

微机系统与接口

Leo

2024 年 5 月 6 日

目录

第一章 汇编语言	1
1.1 基本概念	1
1.2 表达式	1
1.3 伪指令	2
1.4 DOS 功能调用	2
1.4.1 键盘功能调用	2
1.4.2 显示功能调用	3
1.4.3 打印功能调用	4
1.4.4 日期时间调用	4
1.4.5 返回操作系统	5
第二章 中断	6
2.1 8086 中断系统	6
2.1.1 中断概述	6
2.1.2 中断源	6
2.1.3 硬件中断	6
2.1.4 软件中断	6
2.1.5 中断矢量表	7
2.1.6 响应过程	7
2.2 8259A 基本原理	7
2.2.1 引脚功能	7
2.2.2 工作方式	8
第三章 并行接口芯片 8255A	10
3.1 引脚简介	10
3.2 内部结构	10
3.3 控制字	10
3.4 工作方式	11

第一章 汇编语言

1.1 基本概念

汇编语言程序：用指令的助记符、符号地址、标号、伪指令等汇编语言编写的程序称为源程序，用.asm 文件表示。源程序要经过汇编程序翻译为机器能认识的二进制码.exe 文件后才能让机器执行。

基本流程：编辑程序-汇编程序-连接程序-得到可执行文件

汇编语言源程序格式：

1. 指令性语句：“标号：指令助记符操作数；注释”
2. 指示性语句：名字伪指令助记符；注释

1.2 表达式

表达式由运算对象和运算符组成，运算结果做操作数使用，运算对象可以是常数、变量或标号。常用运算符如下表

类型	运算符和运算结果	举例
算数运算符	+-* /MOD,SHL,SHR	MOV AL,32/5, 汇编成 MOV AL,6
逻辑运算符	AND,OR,XOR,NOT	MOV AL,30H OR,05H 汇编成 MOV AL,35H
关系运算符	EQ,NE,LT,GT,LE,GE	MOV AX,15 EQ 0FH 汇编成 MOV AX,0FFFFH(因为 15 等于 0FH)
数值返回	OFFSET,SEG,TYPE(返回字节数),LENGTH(返回单元数),SIZE(返回总字节数)	
修改属性	段超越前缀, PTR 修改类型属性, HIGH/LOW 分离高低字节, SHORT 短转移	HIGH 1234=12H
其他	圆括号改变优先级, 方括号间接寻址或下标	

1.3 伪指令

伪指令为指示性语句，没有对应的机器码，不占用内存单元，仅起说明作用。这里列出了常用的伪指令。

伪指令语句	说明	举例
等值伪指令 EQU	用于给常数、变量和表达式定义一个符号名	COUNT EQU 100
段定义语句 SEGMENT...ENDS		
段分配语句 ASSUME	任何时候都只允许四个段同时存在，该语句用来确定哪些段做 DS, CS, ES, SS。CS 必须要有	
过程定义语句 PROC...ENDP	定义一些子程序供主程序调用	PRO1 PROC FAR ... RET PRO1 ENDP
程序结束语句 END	放在源程序最后一行，汇编程序遇到 END 语句就停止进行	

1.4 DOS 功能调用

DOS 相当于一些提前写好的子程序，让用户可以方便地与设备和接口打交道。
调用方法：

- 1. 将功能号放入 AH 中
- 2. 入口参数放入指定寄存器中（可选）
- 3. 执行 INT 21H
- 4. 调用结束后得到出口参数，或显示在 CRT 上

1.4.1 键盘功能调用

功能号 =1、6、7、8

等待键盘，有键压下时，将该键的 ASCII 码存入 AL。区别在于

- 1. 1,6 号功能调用会回显该字符；7,8 号不回显
- 2. 1 号功能调用还要检查是否按下 ctrl+break 键，若是，则执行 INT 23H 中止程序。

功能号 =0AH

键入字符串，将其存入显示缓冲区。该缓冲区由 DS:DX 指向。

```
1  DATA SEGMENT ;定义显示缓冲区
2  BUF DB 50 ;最大字节数为50
3      DB ?;用来存储实际键入的字节数
4      DB 50 DUP(?);存键入字符的ASCII码
5  DATA ENDS
6  ...
7  CODE SEGMENT
8  MOV AX,DATA
9  MOV DS,AX ;DS指向缓冲区段址
10 MOV DX,OFFSET BUF ;DX指向缓冲区偏移地址
11 MOV AH,0AH;调用DOS功能
12 INT 21H
13 ...
14 CODE ENDS
15 ;缓冲区的信息为32 实际键入数 键入字符的ASCII码 0D（回车符） 0 0 0 0（后续
    未用到的地方填0）
```

1.4.2 显示功能调用

功能号 =2

显示单个字符，入口参数为 DL，其中存储要显示的字符的 ASCII 码

```
1  MOV DL,0EAH
2  MOV AH,2
3  INT 21H
```

功能号 =9

显示以 \$ 为结尾的字符串（\$ 符号不显示），字符串起始地址为 DS:DX，作为入口参数。

```
1  DATA SEGMENT
2  MESS DB 'TRY AGAIN! ',0DH,0AH,'$'
3  DATA ENDS
4  CODE SEGMENT
5  ...
6  MOV SEG MESS
7  MOV DS,AX
8  MOV DX,OFFSET MESS
9  MOV AH,9
10 INT 21H
11 ...
12 CODE ENDS
```

1.4.3 打印功能调用

功能号 =5，入口参数为 DL，里面存放要打印的字符

```
1 PRNT DB 'YOU ARE RIGHT!'  
2 COUNT EQU $-PRNT;表示将当前地址的值减去PRNT的地址，实际上就是PRNT的长度  
3     ...  
4     MOV CX,COUNT  
5     MOV BX,0  
6 NEXT: MOV AH,5  
7     MOV DL,PRNT[BX]  
8     INT 21H  
9     INC BX  
10    LOOP NEXT
```

1.4.4 日期时间调用

功能号 =2AH

取日期，不要入口参数，返回的日期放在下述位置

1. CX: 年号 (1980-2099), 二进制
2. DH: 月号
3. DL: 日
4. AL: 星期

功能号 =2BH

设置日期。调用前 CX, DX 必须存有一个有效日期。调用后，若设置日期有效，则 AL=00，若设置日期无效，则 AL=0FFH.

```
1     MOV CX,1999;1999年(存为3CFH)  
2     MOV DH,04;4月  
3     MOV DL,15;15日  
4     MOV AH,2BH  
5     INT 21H
```

功能号 =2CH

取时间，不要入口参数，返回值在 CX, DX 中

1. CH: 小时
2. CL: 分钟

3. DH: 秒

4. DL:1/100 秒

功能号 =2DH

设置时间。调用前 CX 和 DX 中存放需设置的时间值，格式同 2CH 号调用。调用后 AL 的取值也同 2CH 号调用。

```
1  MOV CX,0824H;时间8:36
2  MOV DX,1532H;21.5秒(15H=21,32H=50),50/100=0.5秒
3  MOV AH,2DH
4  INT 21H
```

1.4.5 返回操作系统

功能号 4CH，调用时可以强行返回 DOS。一般用于程序末尾。

第二章 中断

2.1 8086 中断系统

2.1.1 中断概述

CPU 执行正常的程序时，由于某些事件产生，暂时中止当前程序的执行，转去执行为时间服务的“中断服务程序”，然后返回原程序继续执行，这个过程称为“中断”。中断的基本概念包括中断源、中断屏蔽、中断嵌套、中断矢量表和中断优先级。

2.1.2 中断源

$$\text{中断源} \left\{ \begin{array}{l} \text{硬件中断（外部中断）} \left\{ \begin{array}{l} NMI \\ INTR \end{array} \right. \\ \text{软件中断（内部中断）} \left\{ \begin{array}{l} INT\ n \\ INTO \\ Division\ error \\ IF = 1 \end{array} \right. \end{array} \right. \quad (2.1)$$

2.1.3 硬件中断

从 8086/8088 的 NMI 引脚引入的为不可屏蔽中断，如奇偶校验错或电源掉电。

从 INTR 引脚引入的为可屏蔽中断，中断源接到 8259A 的 IR0-IR7 引脚，经过 8259A 判优之后，选择优先级最高的中断向 CPU 提出中断请求。

2.1.4 软件中断

INT n(n=00-FFH) 以指令形式安排在程序中。

INTO 溢出标志 OF 为 1 时，若再执行 INTO 指令，则自动产生溢出中断。

除法错中断：当除数为 0 或商超过寄存器的最大容纳值时产生该中断。

单步中断：IF=1 时，CPU 每执行一条指令后都产生单步中断，将各寄存器和有关寄存器的内容显示在 CRT 上。

2.1.5 中断矢量表

1. 中断矢量表用于存放中断服务程序的入口地址
2. 共有 256 种中断
3. 每种中断的入口地址 (CS:IP) 占 4 个内存单元
4. 共占用 1K 单元, 位于 000-3FFH 范围内

中断矢量表的格式如图!!!

2.1.6 响应过程

中断响应流程图如图所示。!!! 其中, 中断处理的程序流程为

1. 标志寄存器入栈
2. IF——>TEMP
3. 清除 IF 和 TF: 清除 IF 禁止其他可屏蔽中断进入, 清除 TF 禁止单步运行行模式。
4. CS 和 IP 入栈
5. 进入中断处理程序
6. 检查 NMI, 若有, 重新第一步, 若无, STI 打开中断, 允许其他中断进入
7. 检查 TEMP=1? 若是, 重新第 1 步, 若否, 执行中断处理程序
8. 用 IRET 弹出 IP、CS、标志寄存器
9. 返回断点

其中进入中断处理程序前都是由硬件自动完成的。中断可以嵌套, 优先级从高到低为 IR0, ... IR7.

2.2 8259A 基本原理

可编程中断控制器, 允许 8 级硬件中断, 级联后可以扩展到 64 级, 每一级都可以允许或屏蔽中断。判别优先级后, 选择高级中断类型号送给 CPU

2.2.1 引脚功能

1. $D_0 - D_7$: 数据总线, 与系统 DB 或通过总线驱动器与系统 DB 连接
2. $\overline{CS}, \overline{WR}, \overline{RD}$: 片选、写、读信号
3. \overline{INTA} : 中断响应信号。CPU 响应 INTR 中断后, 发出两个连续的 \overline{INTA} 信号, 第一个表示 CPU 已经开始处理, 第二个发来后, 8259 从 DB 上送出中断类型号。

H	\overline{CS}	\overline{WR}	\overline{RD}	A_0	读写功能
	0	1	0	0	CPU 往 8259 写 ICW1, OCW2, OCW3
	0	1	0	1	CPU 往 8259 写 ICW2, ICW3, ICW4, OCW1
	0	0	1	0	IRR/ISR 向 CPU 传送数据
	0	0	1	1	IMR 向 CPU 传送数据
	1	x	x	x	高阻态
	x	1	1	x	高阻态

4. A_0 : 端口选择线, 0 表示选中偶地址
5. $CAS_0 - CAS_2$: 级联信号, 主片的 $CAS_0 - CAS_2$ 做输出信号, 从片的 $CAS_0 - CAS_2$ 做输入信号
6. $\overline{SP}/\overline{EN}$: 从设备编程/允许缓冲线
7. $IR_7 - IR_0$: 中断设备的中断请求信号输入端, 可以是电平也可以是上边沿触发

8259A 内部结构框图!!!

1. 数据总线缓冲器: 8 位双向三态缓冲器, 传送命令字 (写入)、状态字 (读出)、中断向量字 (读出)
2. 控制电路: 有一组初始化命令字寄存器和一组操作命令字寄存器, 按编程方式管理 8259A 工作
3. 读写控制电路:
4. 级联缓冲器/比较器
5. 中断服务寄存器 ISR: 存放正在处理中的中断请求信号。任一级中断被响应时, 相应的 IS_n 位置为 1, 中断处理结束后清零。多重中断时可能会多位置一
6. 优先权判别器 PR: 对 IRR 寄存器的中断请求进行判定, 让优先级最高的中断服务送到 ISR 中
7. 中断请求寄存器 IRR: 存放 $IR_7 - IR_0$ 上的中断请求信号, 哪个引脚上有请求, 相应位就置一。电平触发还是前边沿触发由编程确定。
8. 中断屏蔽寄存器 IMR: 用软件使某位置一, 则禁止对应位的中断请求信号进入 PR 和 ISR

2.2.2 工作方式

设置优先级的方式

1. 完全嵌套方式: 优先级固定, 禁止同级中断
2. 特殊全嵌套方式: 优先级固定, IR_0 最高, 允许同级中断

3. 优先级自动循环方式：某设备服务后，优先级排到最后，将最高权利赋给比他低一级的中断，初始 IR_0 最高
4. 优先级特殊循环方式：同上，最低优先级由编程决定，如可以编程让 $IR_4 > \dots > IR_7 > IR_0 > \dots > IR_3$

屏蔽中断源的方式

1. 普通屏蔽：用 OCW1 对 IMR 置位为 1 来实现
2. 特殊屏蔽方式：先通过 OCW3 使其工作在特殊屏蔽方式，再用 OCW1 使本级中断的屏蔽寄存器的相应位置一，仅屏蔽当前正在处理的这级中断本身，允许高级或低级中断进入

结束中断的方式

1. 自动结束中断 AEOI：收到第二个 \overline{INTA} 信号后沿，发中断结束信号使 $IS_n = 0$ ，仅用于单片 8259A 场合。
2. 一般中断结束命令 EOI：收到 CPU 发来的 EOI 命令后，将 ISR 中优先级最高的 IS_n 复位，适用于完全嵌套方式
3. 特殊中断结束命令 SEOI：ISR 不知道哪一级中断是最后处理的，需要 OCW2 的 L2-L0 字段来指出对哪个 IS_n 复位。适用于非完全嵌套方式（优先级循环方式）

系统总线的连接方式

1. 缓冲方式：8259A 需要总线驱动器才能与 DB 连接
2. 非缓冲方式：适用于单片或少量级联的系统

中断查询方式

中断查询方式通常用在 64 级中断的场合。用 OCW3 使 $P=1$ ，可以发出查询命令，再执行 IN 指令，读入状态字，了解中断状态。状态字格式为“ $IxxxW_1W_2W_3$ ”， $I=0$ 表示无中断命令，3 位 W 给出优先级最高的设备的编码，并使相应的 $IS_n = 1$

第三章 并行接口芯片 8255A

通用可编程并行 IO 接口芯片，内部有 ABC3 个数据端口，可工作于三种方式

3.1 引脚简介

1. $D_0 - D_7, \overline{CS}, \overline{WR}, \overline{RD}$: 数据位和片选、读写控制信号
2. A_1A_0 : 端口选择信号，从 0 到 3 分别表示选中 A、B、C 和控制字寄存器
3. $PA_0 - PA_7$: A 口输入输出引线，与外设相连传送数据信息
4. $PB_0 - PB_7$: B 口输入输出引线，与外设相连传送数据信息
5. $PC_0 - PC_7$: C 口输入输出引线，与外设相连传送数据信息, 可分为两组使用

3.2 内部结构

如图所示!!!

3.3 控制字

控制字分为方式选择字和置位复位字。方式选择字定义各端口的工作方式，置位复位字使 C 口的任一位置位或复位。

方式选择字的格式

1. D7=1: 标志位，表示本字为方式选择字
2. D6D5: 指定 A 口方式，0 为方式 0，1 为方式 1，其余为方式 2
3. D4: 决定 A 口 I/O
4. D3: C 口高 4 位 I/O
5. D2: 决定 B 口方式: 0 为方式 0，1 为方式 1
6. D1: B 口 I/O
7. D0: C 口低四位 I/O

置位复位字格式

1. D7=0: 标志位
2. D6D5D4: 任意, 通常置为 0
3. D3D2D1: 用二进制数的数值来选中 $PC_0 - PC_7$
4. D0: 指示所选中位想要置为 1 或 0

1	<code>MOV AL,00000100B</code>
2	<code>OUT 63H,AL ;控制口地址为63H, 让PC2输出低电平</code>

3.4 工作方式

1. 方式 0: 基本输入输出方式: A、B、C、任意口都可以做输入输出, 输出信号能锁存, 输入不能锁存
2. 方式 1: 选通输入输出方式: A、B 做 8 位数据口输入输出, 但要在联络信号的控制下才能完成 IO。C 口 6 根先做联络信号, 余下两根传送数据。A、B 的 IO 数据均能锁存
3. 方式 2: 双向传送方式。仅 A 口可以工作在本方式, IO 均能锁存但不能同时进行。C 口的 $PC_7 - PC_3$ 做 A 口的联络控制信号, 余下 3 根用来 IO