram1(clk, we, inaddr, outaddr, indata, outdata);
2   input clk;
3   input we;
4   input [3:0] inaddr;
5   input [3:0] outaddr;
6   input [7:0] indata;
7   //output reg [6:0] HEX0;
8   //output reg [6:0] HEX1;
9   output reg [7:0] outdata;
10
11   reg [7:0] ram [15:0];
12
13
14   initial
15   begin
16     $readmemh("C:/intelFPGA_lite/exp/exp07/mem1.txt", ram, 0, 15);
17   end
18
19   always @(posedge clk)
20   begin
21     if (we)
22       ram[inaddr] <= indata;
23     else
24       outdata <= ram[outaddr];
25   end
26
27 endmodule
28
```

ram2port 直接按照实验手册的来：
初始化：

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	00	01	02	03	04	05	06	07	-----
8	08	09	10	11	12	13	14	15	-----

show.v 是用数码管显示 ram1 的 outdata, 略...

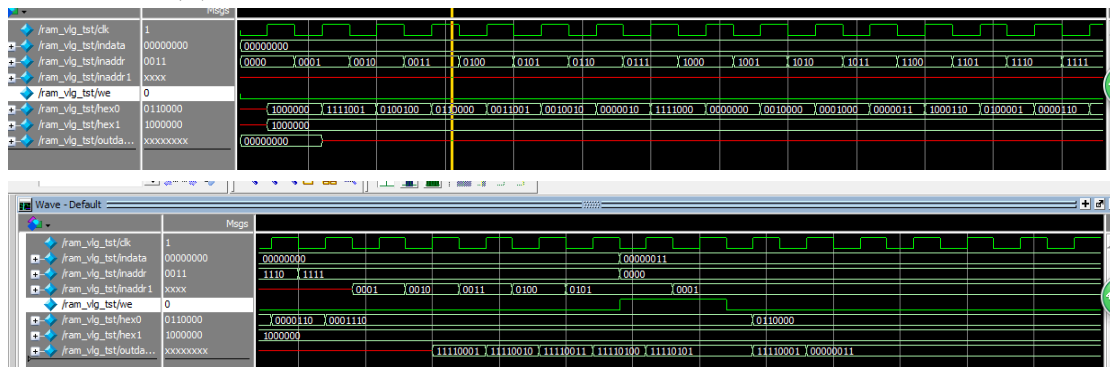
激励代码：

```

57 // code that executes only once
58 // insert code here --> begin
59 clk=0;we=0;indata=8'b00000000;inaddr=4'b0000;#8;
60                                     inaddr=4'b0001;#8;
61                                     inaddr=4'b0010;#8;
62                                     inaddr=4'b0011;#8;
63                                     inaddr=4'b0100;#8;
64                                     inaddr=4'b0101;#8;
65                                     inaddr=4'b0110;#8;
66                                     inaddr=4'b0111;#8;
67                                     inaddr=4'b1000;#8;
68                                     inaddr=4'b1001;#8;
69                                     inaddr=4'b1010;#8;
70                                     inaddr=4'b1011;#8;
71                                     inaddr=4'b1100;#8;
72                                     inaddr=4'b1101;#8;
73                                     inaddr=4'b1110;#8;
74                                     inaddr=4'b1111;#8;
75                                     inaddr1=4'b0001;#8;
76                                     inaddr1=4'b0010;#8;
77                                     inaddr1=4'b0011;#8;
78                                     inaddr1=4'b0100;#8;
79                                     inaddr1=4'b0101;#8;|
80                                     we=1;indata=2'b11;inaddr=4'b0000;#8;
81                                     inaddr1=4'b0001;#8;
82                                     we=0;indata=8'b00000011;inaddr=4'b0000;#8;
83                                     inaddr1=4'b0001;#8;
84
85 // end

```

ModelSim 仿真：



引脚分配：

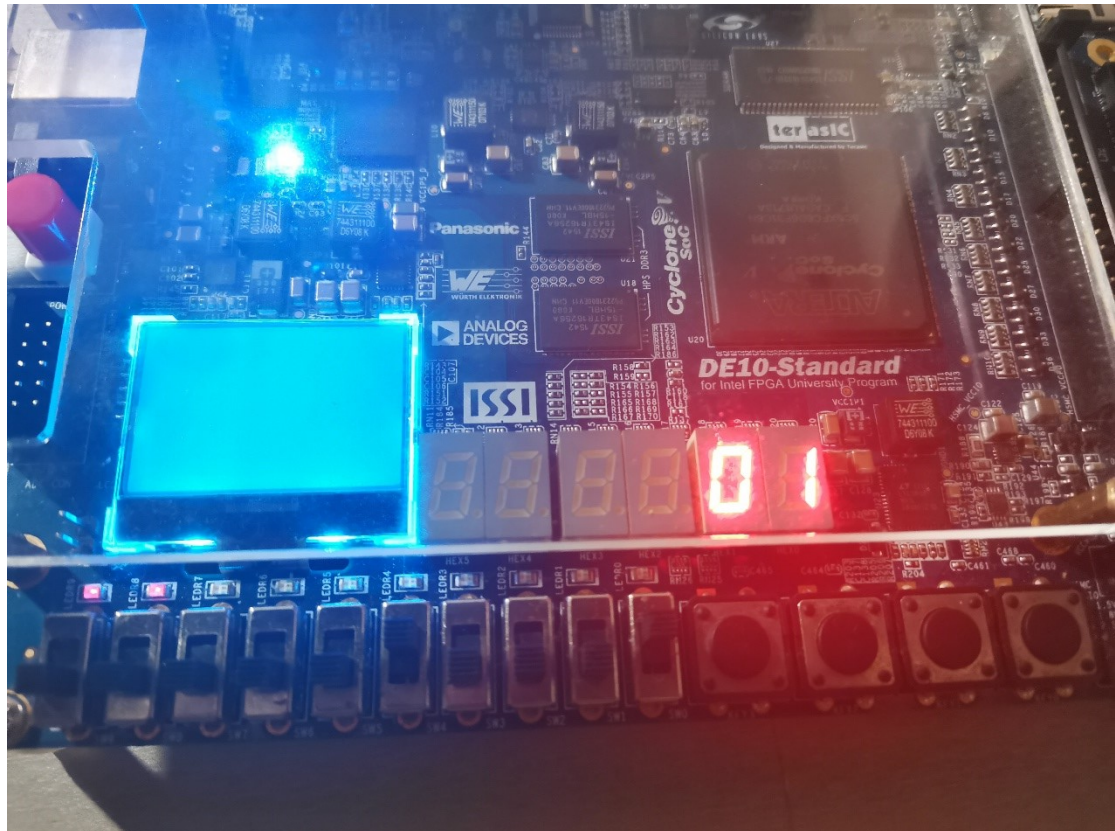
	Node Name	Direction	Location	I/O Bank
in	clk	Input	PIN_AJ4	3B
out	hex0[6]	Output	PIN_AH18	4A
out	hex0[5]	Output	PIN_AG18	4A
out	hex0[4]	Output	PIN_AH17	4A
out	hex0[3]	Output	PIN_AG16	4A
out	hex0[2]	Output	PIN_AG17	4A
out	hex0[1]	Output	PIN_V18	4A
out	hex0[0]	Output	PIN_W17	4A
out	hex1[6]	Output	PIN_V17	4A
out	hex1[5]	Output	PIN_AE17	4A
out	hex1[4]	Output	PIN_AE18	4A
out	hex1[3]	Output	PIN_AD17	4A
out	hex1[2]	Output	PIN_AE16	4A
out	hex1[1]	Output	PIN_V16	4A
out	hex1[0]	Output	PIN_AF16	4A
in	inaddr[3]	Input	PIN_AC30	5B
in	inaddr[2]	Input	PIN_AB28	5B

	Node Name	Direction	Location	I/O Bank
in	inaddr[1]	Input	PIN_Y27	5B
in	inaddr[0]	Input	PIN_AB30	5B
in	inaddr1[3]	Input	PIN_AD30	5B
in	inaddr1[2]	Input	PIN_AC28	5B
in	inaddr1[1]	Input	PIN_V25	5B
in	inaddr1[0]	Input	PIN_W25	5B
in	indata[7]	Input		
in	indata[6]	Input		
in	indata[5]	Input		
in	indata[4]	Input		
in	indata[3]	Input		
in	indata[2]	Input		
in	indata[1]	Input	PIN_AA30	5B
in	indata[0]	Input	PIN_AC29	5B
out	outdata2[7]	Output	PIN_AF24	4A
out	outdata2[6]	Output	PIN_AE24	4A
out	outdata2[5]	Output	PIN_AF25	4A

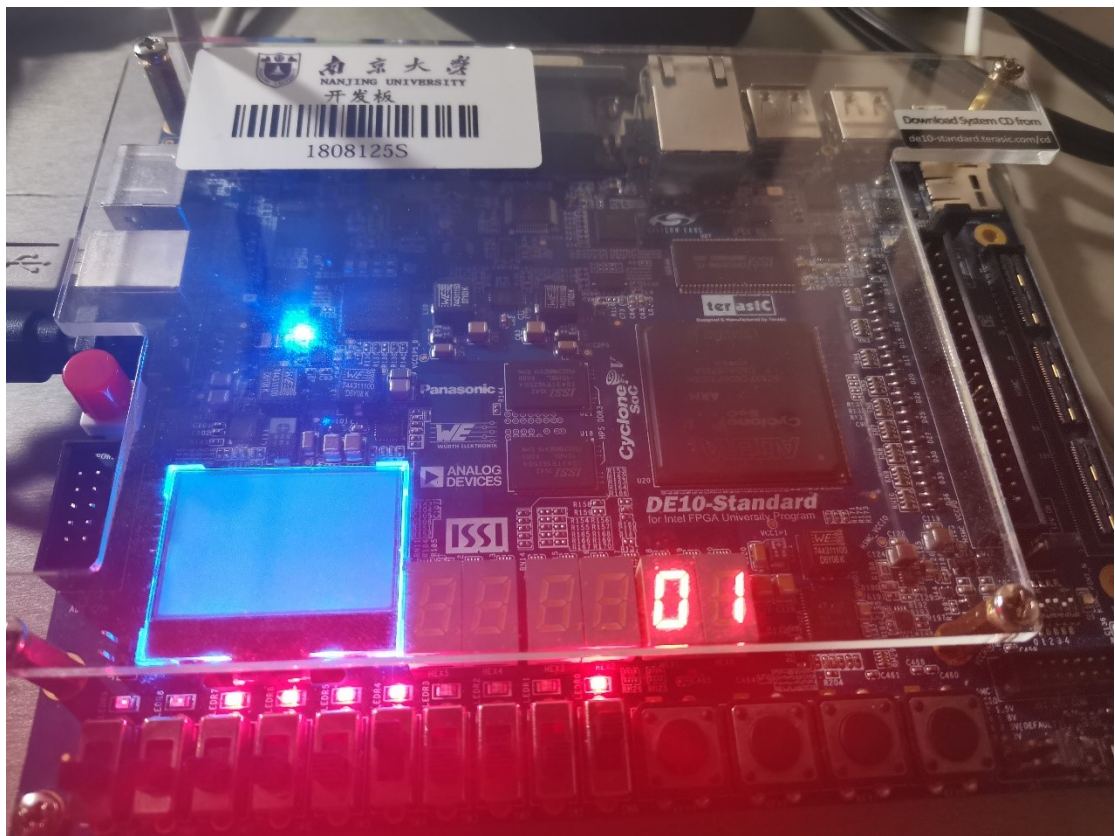
out	outdata2[4]	Output	PIN_AG25	4A
out	outdata2[3]	Output	PIN_AD24	4A
out	outdata2[2]	Output	PIN_AC23	4A
out	outdata2[1]	Output	PIN_AB23	5A
out	outdata2[0]	Output	PIN_AA24	5A
in	we	Input	PIN_AK4	3B
<<new node>>				

实验结果：

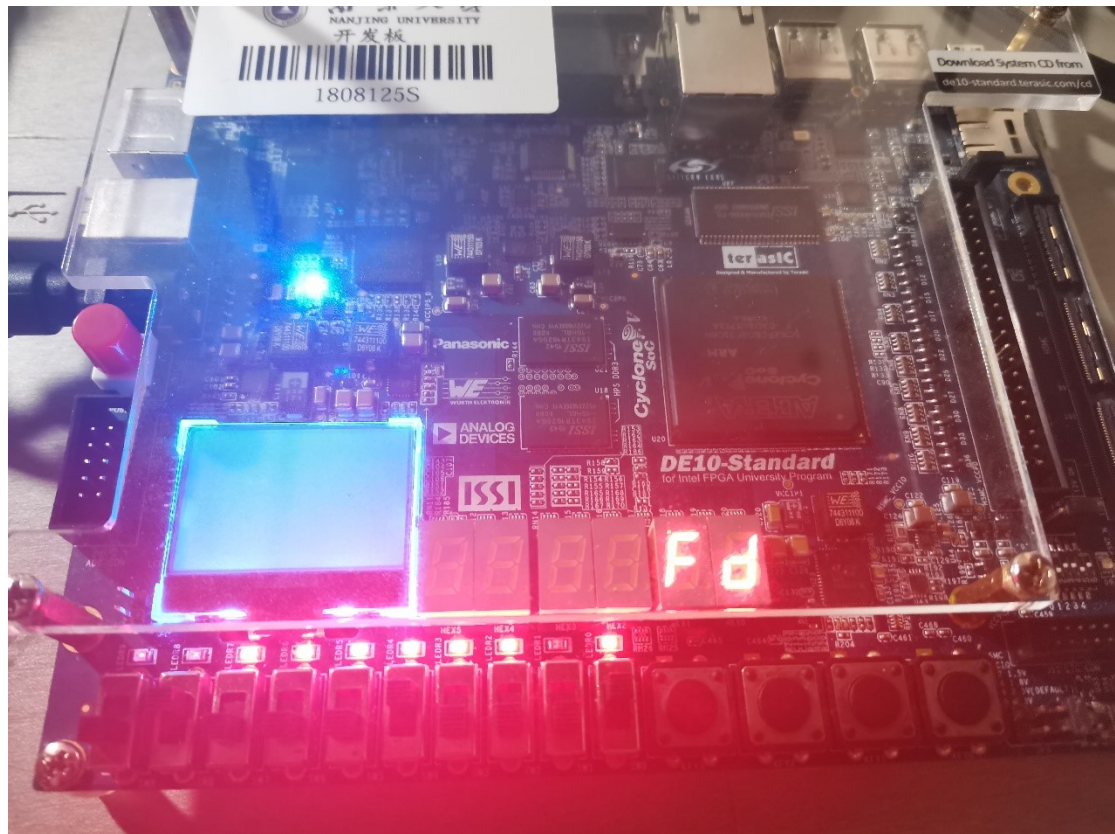
先读初始化的数据，地址都为1，一个时钟周期，ram1 显示 0x1:



两个时钟周期后，ram2port 显示 0xF1，因为 ram2port 里有锁存器：



然后写数据，两个存储器都只写入两位 01，再读，都显示 0xFD：



五、 实验中遇到的问题及解决办法

1、对 wire 和 reg 的使用：一开始习惯性的把模块间的参数用 reg 表示，但其实应该用 wire，因为 wire 相当于线。（??? 应该是这样的吧 2333...）

六、 启示

```
12
13     reg [RAM_WIDTH:0] ram [(2**RAM_ADDR_WIDTH)-1:0];
14
15     always @(posedge clk)
16         if (we)
17             ram[inaddr] <= din;
18
19     assign dout = ram[outaddr];
20
21 endmodule
```

📖 思考题

如果将表 7-2 中存储器实现部分改为

```
1 always @(posedge clk)
2     if (we)
3         ram[inaddr] <= din;
4     else
5         dout <= ram[outaddr];
```

该存储器的行为是否会发生变化?

修改之后，只有 clk 上升沿且 we=0 时，才会输出

七、 意见与建议

虽然之前我并没有上过数字电路这门课，但是实验手册前面的讲解非常的清楚，由浅入深，帮助我学习和完成了这次实验。