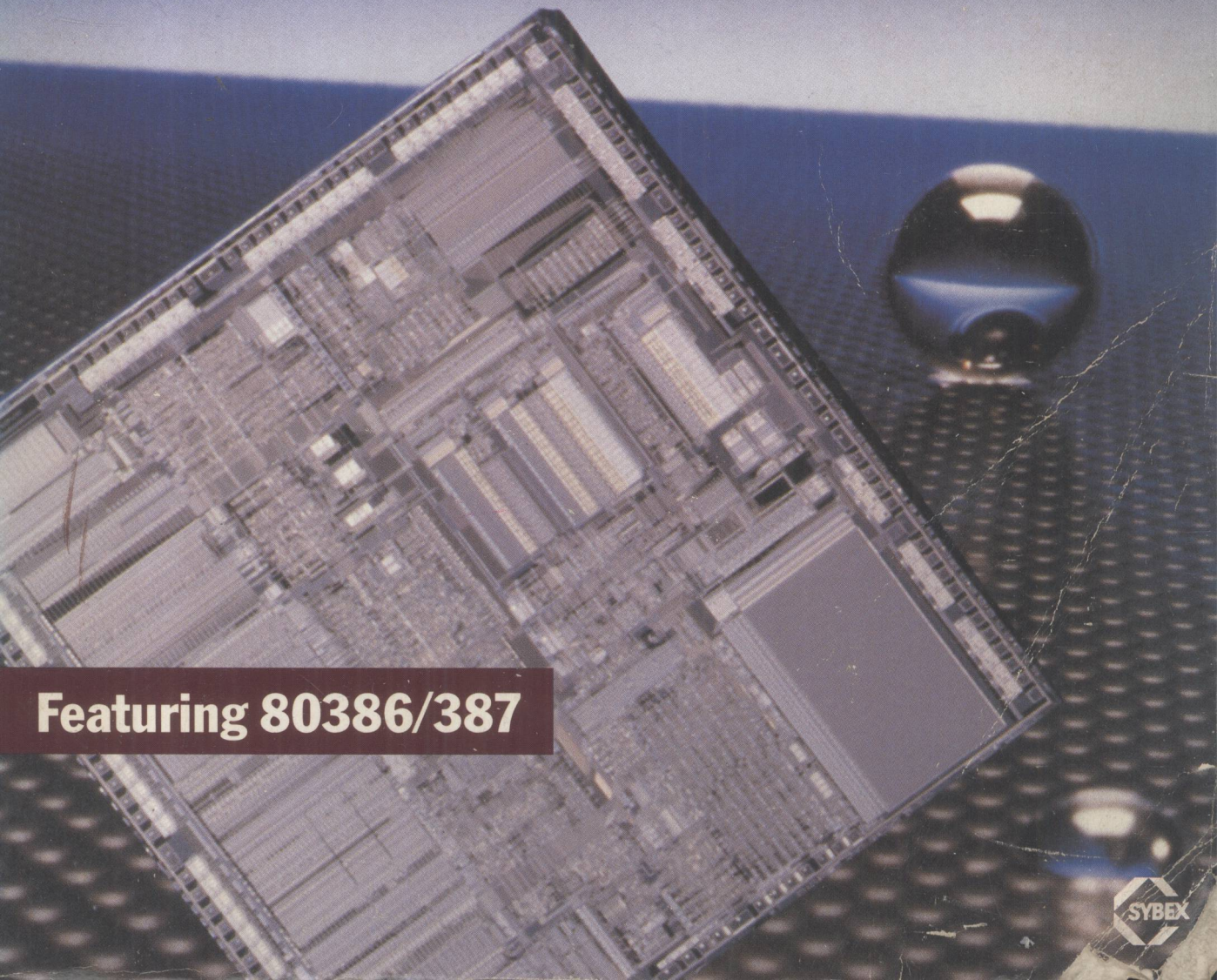
SYBEX

PROGRAMMING THE 80386

John H. Crawford Patrick P. Gelsinger



Featuring 80386/387

SYBEX

Programming the 80386

John H. Crawford
Patrick P. Gelsinger

San Francisco • Paris • Düsseldorf • London

Cover design by Thomas Ingalls + Associates
Cover photography by Casey Cartwright

Library of Congress Card Number: 87-61199
ISBN 0-89588-381-3
Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

Programming the 80386

Two-Byte 80386 Opcode Map
(First byte is OFH)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Grp6	Grp7	LAR Gv,Ew	LSL Gv,Ew			CLTS										0
1																	1
2	MOV Cd,Rd	MOV Dd,Rd	MOV Rd,Cd	MOV Rd,Dd	MOV Td,Rd		MOV Rd,Td										2
≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
8	Long-displacement jump on condition (Jv)																8
	JO	JNO	JB	JNB	JZ	JNZ	JBE	JNBE	JS	JNS	JP	JNP	JL	JNL	JLE	JNLE	
9	Byte Set on condition (Eb)								SETS	SETNS	SETP	SETNP	SETL	SETNL	SETLE	SETNLE	9
	SETO	SETNO	SETB	SETNB	SETZ	SETNZ	SETBE	SETNBE									
A	PUSH FS	POP FS		BT Ev,Gv	SHLD EvGvIb	SHLD EvGvCL			PUSH GS	POP GS		BTS Ev,Gv	SHRD EvGvIb	SHRD EvGvCL		IMUL Gv,Ev	A
B			LSS Mp	BTR Ev,Gv	LFS Mp	LGS Mp	MOVZX				Grp-8 Ev,Ib	BTC Ev,Gv	BSF Gv,Ev	BSR Gv,Ev	MOVSX		B
							Gv,Eb	Gv,Ew							Gv,Eb	Gv,Ew	
≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F																	F

Opcodes Determined by Bits 5,4,3 of MODRM Field

mod	nnn	R/M

Group	000	001	010	011	100	101	110	111
1	ADD	OR	ADC	SBB	AND	SUB	XOR	CMP
2	ROL	ROR	RCL	RCR	SHL	SHR		SAR
3	TEST Ib/Iv		NOT	NEG	MUL AL/eAX	IMUL AL/eAX	DIV AL/eAX	IDIV AL/eAX
4	INC Eb	DEC Eb						

Group	000	001	010	011	100	101	110	111
5	INC Ev	DEC Ev	CALL Ev	CALL eP	JMP Ev	JMP Ep	PUSH Ev	
6	SLDT Ew	STR Ew	LLDT Ew	LTR Ew	VERR Ew	VERW Ew		
7	SGDT Ms	SIDT Ms	LGDT Ms	LIDT Ms	SMSW Ew		LMSW Ew	
8					BT	BTS	BTR	BTC

One-Byte 80386 Opcode Map

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	ADD Eb,Gb	ADD Ev,Gv	ADD Gb,Eb	ADD Gv,Ev	ADD AL,Ib	ADD eAX,Iv	PUSH ES	POP ES	OR Eb,Gb	OR Ev,Gv	OR Gb,Eb	OR Gv,Ev	OR AL,Ib	OR eAX,Iv	PUSH CS	2-byte escape	0
1	ADC Eb,Gb	ADC Ev,Gv	ADC Gb,Eb	ADC Gv,Ev	ADC AL,Ib	ADC eAX,Iv	PUSH SS	POP SS	SBB Eb,Gb	SBB Ev,Gv	SBB Gb,Eb	SBB Gv,Ev	SBB AL,Ib	SBB eAX,Iv	PUSH DS	POP DS	1
2	AND Eb,Gb	AND Ev,Gv	AND Gb,Eb	AND Gv,Ev	AND AL,Ib	AND eAX,Iv	SEG =ES	DAA	SUB Eb,Gb	SUB Ev,Gv	SUB Gb,Eb	SUB Gv,Ev	SUB AL,Ib	SUB eAX,Iv	SEG =CS	DAS	2
3	XOR Eb,Gb	XOR Ev,Gv	XOR Gb,Eb	XOR Gv,Ev	XOR AL,Ib	XOR eAX,Iv	SEG =SS	AAA	CMP Eb,Gb	CMP Ev,Gv	CMP Gb,Eb	CMP Gv,Ev	CMP AL,Ib	CMP eAX,Iv	SEG =CS	AAS	3
4	INC general register eAX	INC general register eCX	INC general register eDX	INC general register eBX	INC general register eSP	INC general register eBP	INC general register eSI	INC general register eDI	DEC general register eAX	DEC general register eCX	DEC general register eDX	DEC general register eBX	DEC general register eSP	DEC general register eBP	DEC general register eSI	DEC general register eDI	4
5	PUSH general register eAX	PUSH general register eCX	PUSH general register eDX	PUSH general register eBX	PUSH general register eSP	PUSH general register eBP	PUSH general register eSI	PUSH general register eDI	POP into general register eAX	POP into general register eCX	POP into general register eDX	POP into general register eBX	POP into general register eSP	POP into general register eBP	POP into general register eSI	POP into general register eDI	5
6	PUSHA	POPA	BOUND Gv,Ma	ARPL Ew,Rw	SEG =FS	SEG =GS	Operand Size	Address Size	PUSH Iv	IMUL GvEvIv	PUSH Ib	IMUL GvEvIb	INSB Yb,DX	INSW/D Yv,DX	OUTSB DX,Xb	OUTSW/D DX,Xv	6
7	Short-displacement jump on condition (Jb) JO	Short-displacement jump on condition (Jb) JNO	Short-displacement jump on condition (Jb) JB	Short-displacement jump on condition (Jb) JNB	Short-displacement jump on condition (Jb) JZ	Short-displacement jump on condition (Jb) JNZ	Short-displacement jump on condition (Jb) JBE	Short-displacement jump on condition (Jb) JNBE	Short-displacement jump on condition (Jb) JS	Short-displacement jump on condition (Jb) JNS	Short-displacement jump on condition (Jb) JP	Short-displacement jump on condition (Jb) JNP	Short-displacement jump on condition (Jb) JL	Short-displacement jump on condition (Jb) JNL	Short-displacement jump on condition (Jb) JLE	Short-displacement jump on condition (Jb) JNLE	7
8	Immediate Grp1 Eb,Ib	Immediate Grp1 Ev,Iv		Grp1 Ev,Ib	TEST Eb,Gb	TEST Ev,Gv	XCHG Eb,Gb	XCHG Ev,Gv	MOV Eb,Gb	MOV Ev,Gv	MOV Gb,Eb	MOV Gv,Ev	MOV Ew,Sw	LEA Gv,M	MOV Sw,Ew	POP Ev	8
9	NOP	XCHG word or double-word register with eAX eCX	XCHG word or double-word register with eAX eDX	XCHG word or double-word register with eAX eBX	XCHG word or double-word register with eAX eSP	XCHG word or double-word register with eAX eBP	XCHG word or double-word register with eAX eSI	XCHG word or double-word register with eAX eDI	CBW	CWD	CALL Ap	WAIT	PUSHF Fv	POPF Fv	SAHF	LAHF	9
A	MOV AL,Ob	MOV eAX,Ov	MOV Ob,AL	MOV Ov,eAX	MOVSB Xb,Yb	MOVSW/D Xv,Yv	CMPSB Xb,Yb	CMPSW/D Xv,Yv	TEST AL,Ib	TEST eAX,Iv	STOSB Yb,AL	STOSW/D Yv,eAX	LODSB AL,Xb	LODSW/D eAX,Xv	SCASB AL,Xb	SCASW/D eAX,Xv	A
B	MOV immediate byte into byte register AL	MOV immediate byte into byte register CL	MOV immediate byte into byte register DL	MOV immediate byte into byte register BL	MOV immediate byte into byte register AH	MOV immediate byte into byte register CH	MOV immediate byte into byte register DH	MOV immediate byte into byte register BH	MOV immediate word or double into word or double register eAX	MOV immediate word or double into word or double register eCX	MOV immediate word or double into word or double register eDX	MOV immediate word or double into word or double register eBX	MOV immediate word or double into word or double register eSP	MOV immediate word or double into word or double register eBP	MOV immediate word or double into word or double register eSI	MOV immediate word or double into word or double register eDI	B
C	Shift Grp2 Eb,Ib	Shift Grp2 Ev,Ib	RET near Iw	RET near	LES Gv,Mp	LDS Gv,Mp	MOV Eb,Ib	MOV Ev,Iv	ENTER Iw,Ib	LEAVE	RET far Iw	RET far	INT 3	INT Ib	INTO	IRET	C
D	Shift Grp2 Eb,1	Shift Grp2 Ev,1	Shift Grp2 Eb,CL	Shift Grp2 Ev,CL	AAM	AAD		XLAT	ESC (Escape to coprocessor instruction set)								D
E	LOOPNE Jb	LOOPE Jb	LOOP Jb	JCXZ Jb	IN AL,Ib	IN eAX,Ib	OUT Ib,AL	OUT Ib,eAX	CALL Av	JMP Jv	JMP Ap	JMP Jb	IN AL,DX	IN eAX,DX	OUT DX,AL	OUT DX,eAX	E
F	LOCK		REPNE	REP/ REPE	HLT	CMC	Unary Grp3 Eb	Unary Grp3 Ev	CLC	STC	CLI	STI	CLD	STD	INC/DEC Grp4	Indirct Grp5	F

To our wives, Norma and Linda, who were first chip widows and then book widows.

— John Crawford
— Patrick Gelsinger
Santa Clara, 1987

Acknowledgments

CREATING A BOOK LIKE THIS ONE IS A COMPLEX AND EXACTING PROJECT. We would like to thank Intel Corporation for producing the 80386 and their customers for making the 80386 a success. Thanks to Norma Crawford for word processing of early drafts. Thanks to David Perlmutter for educating us in the operation of the 80387.

We would also like to thank the people at SYBEX who helped bring *Programming the 80386* from the early stages of development to the finished work you see. Our thanks to Dr. R.S. Langer, editor-in-chief, for his enthusiastic support and his choice of fine restaurants; David Kolodney, developmental and project editor; Tanya Kucak, editor, for her fine word chiseling by her red pen that never ran dry; Dan Tauber, technical editor; Olivia Shinomoto, word processor; Charles Cowens, typesetter; Jeff Green, proofreader; Jeffrey James Giese, technical illustrator; Suzy Anger, production coordinator; Evelyn Ong Sy and Jenny Wong, paste-up artists; and Paula Alston, indexer. We would also like to acknowledge the work of Skillful Means, typesetters for Chapter 3.

Introduction

THIS BOOK PRESENTS THE ASSEMBLY LANGUAGE PROGRAMMER'S VIEW OF the 80386, the latest member of the popular Intel 86 family of microcomputers. Throughout the book we focus on the 32-bit features of the chip. The 80386 is entirely compatible with the 8086 and 80286, and we summarize these features in Chapter 9. In addition to complete coverage of the 80386, we also cover the 80387, the numerics coprocessor of the 80386. Rather than presenting the 80387 in an appendix or in a separate chapter, as many books do, we present it in an integrated fashion.

Having spent years developing the chip itself, we are pleased to present the *insider's view* of how to program and use the 80386. Throughout the book, we have strived to be accurate and authoritative, as only the chip designers could be.

An important question to answer is: why should you be reading this book? Why will your understanding and programming of the 80386 benefit you for the next decade or two of your programming career? The answer is in the tremendous cumulative investment in the 86 family. To design and use a computer, investments are continually made. These investments are by those designing computers (IBM PC, PC/AT, PS/2), operating systems (UNIX, MS-DOS), programming languages (C, FORTRAN), application programs (Lotus 1-2-3, MultiMate, dBASE III), and additional hardware (graphics, extra disks, network connections, add-on memory). The investments also include programs you may write yourself and, of course, your time to learn. Thus, computer families, such as the 86, evolve and share compatibility from one generation to the next. This compatibility allows the use and leverage of massive investments already made into a computer family.

We assume you have experience in the basic theory of computer operations. We also assume this is not your first assembly language experience. We thus purposely avoid these introductory topics and recommend the less experienced reader in these areas to first read an introductory

text. The book is divided into roughly three parts. Chapters 1–4 present the applications programmer's view of the 80386. Applications programmers can limit their reading to these chapters with little loss in completeness. Chapters 5–7 present the operating-system programmer's view of the 80386. These chapters are less tutorial than Chapters 1–4 and conclude with reference material on the detailed operation of the operating-system facilities. Chapters 5–7 are required reading for the operating-system programmer. Chapters 8 and 9 pick up the loose ends: debugging and 80386 compatibility with the 8086 and 80286. A more detailed description of each chapter follows.

In **Chapter 1,** we give a brief introduction to the 8086 family of processors. We also present other introductory items, such as memory organization and number representations. The bulk of the chapter is dedicated to the data types supported by the 80386 and 80387.

In **Chapter 2,** we present the internal machine state, general registers, processor control registers, and segment registers of the 80386. This is followed by an introduction to memory addressing. Instruction encodings and I/O space addressing are next presented. The chapter concludes with the 80387 internal machine state, general registers, and control registers.

Chapter 3, the most voluminous of the book, presents every instruction of the 80386 and 80387. The instruction presentation is broken into four sections: integer instructions, multiple-segment instructions, instructions for the operating-system writer, and instructions that operate on floating-point data.

Chapter 4 presents several examples of the applications programmer's instructions and machine state. It summarizes the applications programmer's view of the 80386 and 80387.

Chapter 5 presents the memory-management, protection, and multitasking facilities of the 80386. Several registers and system segments used by these facilities are introduced here rather than in Chapter 2. The chapter includes the exact semantics of all segmentation, memory access, control-transfer, and task-switching operations.

Chapter 6 presents the interrupts and exceptions of the 80386 and how they are processed. This includes the priorities of interrupts, how they are masked, and details of control transfers during interrupt processing. As in Chapter 5, an authoritative presentation of the interrupt and exception processing details is given. The chapter concludes with the 80387 exception causes and methods of processing.

Chapter 7 presents examples of the operating-system facilities of the 80386. These examples demonstrate many of the segmentation, paging,

and exception facilities discussed in Chapters 5 and 6, and the operating-system and multiple-segment instructions of Chapter 3.

Chapter 8 presents the facilities included in the 80386 specifically to support debugging.

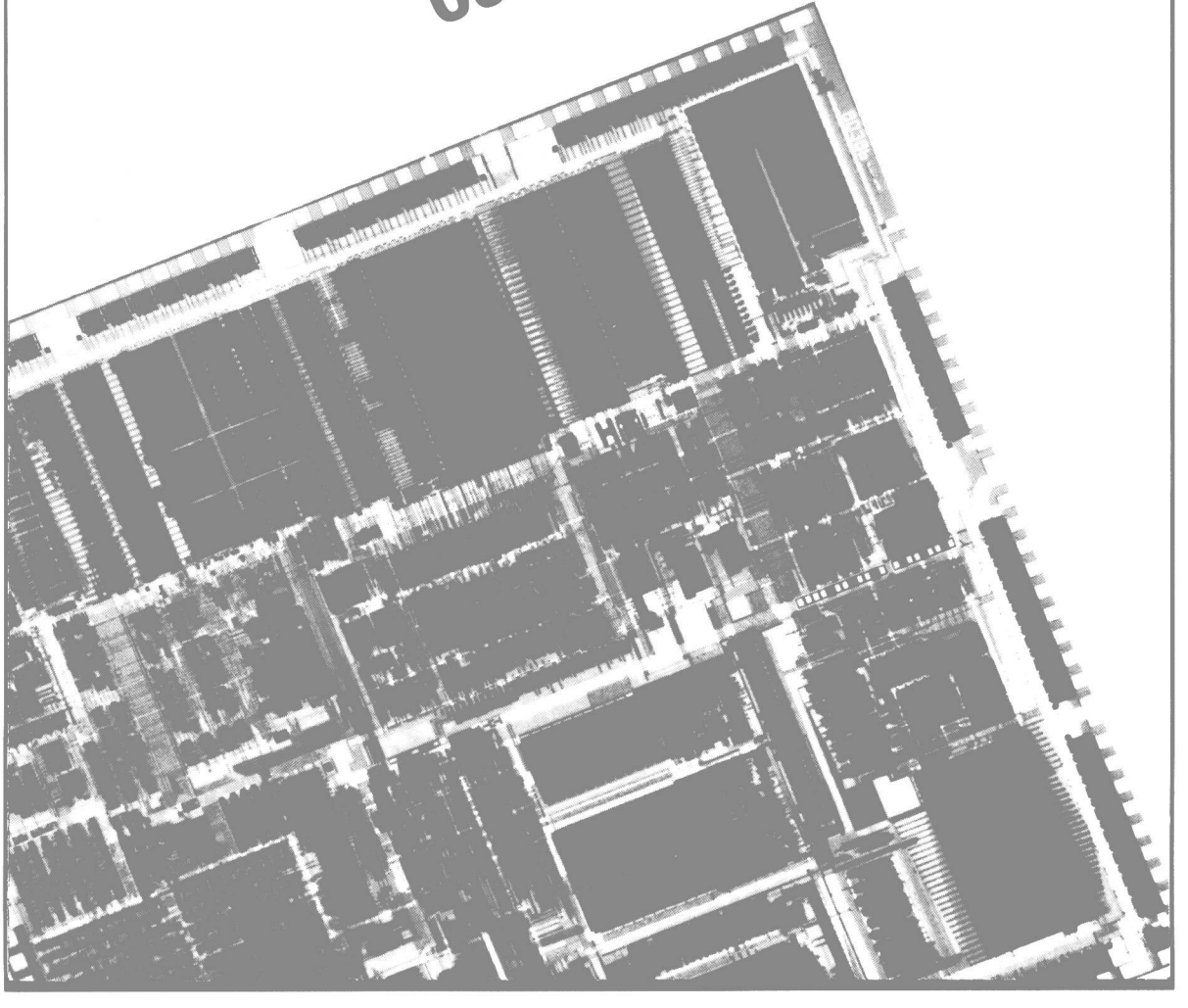
Chapter 9 takes a step backward and discusses executing 16-bit code on the 80386. This includes descriptions of real (8086), virtual-8086, and protected 16-bit modes of operation.

The following references provide additional material on the 80386 and 80387. Since we make no mention of the hardware aspects of the 80386, items 2 and 3 below are particularly useful, as they cover this area.

1. *80386 Programmer's Reference Manual,* Intel Corporation, Order No. 230985.
2. *80386 Hardware Reference Manual,* Intel Corporation, Order No. 231732.
3. *80386 Data Sheet,* Intel Corporation, Order No. 231630.
4. *80386 Assembly Language Reference Manual,* Intel Corporation, Order No. 122332.
5. *80387 Data Sheet,* Intel Corporation, Order No. 231920.

With this brief introduction, let's begin our study of the 80386. It will be challenging, but not without reward, as you make an investment in a very popular microcomputer family.

Basic Concepts



SYBEX Computer Books are different.

Here is why . . .

At SYBEX, each book is designed with you in mind. Every manuscript is carefully selected and supervised by our editors, who are themselves computer experts. We publish the best authors, whose technical expertise is matched by an ability to write clearly and to communicate effectively. Programs are thoroughly tested for accuracy by our technical staff. Our computerized production department goes to great lengths to make sure that each book is well-designed.

In the pursuit of timeliness, SYBEX has achieved many publishing firsts. SYBEX was among the first to integrate personal computers used by authors and staff into the publishing process. SYBEX was the first to publish books on the CP/M operating system, microprocessor interfacing techniques, word processing, and many more topics.

Expertise in computers and dedication to the highest quality product have made SYBEX a world leader in computer book publishing. Translated into fourteen languages, SYBEX books have helped millions of people around the world to get the most from their computers. We hope we have helped you, too.

For a complete catalog of our publications:

SYBEX, Inc. 2021 Challenger Drive, #100, Alameda, CA 94501
Tel: (415) 523-8233/(800) 227-2346 Telex: 336311

Contents

Introduction xviii

Chapter 1 1
The Basics

History of Intel Microprocessors 1
Compatibility with the 8086 and 80286 2

Data Formats 3
Memory 3
Notation 4
Unsigned Numbers 5
Signed Integers 5
Strings 9
Bits 11
BCD 12

Floating-Point Data Types 14
Introduction to Floating Point 14
IEEE Floating-Point Standard 15
What If the 80387 Is Missing? 16
Data Formats 17
Integer Data Types 18
BCD 19
Real Formats 20
Temporary Reals 23
Special Cases 23
Exceptions 29

Chapter 2 **33**

Machine State and Memory Addressing

Registers 34
The General Registers 35
The Processor-Control Registers 36
Segment Registers 41

Memory Addressing Concepts 42
Two-Part Addressing 42
Notation 43

Memory Addressing Mechanism 43
The Segment Part: Segment Register 44
The Offset Part: Address Modes 47
Program Stack 48
Pointer Data Type 53
Address Modes and Data Structures 54
Segmentation Strategies 56

Instruction Encoding 59
Immediate Constants 61
Register Operands 63
Memory Operands 66

I/O Space 76

Floating-Point Registers 77
Floating-Point Accumulator Stack 77
Sixteen-Bit Status and Control Registers 80
Error-Pointer Registers 86

Chapter 3 **91**

Instruction Set

Table of Contents for Chapter 3 91

Alphabetical Index to Instructions 98

Instruction Description Format 105

Integer 119

Multiple Segment 269

Operating System 290

Floating Point 320

Chapter 4 **401**

Instruction Set Examples

Syntax 401

Integer Examples 405
Signed Divide 405
Sort 406
Factorial 408
Semaphore 411
String Search 412
Bit Block Transfer 415

Floating-Point Examples 420
Floating-Point Flags 420
Partial Remainder 422
Exponential Computations 422
Matrix Multiplication 423
Statistics 426

Chapter 5 **431**

Memory Management, Protection, and Tasks

Memory-Management Facilities 433
Address Translation 433
Protection 437

Segmentation 446
Segment Descriptor Tables 448

Segment Selectors 451
Segment Descriptors 453

Paging 463
Page Table Structure 465
Page Table Entry Format 470
Virtual Memory 473
Page-Level Protection 473
Software Issues in Modifying Page Table Entries 475

Processor-Control Registers and System Segments 477
Processor-Control Registers 477
Segmentation Table Base Registers 481
Task State Segment Format 483

Instructions Sensitive to Privilege Level 488
Privileged Instructions 489
I/O Space Protection 490
Instructions That Change EFLAGS 496

Control-Transfer Methods 497
Same Level, Same Task 497
Different Level, Same Task 498
Outward Returns 504

Segmentation Details 505
Exceptions Summary 506
Memory Data Access Details 509
Control-Transfer Details 528
Task Switches 540

Chapter 6 **553**
Interrupts and Exceptions

Interrupts 554
INTR Interrupts 555
NMI 555

Exceptions 555
Instruction Restart 557