汇编语言教程

INTEL 8086/8088 80286
80386 8087 8089

适用机种 IBM-PC/XT PC-AT 5550
国产0520 0530及其兼容机

朱慧真 编著

国防工业出版社

汇 编 语 言 教 程

INTEL 8086/8088 80286 80386 8087 8089

适用机种 IBM-PC/XT、PC-AT 5550

国产0520 0530及其兼容机

朱慧真 编著

国防工业出版社

内容简介

本书以IBM-PC/XT和IBM-PC/AT机型为背景，着重讲述了INTEL 8086/8088和80286的宏汇编语言MASM，同时介绍了INTEL 8086/8088、8087、8089、80286、80386五个微处理器的指令系统。全书共分九章，分别介绍了8086/8088汇编语言程序设计的基本概念和基本方法、汇编语言的扩展、中断系统与控制性语句、输入输出程序设计、基本程序设计技巧以及80286与80386的扩充功能。

本书可作为大学计算机系学生使用的教材，也可作为学习汇编语言的自学读本。

汇编语言教程

INTEL 8086/8088 80286 80386 8087 8089

适用机种 IBM-PC/XT、PC—AT 5550 国产0520 0530及其兼容机

朱慧彦 编著

国防工业出版社出版 发行

（北京市海淀区西路老虎庙七号）

新华书店经售

蓟县印刷厂印装

787×1092 1/16 印张22 1/4 511千字

1988年10月第一版 1988年10月第一次印刷 印数：00,0001—005000册

ISBN 7-118-00432-4/TP.52 定价：10.75元

前　言

汇编语言是介于计算机能够直接理解的代码语言(又称机器语言)与使用者容易理解的高级语言之间的一种语言。它以助记符代替二进制码帮助使用者记忆和使用代码语言，所以又称为符号语言。由于汇编语言比高级语言更接近于代码语言，因而能透彻地反映计算机硬件的功能与特点，便于使用者灵活地根据自己的要求编制最为经济(省时间、省内存)的程序以及高级语言无法实现的程序，便于使用者编制实现各种控制和测量功能的程序。所以，尽管存在着多种高级语言，汇编语言作为一个强有力的软件工具仍不失它的重要性。

本书以IBM-PC／XT和IBM-PC/AT为背景机，着重讲述了INTEL 8086/8088以及80286的宏汇编语言MASM，同时，较完整地介绍了它涉及的INTEL 8086／8088、8087、8089、80286、80386五个微处理器的指令系统。考虑到作为中心处理器的8086／8088、80286、80386是一组功能递增的系列元件，在讲解上我们首先以8086／8088为基础，详细地介绍了与之相关的汇编语言，而后，再进一步介绍了80286、80386的扩展功能。这样，不仅便于初学者进行逐步加深的学习，而且便于已经了解8086／8088汇编语言的读者直接进入80286汇编语言的学习，同时对80386的有关性能也会有所了解。

本书第一章到第四章为基础部分，讲述了8086／8088程序设计的基本概念及方法；第五章讲述汇编语言的多段连接及宏扩展部分；第六章讲中断系统、控制性指令及系统功能模块的应用；第七章介绍I/O端口、I/O指令及IO程序，并讲解了MA89汇编语言。第八章讲软件设计中常用的查列表方法、浮动程序、再入式程序以及进程概念。第九章讲8086扩展成80286及80386后的扩充结构及功能，每章后都有足够多的练习题供读者上机实践使用。

本书是在编者多年为北京大学计算机科学技术系讲授《汇编语言》课程所编写的讲义基础上反复修改而成的。在讲解上，避免引入不必要的代码语言，而直接介绍汇编语言。全书自成体系，便于读者理解和掌握。

本书由中国科学院北京科海培训中心组织编写。

在本书的编写过程中，得到了北京大学计算机科学技术系杨芙清教授、许卓群副教授的热情指导和帮助。北京大学计算机科学技术研究所的俞士汶副教授对原稿进行了认真的校对并提出了许多宝贵意见，编者谨在此表示衷心的感谢！

最后，请各位专家及广大读者批评指正。

编者

一九八八年四月于北京大学

目　　录

第一章 预备知识

§1.1 基本概念

汇编语言 (assembler language/assembly language)是一种面向机器的程序设计语言，它是一种初级语言，它的基本内容是机器语言符号化的描述。通常汇编语言的执行语句与机器语言的指令是一对一的关系，即汇编语言的一个执行语句对应一条机器语言指令，反之亦然。

用汇编语言书写的程序称为汇编语言源程序或汇编语言程序。在本书中也简称为源程序。

汇编语言源程序也和其它高级语言的源程序一样，必须由翻译程序翻译成代码程序(即机器语言程序)方能在机器中执行。此代码程序称为相应源程序的目标程序。

汇编程序 (assembler/assembly program)是把源程序转变为相应目标程序的翻译程序。这个转变过程称为汇编 (assemble) 。

汇编语言与机器关系很密切，它通常是为特定的计算机或特定的计算机系列设计的。

§1.2 系统结构

1.2.1 iAPX86/88微处理器系列概况

iAPX86/88是以Intel 8086微处理器为基础发展起来的微处理器系列，其中的每一种都是由8086、8088、80186、80286等这些微处理器之一作为中心处理器 (CPU) ，并配以适当的支持芯片 (如8087、8089、80287、80130中的一片或数片) 构成具有各种扩展或增强功能的微处理器系统，表1.1给出了iAPX 86/88系列的构成例子。

8086 CPU是一个16位微处理器。由它联系与控制其它的处理器的运行。程序主要在该CPU上运行。下面对它将有进一步的介绍。

8088 CPU是一个准16位微处理器，它采用八位数据总线，而使用16位内部结构，它与8086的区别只是在存取16位数据时，内部处理有所不同，其它功能完全相同。对软件工作者来说，8088与8086除了运行速度上的差别外，其它方面没有区别。因此凡是本书围绕8086 CPU讲解的内容都适合8088 CPU。两者的指令系统相同，汇编语言一致。

8087是浮点运算处理器。8089是I/O处理器。对这两个处理器，下面也有进一步介绍。

80130是操作系统的固件。

80186是8086的增强型，增加了可靠性及速度，指令系统中有8条指令增强了功能，还增加了10条新指令。

80286除了具有8086的基本结构外，还集成了存储管理及虚地址保护机构。寻址容量

表 1.1 iAPX86/88微处理器系列的一些构成例子

器件名称 \ 类别	中心处理器（CPU）				支持芯片			
	8086	8088	80186	80286	8087	80287	8089	80130
iAPX86-10	1							
iAPX88-10		1						
iAPX86-11	1						1	
iAPX88-11		1					1	
iAPX86-20	1				1			
iAPX88-20		1			1			
iAPX86-21	1				1		1	
iAPX88-21		1			1		1	
iAPX86-30	1							1
iAPX88-30		1						1
iAPX186-10			1					
iAPX186-20			1		1			
iAPX186-30			1					1
iAPX286-10				1				
iAPX286-20				1		1		

达16兆字节。可提供给每个用户1000兆字节的虚存空间。它的指令系统包含了8086的全部指令，除增强了其中九种指令的功能外，还增加了二十五种新的指令，对此，本书第九章有详细讲解。

80287与8087一样，也是浮点运算处理器，它是与80286 CPU配套的协处理器。

此外，作为80286的增强型还有80386（表1.1没有列入），它是32位微处理器，虚存空间达到64T(MM)字节，运算速度也有大幅度的提高，与80286的指令系统向上兼容并增

加了三种新的指令（详见第九章）。

1.2.2 系统组成

相应iAPX86/88处理器系列可组成许多种微处理机系统。最简单的是由iAPX86-10或iAPX88-10组成的微处理机。IBM-PC和IBM-PC/XT分别属于iAPX88-10及iAPX88-20。它的系统组成可参看图1.1。

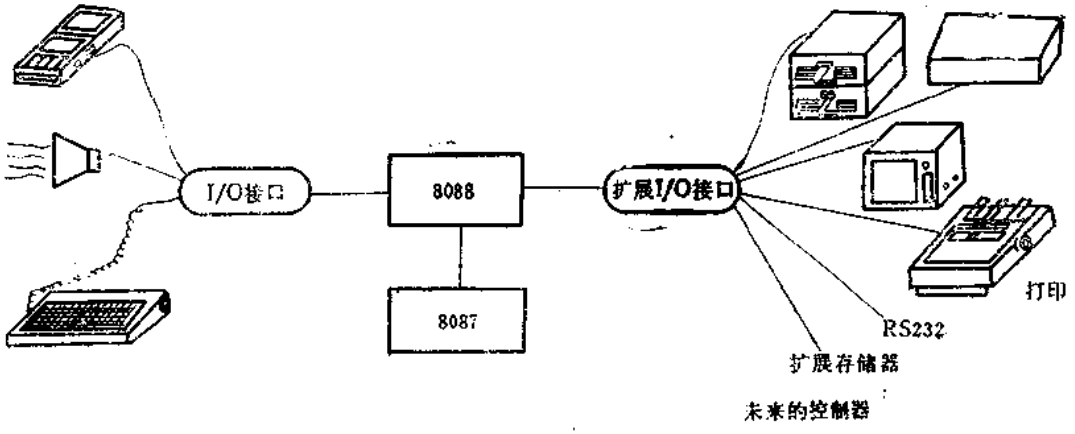

图 1.1 IBM-PC系统结构示意图

IBM-PC/XT系统板内存容量为64KB(16+48)到304KB(256+48)，加上扩充板的存储器容量后，整个系统内存可到1MB。通常它的外存储器是一台5¼吋软盘机和一台温彻斯特硬磁盘机。软盘存储容量为320KB或360KB，温盘存储容量为10MB或20MB。外部设备通常有键盘输入器、彩色图形终端及一台点阵打印机。

iAPX86-21处理器组成的微处理机，其系统组成可见图1.2。AI公司的AI-M16微处理机属于这种类型，它可以带多个终端。IBM-PC/AT微机属于iAPX286-20型，它也可以带多个终端。

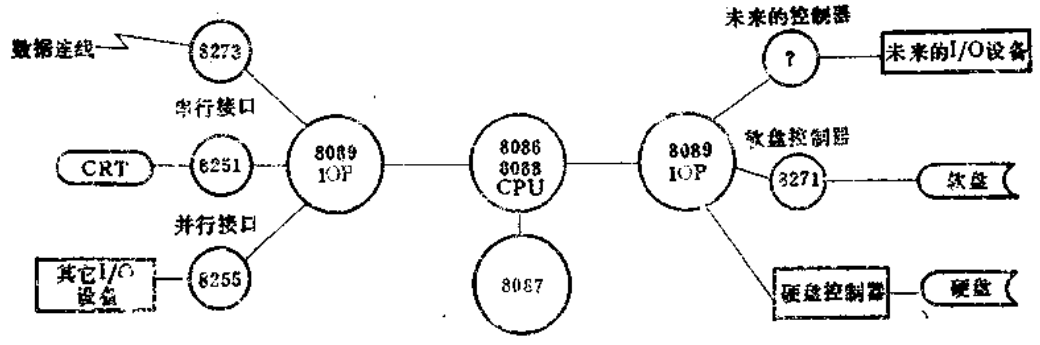

图 1.2 iAPX86-21型微处理机结构示意图

我国生产的长城0520-C微机与IBM-PC/XT微机兼容。0530微机与IBM-PC/AT微机兼容。

这些微机中能运行的软件很多，并且在不断地增加。目前在IBM-PC系列中运行的单

任务操作系统是MS-DOS(PC-DOS)及CCDOS。多任务操作系统是XENIX，后者只能在IBM-PC/AT及长城0530微机上运行。在AI-M16微机上运行的单任务操作系统是CP/M，多任务操作系统是MP/M。

这些微机系统能够提供使用的语言有很多种。通常的BASIC、COBOL、FORTRAN、PASCAL、LISP及C语言都能提供使用，汇编语言是ASM及MASM，还有True-BASIC及TurboPASCAL语言等。

以下将从软件角度介绍三个处理器：8086,8087及8089。

1.2.3 8086处理器

8086处理器是系统中心处理器（CPU），它负责联接和协调8087与8089处理器。

它有较完备的91种指令（其中，无浮点运算指令），有两级可达256种中断源的中断系统。操作系统在8086处理器上运行，其它程序也主要在8086处理器上运行。

8086处理器由两部分组成，一部分是执行单元EU，它负责执行指令；另一部分是总线接口单元BIU，它负责提供指令及数据，并负责形成地址和执行总线周期。它设立了一个指令队列排队器，可先行取6个字节的指令。图1.3为8086的结构图。

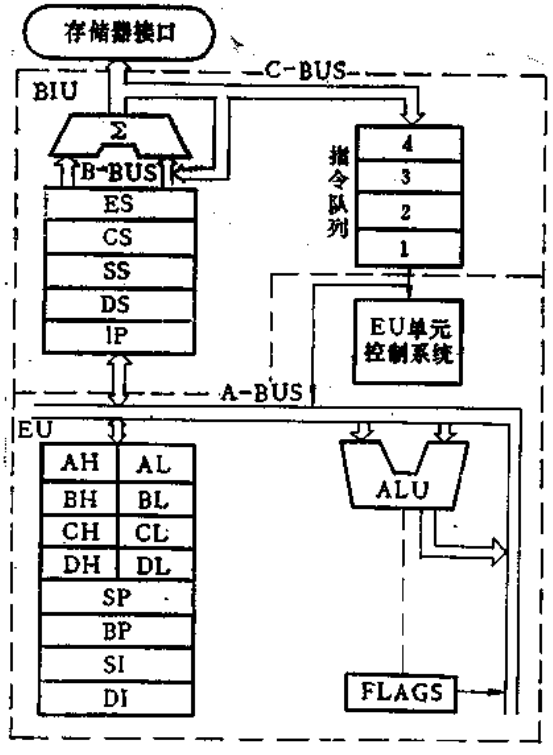

图 1.3 8086/8088 CPU结构图

8086处理器的内部寄存器如图1.4所示。它有4个16位的通用寄存器AX、BX、CX及DX、它们也可以做8个8位的寄存器使用。分别记为：AH、AL、BH、BL、CH、CL、DH及DL。16位寄存器还有指针寄存器BP与SP，变址寄存器SI与DI，指令指针寄存器IP，标志位寄存器FLAGS及四个段界寄存器CS、DS、ES及SS。其中AX～DI寄存器和

AH～DL寄存器都可作累加器使用。此外它们还有一些特殊用法，以后再讲。指令指针寄存器IP存放当前要执行的指令地址，因此它将随指令的执行自动地变化。标志寄存器FLAGS见图1.4，它是一个16位字寄存器，其内容标志出程序执行时CPU的状态。指令执行时，由于标志寄存器的内容不同（即程序状态的不同）将产生不同的功效，并且程序的状态将随指令的执行而变化。标志寄存器中的各位都有其专门的定义及作用。这里只作简单的介绍，以后将在涉及到的指令中进一步讲解。

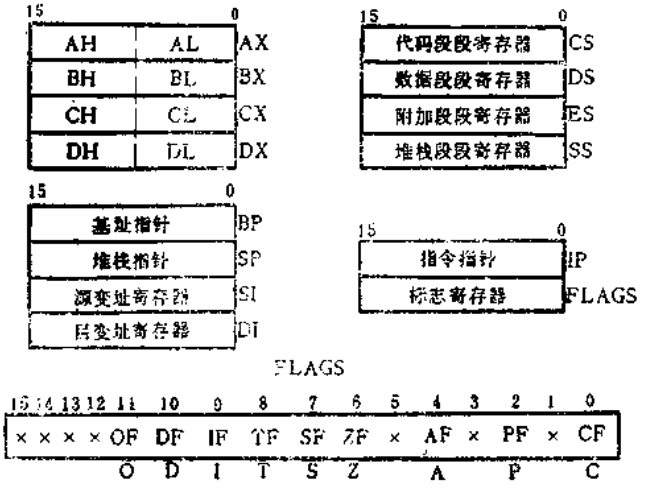

图 1.4 8086/8088内部寄存器

× ——无定义。

O——of溢出标志位，运算结果超过系统所能表示的数的范围时为溢出（或上溢）。例如：字节整数表示范围为－128～＋127，字整数表示范围为－32768～＋32767，溢出时，(of) =1。

D——df方向标志位，它标志字符串指令处理的方向。(df) =0是递增方向；(df)=1是递减方向。

I——if中断标志位，它表示系统可否接受中断（除不可屏蔽中断之外）。(if) =0屏蔽中断；(if)=1允许中断。

T——tf单步中断标志位。

S——sf符号标志位，它表示运算结果最高位（符号位）的值，(sf) =1结果为负数，(sf)=0结果为正数。

Z——zf全零标志位，运算结果为全零时，该位置“1”即(zf) = 1，否则(zf) =0

A——af辅助进位标志位，在十进制运算中，低四位产生进位或借位时，(af) =1，否则(af)=0

P——pf奇偶校验标志位，它用于核对ASCII码的奇偶性，(pf)=1时表示运算结果后八位中有偶数个“1”，否则(pf)=0(奇数个“1”)。

C——cf进位标志位，运算结果在最高位产生进位或借位时，(cf) =1。(字节的最高位为D_7；字的最高位为D_{15})，否则(cf)=0

四个段界寄存器，分别介绍如下：

CS——代码段寄存器，它存放当前执行程序段的界地址。

DS——数据段寄存器，它存放当前数据段的界地址。

SS——栈段寄存器，它存放当前栈段的界地址。

ES——附加段寄存器，它存放附加数据段的界地址。

1.2.4 存储器及其分配方式

8086可达1048576字节的内存。内存按字节编址，每个内存单元的地址为二进制20位。地址以16进制表示是从 00000到FFFFF。字长为16位，字地址最好从偶数地址开始。习惯上在书写时内存地址一律用十六进制表示。

一、物理地址与逻辑地址

物理地址是内存真实的地址，也称实际地址。它是一个20位的地址。存取数据时必须使用物理地址。但是在多道系统中物理地址是不能直接使用的，因为用户使用内存的情况是不可预测的，从而引入逻辑地址的概念。逻辑地址是用户定义的地址，一般逻辑地址是指系统分配给用户空间的相对地址。

二、主存储器的分段结构

8086的主存储器基本上是界地址分配方式，但没有越界检查。系统设立的四个段界寄存器专门用于存放界地址。

内存以块为分配单位，每段可分给若干个连续块，而块的大小可按下列公式推算：

$$块长 = 2^N$$

其中N为内存地址位数减界地址寄存器位数，因8086／8088内存地址位数为20，且其所有的寄存器都是十六位的，故N＝4，块长为16字节。1048576的内存可分为64K个块。块号从0000编到FFFF。

系统根据用户的需要，把连续的若干块组成的段分配给用户，各段的首块号即为该段的界地址，段界寄存器中存放的是当前段的界地址。由界地址变为相应的物理地址显然只要在其尾部加上一个十六进制的“0”。例如：下列是三个界地址与其相应的物理地址

界地址：	00FF	0001	0006
相应物理地址：	00FF0	00010	00060

其实段的界地址相应的物理地址就是该段的段始址。8086中的逻辑地址就是段内相对于段始址的相对地址，也称偏移地址。由于指令中使用的偏移地址最多只能是16位，故段最长可达64K字节。

三、逻辑地址与物理地址的转换

逻辑地址只有转换成物理地址才能访问内存，这个转换由硬件来完成。一般，此转换遵循下列公式：

逻辑地址的相应物理地址＝当前该段始址＋该逻辑地址

例如设程序段的段界地址为AAAA，这意味着在执行程序时CS＝AAAA。并设该段

中有逻辑地址为BBBB，要访问BBBB时，一定要求出BBBB相对应的物理地址，由于段始址是段界地址（CS的内容）对应的物理地址，这里CS = AAAA，故段始址为 AAAA0，则

BBBB的物理地址 = AAAA0 + BBBB

= B665B

因而程序中要访问的是BBBB，实际上访问的是B665B。

上例中只讲了访问指令时逻辑地址到物理地址的转换。对数据的访问也有类似的转换，只是使用的段界寄存器不同，可能是DS或ES或SS。

程序必须按段使用内存，程序中出现的地址（除直接段间转移指令外）一般都是逻辑地址，并且在执行时逻辑地址与其对应的段界寄存器应是明确的，而其段界寄存器的值也是确定的。

系统虽只有四个段界寄存器，但程序中可有任意个段，并且各段互相独立，可以不邻接，还可以互相重叠。只是当前参与运行的最多只能有四段。显然段界寄存器的内容软件可置，用户也可置。

这种内存分配方式是很灵活的，同时也扩展了使用空间，但是内存缺乏必要的保护，用户必须正确置段界寄存器的内容，否则后果严重。

四、堆栈

堆栈是一种后进先出的存储区。它常用于存放调用程序的现场及参数。有了堆栈，多重调用的层次包括递归调用的层次都是很清楚的。

8086的堆栈设在存储器中，并且由栈段界寄存器SS指示堆栈段的始址。栈指针寄存器SP指示栈段使用到的位置到栈段始址的偏移地址。堆栈的信息是从高地址到低地址的方向

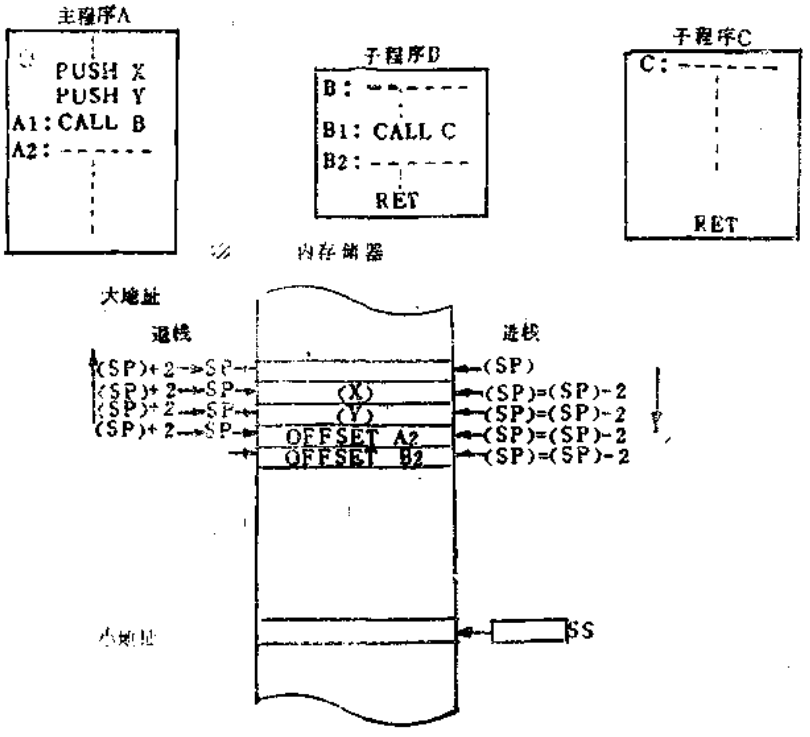

图 1.5 堆栈操作图示

存放。栈中每个元素为一个字长，堆栈按后进先出的原则退栈。进栈时(SP)－2→SP，退栈时 (SP)＋2→SP。一个栈段最长可达64K字节，超过64K字节，可再定义一个栈段。程序中可有若干个栈段。但只有一个当前栈段，只有当前栈段的段界地址及栈顶位置分别由寄存器SS及SP指示。图1.5为堆栈操作示意图。

1.2.5 8087处理器

8087是高速运算处理器，专门用于计算浮点数及长整数。它可以计算32位到64位的浮点数及整数，还可计算80位的压缩十进制数及一些初等函数。

它有8个浮点栈，这八个浮点栈由80位长的寄存器组成，记为st(n)，n＝0,1,2,…,7。它有62种指令，相应有77种汇编语句，运算绝大部分是在浮点栈上运行。系统中8087的指令和8086的指令可以在同一个程序中使用，8087的指令是不完备的、不能构成循环程序。它一定要和8086的指令配合使用，但它的浮点运算及整数运算速度快且精度高。

1.2.6 8089处理器

8089输入输出处理器是一个高性能通用I/O系统，它具有两个独立的I/O通道。每个通道除管理I/O之外，还具有CPU和DMA（直接存储器存取）控制器的作用。它有自己的64K字节的I/O存储器，并且还能使用系统内存。它本身具有50种指令的指令系统及相应的汇编语言MA89汇编程序，它能执行编程和实现数据的简单翻译与比较，它能独立地实现内存到内存或外设到外设的数据传送。但是这些功能体现在系统软件中，用户不必直接使用8089的指令，只把它看成是具有两个通道的控制器即可。

§1.3 数据表示

1.3.1 十六进制数

十六进制数是逢十六进一位，每一位有十六种状态，用0,1,2,…,9,A,B,C,D,E,F,表示。恰好四位二进制数可以代表一位十六进制数。表1.2是十进制、二进制与十六进制数码对照表。

表 1.2 十进制、二进制与十六进制数码对照表

十进制	0	1	2	3	4	5	6
十六进制	0	1	2	3	4	5	6
二进制	0000	0001	0010	0011	0100	0101	0110