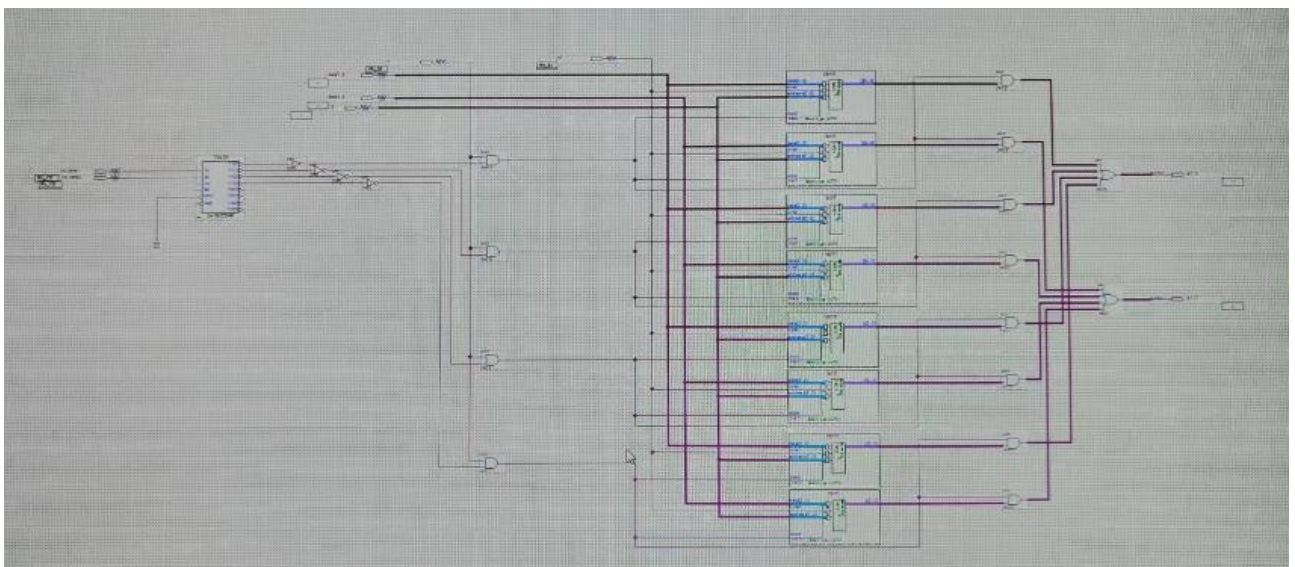


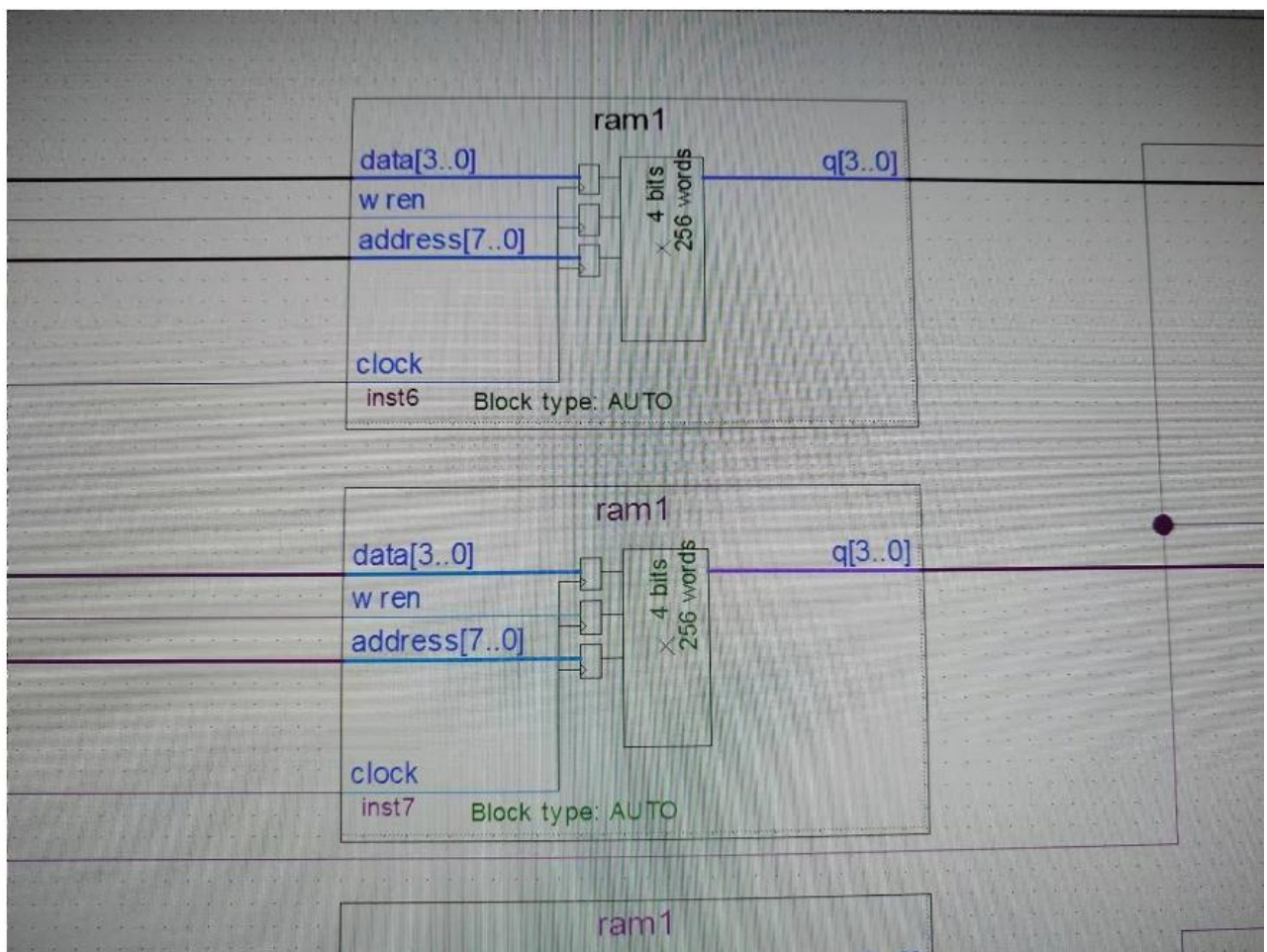
学号：	姓名：	班级：
实验题目：实验九 RAM 扩展实验		
实验学时： 2	实验日期：2023. 04. 04	
<p>实验目的：</p> <ol style="list-style-type: none"> 1. 了解半导体静态随机读写存储器 RAM 的工作原理及其使用方法。 2. 掌握半导体存储器的字、位扩展技术。 		
<p>硬件环境：</p> <ol style="list-style-type: none"> 1. 实验室台式机 2. 计算机组成与设计实验箱 		
<p>软件环境：</p> <p>QuartusII 13.0</p>		
<p>实验内容与设计：</p> <p>1、实验内容</p> <p>实验要求：</p> <ol style="list-style-type: none"> 1. 采用实验 8 所设计的 256 x 4 的 RAM 的结构（参考实验 8 生成器件），构成 1Kx8 的存储器。根据课本第 4 章的内容自行设计实现方案。 2. 实验 8 中，因为 RAM 的数据输入和数据输出是不同的端口，设计时不用隔离器件。要利用 2-4 译码器 74139. 3. 选择五个不连续的存储单元地址，分别存入不同内容，作单个存储器单元的读/写操作实验。 <p>首先我们要设计一个 1K*8 的 RAM 存储器，则根据实验要求我们需要使用到一个 2-4 译码器 74139，和 8 个 256*4 的 RAM 存储器。对于 RAM 元件来说，需要有地址线输入单元地址，数据线输入数据，在本实验中，我们先用两个 RAM 进行位拓展，构成一个 256*8 的 RAM 存储器，接下来在进行字拓展，即用四个 256*8 的 RAM 存储器来构成一个 1K*8 的 RAM 存储器，我们使用 2-4 译码器的输出来做片选信号。</p> <p>2、实验原理图</p>		



3、实验步骤

(1) 设计 RAM

首先我们需要设计一个 256*4 的 RAM 存储器，并且我们在初始时并不会往 RAM 中写入数据，而是在过程中写入，RAM 示意图：



(2) 原理图输入

我们先做 RAM 位拓展，即先由两个 256*4 的 RAM 存储器构成一个 256*8 的 RAM 存储器，再有

四个 256*8 的 RAM 存储器构成一个 1K*8 的 RAM 存储器,同时我们还需要 and2 和 or4 以及 2-4 译码器 74139, 按照上图即可完成原理图的输入。

(3) 管脚锁定

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 输入存储器的低 8 位地址，高 2 位地址由红色的拨码开关提供。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 RAM 进行写/读操作；下为管脚图：

address[7]	Input	PIN_39	3	B3_NO	PIN_39	2.5 V (default)	8mA (default)
address[6]	Input	PIN_42	3	B3_NO	PIN_42	2.5 V (default)	8mA (default)
address[5]	Input	PIN_83	5	B5_NO	PIN_83	2.5 V (default)	8mA (default)
address[4]	Input	PIN_77	5	B5_NO	PIN_77	2.5 V (default)	8mA (default)
address[3]	Input	PIN_74	5	B5_NO	PIN_74	2.5 V (default)	8mA (default)
address[2]	Input	PIN_70	4	B4_NO	PIN_70	2.5 V (default)	8mA (default)
address[1]	Input	PIN_65	4	B4_NO	PIN_65	2.5 V (default)	8mA (default)
address[0]	Input	PIN_60	4	B4_NO	PIN_60	2.5 V (default)	8mA (default)
c	Input	PIN_58	4	B4_NO	PIN_58	2.5 V (default)	8mA (default)
data[7]	Input	PIN_84	5	B5_NO	PIN_84	2.5 V (default)	8mA (default)
data[6]	Input	PIN_34	2	B2_NO	PIN_34	2.5 V (default)	8mA (default)
data[5]	Input	PIN_75	5	B5_NO	PIN_75	2.5 V (default)	8mA (default)
data[4]	Input	PIN_67	4	B4_NO	PIN_67	2.5 V (default)	8mA (default)
data[3]	Input	PIN_66	4	B4_NO	PIN_66	2.5 V (default)	8mA (default)
data[2]	Input	PIN_64	4	B4_NO	PIN_64	2.5 V (default)	8mA (default)
data[1]	Input	PIN_55	4	B4_NO	PIN_55	2.5 V (default)	8mA (default)
data[0]	Input	PIN_52	3	B3_NO	PIN_52	2.5 V (default)	8mA (default)
pin_name1	Input	PIN_133	8	B8_NO	PIN_133	2.5 V (default)	8mA (default)
pin_name2	Input	PIN_129	8	B8_NO	PIN_129	2.5 V (default)	8mA (default)
q[7]	Output	PIN_80	5	B5_NO	PIN_80	2.5 V (default)	8mA (default) 2 (default)
q[6]	Output	PIN_85	5	B5_NO	PIN_85	2.5 V (default)	8mA (default) 2 (default)
q[5]	Output	PIN_73	5	B5_NO	PIN_73	2.5 V (default)	8mA (default) 2 (default)
q[4]	Output	PIN_76	5	B5_NO	PIN_76	2.5 V (default)	8mA (default) 2 (default)
q[3]	Output	PIN_71	4	B4_NO	PIN_71	2.5 V (default)	8mA (default) 2 (default)
q[2]	Output	PIN_72	4	B4_NO	PIN_72	2.5 V (default)	8mA (default) 2 (default)
q[1]	Output	PIN_68	4	B4_NO	PIN_68	2.5 V (default)	8mA (default) 2 (default)
q[0]	Output	PIN_69	4	B4_NO	PIN_69	2.5 V (default)	8mA (default) 2 (default)
wr	Input	PIN_53	3	B3_NO	PIN_53	2.5 V (default)	8mA (default)

4、实验结果

如下图所示，是我们在实验过程中 8 个 RAM 存储器中存储的数据

Index	Instance ID	Status	Width	Depth	Type	Mode
0	ram2	Not running	4	256	RAM/ROM	Read/Write
1	ram2	Not running	4	256	RAM/ROM	Read/Write
2	ram2	Not running	4	256	RAM/ROM	Read/Write
3	ram2	Not running	4	256	RAM/ROM	Read/Write
4	ram2	Not running	4	256	RAM/ROM	Read/Write
5	ram2	Not running	4	256	RAM/ROM	Read/Write
6	ram2	Not running	4	256	RAM/ROM	Read/Write
7	ram2	Not running	4	256	RAM/ROM	Read/Write

Instance 0: ram2															
000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Instance 1: ram2															
000000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Instance 2: ram2															
000000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Instance 3: ram2															
000000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

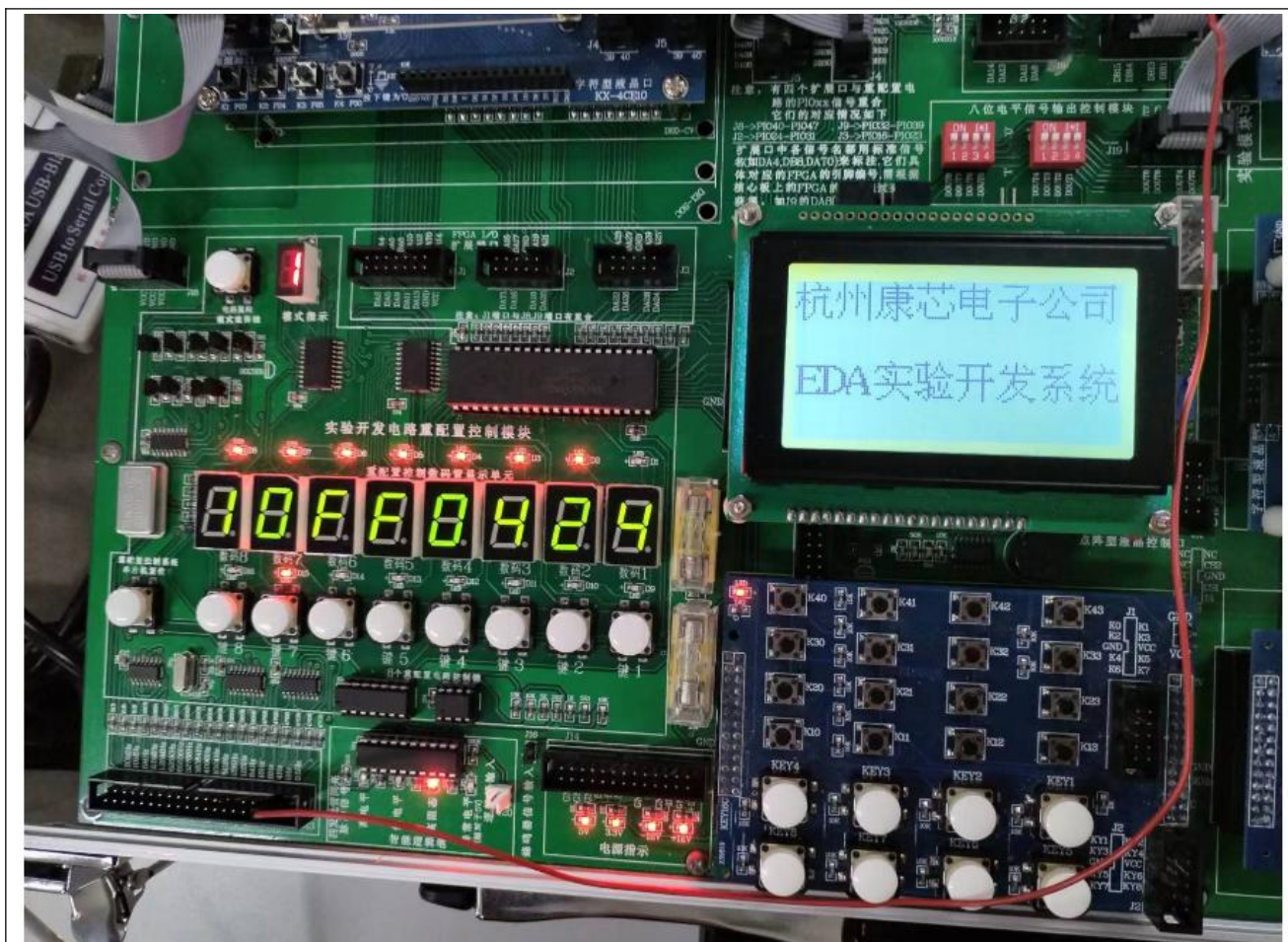
Instance 4: ram2															
000000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Instance 5: ram2															
000000	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

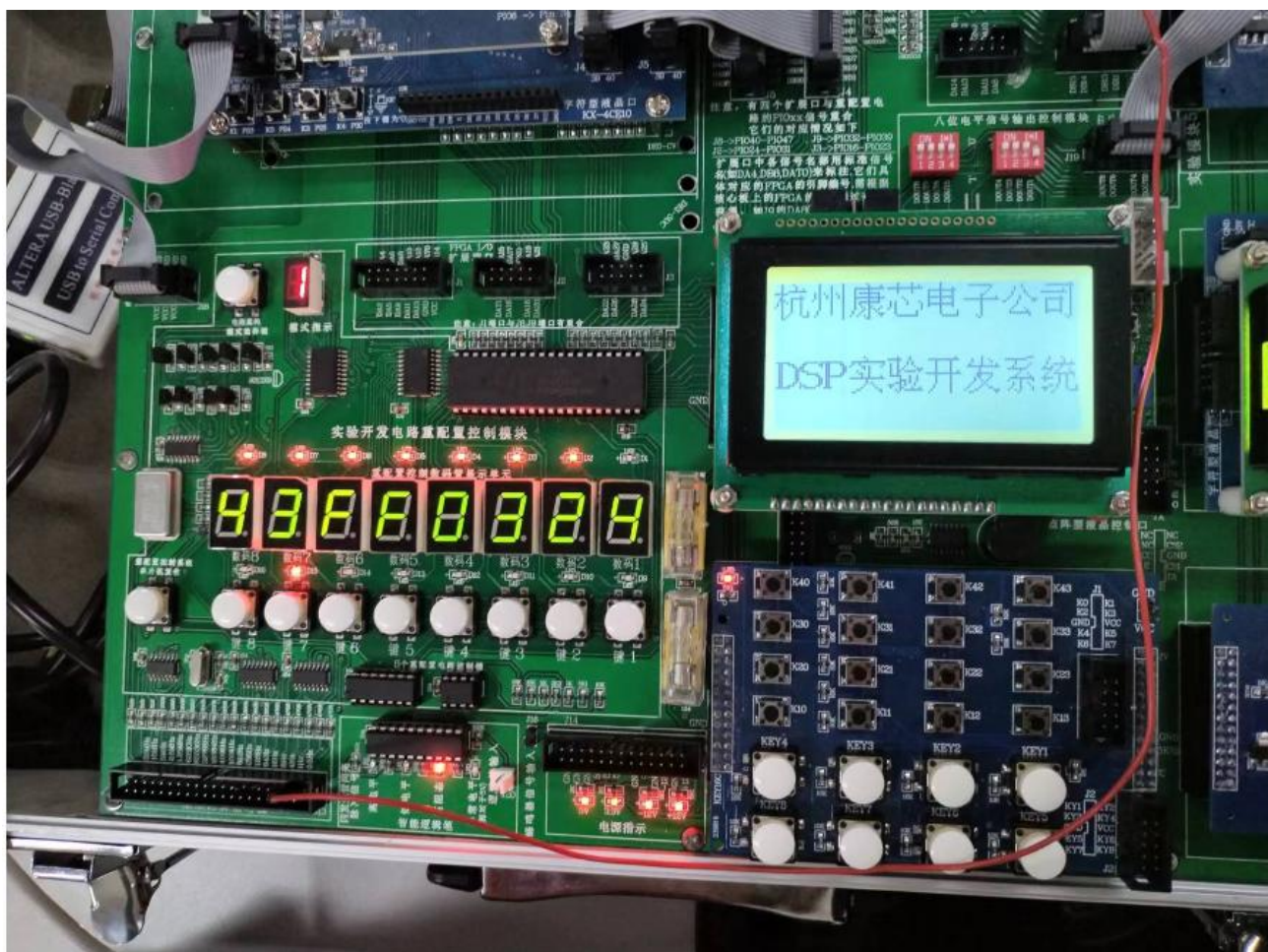
Instance 6: ram2															
000000	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Instance 7: ram2															
000000	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
000072	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000e4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

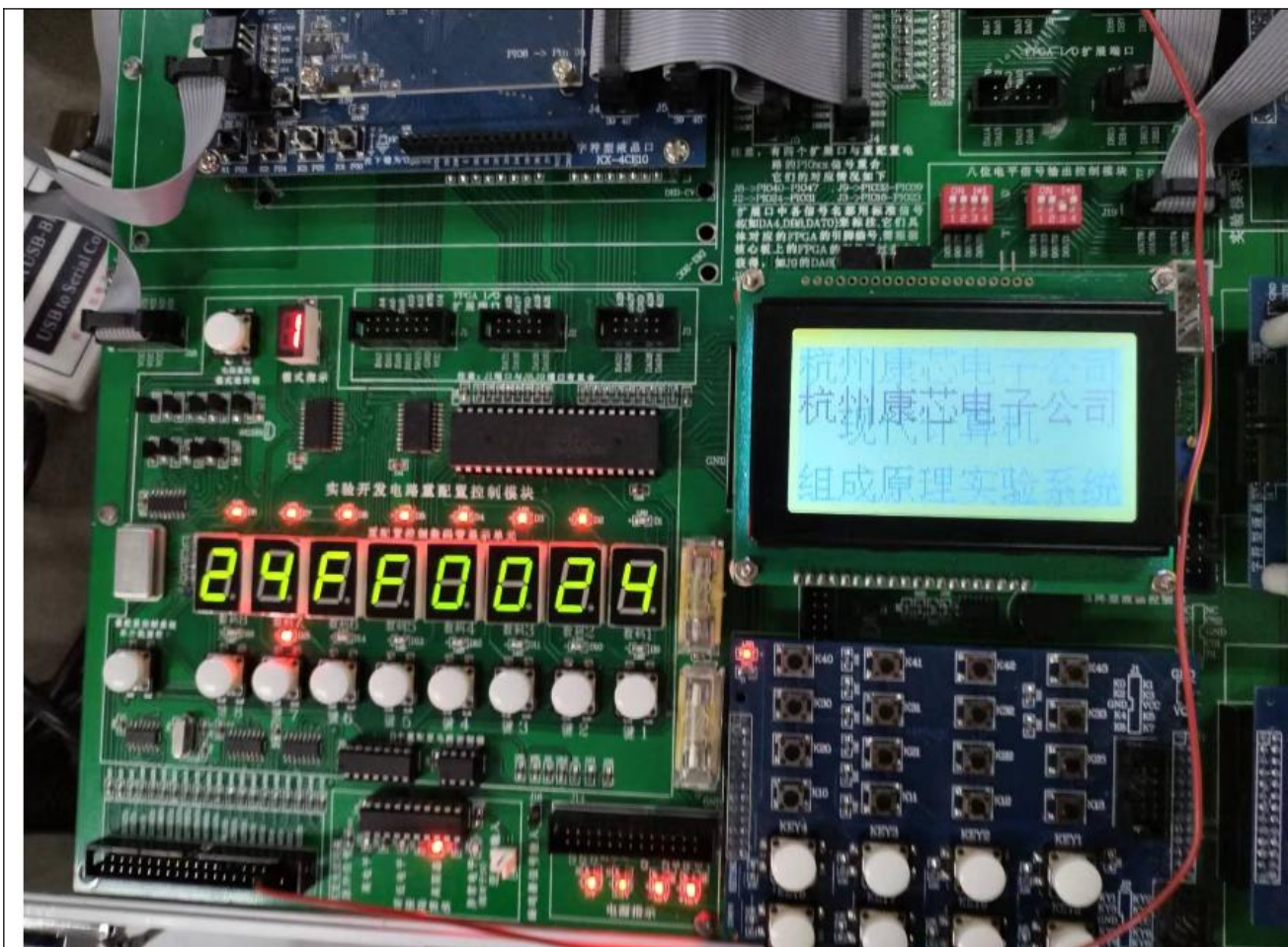
其中 RAM4 和 RAM0 是一组，RAM5 和 RAM1 是一组，RAM6 和 RAM2 是一组，RAM7 和 RAM3 是一组，并且 RAM4、RAM5、RAM6、RAM7 中数据是高位，RAM0、RAM1、RAM2、RAM3 中数据是低位。我们先选中 RAM4 和 RAM0 这一组，输入地址 04（十六进制数），得到的数据是 10（十六进制数），经过对比测试正确。



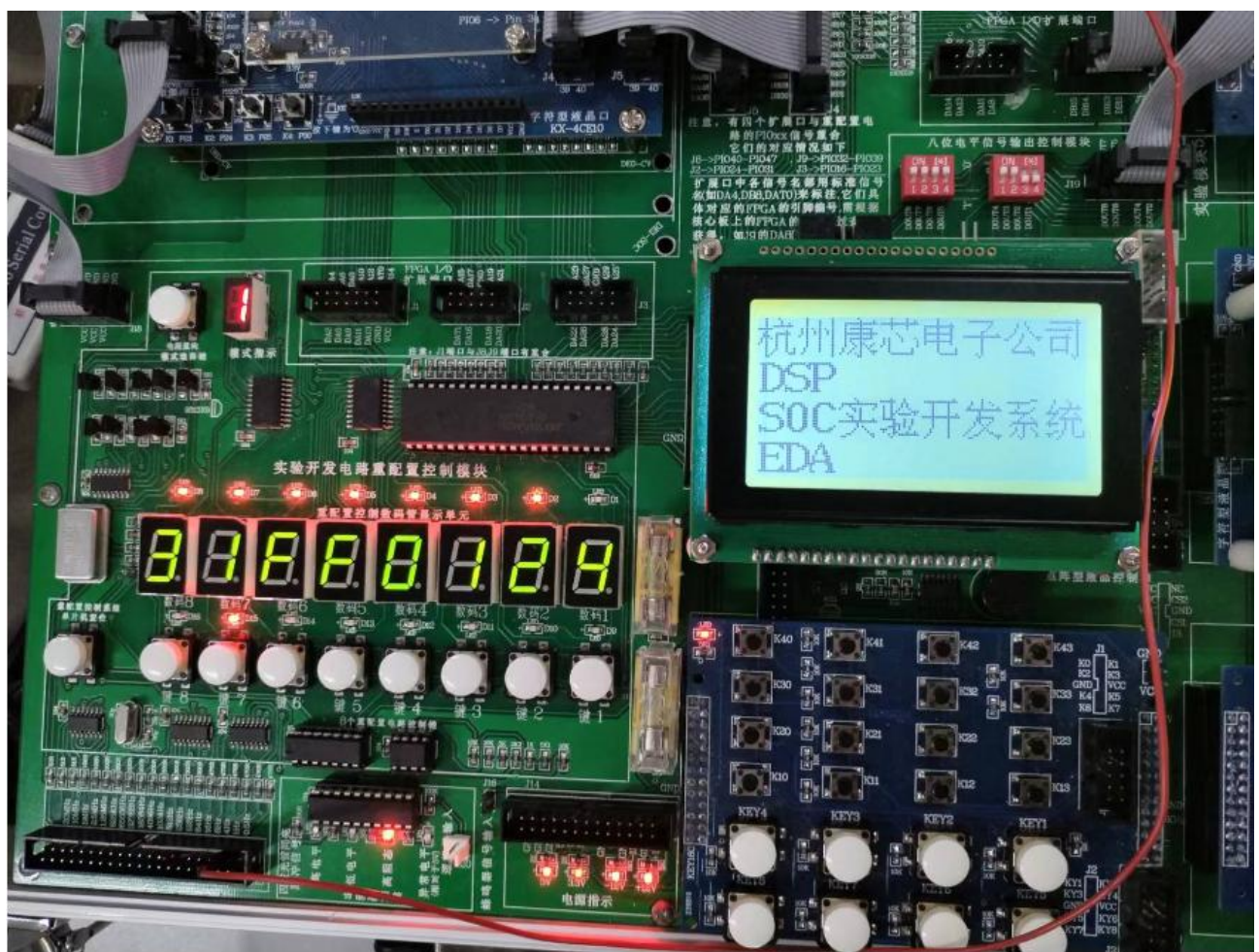
接下来我们选中 RAM7 和 RAM3 这一组，并且输入地址是 03（十六进制数），得到的数据是 43（十六进制数），经过对比测试正确。



接下来我们选中 RAM5 和 RAM1 这一组，并且输入地址是 00（十六进制数），得到的数据是 24（十六进制数），经过对比测试正确。



最后我们选中 RAM6 和 RAM2 这一组，并且输入地址是 01（十六进制数），得到的数据是 31（十六进制数），经过对比测试正确。



结论分析与体会：

通过本次实验，了解到了如何进行字拓展和位拓展，同时对片选信号有了进一步的认知。对于位拓展来说，使用同种输入控制信号，同时输入两个 RAM，其中输入的数据，一部分为高位，一部分为低位，对应的输出亦然，就实现了存储字长的扩展。对于字扩展，可以使用部分高位地址作为译码信号输入到译码器，从而生成对两个 RAM 的片选信号，但在本实验中，我们采用的是 2-4 译码器 74139。