

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

汇编语言程序设计 实验指导及习题解答

牛斗 宋人杰 张洪业 李红彪 编著



清华大学出版社

21世纪普通高校计算机公共课程规划教材

内容简介

汇编语言程序设计 实验指导及习题解答

牛斗 宋人杰 张洪业 李红彪 编著

清华大学出版社
北京

内 容 简 介

本书是汇编语言程序设计学习过程中的辅助教材,与宋人杰教授主编的《汇编语言程序设计》教材配套使用。本教材共分为两大部分,第一部分是程序设计实验指导教材,第二部分是习题解答。在实验指导部分结合实际的实验环境和上机实验方法给出了具体的实验内容,第1~9章为基础实验,第10章为4个综合性程序设计实验。在习题解答部分给出了宋人杰教授主编的《汇编语言程序设计》一书各章节的习题参考答案。

本书适于作为普通高等院校计算机科学与技术专业及相关专业的汇编语言程序设计的配套教材,也可以作为自学考试参考教材和相关专业技术人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

汇编语言程序设计实验指导及习题解答/牛斗等编著. —北京:清华大学出版社,2009.8
(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-20076-5

I. 汇… II. 牛… III. 汇编语言—程序设计—高等学校—教学参考资料 IV. TP313

中国版本图书馆CIP数据核字

责任编辑:梁颖 赵晓宁

责任校对:梁毅

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:9.75 字 数:241千字

版 次:2009年8月第1版 印 次:2009年8月第1次印刷

印 数:1~4000

定 价:17.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:032920-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。本规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前 言

本书是汇编语言程序设计学习过程中的辅助教材，它与宋人杰教授主编的《汇编语言程序设计》教材配套使用。汇编语言程序设计技术是计算机专业学生必须掌握的基本技能之一。使用汇编语言编写的程序可以使用户直接访问计算机系统内部资源，具有时性强、占用存储资源少、执行速度快、代码效率高等优点。此外通过汇编语言的学习可以使学生更好地理解计算机系统的组成及工作原理。因此“汇编语言程序设计”这门课程始终是高校计算机专业及相关学科的经典课程之一。但是学生们在学习汇编语言的过程中，特别是在程序设计实验和习题解答这些环节中经常遇到参考资料少的问题，为此特编写本教材为学生们学习汇编语言提供帮助。本教材共分为两大部分，第一部分是程序设计实验指导教材，第二部分是《汇编语言程序设计》一书的习题解答。在实验指导部分，编者结合自身的教学和实践经验，尽可能选用一些实用性较强的应用程序作为实验内容，有助于提高学生分析和解决实际问题的能力和实际编程能力。

本书分为两篇共 19 章。其中第 1 篇（10 章）是实验指导内容；第 2 篇（9 章）是习题解答内容。本书第 1 篇由牛斗编写，第 2 篇由张洪业编写，宋人杰教授负责审稿，李红彪高级实验师负责实验部分的上机调试。

由于编者水平有限，书中难免存在错误和不妥之处，敬请广大读者批评指正。

编 者
2009 年 3 月

目 录

第 1 篇 实验指导篇	1
第 1 章 汇编软件 MASM 和调试工具软件 DEBUG 使用	2
1.1 实验目的	2
1.2 相关基础知识	2
1.2.1 汇编语言程序上机步骤	2
1.2.2 调试工具软件 DEBUG	5
1.3 汇编语言上机实验	9
1.4 应用 DEBUG 软件调试程序实验	10
1.5 实验报告要求	13
第 2 章 8086 指令使用	14
2.1 实验目的	14
2.2 相关基础知识	14
2.2.1 8086 CPU 的寄存器组	14
2.2.2 存储器	15
2.2.3 寻址方式	17
2.2.4 8086 CPU 常用指令	17
2.2.5 常用伪指令	22
2.3 实验内容	23
2.4 实验报告要求	25
第 3 章 顺序程序设计	26
3.1 实验目的	26
3.2 相关基础知识	26
3.2.1 汇编语言源程序基本框架	26
3.2.2 顺序程序设计框图	27
3.3 实验内容	27
3.4 实验报告要求	30
第 4 章 分支程序设计	31
4.1 实验目的	31

4.2	相关基础知识	31
4.2.1	控制转移指令	31
4.2.2	分支程序框图	33
4.3	实验内容	33
4.4	实验报告要求	37
第5章	循环程序设计	38
5.1	实验目的	38
5.2	相关基础知识	38
5.2.1	串处理指令	38
5.2.2	循环指令	41
5.2.3	循环程序框图	41
5.3	实验内容	41
5.4	实验报告要求	44
第6章	子程序设计	45
6.1	实验目的	45
6.2	相关基础知识	45
6.2.1	子程序调用和返回指令	45
6.2.2	子程序定义及结构	46
6.2.3	子程序使用注意事项	48
6.3	实验内容	48
6.4	实验报告要求	51
第7章	中断程序设计	52
7.1	实验目的	52
7.2	相关基础知识	52
7.2.1	中断操作概述	52
7.2.2	中断系统概述	52
7.2.3	常用中断指令介绍	53
7.2.4	处理器控制指令	54
7.3	实验内容	55
7.4	实验报告要求	57
第8章	DOS 调用与 BIOS 调用	58
8.1	实验目的	58
8.2	相关基础知识	58
8.2.1	DOS 与 BIOS 中断概述	58
8.2.2	DOS 系统功能调用	59

8.2.3 BIOS 功能调用	59
8.3 实验内容	61
8.4 实验报告要求	65
第 9 章 PWB 软件平台及 32 位指令的使用	66
9.1 实验目的	66
9.2 相关基础知识	66
9.2.1 PWB 的安装	66
9.2.2 PWB 的进入和退出	67
9.2.3 PWB 主菜单	67
9.2.4 PWB 开发环境的设置	68
9.2.5 源代码级调试工具软件 CodeView	68
9.3 实验内容	69
9.4 实验报告要求	70
第 10 章 综合性程序设计实验	71
10.1 实验目的	71
10.2 实验内容	71
10.3 实验报告要求	78
第 2 篇 习题解答篇	79
第 1 章 汇编语言基础知识	80
第 2 章 汇编语言源程序格式	81
第 3 章 8086/8088 寻址方式及指令系统	83
第 4 章 顺序、分支与循环程序设计	92
第 5 章 子程序及宏指令设计	99
第 6 章 32 位指令系统及程序设计	107
第 7 章 综合程序设计	109
第 8 章 输入输出与中断控制	109
第 9 章 C 语言与汇编语言混合编程	118
自测题 1	123
自测题 2	126
自测题 1 参考答案	129
自测题 2 参考答案	131
附录 A DOS 功能调用 (INT 21H) 及错误码表	134
附录 B BIOS 中断调用表 (INT N)	139
参考文献	143

第1篇

实验指导篇

第 1 章

汇编软件 MASM 和调试工具软件 DEBUG 使用

汇编语言是唯一能够充分利用计算机硬件特性并直接控制硬件设备的语言。对于诸如实时控制、软件加密解密、病毒分析、软件调试等领域，汇编语言是最有效的程序设计语言。在学习汇编语言的过程中，只有通过大量的上机实验才能更好地掌握汇编语言程序设计的技能。

1.1 实验目的

- (1) 掌握汇编软件 MASM 的使用，熟悉汇编语言的上机过程。
- (2) 掌握调试工具软件 DEBUG 的应用，掌握汇编语言程序的调试技能。

1.2 相关基础知识

1.2.1 汇编语言程序上机步骤

1. 汇编语言程序设计上机步骤

汇编语言源程序的上机步骤包括编辑、汇编、链接、调试等几个过程，如图 1-1 所示。



图 1.1 汇编语言源程序上机步骤

源程序文件的编辑就是编写一个汇编语言源程序，它可以使用任何一个文件编辑器实现。例如，Windows 2000 操作系统附件中的记事本。编辑形成的文件最后一定要以 .ASM 后缀保存。在编辑源程序的时候一定要将输入法切换到英文环境下，否则会出错。

2. 汇编

汇编是把用汇编语言编写的源代码程序翻译成计算机能够识别的二进制机器语言程序。目前常用的汇编程序是 MASM，称为宏汇编程序。宏汇编程序当前主要有两个版本：MASM5.0 和 MASM6.0。其中，MASM5.0 的可执行文件是 MASM.EXE，MASM5.0 用于 DOS 环境下。初学者首先学习使用 MASM5.0。而 MASM6.0 的可执行文件是 ML.EXE。

汇编过程是将源程序翻译为等价的二进制机器语言的过程。所产生的文件称为目标程序，其后缀为 .OBJ。在这个阶段中，将对源程序的语法进行检验，如果发现错误将给予提

示。错误提示分为严重错误和警告错误两种。严重错误指示某些指令存在语法错误，不能形成对应的二进制机器指令。而警告错误指示某些指令含义不够明确，需要提醒程序员注意。程序员可以根据提示对源程序进行修改，直到得到正确的结果为止。

在汇编过程中还会产生列表文件和交叉引用文件。列表文件的后缀为.LST，主要用于给出源程序和机器代码程序的清单，以供检查程序使用。交叉引用文件的后缀为.CRF，它是一个为源程序所引用的各种符号进行前后对照的文件，主要用于给出不同程序段中互相引用的全部符号列表。这两个文件是可有可无的，如果不需要生成这两个文件，可以在汇编过程中加以控制。

3. 链接

在汇编过程中形成了目标文件(.OBJ)，但是在该文件中，只是将源程序的操作码部分变成了机器码，而地址操作数还是可浮动的相对地址，不是实际地址，浮动地址需要再定位。只有经过链接过程才能形成最后的可执行文件。链接程序的功能就是把一个或多个目标文件合并成一个可执行文件，其后缀为.EXE。其实在链接过程中一共形成两个文件：第一个是可执行文件(.EXE)。默认的可执行文件名字与源程序文件相同，也可以根据用户要求进行修改可执行文件的名字；链接过程中形成的第二个文件是内存映像文件(.MAP)，它给出了每个段的地址分配情况和长度。如果不需要该文件则可以在链接命令执行时直接输入 Enter 键。在链接过程中还与一个库文件(.LIB 文件)相关。一般的汇编程序不需要库文件。但是当汇编语言与高级语言(如 C 语言)接口时需要用到库文件。此时输入相应的库文件名字即可。

4. 运行

通过汇编和链接，最终形成的可执行文件已经排出了程序中的语法错误，可以直接执行。但是可能还存在一些算法错误，这样的错误则要通过调试过程来修正。常用的调试工具有 DEBUG 等。

5. 汇编语言上机举例

例 编写一个名字为 ABCD.ASM 的源程序，该程序的功能是在计算机屏幕上显示一个字符串“HELLO!”。

1) 编辑源程序

利用 Windows 2000 操作系统附件中的“记事本”编辑如下源程序。

```
DATA    SEGMENT
        S1 DB 'HELLO!','$'
DATA    ENDS
STACK   SEGMENT PARA STACK
        DB 64 DUP(?)
STACK   ENDS
CODE    SEGMENT
        MAIN PROC FAR
            ASSUME CS:CODE,DS:DATA,SS:STACK
START:  PUSH    DS
            MOV     AX,0
            PUSH    AX
```

```
MOV    AX, DATA
MOV    DS, AX
MOV    AX, STACK
MOV    SS, AX
MOV    AH, 09H
MOV    DX, OFFSET S1
INT     21H
RET

MAIN ENDP
CODE    ENDS

END START
```

2) 汇编

在 MASM5.0 中的汇编程序是 MASM.EXE。

其命令格式为：

```
C:\> MASM ABCD.ASM
```

执行上述操作命令之后将在屏幕上显示如下信息：

```
Microsoft(R) Macro Assembler Version 5.00
Copyright(C) Microsoft Corp 1981-1985,1987.All rights reserved
Object filename[ABCD.OBJ]:
Source listing[NUL.LST]:
Cross reference[NUL.CRF]:
50678 + 410090 Bytes symbol space free
0 warning Errors
0 Severe Errors
```

在汇编过程中产生了三个文件，其默认的文件名显示在屏幕上，如果不需要做任何改变就直接按 Enter 键。

通过屏幕上的显示，可以了解到在本次汇编过程中没有发生错误和警告，因此可以继续链接操作。

3) 链接

在 MASM5.0 中的汇编程序是 LINK.EXE。

其命令格式为：

```
C:\> LINK ABCD
```

执行上述命令后将在屏幕上显示如下信息：

```
Microsoft(R) Overlay Linker Version 3.6
Copyright(C) Mirosoft Corp 1983-1987.All rights reserved
Run File[ABCD.EXE]
List File[NUL.MAP]:
Libraries[.LIB]:
```

在链接过程中产生了两个文件，其默认的文件名显示在屏幕上，如果不需要做任何改

变就直接按 Enter 键。对于库文件一栏, 如果不需要使用库文件, 可以直接按 Enter 键; 如果需要使用库文件, 则要求输入所使用的库文件名字。

4) 运行

其命令格式为:

C: > ABCD.┐

在屏幕上就可以看到“HELLO!”的字样了。

1.2.2 调试工具软件 DEBUG

如果汇编语言的源程序中存在一些语法或格式上的错误, 在汇编和链接过程中都可以被发现并得到纠正。但是如果在源程序中存在算法错误和逻辑错误, 那就只有在调试过程中才能发现并纠正了。常用的调试工具软件是 DEBUG。DEBUG 提供了修改寄存器和内存单元内容的命令, 可以很方便地修改寄存器、内存单元的内容, 为调试程序、修改程序带来了方便。在 DEBUG 状态下, 还提供了调试、控制测试的环境。用户可以在此环境下进行编辑、调试和执行等一些简单的汇编程序。DEBUG 有显示命令, 通过显示命令, 可以观察某个内存单元的内容、CPU 内部某个寄存器的内容, 并可以根据这些内容的变化情况、调试程序。

1. DEBUG 的启动

假设在 Windows 2000 的环境下, 通过选择“程序”→“附件”→“命令提示符”进入 DOS 状态。然后输入: Debug [[drive:][path] filename [parameters]]┐ 即可进入 DEBUG 环境。

其中, [drive:][path] filename: 指定要测试的可执行文件的位置和名称。

[parameters]: 指定所需要的命令。

如果已经进入存放 DEBUG 文件的目录下, 则可以使用简化的命令进入 DEBUG。简化命令的格式为: Debug┐。进入 DEBUG 后, 显示 DEBUG 的提示符号, 该提示符号为连字符 (-)。

2. DEBUG 的主要命令

这里只介绍常用到的几个主要 DEBUG 命令, 全部的 DEBUG 命令请参考相关文献。

1) A (汇编命令)

该命令的功能是从汇编语言程序语句创建可执行的机器码。所有数值都是十六进制格式, 必须按 1~4 个字符输入这些数值。

命令格式: A address

参数 address 是指定输入汇编语言指令的位置。

对 address 使用十六进制值, 但是输入十六进制地址时不能以 h 字符结尾。如果不指定地址, 执行 A 命令后将在它上次停止处的地址开始汇编。

A 命令使用举例如下。

```
-A 0100:0500
```

```
0100:0500 MOV AL,05
```

```
0100:0502 JMP NEAR 505
```

```
0100:0505 JMP FAR 50A
:
```

2) C (比较命令)

该命令的功能是比较内存的两个区域存放的内容。

命令格式: C range address

参数 range 指定要比较的内存第一个区域的起始和结束地址,或起始地址和长度。

参数 address 指定要比较的第二个内存区域的起始地址。

C 命令使用举例如下。

```
-C 100,10F 300
```

该命令对 100h~10Fh 的内存数据块与 300h~30Fh 的内存数据块进行比较。

Debug 响应前面的命令并显示如下信息 (假定此时 DS = 197F) :

```
197F:0100 4D E4 197F:0300
197F:0101 67 99 197F:0301
197F:0102 A3 27 197F:0302
197F:0103 35 F3 197F:0303
197F:0104 97 BD 197F:0304
197F:0105 04 35 197F:0305
197F:0107 76 71 197F:0307
197F:0108 E6 11 197F:0308
197F:0109 19 2C 197F:0309
197F:010A 80 0A 197F:030A
197F:010B 36 7F 197F:030B
197F:010C BE 22 197F:030C
197F:010D 83 93 197F:030D
197F:010E 49 77 197F:030E
197F:010F 4F 8A 197F:030F
```

注意: 列表中缺少地址 197F:0106 和 197F:0306, 这表明地址中的数据值是相同的。

3) D (显示内存命令)

该命令的功能是显示一定范围内存地址的内容。

命令格式: D [range]

参数 range 指定要显示其内容的内存区域的起始和结束地址,或起始地址和长度。如不指定 range, 将从以前 D 命令中所指定地址范围末尾开始显示 128 字节的内容。

D 命令举例如下。

```
-D CS:100 10F
```

Debug 按以下格式显示范围中的内容:

```
04BA:0100 54 4F 4D 00 53 41 57 59-45 52 00 00 00 00 00 00 TOM.SAWYER...
...
```

当使用 D 命令时, Debug 以两个部分显示内存内容: 十六进制部分 (每个字节的值都

用十六进制格式表示) 和 ASCII 码部分(每个字节的值都用 ASCII 码字符表示)。每个非打印字符在显示的 ASCII 部分由英文句号(.)表示。每个显示行显示 16 字节的内容, 其中在第 8 和第 9 字节之间有一个连字符。每个显示行从 16 字节的边界上开始。

4) G (运行命令)

该命令功能是运行当前在内存中的程序。

命令格式: G [=address] [breakpoints]

参数 address 指定当前在内存中要开始执行的程序地址。如果不指定 address 参数, 将从 CS:IP 寄存器中的当前地址开始执行程序。

参数 breakpoints 为 G 命令设置的临时断点。

如果命令中有指定断点参数, 则程序在遇到的第一个断点处停止, 当程序到达断点时, Debug 将显示所有寄存器的内容、所有标记的状态以及最后执行指令的解码形式。

5) Q (退出命令)

该命令功能是该停止 Debug 会话, 返回到命令提示符。

命令格式: Q

参数: 无。

6) R (寄存器显示命令)

该命令功能是显示或改变一个或多个 CPU 寄存器的内容。

R [register-name]

参数: 无。

如果在没有参数的情况下使用, 则 R 命令显示所有寄存器的内容以及寄存器存储区域中的标志。

参数: register-name 指定要显示其内容的寄存器名。

举例如下。

要查看所有寄存器的内容、所有标记的状态和当前位置的指令解码表, 请输入以下命令:

R

如果当前位置是 04BA:011A, 显示结果将类似如下:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A NV UP DI NG NZ AC PE NC
04BA:011A CD21 INT 21
```

如果只想查看标志的状态, 请输入以下命令:

RF

Debug 按以下格式显示信息:

```
NV UP DI NG NZ AC PE NC - _
```

7) T (跟踪命令)

该命令功能是执行一条指令，并显示所有寄存器的内容、所有标志的状态和所执行指令的源代码形式。

命令格式: T [=address] [number]

参数=address 指定 Debug 启动跟踪指令的地址。如果省略 address 参数，跟踪将从当前 CS:IP 寄存器所指定的地址开始。

参数 number 指定要跟踪的指令数。该值必须是十六进制数，默认值为 1。

举例如下。

要执行一个指令 (CS:IP 指向的指令)，然后显示寄存器的内容、标志的状态以及指令的解码形式，请输入以下命令：

T

如果程序中的指令位于 04BA:011A，Debug 可能显示下列信息：

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A NV UP DI NG NZ AC PE NC
04BA:011A CD21 INT 21
```

8) U (反汇编命令)

该命令功能是反汇编字节并显示相应的原语句，其中包括地址和字节值。反汇编代码像已汇编文件的列表。

命令格式: U [range]

参数无。

如果在没有参数的情况下使用，则 U 命令将反汇编 20H 字节的内容 (默认值)，从前面 U 命令所显示地址后的第一个地址开始。

参数 range 指定要反汇编代码的起始地址和结束地址，或起始地址和长度。

举例如下。

要反汇编 16 (10h) 字节，从地址 04BA:0100 开始，请输入以下命令：

U04ba:010010

Debug 按以下格式显示结果：

```
04BA:0100 206472 AND [SI+72],AH
04BA:0103 69 DB 69
04BA:0104 7665 JBE 016B
04BA:0106 207370 AND [BP+DI+70],DH
04BA:0109 65 DB 65
04BA:010A 63 DB 63
04BA:010B 69 DB 69
04BA:010C 66 DB 66
04BA:010D 69 DB 69
04BA:010E 63 DB 63
04BA:010F 61 DB 61
```