学号       **E21714049**      专业 **计算机科学与技术**     姓名 **梅世祺**

实验日期 **2019.10.1**     教师签字     成绩

实验报告

【实验名称】 **微机原理实验一**

【实验目的】

**实验一**

1. 了解存储器扩展的方法和存储器的读/写。
2. 掌握CPU对16位存储器的访问方法。

**实验二**

1. 了解存储器扩展的方法和存储器的读/写。
2. 掌握CPU对16位存储器的访问方法以及如何正确输入和显示字符串。

**实验三**

1. 学习并掌握8255的工作方式及其应用。
2. 掌握8255典型应用电路的接法。
3. 掌握程序固化及脱机运行程序的方法。

【实验内容】

**实验一**  编写程序，将内存3500H单元的8个数据复制到3600H单元开始的数据区中，通过调试验证数据功能，使用E命令修改3500H单元开始的数据，运行程序后使用D命令查看3600H单元开始的数据。

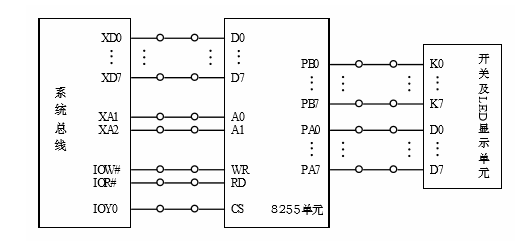
**实验二** 定义一个数据段，里面有2个字符串变量，一个里面有初值，另一个没有，用0AH，输入字符串，然后利用09H，显示两个字符串。

**实验三** 使8255的A口和B口均为输出，数据灯D7~D0由左向右，每次仅亮一个灯，循环显示，D15~D8与与D7~D0正好相反，由右向左，每次仅点亮一个灯，循环显示，实验接线图如下图所示，实验步骤如下所述：

1. 按图2-6-5连接实验线路图；
2. 编写实验程序，经编译、链接无误后装入系统；
3. 运行程序，观察LED灯的显示，验证程序功能；
4. 自己改变流水灯的方式，编写程序。

额外要求：

1. 要求每次间隔地显示两个灯
2. 要求流水灯每次走两步



 【实验原理】

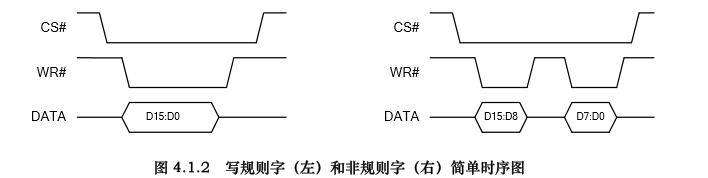
**实验一、二**

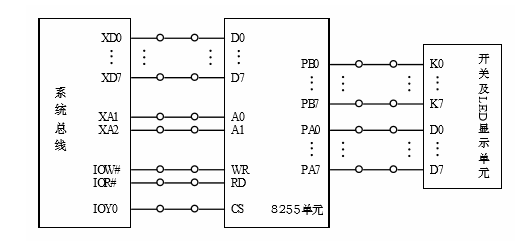
存储器是用来存储信息的部件,是计算机的重要组成部分，静态RAM是由MOS管组成的触发器电路，每个触发器可以存放1位信息。只要不掉电，所储存的信息就不会丢失。因此，静态RAM工作稳定，  不要外加刷新电路，使用方便。但一般SRAM的每-个触发器是由6个晶体管组成，SRAM芯片的集成度不会太高，目前较常用的有6116 (2Kx8位)，6264 (8Kx8位)和62256 (32Kx8位)。本实验平台上选用的是62256，两片组成32KX16位的形式，共64K字节。62256的外部引脚图如图4.1.1所示。



      本系统采用准32位CPU,具有16位外部数据总线，即D0、D1、. D15,地址总线为BHE# ( #表示该信号低电平有效)、BLE#、A1、A2、A20。存储器分为奇体和偶体，分别由字节允许线BHE#和BLE#选通。

      存储器中,从偶地址开始存放的字称为规则字，从奇地址开始存放的字称为非规则字。处理器访问规则字只需要-一个时钟周期，BHE#和BLE#同时有效，从而同时选通存储器奇体和偶体。处理器访问非规则字却需要两个时钟周期，第-一个时钟周期BHE#有效，访问奇字节;第二个时钟周期BLE#有效，访问偶字节。处理器访间字节只需要-一个时钟周期，视其存放单元为奇或偶，而BHE#或BLE #有效，从而选通奇体或偶体。写规则字和非规则字的简单时序图如图4.1.2所示。





**8255基本输入输出实验接线图**

**实验步骤**

1. 运行Wmd86软件，进入Wmd86集成开发环境。
2. 根据程序设计使用语言的不同，通过在“设置”下拉列表来选择需要使用的语言和寄存器类型，这里我们设置成“汇编语言”和“16位寄存器”。



1. 语言和寄存器选择后，点击新建或按CRTL+N组合键来新建一个文档。默认文件名位Wmd861.

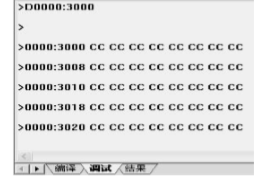


1. 编写实验程序并保存。（具体代码将在之后给出）。
2. 点击，编译文件，若程序编译无误，则可以继续点击，进行连接，连接无误后方可以加载程序。编译、链接后输出如图所示信息。



6 .连接PC与实验系统的通讯电缆，打开实验系统电源。

7. 编译、链接都正确并且上下位机通讯成功后，就可以下载程序，联机调试了。可以通过端口列表中“端口测试”来检查是否通讯正常。接着按下载、编译、链接组合按钮。下载成功后，会显示“加载成功”，表示程序已正确下载，起始运行语句下会有一条绿色的背景。

8. 将输出去切换到调试窗口，使用D0000：3000命令查看内存3000H起始地址的数据没如图所示。

9.最后点击运行程序。

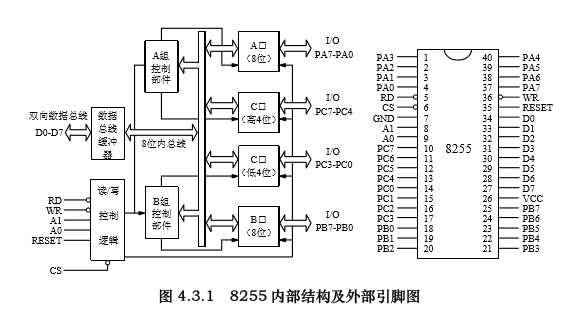
实验一 代码实现

|  |
| --- |
| SSTACK SEGMENT STACK ; 定义堆栈段 |
|  | DW 32 DUP(?) |
|  | SSTACK ENDS |
|  |  |
|  | CODE SEGMENT |
|  | ASSUME CS:CODE, SS:SSTACK |
|  | START: |
|  | PUSH DS |
|  | XOR AX, AX |
|  | MOV DS, AX |
|  | MOV SI, 3000H ; 建立数据起始地址 |
|  | MOV CX, 16 ; 循环次数16次 |
|  | AA1: |
|  | MOV [SI], AL |
|  | INC SI ; 地址自加1 |
|  | INC AL ; 数据自加1 |
|  | LOOP AA1 |
|  | MOV AX, 4C00H |
|  | INT 21H ; EXIT |
|  | CODE ENDS |
|  | END START |

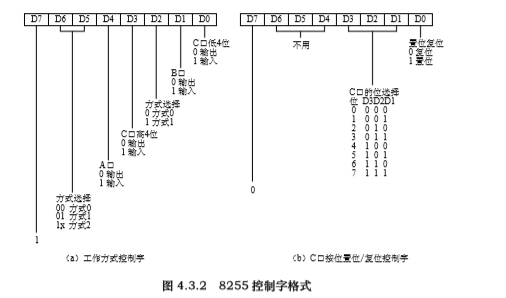
实验二 代码实现

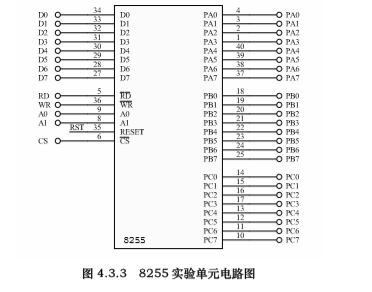
|  |  |
| --- | --- |
|  | DATA SEGMENT |
|  | STR1 DB "NO.4 E1714049 Z41714016$" |
|  | BUFFER DB 17 |
|  | NUM\_BUFFER DB ? |
|  | INPUT DB 16 DUP(?) |
|  | DATA ENDS |
|  |  |
|  | CODE SEGMENT |
|  | ASSUME CS:CODE, DS:DATA |
|  | START: |
|  | MOV AX, DATA |
|  | MOV DS, AX |
|  | ; PRINT EXISTING STRING |
|  | LEA DX, STR1 |
|  | MOV AH, 09H |
|  | INT 21H |
|  | ; INPUT A STRING AND PRINT IT |
|  | LEA DX, BUFFER |
|  | MOV AH, 0AH |
|  | INT 21H |
|  | LEA DX, INPUT |
|  | MOV AH, 09H |
|  | INT 21H |
|  | LEA DX, STR1 |
|  | MOV AH, 09H |
|  | INT 21H |
|  | MOV AH,4CH |
|  | INT 21H |
|  | CODE ENDS |
|  | END START |
|  |  |

**实验三**

****

并行接口是以数据的字节为单位与I/O首个被或被控制对象之间传递信息。CPU与接口之间的数据传送是并行的，即可以同时传递8位、16位、或32位等。8255可编程外围接口芯片是INTERNET公司生产的通用并行I/O接口芯片，它具有A,B,C三个并行接口，用+5V单电源供电，能在以下三种方式下工作：方式0--基本输入/输出方式、方式1--选通输入/输出方式、方式2--双向选通工作方式。8255的内部结构及引脚如图4.3.1所示，8255工作方式控制字和C口按位置置位/复位控制字格式如图4.3.2所示。





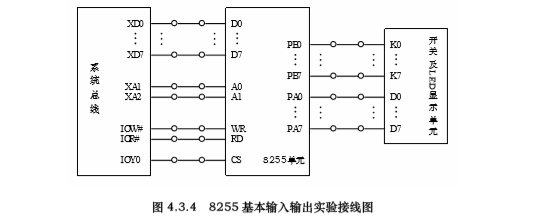
**实验步骤**：

1. 基本输入输出实验

本实验使用8255端口A工作在方式0并作为输出口，端口B工作在方式0并作为输入口。用一组开关信号接入端口B，端口A输出线接至一组数据灯上，然后通过对8255芯片编程来实现输入输出功能。j

1. 实验接线如题4.3.4所示，先按图连接实验线路图（第一次忘记拍照纪念，不过做完后已经经过助教检查）。
2. 编写实验程序（经编译无误后装入系统）。
3. 运行程序，改动拨动开关，同时观察LED显示，验证程序功能。
4. 点击“调试”下拉菜单中的“固化程序“项，将程序固化到系统存储器中。
5. 将386EX单板机系统的短路跳线JDBG的短路接到RUN端，然后按复位按键，观察程序是否正常运行；关闭实验箱电源，稍等后再次打开电源，看固化的程序是否运行，验证程序功能。
6. 实验完毕后，将短路跳线JDBG的短路块接到DBG端。

小提示：I386EX CPU单板机支持联机调试模式和脱机独立运行模式。两种模式的切换时同通过I386EX CPU单板机单元的右下角下层基板处的短路条线JDBG来实现。短路块短接到DBG档，CPU与软件处于联机模式，该模式下，通过软件界面可对CPU进行下载程序，单步、断点、连续运行等调试，通过固化功能菜单，可将加载到CPU单板机存储器中的程序固化到FLASH存储器中。固化完成后，将短路块短接到RUN档，并复位或另加电，CPU将启动FLASH存储器中的程序进行独立运行，此时I386EX CPU单板机就工作在脱机独立于运行模式了。



实验三 代码实现：

|  |
| --- |
| IOY0 EQU 0600H |
|  | MY8255\_A EQU IOY0+00H\*2 |
|  | MY8255\_B EQU IOY0+01H\*2 |
|  | MY8255\_C EQU IOY0+02H\*2 |
|  | MY8255\_MODE EQU IOY0+03H\*2 |
|  |  |
|  | SSTACK SEGMENT STACK |
|  | DW 32 DUP(?) |
|  | SSTACK ENDS |
|  |  |
|  | CODE SEGMENT |
|  | ASSUME CS:CODE |
|  | START: |
|  | MOV DX, MY8255\_MODE |
|  | MOV AL, 80H |
|  | OUT DX, AL |
|  | MOV BX, 8005H |
|  | AA1: |
|  | MOV DX, MY8255\_A |
|  | MOV AL, BL |
|  | OUT DX, AL |
|  |  |
|  | MOV DX, MY8255\_B |
|  | MOV AL, BL |
|  | OUT DX, AL |
|  |  |
|  | ROL BL, 1 |
|  |  |
|  | CALL DELAY |
|  | CALL DELAY |
|  | JMP AA1 |
|  | DELAY: |
|  | PUSH CX |
|  | MOV CX, 0F000H |
|  | AA2: |
|  | PUSH AX |
|  | POP AX |
|  | LOOP AA2 |
|  | POP CX |
|  | RET |
|  | CODE ENDS |
|  | END START |

【小结或讨论】

1. 在最终结果与预期不相符时，应首先检查实验箱接线是否正确，其次再检查代码是否逻辑上存在错误；
2. 在实验三流水灯实验中，传给 BX 的值不同（即MOV BX, 8005H），最后流水灯的显示方式也就会随之不同。