

# 汇编复习整理

---

- 计算机基本知识
- 8086系统结构
- 指令系统
- 汇编程序设计
- 外设通信
- 可编程接口芯片
- 中断

## 计算机基本知识

---

- 微型计算机的组成

名称	功能
CPU（运算器+控制器）	取指令，简单逻辑运算，程序控制
存储器	存储程序和数据
外部设备（I/O接口）	外设与CPU之间的信息协调
总线	数据信息，地址信息，控制信息的通道

## 8086系统结构

---

### CPU内部结构

- 指令执行部件  
EU 从BIU的指令队列中取指令，执行指令
- 总线接口部件  
BIU 负责与外部存储器或I/O打交道

### 寄存器结构

- 通用寄存器（EU内）

寄存器	功能
AX	存放参加运算的结果和数据，I/O
BX	数据寄存器，内存偏移地址，基址寄存器
CX	数据寄存器，循环计数
DX	数据寄存器
SI	源变址寄存器，内存逻辑偏移地址，存数
DI	目标变址寄存器
BP	内存逻辑偏移地址
SP	栈顶逻辑偏移地址

- 段寄存器（BIU内）

寄存器	功能
DS	数据段寄存器
SS	堆栈段寄存器
ES	扩展段寄存器
CS	代码段寄存器

逻辑段： 64KB 寻址范围不超过64KB 因为8086所有寄存器都是**16**位寄存器

- 标志寄存器（EU内）

标志寄存器	功能
CF（进位）	最高位有进位或借位
PF（奇偶）	
AF（辅助进位）	
ZF（零）	ZF=1表示本次运算结果为0
SF（符号）	最高位为1，SF=0
OF（溢出）	运算结果溢出
IF（中断）	
DF（方向）	
TF（单步）	

# 指令系统

- 寻址方式

MOV 目的操作数，源操作数

操作数分为：存储器操作数，寄存器操作数，立即数

寻址方式	例子
立即寻址	MOV AX,34EAH
寄存器寻址	MOV BP,SP MOV AX,1234H
直接寻址	MOV AX,[3E4CH] MOV ES:[1234H],BL
寄存器间接寻址	MOV SI,61A8H MOV DX,[SI]
基址/变址寻址	MOV CX,36H[BX]
基址+变址寻址	
I/O端口寻址	IN AL,25H

- 常考指令

- 传送

- MOV
- PUSH POP
- XCHG
- IN OUT
- LEA

- 算术

- ADD INC DAA
- SUB DEC DAS
- INC DEC 不影响标志位
- CMP

- 位操作

- NOT 取反
- AND
- OR
- XOR
- SHL SAL 左移位
- SHR SAR 右移位
- ROL ROR 循环移位

- 程序转移

- JE/JZ JNE/JNZ
- ZF标志位 一般判断cmp之后是不是相等
- JG JGE

判断是否大于

- JL JLE

判断是否小于

## 汇编程序设计

### 显示字符串

DOS功能调用：

调用号送至AH寄存器

中断指令 INT 21H

功能	调用号
单个字符输出	2号
多个字符输出	9号
过程终止	4ch号
单字符输入	1号 AL=ASCII码

- 屏幕显示一行字符串（重点）

```
;DS:DX指向欲显示字符串的首址
DATA SEGMENT
BUFF DB 'HELLOWORLD!$' ;必须以$结尾
DATA ENDS
CODE SEGMENT
    ASSUME DS: DATA , CS: CODE
START:
    MOV AX, DATA
    MOV DS, AX
    LEA DX, BUFF ;获得有效地址
    #或
    MOV DX, OFFSET BUFF ;取标号或变量的距段首址的偏移量
    MOV AH, 9
    INT 21H
    MOV AH, 4CH
    INT 21H
CODE ENDS
    END START
;OFFSET只能取得用"数据定义伪指令"定义的变量的有效地址,不能取得一般操作数的有效地址
```

- 显示字符

```

# 光标显示
MOV DL, 'A'
MOV AH, 2
INT 21H
#####
# 键盘输入
MOV AH, 1
INT 21H
MOV [2000H], AL

```

- 显示数据

```

# 以23为例 即00100011
MOV DL, BL
MOV AL, 4
SHR DL, AL #00000010
ADD DL, 30H #32H, 先显示高四位
MOV AH, 02H
INT 21H
MOV DL, BL
AND DL, 0FH #00000011
OR DL, 30H #33H
MOV AH, 02H
INT 21H

```

## 循环操作

- 数据段的定义

```

MOV AX, BUFF = MOV AX, [0000]
MOV BX, BUFF+2 = MOV BX, [0002]
SUM DW ?, ? ;保留四个字节内存单元
MAX DB 0,0,0,0,0 = MAX DB 5 DUP (0)

```

- 冒泡排序

```

DATA SEGMENT
BUFF DB 5,3,-7,-9,2,23,55,88
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    MOV DS,AX
    MOV CX,7
LOOP1:

```

```

MOV DX,CX
MOV SI,0
LOOP2:
MOV AL,BUFF[SI]
CMP AL,BUFF[SI+1]
JLE CONTINUE;从小到大 从大到小改成 JGE
XCHG AL,BUFF[SI+1]
CONTINUE:
INC SI
LOOP LOOP2
MOV CX,DX
LOOP LOOP1
MOV AH,4CH
INT 21H

```

- 最大值

```

#求数组里的最大值
DATA SEGMENT
BUFF DB 60 DUP(?)
MAX DB ?
DATA ENDS
CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    MOV SI,OFFSET BUFF ;取数组第一个数的地址
    MOV CX,60
    MOV AX,[SI]
LOOP1:
    CMP AX,[SI]
    JGE CONTINUE
    MOV AX,[SI]
CONTINUE:
    INC SI
    LOOP LOOP1
    MOV MAX,AX
    MOV AX,4C00H
    INT 21H
CODE ENDS
END START

```

- 累加

```

#大于等于60的数
DATA SEGMENT
BUFF DB 10 DUP(?)
SUM DB ?
DATA ENDS

```

```

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    MOV SI,OFFSET BUFF
    MOV CX,10
    MOV AX,60
    MOV DX,0;个数
LOOP1:
    CMP [SI],AX;
    JL CONTINUE;小于60
    INC DX
CONTINUE:
    INC SI
    LOOP LOOP1
    MOV SUM,DX
    MOV AX,4C00H
    INT 21H
CODE ENDS
    END START

```

## 画方框

```

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:
    MOV AL,13H;分辨率
    MOV AH,0;设定模式
    INT 10H

    MOV CX,10;X起始坐标
    MOV DX,10;Y坐标
    MOV BX,200;终点坐标
    MOV AL,0010B
    MOV AH,0CH
HENG:
    INT 10H
    INC CX
    CMP CX,BX
    JNE HENG

    MOV CX,10
    MOV DX,10
    MOV BX,200;终点坐标
SHU:
    INT 10H
    INC DX
    JNE SHU
    . . . . .

```

# 可编程接口芯片

- 初始化

```
# 8255
MOV DX, CTRL_PORT
MOV AL, 10010000B; a口输入b口输出
MOV AL, 10000000B; ab口都是输出
OUT DX, AL
# 8254
MOV DX, CTRL_PORT
MOV AL, 00110111; 通道00, 读写格式11, 工作方式011, 计数码1
MOV AX, 0000; 计数初值
OUT DX, AL
MOV AL, AH; 先输出低八位再输出高八位
OUT DX, AL
```

计数初值的设定:

$n = f * t$  是一个周期时间,  $f$  是脉冲波频率

## 外设通信

### 接口

接口是每个外设或每种外设与微处理器相连时使用的电路,接口上可以包含唯一的端口,也可以包含几个端口。

### 数码管

供阳极低电平有效

## CPU与外设数据传送的方式

### 1. 程序控制方式

- 无条件传送方式：
  - 优点：程序设计简单
  - 缺点：只能用于CPU与外设同步时，易出错
- 查询传送方式
  - 优点：接口电路简单
  - 缺点：CPU需和外设串行工作，效率不高

### 2. 中断传送方式

- 优点：相对提高了CPU利用率与实时性
- 缺点：接口电路较复杂，增加负载

### 3. DMA方式(直接存储器存取方式)

- 优点：传输数据不经过CPU，速度最快
- 缺点：硬件复杂



## io接口作用

1.数据寄存与缓冲 2.设备选择 3.信号转换 4.外设检查与控制 5.中断或dma管理 6.可编程

## 中断

---

- 中断处理的基本过程

- 中断请求

CPU设有两个中断请求输入引脚：可屏蔽中断请求输入引脚和不可屏蔽中断请求输入引脚。

- 中断判优

CPU必须找出中断优先级最高的中断源

- 软件方法

- 硬件方法

- 中断响应

- 中断服务

- 中断返回

- 中断的功能

- 故障处理

- 实时处理

- 分时处理

## AD转换的三种方式

---

- 计数式，速度慢，比较次数多
- 双积分，速度慢，精度高,抗干扰
- 逐次逼近，速度快，最常用