一、实验目的

1. 深入理解基本模型计算机的功能、组成知识；
2. 深入学习计算机各类典型指令的执行流程；
3. 学习微程序控制器的设计过程和相关技术，掌握 LPM_ROM 的配置方法。
4. 在掌握部件单元电路实验的基础上，进一步将单元电路组成系统，构造一台基本模型计算机。
5. 定义五条机器指令，并编写相应的微程序，上机调试，掌握计算机整机概念。掌握微程序的设计方法，学会编写二进制微指令代码表。
6. 通过熟悉较完整的计算机的设计，全面了解并掌握微程序控制方式计算机的设计方法。

二、实验环境

- PC 计算机
- Quartus II 开发软件
- 计算机组成原理试验箱

三、实验原理

1．在部件实验过程中，各部件单元的控制信号是人为模拟产生的，而本实验将能在微过程控制下自动产生各部件单元控制信号，实现特定的功能。实验中，计算机数据通路的控制将由微过程控制器来完成，CPU 从内存中取出一条机器指令到指令执行结束的一个指令周期，全部由微指令组成的序列来完成，即一条机器指令对应一个微程序。

2．指令格式

（1）指令格式

采用寄存器直接寻址方式，其格式如下：

位	7654	32	10
功能	OP-CODE	rs	rd

本实验采用五条机器指令：IN（输入）、ADD（二进制加法）、STA（存数）、OUT（输出）、JMP（无条件转移），其指令格式如下（最高 4 位二进制数为操作码）：

其中，OP-CODE 为操作码，rs 为源寄存器，rd 为目的寄存器，并规定：

Rs 或 rd	选定的寄存器
00	R0
01	R1
10	R2

助记符	机器指令码	Addr 地址码	功能说明
IN	0 0H		“INPUT”中的数据→R0
ADD addr	1 0H	XX H	R0+[addr] ->R0
STA addr	2 0H	XX H	R0 -> [addr]
OUT addr	3 0H	XX H	[addr] -> BUS
JMP addr	4 0H	XX H	addr →PC

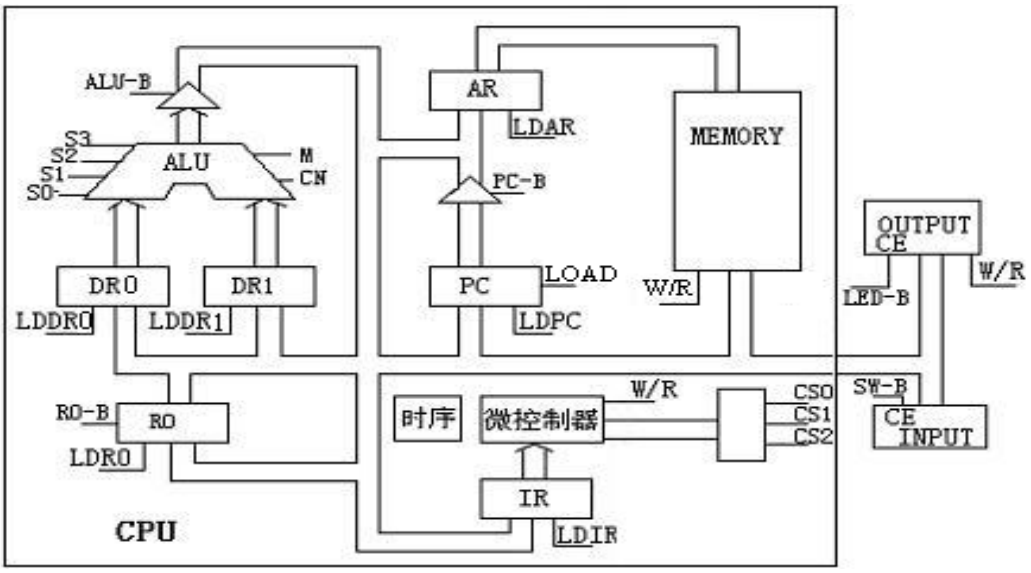


图 1-1 数据通路框图

注：基本模型机无 R1,R2 寄存器，但实际电路中有 R1,R2 寄存器。

其中 IN 为单字长（8 位二进制），其余为双字长指令，XX H 为 addr 对应的十六进制地址码。为了向 RAM 中装入程序和数据，检查写入是否正确，并能启动程序执行，还必须设计三个控制台操作微程序。

1、存储器读操作（KRD）：下载实验程序后按总清除按键（CLR）后，控制台 SWA、

SWB 为“0 0”时，可对 RAM 连续手动读出操作。

2、存储器写操作（KWE）：下载实验程序后按总清除按键（CLR）后，控制台

SWA、SWB 为“0 1”时，可对 RAM 连续手动写操作。

1. 启动程序（RP）：下载实验程序后按总清除按键（CLR）后，控制台 SWA、SWB 为“1 1”时，即可转入到微地址“01”号“取指令”微指令，启动程序运行。

根据以上要求设计数据通路框图，如图 1-1 所示。

SWB	SWA	控制台指令
0	0	读内存（KRD）
0	1	写内存（KWE）
1	1	启动程序（RP）

表 1-1 24 位微代码定义：

24	23	22	21	20	19	18	17	16	15 14 13	12 11 10	987	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	uA5	uA4	uA3	uA2	uA1	uA0

表 1-2 A、B、C 各字段功能说明：

A 字段				B 字段				C 字段			
15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	0		0	0	0		0	0	0	
0	0	1	LDRi	0	0	1	RS-B	0	0	1	P（1）
0	1	0	LDDR1	0	1	0		0	1	0	
0	1	1	LDDR2	0	1	1		0	1	1	
1	0	0	LDIR	1	0	0		1	0	0	P（4）
1	0	1	LOAD	1	0	1	ALU-B	1	0	1	LDAR
1	1	0	LDAR	1	1	0	PC-B	1	1	0	LDPC

24 位微代码中各信号的功能

(1) uA5—uA0：微程序控制器的微地址输出信号，是下一条要执行的微指令的微地址。

(2) S3、S2、Sl、S0：由微程序控制器输出的 ALU 操作选择信号，以控制执行 16 种算术操作或 16 种逻辑操作中的某一种操作。

(3) M：微程序控制输出的 ALU 操作方式选择信号端。M＝0 执行算术操作；M＝l 执行逻辑操作。

(4) /Cn：微程序控制器输出的进位信号，/Cn＝0 表示 ALU 运算时最低位有进位，/Cn＝1 则表示无进位。

(5) WE：微程序控制器输出的 RAM 控制信号。当/CE＝0 时，如 WE＝0 为存储器读；如 WE＝1 为存储器写。

(6) A9、A8——译码后产生 CS0、CS1、CS2 信号，分别作为 SW_B、RAM、LED 的选通控制信号。

(7) A 字段（15、14、13）——译码后产生与总线相连接的各单元的输入选通信号（见表 6-1）。

(8) B 字段（12、11、10）——译码后产生与总线相连接的各单元的输出选通信号。

(9) C 字段（9、8、7） ——译码后产生分支判断测试信号 P(1)~P(4)和 LDPC 信号。

系统涉及到的微程序流程见图 1-2。当执行“取指令”微指令时，该微指令的判断测试字段为 P(1)测试。由于“取指令”微指令是所有微程序都使用的公用微指令，因此 P(1)的测试结果出现多路分支（见图 1-2 左图）。用指令寄存器的高 4 位（IR7-IR4）作为测试条件，出现 5 路分支，占用 5 个固定地址单元。

控制台操作为 P(4)测试（见图 1-2 右图），它以控制台信号 SWB、SWA 作为测试条件，出现了 3 路分支，占用 3 个固定微地址单元。当分支微地址单元固定后，剩下的其它地方就可以一条微指令占用控制存储器的一个微地址单元。

当全部微程序设计完毕后，应将每条微指令代码化、

指令寄存器（IR）：指令寄存器用来保存当前正在执行的一条指令。当执行一条指令时，先把它从内存取到缓冲寄存器中，然后再传送至指令寄存器。指令划分为操作码和地址码段，由二进制数构成，为了执行任何给定的指令，必须对操作码进行测试“P(1)”，通过节拍脉冲 T4 的控制，以便识别所要求的操作。

指令译码器: 根据指令中的操作码强置微控制器单元的微地址，使下一条微指令指向相应的微程序首地址。

实验中 LCD 液晶显示屏可以用来显示模型机 CPU 中各组成单元的内容。将 B100_C.sof 文件下载到实验台后，按系统复位键，LCD 液晶显示屏即显示 CPU 中各组成单元的内容。其功能说明如下：

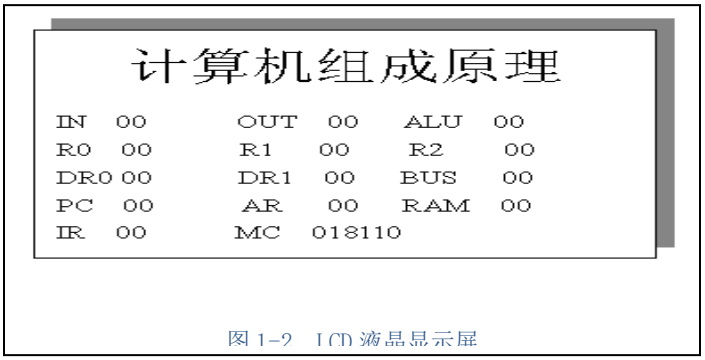


图 1-2 LCD 液晶显示屏

LCD 液晶显示屏功能说明

名称	作 用	名称	作用
IN	输入单元 INPUT	DR1	暂存器 DR1

OUT	输出单元 OUTPUT	DR2	暂存器 DR2
ALU	算术逻辑单元	PC	程序计数器
BUS	内部数据总线	AR	地址寄存器
R0	寄存器 R0	RAM	程序/数据存储器
R1	寄存器 R1	IR	指令寄存器
R2	寄存器 R2	MC	微程序控制器

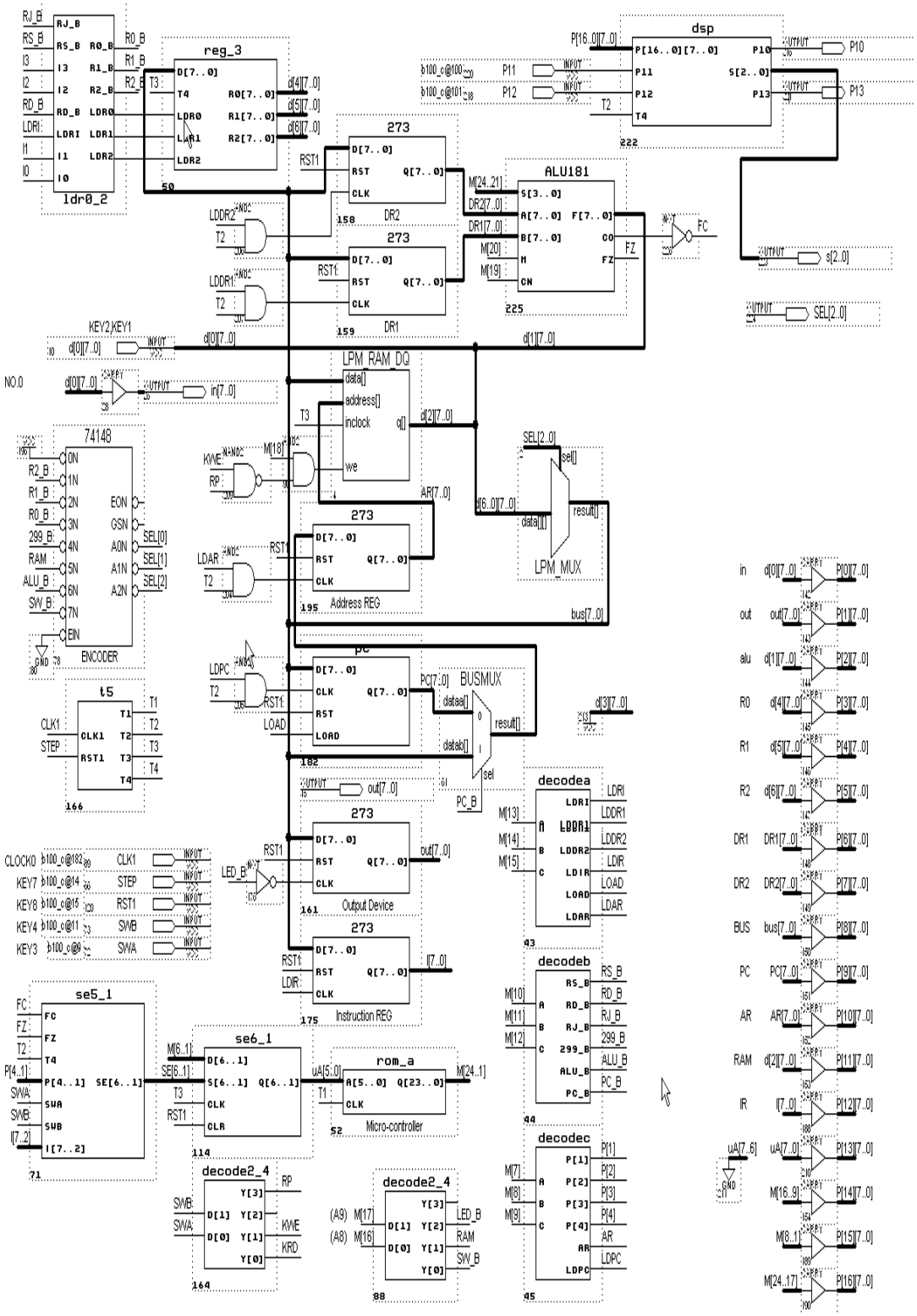


图 1-3 模型计算机电路原理图

四、实验步骤

1. 指令功能设计

本次实验预计完成设计八条指令，他们分别是

指令	功能
IN	从控制台读入数据
SUB	两数相减
INC	自增一运算
XOR	异或运算
CLR	清空 R0 寄存器
NOT	R0 寄存器取反
STA	写内存
SHL	左移运算

2. 指令微程序设计

根据各指令所实现的功能，设计 CPU 内部的数据流操作指令，以及 ALU 运算指令，可以得到如下微程序流程图

3. 二进制微代码设计

全部微程序设计完毕之后，将每条微指令按照微指令格式代码化。转化形成二进制微代码表：

微地址	微指令	S3	S2	S1	S0	M	CN	WE	A9	A8	A	B	C	UA5—UA0
00	018110	0	0	0	0	0	0	0	1	1	000	000	100	010000
01	01ED82	0	0	0	0	0	0	0	1	1	110	110	110	000010
02	00C048	0	0	0	0	0	0	0	0	1	100	000	001	001000
03	00E004	0	0	0	0	0	0	0	0	1	110	000	000	000100
04	00B005	0	0	0	0	0	0	0	0	1	011	000	000	000101
05	01A206	0	0	0	0	0	0	0	1	1	010	001	000	000110
06	619A01	0	1	1	0	0	0	0	1	1	001	101	000	000001
07	059A01	0	0	0	0	0	1	0	1	1	001	101	000	000001
10	001001	0	0	0	0	0	0	0	0	0	001	000	000	000001
11	01ED83	0	0	0	0	0	0	0	1	1	110	110	110	000011
12	01A207	0	0	0	0	0	0	0	1	1	010	001	000	000111

13	01ED95	0	0	0	0	0	0	0	1	1	110	110	110	010101
14	01A21A	0	0	0	0	0	0	0	1	1	010	001	000	011010
15	01A21B	0	0	0	0	0	0	0	1	1	010	001	000	011011
16	01ED9C	0	0	0	0	0	0	0	1	1	110	110	110	011100
17	01A21E	0	0	0	0	0	0	0	1	1	010	001	000	011110
20	01ED92	0	0	0	0	0	0	0	1	1	110	110	110	010010
21	01ED94	0	0	0	0	0	0	0	1	1	110	110	110	010100
22	01A010	0	0	0	0	0	0	0	1	1	010	000	000	010000
23	018001	0	0	0	0	0	0	0	1	1	000	000	000	000001
24	062011	0	0	0	0	0	1	1	0	0	010	000	000	010001
25	00E017	0	0	0	0	0	0	0	0	1	110	000	000	010111
26	00D181	0	0	0	0	0	0	0	0	1	101	000	110	000001
27	00B018	0	0	0	0	0	0	0	0	1	011	000	000	011000
30	01A219	0	0	0	0	0	0	0	1	1	010	010	000	011001
31	699A01	0	1	1	0	1	0	0	1	1	001	101	000	000001
32	01B21B	0	0	0	0	0	0	0	1	1	011	001	000	011011
33	099A01	0	0	0	0	1	0	0	1	1	001	101	000	000001
34	00E01D	0	0	0	0	0	0	0	0	1	110	000	000	011101
35	038201	0	0	0	0	0	0	1	1	1	000	001	000	000001
36	C19A01	1	1	0	0	0	0	0	1	1	001	101	000	000001

4. 编写模型机测试程序

地址（二进制）	内容（二进制）	助记符	说明
0000 0000	0000 0000	IN	“INPUT DEVICE—〉R0
0000 0001	0111 0000	SHL	R0 = R0+R0
0000 0010	0000 1010		
0000 0011	0001 0000	SUB [0BH]	R0 = R0 - [0BH]
0000 0100	0000 1011		
0000 0101	0010 0000	INC	R0 = R0+1
0000 0110	0011 0000	XOR [0CH]	R0 = RO xor [0CH]
0000 0111	0000 1100		
0000 1000	0100 0000	CLR	R0 = 0
0000 1001	0101 0000	NOT	R0 = !R0
0000 1010	0110 0000	STA [0D]	[0D] = R0
0000 1011	0000 1101		

0000 1100	1101 0101	被减数
0000 1101	1010 1100	异或操作数
0000 1110	0000 0000	结果存储位置

5. 实验台调试

实验中 LCD 液晶显示屏可以用来显示模型机 CPU 中各组成单元的内容。将修改好的 ROM 和 RAM 文件添加到工程中，并随同模型 CPU 设计文件一同编译进 SOF 下载文件中，直接下载进入 FPGA，按系统复位键，LCD 液晶显示屏即显示 CPU 中各组成单元的内容。

- 执行程序

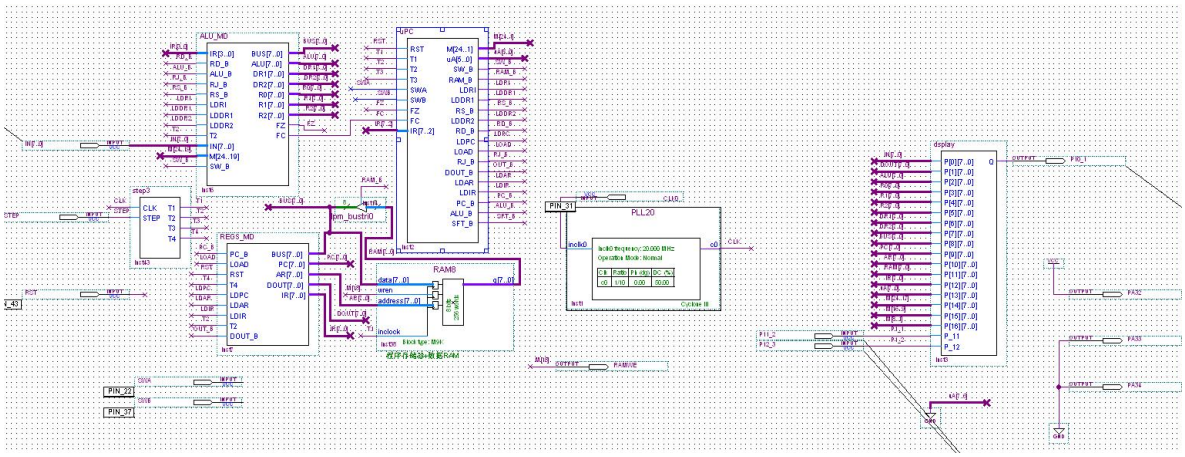
  1. 按 1 次系统复位键 8，并置键 8 为高电平，使 CPU 允许正常工作；
  2. 控制开关（键 4、键 3）设置为 SWB、SWA=1,1，处于程序执行方式
  3. 通过键 2、键 1 输入运算数据，如 56H，按 4 次单步键 7，产生 2 个脉冲，执行 2 条微指令，进入到图 1-4 控制台的 RP（11），此时的微指令地址是“23”，微指令码 MC=008001；IN=56H
  4. 再用键 7 产生 1 个脉冲，执行 1 条微指令，微程序流程进入图 1-4 左的“运行微程序”的最上块：此时 PC=00 送地址寄存器 AR=00，PC 自动加 1，PC=01，MC=00ED82，IN=56；
  5. 数码管 3 显示的是进位情况，有进位，LED3 显示 1，无进位 LED3 显示 0。
  6. 键 7 产生 1 个脉冲，执行微指令 MC=00C048（图 1-4），RAM 中的第一条指令码 00 进入 BUS，再由 BUS 进入指令寄存器 IR=00。键 7 再进 1 个脉冲，进入 MC=001001，执行指令 IN，送数 IN -> R0=56;
  7. 键 7 产生 1 个脉冲，执行完 IN 指令后，返回到初始端，执行微指令 MC=00ED82；
  8. 以此往复执行，直到模型机测试程序执行结束

- EAB 在系统读写

  使用在系统 EAB 读写工具对模型 CPU 中的存放微程序的 ROM 和存放程序与数据的 RAM 进行观察和改写

五、实验结果

模型 CPU 设计电路图：



CPU ROM 内容：

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	018110	01ED82	00C048	00E004	00B005	01A206	619A01	059A01
08	001001	01ED83	01A207	01ED95	01A21A	01A21B	01ED9C	01A21E
10	01ED92	01ED94	01A010	018001	062011	00E016	00B017	00B018
18	01A219	699A01	01B219	099A01	00E01D	038201	C19A01	000000
20	000000	000000	000000	000000	000000	000000	000000	000000
28	000000	000000	000000	000000	000000	000000	000000	000000
30	000000	000000	000000	000000	000000	000000	000000	000000
38	000000	000000	000000	000000	000000	000000	000000	000000

CPU RAM 内容：

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	00	70	10	0B	20	30	0C	40
08	50	60	0D	D5	AC	0D	03	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00

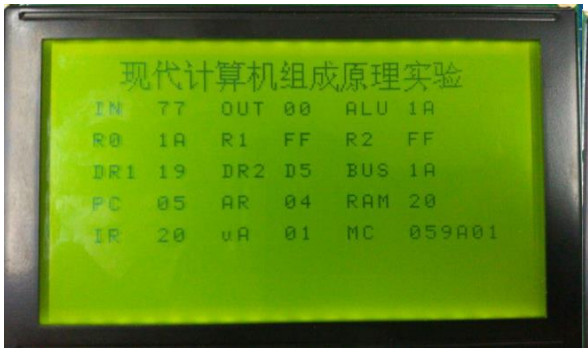
FPGA 测试：

初始状态

IN 指令读入 77H

左移运算 EEH

减法运算 EEH-D5H



增一运算

异或运算

清零运算

写入 RAM

在线读取 RAM

可以看到指定位置已经被写入数据

六、实验心得

这学期的计算机组成原理课程设计让我受益菲浅。在实验课即将结束之时，我对在这一学期的学习进行了总结，总结这一学期的收获与不足。取之长、补之短，在今后的学习和工作中有所受用。

没有接触过计算机或者对计算机不是特别了解的人可能觉得计算机特别神秘而且不知道为什么它可以实现那么复杂的功能，而就我们而言越是深入学习越是渴望了解其工作原理。很幸运这学期我们开设了计算机组成原理课程设计，之所以将计算机组成原理实验独立成一门课程，足见其重要性。也体现了现在大学教学中对大学生动手能力的重视。《计算机组成原理实验系统》这门课程是我们计算机专业一门很重要的专业课起着承上启下的作用。这学期的课程设计，通过自己的亲自操作让我对计算机的基本结构，基本组成与结构原理有了更加深入的了解，课程设计对于我们了解现代计算机的各个组成部分及其工作原理有重要作用，通过自己动手做实验使我们更加深入的了解了计算机 CPU 的实现方法。本次课程设计，从微指令的设计，到二进制微代码的翻译，八条指令设计完毕后，我对 CPU 具体功能的实现方法有了更深刻的理解。但是由于课程及实验时间的限制，我想我们学到的东西还是太少了，不过没关系，这毕竟为我们以后的学习打下了基础。

总之，这次课程设计给我提供了动手实验的机会，使我对计算机组成原理的相关知识有了更深的印象和认识。计算机组成原理是计算机专业的基础课。这门课对于使我们了解现代计算机的各个组成部分及其工作原理具有重要作用，对于我们后续课程的学习无疑也具有积极的意义。计算机专业是一个很渊博的专业，我们现在有很好的机会站在巨人的肩膀上学习，虽然通过这学期的课程设计学到了很多知识，但那只是计算机知识海洋中的一滴，我将继续努力对计算机组成原理方面进行深入的研究，了解更多计算机方面的知识，为以后打下坚实的基础。