

IBM Technical Newsletter

File Number 1410/7010-22
Re: Form No. C28-0326
This Newsletter No. N28-1069
Date July 3, 1963
Previous Newsletter Nos. None

IBM 1410/7010 Operating System
Autocoder: Preliminary Specifications

Attached are replacement pages for the IBM Systems Reference Library publication, "IBM 1410/7010 Operating System; Autocoder: Preliminary Specifications," Form C28-0326. The replacement pages are 29 and 30, and the Appendix.

Amendments on pages 29 and 30 are indicated by a • placed to the left of the heading or paragraph affected.

Please include the title, "BASE2--Base Address (Common Data Area)," after "BASE1" in the following places as appropriate:

1. Under "Linkage Loader Control Operations" on page 28.
2. In the table of Contents (reference: page 30).
3. In the Index (reference: page 30).

The following operands are permissible in a BASEL statement:

- Actual
- Symbol
- Asterisk plus X00 (*+X00)

Actual: The actual core storage location is written in the operand field (see Figure 64).

Line	Label	Operation
3	5/6	15/16 20/21 25 30 35 40
0.1		BASEL 15000
0.2		

Figure 64. Use of an Actual Address in a BASEL Statement

Symbol: A symbolic address can be a symbolic name or Linkage Symbol defined in a subprogram that is processed by the Linkage Loader prior to the subprogram in which the BASEL statement appears. This symbolic address is defined by means of the DEFIN or TITLE statements (see below).

In Figure 65, the operand PROGRAM2 is the name of a subprogram that has been previously processed.

Line	Label	Operation
3	5/6	15/16 20/21 25 30 35 40
0.1		BASEL PROGRAM2
0.2		

Figure 65. Symbolic Address in a BASEL Statement

Asterisk Plus X00 (+X00):* If an *+X00 is the operand of a BASEL statement, the relocation factor will be incremented to the next even-hundred location. For example, if the asterisk has the value of 18279, the relocation factor will be 18300. If the value of the asterisk is already an even hundred address, that value will remain the relocation factor.

PRCT — Protect

The PRCT statement is used to set a limit for erasure of Linkage Symbols from the symbol table. The Linkage Loader will retain in its symbol table all Linkage Symbols equal to or higher than the address value specified by the operand of the PRCT statement.

The operand of a PRCT statement can be either a Linkage Symbol or an actual value (see Figure 66).

Line	Label	Operation
3	5/6	15/16 20/21 25 30 35 40
0.1		PRCT LABEL
0.2		

Figure 66. The PRCT Statement

•TITLE — Title

The TITLE statement is used to establish an identifying name for a subprogram, to indicate the size of the common data area the subprogram will use, and to state the lowest origin point in the subprogram.

ENTRIES

The name of the subprogram must be a conventional label (1-10 alphameric characters in length) written as the first entry in the operand field. This name can be used in DCWS, DCWF, and CALL statements.

The size of the common data area, if required, is written as the second entry in the form of an integer, one to five positions in length.

The lowest origin point is the third entry and is optional. If it is omitted, the processor will automatically compute the lowest origin point in the program and pass it to the Linkage Loader along with the subprogram name and the size of the common data area required.

If the third entry is included, the automatic computation of the processor is negated, and the value declared by the entry is passed on to the Linkage Loader. This entry can be an actual value or a label within the assembly.

Although the third entry is normally omitted, it can be useful under the following conditions:

1. A program being assembled contains one or more SPEND statements, and
2. The low origins of the subprograms are different and it is desired to indicate these different low origin points. In the absence of a third parameter, the processor will automatically place the same lowest origin encountered in all the TITLE cards within an assembly.

FORMAT CONSIDERATIONS

The entries are written in the operand field separated by commas. If the second entry is omitted and a third entry is used, the third entry must be separated from the first entry by two commas. If both the second and third entries are omitted, only the name is required with no trailing commas. See Figure 67.

The second and third entries are inverted in sequence when passed to the Linkage Loader.

Line	Label	Operation
3	5/6	15/16 20/21 25 30 35 40
0.1		TITLE NAME
0.2		TITLE NAME, 145
0.3		TITLE NAME, 145, LABEL
0.4		TITLE NAME, LABEL
0.5		TITLE NAME, 145, 18000
0.6		

•Figure 67. Permissible Forms of the TITLE Statement

DEFIN — Definition

The DEFIN control operation is used to define a Linkage Symbol for use by the Linkage Loader. This Linkage Symbol represents an alternative entry point or data

field within the subprogram being assembled by the Autocoder processor. This symbol can be referenced by other subprograms. This symbol can appear in the form of a one- to ten-position conventional label (see "DCWF - Subprogram Address Constant") or in the form, LABEL/.

See the publication, *The System Monitor* for details concerning the treatment of these symbols by the Linkage Loader. Figure 68 illustrates the format of the DEFIN statement.

The operand of the DEFIN statement can be indexed and/or address adjusted.

Line	Label	Operation
3 5 6	15	16 20 21 25 30 35 40
0 1	LABEL	DEFIN ABCDEF
0 2		

Figure 68. The DEFIN Statement

CALL - Subprogram Call

The CALL statement provides the Linkage Loader with the name of a subprogram that is to be loaded from the System Library File or from the Go File. It serves the same call function provided by the DCWF and DCWS

statements except that it requires no core storage in the object program in which the CALL appears. When the CALL Card produced by the processor is loaded with the object deck, the operand of the CALL statement is passed directly to the Linkage Loader. The operand of the CALL Card is the name of the subprogram to be processed by the Linkage Loader. The format of the CALL Card is illustrated in Figure 69.

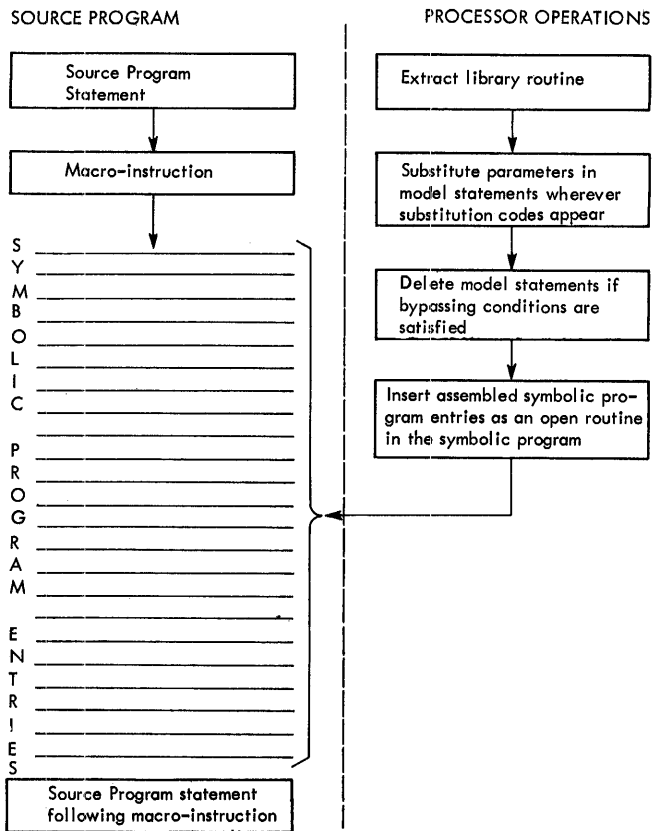
Line	Label	Operation
3 5 6	15	16 20 21 25 30 35 40
0 1		CALL SUBPROGRAM
0 2		

Figure 69. The CALL Statement

BASE2 - Base Address (Common Data Area)

The BASE2 card is used to relocate the COMMON data area to a location other than the value of /AMS/. The operand of the BASE2 card, which can be either a Linkage Symbol or an actual address, sets the upper limit of this common data area.

See the publication, *The System Monitor*, for details concerning the treatment of the BASE2 card by the Linkage Loader.



When a macro-instruction is encountered in the source program, the processor extracts the specified library routine, tailors it, and inserts it in-line in the users source program.

Figure 100. Macro Processing

Appendix: Autocoder Operation Codes

- This Appendix contains a complete listing of Autocoder language statements and, where applicable, their machine-language equivalents. In this listing, A-addresses in operands have been equated to location 12345, B-addresses have been equated to location 34567, and I-addresses have been equated to location 56789. C-addresses (for the Store Register instructions) have been equated to 45678. The character "D" in an operand indicates that an appropriate machine-language d-modifier is to be coded in the Autocoder source statement.

Included in the listing are several generalized forms

for source statements. For example, the programmer can cause the generation of any machine-language instruction with an operation code of "J" by using the generalized form:

JID I,D

Place JID in the Op Code field and the branch address (I-address) and appropriate d-modifier in the Operand field. The generalized forms permit the coding of instructions for which there are no specific Autocoder mnemonics. Addresses in the operand of the generalized forms can be specified symbolically with labels and can have address adjustment and indexing.

AUTOCODER MNEMONIC OPERATION CODES			
LABEL	OPCODE	OPERAND	INSTRUCTION
*	PROCESSOR CONTROL OPERATIONS		
	ORG	10000 ORIGIN	
	LTORG	* LITERAL ORIGIN	
*	HEADR	THIS INFORMATION WILL BE THE HEADING OF LISTING	
*	RESEQ	RESEQUENCE AT 001, NEW PAGE, NEW IDENT	
*	EJECT	CONTINUE LISTING ON NEW PAGE	
*	SPEND	START	END PRIMARY SUBPROGRAM
*	SPEND		END SECONDARY SUBPROGRAM
*	END	START	END PRIMARY SUBPROGRAM AND AUTOCODER RUN
*	END		END SECONDARY SUBPRGRM AND AUTOCODER RUN
	NOPWM		NO OPERATION WORD MARK N
*	LINKAGE LOADER CONTROL OPERATIONS		
*	BASE1	12000	CONTROLS RELOCATION FACTOR
*	BASE2	38000	UPPER LIMIT FOR COMMON DATA AREA
*	TITLE	COSINE,20,0	DECLARES NAME OF SUBPROGRAM,
*			SIZE OF COMMON, ORIGIN POINT
SINE/	DEFIN	COSINE+42	DECLARES LINKAGE SYMBOL
*	PRTCT	ABCD/	PREVENTS ERASURE OF LINKAGE SYMBOLS
*			FROM LINKAGE LOADER TABLE
*	CALL	THIRD	SUBPROGRAM CALL
*	DECLARATIVE OPERATIONS		
A	EQU	12345	THE EQUATE INSTRUCTION
B	EQU	34567	
C	EQU	45678	
I	EQU	56789	

AUTOCODER MNEMONIC OPERATION CODES

LABEL	OPCOD	OPERAND	INSTRUCTION
DA	1X2,G	DEFINE AREA	3 10001
DAV	1X2,G	DEFINE AREA IN COMMON AREA	3 10004
DCW	--	DEFINE CONSTANT WITH WORD MARK	1 10007
DC	+2	DEFINE CONSTANT	1 10008
DS	1	DEFINE SYMBOL	1 10009
DCWF	NAME	ADCON FOR ENTRY POINT OF NAMED SUBPROGRAM	5 10010
DCWS	NAME	BRANCH TO NAMED SUBPROGRAM	7 10015
RSV	LABEL	APPLY DOWNWARD RELOCATION TO LABEL	
*	ARITHMETIC OPERATIONS		
A	A,B	ADD A-FIELD TO B-FIELD	A 12345 34567
S	A,B	SUBTRACT A FROM B	S 12345 34567
ZA	A,B	ZERO AND ADD A TO B	+ 12345 34567
ZS	A,B	ZERO AND SUBTRACT A FROM B	- 12345 34567
M	A,B	MULTIPLY	- 12345 34567
D	A,B	DIVIDE	(12345 34567
*	MOVE RIGHT TO LEFT COMMANDS		
*	MOVE SINGLE POSITION		
MLNS	A,B	MOVE LEFT NUMERIC SINGLE	D 12345 34567 1
MLZS	A,B	ZONES SINGLE	D 12345 34567 2
MLCS	A,B	CHARACTERS SINGLE	D 12345 34567 3
MLWS	A,B	WORD MARKS SINGLE	D 12345 34567 4
MLNWS	A,B	NUMERIC AND WORD MARK SINGLE	D 12345 34567 5
MLZWS	A,B	ZONE AND WORD MARK SINGLE	D 12345 34567 6
MLCWS	A,B	CHARACTER AND WORD MARK SINGLE	D 12345 34567 7
*	STOP MOVE AT WORD MARK IN A-FIELD		
MLNA	A,B	MOVE LEFT NUMERIC TO A-FIELD WORD MARK	D 12345 34567 /
MLZA	A,B	ZONES TO A-FIELD WORD MARK	D 12345 34567 S
MLCA	A,B	CHARACTERS TO A-FIELD WORD MARK	D 12345 34567 T
MLWA	A,B	WORD MARKS TO A-FIELD WORD MARK	D 12345 34567 U
MLNWA	A,B	NUMERIC AND WM TO WORD MARK IN A	D 12345 34567 V
MLZWA	A,B	ZONES AND WM TO WORD MARK IN A	D 12345 34567 W
MLCWA	A,B	CHARACTERS AND WM TO WORD MK IN A	D 12345 34567 X
*	STOP MOVE AT WORD MARK IN B-FIELD		
MLNB	A,B	MOVE LEFT NUMERIC TO B-FIELD WORD MARK	D 12345 34567 J
MLZB	A,B	ZONES TO B-FIELD WORD MARK	D 12345 34567 K
MLCB	A,B	CHARACTERS TO B-FIELD WORD MARK	D 12345 34567 L
MLWB	A,B	WORD MARKS TO B-FIELD WORD MARK	D 12345 34567 M
MLNWB	A,B	NUMERIC AND WM TO WORD MARK IN B	D 12345 34567 N
MLZWB	A,B	ZONES AND WM TO WORD MARK IN B	D 12345 34567 O
MLCWB	A,B	CHARACTERS AND WM TO WORD MK IN B	D 12345 34567 P
*	STOP MOVE AT WORD MARK IN A- OR B-FIELD		
MLN	A,B	MOVE LEFT NUMERIC	D 12345 34567 A
MLZ	A,B	ZONES	D 12345 34567 B

AUTCCODER MNEMONIC OPERATION CODES

LABEL	OPCOD	OPERAND	INSTRUCTION
	MLC	A,B	CHARACTERS D 12345 34567 C
	MLW	A,B	WORD MARKS D 12345 34567 D
	MLNW	A,B	NUMERIC AND WORD MARKS D 12345 34567 E
	MLZW	A,B	ZONES AND WORD MARKS D 12345 34567 F
	MLCW	A,B	CHARACTERS AND WORD MARKS D 12345 34567 G
*	MOVE LEFT TO RIGHT COMMANDS		
*	STOP MOVE AT WORD MARK IN A- OR B-FIELD		
	MRN	A,B	MOVE RIGHT NUMERIC D 12345 34567 9
	MRZ	A,B	ZONES D 12345 34567 0
	MRC	A,B	CHARACTERS D 12345 34567 =
	MRW	A,B	WORD MARKS D 12345 34567 -
	MRNW	A,B	NUMERIC AND WORD MARKS D 12345 34567 .
	MRZW	A,B	ZONES AND WORD MARKS D 12345 34567 T
	MRCW	A,B	CHARACTERS AND WORD MARKS D 12345 34567 M
*	STOP MOVE AT RECORD MARK IN A-FIELD		
	MRNR	A,B	MOVE RIGHT NUMERIC TO RECORD MARK IN A-FIELD D 12345 34567 Z
	MRZR	A,B	ZONES TO RECORD MARK IN A-FIELD D 12345 34567 #
	MRCR	A,B	CHARACTERS TO RECORD MARK IN A D 12345 34567 ,
	MRWR	A,B	WORD MARKS TO RECORD MARK IN A D 12345 34567 (
	MRNWR	A,B	NUMERIC AND WM TO RM IN A-FIELD D 12345 34567 S
	MRZWR	A,B	ZONES AND WM TO RM IN A-FIELD D 12345 34567 '
	MRCWR	A,B	CHARACTERS AND WM TO RM IN A D 12345 34567 Q
	MRNG	A,B	MOVE RIGHT NUMERIC TO GM-WM IN A-FIELD D 12345 34567 R
	MRZG	A,B	ZONES TO GM-WM IN A-FIELD D 12345 34567 0
	MRCG	A,B	CHARACTERS TO GM-WM IN A-FIELD D 12345 34567 \$
	MRWG	A,B	WORD MARKS TO GM-WM IN A-FIELD D 12345 34567 *
	MRNWG	A,B	NUMERIC AND WM TO GM-WM IN A D 12345 34567 P
	MRZWG	A,B	ZONES AND WM TO GM-WM IN A-FIELD D 12345 34567 ;
	MRCWG	A,B	CHARACTERS AND WM TO GM-WM IN A D 12345 34567 L
*	STOP AT RM OR GM-WM IN A-FIELD		
	MRNM	A,B	MOVE RIGHT NUMERIC TO RM OR GM-WM D 12345 34567 I
	MRZM	A,B	ZONES TO RM OR GM-WM D 12345 34567 0
	MRCM	A,B	CHARACTERS TO RM OR GM-WM D 12345 34567 .
	MRWM	A,B	WORD MARKS TO RM OR GM-WM D 12345 34567)
	MRNWM	A,B	NUMERIC AND WM TO RM OR GM-WM D 12345 34567 L
	MRZWM	A,B	ZONES AND WM TO RM OR GM-WM D 12345 34567 T
	MRCWM	A,B	CHARACTERS AND WM TO RM OR GM-WM D 12345 34567 M
*	SCAN LEFT AND RIGHT COMMANDS		
	SCNRR	A,B	SCAN RIGHT TO RM IN A-FIELD D 12345 34567 Y
	SCNRG	A,B	TO GM-WM IN A-FIELD D 12345 34567 Q
	SCNRM	A,B	TO RM OR GM-WM IN A-FIELD D 12345 34567 H
	SCNR	A,B	TO WORD MARK IN A- OR B-FIELD D 12345 34567 8

AUTOCODER MNEMONIC OPERATION CODES

LABEL	OPCOD	OPERAND	INSTRUCTION
	SCNLA	A,B SCAN LEFT TO WORD MARK IN A-FIELD	D 12345 34567 ^C T
	SCNLB	A,B TO WORD MARK IN B-FIELD	D 12345 34567 -
	SCNL	A,B TO WM IN A- OR B-FIELD	D 12345 34567 +
	SCNLS	A,B SINGLE POSITION	D 12345 34567
*	SPECIAL MOVE COMMANDS		
	MCS	A,B MOVE CHARACTERS AND SUPPRESS ZEROS	Z 12345 34567
	MCE	A,B MOVE CHARACTERS AND EDIT	E 12345 34567
*	COMPARE AND LOOKUP COMMANDS		
	C	A,B COMPARE B-FIELD TO A-FIELD	C 12345 34567
	LL	A,B LOOKUP LOW	T 12345 34567 1
	LE	A,B LOOKUP EQUAL	T 12345 34567 2
	LLE	A,B LOOKUP LOW OR EQUAL	T 12345 34567 3
	LH	A,B LOOKUP HIGH	T 12345 34567 4
	LLH	A,B LOOKUP LOW OR HIGH	T 12345 34567 5
	LEH	A,B LOOKUP EQUAL OR HIGH	T 12345 34567 6
*	LOGICAL OPERATIONS		
	BW	I,B BRANCH TO I-ADDR IF WORD MARK AT B-ADDRESS	V 56789 34567 1
	BZN	I,B BRANCH TO I IF B HAS NO AB-BITS	V 56789 34567 2
	BZN	I,B,AB IF B HAS A-BIT AND B-BIT	V 56789 34567 B
	BZN	I,B,+ A-BIT AND B-BIT	V 56789 34567 B
	BZN	I,B,A IF B HAS A-BIT AND NO B-BIT	V 56789 34567 S
	BZN	I,B,T A-BIT AND NO B-BIT	V 56789 34567 S
	BZN	I,B,B IF B HAS B-BIT AND NO A-BIT	V 56789 34567 K
	BZN	I,B,- B-BIT AND NO A-BIT	V 56789 34567 K
	BWZ	I,B BRANCH TO I IF B HAS WM AND NO AB-BITS	V 56789 34567 3
	BWZ	I,B,AB AND AB-BITS	V 56789 34567 C
	BWZ	I,B,+ AND AB-BITS	V 56789 34567 C
	BWZ	I,B,A AND A-BIT	V 56789 34567 T
	BWZ	I,B,T AND A-BIT	V 56789 34567 T
	BWZ	I,B,B AND B-BIT	V 56789 34567 L
	BWZ	I,B,- AND B-BIT	V 56789 34567 L
	BCE	I,B,D BRANCH TO I IF CHARACTER AT B EQU D-MOD	B 56789 34567 D
	BBE	I,B,D BRNCH IF ANY BIT AT B MATCHES BIT IN D-MOD	W 56789 34567 D
	B	I UNCONDITIONAL BRANCH	J 56789
	BU	I BRANCH IF COMPARE UNEQUAL	J 56789 /
	BE	I EQUAL	J 56789 S
	BL	I LOW	J 56789 T
	BH	I HIGH	J 56789 U
	BZ	I BRANCH IF ZERO BALANCE	J 56789 V
	BAV	I BRANCH IF ARITHMETIC OVERFLOW	J 56789 Z
	BDV	I BRANCH IF DIVIDE OVERFLOW	J 56789 W

AUTOCODER MNEMONIC OPERATION CODES

LABEL	OPCOD	OPERAND	INSTRUCTION	
MISCELLANEOUS OPERATIONS				
SAR	C	STORE A-REGISTER	G 45678 A	
SBR	C	STORE B-REGISTER	G 45678 B	
SW	A	SET WORD MARK AT A	, 12345	
SW	A,B	SET WORD MARK AT A AND B	, 12345 34567	
CW	A	CLEAR WORDMARK AT A) 12345	
CW	A,B	CLEAR WORD MARK AT A AND B) 12345 34567	
CS	B	CLEAR STORAGE	/ 34567	
CS	I,B	CLEAR STORAGE AND BRANCH	/ 56789 34567	
NOP		NO OPERATION	N	
STC	A	STORE TIME CLOCK	G 12345	T
SR	C,D	GENERALIZED STORE REGISTER	G 45678	D
STCPU	I	STORE CPU STATUS	\$ 56789	S
RSCPU	I	RESTORE CPU STATUS	\$ 56789	R
FLOATING POINT ARITHMETIC INSTRUCTIONS				
FRA	A	FLOATING RESET ADD	= 12345	R
FST	A	FLOATING STORE	= 12345	L
FA	A	FLOATING ADD	= 12345	A
FS	A	FLOATING SUBTRACT	= 12345	S
FM	A	FLOATING MULTIPLY	= 12345	M
FD	A	FLOATING DIVIDE	= 12345	D
BXO	I	BRANCH EXPONENT OVERFLOW	J 56789	Y
BXU	I	BRANCH EXPONENT UNDERFLOW	J 56789	X

NOTE: The remainder of this listing presents instructions that are (or can be) related to input/output functions, including the use of priority and overlap. They are "restricted" in that special care must be given to their use, since they are potential hazards to such Resident Monitor control functions as input/output scheduling and assignment of input/output units. Because all programs within the Operating System are provided with input/output facilities by the

system's IOCS, the majority of these instructions will not normally be used in coding an Autocoder program. The programmer who wishes to use any of these instructions is advised to be familiar with the extended use of the IOCS, as explained in the publication IBM 1410/7010 Operating System; Basic Input/Output Control System: Preliminary Specifications, Form C28-0322.

CONDITIONAL BRANCHES FOR I/O, OVERLAP, AND PRIORITY				
BEX1	I,D	BRANCH EXTERNAL INDICATOR - CHANNEL 1	R 56789	D
BEX2	I,D	- CHANNEL 2	X 56789	D
BEX3	I,D	- CHANNEL 3	3 56789	D
BEX4	I,D	- CHANNEL 4	1 56789	D

NOTE: The Branch External Indicator instructions must not be used in any form, for any purpose, by dependent programs within the Operating System. These instructions reset certain interrupt indicators

and can result in either an I/O interlock error or failure of IOCS functions. The BEX mnemonic is included in this listing merely as an aid for reading assembly listings of the Resident IOCS.

BOL1	I	BRANCH OVERLAP IN PROCESS - CHANNEL 1	J 56789	1
BOL2	I	- CHANNEL 2	J 56789	2
BOL3	I	- CHANNEL 3	J 56789	3
BOL4	I	- CHANNEL 4	J 56789	4
BB1	I	BRANCH IF BINARY CARD - CHANNEL 1	J 56789	M
BB2	I	- CHANNEL 2	J 56789	(

AUTOCODER MNEMONIC OPERATION CODES

LABEL	OPCOD	OPERAND	INSTRUCTION	
BPCB	I	BRANCH PRINTER CARRIAGE BUSY - CHANNEL 1	J 56789	R
BPCB1	I	- CHANNEL 1	J 56789	R
BPCB2	I	- CHANNEL 2	J 56789	L
BCV	I	BRANCH CARRIAGE OVERFLOW - CHANNEL 1	J 56789	-
BCV1	I	- CHANNEL 1	J 56789	-
BCV2	I	- CHANNEL 2	J 56789)
BC9	I	BRANCH CARRIAGE CHANNEL 9 - CHANNEL 1	J 56789	9
BC91	I	- CHANNEL 1	J 56789	9
BC92	I	- CHANNEL 2	J 56789	0
BXPA	I	BRANCH AND EXIT PRIORITY ALERT	Y 56789	X
BEPA	I	BRANCH AND ENTER PRIORITY ALERT	Y 56789	E
* JID	I,D	GENERALIZED TEST AND BRANCH		
* THE ABOVE IS A GENERALIZED FORM, PERTAINING TO				
* THE J OPCODE. THE PROPER D MODIFIER MUST BE				
* SUBSTITUTED BY THE USER FOR THE D SHOWN IN THE OPERAND				
* BPI	I,D	GENERALIZED PRIORITY TEST AND BRANCH		
* THE ABOVE IS A GENERALIZED FORM, PERTAINING TO				
* THE Y OPCODE. THE PROPER D MODIFIER MUST BE				
* SUBSTITUTED BY THE USER FOR THE D SHOWN IN THE OPERAND				
NOTE: The Priority Test and Branch instructions		thorough knowledge of the internal functions of the		
reset indicators tested by the Resident IOCS and,		IOCS is prerequisite for use of these instructions in		
therefore, should be used with special care. A		a dependent program.		
STATS	I,D	GENERALIZED STORE AND RESTORE STATUS	\$ 56789	D
* THE ABOVE IS A GENERALIZED FORM. THE PROPER				
* D MODIFIER MUST BE SUBSTITUTED FOR THE D SHOWN.				
* FOUR EXAMPLES OF THIS USAGE ARE				
STATS	I,E	STORE CHANNEL 1 STATUS	\$ 56789	E
STATS	I,G	STORE CHANNEL 3 STATUS	\$ 56789	G
STATS	I,1	RESTORE CHANNEL 1 STATUS	\$ 56789	1
STATS	I,4	RESTORE CHANNEL 4 STATUS	\$ 56789	4
SSF	0	SELECT STACKER 0 AND FEED - CHANNEL 1	K	0
CC	1	CARRIAGE CONTROL I/O CHANNEL 1	F	1
BSP	(B1	BACKSPACE TAPE	U (B1	B
WTM	(B1	WRITE TAPE MARK	U (B1	M
RWD	(B1	REWIND	U (B1	R
RWU	(B1	REWIND AND UNLOAD	U (B1	U
CU	(B1,W	CONTROL UNIT	U (B1	W
MU	(B1,B,D	TO BUILD MOVE MODE I/O COMMAND	M (B1 34567	D
LU	(B1,B,D	TO BUILD LOAD MODE I/O COMMAND	L (B1 34567	D
H	I	HALT AND BRANCH	. 56789	

NOTE: The Halt instruction, although not necessarily related to input/output functions, is included in the "restricted" category because its use is in opposition to the Operating System convention that dependent programs should not interrupt Monitor control with machine halts. This convention is especially significant for dependent programs that run under control of a Resident Monitor that includes the Tele-Processing[®] Supervisor.

EOJ END

Actual Address	11	Address Constant	14
Address Adjustment	12	LTORC - Literal Origin	26
ADDRS (Address)	8	Machine Requirements	5
Assembly Listing	8	Macro Operations	31
Asterisk Address	12	Macro Statements	6
Autocoder Operation Codes	17	Macro System	31
Autocoder Processor	5	MATH (Pseudo-Macro)	35
Autocoder Statements	6	MEND (Pseudo-Macro)	39
BASE1 - Base Address	28	"NO" Relocation	7
BASE2 - Base Address (COMMON Data Area)	30	NOPWM - No Operation; Word Mark	28
Blank Operand	15	NOTE	
BOOL (Pseudo-Macro)	37	Assembly Listing	8
CALL - Subprogram Call	30	Pseudo Macro	39
CARD (Card Number)	9	Op Code (Operation Code)	8
Comments and Comments Card	10	Operand	7
COMMON	22	Types of Operands	11
COMP (Pseudo-Macro)	38	Assembly Listing	8
Control Operation Codes	25	Operation Code	6
Control Statements	6	Assembly Listing	8
CT (Character Count)	8	Machine Language (see Appendix)	
DA - Define Area	19	Options, Processing	8
Header Line	19	ORG - Origin	25
Header Line (with Fields)	20	Actual	26
Sample Problem	21	Symbolic	26
Review	22	Blank	26
DAV - Define Area in COMMON	22	Asterisk	26
The COMMON Data Area	22	Asterisk + X00	26
Coding Examples	23	Page-and-Line Number	8
DC - Define Constant	19	PCLIN (Page-and-Line Number)	8
DCW - Define Constant with Word Mark	17	Processing Options	8
Address Constant	18	Processor, Autocoder	5
Alphameric	18	PRTCT - Protect	29
Blank	18	Pseudo-Macro Instructions	35
Numerical	17	Coding Example	40
DCWF - Subprogram Address Constant	19	Reference Address	11
DCWS - Subprogram Branch Instruction	19	Actual	11
Declarative Operation Codes	17	Symbolic	11
Declarative Statements	6	Asterisk	12
DEFIN - Definition	29	REL (Relocation Indicator)	8
"Downward" Relocation	7	Relocation	7
DS - Define Symbol	19	RESEQ - Resequence	27
EJECT - Eject	27	RSV - Reserve	27
END - End	28	SEQNO (Sequence Number)	8
EQU - Equate	23	Sequence Numbers	8
Actual or Symbolic Address	23	SPEND - End Subprogram	28
Adjusted or Modified Address	24	Statements, Autocoder	6
Index Register	24	Imperative	6
Asterisk	25	Declarative	6, 17
EXEQ Card	8	Control	6, 25
Flag	9	Macro	6, 31
HEADR - Header Line	27	Sum Boxes	35
Imperative Statements	6	Symbolic Address	11
Indexing	12	Termination, Object Program	7
Instruction (see Assembly Listing)	8	TITLE - Title	29
Label	6	Types of Operands	11
Assembly Listing	8	Reference Address	11
Linkage Loader		Address Adjustment	12
Control Operations	28	Indexing	12
Linkage Symbols	15	Literals	13
Linkage Symbols	15	Linkage Symbols	15
Literals	13	Miscellaneous	15
Numerical	13	"Upward" Relocation	7
Alphameric	13	Zero Operand	15
Area-Defining	14		

