INSTRUCTOR'S SOLUTION MANUAL WITH TRANSPARENCY MASTERS

to accompany

THE 8088 AND 8086 MICROPROCESSORS

Programming, Interfacing, Software, Hardware, and Applications

Including the 80287, 80386, 80486, and Pentium™ Processors

Third Edition

Walter A. Triebel
Intel Corporation

Avtar Singh
San Jose State University

Prentice Hall
Upper Saddle River, New Jersey *Columbus, Ohio*

INSTRUCTOR'S SOLUTION MANUAL WITH TRANSPARENCY MASTERS
to accompany

THE 8088 AND 8086 MICROPROCESSORS

Programming, Interfacing, Software, Hardware, and Applications

Including the 80287, 80386, 80486, and Pentium™ Processors

Third Edition

Walter A. Triebel
Intel Corporation

Avtar Singh
San Jose State University

江苏工业学院图书馆
藏书章

Prentice Hall
Upper Saddle River, New Jersey *Columbus, Ohio*

© 2000 by Prentice-Hall, Inc.
Pearson Education
Upper Saddle River, New Jersey 07458

All rights reserved. Instructors of classes using Triebel & Singh, ***The 8088 and 8086 Microprocessors, Third Edition,*** may reproduce material from the instructor's solution manual for classroom use. Otherwise, no part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN: 0-13-012842-2

CONTENTS

PREFACE

This manual contains solutions or answers to the assignment problems at the end of each chapter. These brief answers and solutions are strictly for use by the instructor. Throughout the solutions the notation ---- after a signal name stands for overbar (NOT) and (ENTER) stands for the depression of the return/enter key.

Also, included in the manual are transparency masters for many of the illustrations in the book. Transparency masters for most of the illustrations which are not supplied here were published in the solutions manual for other textbooks by the authors. A chart is provided in this manual to cross reference between figure numbers in this textbook and the corresponding figure in the textbooks:

1. THE 80386, 80486, AND PENTIUM[R] MICROPROCESSORS: Hardware, Software, and Interfacing, Walter A. Triebel c. 1998 Prentice-Hall

Solutions Manual: ISBN 0-13-533225-7

2. THE 8088 AND 8086 MICROPROCESSORS: Programming, Interfacing, Software, Hardware, and Applications, 2nd Edition Walter A. Triebel and Avtar Singh c. 1997 Prentice-Hall

Solutions Manual: ISBN 0-13-367897-0

Copies of transparency masters from the solution manual for the book identified as number 2 above can be obtained by requesting them by letter from

Walter A. Triebel
P.O. Box 1847
Wayne, NJ 07470.

Other supplements available from Prentice-Hall for the textbook are:

Laboratory Manual: **ISBN: 0-13-367913-6**
80x86 Microprocessor Experiments
Walter A. Triebel and Avtar Singh
c. 1997 Prentice-Hall, Inc.
A Simon & Schuster Company

Student's Laboratory Programs Diskette: **ISBN 0-13-788555-5**
Instructor's Laboratory Solution Manual: **ISBN 0-13-700154-1**
Instructor's Laboratory Solution Diskette: **ISBN 0-13-788563-6**

此为试读,需要完整PDF请访问: www.ertongbook.com

Support products available from third parties are as follows:

Microsoft Macroassembler

Microsoft Corporation, Redmond, WA 98052
800-426-9400

PCμLAB- Laboratory Interface Circuit Test Unit

Microcomputer Directions, Inc.
P.O. Box 15127, Fremont, CA 94539
973-872-9082

Avtar Singh

Walter A. Triebel

CHAPTER 1

Section 1.1

1. Original IBM PC.
2. The functionality of the PC can be expanded by simply adding special function boards into the system.
3. I/O channel.
4. Personal computer advanced technology.
5. Industry standard architecture.
6. Microchannel architecture.
7. A reprogrammable microcomputer is a general-purpose computer designed to run programs for a wide variety of applications, for instance, accounting, word processing, and languages such as BASIC.
8. Mainframe computer, minicomputer, and microcomputer.
9. The microcomputer is similar to the minicomputer in that it is designed to perform general-purpose data processing; however, it is smaller in size, has reduced capabilities, and is lower in cost than a minicomputer.
10. Very large scale integration.

Section 1.2

11. Input unit, output unit, microprocessing unit, and memory unit.
12. Microprocessing unit (MPU).
13. 16-bit.
14. Keyboard; mouse and joy stick.
15. Video display and printer.
16. Primary storage and secondary storage memory.
17. 360Kbytes; 10Mbytes.
18. Read-only memory (ROM) and random access read/write memory (RAM).
19. 48Kbytes; 256Kbytes.
20. The DOS program is loaded from the hard disk into RAM and then run. Since RAM is volatile, the disk operating system is lost whenever power is turned off.

Section 1.3

21. 4-bit, 8-bit, 16-bit, 32-bit, and 64-bit.
22. 4004, 8008, 8086, 80386DX.
23. 8086, 8088, 80186, 80188, 80286.
24. Million instructions per second.
25. 27 MIPS.
26. Drystone program.
27. 39; 49.
28. 30,000, 140,000, 275,000, 1,200,000, 3,000,000.
29. A special purpose microcomputer that performs a dedicated control function.

30. Event controller and data controller.
31. A multichip microcomputer is constructed from separate MPU, memory, and I/O ICs. On the other hand, a single chip microcomputer has the MPU, memory, and I/O functions all integrated into a single IC.
32. 8088, 8086, 80286, 80386DX, 80486DX, and Pentium^{R} processor.
33. Real mode and protected mode.
34. Upward software compatible means that programs written for the 8088 or 8086 will run directly on the 80286, 80386DX, and 80486DX.
35. Memory management, protection, and multitasking.
36. Floppy disk controller, communication controller, and local area network controller.

CHAPTER 2

Section 2.1

1. Bus interface unit and execution unit.
2. BIU.
3. 20 bits; 16 bits.
4. 4 bytes; 6 bytes.
5. General purpose registers, temporary operand registers, arithmetic logic unit (ALU), and status and control flags.

Section 2.2

6. Aid to the assembly language programmer for understanding a microprocessor's software operation.
7. Their purpose, function, operating capabilities, and limitations.
8. 14.
9. 1,048,576 (1M) bytes.
10. 65,536 (64K) bytes.

Section 2.3

11. $FFFFF_{16}$ and 00000_{16}.
12. Bytes.
13. $00FF_{16}$; aligned word.
14. 44332211_{16}; misaligned double word.
15.

Address	Contents
0A002H	CDH
0A003H	ABH

aligned word.

16.

Address	Contents
0A001H	78H
0A002H	56H
0A003H	34H
0A004H	12H

misaligned double word.

Section 2.4

17. Unsigned integer, signed integer, unpacked BCD, packed BCD, and ASCII.
18. **(a)** 7FH
(b) F6H
(c) 80H
(d) 01F4H
19. (0A000H) = F4H
(0A001H) = 01H
20. -1000 = 2's complement of 1000
= FC18H
21. **(a)** 00000010, 00001001; 00101001.
(b) 00001000, 00001000; 10001000.
22. (0B000H) = 09H
(0B001H) = 02H
23. NEXT I
24. (0C000H) = 34H
(0C001H) = 33H
(0C002H) = 32H
(0C003H) = 31H

Section 2.5

25. 64Kbytes.
26. Code segment (CS) register, stack segment (SS) register, data segment (DS) register, and extra segment (ES) register.
27. CS.
28. Up to 256Kbytes.
29. Up to 128Kbytes.

Section 2.6

30. Pointers to interrupt service routines.
31. 80_{16} through $FFFEF_{16}$.
32. Instructions of the program can be stored anywhere in the general-use part of the memory address space.
33. Control transfer to the reset power-up initialization software routine.

Section 2.7

34. The instruction pointer is the offset address of the next instruction to be fetched by the 8088 relative to the current value in CS.
35. The instruction is fetched from memory; decoded within the 8088; operands are read from memory or internal registers; the operation specified by the instruction is performed on the data; and results are written back to either memory or an internal register.
36. IP is incremented such that it points to the next sequential word of instruction code.

Section 2.8

37. Accumulator (A) register; base (B) register; count (C) register; and data (D) register.
38. With a postscript X to form AX, BX, CX, and DX.
39. DH and DL.
40. Count for string operations and count for loop operations.

Section 2.9

41. Offset address of a memory location relative to a segment base address.
42. Base pointer (BP) and stack pointer (SP).
43. SS.
44. DS.
45. Source index register; destination index register.
46. The address in SI is the offset to a source operand and DI contains the offset to a destination operand.

Section 2.10

47.

Flag	*Type*
CF	Status
PF	Status
AF	Status
ZF	Status
SF	Status
OF	Status
TF	Control
IF	Control
DF	Control

48. CF = 1, if a carry-out/borrow-in results for the MSB during the execution of an arithmetic instruction. Else it is 0.

PF = 1, if the result produced by execution of an instruction has even parity. Else it is 0.

AF = 1, if there is a carry-out/borrow-in for the fourth bit during the execution of an arithmetic instruction.

ZF = 1, if the result produced by execution of an instruction is zero. Else it is 0.

SF = 1, if the result produced by execution of an instruction is negative. Else it is 0.

OF = 1, if an overflow condition occurs during the execution of an arithmetic instruction. Else it is 0.

49. Instructions can be used to test the state of these flags and, based on their setting, to modify the sequence in which instructions of the program are executed.

50. Trap flag.

51. DF.

52. Instructions are provided that can load the complete register or modify specific flag bits.

Section 2.11

53. 20 bits.

54. Offset and segment base.

55. (a) 11234H
(b) 0BBCDH
(c) A32CFH
(d) C2612H

56. (a) ? = 0123H
(b) ? = 2210H
(c) ? = 3570H
(d) ? = 2600H

57. $021AC_{16}$.

58. $A000_{16}$.

59. 1234_{16}.

Section 2.12

60. The stack is the area of memory used to temporarily store information (parameters) to be passed to subroutines and other information such as the contents of IP and CS that is needed to return from a called subroutine to the main part of the program.

61. $CFF00_{16}$.

62. 128 words.

63. FEFEH → (SP)
(AH) = EEH → (CFEFFH)
(AL) = 11H → (CFEFEH)

Section 2.13

64. Separate.

65. 64Kbytes.

66. Page 0.

CHAPTER 3

Section 3.1

此为试读,需要完整PDF请访问: www.ertongbook.com

1. Data transfer instructions, arithmetic instructions, logic instructions, string manipulation instructions, control transfer instructions, and processor control instructions.

Section 3.2

2. Software.
3. Program.
4. 80386DX machine code.
5. Instructions encoded in machine language are coded in 0s and 1s, while assembly language instructions are written with alphanumeric symbols such as MOV, ADD, or SUB.
6. Mnemonic that identifies the operation to be performed by the instruction; ADD and MOV.
7. The data to be processed during execution of an instruction; source operand and destination operand.
8. START; ;Add BX to AX
9. An assembler is a program used to convert an assembly language source program to its equivalent program in machine code. A compiler is a program that converts a program written in a high-level language to equivalent machine code.
10. Programs written is assembly language or high level language statements are called source code. The machine code output of an assembler or compiler is called object code.
11. It takes up less memory and executes faster.
12. A real-time application is one in which the tasks required by the application must be completed before any other input to the program occurs that can alter its operation.
13. Floppy disk subsystem control and communications to a printer; code translation and table sort routines.

Section 3.3

14. $0000001111000010_2 = 03C2H$.
15. **(a)** $1000100100010101_2 = 8915H$; **(b)** $1000100100011000_2 = 8918H$;
(c) $10001010010101110001000 0_2 = 8A5710H$.
16. **(a)** $00011110_2 = 1EH$; **(b)** $1101001011000011_2 = D2C3H$;
(c) $11000001100011010000010010_2 = 03063412H$.

Section 3.4

17. An addressing mode means the method by which an operand can be specified in a register or a memory location.
18. Register operand addressing mode
Immediate operand addressing mode
Memory operand addressing modes
19. Base, index, and displacement.
20. Direct addressing mode
Register indirect addressing mode
Based addressing mode
Indexed addressing mode

Based-indexed addressing mode

21.

Instruction	Destination	Source
(a)	Register	Register
(b)	Register	Immediate
(c)	Register indirect	Register
(d)	Register	Register indirect
(e)	Based	Register
(f)	Indexed	Register
(g)	Based-indexed	Register

22.

(a) PA = $0B200_{16}$
(b) PA = $0B100_{16}$
(c) PA = $0B700_{16}$
(d) PA = $0B600_{16}$
(e) PA = $0B900_{16}$

Section 3.5

23. 3 bytes.
24. 24 bytes.

CHAPTER 4

Section 4.1

1. The DEBUG program allows us to enter a program into the PC's memory, execute it under control, view its operation, and modify it to fix errors.
2. Yes.
3. Error.
4.

```
-R CX          (ENTER)
CX XXXX
:0010          (ENTER)
```

5.

```
-R F           (ENTER)
NV UP EI PL NZ NA PO NC  -PE   (ENTER)
```

6.

```
-R             (ENTER)
```

Section 4.2

7.

```
-D CS:0000 000F      (ENTER)
```

8.

```
-E CS:0        (ENTER)
```

1342:0000 CD. 20. 00. 40. 00. 9A. EE. FE.
1342:0008 1D. F0. F5. 02. A7. 0A. 2E. 03. (ENTER)
After a byte of data is displayed, the space bar is depressed to display the next byte. The values displayed may not be those shown but will be identical to those displayed with the DUMP command.

9.

-E CS:100 FF FF FF FF FF (ENTER)

10.

-E SS:(SP) 0 .....0 (32 zeros) (ENTER)
-

11.

-F CS:100 105 11 (ENTER)
-F CS:106 10B 22 (ENTER)
-F CS:10C 111 33 (ENTER)
-F CS:112 117 44 (ENTER)
-F CS:118 11D 55 (ENTER)
-E CS:105 (ENTER)
CS:0105 XX.FF (ENTER)
-E CS:113 (ENTER)
CS:0113 XX.FF (ENTER)
-D CS:100 11D (ENTER)
-S CS:100 11D FF (ENTER)

Section 4.3

12. Input command and output command.
13. Contents of the byte-wide input port at address 0123_{16} is input and displayed on the screen.
14.

O 124 5A (ENTER)

Section 4.4

15. The sum and difference of two hexadecimal numbers.
16. 4 digits.
17.

H FA 5A (ENTER)

Section 4.5

18.

-E CS:100 32 0E 34 12 (ENTER)
-U CS:100 103 (ENTER)
1342:100 320E3412 XOR CL,[1234]
-W CS:100 1 50 1 (ENTER)

19.

```
-L CS:400 1 50 1      (ENTER)
-U CS:400 403         (ENTER)
1342:0400 320E3412     XOR   CL,[1234]
-
```

Section 4.6

20.

```
-A CS:100             (ENTER)
1342:0100 MOV [DI],DX    (ENTER)
1342:0102             (ENTER)
```

21.

```
-A CS:200             (ENTER)
1342:0200 ROL BL,CL                (ENTER)
1342:0202             (ENTER)
-U CS:200 201         (ENTER)
1342:0200 D2C3  ROL BL,CL
-
```

Section 4.7

22.

```
-L CS:300 1 50 1      (ENTER)
-U CS:300 303         (ENTER)
-R CX                 (ENTER)
CX XXXX
:000F                 (ENTER)
-E DS:1234 FF         (ENTER)
-T =CS:300            (ENTER)
-D DS:1234 1235       (ENTER)
```

23.

```
-N A:BLK.EXE          (ENTER)
-L CS:200             (ENTER)
-R DS                 (ENTER)
DS 1342
:2000                 (ENTER)
-F DS:100 10F FF      (ENTER)
-F DS:120 12F 00      (ENTER)
-D DS:100 10F         (ENTER)
2000:0100  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF
-D DS:120 12F         (ENTER)
2000:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
-R DS                 (ENTER)
DS 2000
:1342                 (ENTER)
```

```
-R                    (ENTER)
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1342 ES=1342 SS=1342 CS=1342 IP=0100 NV UP EI PL NZ NA PO NC
1342:0100 8915     MOV     [DI],DX          DS:0000=20CD
-U CS:200 217      (ENTER)
1342:0200 B80020   MOV  AX,2000
1342:0203 8ED8     MOV  DS,AX
1342:0205 BE0001   MOV  SI,0100
1342:0208 BF2001   MOV  DI,0120
1342:020B B91000   MOV  CX,0010
1342:020E 8A24     MOV  AH,[SI]
1342:0210 8825     MOV  [DI],AH
1342:0212 46       INC  SI
1342:0213 47       INC  DI
1342:0214 49       DEC  CX
1342:0215 75F7      JNZ  020E
1342:0217 90       NOP
-G =CS:200 217     (ENTER)
AX=FF00 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0110 DI=0130
DS=2000 ES=1342 SS=1342 CS=1342 IP=0217 NV UP EI PL ZR NA PE NC
1342:0217 90       NOP
-D DS:100 10F      (ENTER)
2000:0100  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF
-D DS:120 12F      (ENTER)
2000:0120  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF
```

Section 4.8

24. A syntax error is an error in the rules of coding the program. On the other hand, an execution error is an error in the logic of the planned solution for the problem.

25. Bugs.

26. Debugging the program.

27.

```
-N A:BLK.EXE       (ENTER)
-L CS:200          (ENTER)
-U CS:200 217      (ENTER)
1342:0200 B80020   MOV  AX,2000
1342:0203 8ED8     MOV  DS,AX
1342:0205 BE0001   MOV  SI,0100
1342:0208 BF2001   MOV  DI,0120
1342:020B B91000   MOV  CX,0010
1342:020E 8A24     MOV  AH,[SI]
1342:0210 8825     MOV  [DI],AH
1342:0212 46       INC  SI
1342:0213 47       INC  DI
```

```
1342:0214 49          DEC   CX
1342:0215 75F7        JNZ   020E
1342:0217 90          NOP
-R DS                 (ENTER)
DS 1342
:2000                 (ENTER)
-F DS:100 10F FF      (ENTER)
-F DS:120 12F 00      (ENTER)
-R DS                 (ENTER)
DS 2000
:1342                 (ENTER)
-G =CS:200 20E        (ENTER)
AX=2000 BX=0000 CX=0010 DX=0000 SP=FFEE BP=0000 SI=0100 DI=0120
DS=2000 ES=1342 SS=1342 CS=1342 IP=020E NV UP EI PL NZ NA PO NC
1342:020E 8A24        MOV AH,[SI]           DS:0100=FF
-D DS:120 12F         (ENTER)
2000:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
-G 212                (ENTER)
AX=FF00 BX=0000 CX=0010 DX=0000 SP=FFEE BP=0000 SI=0100 DI=0120
DS=2000 ES=1342 SS=1342 CS=1342 IP=0212 NV UP EI PL NZ NA PO NC
1342:0212 46          INC   SI
-D DS:120 12F         (ENTER)
2000:0120  FF 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
-G 215                (ENTER)
AX=FF00 BX=0000 CX=000F DX=0000 SP=FFEE BP=0000 SI=0101 DI=0121
DS=2000 ES=1342 SS=1342 CS=1342 IP=0215 NV UP EI PL NZ AC PE NC
1342:0215 75F7        JNZ   020E
-G 20E                (ENTER)
AX=FF00 BX=0000 CX=000F DX=0000 SP=FFEE BP=0000 SI=0101 DI=0121
DS=2000 ES=1342 SS=1342 CS=1342 IP=020E NV UP EI PL NZ AC PE NC
1342:020E 8A24        MOV AH,[SI]           DS:0101=FF
-G 215                (ENTER)
AX=FF00 BX=0000 CX=000E DX=0000 SP=FFEE BP=0000 SI=0102 DI=0122
DS=2000 ES=1342 SS=1342 CS=1342 IP=0215 NV UP EI PL NZ NA PO NC
1342:0215 75F7        JNZ   020E
-D DS:120 12F         (ENTER)
2000:0120  FF FF 00 00 00 00 00 00-00 00 00 00 00 00 00 00
-G 20E                (ENTER)
AX=FF00 BX=0000 CX=000E DX=0000 SP=FFEE BP=0000 SI=0102 DI=0122
DS=2000 ES=1342 SS=1342 CS=1342 IP=020E NV UP EI PL NZ NA PO NC
1342:020E 8A24        MOV AH,[SI]           DS:0102=FF
-G 217                (ENTER)
AX=FF00 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0110 DI=0130
DS=2000 ES=1342 SS=1342 CS=1342 IP=0217 NV UP EI PL ZR NA PE NC
1342:0217 90          NOP
```


此为试读,需要完整PDF请访问: www.ertongbook.com