

社 第 1 章 汇编语言基础知识（全）

2010-10-18 19:32:40| 分类： 答案集锦 阅读 785 评论 6 字号：大中小 订阅

第 1 章 汇编语言基础知识

〔习题 1.1〕简述计算机系统的硬件组成及各部分作用。

〔解答〕

CPU：包括运算器、控制器和寄存器组。运算器执行所有的算术和逻辑运算；控制器负责把指令逐条从存储器中取出，经译码分析后向机器发出各种控制命令，并正确完成程序所要求的功能；寄存器组为处理单元提供所需要的数据。

存储器：是计算机的记忆部件，它用来存放程序以及程序中所涉及的数据。

外部设备：实现人机交换和机间的通信。

〔习题 1.2〕明确下列概念或符号：

主存和辅存，**RAM** 和 **ROM**，存储器地址和 **I/O** 端口，**KB**、**MB**、**GB** 和 **TB**

〔解答〕

主存又称内存是主存储器的简称，主存储器存放当前正在执行的程序和使用的数据，**CPU** 可以直接存取，它由半导体存储器芯片构成其成本高、容量小、但速度快。辅存是辅助存储器的简称，辅存可用于长期保存大量程序和数据，**CPU** 需要通过 **I/O** 接口访问，它由磁盘或光盘构成，其成本低、容量大，但速度慢。

RAM 是随机存取存储器的英语简写，由于 **CPU** 可以从 **RAM** 读信息，也可以向 **RAM** 写入信息，所以 **RAM** 也被称为读写存储器，**RAM** 型半导体存储器可以按地址随机读写，但这类存储器在断电后不能保存信息；而 **ROM** 中的信息只能被读出，不能被修改，**ROM** 型半导体通常只能被读出，但这类存储器断电后能保存信息。

存储器由大量存储单元组成。为了区别每个单元，我们将它们编号，于是，每个存储单元就有了一个存储地址，**I/O** 接口是由一组寄存器组成，为了区别它们，各个寄存器进行了编号，形成 **I/O** 地址，通常称做 **I/O** 端口。

KB 是千字节、**MB** 是兆字节、**GB** 是吉字节和 **TB** 是太字节，它们都是表示存储器存储单元的单位。

〔习题 1.3〕什么是汇编语言源程序、汇编程序、目标程序？

〔解答〕

用汇编语言书写的程序就称为汇编语言源程序；完成汇编工作的程序就是汇编程序；由汇编程序编译通过的程序就是目标程序。

〔习题 1.4〕汇编语言与高级语言相比有什么优缺点？
〔解答〕

汇编语言与高级语言相比的优点：由于汇编语言本质就是机器语言，它可以直接地、有效地控制计算机硬件，因而容易产生运行速度快，指令序列短小的高效目标程序，可以直接控制计算机硬件部件，可以编写在“时间”和“空间”两方面最有效的程序。

汇编语言与高级语言相比的缺点：由于与处理器密切相关导致通用性差、可移植性差，汇编语言功能有限，又涉及寄存器、主存单元等硬件细节，编写汇编语言比较繁琐，调试起来也比较困难，编译程序产生的目标程序往往比较庞大、程序难以优化，运行速度慢。

〔习题 1.5〕将下列十六进制数转换为二进制和十进制表示

- (1) FFH (2) 0H (3) 5EH (4) EFH
- (5) 2EH (6) 10H (7) 1FH (8) ABH

〔解答〕

(1) FFH	11111111B	255D
(2) 0H	0B	0D
(3) 5EH	1011110B	94D
(4) EFH	11101111B	239D
(5) 2EH	101110B	46D
(6) 10H	10000B	16D
(7) 1FH	11111B	31D
(8) ABH	10101011B	171D

〔习题 1.6〕
将下列十进制数转换为 BCD 码表示

- (1) 12 (2) 24 (3) 68 (4) 127
- (5) 128 (6) 255 (7) 1234 (8) 2458

〔解答〕

(1) 12	00010010
(2) 24	00100100
(3) 68	01101000
(4) 127	000100100111
(5) 128	000100101000

- | | |
|----------|------------------|
| (6) 255 | 001001010101 |
| (7) 1234 | 0001001000110100 |
| (8) 2458 | 0010010001011000 |

〔习题 1.7〕

将下列 BCD 码转换为十进制数

- | | | | |
|--------------|--------------|--------------|--------------|
| (1) 10010001 | (2) 10001001 | (3) 00110110 | (4) 10010000 |
| (5) 00001000 | (6) 10010111 | (7) 10000001 | (8) 00000010 |

〔解答〕

(1) 91

(2) 89

(3) 36

(4) 90

(5) 08

(6) 97

(7) 81

(8) 02

〔习题 1.8〕将下列十进制数分别用 8 位二进制数的原码、反码和补码表示

- | | | | |
|---------|----------|----------|---------|
| (1) 0 | (2) -127 | (3) 127 | (4) -57 |
| (5) 126 | (6) -126 | (7) -128 | (8) 68 |

〔解答〕

- | | | | | |
|----------|----|----------|----------|----------|
| (1) 0 | +0 | 00000000 | 00000000 | 00000000 |
| | -0 | 10000000 | 11111111 | 00000000 |
| (2) -127 | | 11111111 | 10000000 | 10000001 |
| (3) 127 | | 01111111 | 01111111 | 01111111 |
| (4) -57 | | 10101111 | 11010000 | 11010001 |
| (5) 126 | | 01111110 | 01111110 | 01111110 |
| (6) -126 | | 11111110 | 10000001 | 10000010 |
| (7) -128 | | | | 10000000 |
| (8) 68 | | 01000100 | 01000100 | 01000100 |

〔习题 1.9〕完成下列二进制数的运算

(1) $1011+1001$ (2) $1011-1001$ (3) 1011×1001 (4) $10111000\div 1001$

(5) $1011\wedge 1001$ (6) $1011\vee 1001$ (7) ~ 1011 (8) $1011?1001$

〔解答〕

(1) $1011+1001=10100$

(2) $1011-1001=0010$

(3) $1011\times 1001=1100011$

(4) $10111000\div 1001=10100$, 余数 1000

(5) $1011\wedge 1001=1001$

(6) $1011\vee 1001=1011$

(7) $\sim 1011=0100$

(8) $1011?1001=0010$ (?代表异或)

〔习题 1.10〕数码 0~9、大写字母 A~Z、小写字母 a~z 对应的 ASCII 码分别是多少? ASCII 码为 0dh、0ah 对应的是什么字符?

〔解答〕

数码 0~9: 30H~39H

大写字母 A~Z: 41H~5AH

小写字母 a~z: 61H~7AH

ASCII 码为 0dh、0ah 分别对应回车和换行控制字符。

〔习题 1.11〕计算机中有一个“01100001”编码, 如果把它认为是无符号数, 它是十进制什么数? 如果认为它是 BCD 码, 则表示什么数? 又如果它是某个 ASCII 码, 则代表哪个字符?

〔解答〕

十进制无符号数: $01100001B=61H=97$

BCD 码: 61

ASCII 码: a

〔习题 1.12〕简述 Intel 80x86 系列微处理器在指令集方面的发展。

〔解答〕

1978 年 Intel, 正式推出了 16 位 8086CPU, 1979 年 Intel 推出了准 16 位微处理器 8088, 随后, Intel 推出了 80186/80188, 80186/80188 指令系统比 8086 指令系统新增了若干条实用的指令, 涉及堆栈操作、移位指令、过程指令和边界检测及乘法指令, 1982 年 Intel 推出 80286 CPU, 80286 指令系统包括全部 80186 指令及新增的保护指令 15 条, 其中有些保护方式在实方式下也可以使用, 1985 年, Intel80x86 推

出微处理器地进入第三代 80386 CPU, 80386 指令系统在兼容原来 16 位指令系统的基础上, 全面升级为 32 位, 还新增了有关位操作、条件设置指令以及控制、调试和测试寄存器的传送指令等, 1989 年, Intel 推出了 80486CPU,80486 将浮点处理单元 FPU 集成进来, 还采用了精简指令集计算机技术 RISC 和指令流水线方式, 还新增了用于多处理器和内部 Cache 操作的 6 条指令, 1993 年 Intel 制成了俗称 586 的微处理器, 取名 Pentium。Pentium 仍为 32 位结构, 地址总线为 32 位, 对常用的简单指令用硬件实现, 重新设计指令的微代码等, Pentium 新增了一条 8 字节比较交换指令和一条处理器识别指令, 以及 4 条系统专用指令, 1996 年推出了 MMX Pentium,新增了 57 条多媒体指令, 1995 年 Intel 推出 Pentium Pro 新增了 3 条指令, 1999 年推出了 PentiumIII新增了 70 条 SSE 指令, 2000 年推出的 Pentium4 新增了 76 条 SSE2 指令

〔习题 1.13〕什么是 DOS 和 ROM-BIOS?

〔解答〕

DOS 是 Diskette Operating system 的缩写, 意思是磁盘操作系统,DOS 主要是面向磁盘的系统软件, 说得简单些, 就是人与机器的一座桥梁, 是罩在机器硬件外面的一层“外壳”, 是 1981~1995 年的个人电脑上使用的一种主要的操作系统。BIOS (Basic Input / Output System) 即基本输入输出系统, 通常是固化在只读存储器 (ROM) 中, 所以又称为 ROM-BIOS。它直接对计算机系统输入、输出设备进行设备级、硬件级的控制, 是连接软件程序和硬件设备之间的枢纽。ROM-BIOS 是计算机系统中用来提供最低级、最直接的硬件控制的程序。

〔习题 1.14〕简述 PC 机最低 1MB 主存空间的使用情况。

〔解答〕

- (1) 基本 RAM 区(00000H—9FFFFH)该区共 640KB, 由 DOS 进行管理。在这个区域中操作系统要占用掉一部分低地址空间,其它则向用户程序开放。
- (2) 保留区 RAM (A0000H--BFFFFFFH) 该区为系统安排的“显示缓冲存储区”, 共 126KB, 是显卡上的芯片提供支持, 用于存放屏幕显示信息。但这部分地址空间实际上并没有全部使用。
- (3) 扩展区 ROM (C0000H--DFFFFFFH) 该区 128KB, 由接口卡上的芯片提供支持, 用于为系统不直接支持的外设安排设备驱动程序。用户固化的程序就可[安排在这一段, 系统的会对它进行确认和连接。
- (4) 系统区 ROM (E0000H--FFFFFFH) 该区共 128KB, 由系统占用, 它主要提供 ROM-BIOS 程序, 基本输入输出程序 BIOS, 是操作系统的重要组成部分, 主要用来驱动输入输出设备, 也负责系统的上电检测, 磁盘引导等初始化操作, 在 ROM-BIOS 中还有 CMOS 微机设置程序以及使用的字符图符信息等内容。

〔习题 1.15〕罗列 8086CPU 的 8 个 8 位和 16 位通用寄存器, 并说明各自的作用。

〔解答〕

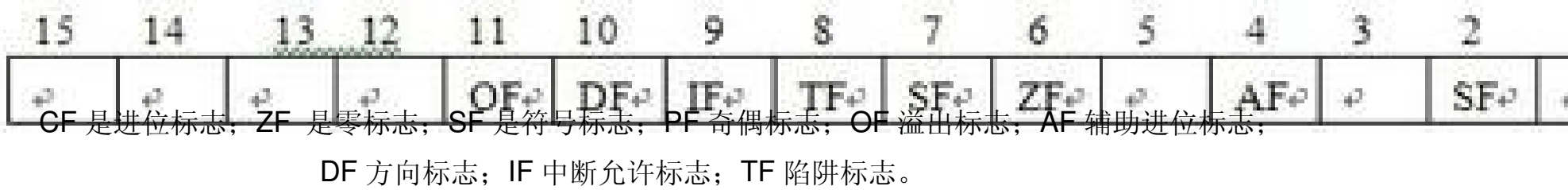
- (1) 数据寄存器: AX 称为累加器, 使用频度最高, 用于算术、逻辑运算以及与外设传送信息等; BX 称为基址寄存器, 常用做存放存储器地址; CX 称为计数器, 作为循环和串操作等指令中的隐含计数器; DX 称为数据寄存器, 常用来存放双字长数据的高 16 位, 或存放外设端口地址。
- (2) 指针及变址寄存器包括 SI,DI,BP,SP,四个寄存器, 常用于存储器寻址时提供地址。SI 是源变址寄存器, DI 是目的变址寄存器, 一般与 DS 联用确定数据段和附加段中某一存储单元地址, 在串指令中, SI 与 DS 联用、DI 和 ES 联用, 分别寻址数据段和附加段; 同时, 在串指令中, SI 和 DI 还都具有自动增

量或减量的功能。**SP**为堆栈指针寄存器，指示栈顶的偏移地址；**BP**为基地址指针寄存器，表示堆栈段中的基地址。**SP**与**BP**寄存器均可与**SS**段寄存器联合使用以确定堆栈段中的存储单元地址。

〔习题 1.16〕什么是标志，它有什么用途？状态标志和控制标志有什么区别？画出标志寄存器 **FLAGS**，说明各个标志的位置和含义。

〔解答〕

标志用于反映指令执行结果或控制指令执行形式。它是汇编语言程序设计中必须特别注意的一个方面，状态用来记录运行的结果的状态信息，许多指令的执行都将相应地设置它，控制标志位可由程序根据需要由指令设置，用来控制处理器执行指令的方式。



〔习题 1.17〕举例说明 **CF** 和 **OF** 标志的差异。

〔解答〕

溢出标志 **OF** 和进位标志 **CF** 是两个意义不同的标志。

进位标志表示无符号数运算结果是否超出范围，运算结果仍然正确；溢出标志表示有符号数运算结果是否超出范围，运算结果已经不正确。

例 1：3AH + 7CH = B6H

无符号数运算：58 + 124 = 182，范围内，无进位

有符号数运算：58 + 124 = 182，范围外，有溢出

例 2：AAH + 7CH = (1) 26H

无符号数运算：170 + 124 = 294，范围外，有进位

有符号数运算：-86 + 124 = 28，范围内，无溢出

〔习题 1.18〕字和双字在存储器中如何存放，什么是“小端方式”？对字和双字存储单元，什么是它们的对齐地址？为什么要对齐地址？

〔解答〕

字或双字在存储器中占相邻的 2 个或 4 个存储单元；存放时，低字节存入低地址，高字节存入高地址；字或双字单元的地址用它的低地址来表示。**80x86** 处理器采用的这种“低对低，高对高”的存储形式，被称为“小端方式”；将字单元安排在偶地址，双字节单元安排在模 4 地址，被称为“地址对齐方式”因为对于不对齐地址的数据，处理器访问时，需要额外的访问时间，所以通常应该将数据的地址对齐，以取得较高的存取速度。

〔习题 1.19〕什么是 **8086** 中的逻辑地址和物理地址？逻辑地址如何转换成物理地址？请将如下逻辑地址用物理地址表达：

(1) FFFFh:0 (2) 40h:17h (3) 2000h:4500h (4) B821h:4567h

〔解答〕

在 8086 处理器中，对应每个物理存储单元都有一个唯一的 20 位编号，就是物理地址，从 00000H ~ FFFFFH。

在 8086 内部和用户编程时，采用的段基地址：段内偏移地址形式称为逻辑地址。

将逻辑地址中的段地址左移二进制 4 位（对应 16 进制是一位，即乘以 16），加上偏移地址就得到 20 位物理地址

如下逻辑地址用物理地址表达：

(1) FFFFh:0 = FFFF0H

(2) 40h:17h = 00417H

(3) 2000h:4500h = 24500H

(4) B821h:4567h = BC777H （不要算错）

〔习题 1.20〕8086 有哪 4 种逻辑段，各种逻辑段分别是什么用途？

〔解答〕

代码段（Code Segment）用来存放程序的指令序列。处理器利用 CS : IP 取得下一条要执行的指令。

堆栈段（Stack Segment）确定堆栈所在的主存区域。处理器利用 SS : SP 操作堆栈中的数据。

数据段（Data Segment）存放当前运行程序所用的数据。处理器利用 DS : EA 存取数据段中的数据。

附加段（Extra Segment）是附加的数据段，也用于数据的保存。处理器利用 ES : EA 存取数据段中的数据

〔习题 1.21〕数据的默认段是哪个，是否允许其他段存放数据？如果允许，如何实现，有什么要求？

〔解答〕

数据的默认段是安排在数据段，也经常安排在附加段，尤其是串操作的目的区必须是附加段，允许其它段存放数据，数据的存放比较灵活的，实际上可以存放在任何一种逻辑段中，这时，只要明确指明是哪个逻辑段就可以了。

〔习题 1.22〕什么是操作码、操作数和寻址方式？有哪三种给出操作数的方法？

〔解答〕

操作码说明计算机要执行哪种操作，它是指令中不可缺少的组成部分，操作数是指令执行的参与者，也是各种操作的对象，我们把寻找数的方式叫做操作数的寻址方式。给出操作数的三种方法是直接给出，间接给出，隐藏操作数方式给出。

〔习题 1.23〕什么是有效地址 EA？8086 的操作数如果在主存中，有哪些寻址方式可以存取它？

〔解答〕

DS 存放数据段的段地址，存储器中操作数的偏移地址则由各种主存方式得到，称之为有效地址 EA。
8086 的操作数如果在主存中，可以存取它的寻址方式有直接寻址方式、寄存器间接寻址方式、寄存器相对寻址方式、基址变址寻址方式、相对基址变址寻址方式。

〔习题 1.24〕说明下列指令中源操作数的寻址方式？如果 BX=2000H，DI=40H，给出 DX 的值或有效地址 EA 的值。

(1) mov dx,[1234h]

(2) mov dx,1234h

(3) mov dx,bx

(4) mov dx,[bx]

(5) mov dx,[bx+1234h]

(6) mov dx,[bx+di]

(7) mov dx,[bx+di+1234h]

〔解答〕

(1) 直接寻址，EA=1234H

(2) 立即数寻址，DX=1234H

(3) 寄存器寻址，DX=2000H

(4) 间接寻址，EA=2000H

(5) 相对寻址，EA=3234H

(6) 基址变址寻址，EA=2040H

(7) 相对基址变址寻址，EA=3274H