专业：电子科学与技术
姓名：
学号：
日期：
地点：玉泉

浙江大学实验报告

课程名称：微机原理与接口技术　指导老师：　成绩：
实验名称：第一次软件实验　实验类型：软件实验　同组学生姓名：无

实验一

1、实验目的

掌握数据传送指令；掌握各种数据传送指令的寻址方式；掌握数据传送、复制等程序的设计与调试方法。

2、基础实验部分

（1）给外部 RAM8000～80FFH 的 256 个单元的内容赋值，赋值的内容取决于程序中 A 的赋值。实验结果如图 1~图 3 所示。

8000	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8010	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8020	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8030	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8040	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8050	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8060	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8070	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8080	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
8090	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80A0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80B0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80C0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80D0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80E0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
80F0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01

图 1 XDATA 对应内存段的数据

名称	值	名称	值
ACC	01	.7	1
B	00	.6	0
DPH	81	.5	0
DPL	00	.4	0
IE	00	.3	0
IP	00	.2	0
P0	FF	.1	0
P1	FF	.0	1
P2	FF		
P3	FF		
PCON	00		
PSW	01		
SBUF	00		
SCON	00		
SP	07		
TCON	00		
TH0	00		
TH1	00		
TL0	00		

图 2 SFR 数据

R0	00	.7	0
R1	00	.6	0
R2	00	.5	0
R3	00	.4	0
R4	00	.3	0
R5	00	.2	0
R6	00	.1	0
R7	00	.0	1
A	01		
B	00		
DPH	81		
DPL	00		
PSW	01		
SP	07		

图 3 寄存器数据

程序运行结束后，片外 RAM 中 8000H~80FFH 地址所保存的内容为寄存器 A 中的内容 01H，寄存器 R0 的最终值为 0，DPTR 的最终值为 8100H。

（2）将 3000H 起始的 256 个字节存储块移动到 4000H 起始的 256 个字节存储块，实验结果如图 4~图 7。

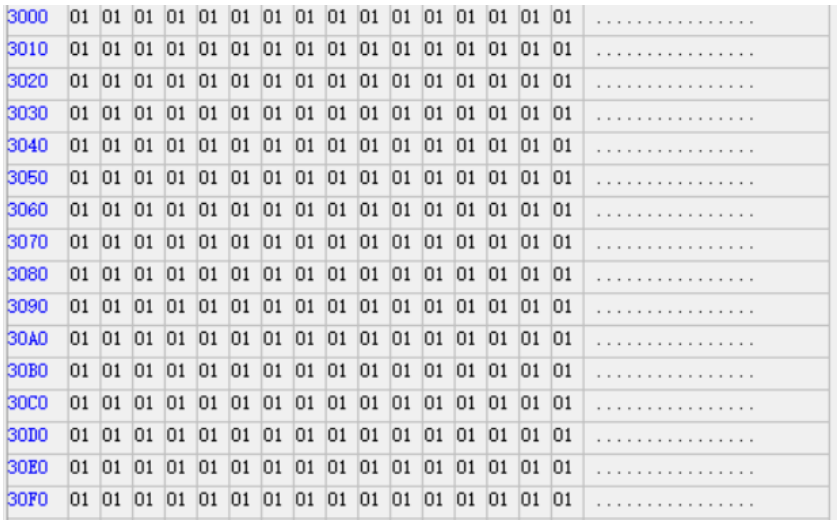
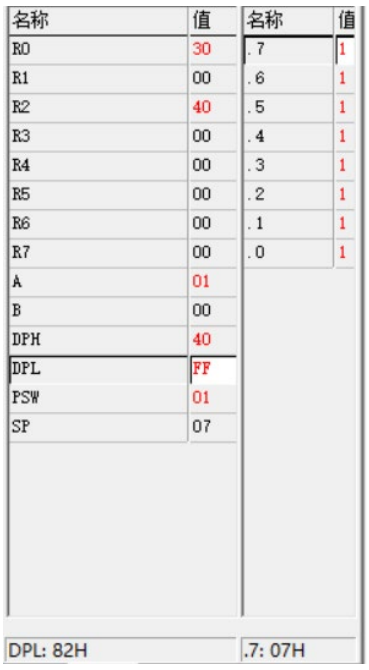


图 4 片外 RAM 源数据及地址段（3000H~30FFH）

图 5 片外 RAM 目标地址段（4000H~40FFH）及数据

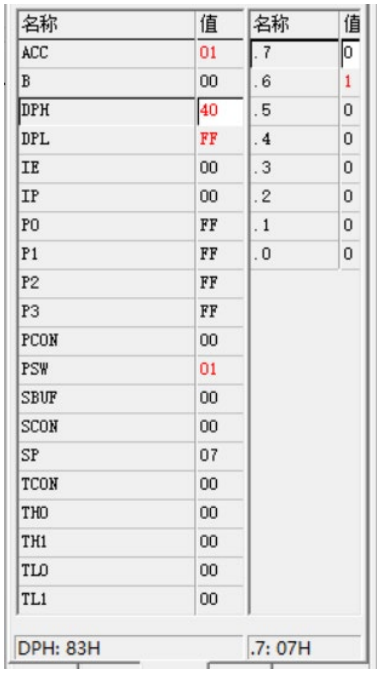
程序运行结束后，片外 RAM 中地址 3000H~30FFH 的数据被搬运到 4000H~40FFH 中，其值为寄存器 A 的值 01H。寄存器 R1，R3 的最终值为 00H（在 INC R1、INC R3 指令中各自多经历了一次自增），DPTR 的最终值为 40FFH。



名称	值	名称	值
R0	30	.7	1
R1	00	.6	1
R2	40	.5	1
R3	00	.4	1
R4	00	.3	1
R5	00	.2	1
R6	00	.1	1
R7	00	.0	1
A	01		
B	00		
DPH	40		
DPL	FF		
PSW	01		
SP	07		

DPL: 82H .7: 07H

图 7 寄存器数据

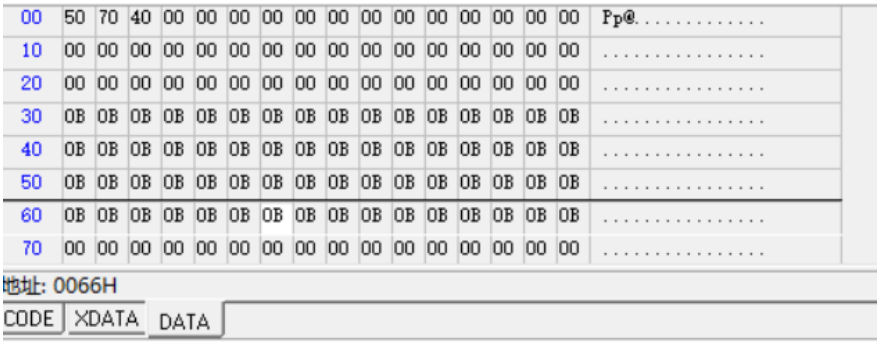


名称	值	名称	值
ACC	01	.7	0
B	00	.6	1
DPH	40	.5	0
DPL	FF	.4	0
IE	00	.3	0
IP	00	.2	0
P0	FF	.1	0
P1	FF	.0	0
P2	FF		
P3	FF		
PCON	00		
PSW	01		
SBUF	00		
SCON	00		
SP	07		
TCON	00		
TH0	00		
TH1	00		
TL0	00		
TL1	00		

DPH: 83H .7: 07H

图 7 SFR 数据

（3）原程序的功能为：将片内 RAM 中地址 30H~4FH 中的内容复制到 50H 到 6FH。用 VW 软件将 30H~40H 中的数据设为 0BH，实验结果如图 8~图 9 所示。

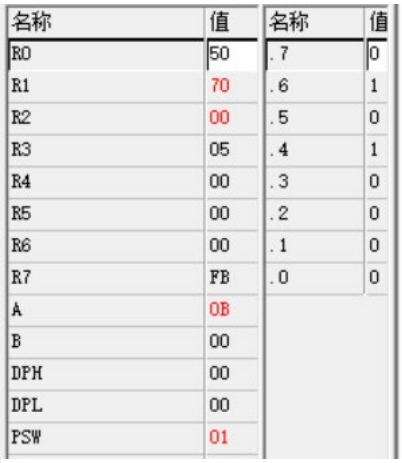


00	50	70	40	00	00	00	00	00	00	00	00	00	00	00	00	00	Pp@. . . . .
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
30	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	. . . . .
40	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	. . . . .
50	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	. . . . .
60	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B	. . . . .
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .

地址: 0066H

CODE XDATA DATA

图 8 片内 RAM 地址 30H~4FH 中的数据



名称	值	名称	值
R0	50	.7	0
R1	70	.6	1
R2	00	.5	0
R3	05	.4	1
R4	00	.3	0
R5	00	.2	0
R6	00	.1	0
R7	FB	.0	0
A	0B		
B	00		
DPH	00		
DPL	00		
PSW	01		

图 9 寄存器数据

按要求更改程序如下：

```
        ORG 0000H
        MOV P2, #00H      ;设置高八位地址为 0
        MOV R0, #30H      ;设置 R0
        MOV R1, #50H      ;设置 R1
        MOV R2, #20H      ;设置 R2
L1:     MOVX A, @R0       ;将 R0 指向的内容赋值给 A
        MOVX @R1, A       ;将 A 的内容赋值给 R1 指向的地址
        INC R0            ;R0 自增改变地址
        INC R1            ;R1 自增改变地址
        DJNZ R2, L1       ;循环判断
        SJMP $
        END
```

Code 1 **修改后的程序**

修改后的程序功能为：将外部 RAM 地址 0030H~004FH 中的数据复制到 0050H 到 006FH。由于 MOVX A, @R0 指令中 R0 只提供外部 RAM 的低八位地址，高八位为 P2 口提供，故指令 MOV P2, #00H 将 P2 口设置为 00H，这样就可以实现对应的功能。实验结果如图 10~图 11 所示。

图 10 **片外** RAM **地址** 0030H~004FH **块中的数据与** 0050H~006FH **块中的数据是相同的，说明复制正确**

3、设计实验部分

（1）在 WAVE 环境修改内部 RAM 30H~3FH 的内容分别为#00H~#0FH，设计程序实现将内部 RAM30H~3FH 到 40H~4FH 的数据块拷贝。

```
        ORG 0000H
        MOV R0, #30H      ;设置源数据起始地址
        MOV R1, #40H      ;设置目标位置起始地址
        MOV R2, #10H      ;设置循环计数
L1:     MOV A , @R0       ;将 R0 所指向的源数据赋值给寄存器 A
        MOV @R1, A        ;将 A 中的值赋值给 R1 所指向的目标地址
        INC R0            ;改变源地址
        INC R1            ;改变目标地址
        DJNZ R2, L1       ;循环判断
        SJMP $
        END
```

Code 2 **实验一设计实验** 1 **程序**

程序如 Code2 所示，实验结果如图 11 所示。可见该程序成功实现了数据拷贝复制。

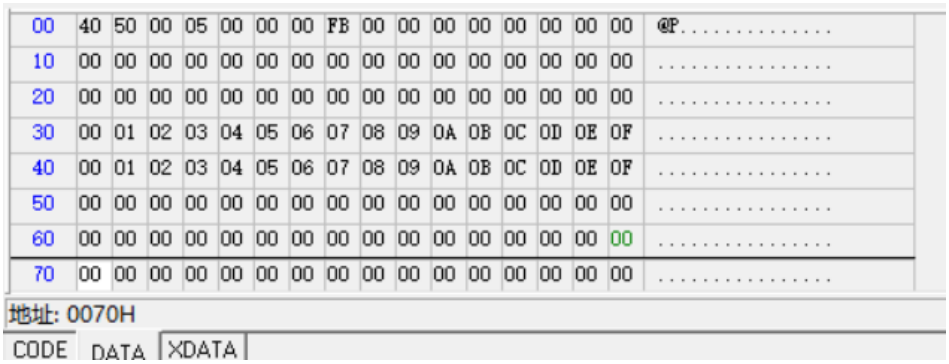


图 11 程序运行后片内 RAM 相应地址的数据

（2）在 WAVE 环境修改内部 RAM 30H~3FH 的内容分别为#00H~#0FH，设计程序实现将片内 30H~3FH 单元的内容复制到片外 1030H~103FH 中。

程序如 Code3 所示，实验结果如图 12 所示。

```
        ORG 0000H
        MOV R0, #30H          ;工作寄存器 R0 指向源数据首地址
        MOV DPTR, #1030H      ;DPTR 指向目的地址
        MOV R7, #10H          ;传送的字节数
MLP:    MOV A, @R0            ;取出源数据
        MOVX @DPTR, A         ;送入目标地址
        INC R0                ;改变源地址指针
        INC DPTR              ;改变目标地址指针
        DJNZ R7, MLP          ;未完成则循环
        NOP
        SJMP $
```

Code 3 实验一设计实验 2 程序

图 12 片外 RAM 数据，1030H~103FH 中的数据为片内 30H~3FH 的拷贝

（3）修改内部 RAM 30H~3FH 的内容分别为#00H~#0FH，设计程序实现将内部 RAM 30H~3FH 内容逆序拷贝到外部数据 XRAM：0000H~000FH 中。

```
        ORG 0000H
        MOV R0, #3FH          ;工作寄存器 R0 指向源数据末尾地址
        MOV DPTR, #0000H      ;DPTR 指向目的首地址
        MOV R7, #10H          ;传送的字节数
MLP:    MOV A, @R0            ;取出源数据
        MOVX @DPTR, A         ;送入目标地址
        DEC R0                ;改变源地址指针
        INC DPTR              ;改变目标地址指针
        DJNZ R7, MLP          ;未完成则循环
        NOP
        SJMP $
```

Code 4 **实验一设计实验 3 程序**

程序如 Code4 所示，实验结果如图 13 所示。由于 DPTR 没有递减指令，故选择从源数据的末尾地址开始递减，用 R0 储存源数据地址，以此实现倒序复制。

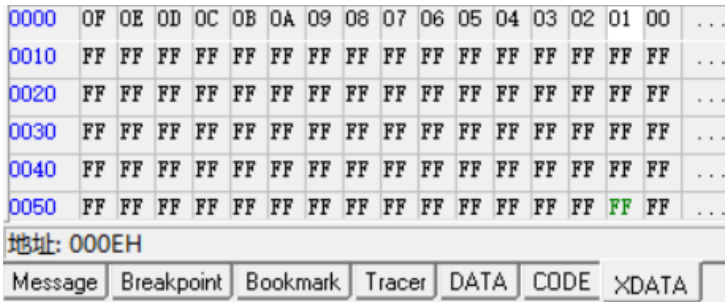


图 13 **片外 RAM 相应数据，0000H~000FH 中的数据为片内 RAM30H~3FH 数据的倒序**

实验二

1、实验目的

了解微机系统中的数制与代码表示方法；掌握计算机中使用的各种代码转换方法；掌握实现分支、循环的指令及其程序的编写方法。

2、基础实验部分

（1）完成单字节的 ASCII 码到十六进制数转换。补全书中的程序，如 Code5。实验结果如图 14~15。

```
RESULT EQU 30H
ORG 0000H
MOV A, #41H
CLR C
SUBB A, #37H          ;ASCII 码转换为 16 进制，相差 37H
MOV RESULT, A
LJMP $
END
```

Code 5 **实验二基础实验 1 程序**

00	50	70	00	05	00	00	0F	00	00	00	00	00	00	00	00	00	Pp. . . . .
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
30	0A	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	. . . . .
40	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	. . . . .
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .

图 14 片内 RAM 相应地址的数据

名称	值	名称	值
R0	50	.7	0
R1	70	.6	0
R2	00	.5	0
R3	05	.4	0
R4	00	.3	1
R5	00	.2	0
R6	0F	.1	1
R7	00	.0	0
A	0A		
B	00		
DPH	00		
DPL	00		
PSW	40		
SP	07		

图 15 寄存器数据

程序运行后，字母‘A’对应的十六进制数 0AH 被存放到片内 RAM 地址为 30H 的区域，寄存器 A 中的值为 0AH。

（2）单字节 BCD 码到十六进制数转换。书中程序有误，重写如 Code6 所示。

```
RESULT EQU 30H
ORG 0000H
MOV A, #23H
MOV R0, A
ANL A, #0F0H
SWAP A
MOV B, #0AH
MUL AB
MOV RESULT, A        ;转换高位
MOV A, R0
ANL A, #0FH
ADD A, RESULT
MOV RESULT, A        ;转换低位
SJMP $
END
```

Code 6 实验二基础实验 2 程序

实验结果如图 16~图 17 所示。片内 RAM 地址 30H 的单元储存了转换的结果。单字节 BCD 码 23H 对应的十进制数为 23，十六进制数为 17H。

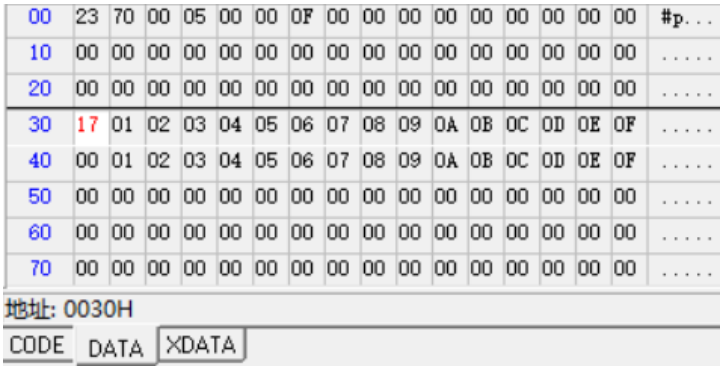


图 16 片内 RAM 相应地址的数据

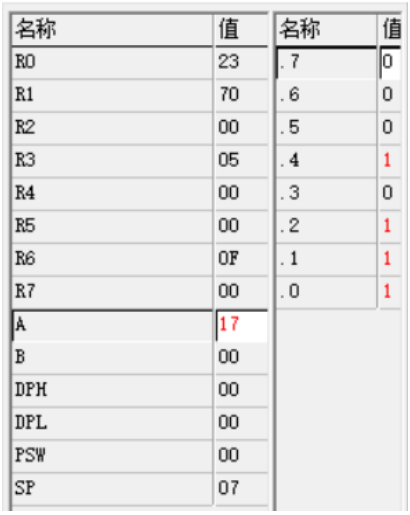


图 17 寄存器数据

（3）将单字节十六进制数 7BH 的值转换为十进制数，存放在 30H～32H 中。补全程序如 Code7 所示，实验结果如图 18~图 19 所示。

```
RESULT EQU 30H
ORG 0000H
MOV A, #7BH
MOV B, #64H          ;除以 100 获得百位
DIV AB
MOV RESULT, A
MOV A, B
MOV B, #0AH          ;除以 10 获得十位
DIV AB
MOV RESULT+1, A
MOV RESULT+2, B
SJMP $
END
```

Code 7 实验二基础实验 3 程序

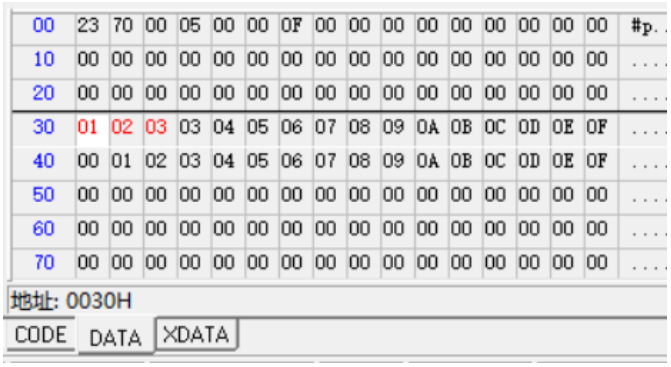


图 18 片内 RAM 相应地址的数据

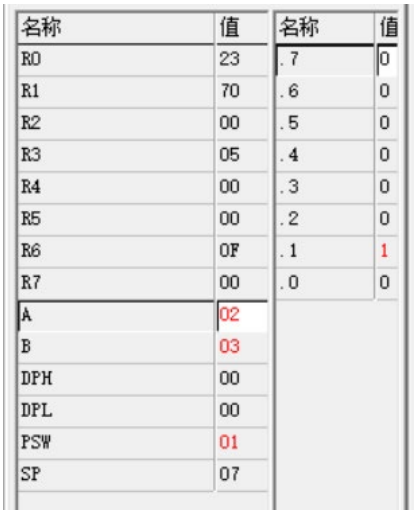


图 19 寄存器数据

7BH 对应的十进制数为 123，故片内 RAM 30H~32H 中的数据分别为 1、2、3。

3、设计实验部分

（1）设一串字母的 ASCII 存于 30H 起始的单元中，设计程序判断字母是否为大写字母，是则将大写字母的 ASCII 字符转换成小写字母的 ASCII 字符，为小写则不转换。程序如 Code8 所示，实验结果如图 20 所示。

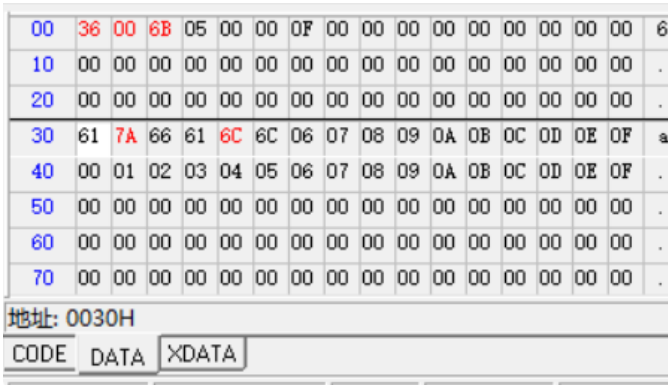


图 20 片内 RAM 相应地址的数据

```
        ORG 0000H
        MOV R0, #30H          ;设置首地址并开始初始化内存数据
        MOV 30H, #41H         ;(30H) = 'A'
        MOV 31H, #5AH         ;(31H) = 'Z'
        MOV 32H, #66H
        MOV 33H, #61H         ;(33H) = 'a'
        MOV 34H, #4CH         ;(34H) = 'L'
        MOV R1, #06H
LOOP:   CJNE @R0, #41H, NEXT1
NEXT1:  JC LAST               ;比‘A’（41H）小，不变
        MOV 02H, @R0          ;判断比‘Z’（5AH）大时，因为 JNC 在 Cy=0 时转移
        DEC R2                ;而 5AH-5AH=0，Cy=0，故需要将带判断的数减去 1
        CJNE R2, #5AH, NEXT2  ;用修正后的值进行判断，确保正确
NEXT2:  JNC LAST              ;比‘Z’（5AH）大，不变
        MOV A, #20H           ;大小写字母的 ASCII 码相差 20H。
        ADD A, @R0
        MOV @R0, A
LAST:   INC R0                ;修改源地址
        DJNZ R1, LOOP         ;未完成则继续循环
        SJMP $
        END
```

Code 8 **实验二设计实验 1 程序**

可以看到，程序将大写字母‘A’（41H）、‘Z’（5AH）、‘L’（4CH）分别转换为了小写字母‘a’（61H）、‘z’（7AH）、‘l’（6CH），而小写字母和其他字符则未进行转换。

（2）将单字节十六进制数 D8H 转换为十进制数，存放在 30H～32H 中。程序如 Code9 所示，实验结果如图 21 所示。

```
RESULT EQU 30H
ORG 0000H
MOV A, #0D8H
MOV B, #64H
DIV AB
MOV RESULT, A         ;除以 100 得百位数
MOV A, B
MOV B, #0AH
DIV AB
MOV RESULT+1, A       ;除以 10 得十位数
MOV RESULT+2, B       ;余数为个位数
SJMP $
END
```

Code 9 **实验二设计实验 2 程序**

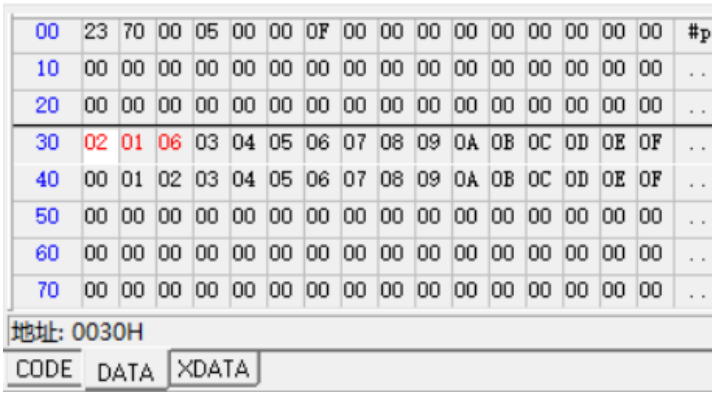


图 21 片外 RAM 相应地址的数据

十六进制数 D8H 对应的十进制数为 216，因此片内 RAM 地址 30H~32H 中的数据分别为 2、1、6。

实验总结与心得

本次实验主要熟悉了 51 微控制器指令，掌握了数据传输、进制转换等操作，对片内 RAM、片外 RAM、寄存器等不同地址空间的操作有了更深的认识。同时熟悉并掌握了汇编程序的编写、调试、验证过程。

本次实验遇到的主要问题是用 R0、R1 对片外 RAM 寻址时，忽略了高八位的地址受到了 P2 口的影响，误以为高八位是 00H，而实际上 P2 口初始化的结果为 FFH，导致实验结果与预期结果不同。经过助教的帮助和解释后理解了这个问题，修改程序，将 P2 口设 00H，成功达到预期结果。