

微机原理与接口技术 实验教程

Experimental Course of
Microcomputer Principle and
Interface Technology

冯旭刚 章家岩 主编

中国科学技术大学出版社

微机原理与接口技术 实验教程

冯旭刚 章家岩 主编



中国科学技术大学出版社

内 容 简 介

本书以唐都 TD-PITE 型教学实验系统为实验平台,结合多年教学和应用研究经验,配合理论教学,详尽地介绍了“微机原理与接口技术”课程所需要的各类实验,并在接口及应用方面进行了适当扩展。本书共 8 章,前 5 章主要介绍了汇编语言程序设计实验、80x86 微机原理及相应的实验;考虑到微机原理接口实验和单片机原理实验的相通性以及 TD-PITE 型教学实验系统对单片机原理及应用实验的支持,在第 6~8 章介绍了单片机相关方面的实验,每一个实验都包括实验目的、实验设备、实验内容、实验原理、实验步骤等,并附有自行设计内容,同时,对实验需要的一些理论知识也进行了必要的介绍。

本书既可作为高等学校工科电类专业和其他相近专业本科教育“微机原理与接口技术”和“单片机原理及应用”课程的配套实验指导教材,也可供非电专业师生用作微型系统设计和应用的参考书。

图书在版编目(CIP)数据

微机原理与接口技术实验教程/冯旭刚,章家岩主编. —合肥:中国科学技术大学出版社, 2017.9

ISBN 978-7-312-04224-9

I. 微… II. ①冯… ②章… III. ①微型计算机—理论—教材 ②微型计算机—接口技术—教材 IV. TP36

中国版本图书馆 CIP 数据核字(2017)第 119129 号

出版 中国科学技术大学出版社

安徽省合肥市金寨路 96 号,230026

<http://press.ustc.edu.cn>

<https://zgkxjsdxcbs.tmall.com>

印刷 合肥华苑印刷包装有限公司

发行 中国科学技术大学出版社

经销 全国新华书店

开本 787 mm×1092 mm 1/16

印张 9.75

字数 250 千

版次 2017 年 9 月第 1 版

印次 2017 年 9 月第 1 次印刷

定价 28.00 元

前 言

本书是为配合高等学校工科电类专业学生学习“微机原理与接口技术”和“单片机原理及应用”而编写的实验及课程设计指导教材。

“微机原理与接口技术”课程是多数高等院校电子类专业学生的必修课程,其实验课程可以帮助学生提高对微机原理的认知和操作能力,是微机原理教学当中的重要环节。实验教学的独特功能和作用,是其他教学环节无法替代的。为了适应我校需求,结合我校电子类专业课程教学实际,我们编写了这本用于“微机原理与接口技术”课程的实验指导教材。

全书共 8 章。第 1 章为汇编语言程序设计,详细介绍了如何在 PC 计算机上进行汇编语言程序的建立和运行,在 MS-DOS 环境下进行汇编语言程序的编写、汇编、连接和调试运行,每位学生均可在个人计算机上完成实验而不需要专用实验平台,增加了实验的方便性。第 2 章为实验平台硬件基础介绍,主要包括 80x86 CPU 寄存器结构和寻址方式,唐都 TD-PITE 实验系统硬件结构和各扩展接口原理图。第 3 章主要介绍了 80x86 的两个基础实验——80x86 指令和 32 位指令寻址,目的是让学生了解和学习 80x86 CPU 的基本工作方式。第 4 章主要为微机原理接口实验,包括中断实验、DMA 应用实验、定时/计数器实验、AD 转换实验、串行/并行接口实验和部分功能扩展实验,根据以往教学经验,本章包含了配合课程教学所需的所有实验内容。第 5 章为微机应用综合设计,结合我校以往微机原理课程设计要求,总结出常用的 3 个设计题目,完成综合设计实验对学生学习 80x86 微机有很大帮助。第 6~8 章为单片机原理实验部分,其中第 6 章主要介绍了 SST89E554RC 单片机结构和片内资源,Keil C51 软件的安装和使用说明,51 单片机的开发环境和仿真调试方法;第 7 章主要是单片机基础应用实验,包括基本 IO 实验、中断实验、定时/计数器实验、看门狗实验、串口通信实验等,完成此章节实验可实现对 51 单片机大部分基础功能的掌握;第 8 章为 51 单片机应用扩展功能实验,属于单片机学习的进阶实验,包括步进电机实验、直流电机 PWM 调速实验和温度闭环控制实验等。

本书在内容选材上,既考虑了实验教学与理论的相关性,又注意了使其具有一定的独立性和实用性;既有简明的理论和方法,又有实验必要的操作步骤和设计要求。在内容编排上,注意结合微机技术领域中的新知识、新技术、新器件,提出新的实验要求,力求以综合性、设计性为主,着眼于培养学生的综合能力、创新能力和学习能力。

本书作为“安徽省重大教学改革研究项目(2015zdjy059)——基于能力素质模型的电类专业工程应用型人才培养模式的研究与实践”课题成果,由安徽工业大学冯旭刚副教授和章家岩教授担任主编。参加编写的老师还有刘一帆、王伟、陈飞、顾凌明、游春豹等。安徽工业大学电气与信息工程学院工业窑炉先进运维与节能优化研究所硕士生孟庆喜、杜罗通、冯同磊、梁聚齐为完成本书插图、校对和排版工作付出了辛勤的劳动,在此对他们表示衷心的感谢。

由于时间紧迫,书中存在疏漏之处在所难免,敬请读者批评、指正。

编 者

2017年3月17日

于安徽工业大学

目 录

前言	(i)
----------	-------

第 1 篇 微 机 篇

第 1 章 汇编语言程序设计	(3)
1.1 汇编语言程序的建立和运行	(3)
1.2 调试程序 DEBUG.EXE 使用介绍	(8)
1.2.1 DEBUG 的启动	(8)
1.2.2 DEBUG 常用命令格式及其功能	(9)
1.3 在 DEBUG 状态下建立和运行程序的操作	(14)
1.4 汇编语言源程序的建立和运行	(17)
1.5 分支程序设计	(21)
1.6 循环程序设计	(21)
1.7 从键盘输入数据并显示的程序设计	(23)
1.8 综合程序设计	(25)
第 2 章 80x86 微机原理及实验平台介绍	(27)
2.1 80x86 寄存器	(27)
2.1.1 通用寄存器	(27)
2.1.2 段寄存器	(28)
2.1.3 指令指针和标志寄存器	(28)
2.2 80x86 存储器寻址	(29)
2.2.1 存储器寻址	(29)
2.2.2 存储器寻址方式	(30)
2.2.3 支持各种数据结构	(30)
2.3 TD-PITE 微机教学实验系统介绍	(30)
2.3.1 系统概述	(30)
2.3.2 TD-PITE 微机教学实验系统简介	(31)
2.3.3 TD-PITE 系统硬件操作环境	(31)
2.4 系统认识实验	(40)

第 3 章	32 位指令及其程序设计实验	(44)
3.1	80x86 指令及程序设计	(44)
3.2	32 位指令及寻址实验	(46)
第 4 章	80x86 微机接口技术及其应用实验	(55)
4.1	8259 中断控制实验	(55)
4.2	DMA 特性及 8237 应用实验	(61)
4.3	8254 定时/计数器应用实验	(67)
4.4	8255 并行接口实验	(70)
4.5	8251 串行接口应用实验	(73)
4.6	A/D 转换实验	(80)
4.7	D/A 转换实验	(82)
4.8	电子发声设计实验	(85)
4.9	步进电机实验	(89)
第 5 章	微机应用综合设计	(92)
5.1	综合设计一	(92)
5.2	综合设计二	(94)
5.3	综合设计三	(99)

第 2 篇 单片机篇

第 6 章	单片机相关知识概述	(105)
6.1	SST89E554RC 简介	(105)
6.2	实验项目	(107)
6.3	Keil C51 的安装	(108)
6.4	μVision2 集成开发环境	(110)
6.5	仿真调试与脱机运行间的切换方法	(114)
第 7 章	单片机基础应用实验	(119)
7.1	数字量输入/输出实验	(119)
7.2	中断系统实验	(120)
7.3	定时/计数器实验	(122)
7.4	看门狗实验	(124)
7.5	低功耗实验	(126)
7.6	PCA 实验	(128)
7.7	串口通信实验	(132)
7.8	SPI 总线实验	(135)

第 8 章 单片机控制应用实验 (140)

8.1 步进电机实验 (140)

8.2 直流电机 PWM 调速实验 (142)

8.3 温度闭环控制实验 (143)

参考文献 (146)

第 1 篇 微 机 篇

第1章 汇编语言程序设计

本章主要介绍汇编语言程序设计,通过在计算机上使用 DEBUG 工具进行汇编语言编程和实验运行,通过计算机来学习 80x86 的指令系统、寻址方式以及程序的设计方法,这种实验方式不需要特殊的实验设备,只要有计算机就能进行实验,便于学生在空闲时间学习与操作。

1.1 汇编语言程序的建立和运行

汇编语言是一种面向机器的符号语言,要想在计算机上运行汇编语言程序,首先要用编辑程序产生汇编语言的源程序(扩展名为 .ASM 的文件);然后用汇编程序把源文件转换成用二进制代码表示的目标文件(扩展名为 .OBJ 的文件);再经过连接程序,形成可执行文件(扩展名为 .EXE 或 .COM 的文件)。因此,汇编语言程序的建立和运行过程如图 1.1 所示。

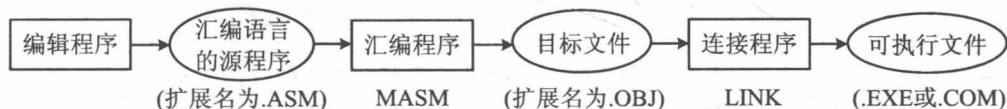


图 1.1 汇编语言程序的建立和运行过程

可见要建立和运行汇编语言程序,系统盘上应该有以下文件:

EDIT.EXE(或 PE.EXE、WS.EXE ...)	全屏幕编辑程序
MASM.EXE(或 ASM.EXE)	汇编程序
LINK.EXE	连接程序
DEBUG.EXE	动态调试程序

其中 ASM.EXE 是普通汇编程序,不支持宏汇编。若要用宏汇编,则必须用 MASM.EXE。

当用户设计好汇编语言程序后,就应按图 1.2 中的操作步骤对汇编语言源程序进行编辑、修改、连接和运行。如果需要,可用动态调试程序 DEBUG.EXE 对汇编语言源程序直接进行汇编、动态调试和运行,直观地了解程序的执行情况,进行跟踪查错,直到程序执行出正确的结果。汇编语言源程序上机运行的流程如图 1.2 所示。

【说明】

① 在用 MASM.EXE 汇编过程中,汇编程序将对源程序进行两遍扫描,第一遍扫描将源程序中所用各标识符的位置确定下来,第二遍扫描产生机器代码。如果源程序中有语法错误,则汇编程序将指出源程序中的错误,以便修改,直到最后得到无语法错误的 .OBJ 文件。

在 MS-DOS 支持下经汇编所生成的目标程序是浮动的,需通过定位才能运行。

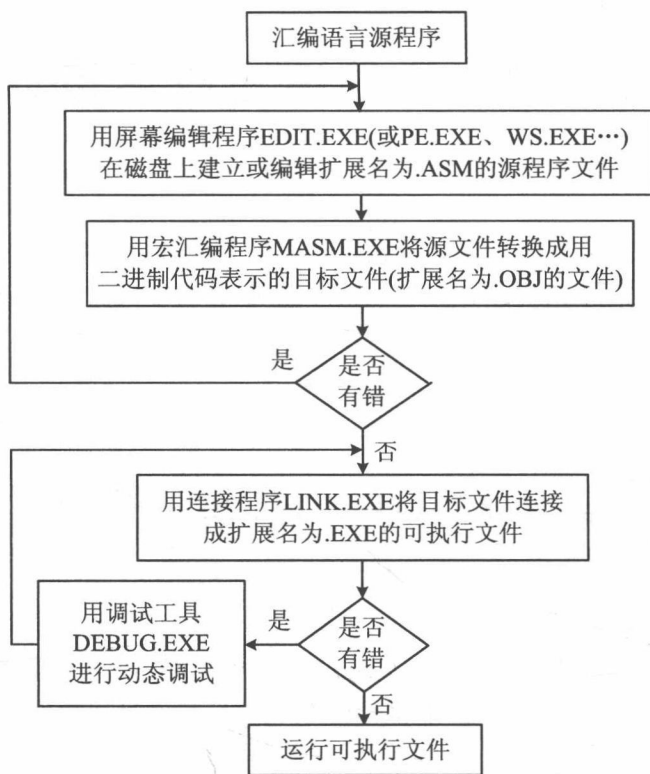


图 1.2 汇编语言源程序上机运行的流程

② 连接程序 LINK.EXE 把程序的各个模块连接在一起,把目标程序的相对地址变为绝对地址,形成可执行文件(扩展名为 .EXE)。

③ 生成可执行文件(.EXE)后,以 DOS 命令的形式直接键入文件名就可执行该程序。若程序运行结果有错,可以用 DEBUG 来调试,直到得出正确结果。

下面以建立和运行用户程序 ABC.EXE 为例介绍汇编语言源程序的输入、汇编、连接和调试运行过程。

1. 用编辑命令 EDIT(或 PE、WS、TC …)建立或编辑汇编语言源程序(扩展名为 .ASM 的文件)

源程序就是用汇编语言的语句编写的程序,它不能被机器识别。源程序必须以 .ASM 作为扩展名(扩展名即为文件的属性,如 FORTRAN 源程序的扩展名为 .FOR, BASIC 源程序的扩展名为 .BAS 等)。

【操作方法】(带下划线部分为键盘输入,后同。)

F:\MASM>EDIT ABC.ASM ↵

回车后显示 EDIT 编辑器的初始界面,如图 1.3 所示。

按回车键,进入 EDIT 编辑器的程序编辑窗口,如图 1.4 所示。

此时便可在编辑窗口内输入、编辑源程序。输入源程序后如图 1.5 所示。

源程序输入完毕,按 F3 功能键便可在屏幕下端显示图 1.6 中所示的操作提示,然后根据需要选择下一步操作。

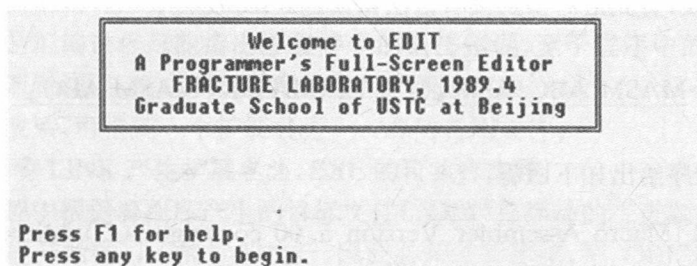


图 1.3 全屏幕编辑程序(EDIT)的初始界面

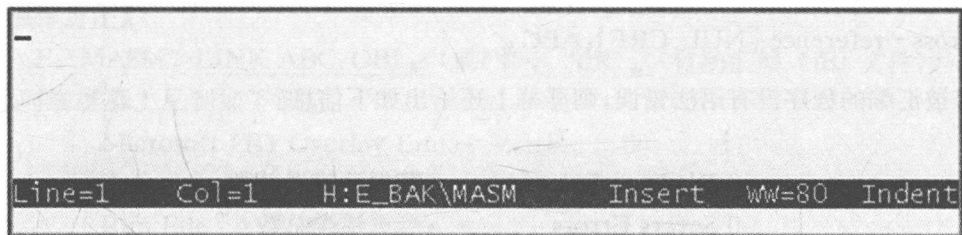


图 1.4 EDIT 编辑器输入初始画面

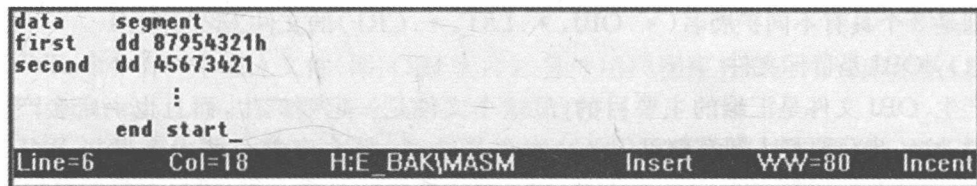


图 1.5 输入源程序



图 1.6 操作提示

例如：

键入 E (E 命令功能: 存盘后退出)

F:\MASM> (存盘完毕, 返回 DOS 提示符)

若要修改、编辑一个磁盘上已存在的源文件(修改文件), 操作方法同上。例如：

F:\MASM> EDIT 文件名. ASM

2. 用汇编命令 MASM(或 ASM)将源文件汇编成扩展名为 .OBJ 的目标文件

源程序建立后, 就可以用汇编程序 MASM.EXE(或 ASM.EXE)进行汇编。所谓汇编, 实际上就是把以 .ASM 为扩展名的源文件转换成用二进制代码表示的目标文件。目标文件以 .OBJ 为扩展名。在汇编过程中, 汇编程序对源文件进行两次扫描。若源程序中有语法错误, 在汇编过程结束后, 汇编程序会指出源程序中的错误, 这时用户可以用再用编辑程序来修

【操作方法】

此时,汇编程序给出如下回答:

Object filename [Abc.OBJ]: ↵

Source Listing [NUL.LST]:ABC ✓

Cross-reference [NUL.CRF]:ABC ↙

...

0 Warning Errors ;警告性错误总数

0 Severs Errors ;严重错误总数

F:\MASM>

(1) .OBJ 是目标文件

(2) .LST 是列表文件

(3) .CRF 是交叉符号表文件

【说明】

汇编结束后,若要显示一下汇编的结果,可查看.LST 清单文件。

【操作方法】

F:\MASM>TYPE ABC.LST ↵

此时可以看到汇编语言源程序及其对应的机器语言程序、符号表等。

在汇编过程中,如果发现程序中有语法错误,则报告错误信息;汇编结束时,进行警告性错误(Warning Errors)和严重错误(Severs Errors)汇总统计,前者指一般性错误,后者指语法性错误。具体错误信息代号可查看错误信息表。如果程序有错,则应重新调用编辑程序

修改错误和再一次汇编,直到汇编正确通过。

需要指出的是汇编过程只能指出源程序中的语法错误,至于程序中的算法错误或在程序设计中存在的逻辑错误,应在调试程序的过程中予以解决。

源程序经修改后,可得到一个扩展名为 .~AS 的备用文件。

3. 用连接命令 LINK 产生扩展名为 .EXE 的可执行文件

由于汇编过程中根据源程序产生的目标文件(.OBJ)是浮动的二进制代码文件(即相对地址都是从 0 地址开始的),不能立即执行,因此必须进行连接,把 .OBJ 文件转换成可执行的 .EXE 文件。LINK 命令还可将一个或多个 .OBJ 文件(可由用户编写,也可能是某个程序库中存在的)连接起来,生成一个可执行的 .EXE 文件。

【操作方法】

F:\MASM>LINK ABC.OBJ ↵ (或 LINK ABC ↵ (自动汇编 .OBJ 文件))

此时在屏幕上见到如下信息:

Microsoft (R) Overlay Linker Version 3.60

Copyright (C) Microsoft Corp 1983 - 1987. All rights reserved

Run File [ABC.EXE]: ↵

List File [NUL.MAP]: ABC ↵

Libraries [.LIB]: ↵

F:\MASM>

LINK 命令有一个输入文件(即 .OBJ 文件),另外用户程序有时会用到库函数,对于提示信息 Libraries[.LIB],要输入库名,若不需要则直接回车。

LINK 过程产生两个输出文件,一个是扩展名为 .EXE 的可执行文件,这个文件是 LINK 过程的主要目的;另一个是 .MAP 文件,.MAP 文件是连接程序的列表文件,又称为映像文件,它给出每个段在内存中的分配情况。需要时,可查看 ABC 文件在内存中的分配情况。

【操作方法】

F:\MASM>TYPE ABC.MAP ↵

即可显示 ABC 文件的内存分配情况。

MAP 文件也是可有可无的。

在 LINK 过程中,有时会出现“Warning: NO STACK Segment”信息,即报告“无堆栈段”的警告性错误,但这个错误并不影响程序的执行。如果在源程序中设置了堆栈段,则无此提示信息。

4. 程序的执行

在建立了 .EXE 文件后,就可执行程序了。在 DOS 状态下,可直接打入文件名,即可出现运行结果。

【操作方法】

F:\MASM>ABC ↵

此时即可出现程序运行结果。

实际上,大部分程序必须经过调试才能纠正程序设计中的错误,从而得到正确的结果。所谓调试,就是用调试程序(DEBUG.EXE 程序)来发现错误,再经过编辑、汇编、连接,达到纠正错误的目的。

1.2 调试程序 DEBUG.EXE 使用介绍

DEBUG 是 DOS 支持下的专用于 80x86 汇编语言动态调试的工具软件,它的主要功能如下:

① 从磁盘调入或接受从键盘上输入的被调试程序,汇编或反汇编一段汇编语言源程序。

② 显示/修改寄存器或存储器的内容、填充指定内存区域、内存区域内容的移动、两块内存区域内容的比较、在指定区域中查找字符等。

③ 用不同方式跟踪程序执行的轨迹,如跟踪、设置断点运行程序等。

④ 读/写磁盘文件或扇区等。

DEBUG 共为用户设置了 19 条命令,每条命令执行后都会在屏幕上给出回答信息。利用 DEBUG 命令分析、调试、修改、扩充一个代码程序十分方便。因此,掌握用 DEBUG 来对程序进行调试的方法,对学习汇编语言程序设计将是十分有用的。尤其是大程序的调试,没有 DEBUG 的帮助将会十分困难。

1.2.1 DEBUG 的启动

在 DOS 提示符 F:\>(或 C:\>或 A:\>)下,键入如下命令:

F:\MASM>DEBUG ↵

或

F:\MASM>DEBUG[d:][Path]文件名[.EXE 或 .COM]↵

- (“-”为 DEBUG 提示符)

【说明】

① 对于不带参数的 DEBUG,启动后,在 DEBUG 状态提示符“-”下,可利用 DEBUG 命令对内存现有的内容进行操作。若需对指定文件操作,则应使用 N 命令和 L 命令,将指定文件装入内存,再进行操作。操作方法为

F:\MASM>DEBUG ↵

- N[d:][path]文件名.EXE ↵

- L ↵

-

② 对带参数的 DEBUG,其参数是指被调试文件所在的驱动器的盘符、路径以及文件名和扩展名。

③ “-”是 DEBUG 的提示符,在“-”状态下,可以使用 DEBUG 的所有命令。

④ 在启动 DEBUG 时同时给出被调试程序的文件名,则 DEBUG 启动后,自动将程序(扩展名为 .COM 或 .EXE 的文件)从磁盘调入内存,存于 CS:0100H 开始的存储区,并在 CX 中装入该文件的长度(文件大于 64 KB,则在 BX、CX 中装入文件长度)。

⑤ 不带被调试文件名的 DEBUG 程序启动后,各寄存器的初始状态是 CS、DS、ES、SS

置于用户可用的内存空间的最低地址,即 DEBUG 程序结束后的第一个段;IP 置为 0100H;AX、BX、CX、DX、BP、SI、DI 置为 0000H。

带被调试文件名的 DEBUG 程序启动后,则 BX、CX 置为文件长度,段寄存器、SP、IP 置成 .EXE 文件规定的值,其余的均置为 0。

1.2.2 DEBUG 常用命令格式及其功能

1.2.2.1 汇编和反汇编命令

1. 汇编命令 - A

格式:

A [地址]

功能:将键盘输入的汇编语言语句直接汇编为机器指令存于指定的内存单元中。

【说明】

① 在 DEBUG 状态下,所有输入的数据均认为是十六进制数,故不用 H 后缀。命令前缀 CS、DS、ES、SS 和 NEAR、FAR 必须在相关指令前输入。

② 因每送入一条指令,立即被汇编成机器码,故不能使用标号、符号地址,也不能在操作数中引用标号,只能使用 DB、DW 两条伪指令。

③ DEBUG 不能识别操作数是字单元还是字节单元,须用前缀 WORD PTR (或 WORD 或 WO)、BYTE PTR (或 BYTE 或 BY)说明。

④ A 命令中的地址形式为〈段地址〉〈偏移地址〉,若地址参数缺省,DEBUG 启动后第一次使用 A 命令,则以 CS:0100H 为起始地址;若已执行过 A 命令,则以上一次使用 A 命令后的最后一个存储单元的下一个单元地址为汇编的起始地址。

⑤ 如果送入的指令格式错误,将在出错的位置显示 Error,并重新显示汇编地址,用户可重新输入正确的汇编语句。

⑥ 用 A 命令输入的程序,可用写盘命令 W 存盘(用 W 命令须慎重,不可将盘中其他有用信息冲掉),用运行命令 G 运行该程序。例如:

```
- A ✓
136E:0100 MOV AX, 2000 ✓
136E:0103 MOV DS, AX ✓
136E:0105 MOV DI, 3500 ✓
136E:0108 MOV CX, 0064 ✓
136E:010B MOV AL, 00 ✓
136E:010D MOV [DI], AL ✓
136E:010F INC DI ✓
136E:0110 ADD AL, 01 ✓
136E:0112 DAA ✓
136E:0113 LOOP 010D ✓
136E:0115 MOV AH, 02 ✓
```