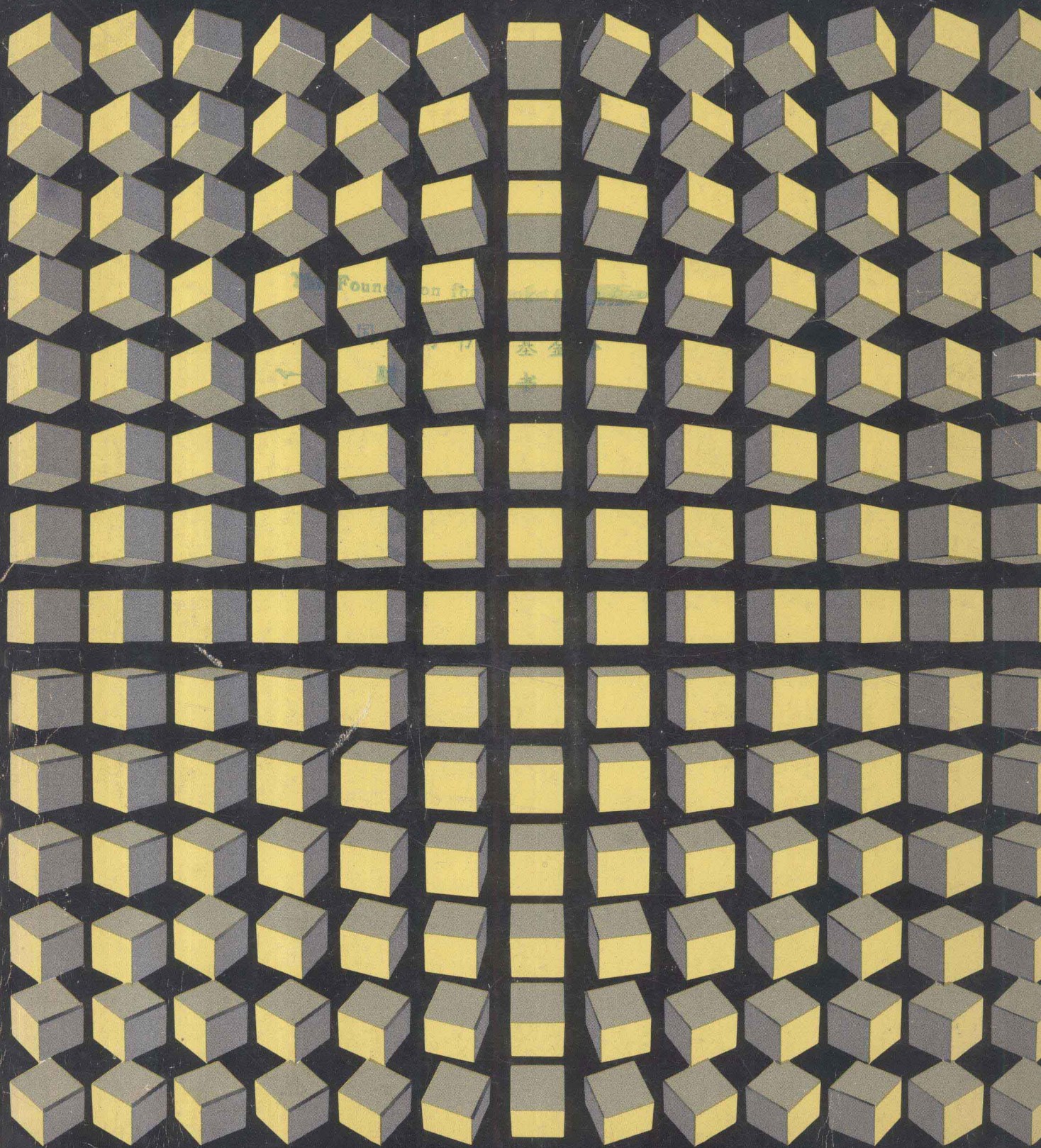
Introduction to Assembler Language Programming

Carl Feingol



Introduction to Assembler Language Programming

Carl Feingold, C.P.A., C.D.P.

West Los Angeles College, Culver City, California

Wm. C. Brown Company Publishers
Dubuque, Iowa

ISBN 0—697—08124—9

Printed in the United States of America

Introduction to Assembler Language Programming

Preface

The Assembler programming language is tailored to take full advantage of the features of the IBM system 360/370, the most widely used family of computers. The choice of a programming language can be governed, at least in part, by the degree of efficiency one seeks in an object program compared to the time that will be required to write the program. Assembler language provides a source statement for each instruction in the 360/370 machine language. Of the various symbolic programming languages, Assembler language is closest to machine language in format and content. It permits the programmer to use all system 360/370 functions as if the programmer were coding in 360/370 machine language. One has to write only the appropriate source statements and the special features can be utilized. High level programming languages, such as FORTRAN and COBOL, do not provide these features since they are written for a standard configuration and generally do not have the ability to compile statements that activate these special machine features.

The effective use of the Assembler language in the hands of a skilled programmer can save both computer storage and time. A knowledge of Assembler programming has important benefits for the programmer working in a high level language. It can be helpful to the programmer in analyzing and debugging programs as well as including certain language routines in the program to meet special systems or other requirements. For this and other reasons, Assembler language is the principal language used in writing software programs for compilers, supervisory programs, and many other programs that are to be used repeatedly.

The purpose of the text is to provide the general reader with a broad and comprehensive coverage of the features of Assembler language. In a systematic fashion, the reader is introduced to the basic computer concepts, computer programming, and the features of the IBM system 360/370 and proceeds early in the text to writing simple programs thus gathering information relative to the format of compiler listings, diagnostics, dumps, and other basic essentials of computer programming. All the salient features of Assembler programming are explained and illustrated through numerous illustrations. Each illustration is complete within itself so that the reader does not have to "wade"

through numerous pages of narrative for explanations of the illustration. The text is comprehensive providing the reader with the introductory concepts through the basic components of Assembler language programming and advanced programming concepts. The text is meant to be all inclusive, with little or no need, for reference to the numerous reference manuals. Some of the important features are,

Readers Are Given the Basic Tools of Programming with Which They Can Carry Out More Complex Procedural Techniques.

After a concise explanation of the components of the computer, such as Input/Output devices, the Central Processing unit, and storage elements, readers receive a step-by-step introduction to the process of programming. Each step is clearly illustrated with sample programs and flowcharts. Then, after an explanation of the unique features of the system 360/370 statements and Assembler language, readers are given in-depth instructions for writing simple Assembler language programs, with definitions of storage areas, instruction statements, and types of constants.

Clear, Concise Instructions Allow the Reader to Grasp Difficult Programming Procedures.

All subjects in Assembler language are covered in sufficient detail to reduce or eliminate the need for outside information from computer manufacturers reference manuals. Topics include rounding and editing in decimal operations; logical operations on characters and bits such as comparing, testing, translating, and shifting; fixed point procedures commonly used in scientific and engineering calculations but being used more and more in business applications; and subroutines and subprograms.

Comprehensive Coverage of Major Concepts Give the Reader the Ability to More Fully Utilize the Capabilities of the System 360/370.

Directions for programming table and array handling are reinforced by detailed sample programs for each function. Included in these samples are the input data, outline of the computations, and a copy of the printed program and generated output. A chapter on magnetic tape and direct access (disk, drum, and data cell) processing includes typical input and output macros with directions for several types of operations, such as creating, modifying, and adding to sequential, indexed sequential, and direct data sets.

Over 400 Illustrations Help the Reader to Visualize Technical Concepts.

Line drawings, sample programs, charts, and graphs all combine to provide helpful examples of the topics discussed. The sample programs illustrate procedures outlined in the text and make it easier for the reader to grasp the details of actual Assembler language programming. Each instruction is complete with detailed explanations for its use, together with illustrations showing how it is written with resulting output.

Study Aids in Each Chapter Help the Reader to Learn the Concepts as They Are Being Presented.

Helpful study aids, such as review questions, fill-ins, matching, flowcharting problems, multiple choice, and of course programming problems, help to test the level of reader comprehension of various mechanical and conceptual procedures.

Extensive Appendices Provide the Reader with Useful Reference Material.

In addition to programming problems, the appendices provide instruction formats, hexadecimal number system calculation and tables, job control card formats, examples of computer printouts, dumps, and debugging procedures.

The DOS and OS input/output configurations are given early in the text so that the reader can proceed with program writing without being concerned with the different formats for each system.

The text is so arranged so that the first half (chapters 1-7) may be used as an introductory course in Assembler language programming with the second half (chapters 8-15) being used as an advanced Assembler language programming course depending on the needs and objectives of the instructor.

Chapters 1 and 2 provide an insight to the operation of the computer and the numerous problems encountered in the study, planning, and preparation of computer programs. This chapter can serve as a review for those who have had an intro-

ductory course. For those who have not had a course, the chapters will provide the necessary background material for Assembler language programming.

Chapters 3 and 4 present an overview of the system 360/370 features with the basic programming elements for writing Assembler programs. This knowledge provides a background for later chapters in which many of the instructions in the system 360/370 instruction set are introduced as well as illustrated by sample Assembler language programs.

Chapter 5 provides the basic macros for input/output operations for both DOS and OS systems. Since most DOS and OS input/output configurations are standard for each installation, the reader can copy the necessary input/output statements applicable to his particular installation verbatim and use them in the early programs. Later on, when the more advanced concepts are discussed, these statements can be modified to suit the individual needs thus eliminating the need for separate discussions of DOS and OS systems. The reader should be able to write his first program.

Chapters 6 and 7 discuss the important decimal features of the Assembler language. The decimal feature provides a series of instructions that permit arithmetic operations to be performed without first converting the data to binary format. Data processed by these instructions may be in fields of varying lengths, starting at any address in storage and includes instructions for editing data (preparing data for printing by the insertion of characters such as $, etc.).

Chapters 8 and 9 explain the logical operations on characters and bits which can only be performed in Assembler language programming. Logical operations permit bit-by-bit operations in making decisions and bit operations on sequence of characters. Bit manipulation features are only available to Assembler language programmers and cannot be performed on high level languages such as FORTRAN and COBOL.

Chapters 10 and 11 discuss fixed point operations which involve registers. Many important procedures in Assembler language can be performed using registers as they are an important part of the Assembler language. Using fixed point format requires that data be in binary format and on fixed boundaries in storage. These include the arithmetic instructions as the central topic with important consideration also given to certain logical operations (comparing, branching, etc.) and loop methods.

Decision making and branching are important parts of data processing and the programming methods by which these operations are carried out are important aspects of the programming task.

Chapters 12 through 15 provide the advanced concepts of Subroutines, subprograms, Macro Language, Table Handling, Magnetic Tape and Direct Access applications.

In the appendices, the important features of operation codes, instruction formats, hexadecimal calculations and tables, character codes, control characters, condition code settings, assembler instructions, job control language, debugging a program, dump and a system 370 reference summary are provided.

There are exercises, questions for review, and problems behind each chapter, as well as programming problems. The exercises have answers so that the reader can have immediate "feedback" to reinforce the learning process. The questions for review and problems can be assigned to fulfill the needs of the instructor. The instructor may assign programming problems as term projects.

The instructors manual will include a brief commentary of each chapter. The commentary states the intent of the chapter, useful teaching suggestions, and a summary of the important points. The instructors manual also provides answers for each of the questions for review, and all answers to problems, and program assignments.

I am indebted to the IBM Corporation for gratuitously granting permission to use the numerous illustrations, charts, photos, and diagrams that made the text more meaningful.

My special thanks to my wife Sylvia, who served as chief typist, confidant, proofreader, and without whose encouragement, the book would never have been written.

This section may be detached for the convenience of the student.

IBM

System/370 Reference Summary

GX20-1850-3

Fourth Edition (November 1976)

This reference summary is a minor revision and does not obsolete the previous edition. Changes include the addition of some new DASD and 3203 printer commands, the EBCDIC control characters GE and RLF, and minor editorial revisions.

The card is intended primarily for use by S/370 assembler language application programmers. It contains basic machine information on Models 115 through 168 summarized from the *System/370 Principles of Operation* (GA22-7000-4), frequently used information from the VS and VM assembler language manual (GC33-4010), command codes for various I/O devices, and a multi-code translation table. The card will be updated from time to time. However, the above manuals and others cited on the card are the authoritative reference sources and will be first to reflect changes.

To distinguish them from instructions carried over from S/360, the names of instructions essentially new with S/370 are shown in italics. Some machine instructions are optional or not available for some models. For those that are available on a particular model, the user is referred to the appropriate systems reference manual. For a particular installation, one must ascertain which optional hardware features and programming system(s) have been installed. The floating-point and extended floating-point instructions, as well as the instructions listed below, are not standard on every model. Monitoring (the MC instruction) is not available on the Model 165, except by field installation on purchased models.

Conditional swapping	CDS, CS
CPU timer and clock comparator	SCKC, SPT, STCKC, STPT
Direct control	RDD, WRD
Dynamic address translation	LRA, PTLB, RRB, STNSM, STOSM
Input/output	CLRIO, SIOF
Multiprocessing	SIGP, SPX, STAP, STPX
PSW key handling	IPK, SPKA

Comments about this publication may be sent to the address below. All comments and suggestions become the property of IBM.

IBM Corporation, Technical Publications/Systems, Dept. 824, 1133 Westchester Avenue, White Plains, N.Y. 10604.

MACHINE INSTRUCTIONS

(2)

NAME	MNEMONIC	OP CODE	FOR MAT	OPERANDS
Add (c)	AR	1A	RR	R1,R2
Add (c)	A	5A	RX	R1,D2(X2,B2)
Add Decimal (c)	AP	FA	SS	D1(L1,B1),D2(L2,B2)
Add Halfword (c)	AH	4A	RX	R1,D2(X2,B2)
Add Logical (c)	ALR	1E	RR	R1,R2
Add Logical (c)	AL	5E	RX	R1,D2(X2,B2)
AND (c)	NR	14	RR	R1,R2
AND (c)	N	54	RX	R1,D2(X2,B2)
AND (c)	NI	94	SI	D1(B1),I2
AND (c)	NC	D4	SS	D1(L,B1),D2(B2)
Branch and Link	BALR	05	RR	R1,R2
Branch and Link	BAL	45	RX	R1,D2(X2,B2)
Branch on Condition	BCR	07	RR	M1,R2
Branch on Condition	BC	47	RX	M1,D2(X2,B2)
Branch on Count	BCTR	06	RR	R1,R2
Branch on Count	BCT	46	RX	R1,D2(X2,B2)
Branch on Index High	BXH	86	RS	R1,R3,D2(B2)
Branch on Index Low or Equal	BXLE	87	RS	R1,R3,D2(B2)
Clear I/O (c,p)	CLRIO	9D01	S	D2(B2)
Compare (c)	CR	19	RR	R1,R2
Compare (c)	C	59	RX	R1,D2(X2,B2)
Compare and Swap (c)	CS	BA	RS	R1,R3,D2(B2)
Compare Decimal (c)	CP	F9	SS	D1(L1,B1),D2(L2,B2)
Compare Double and Swap (c)	CDS	BB	RS	R1,R3,D2(B2)
Compare Halfword (c)	CH	49	RX	R1,D2(X2,B2)
Compare Logical (c)	CLR	15	RR	R1,R2
Compare Logical (c)	CL	55	RX	R1,D2(X2,B2)
Compare Logical (c)	CLC	D5	SS	D1(L,B1),D2(B2)
Compare Logical (c)	CLI	95	SI	D1(B1),I2
Compare Logical Characters under Mask (c)	CLM	BD	RS	R1,M3,D2(B2)
Compare Logical Long (c)	CLCL	0F	RR	R1,R2
Convert to Binary	CVB	4F	RX	R1,D2(X2,B2)
Convert to Decimal	CVD	4E	RX	R1,D2(X2,B2)
Diagnose (p)		83		Model-dependent
Divide	DR	1D	RR	R1,R2
Divide	D	5D	RX	R1,D2(X2,B2)
Divide Decimal	DP	FD	SS	D1(L1,B1),D2(L2,B2)
Edit (c)	ED	DE	SS	D1(L,B1),D2(B2)
Edit and Mark (c)	EDMK	DF	SS	D1(L,B1),D2(B2)
Exclusive OR (c)	XR	17	RR	R1,R2
Exclusive OR (c)	X	57	RX	R1,D2(X2,B2)
Exclusive OR (c)	XI	97	SI	D1(B1),I2
Exclusive OR (c)	XC	D7	SS	D1(L,B1),D2(B2)
Execute	EX	44	RX	R1,D2(X2,B2)
Halt I/O (c,p)	HIO	9E00	S	D2(B2)
Halt Device (c,p)	HDV	9E01	S	D2(B2)
Insert Character	IC	43	RX	R1,D2(X2,B2)
Insert Characters under Mask (c)	ICM	BF	RS	R1,M3,D2(B2)
Insert PSW Key (p)	IPK	B20B	S	
Insert Storage Key (p)	ISK	09	RR	R1,R2
Load	LR	18	RR	R1,R2
Load	L	58	RX	R1,D2(X2,B2)
Load Address	LA	41	RX	R1,D2(X2,B2)
Load and Test (c)	LTR	12	RR	R1,R2
Load Complement (c)	LCR	13	RR	R1,R2
Load Control (p)	LCTL	B7	RS	R1,R3,D2(B2)
Load Halfword	LH	48	RX	R1,D2(X2,B2)
Load Multiple	LM	98	RS	R1,R3,D2(B2)
Load Negative (c)	LNR	11	RR	R1,R2
Load Positive (c)	LPR	10	RR	R1,R2
Load PSW (n,p)	LPSW	82	S	D2(B2)
Load Real Address (c,p)	LRA	B1	RX	R1,D2(X2,B2)
Monitor Call	MC	AF	SI	D1(B1),I2
Move	MVI	92	SI	D1(B1),I2
Move	MVC	D2	SS	D1(L,B1),D2(B2)
Move Long (c)	MVCL	0E	RR	R1,R2
Move Numerics	MVN	D1	SS	D1(L,B1),D2(B2)
Move with Offset	MVO	F1	SS	D1(L1,B1),D2(L2,B2)
Move Zones	MVZ	D3	SS	D1(L,B1),D2(B2)
Multiply	MR	1C	RR	R1,R2
Multiply	M	5C	RX	R1,D2(X2,B2)
Multiply Decimal	MP	FC	SS	D1(L1,B1),D2(L2,B2)
Multiply Halfword	MH	4C	RX	R1,D2(X2,B2)
OR (c)	OR	16	RR	R1,R2

MACHINE INSTRUCTIONS (Contd)

NAME	MNEMONIC	OP CODE	FOR MAT	OPERANDS
OR (c)	O	56	RX	R1,D2(X2,B2)
OR (c)	OI	96	SI	D1(B1),I2
OR (c)	OC	D6	SS	D1(L,B1),D2(B2)
Pack	PACK	F2	SS	D1(L1,B1),D2(L2,B2)
Purge TLB (p)	PTLB	B20D	S	
Read Direct (p)	RDD	85	SI	D1(B1),I2
Reset Reference Bit (c,p)	RRB	B213	S	D2(B2)
Set Clock (c,p)	SCK	B204	S	D2(B2)
Set Clock Comparator (p)	SCKC	B206	S	D2(B2)
Set CPU Timer (p)	SPT	B208	S	D2(B2)
Set Prefix (p)	SPX	B210	S	D2(B2)
Set Program Mask (n)	SPM	04	RR	R1
Set PSW Key from Address (p)	SPKA	B20A	S	D2(B2)
Set Storage Key (p)	SSK	08	RR	R1,R2
Set System Mask (p)	SSM	80	S	D2(B2)
Shift and Round Decimal (c)	SRP	F0	SS	D1(L1,B1),D2(B2),I3
Shift Left Double (c)	SLDA	8F	RS	R1,D2(B2)
Shift Left Double Logical	SLDL	8D	RS	R1,D2(B2)
Shift Left Single (c)	SLA	8B	RS	R1,D2(B2)
Shift Left Single Logical	SLL	89	RS	R1,D2(B2)
Shift Right Double (c)	SRDA	8E	RS	R1,D2(B2)
Shift Right Double Logical	SRDL	8C	RS	R1,D2(B2)
Shift Right Single (c)	SRA	8A	RS	R1,D2(B2)
Shift Right Single Logical	SRL	88	RS	R1,D2(B2)
Signal Processor (c,p)	SIGP	AE	RS	R1,R3,D2(B2)
Start I/O (c,p)	SIO	9C00	S	D2(B2)
Start I/O Fast Release (c,p)	SIOF	9C01	S	D2(B2)
Store	ST	50	RX	R1,D2(X2,B2)
Store Channel ID (c,p)	STIDC	B203	S	D2(B2)
Store Character	STC	42	RX	R1,D2(X2,B2)
Store Characters under Mask	STCM	BE	RS	R1,M3,D2(B2)
Store Clock (c)	STCK	B205	S	D2(B2)
Store Clock Comparator (p)	STCKC	B207	S	D2(B2)
Store Control (p)	STCTL	B6	RS	R1,R3,D2(B2)
Store CPU Address (p)	STAP	B212	S	D2(B2)
Store CPU ID (p)	STIDP	B202	S	D2(B2)
Store CPU Timer (p)	STPT	B209	S	D2(B2)
Store Halfword	STH	40	RX	R1,D2(X2,B2)
Store Multiple	STM	90	RS	R1,R3,D2(B2)
Store Prefix (p)	STPX	B211	S	D2(B2)
Store Then AND System Mask (p)	STNSM	AC	SI	D1(B1),I2
Store Then OR System Mask (p)	STOSM	AD	SI	D1(B1),I2
Subtract (c)	SR	1B	RR	R1,R2
Subtract (c)	S	5B	RX	R1,D2(X2,B2)
Subtract Decimal (c)	SP	FB	SS	D1(L1,B1),D2(L2,B2)
Subtract Halfword (c)	SH	4B	RX	R1,D2(X2,B2)
Subtract Logical (c)	SLR	1F	RR	R1,R2
Subtract Logical (c)	SL	5F	RX	R1,D2(X2,B2)
Supervisor Call	SVC	0A	RR	I
Test and Set (c)	TS	93	S	D2(B2)
Test Channel (c,p)	TCH	9F00	S	D2(B2)
Test I/O (c,p)	TIO	9D00	S	D2(B2)
Test under Mask (c)	TM	91	SI	D1(B1),I2
Translate	TR	DC	SS	D1(L,B1),D2(B2)
Translate and Test (c)	TRT	DD	SS	D1(L,B1),D2(B2)
Unpack	UNPK	F3	SS	D1(L1,B1),D2(L2,B2)
Write Direct (p)	WRD	84	SI	D1(B1),I2
Zero and Add Decimal (c)	ZAP	F8	SS	D1(L1,B1),D2(L2,B2)

Floating-Point Instructions

NAME	MNEMONIC	OP CODE	FOR MAT	OPERANDS
Add Normalized, Extended (c,x)	AXR	36	RR	R1,R2
Add Normalized, Long (c)	ADR	2A	RR	R1,R2
Add Normalized, Long (c)	AD	6A	RX	R1,D2(X2,B2)
Add Normalized, Short (c)	AER	3A	RR	R1,R2
Add Normalized, Short (c)	AE	7A	RX	R1,D2(X2,B2)
Add Unnormalized, Long (c)	AWR	2E	RR	R1,R2
Add Unnormalized, Long (c)	AW	6E	RX	R1,D2(X2,B2)
Add Unnormalized, Short (c)	AUR	3E	RR	R1,R2
Add Unnormalized, Short (c)	AU	7E	RX	R1,D2(X2,B2)

c. Condition code is set.
n. New condition code is loaded.
p. Privileged instruction.
x. Extended precision floating-point.

Floating-Point Instructions (Contd)

NAME	MNEMONIC	OP CODE	FOR MAT	OPERANDS
Compare, Long (c)	CDR	29	RR	R1,R2
Compare, Long (c)	CD	69	RX	R1,D2(X2,B2)
Compare, Short (c)	CER	39	RR	R1,R2
Compare, Short (c)	CE	79	RX	R1,D2(X2,B2)
Divide, Long	DDR	2D	RR	R1,R2
Divide, Long	DD	6D	RX	R1,D2(X2,B2)
Divide, Short	DER	3D	RR	R1,R2
Divide, Short	DE	7D	RX	R1,D2(X2,B2)
Halve, Long	HDR	24	RR	R1,R2
Halve, Short	HER	34	RR	R1,R2
Load and Test, Long (c)	LTDR	22	RR	R1,R2
Load and Test, Short (c)	LTER	32	RR	R1,R2
Load Complement, Long (c)	LCDR	23	RR	R1,R2
Load Complement, Short (c)	LCER	33	RR	R1,R2
Load, Long	LDR	28	RR	R1,R2
Load, Long	LD	68	RX	R1,D2(X2,B2)
Load Negative, Long (c)	LNDR	21	RR	R1,R2
Load Negative, Short (c)	LNER	31	RR	R1,R2
Load Positive, Long (c)	LPDR	20	RR	R1,R2
Load Positive, Short (c)	LPER	30	RR	R1,R2
Load Rounded, Extended to Long (x)	LRDR	25	RR	R1,R2
Load Rounded, Long to Short (x)	LRER	35	RR	R1,R2
Load, Short	LER	38	RR	R1,R2
Load, Short	LE	78	RX	R1,D2(X2,B2)
Multiply, Extended (x)	MXR	26	RR	R1,R2
Multiply, Long	MDR	2C	RR	R1,R2
Multiply, Long	MD	6C	RX	R1,D2(X2,B2)
Multiply, Long/Extended (x)	MXDR	27	RR	R1,R2
Multiply, Long/Extended (x)	MXD	67	RX	R1,D2(X2,B2)
Multiply, Short	MER	3C	RR	R1,R2
Multiply, Short	ME	7C	RX	R1,D2(X2,B2)
Store, Long	STD	60	RX	R1,D2(X2,B2)
Store, Short	STE	70	RX	R1,D2(X2,B2)
Subtract Normalized, Extended (c,x)	SXR	37	RR	R1,R2
Subtract Normalized, Long (c)	SDR	2B	RR	R1,R2
Subtract Normalized, Long (c)	SD	6B	RX	R1,D2(X2,B2)
Subtract Normalized, Short (c)	SER	3B	RR	R1,R2
Subtract Normalized, Short (c)	SE	7B	RX	R1,D2(X2,B2)
Subtract Unnormalized, Long (c)	SWR	2F	RR	R1,R2
Subtract Unnormalized, Long (c)	SW	6F	RX	R1,D2(X2,B2)
Subtract Unnormalized, Short (c)	SUR	3F	RR	R1,R2
Subtract Unnormalized, Short (c)	SU	7F	RX	R1,D2(X2,B2)

EXTENDED MNEMONIC INSTRUCTIONS†

Use	Extended Code* (RX or RR)	Meaning	Machine Instr.* (RX or RR)
General	B or BR	Unconditional Branch	BC or BCR 15,
	NOP or NOPR	No Operation	BC or BCR 0,
After Compare Instructions (A:B)	BH or *BHR*	Branch on A High	BC or BCR 2,
	BL or *BLR*	Branch on A Low	BC or BCR 4,
	BE or *BER*	Branch on A Equal B	BC or BCR 8,
	BNH or *BNHR*	Branch on A Not High	BC or BCR 13,
	BNL or *BNLR*	Branch on A Not Low	BC or BCR 11,
	BNE or *BNER*	Branch on A Not Equal B	BC or BCR 7,
After Arithmetic Instructions	BO or *BOR*	Branch on Overflow	BC or BCR 1,
	BP or *BPR*	Branch on Plus	BC or BCR 2,
	BM or *BMR*	Branch on Minus	BC or BCR 4,
	BNP or *BNPR*	Branch on Not Plus	BC or BCR 13,
	BNM or *BNMR*	Branch on Not Minus	BC or BCR 11,
	BNZ or *BNZR*	Branch on Not Zero	BC or BCR 7,
	BZ or *BZR*	Branch on Zero	BC or BCR 8,
After Test under Mask Instruction	BO or *BOR*	Branch if Ones	BC or BCR 1,
	BM or *BMR*	Branch if Mixed	BC or BCR 4,
	BZ or *BZR*	Branch if Zeros	BC or BCR 8,
	BNO or *BNOR*	Branch if Not Ones	BC or BCR 14,

†Source: GC33-4010; for OS/VS,VM/370 and DOS/VS.

*Second operand, not shown, is D2(X2,B2) for RX format and R2 for RR format.

SOME EDIT AND EDMK PATTERN CHARACTERS (in hex)

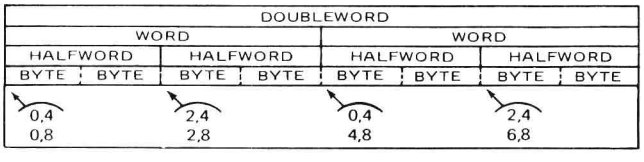
20—digit selector	40—blank	5C—asterisk
21—start of significance	4B—period	6B—comma
22—field separator	5B—dollar sign	C3D9—CR

CONDITION CODES

(5)

Condition Code Setting	0	1	2	3
Mask Bit Value	8	4	2	1
General Instructions				
Add, Add Halfword	zero	<zero	>zero	overflow
Add Logical	zero, no carry	not zero, no carry	zero, carry	not zero, carry
AND	zero	not zero	–	–
Compare, Compare Halfword	equal	1st op low	1st op high	–
Compare and Swap/Double	equal	not equal	–	–
Compare Logical	equal	1st op low	1st op high	–
Exclusive OR	zero	not zero	–	–
Insert Characters under Mask	all zero	1st bit one	1st bit zero	–
Load and Test	zero	<zero	>zero	–
Load Complement	zero	<zero	>zero	overflow
Load Negative	zero	<zero	–	–
Load Positive	zero	–	>zero	overflow
Move Long	count equal	count low	count high	overlap
OR	zero	not zero	–	–
Shift Left Double/Single	zero	<zero	>zero	overflow
Shift Right Double/Single	zero	<zero	>zero	–
Store Clock	set	not set	error	not oper
Subtract, Subtract Halfword	zero	<zero	>zero	overflow
Subtract Logical	–	not zero, no carry	zero, carry	not zero, carry
Test and Set	zero	one	–	–
Test under Mask	zero	mixed	–	ones
Translate and Test	zero	incomplete	complete	–
Decimal Instructions				
Add Decimal	zero	<zero	>zero	overflow
Compare Decimal	equal	1st op low	1st op high	–
Edit, Edit and Mark	zero	<zero	>zero	–
Shift and Round Decimal	zero	<zero	>zero	overflow
Subtract Decimal	zero	<zero	>zero	overflow
Zero and Add	zero	<zero	>zero	overflow
Floating-Point Instructions				
Add Normalized	zero	<zero	>zero	–
Add Unnormalized	zero	<zero	>zero	–
Compare	equal	1st op low	1st op high	–
Load and Test	zero	<zero	>zero	–
Load Complement	zero	<zero	>zero	–
Load Negative	zero	<zero	–	–
Load Positive	zero	–	>zero	–
Subtract Normalized	zero	<zero	>zero	–
Subtract Unnormalized	zero	<zero	>zero	–
Input/Output Instructions				
Clear I/O	no oper in progress	CSW stored	chan busy	not oper
Halt Device	interruption pending	CSW stored	channel working	not oper
Halt I/O	interruption pending	CSW stored	burst op stopped	not oper
Start I/O, SIOF	successful	CSW stored	busy	not oper
Store Channel ID	ID stored	CSW stored	busy	not oper
Test Channel	available	interruption pending	burst mode	not oper
Test I/O	available	CSW stored	busy	not oper
System Control Instructions				
Load Real Address	translation available	ST entry invalid	PT entry invalid	length violation
Reset Reference Bit	R=0, C=0	R=0, C=1	R=1, C=0	R=1, C=1
Set Clock	set	secure	–	not oper
Signal Processor	accepted	stat stored	busy	not oper

CNOP ALIGNMENT



ASSEMBLER INSTRUCTIONS†

(6)

Function	Mnemonic	Meaning
Data definition	DC	Define constant
	DS	Define storage
	CCW	Define channel command word
Program sectioning and linking	START	Start assembly
	CSECT	Identify control section
	DSECT	Identify dummy section
	DXD*	Define external dummy section
	CXD*	Cumulative length of external dummy section
	COM	Identify blank common control section
	ENTRY	Identify entry-point symbol
	EXTRN	Identify external symbol
	WXTRN	Identify weak external symbol
Base register assignment	USING	Use base address register
	DROP	Drop base address register
Control of listings	TITLE	Identify assembly output
	EJECT	Start new page
	SPACE	Space listing
	PRINT	Print optional data
Program Control	ICTL	Input format control
	ISEQ	Input sequence checking
	PUNCH	Punch a card
	REPRO	Reproduce following card
	ORG	Set location counter
	EQU	Equate symbol
	OPSYN*	Equate operation code
	*PUSH**	Save current PRINT or USING status
	*POP**	Restore PRINT or USING status
	LTORG	Begin literal pool
	CNOP	Conditional no operation
	COPY	Copy predefined source coding
	END	End assembly
Macro definition	MACRO	Macro definition header
	MNOTE	Request for error message
	MEXIT	Macro definition exit
	MEND	Macro definition trailer
Conditional assembly	ACTR	Conditional assembly loop counter
	AGO	Unconditional branch
	AIF	Conditional branch
	ANOP	Assembly no operation
	GBLA	Define global SETA symbol
	GBLB	Define global SETB symbol
	GBLC	Define global SETC symbol
	LCLA	Define local SETA symbol
	LCLB	Define local SETB symbol
	LCLC	Define local SETC symbol
	SETA	Set arithmetic variable symbol
	SETB	Set binary variable symbol
	SETC	Set character variable symbol

SUMMARY OF CONSTANTS†

TYPE	IMPLIED LENGTH, BYTES	ALIGNMENT	FORMAT	TRUNCATION/ PADDING
C	–	byte	characters	right
X	–	byte	hexadecimal digits	left
B	–	byte	binary digits	left
F	4	word	fixed-point binary	left
H	2	halfword	fixed-point binary	left
E	4	word	short floating-point	right
D	8	doubleword	long floating-point	right
L	16	doubleword	extended floating-point	right
P	–	byte	packed decimal	left
Z	–	byte	zoned decimal	left
A	4	word	value of address	left
Y	2	halfword	value of address	left
S	2	halfword	address in base-displacement form	–
V	4	word	externally defined address value	left
Q*	4	word	symbol naming a DXD or DSECT	left

†Source: GC33-4010; for OS/VS, VM/370, and DOS/VS.
*OS/VS and VM/370 only.

I/O COMMAND CODES

(7)

Standard Command Code Assignments (CCW bits 0-7)

xxxx 0000	Invalid	†††† ††01	Write
†††† 0100	Sense	†††† ††10	Read
xxxx 1000	Transfer in Channel	†††† ††11	Control
†††† 1100	Read Backward	0000 0011	Control No Operation

x—Bit ignored. †Modifier bit for specific type of I/O device

CONSOLE PRINTERS

Write, No Carrier Return	01	Sense	04
Write, Auto Carrier Return	09	Audible Alarm	0B
Read Inquiry	0A		

3504, 3505 CARD READERS/3525 CARD PUNCH

Source: GA21-9124

Command	Binary	Hex
Sense	0000 0100	04
Feed, Select Stacker	SS10 F011	
Read Only*	11D0 F010	
Diagnostic Read (invalid for 3504)	1101 0010	D2
Read, Feed, Select Stacker*	SSD0 F010	
Write RCE Format*	0001 0001	11
3504, 3505 only		
Write OMR Format†	0011 0001	31
3525 only		
Write, Feed, Select Stacker	SSD0 0001	
Print Line*	LLLL L101	

Bit Meanings

SS Stacker
00 1
01/10 2

F Format Mode
0 Unformatted
1 Formatted

D Data Mode
0 1—EBCDIC
1 2—Card image

L Line Position
5-bit binary value

*Special feature on 3525. †Special feature.

PRINTERS: 3211/3811 (GA24-3543), 3203/IPA, 1403*/2821 (GA24-3312)

	After Write	*Immed*
Space 1 Line	09	0B
Space 2 Lines	11	13
Space 3 Lines	19	1B
Skip to Channel 0†	–	83
Skip to Channel 1	89	8B
Skip to Channel 2	91	93
Skip to Channel 3	99	9B
Skip to Channel 4	A1	A3
Skip to Channel 5	A9	AB
Skip to Channel 6	B1	B3
Skip to Channel 7	B9	BB
Skip to Channel 8	C1	C3
Skip to Channel 9	C9	CB
Skip to Channel 10	D1	D3
Skip to Channel 11	D9	DB
Skip to Channel 12	E1	E3
Adv. to End of Sheet (3203 only)	5B	

Write without spacing	01
Sense	04
Load UCSB without folding	FB
Fold†	43
Unfold†	23
Load UCSB and Fold (exc. 3211)	F3
UCS Gate Load (1403 only)	EB
Load FCB (exc. 1403)	63
Block Data Check	73
Allow Data Check	7B
Read PLB†	02
Read UCSB†	0A
Read FCB†	12
Diag. Check Read (exc. 3203)	06
Diagnostic Write†	05
Raise Cover†	6B
Diagnostic Gate†	07
Diagnostic Read (1403 only)	02

*UCS special feature; IPA diagnostics are model-dependent. †3211 only.

3420/3803, 3410/3411 MAGNETIC TAPE

(**Indicates 3420 only)

See GA32-0020, -0021, -0022 for special features and functions of specific models.

Write	01
Read Forward	02
Read Backward	0C
Sense	04
Sense Reserve**	F4
Sense Release**	D4
Request Track-in-Error	1B
Loop Write-to-Read**	8B
Set Diagnose**	4B
Rewind	07
Rewind Unload	0F
Erase Gap	17
Write Tape Mark	1F
Backspace Block	27
Backspace File	2F
Forward Space Block	37
Forward Space File	3F
Data Security Erase**	97
Diagnostic Mode Set**	0B

Mode Set 1 (7-track)

Density	Parity	DC	Trans	Cmd
200	odd	on	off	13
		off	off	33
			on	3B
	even	off	off	23
			on	2B
556	odd	on	off	53
		off	off	73
			on	7B
	even	off	off	63
			on	6B
800	odd	on	off	93
		off	off	B3
			on	BB
	even	off	off	A3
			on	AB

Mode Set 2 (9-track), 800 bpi	CB
Mode Set 2 (9-track), 1600 bpi	C3
Mode Set 2 (9-track), 6250 bpi**	D3

DIRECT ACCESS STORAGE DEVICES

(8)

3330-3340-3350 SERIES (GA26-1592, -1617, -1619, -1620, -1638); **2305/2835** (GA26-1589); **2314, 2319** (GA26-3599, -1606)

See systems reference manuals for restrictions.

Command		MT Off	MT On*	Count
Control	Orient (c)	2B		Nonzero
	Recalibrate	13		Nonzero
	Seek	07		6
	Seek Cylinder	0B		6
	Seek Head	1B		6
	Space Count	0F		3 (a); nonzero (d)
	Set File Mask	1F		1
	Set Sector (a,f)	23		1
	Restore (executes as a no-op)	17		Nonzero
	Vary Sensing (c)	27		1
	Diagnostic Load (a)	53		1
	Diagnostic Write (a)	73		512
Search	Home Address Equal	39	B9	4
	Identifier Equal	31	B1	5
	Identifier High	51	D1	5
	Identifier Equal or High	71	F1	5
	Key Equal	29	A9	KL
	Key High	49	C9	KL
	Key Equal or High	69	E9	KL
	Key and Data Equal (d)	2D	AD	Number of bytes (including mask bytes) in search argument
	Key and Data High (d)	4D	CD	
	Key and Data Eq. or Hi (d)	6D	ED	
Continue Scan	Search Equal (d)	25	A5	
	Search High (d)	45	C5	
	Search High or Equal (d)	65	E5	
	Set Compare (d)	35	B5	
	Set Compare (d)	75	F5	
	No Compare (d)	55	D5	
Read	Home Address	1A	9A	5
	Count	12	92	8
	Record 0	16	96	Number of bytes to be transferred
	Data	06	86	
	Key and Data	0E	8E	
	Count, Key and Data	1E	9E	
	IPL	02		
	Multiple Count, Key, Data (b)	5E		> Max. track len.
	Sector (a,f)	22		1
Sense	Sense I/O	04		24 (a); 6 (d)
	Sense I/O Type (b)	E4		7
	Read, Reset Buffered Log (b)	A4		24
	Read Buffered Log (c)	24		128
	Device Release (e)	94		24 (a); 6 (d)
	Device Reserve (e)	B4		24 (a); 6 (d)
	Read Diagnostic Status 1 (a)	44		16 or 512
Write	Home Address	19		5, 7, or 11
	Record 0	15		8+KL+DL of RO
	Erase	11		8+KL+DL
	Count, Key and Data	1D		8+KL+DL
	Special Count, Key and Data	01		8+KL+DL
	Data	05		DL
	Key and Data	0D		KL+DL

* Code same as MT Off except as listed.
a. Except 2314, 2319.
b. 3330-3340-3350 series only.
c. 2305/2835 only.
d. 2314, 2319 only.
e. String switch or 2-channel switch required.
f. Special feature required on 3340.

GX20-1850-3

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
360 Hamilton Avenue, White Plains, New York 10601
(International)

Printed in U.S.A.

CODE TRANSLATION TABLE

Dec.	Hex	Instruction (RR)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code EBCDIC	Binary
0	00			NUL	NUL		12-0-1-8-9	0000 0000
1	01			SOH	SOH		12-1-9	0000 0001
2	02			STX	STX		12-2-9	0000 0010
3	03			ETX	ETX		12-3-9	0000 0011
4	04	SPM		PF	EOT		12-4-9	0000 0100
5	05	BALR		HT	ENQ		12-5-9	0000 0101
6	06	BCTR		LC	ACK		12-6-9	0000 0110
7	07	BCR		DEL	BEL		12-7-9	0000 0111
8	08	SSK		GE	BS		12-8-9	0000 1000
9	09	ISK		RLF	HT		12-1-8-9	0000 1001
10	0A	SVC		SMM	LF		12-2-8-9	0000 1010
11	0B			VT	VT		12-3-8-9	0000 1011
12	0C			FF	FF		12-4-8-9	0000 1100
13	0D			CR	CR		12-5-8-9	0000 1101
14	0E	MVCL		SO	SO		12-6-8-9	0000 1110
15	0F	CLCL		SI	SI		12-7-8-9	0000 1111
16	10	LPR		DLE	DLE		12-11-1-8-9	0001 0000
17	11	LNR		DC1	DC1		11-1-9	0001 0001
18	12	LTR		DC2	DC2		11-2-9	0001 0010
19	13	LCR		TM	DC3		11-3-9	0001 0011
20	14	NR		RES	DC4		11-4-9	0001 0100
21	15	CLR		NL	NAK		11-5-9	0001 0101
22	16	OR		BS	SYN		11-6-9	0001 0110
23	17	XR		IL	ETB		11-7-9	0001 0111
24	18	LR		CAN	CAN		11-8-9	0001 1000
25	19	CR		EM	EM		11-1-8-9	0001 1001
26	1A	AR		CC	SUB		11-2-8-9	0001 1010
27	1B	SR		CU1	ESC		11-3-8-9	0001 1011
28	1C	MR		IFS	FS		11-4-8-9	0001 1100
29	1D	DR		IGS	GS		11-5-8-9	0001 1101
30	1E	ALR		IRS	RS		11-6-8-9	0001 1110
31	1F	SLR		IUS	US		11-7-8-9	0001 1111
32	20	LPDR		DS	SP		11-0-1-8-9	0010 0000
33	21	LNDR		SOS	!		0-1-9	0010 0001
34	22	LTDR		FS	"		0-2-9	0010 0010
35	23	LCDR			#		0-3-9	0010 0011
36	24	HDR		BYP	$		0-4-9	0010 0100
37	25	LRDR		LF	%		0-5-9	0010 0101
38	26	MXR		ETB	&		0-6-9	0010 0110
39	27	MXDR		ESC	'		0-7-9	0010 0111
40	28	LDR			(		0-8-9	0010 1000
41	29	CDR			)		0-1-8-9	0010 1001
42	2A	ADR		SM	*		0-2-8-9	0010 1010
43	2B	SDR		CU2	+		0-3-8-9	0010 1011
44	2C	MDR			,		0-4-8-9	0010 1100
45	2D	DDR		ENQ	-		0-5-8-9	0010 1101
46	2E	AWR		ACK	.		0-6-8-9	0010 1110
47	2F	SWR		BEL	/		0-7-8-9	0010 1111
48	30	LPER			0		12-11-0-1-8-9	0011 0000
49	31	LNER			1		1-9	0011 0001
50	32	LTER		SYN	2		2-9	0011 0010
51	33	LCER			3		3-9	0011 0011
52	34	HER		PN	4		4-9	0011 0100
53	35	LRER		RS	5		5-9	0011 0101
54	36	AXR		UC	6		6-9	0011 0110
55	37	SXR		EOT	7		7-9	0011 0111
56	38	LER			8		8-9	0011 1000
57	39	CER			9		1-8-9	0011 1001
58	3A	AER			:		2-8-9	0011 1010
59	3B	SER		CU3	;		3-8-9	0011 1011
60	3C	MER		DC4	<		4-8-9	0011 1100
61	3D	DER		NAK	=		5-8-9	0011 1101
62	3E	AUR			>		6-8-9	0011 1110
63	3F	SUR		SUB	?		7-8-9	0011 1111

1. Two columns of EBCDIC graphics are shown. The first gives IBM standard U.S. bit pattern assignments. The second shows the T-11 and TN text printing chains (120 graphics).
2. Add C (check bit) for odd or even parity as needed, except as noted.
3. For even parity use CA.

TWO-CHARACTER BSC DATA LINK CONTROLS

Function	EBCDIC	ASCII
ACK-0	DLE,X'70'	DLE,0
ACK-1	DLE,X'61'	DLE,1
WACK	DLE,X'6B'	DLE,;
RVI	DLE,X'7C'	DLE,<

CODE TRANSLATION TABLE (Contd)

Dec.	Hex	Instruction (RX)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code EBCDIC	Binary
64	40	STH		Sp Sp	@	(3)	no punches	0100 0000
65	41	LA			A		12-0-1-9	0100 0001
66	42	STC			B		12-0-2-9	0100 0010
67	43	IC			C		12-0-3-9	0100 0011
68	44	EX			D		12-0-4-9	0100 0100
69	45	BAL			E		12-0-5-9	0100 0101
70	46	BCT			F		12-0-6-9	0100 0110
71	47	BC			G		12-0-7-9	0100 0111
72	48	LH			H		12-0-8-9	0100 1000
73	49	CH			I		12-1-8	0100 1001
74	4A	AH		¢ ¢	J		12-2-8	0100 1010
75	4B	SH	.	. .	K	B A 8 2 1	12-3-8	0100 1011
76	4C	MH	¤)	< <	L	B A 8 4	12-4-8	0100 1100
77	4D		[	((	M	B A 8 4 1	12-5-8	0100 1101
78	4E	CVD	<	+ +	N	B A 8 4 2	12-6-8	0100 1110
79	4F	CVB	‡	| |	O	B A 8 4 2 1	12-7-8	0100 1111
80	50	ST	& +	& &	P	B A	12	0101 0000
81	51				Q		12-11-1-9	0101 0001
82	52				R		12-11-2-9	0101 0010
83	53				S		12-11-3-9	0101 0011
84	54	N			T		12-11-4-9	0101 0100
85	55	CL			U		12-11-5-9	0101 0101
86	56	O			V		12-11-6-9	0101 0110
87	57	X			W		12-11-7-9	0101 0111
88	58	L			X		12-11-8-9	0101 1000
89	59	C			Y		11-1-8	0101 1001
90	5A	A		! !	Z		11-2-8	0101 1010
91	5B	S	$	$ $	[	B 8 2 1	11-3-8	0101 1011
92	5C	M	•	• •	\	B 8 4	11-4-8	0101 1100
93	5D	D	]	))	]	B 8 4 1	11-5-8	0101 1101
94	5E	AL	;	; ;	¬ ^	B 8 4 2	11-6-8	0101 1110
95	5F	SL	Δ	¬ ¬	–	B 8 4 2 1	11-7-8	0101 1111
96	60	STD	-	- -	`	B	11	0110 0000
97	61		/	/ /	a	A 1	0-1	0110 0001
98	62				b		11-0-2-9	0110 0010
99	63				c		11-0-3-9	0110 0011
100	64				d		11-0-4-9	0110 0100
101	65				e		11-0-5-9	0110 0101
102	66				f		11-0-6-9	0110 0110
103	67	MXD			g		11-0-7-9	0110 0111
104	68	LD			h		11-0-8-9	0110 1000
105	69	CD			i		0-1-8	0110 1001
106	6A	AD		|	j	12-11	0110 1010
107	6B	SD	,	, ,	k	A 8 2 1	0-3-8	0110 1011
108	6C	MD	% (	% %	l	A 8 4	0-4-8	0110 1100
109	6D	DD	¥	_ _	m	A 8 4 1	0-5-8	0110 1101
110	6E	AW	\	> >	n	A 8 4 2	0-6-8	0110 1110
111	6F	SW	**	? ?	o	A 8 4 2 1	0-7-8	0110 1111
112	70	STE			p		12-11-0	0111 0000
113	71				q		12-11-0-1-9	0111 0001
114	72				r		12-11-0-2-9	0111 0010
115	73				s		12-11-0-3-9	0111 0011
116	74				t		12-11-0-4-9	0111 0100
117	75				u		12-11-0-5-9	0111 0101
118	76				v		12-11-0-6-9	0111 0110
119	77				w		12-11-0-7-9	0111 0111
120	78	LE			x		12-11-0-8-9	0111 1000
121	79	CE		`	y		1-8	0111 1001
122	7A	AE	¢	: :	z	A	2-8	0111 1010
123	7B	SE	# =	# #	{	8 2 1	3-8	0111 1011
124	7C	ME	@ '	@ @	¦	8 4	4-8	0111 1100
125	7D	DE	:	' '	}	8 4 1	5-8	0111 1101
126	7E	AU	>	= =	~	8 4 2	6-8	0111 1110
127	7F	SU	✓	" "	DEL	8 4 2 1	7-8	0111 1111

CODE TRANSLATION TABLE (Contd)

(11)

Dec.	Hex	Instruction and Format	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code EBCDIC	Binary
128	80	SSM -SI					12-0-1-8	1000 0000
129	81			a	a		12-0-1	1000 0001
130	82	LPSW -SI		b	b		12-0-2	1000 0010
131	83	Diagnose		c	c		12-0-3	1000 0011
132	84	WRD SI		d	d		12-0-4	1000 0100
133	85	RDD SI		e	e		12-0-5	1000 0101
134	86	BXH SI		f	f		12-0-6	1000 0110
135	87	BXLE SI		g	g		12-0-7	1000 0111
136	88	SRL		h	h		12-0-8	1000 1000
137	89	SLL		i	i		12-0-9	1000 1001
138	8A	SRA					12-0-2-8	1000 1010
139	8B	SLA RS					12-0-3-8	1000 1011
140	8C	SRDL					12-0-4-8	1000 1100
141	8D	SLDL					12-0-5-8	1000 1101
142	8E	SRDA					12-0-6-8	1000 1110
143	8F	SLDA					12-0-7-8	1000 1111
144	90	STM					12-11-1-8	1001 0000
145	91	TM SI		j	j		12-11-1	1001 0001
146	92	MVI SI		k	k		12-11-2	1001 0010
147	93	TS -S		l	l		12-11-3	1001 0011
148	94	NI		m	m		12-11-4	1001 0100
149	95	CLI SI		n	n		12-11-5	1001 0101
150	96	OI SI		o	o		12-11-6	1001 0110
151	97	XI SI		p	p		12-11-7	1001 0111
152	98	LM -RS		q	q		12-11-8	1001 1000
153	99			r	r		12-11-9	1001 1001
154	9A						12-11-2-8	1001 1010
155	9B						12-11-3-8	1001 1011
156	9C	SIO, SIOF S					12-11-4-8	1001 1100
157	9D	TIO, CLRIO S					12-11-5-8	1001 1101
158	9E	HIO, HDV S					12-11-6-8	1001 1110
159	9F	TCH S					12-11-7-8	1001 1111
160	A0						11-0-1-8	1010 0000
161	A1			~			11-0-1	1010 0001
162	A2			s	s		11-0-2	1010 0010
163	A3			t	t		11-0-3	1010 0011
164	A4			u	u		11-0-4	1010 0100
165	A5			v	v		11-0-5	1010 0101
166	A6			w	w		11-0-6	1010 0110
167	A7			x	x		11-0-7	1010 0111
168	A8			y	y		11-0-8	1010 1000
169	A9			z	z		11-0-9	1010 1001
170	AA						11-0-2-8	1010 1010
171	AB						11-0-3-8	1010 1011
172	AC	STNSM SI					11-0-4-8	1010 1100
173	AD	STOSM SI					11-0-5-8	1010 1101
174	AE	SIGP -RS					11-0-6-8	1010 1110
175	AF	MC -SI					11-0-7-8	1010 1111
176	B0						12-11-0-1-8	1011 0000
177	B1	LRA -RX					12-11-0-1	1011 0001
178	B2	See below					12-11-0-2	1011 0010
179	B3						12-11-0-3	1011 0011
180	B4						12-11-0-4	1011 0100
181	B5						12-11-0-5	1011 0101
182	B6	STCTL RS					12-11-0-6	1011 0110
183	B7	LCTL RS					12-11-0-7	1011 0111
184	B8						12-11-0-8	1011 1000
185	B9						12-11-0-9	1011 1001
186	BA	CS RS					12-11-0-2-8	1011 1010
187	BB	CDS RS					12-11-0-3-8	1011 1011
188	BC						12-11-0-4-8	1011 1100
189	BD	CLM					12-11-0-5-8	1011 1101
190	BE	STCM RS					12-11-0-6-8	1011 1110
191	BF	ICM					12-11-0-7-8	1011 1111

Op code (S format)

B202 - STIDP	B207 - STCKC	B20D - PTLB
B203 - STIDC	B208 - SPT	B210 - SPX
B204 - SCK	B209 - STPT	B211 - STPX
B205 - STCK	B20A - SPKA	B212 - STAP
B206 - SCKC	B20B - IPK	B213 - RRB

CODE TRANSLATION TABLE (Contd)

(12)

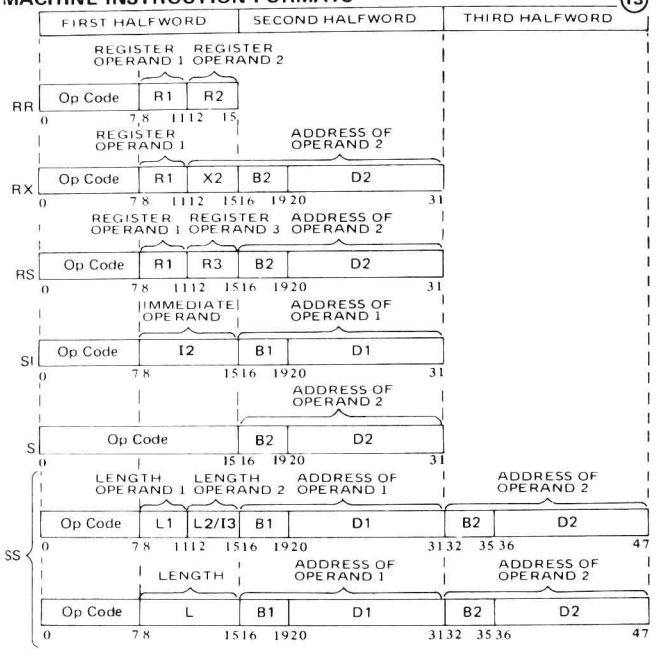
Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code EBCDIC	Binary
192	C0		?	{		B A 8 2	12-0	1100 0000
193	C1		A	A	A	B A 1	12-1	1100 0001
194	C2		B	B	B	B A 2	12-2	1100 0010
195	C3		C	C	C	B A 2 1	12-3	1100 0011
196	C4		D	D	D	B A 4	12-4	1100 0100
197	C5		E	E	E	B A 4 1	12-5	1100 0101
198	C6		F	F	F	B A 4 2	12-6	1100 0110
199	C7		G	G	G	B A 4 2 1	12-7	1100 0111
200	C8		H	H	H	B A 8	12-8	1100 1000
201	C9		I	I	I	B A 8 1	12-9	1100 1001
202	CA						12-0-2-8-9	1100 1010
203	CB						12-0-3-8-9	1100 1011
204	CC			∫			12-0-4-8-9	1100 1100
205	CD						12-0-5-8-9	1100 1101
206	CE			∀			12-0-6-8-9	1100 1110
207	CF						12-0-7-8-9	1100 1111
208	D0		!	}		B 8 2	11-0	1101 0000
209	D1	MVN	J	J	J	B 1	11-1	1101 0001
210	D2	MVC	K	K	K	B 2	11-2	1101 0010
211	D3	MVZ	L	L	L	B 2 1	11-3	1101 0011
212	D4	NC	M	M	M	B 4	11-4	1101 0100
213	D5	CLC	N	N	N	B 4 1	11-5	1101 0101
214	D6	OC	O	O	O	B 4 2	11-6	1101 0110
215	D7	XC	P	P	P	B 4 2 1	11-7	1101 0111
216	D8		Q	Q	Q	B 8	11-8	1101 1000
217	D9		R	R	R	B 8 1	11-9	1101 1001
218	DA						12-11-2-8-9	1101 1010
219	DB						12-11-3-8-9	1101 1011
220	DC	TR					12-11-4-8-9	1101 1100
221	DD	TRT					12-11-5-8-9	1101 1101
222	DE	ED					12-11-6-8-9	1101 1110
223	DF	EDMK					12-11-7-8-9	1101 1111
224	E0		‡	\		A 8 2	0-2-8	1110 0000
225	E1						11-0-1-9	1110 0001
226	E2		S	S	S	A 2	0-2	1110 0010
227	E3		T	T	T	A 2 1	0-3	1110 0011
228	E4		U	U	U	A 4	0-4	1110 0100
229	E5		V	V	V	A 4 1	0-5	1110 0101
230	E6		W	W	W	A 4 2	0-6	1110 0110
231	E7		X	X	X	A 4 2 1	0-7	1110 0111
232	E8		Y	Y	Y	A 8	0-8	1110 1000
233	E9		Z	Z	Z	A 8 1	0-9	1110 1001
234	EA						11-0-2-8-9	1110 1010
235	EB						11-0-3-8-9	1110 1011
236	EC			⊣			11-0-4-8-9	1110 1100
237	ED						11-0-5-8-9	1110 1101
238	EE						11-0-6-8-9	1110 1110
239	EF						11-0-7-8-9	1110 1111
240	F0	SRP	0	0	0	8 2	0	1111 0000
241	F1	MVO	1	1	1	1	1	1111 0001
242	F2	PACK	2	2	2	2	2	1111 0010
243	F3	UNPK	3	3	3	2 1	3	1111 0011
244	F4		4	4	4	4	4	1111 0100
245	F5		5	5	5	4 1	5	1111 0101
246	F6		6	6	6	4 2	6	1111 0110
247	F7		7	7	7	4 2 1	7	1111 0111
248	F8	ZAP	8	8	8	8	8	1111 1000
249	F9	CP	9	9	9	8 1	9	1111 1001
250	FA	AP					12-11-0-2-8-9	1111 1010
251	FB	SP					12-11-0-3-8-9	1111 1011
252	FC	MP					12-11-0-4-8-9	1111 1100
253	FD	DP					12-11-0-5-8-9	1111 1101
254	FE						12-11-0-6-8-9	1111 1110
255	FF			EO			12-11-0-7-8-9	1111 1111

ANSI-DEFINED PRINTER CONTROL CHARACTERS

(A in RECFM field of DCB)

Code	Action before printing record
blank	Space 1 line
0	Space 2 lines
-	Space 3 lines
+	Suppress space
1	Skip to line 1 on new page

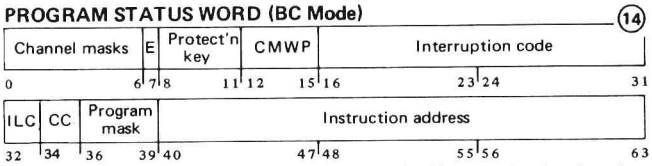
MACHINE INSTRUCTION FORMATS (13)



CONTROL REGISTERS

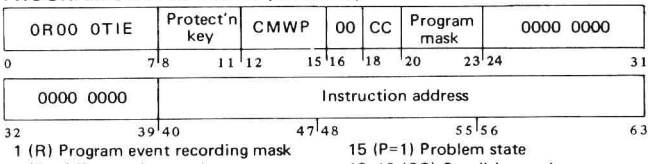
CR	Bits	Name of field	Associated with	Init.
0	0	Block-multiplex'g control	Block-multiplex'g	0
	1	SSM suppression control	SSM instruction	0
	2	TOD clock sync control	Multiprocessing	0
	8-9	Page size control	Dynamic addr. transl.	0
	10	Unassigned (must be zero)		0
	11-12	Segment size control		0
	16	Malfunction alert mask	Multiprocessing	0
	17	Emergency signal mask		0
	18	External call mask		0
	19	TOD clock sync check mask		0
	20	Clock comparator mask	Clock comparator	0
	21	CPU timer mask	CPU timer	0
	24	Interval timer mask	Interval timer	1
	25	Interrupt key mask	Interrupt key	1
	26	External signal mask	External signal	1
1	0-7	Segment table length	Dynamic addr. transl.	0
	8-25	Segment table address		0
2	0-31	Channel masks	Channels	1
8	16-31	Monitor masks	Monitoring	0
9	0	Successful branching event mask	Program-event record'g	0
	1	Instruction fetching event mask		0
	2	Storage alteration event mask		0
	3	GR alteration event mask		0
	16-31	PER general register masks		0
10	8-31	PER starting address	Program-event record'g	0
11	8-31	PER ending address	Program-event record'g	0
14	0	Check-stop control	Machine-check handling	1
	1	Synch. MCEL control		1
	2	I/O extended logout control	I/O extended logout	0
	4	Recovery report mask	Machine-check handling	0
	5	Degradation report mask		0
	6	Ext. damage report mask		1
	7	Warning mask		0
	8	Asynch. MCEL control		0
	9	Asynch. fixed log control		0
15	8-28	MCEL address	Machine-check handling	512

PROGRAM STATUS WORD (BC Mode) (14)



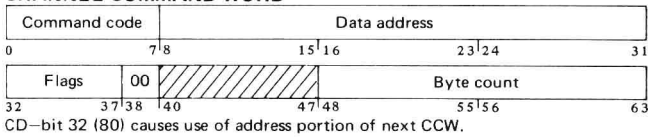
0–5 Channel 0 to 5 masks
6 Mask for channel 6 and up
7 (E) External mask
12 (C=0) Basic control mode
13 (M) Machine-check mask
14 (W=1) Wait state
15 (P=1) Problem state
32–33 (ILC) Instruction length code
34–35 (CC) Condition code
36 Fixed-point overflow mask
37 Decimal overflow mask
38 Exponent underflow mask
39 Significance mask

PROGRAM STATUS WORD (EC Mode)



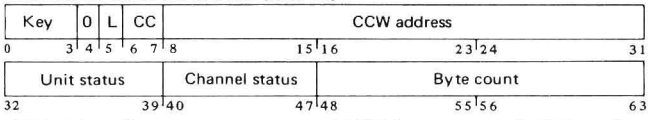
1 (R) Program event recording mask
5 (T=1) Translation mode
6 (I) Input/output mask
7 (E) External mask
12 (C=1) Extended control mode
13 (M) Machine-check mask
14 (W=1) Wait state
15 (P=1) Problem state
18–19 (CC) Condition code
20 Fixed-point overflow mask
21 Decimal overflow mask
22 Exponent underflow mask
23 Significance mask

CHANNEL COMMAND WORD



CD—bit 32 (80) causes use of address portion of next CCW.
CC—bit 33 (40) causes use of command code and data address of next CCW.
SLI—bit 34 (20) causes suppression of possible incorrect length indication.
Skip—bit 35 (10) suppresses transfer of information to main storage.
PCI—bit 36 (08) causes a channel program controlled interruption.
IDA—bit 37 (04) causes bits 8–31 of CCW to specify location of first IDAW.

CHANNEL STATUS WORD (hex 40)



5 Logout pending
6–7 Deferred condition code
32 (80) Attention
33 (40) Status modifier
34 (20) Control unit end
35 (10) Busy
36 (08) Channel end
37 (04) Device end
38 (02) Unit check
39 (01) Unit exception
40 (80) Program-controlled interruption
41 (40) Incorrect length
42 (20) Program check
43 (10) Protection check
44 (08) Channel data check
45 (04) Channel control check
46 (02) Interface control check
47 (01) Chaining check
48–63 Residual byte count for the last CCW used

PROGRAM INTERRUPTION CODES

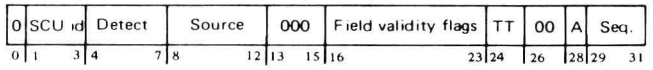
0001	Operation exception	000C	Exponent overflow excp
0002	Privileged operation excp	000D	Exponent underflow excp
0003	Execute exception	000E	Significance exception
0004	Protection exception	000F	Floating-point divide excp
0005	Addressing exception	0010	Segment translation excp
0006	Specification exception	0011	Page translation exception
0007	Data exception	0012	Translation specification excp
0008	Fixed-point overflow excp	0013	Special operation exception
0009	Fixed-point divide excp	0040	Monitor event
000A	Decimal overflow exception	0080	Program event (code may be combined with another code)
000B	Decimal divide exception		

FIXED STORAGE LOCATIONS (15)

Area, dec.	Hex addr	EC only	Function
0- 7	0		Initial program loading PSW, restart new PSW
8- 15	8		Initial program loading CCW1, restart old PSW
16- 23	10		Initial program loading CCW2
24- 31	18		External old PSW
32- 39	20		Supervisor Call old PSW
40- 47	28		Program old PSW
48- 55	30		Machine-check old PSW
56- 63	38		Input/output old PSW
64- 71	40		Channel status word (see diagram)
72- 75	48		Channel address word [0-3 key, 4-7 zeros, 8-31 CCW address]
80- 83	50		Interval timer
88- 95	58		External new PSW
96-103	60		Supervisor Call new PSW
104-111	68		Program new PSW
112-119	70		Machine-check new PSW
120-127	78		Input/output new PSW
132-133	84		CPU address assoc'd with external interruption, or unchanged
132-133	84	X	CPU address assoc'd with external interruption, or zeros
134-135	86	X	External interruption code
136-139	88	X	SVC interruption [0-12 zeros, 13-14 ILC, 15:0, 16-31 code]
140-143	8C	X	Program interrupt. [0-12 zeros, 13-14 ILC, 15:0, 16-31 code]
144-147	90	X	Translation exception address [0-7 zeros, 8-31 address]
148-149	94		Monitor class [0-7 zeros, 8-15 class number]
150-151	96	X	PER interruption code [0-3 code, 4-15 zeros]
152-155	98	X	PER address [0-7 zeros, 8-31 address]
156-159	9C		Monitor code [0-7 zeros, 8-31 monitor code]
168-171	A8		Channel ID [0-3 type, 4-15 model, 16-31 max. IOEL length]
172-175	AC		I/O extended logout address [0-7 unused, 8-31 address]
176-179	B0		Limited channel logout (see diagram)
185-187	B9	X	I/O address [0-7 zeros, 8-23 address]
216-223	D8		CPU timer save area
224-231	E0		Clock comparator save area
232-239	E8		Machine-check interruption code (see diagram)
248-251	F8		Failing processor storage address [0-7 zeros, 8-31 address]
252-255	FC		Region code*
256-351	100		Fixed logout area*
352-383	160		Floating-point register save area
384-447	180		General register save area
448-511	1C0		Control register save area
512†	200		CPU extended logout area (size varies)

*May vary among models; see system library manuals for specific model.
†Location may be changed by programming (bits 8-28 of CR 15 specify address).

LIMITED CHANNEL LOGOUT (hex B0)



4 CPU
5 Channel
6 Main storage control
7 Main storage
8 CPU
9 Channel
10 Main storage control
11 Main storage
12 Control unit
16 Interface address
17-18 Reserved (00)
19 Sequence code
20 Unit status
21 Cmd. addr. and key
22 Channel address
23 Device address
24-25 Type of termination
00 Interface disconnect
01 Stop, stack or normal
10 Selective reset
11 System reset
28(A) I/O error alert
29-31 Sequence code

MACHINE-CHECK INTERRUPTION CODE (hex E8)

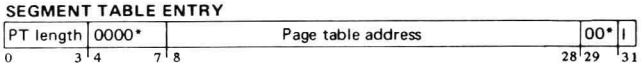
0 System damage
1 Instr. proc'g damage
2 System recovery
3 Timer damage
4 Timing facil. damage
5 External damage
6 Not assigned (0)
7 Degradation
8 Warning
14 Backed-up
15 Delayed
16 Uncorrected
17 Corrected
18 Key uncorrected
20 PSW bits 12-15
21 PSW masks and key
22 Prog. mask and CC
23 Instruction address
24 Failing stg. address
25 Region code
27 Floating-pt registers
28 General registers
29 Control registers
30 CPU ext'd logout
31 Storage logical
46 CPU timer
47 Clock comparator

DYNAMIC ADDRESS TRANSLATION (16)

VIRTUAL (LOGICAL) ADDRESS FORMAT

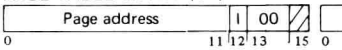
Segment Size	Page Size		Segment Index	Page Index	Byte Index
64K	4K	Bits	8 - 15	16 - 19	20 - 31
64K	2K	0 - 7	8 - 15	16 - 20	21 - 31
1M	4K	are	8 - 11	12 - 19	20 - 31
1M	2K	ignored	8 - 11	12 - 20	21 - 31

SEGMENT TABLE ENTRY



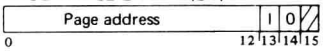
*Normally zeros; ignored on some models.
31 (I) Segment-invalid bit.

PAGE TABLE ENTRY (4K)



12 (I) Page-invalid bit.

PAGE TABLE ENTRY (2K)



13 (I) Page-invalid bit.

HEXADECIMAL AND DECIMAL CONVERSION

From hex: locate each hex digit in its corresponding column position and note the decimal equivalents. Add these to obtain the decimal value.

From decimal: (1) locate the largest decimal value in the table that will fit into the decimal number to be converted, and (2) note its hex equivalent and hex column position. (3) Find the decimal remainder. Repeat the process on this and subsequent remainders.

Note: Decimal, hexadecimal, (and binary) equivalents of all numbers from 0 to 255 are listed on panels 9 - 12.

HEXADECIMAL COLUMNS											
6		5		4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
0 1 2 3		4 5 6 7		0 1 2 3		4 5 6 7		0 1 2 3		4 5 6 7	
BYTE				BYTE				BYTE			

POWERS OF 2

2^n	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$

POWERS OF 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

Contents