四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 微机系统与接口技术 | 实验课时 | 4 |
| 实验项目 | 静态存储器扩展实验 | 实验时间 | 2023.5.6 |
| 实验目的 | 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。 | | |
| 实验环境 | Wdm86 | | |
| 实验内容（算法、程序、步骤和方法） | 【基本实验要求】  1.编写实验程序，将0000H ～ 000FH共16个数写入SRAM的从0000H起始的一段空间中，然后通过系统命令查看该存储空间，检测写入的数据是否正确。2.改变实验程序，按非规则字写存储器，观察实验结果。3.改变实验程序，按字节方式写存储器，观察实验结果。  【扩展实验要求】  将学号写入静态存储器中，然后通过D命令查看写入的数据。（注意高位和地位的顺序。）  【实验原理】  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 的每一个触发器是由 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（2K×8 位），6264（8K×8 位）和 62256（32K×8 位）。  本实验平台上选用的是62256，两片组成 32K×16 位的形式，共 64K 字节。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。处理器访问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。  【实验步骤】   1. 实验接线图如下，按图接线。   C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps3.jpg   1. 编写实验程序，经编译、链接无误后装入系统。   代码如下：  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  START PROC FAR  *ASSUME* CS:CODE  *MOV*    AX, 8000H    *; 存储器扩展空间段地址*  *MOV*    DS, AX      AA0:  *MOV*    SI, 0000H    *; 数据首地址*  *MOV*    CX, 0010H  *MOV*    AX, 0000H      AA1:  *MOV*    [SI], AX  *INC*    AX  *INC*    SI  *INC*    SI  *LOOP*   AA1  *MOV*    AX,4C00H  *INT*    21H          *;程序终止*  START ENDP  CODE ENDS  END START  3. 先运行程序，待程序运行停止。  4. 通过 D 命令查看写入存储器中的数据：  D8000：0000 回车，即可看到存储器中的数据，应为 0000、0001、0002、…、000F  共 16 个字。  5. 改变实验程序，按非规则字写存储器，观察实验结果。  代码如下：  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  START PROC FAR  *ASSUME* CS:CODE  *MOV*    AX, 8000H    *; 存储器扩展空间段地址*  *MOV*    DS, AX      AA0:  *MOV*    SI, 0001H    *; 数据首地址*  *MOV*    CX, 0010H  *MOV*    AX, 0000H      AA1:  *MOV*    [SI], AX  *INC*    AX  *INC*    SI  *INC*    SI  *LOOP*   AA1  *MOV*    AX,4C00H  *INT*    21H          *;程序终止*  START ENDP  CODE ENDS  END START  6. 改变实验程序，按字节方式写存储器，观察实验现象。  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  START PROC FAR  *ASSUME* CS:CODE  *MOV*    AX, 8000H    *; 存储器扩展空间段地址*  *MOV*    DS, AX      AA0:  *MOV*    SI, 0000H    *; 数据首地址*  *MOV*    CX, 0010H  *MOV*    AX, 0000H      AA1:  *MOV*    [SI], AX  *INC*    AX  *INC*    SI  *LOOP*   AA1  *MOV*    AX,4C00H  *INT*    21H          *;程序终止*  START ENDP  CODE ENDS  END START  【扩展实验】  连线图：    代码：  DATA SEGMENT      ID   DB '2021141440374','$'  DATA ENDS  CODE SEGMENT      START:                 PROC   FAR  *ASSUME* CS:CODE,DS:DATA  *MOV*    AX, 8000H          *; 存储器扩展空间段地址*  *MOV*    ES, AX  *MOV*    AX, DATA           *; 存储器扩展空间段地址*  *MOV*    DS, AX  *MOV*    SI, OFFSET ID      *; 数据首地址*  *MOV*    DI, 0000H      LOOP\_START:  *MOV*    AL,[SI]  *CMP*    AL,'$'  *JZ*     END\_LOOP  *SUB*    AL,30H  *MOV*    ES:[DI],AL  *INC*    SI  *INC*    DI  *JMP*    LOOP\_START      END\_LOOP:  *MOV*    AX,4C00H  *INT*    21H                *;程序终止*                 ENDP  CODE ENDS  END START  代码思路：   1. 初始化内存段寄存器。 2. 设置源数据和目标内存位置。 3. 循环读取字符串中的每个字符，跳过字符串结束符$。 4. 将字符转换为数值（减去ASCII码中数字字符的偏移量）。 5. 将转换后的数值存储到扩展内存段。 6. 循环直到遇到结束符$。 7. 程序结束。 | | |
| 数据记录  和计算 | 【基础实验结果】   1. 按规则字在存储空间中写入16个数，然后通过-d 8000:0000指令查看结果是否正确      1. 按非规则字进行存储      1. 按字节进行存储     【扩展实验结果】   1. 输出学号 | | |
| 结 论  （结 果） | 成功完成了三个基础编程实验和一项扩展实验。基础实验包括：   1. 编写程序，将数值0到15连续写入SRAM的0000H起始地址区域。 2. 使用系统命令检查存储区域，以验证数据写入是否准确。 3. 修改程序，实现非规则字的存储器写入。 4. 再次调整程序，以字节为单位进行存储器写入。   对于扩展实验，目标是将学号写入静态存储器，并验证数据。我采用了直接输出的方法：   1. 将13位的学号分别从字符转为数字。 2. 依次将每个数字写入存储器。 3. 访问存储器，按顺序读取并输出，以验证学号的正确性。 | | |
| 小 结 | 因为没区分清楚大端模式和小端模式，所以在规则字和非规则字上耽误了很多时间，在查阅相关资料后成功解决了问题。  通过这系列实验，我不仅掌握了汇编语言编程的基本技巧，还深入理解了存储器地址的构建原理。此外，我对存储器的扩展技术、数据的读写操作以及CPU如何访问存储器有了更深刻的认识。这些实验经历极大地加深了我对计算机硬件与软件交互方式的理解。 | | |
| 指导老师评 议 | 成绩评定： 指导教师签名： | | |