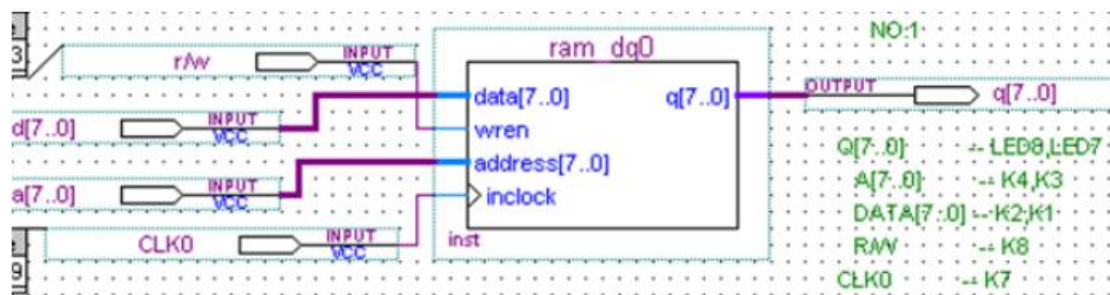


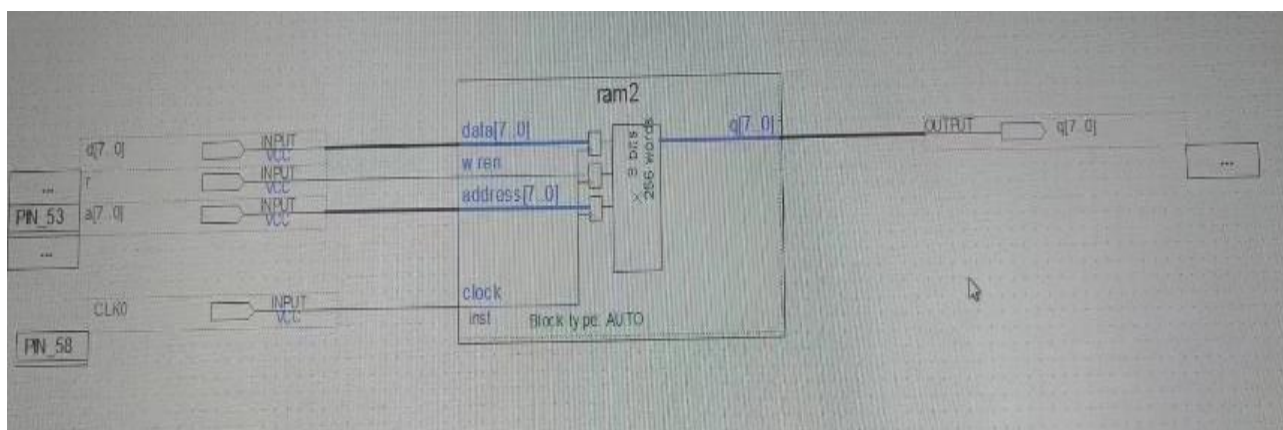
_____计算机组成与设计_____课程实验报告

学号：	姓名：	班级：
实验题目： RAM 扩展实验		
实验学时： 2	实验日期： 2023. 03. 28	
<p>实验目的：</p> <ol style="list-style-type: none"> 1、了解 FPGA 中 RAMlpm_ram_dq 的功能； 2、掌握 lpm_ram_dq 的参数设置和使用方法； 3、掌握 lpm_ram_dq 作为随机存储器 RAM 的仿真测试方法，工作特性和读写方法。 4、验证 FPGA 中 LPM_RAM 的功能。 		
<p>硬件环境：</p> <p>QuartusII 13.1</p> <p>Windows 10</p>		
<p>软件环境：</p> <ol style="list-style-type: none"> 1. 实验室台式机 2. 计算机组成与设计实验箱 		
<p>实验内容与设计：</p> <p>1、实验内容</p> <p>在 FPGA 中利用嵌入式阵列块 EAB 可以构成存储器，RAM 的结构如下图。</p> <p>数据从 ram_dp0 的左边 D[7..0] 输入，从右边 Q[7..0] 输出，R/W 为读/写控制信号端。</p> <p>当输入数据和地址准备好以后，在 inclock 是地址锁存时钟，当信号上升沿到来时，地址被锁存，数据写入存储单元。数据的读出控制是从 A[7..0] 输入存储单元地址，在 CLK 信号上升沿到来时，该单元数据从 Q[7..0] 输出。</p> <p>R/W 是读/写控制端，低电平时进行读操作，高电平时进行写操作；</p> <p>CLK 是读/写时钟脉冲信号；</p> <p>DATA[7..0] 是 RAM_dq0 的 8 位数据输入端；</p> <p>A[7..0] 是 RAM 的读出和写入地址；</p> <p>Q[7..0] 是 RAM_dq0 的 8 位数据输出端。</p>		



2、实验原理图

实验电路图：



3、实验步骤

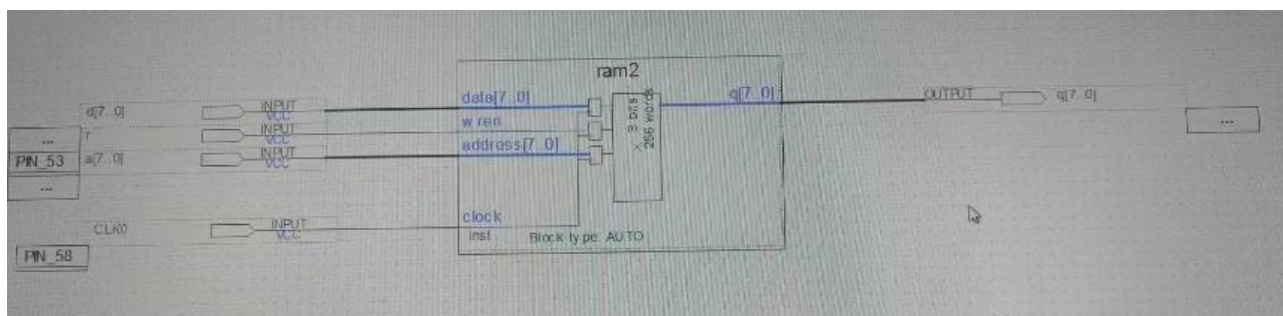
(1) 设计 RAM

按图 7-1 输入电路图，进行编译、引脚锁定、向 FPGA 配置下载，并往 RAM 中写入相关数据，往 RAM 写入数据可以在一开始就写入，也可以在后面使用相关硬件写入。

(2) 定制 RAM 元件

首先完成存放数据 ROM 的设计。利用 MegaWizard Plug-In Manager 定制程序存储器 RAM 宏功能块，并将以上的数据文件加载于此 RAM 中。

(3) 原理图输入：设计原理图如下：



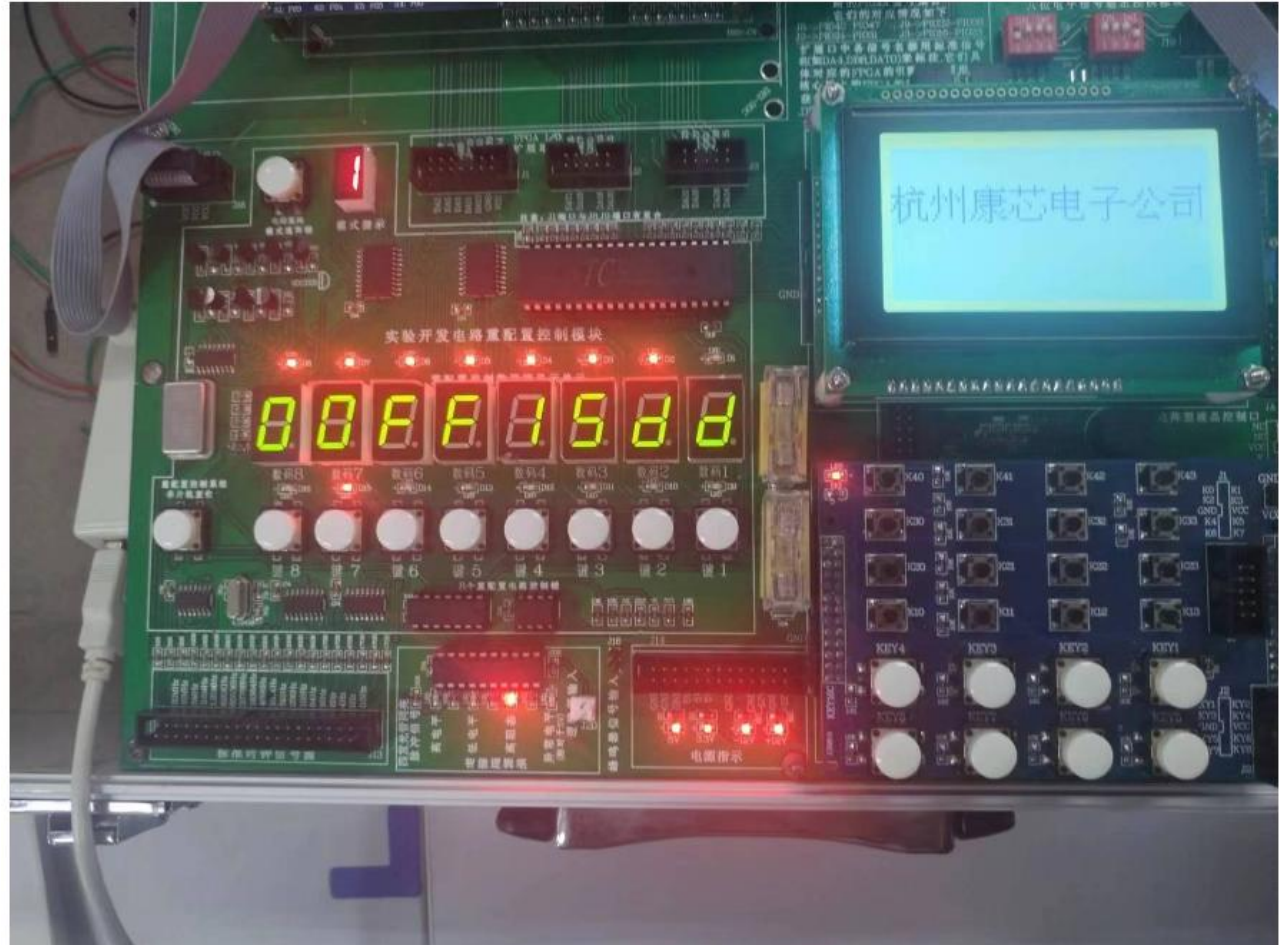
(4) 管脚锁定

如图点击，进行分配引脚工作，下载示例工程文件（图 7）至实验台上的 FPGA，选择实验台模式为 1，通过键 1、键 2 输入 RAM 的 8 位数据（选择实验台工作模式 1），键 3、键 4 输入存储器的 8 位地址。键 8 控制读/写允许，低电平时读允许，高电平时写允许；键 7（CLK0）产生读/写时钟脉冲，即生成写地址锁存脉冲，对 lpm_ram_dq 进行写/读操作。下为引脚分配示例图：

in_a[7]	Input	PIN_39	3	B3_NO	PIN_39	2.5 V (default)	8mA (default)		
in_a[6]	Input	PIN_42	3	B3_NO	PIN_42	2.5 V (default)	8mA (default)		
in_a[5]	Input	PIN_83	5	B5_NO	PIN_83	2.5 V (default)	8mA (default)		
in_a[4]	Input	PIN_77	5	B5_NO	PIN_77	2.5 V (default)	8mA (default)		
in_a[3]	Input	PIN_74	5	B5_NO	PIN_74	2.5 V (default)	8mA (default)		
in_a[2]	Input	PIN_70	4	B4_NO	PIN_70	2.5 V (default)	8mA (default)		
in_a[1]	Input	PIN_65	4	B4_NO	PIN_65	2.5 V (default)	8mA (default)		
in_a[0]	Input	PIN_60	4	B4_NO	PIN_60	2.5 V (default)	8mA (default)		
in_CLK0	Input	PIN_58	4	B4_NO	PIN_58	2.5 V (default)	8mA (default)		
in_d[7]	Input	PIN_84	5	B5_NO	PIN_84	2.5 V (default)	8mA (default)		
in_d[6]	Input	PIN_34	2	B2_NO	PIN_34	2.5 V (default)	8mA (default)		
in_d[5]	Input	PIN_75	5	B5_NO	PIN_75	2.5 V (default)	8mA (default)		
in_d[4]	Input	PIN_67	4	B4_NO	PIN_67	2.5 V (default)	8mA (default)		
in_d[3]	Input	PIN_66	4	B4_NO	PIN_66	2.5 V (default)	8mA (default)		
in_d[2]	Input	PIN_64	4	B4_NO	PIN_64	2.5 V (default)	8mA (default)		
in_d[1]	Input	PIN_55	4	B4_NO	PIN_55	2.5 V (default)	8mA (default)		
in_d[0]	Input	PIN_52	3	B3_NO	PIN_52	2.5 V (default)	8mA (default)		
out_q[7]	Output	PIN_80	5	B5_NO	PIN_80	2.5 V (default)	8mA (default)	2 (default)	
out_q[6]	Output	PIN_85	5	B5_NO	PIN_85	2.5 V (default)	8mA (default)	2 (default)	
out_q[5]	Output	PIN_73	5	B5_NO	PIN_73	2.5 V (default)	8mA (default)	2 (default)	
out_q[4]	Output	PIN_76	5	B5_NO	PIN_76	2.5 V (default)	8mA (default)	2 (default)	
out_q[3]	Output	PIN_71	4	B4_NO	PIN_71	2.5 V (default)	8mA (default)	2 (default)	
out_q[2]	Output	PIN_72	4	B4_NO	PIN_72	2.5 V (default)	8mA (default)	2 (default)	
out_q[1]	Output	PIN_68	4	B4_NO	PIN_68	2.5 V (default)	8mA (default)	2 (default)	
out_q[0]	Output	PIN_69	4	B4_NO	PIN_69	2.5 V (default)	8mA (default)	2 (default)	
in_r	Input	PIN_53	3	B3_NO	PIN_53	2.5 V (default)	8mA (default)		
<<new node>>									

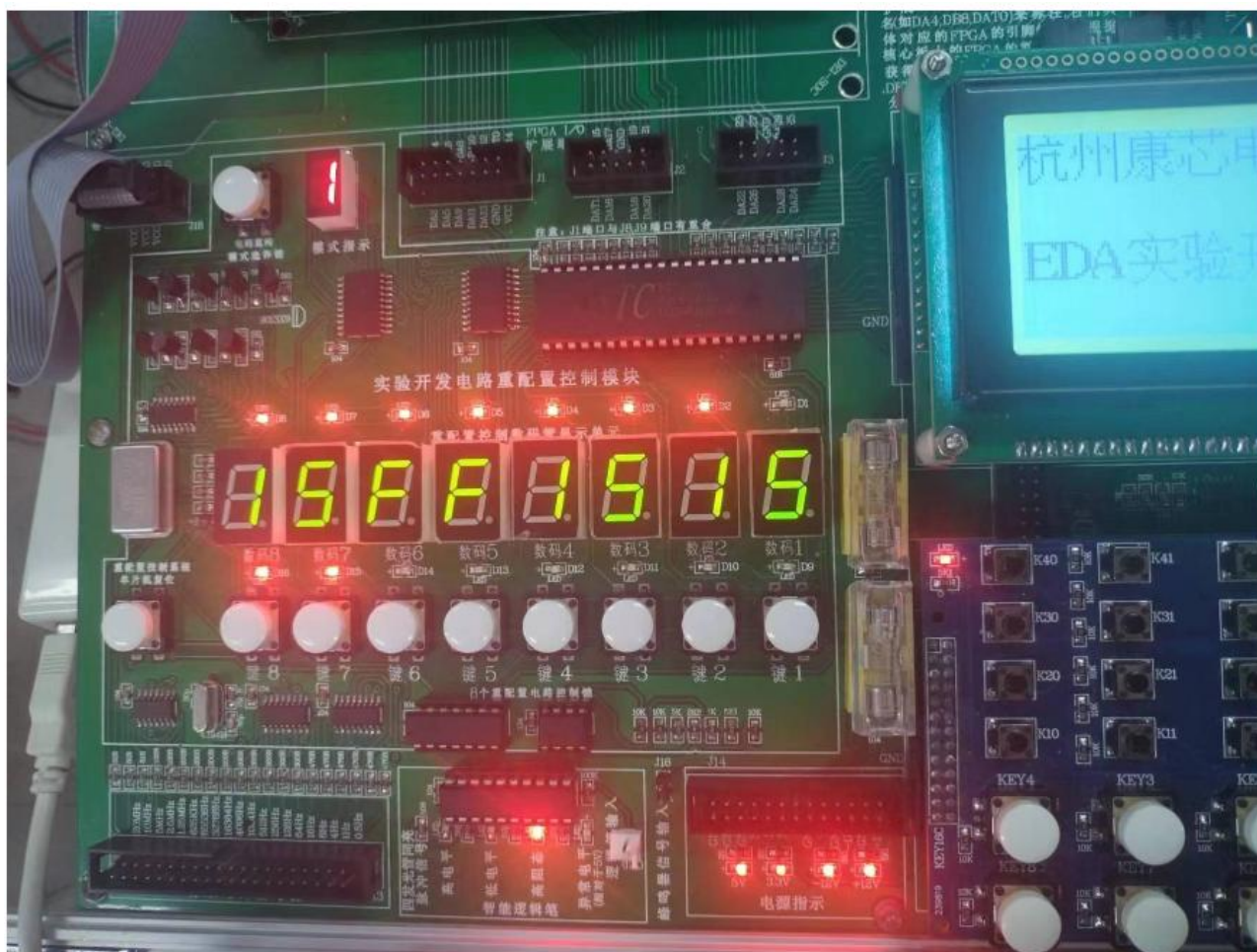
4、实验结果、

由于我们引脚设置，键 8 表示的意思是读写选择，即低电平是读操作，高电平是写操作，而键 7 则是时钟脉冲，键 3 和数键是地址位，数码 8 和数码 7 则是相应的输出，一开始时我们选择地址为 15（十六进制数），由于此时还未写入数据，所以数码 8 和数码 7 显示 0。

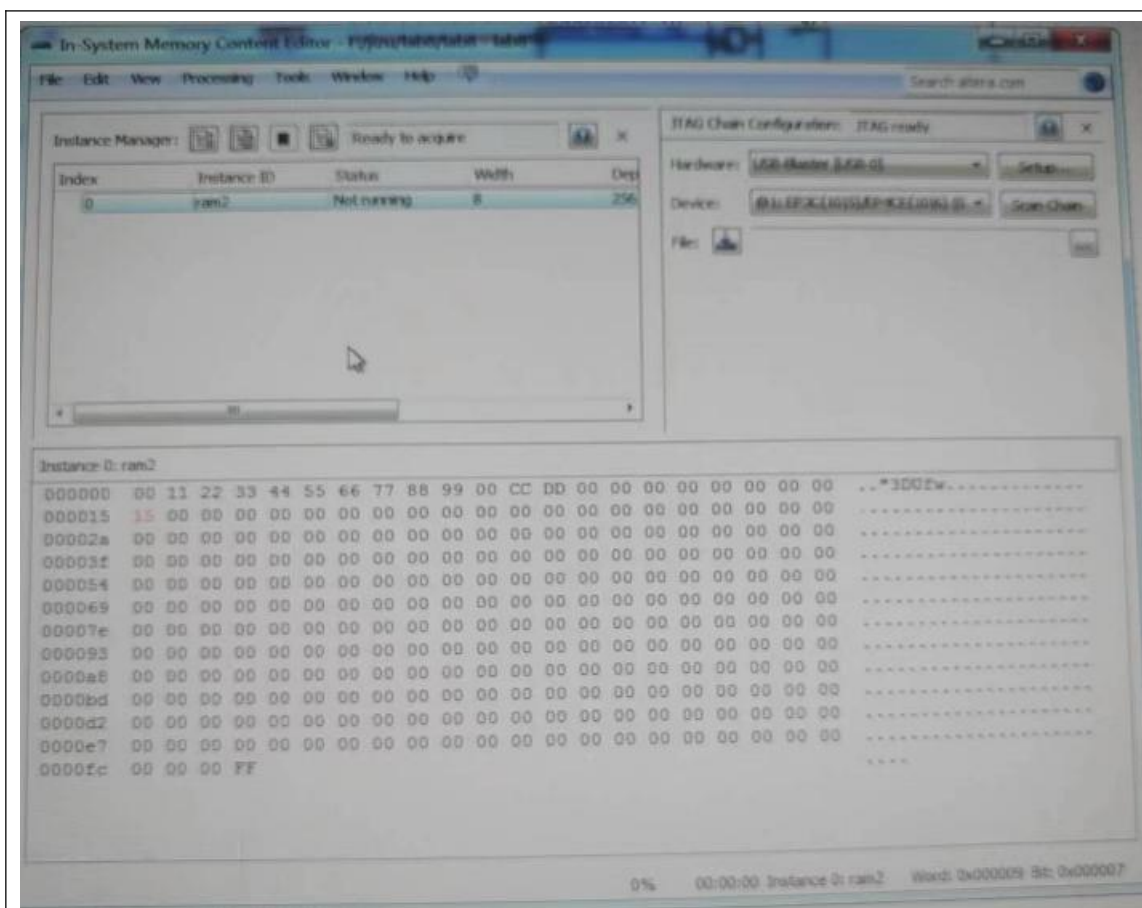


之后我们往地址为 15（十六进制数）的内存单元写入数据，此时需将键 8 置为高电平，同时

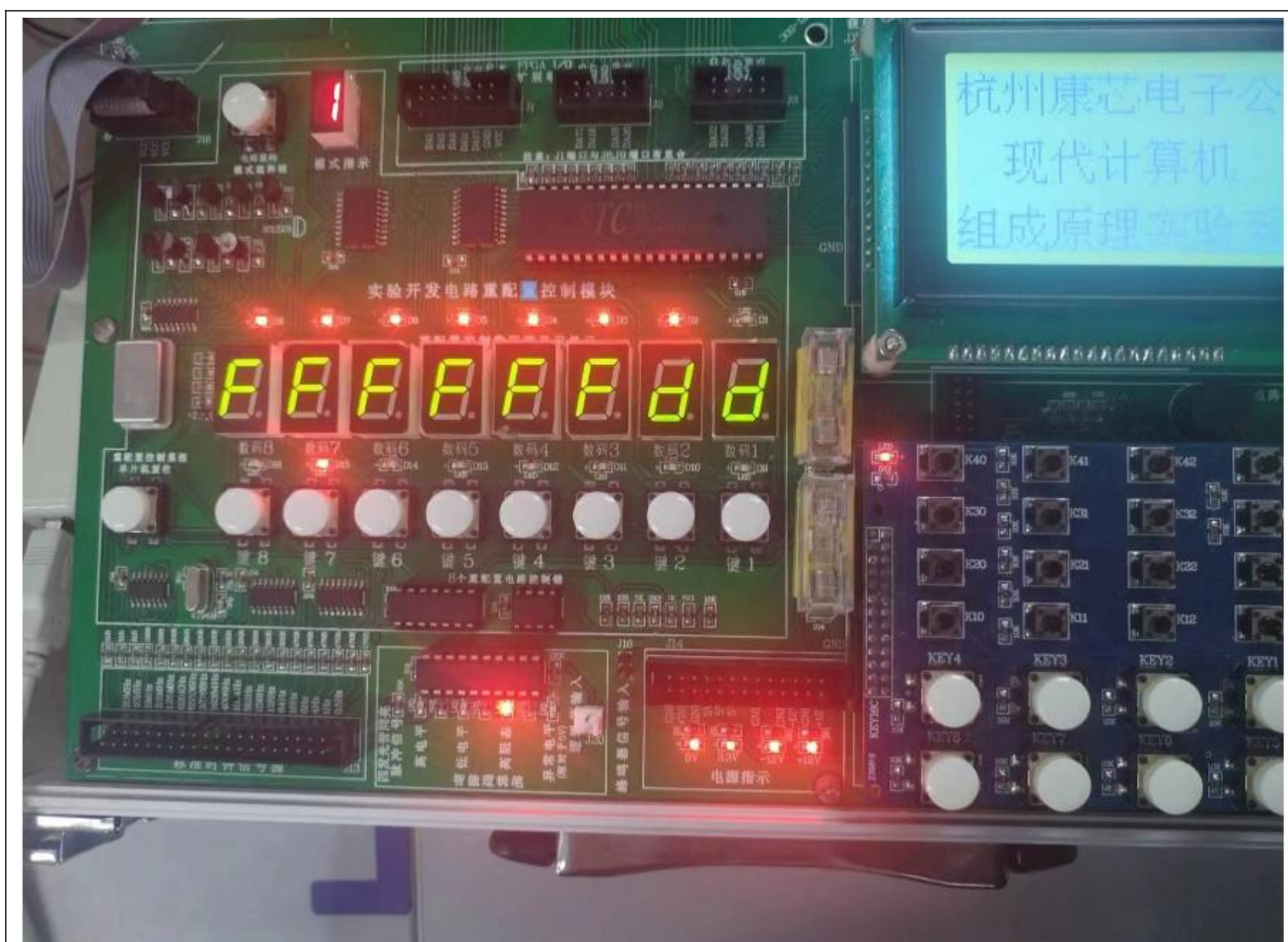
我们往地址为 15（十六进制数）的内存单元写入 15（十六进制数）。



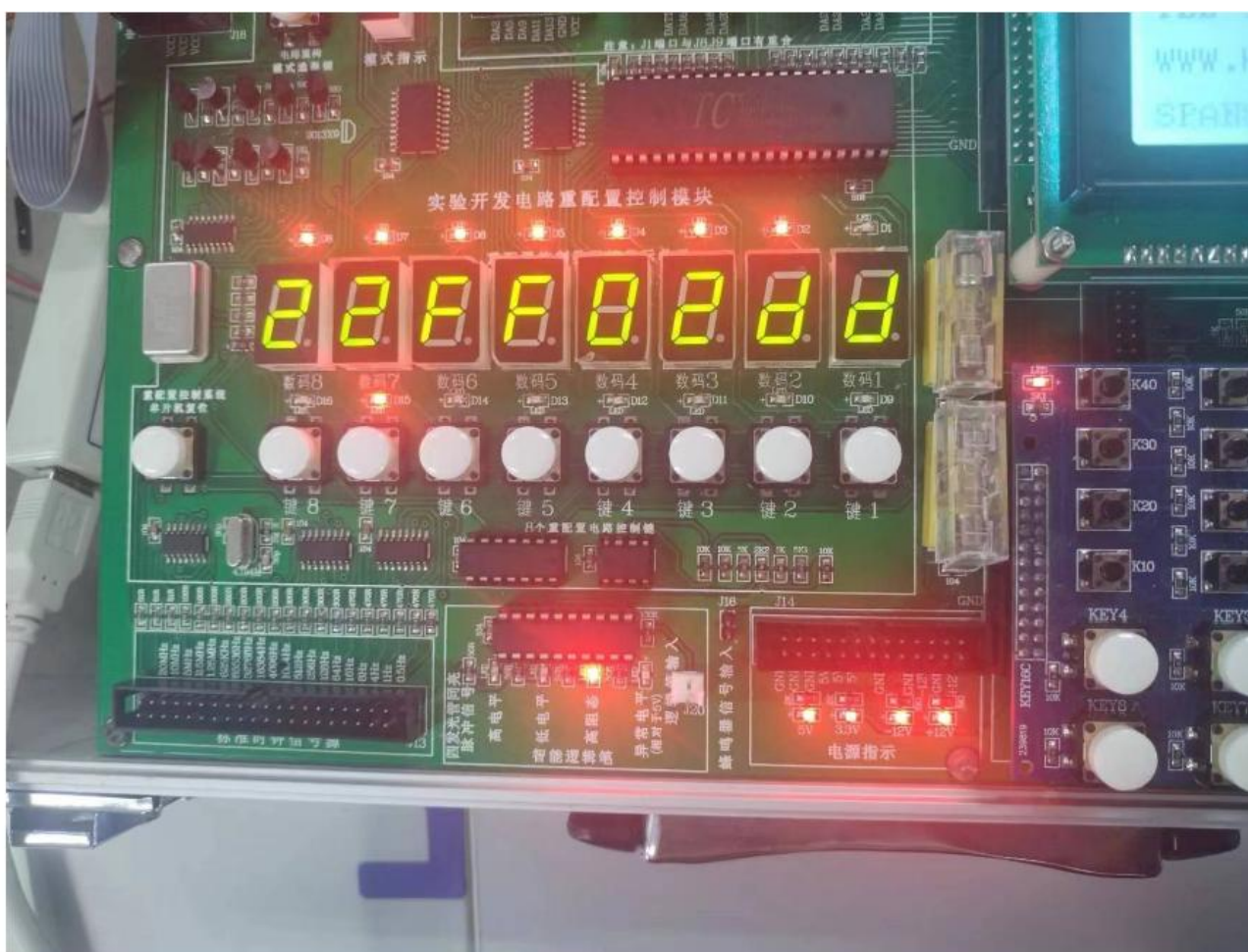
此时 RAM 中数据为：



之后我们选择读操作，将键 8 置为低电平，同时选择地址为 FF（十六进制数）的内存单元，由上图可知，地址为 FF（十六进制数）的内存单元中存储的数据为 FF（十六进制数），而数码 8 和数码 7 显示也为 FF（十六进制数），测试正确。



最后我们选择读操作，并选择地址为 02（十六进制数）的内存单元，由上图可见，地址为 02（十六进制数）的内存单元中所存储的数据为 22（十六进制数），而我们的数码 8 和数码 7 显示的数据也为 22，测试正确。



结论分析与体会：

通过这次实验，我体会到了 RAM 作为随机存取存储器的特点，可以随时读写，而且速度很快，相比于 ROM 多了写数据的功能。我们可以在一开始就往 RAM 中写入数据，也可以后面使用相关硬件来写入，同时我们可以使用用 tool 中的 In-system Memory Content Editor 对 ROM/RAM 内容进行可视化编辑修改，同时检查写入是否正确。