信息与软件工程学院上机实验报告

(第 1 次)

一、实验名称

13.1.1 实验一 数据传送

二、实验目的及要求

1. 实验目的
   通过本次实验掌握 Keil uVision3 开发环境以及单片机汇编语言和C语言源程序编辑、编译、调试方式方法, 熟悉单片机汇编语言指令, 掌握单片机片内外存储器间数据传送的方法, 用汇编语言和C语言编程实现单片机片内外存储器存取数据。
2. 实验要求
   在本次实验前, 在自己的电脑安装 Keil uVision3 以上版本的软件, 根据实验内容和程序框图编写汇编语言和C51程序。

三、实验环境

仿真实验: 装有 Keil uVision3 以上版本软件的笔记本或台式PC机。

四、实验内容

编写软件实现: 设置单片机片内存储器存储区首地址为30H、片外存储器存储区首地址为3000H, 存取数据字节个数16个, 将片内存储区内容设置为01H~10H共16个字节, 读取片内首地址为30H单元内容, 将该内容传送到片外数据存储器存储区中保存, 将保存在片外数据区数据依次取出送P1。观察片内、外存储区数据变化、P1口状态变化。

五、实验设计

1、实验步骤

实验流程图

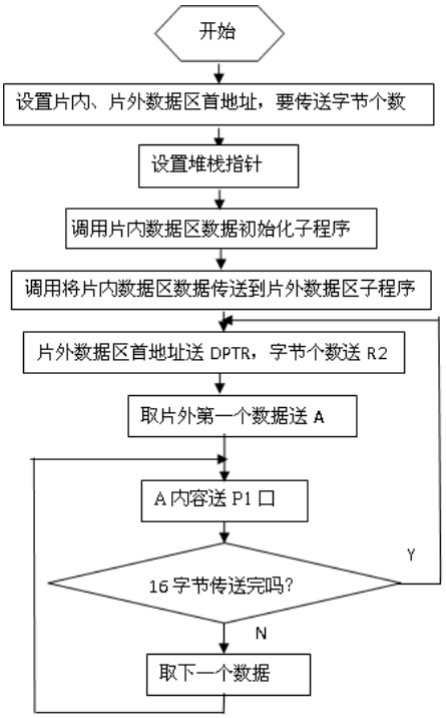


图 13-1 数据传送实验主程序框图

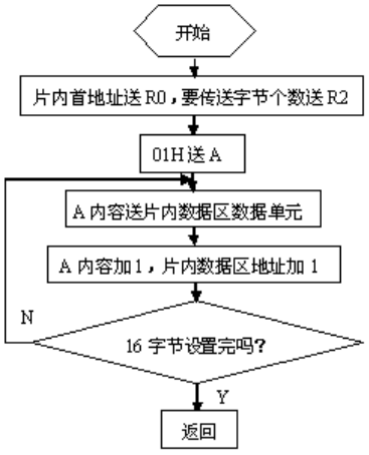


图 13-2 片内数据区数据初始化子程序框图

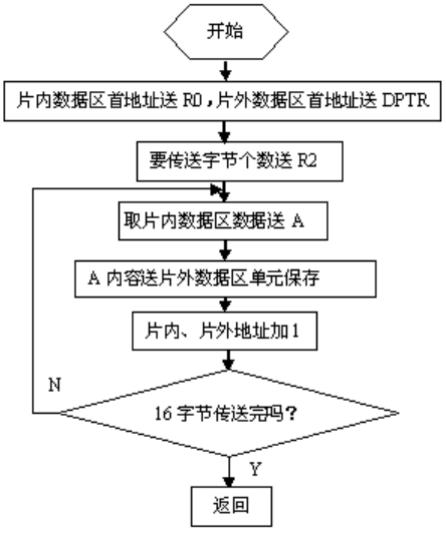


图 13-3 片内数据传送到片外数据区子程序框图

设计思想

本实验利用 8051 单片机的寻址方式和数据传送指令，实现片内 RAM、片外 RAM 和 I/O 端口之间的数据交换。

- **模块化设计**：将程序划分为三个核心模块：片内数据初始化模块、片内到片外数据传送模块、片外数据到 P1 口显示模块。
- **寻址方式**：
  - 片内 RAM 访问：使用寄存器间接寻址方式 (MOV @R0, A 和 MOV A, @R0)，通过 R0 作为地址指针。
  - 片外 RAM 访问：使用数据指针 DPTR 进行寻址 (MOVX A, @DPTR 和 MOVX @DPTR, A)。
  - I/O 端口访问：使用直接寻址方式 (MOV P1, A)。
- **循环控制**：利用 DJNZ 指令来控制16个字节的数据块初始化和传送。
- **数据显示**：设计一个无限循环，将片外 RAM 的数据依次、重复地显示在 P1 端口，以便于在仿真环境中观察结果。

实现步骤

1. **初始化设置**：程序开始后，首先设置堆栈指针 SP 的值（如60H），为后续的子程序调用做准备。
2. **片内数据区初始化**：调用初始化子程序。将片内 RAM 首地址 30H 送入 R0，数据块长度 16 (10H) 送入计数器 R2。将初始数据 01H 送入累加器 A。进入循环：将 A 中的数据存入 R0 指向的地址，然后 A 和 R0 分别加 1，R2 减 1，直到 16 字节（01H~10H）全部写入片内 30H~3FH 区域。
3. **片内到片外数据传送**：调用数据传送子程序。将片内首地址 30H 送入 R0，片外首地址 3000H 送入 DPTR。将数据块长度 16 (10H) 送入计数器 R2。进入循环：先从 R0 指向的片内地址取数到 A，再将 A 中的数据存入 DPTR 指向的片外地址。随后 R0 和 DPTR 分别加 1，R2 减 1，直到 16 字节数据全部复制到片外 3000H~300FH 区域。
4. **数据显示**：执行无限循环。将片外首地址 3000H 送入 DPTR，数据块长度 16 (10H) 送入计数器 R2。进入循环：从 DPTR 指向的片外地址取数到 A，然后将 A 的内容输出到 P1 端口。接着 DPTR 加 1，R2 减 1，循环 16 次。结束后，跳转回数据显示循环的起始位置，重新开始显示。

2、调试过程及实验结果

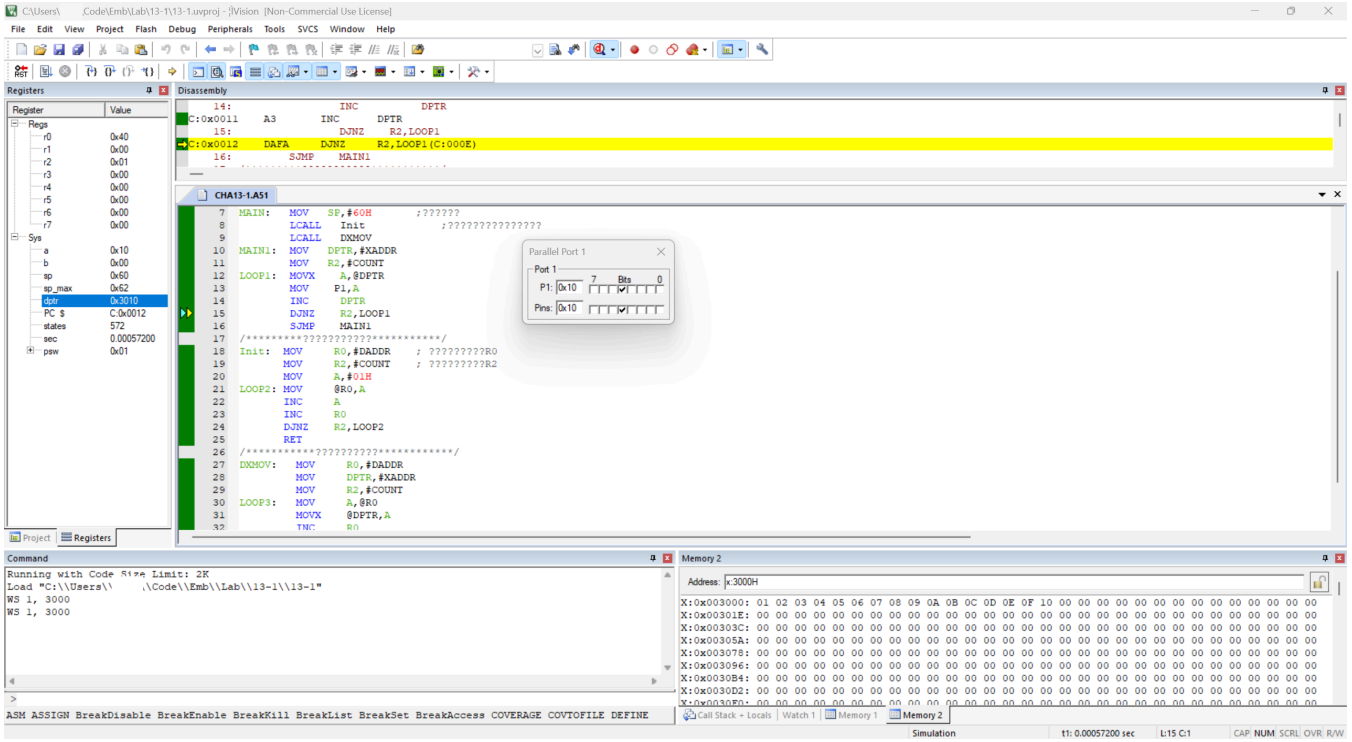
调试过程：

1. 在 Keil uVision3 环境中编译程序，确保没有语法错误。
2. 进入 Debug 调试模式。
3. 打开三个关键窗口：
   - **片内RAM (Internal RAM)**: 打开 `View -> Memory Windows -> Memory 1`。在地址栏输入 `d:30H`。
   - **片外RAM (External RAM)**: 打开 `View -> Memory Windows -> Memory 2`。在地址栏输入 `x:3000H`。
   - **P1 端口**: 打开 `Peripherals -> I/O-Ports -> Port 1`。
4. 使用单步调试 (Step F11) 执行 `LCALL Init` 子程序。观察 Memory 1 (`d:30H`) 窗口，数据从 30H 到 3FH 被依次写入 01H, 02H, ... 10H 。
5. 继续单步调试 (F11) 执行 `LCALL DXMOV` 子程序。观察 Memory 2 (`x:3000H`) 窗口，数据从 3000H 到 300FH 被依次写入 01H, 02H, ... 10H，与片内数据一致。
6. 执行 `MAIN1` 循环，使用连续运行。

实验结果：

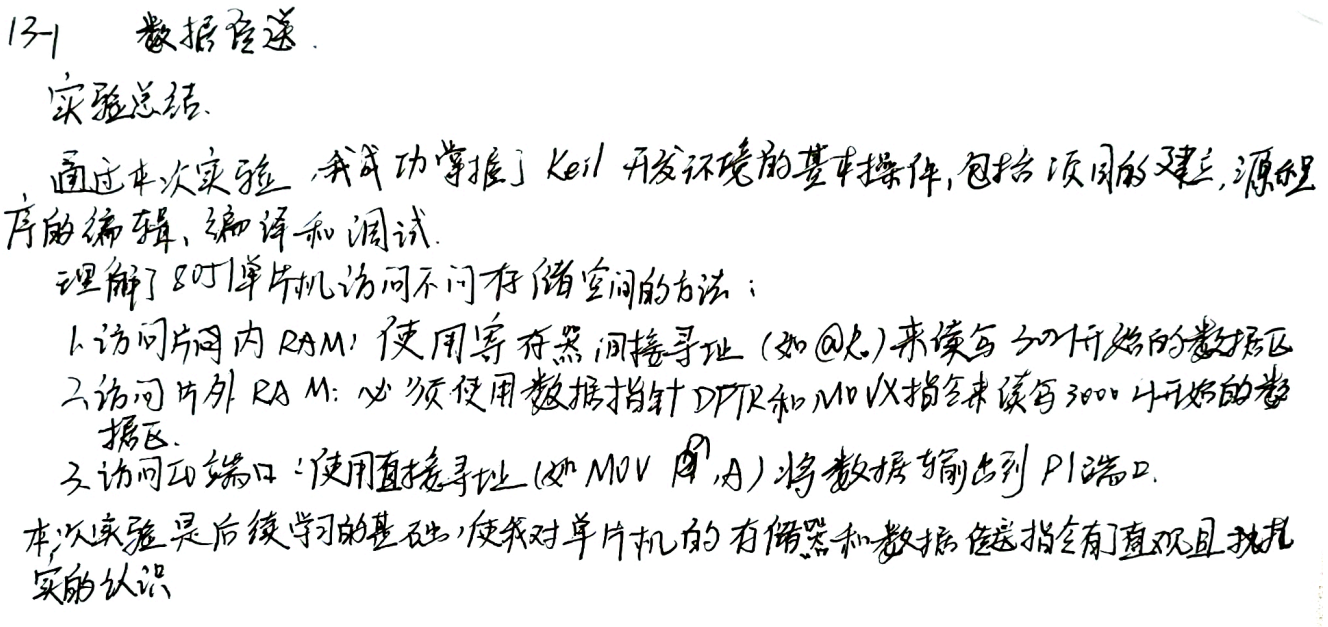
调试运行结果与实验指导书的预期一致。

1. **片内存储器**：`d:30H` 至 `d:3FH` 的 16 个单元内容被成功初始化为 01H 到 10H。
2. **片外存储器**：`x:3000H` 至 `x:300FH` 的 16 个单元内容被成功从片内复制，显示为 01H 到 10H。
3. **P1 端口**：Port 1 窗口的数据循环显示 01H, 02H, 03H... 直到 10H，然后再次从 01H 开始循环。



3、实验改进

4、总结



4、附录

```
/*实验一、文件名：cha13-1.A51*/
        ORG     0000H
DADDR   EQU     30H     ;片内数据区首地址
XADDR   EQU     3000H   ;片外数据区首地址·
COUNT   EQU     10H     ;传送数据大小，共16个字节。

MAIN:   MOV   SP,#60H       ;重置堆栈指针
        LCALL   Init            ;调用片内数据区数据初始化子程序
        LCALL   DXMOV
MAIN1:  MOV   DPTR,#XADDR
        MOV   R2,#COUNT
LOOP1:  MOVX    A,@DPTR
        MOV     P1,A
        INC     DPTR
        DJNZ    R2,LOOP1
        SJMP    MAIN1
/*********片内数据区初始化子程序***********/
Init:    MOV     R0,#DADDR  ; 片内数据区首地址送R0
        MOV     R2,#COUNT    ; 传送数据字节个数送R2
        MOV     A,#01H
LOOP2:  MOV     @R0,A
        INC     A
        INC     R0
        DJNZ    R2,LOOP2
        RET
/***********片内外数据传送子程序************/
DXMOV:   MOV     R0,#DADDR
         MOV     DPTR,#XADDR
         MOV     R2,#COUNT
LOOP3:   MOV     A,@R0
         MOVX    @DPTR,A
         INC     R0
         INC     DPTR
         DJNZ    R2,LOOP3
         RET
```