

IBM

*705
Autocoder
System
Macro-Instruction
Manual*

Logical Macro-Instructions		Page	Input-Output Macro-Instructions		Page
ALTTR	Alternate Transfer	69	RDCD	Read Card	33
FTNOP	First Time Only NOP	73			
FTTR	First Time Only TR	74	DPDR	Dump on Drum	7
IFxxx	If, then CMP and TR	78	RDDR	Read Drum	35
TREH	Transfer Equal or High	104	WRDR	Write Drum	51
TREL	Transfer Equal or Low	105			
TRLOW	Transfer Low	106	DPPRA	Dump on Printer, Auto. Ctl.	9
TRMIN	Transfer Minus	107			
TRNE	Transfer Not Equal	109	DPPRT	Dump on Printer, Pgm. Ctl.	11
TRNZ	Transfer Non-Zero	108			
			PRINT	Print Under Program Ctl.	29
SWNOP	Set Switch to NOP	101	PRNTA	Print Under Auto. Ctl.	27
SWTR	Set Switch to TR	102	WREPA	Write Erase Printer, Auto. Ctl.	53
LOOP	Start Loop	83	WREPR	Write Erase Printer, Pgm. Ctl.	55
END	Test for End of Loop	72			
RPTA	Repeat Loop (ASU Ctr)	89			
RPTM	Repeat Loop (Mem. Ctr)	91	DPPCH	Dump on Punch	13
			PUNCH	Punch Card	31
			WREPN	Write Erase Punch	57
HLTOF	Halt Off	75			
HLTON	Halt On	76			
HLTTR	Halt and Transfer	77	ALTP	Alternate Tape Units	5
			BSTP	Backspace Tape	6
RCMP	Restore CMP Trigger	93	DPTP	Dump on Tape	15
RSGN	Restore Sign Trigger	94	FSTP	Forward Space Tape	19
SCMP	Save CMP Trigger	95	FWDTP	Forward Wind Tape	21
SSGN	Save Sign Trigger	97	RDTP	Read Tape	37
			WRCTP	Write Check Tape	47
MOVE	Move Data	84	RWDTP	Rewind Tape	42
MOVEC	Move Characters	85	RWWLG	Read While Write Tape	23
MOVEI	Move Instruction Addr.	86	WW RTP	Logical (1st time only)	23
			RWWTP	Read While Write Tape	39
LLL	Load Left Location	79	WW RTP		39
LLL14	Load Left Location, ASU14	80	WRETP	Write Erase Tape	59
LRL	Load Right Location	81	WRTM	Write Tape Mark	65
LRL14	Load Right Location, ASU14	82	WRTP	Write Tape	63
DOxxx	DO Subroutine	71	DPTYP	Dump on Typewriter (without checking)	17
CHKT	Check Total	70			17
ORDCH	Order Check	87	TYPCK	Type and Check	45
SEQCH	Sequence Check	99	TYPE	Type (without checking)	43
SETUP	Set ASUs to standard Autocoder configuration	103	WRETY	Write Erase Typewriter (without checking)	61
					61
Floating Decimal Macro-Instructions			Floating Decimal Subroutines		
FLO	Float	119	DO FATN	Arctan	129 & 131
FIX	Fix	118	DO FEX	Exponential	129 & 133
FAB	Absolute	115	DO FLN	Natural Logarithm	129 & 135
FAD	Floating Add	116	DO FSIN	Sine	129 & 137
FDV	Floating Divide	117	DO FSQR	Square Root	129 & 139
FMP	Floating Multiply	120			
FRA	Floating Reset Add	121			
FRS	Floating Reset Subtract	122			
FST	Floating Store	123			
FSU	Floating Subtract	124			
FTM	Floating Transfer Minus	125			
FTP	Floating Transfer Plus	126			
FTZ	Floating Transfer Zero	127			

IBM 705 AUTOCODER MACRO-INSTRUCTION MANUAL

The IBM Autocoder System is a new method of preparing programs for the IBM 705 Electronic Data Processing Machine. Using the speed and logical abilities of the 705, the Autocoder compiles instructions written in a simple notation and translates them into a program in the language of the machine.

In the design of the Autocoder System emphasis was laid on three major points: (1) making the Autocoder simple and easy to use and understand by employing English notation; (2) reducing the time and effort required to write and check out a program by transferring much of the clerical burden of writing, editing and checking to the machine itself, and (3) providing new commands which are desirable either as a convenience in programming or as an extension of the basic logic of the machine.

One of the most important features of the Autocoder resulting from these efforts is its system of macro-instructions. Macro-Instructions are easily written commands to the Autocoder which cause it to create sequences of machine instructions to carry out certain commonly used functions. A typical macro-instruction consists of a single line entry written by the programmer in the same format as a 705 instruction. It will automatically cause the generation of from one to more than a dozen

705 instructions and may also call in desired subroutines of hundreds of instructions. These macro-instructions range in function from input-output instructions to logical aids in programming and to commands for computing in floating point arithmetic.

The ability to use macro-instructions enables a programmer to reduce his coding efforts by writing fewer instructions and his debugging efforts by making it easy for him to use already tested routines.

This manual contains detailed descriptions of the approximately 100 macro-instructions presently available and information on how to construct new ones. It is designed to be used in conjunction with the 705 Autocoder System Manual. The macro-instructions described herein are divided into three sections covering the following subjects: Input-Output Instructions; Logical Instructions; and Floating-Decimal Arithmetic Instructions. A fourth section covers the construction of new macro-instructions.

As new macro-instructions written by IBM and our customers become available for general distribution, they will be released as insertions to the manual.

Working Committee

Goldberg, B.
Goldfinger, R.
Klein, B. B.
Michels, L.
Selden, W. M.
Szabronski, S. J.

Programming Research Dep't.
IBM World Headquarters
590 Madison Avenue
New York 22, New York

SECTION I -

INPUT-OUTPUT MACRO-INSTRUCTIONS

Where applicable the input-output macro-instructions in this section provide for inclusion of the appropriate error subroutine and for an end-of-file transfer address and a restart transfer address. These transfer addresses are optional. If specified, the appropriate transfer will be made. If not, a message will be typed and transfer made to a built-in program stop.

The tape instructions are designed for use with the 754 Control Unit and those for the printer with the 717. Future plans call for the development of suitable macro-instructions for the 777 Tape Record Coordinator and the 760 High-Speed Printer Control Unit.

GENERAL NOTES

The various macro-instructions and their subroutines make use of the accumulator and ASU's 13, 14 and 15. The length setting of the accumulator and ASU 15 is variable. ASU 13 and 14, however, must be set to lengths 10 and 4 respectively before executing a macro-instruction or subroutine. A program produced by the Autocoder System provides for establishing these settings initially and upon completion of execution of a macro-instruction the settings remain the same. Therefore, the programmer need not be concerned with the settings unless he uses ASU's 13 and 14 in his own coding.

Because the accumulator and ASUs are used by the macro-instructions and subroutines, the settings of the sign, zero and comparison triggers may be changed by their execution. In almost all cases the ASU sign trigger is left set to plus. Exceptions to this and changes to the other triggers are noted in each description. It is pointed out that the conditions of the various triggers may be saved prior to execution of a macro-instruction and restored afterwards by use of the Save and Restore series of commands.

The multi-address instructions have succeeding addresses written in the Operand and Comments fields. The convention that has been adopted is that these multiple addresses must each be immediately followed by a lozenge (◊). In cases where an address is optional or left blank, the

lozenge must still be included.

For purposes of modification of and transfer to the various macro-instructions, the Tag written by the programmer is usually applied to the first generated 705 instruction. Reference to the "Coding Produced" portion of each description will show the amount this tag should be incremented to modify any particular generated instruction.

Macro-Instruction: Alternate Tape Units

Operation Code: ALTP

Function: To alternate the tape unit selected by a designated macro-instruction or 705 SEL instruction between two specified tape numbers. May also be used to alternate addresses of other input-output units.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	ALTP		T2 X X1 X	

where: T2 is the tag of the macro-instruction or 705 SEL instruction which is to be alternated.

X1 is the alternate tape unit number, two or three digits, left justified. Leading zeros may be omitted.

Description: The macro-instruction contains two operands which specify the tag of the macro-instruction or 705 SEL instruction which is to have the tape unit number alternated and the alternate tape unit number. The Autocoder generates an in-line sequence of five 705 instructions.

The first time that this sequence is executed in the user's program, the original tape unit number is replaced by the alternate and the alternate is replaced by the original. Thereafter each time the macro-instruction is executed the numbers are reversed.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	14	T2	
	RAD	15	T1 +20	
	UNL	15	T2	
	ST	14	T1 +20	
	AACON	+	X1	(alternate tape unit number.)

- Requirements:
- (a) Applicable Instructions - Macro-instructions RDTP, WRTP, DPTP, WRETP, RWWTP, WW RTP, WRCTP, WRTM, BSTP, FSTP, RWDTP and FWDTP and the 705 SEL instruction.
 - (b) Auxiliary Storage Units - The ASU sign and zero triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters

Macro-Instruction: Backspace Tape (using 754 CU) Operation Code: BSTP

Function: To backspace a designated tape unit a specified number of times.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	BSTP	nn	X2	

where: nn is the last 2 digits of the tape unit number, right justified.
 Zeros may be omitted.
 X2 is the number of times the tape is to be backspaced, left
 justified, of up to 4 digits, leading zeros omitted.

Description: The macro-instruction contains two operands which specify the tape unit and the number of records that the tape is to be backspaced. The Autocoder generates an in-line sequence of seven 705 instructions.

During the execution of this sequence of instructions in the user's program the designated tape unit is selected and the tape is backspaced the specified number of times. Upon completion of the spacing transfer is made to the next instruction written by the user. The tape unit remains selected.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02nn	
	RAD	15	T1 +30	
	BSP			
	SUB	15	(+1)	
	TRZ	15	T1 +35	
	TR		T1 +10	
	AACON	+	X2	

Requirements:

- (a) Auxiliary Storage Units - The ASU sign and zero triggers may be changed, the zero trigger being left on and the sign trigger plus.
- (b) Memory - 35 characters for instructions plus the single character literal constant 1.

Macro-Instruction: Dump on Drum

Operation Code: DPDR

Function: To write a record from a specified address to the end of memory, passing over group marks (WR01) onto a specified drum section or sections. Test for writing and end-of-file error conditions and attempt correction.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DPDR		X1 □ X2 □ X3 □	

where: X1 is the 4 digit number of the starting drum section.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X1, X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the drum section, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the drum error subroutine DRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine DRERR. Here a further analysis is made. If a redundancy has been detected in the memory write-out area, the message "901 R xxxx" is typed and a transfer is made to the restart address. If no restart address is specified the 705 will come to Stop 0901. Pressing the console start key will cause successive rewrites.

If a writing error has occurred the message "902 R xxxx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The specified drum section remains selected. If it is impossible to write on the drum correctly, and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will again cause successive re-writes.

If an erroneous end-of-file condition has occurred, the input-output indicator and the 0902 check indicator will be turned off and the message "EOF R xxxx" typed. If a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0566. Pressing the console start key will cause successive re-writes.

An error in typing will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by the subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		X1	
	WR	01	X2	
	DOA		DRERR	
☒ n	LOD	14	☒ n	
	TRA		DRERR	
	INCL		DRERR	
	RACON		X3	(restart transfer address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program
 - (b) Auxiliary Storage Units - The sign, comparison and zero triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 485 characters for the subroutines DRERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Dump on Printer
(Auto Control, 717 Printer)

Operation Code: DPPRA

Function: To print a line from memory ignoring group marks (WR 01). Test for and take appropriate action upon detection of error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DPPRA	X1	X2 □ X3 □	

where: X1 is the last two digits of the printer number, right justified. Zeros may be omitted.
 X2 is the writing address, either actual, descriptive, literal or blank.
 X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
 X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the printer number, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the printer error subroutine PRERB and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence in the user's program, the ANY indicator is found on it is turned off and a transfer is made to the subroutine PRERB. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 04xx" is typed and a transfer is made to the restart address. If none has been specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 04xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The printer remains selected. If it is impossible to write correctly and a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0902. Again, pressing the console start key will cause successive re-writes.

If a printing error has occurred the message "903 04xx" is typed and the line "PRINT ERROR 2 LINES UP" printed. Transfer is made back to the main program.

An error in typing or printing messages will not be corrected. The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WR	1	X2	
	DOA		PRERB	
n	LOD	14	n	
	TRA		PRERB	
	INCL		PRERB	
	RACON		X3	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 495 characters for the sub-routines PRERB and XOFF and various literal constants which are included in the user's program only once.
 - (d) Printer Control Switch - On single or double space.

Macro-Instruction: Dump on Printer
(Pgm ctl, 717 Printer)

Operation Code: DPPRT

Function: To print a line from memory, ignoring group marks (WR01).
Test for and take appropriate action upon detection of end-of-page
and error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DPPRT	X1	X2 □ X3 □ X4 □	

where: X1 is the last two digits of the printer number, right justified.
Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal
or blank.
X3 is the end-of-page transfer address, either actual; descriptive
or blank (omitted).
X4 is the restart procedure transfer address, either actual,
descriptive or blank (omitted).
X2, X3 and X4 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains four operands which specify the
printer number, the writing address, the end-of-page transfer
address and the restart procedure transfer address. The
Autocoder generates an in-line sequence of six 705 instructions
and causes the inclusion of the printer error subroutine PRERA
and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instruction in the
user's program the ANY indicator is found on, it is turned off
and a transfer is made to the subroutine PRERA. Here a further
analysis is made. If a redundancy error has been detected in
the memory write-out area the message "901 04xx" is typed and
a transfer is made to the restart address. If none has been
specified the 705 will come to Stop 0901. Pressing the console
start key will cause successive re-writes.

If a writing error has occurred the message "902 04xx" is typed
and three re-writes are attempted. If re-writing is successful
a transfer is made back to the main program. The printer re-
mains selected. If it is impossible to write correctly and a
restart address has been specified a transfer is made to it. If
none is specified the 705 will come to Stop 0902. Again, pressing
the console start key will cause successive re-writes.

If a printing error has occurred the message "903 04xx" is typed and the line "PRINT ERROR 2 LINES UP" printed. Transfer is made back to the main program.

If an end-of-page condition has occurred and an end-of-page transfer address has been specified, a transfer is made to it. If none is specified a blank line is printed causing a skip to the start of the next page (channel 1). Again transfer is made back to the main program.

An error in typing or printing messages will not be corrected. The check indicator 0901, 0902 and 0903 and the ANY indicator will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WR	01	X2	
	DOA		PRERA	
M _n	LOD	14	M _n	
	TRA		PRERA	
	INCL		PRERA	
	RAGON		X3	(end-of-page address)
	RACON		X4	(restart address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 and 0903 on Program.
- (b) Auxiliary Storage Units - The Sign, zero and comparison triggers may be changed. The sign trigger is left plus.
- (c) Memory - 30 characters each time the macro-instruction is written plus 716 characters for the sub-routines PRERA and XOFF and various literal constants which are included in the user's program only once.

Macro-Instruction: Dump on Punch

Operation Code: DPPCH

Function: To write a record from memory, ignoring group marks (WR01), punching it on a card. Test for and take appropriate action upon detection of errors.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DPPCH	X1	X2 □ X3 □	

where: X1 is the last two digits of the punch number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the punch number, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the punch error subroutines PNERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine PNERR. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 03xx" is typed and a finder card is punched with the message "ERROR PREV CD." If a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 03xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. If it is impossible to write correctly and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 902. Again, pressing the console start key will cause successive re-writes.

If a punching error has occurred the message "903 03xx" is typed and a finder card is punched with the message "ERROR 3RD CD FWD". Then transfer is made back to the main program.

An error in typing or punching finder cards will not be corrected. The ANY indicator and 0901, 0902 and 0903 check indicators will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		03X1	
	WR	01	X2	
	DOA		PNERR	
H _n	LOD	14	H _n	
	TRA		PNERR	
	INCL		PNERR	
	RACON		X3	(restart transfer address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 and 0903 on Program.
- (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
- (c) Memory - 25 characters each time the macro-instruction is written plus 503 characters for the subroutines PNERR and XOFF and literal constants which are included in the user's program only once.

Macro-Instruction: Dump on Tape (using 754 CU) Operation Code: DPTP

Function: To select a designated tape unit and write the contents of memory on tape from a specified location to the upper end of memory, ignoring group marks (Write 01). Test for and take appropriate action upon detection of end-of-file and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND			COMMENTS
T1	DPTP	X1	X2 ▢	X3 ▢	X4 ▢	

where: X1 is the last 2 digits of the tape unit number, right justified. Zeros may be omitted.
 X2 is the writing address, either actual, descriptive, literal or blank.
 X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted).
 X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2, X3 and X4 must be immediately followed by a lozenge (▢).

Description: The macro-instruction contains four operands which specify the tape unit, the writing address, the end-of-file transfer address and the restart procedure transfer address. The Autocoder generates an inline sequence of six 705 instructions and causes the inclusion of the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine TPERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 R 02xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 R 02xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The tape unit remains selected. If it is impossible to successfully write the record and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-writes.

If an end-of-file condition has been detected by sensing the reflective spot on the end of the tape, the input-output indicator will be turned off and a transfer made to the address specified. If none is specified the message "EOF R 02xx" is typed and the 705 will come to Stop 0566. Again, pressing the console start key will cause successive re-writes.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	WR	01	X2	
	DOA		TPERR	
Π n	LOD	14	Π n	
	TRA		TPERR	
	INCL		TPERR	
	RACON		X3	(end-of-file transfer address)
	RACON		X4	(restart procedure address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches-0902 on Program.
- (b) Auxiliary Storage Units -The setting of the ASU sign, comparison and zero triggers may be changed. The sign trigger will be set plus.
- (c) Memory - 30 characters each time the macro-instruction is written plus 532 characters for the subroutines TPERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Dump on Typewriter (without checking) Operation Code:
DPTYP

Function: To select a designated typewriter and write a record from memory, ignoring group marks (WR01). Turn off the ANY indicator and check indicators 0901 and 0902 if they are turned on.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DPTYP	X1	X2	

where: X1 is the last 2 digits of the typewriter number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.

Description: The macro-instruction contains two operands which specify the typewriter and the writing address. The Autocoder generates an in-line sequence of four 705 instructions and causes the inclusion of the typewriter subroutine XOFF at the end of the program.

If during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer made to the subroutine. The subroutine turns off the 0901 and 0902 check indicators and transfers back to the main program.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		05X1	
	WR	01	X2	
	DOA		XOFF	
H n	LOD	14	H n	
	TRA		XOFF	
	INCL		XOFF	

Requirements:

- (a) Check Indicators - 0902 on Program.
- (b) Memory - 20 characters each time the macro-instruction is written plus 44 characters for the subroutine XOFF and a literal constant which are included only once.

Macro-Instruction: Forward Space Tape (using 754 CU) Operation Code: FSTP

Function: To forward space a designated tape unit a specified number of times without reading records into memory (Read 01).

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FSTP	X1	X2	

where: X1 is the last 2 digits of the tape unit number, right justified.
 Zeros may be omitted.
 X2 is the number of times the tape is to be forward spaced, left justified of up to 4 digits, leading zeros omitted.

Description: The macro-instruction contains two operands which specify the tape unit and the number of records that the tape is to be spaced forward. The Autocoder generates an in-line sequence of thirteen 705 instructions.

During the execution of this sequence of instructions in the user's program the designated tape unit is selected and the tape is spaced forward the specified number of times using the Read 01 instruction. The records read will not enter memory. Upon completion of the spacing the ANY and 0902 indicators are turned off and transfer is made to the next instruction written by the programmer. The tape unit remains selected.

If a tape mark (end-of-file) is sensed before completion of the specified number of forward spaces, the forward spacing is discontinued and the tape is backspaced once. Transfer is then made to the programmer's next instruction after turning off the ANY and 0902 indicators. The input-output indicator is left on. It may either be interrogated by the programmer or the end-of-file condition may be detected during a subsequent read operation when the tape mark is again sensed.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	RAD	15	T1 +35	
	RD	1	0	
	TRS		T1 +40	
	SUB	15	(+1)	
	TRZ	15	T1 +45	
	TR		T1 +10	
	AACON	+	X2	

Coding Produced(continued:)

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	BSP			
	TRA		T1 +50	
	SEL		902	
	TRS		T1 +60	
	SEL		02X1	

Requirements:

- (a) Check Indicator Switches-0902 on Program.
- (b) Auxiliary Storage Units -The ASU sign and zero triggers may be changed. The sign trigger will be set plus. The zero trigger will be on if the required number of spacings are completed before sensing a tape mark.
- (c) Memory - 65 characters for instructions plus the single character literal constant +1.

Macro-Instruction: Forward Wind Tape (using 754 CU) Operation Code: FWDTP

Function: To wind a designated tape forward to the next tape mark and position it to read the next file on the same tape. Used primarily to space over files on a multiple file tape.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FWDTP	X1		

where: X1 is the last 2 digits of the tape unit number, right justified.
Zeros may be omitted.

Description: The macro-instruction contains one operand which specifies the tape unit to be wound forward. The Autocoder generates an in-line sequence of nine 705 instructions.

During the execution of this sequence of instructions in the user's program the designated tape unit is selected and the tape is spaced forward using Read 01 (records do not enter memory). When the next tape mark is sensed (end-of-file) the input-output indicator, ANY indicator and 0902 check indicator are turned off. The tape is now positioned to read the first record of the next file. The 705 then proceeds to the next instruction written by the programmer.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	RD	1	@ 0	
	TRS		T1 +20	
	TR		T1 +5	
	TRA		T1 +25	
	SEL		902	
	TRS		T1 +35	
	SEL		02X1	
	IOF			

Requirements:

- (a) Check Indicator Switches - 0902 on Program.
- (b) Memory - 45 characters.

Macro-Instruction: Logical Read-While-Write Tape Operation Code: RWWLG
and
WW RTP

Function: To select a designated input tape unit and read a record into memory the first time the macro-instruction is executed and thereafter select designated input and output tape units and simultaneously read a record into memory while writing a record on tape. Test for and take appropriate action upon detection of end-of-file and reading and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND			COMMENTS			
T1	RWWLG	X1	X2	□	X3	□	X4	□	
T2	WW RTP	X5	X6	□	X7	□	X8	□	

- where:
- X1 is the last 2 digits of the input tape unit number, right justified. Zeros may be omitted.
 - X2 is the reading address, either actual, descriptive, literal or blank.
 - X3 is the input end-of-file transfer address, either actual, descriptive or blank (omitted).
 - X4 is the input restart procedure transfer address, either actual, descriptive or blank (omitted).
 - X5 is the last 2 digits of the output tape unit number, right justified. Zeros may be omitted.
 - X6 is the writing address, either actual, descriptive, literal or blank.
 - X7 is the output end-of-file transfer address, either actual descriptive or blank (omitted).
 - X8 is the output restart procedure transfer address, either actual descriptive or blank (omitted).

The operation codes RWWLG and WW RTP must be used together and in that order with no intervening operations. X2, X3, X4, X6, X7 and X8 must be immediately followed by a lozenge (□).

Description: The macro-instruction consists of the two operation codes RWWLG (Logical Read-While-Write Tape) and WW RTP (Write While Read Tape) which are used together. Each contains four operands, which for RWWLG specify the input tape unit, reading address, input end-of-file transfer address and input restart procedure transfer address. The operands for WW RTP similarly specify the output addresses. The Autocoder generates an in-line sequence of nineteen 705 instructions and includes the read while write error subroutine RWER, the tape error subroutine TPER and the typewriter subroutine XOFF at end of the program.

The first time that this sequence of instructions is executed in the user's program only the input tape is read. If the ANY indicator is found on, it is turned off and a transfer is made to the tape error subroutine TPERR where correction is attempted in the same manner as explained below for a read error occurring during a read while write. If the reading was correct a switch is set which causes the instruction sequence to perform a simultaneous read while writing operation during subsequent executions.

If, during the read while writing, the ANY indicator is found on, it is turned off and transfer is made to the subroutine RWWER. Here the writing and reading operations are set up as two distinct sequences of instructions and the cause of the error analyzed for each operation in turn using the tape error subroutine TPERR.

If a redundant character has been detected in the output area of memory the message "901 R 02xx" is typed and a transfer is made to the output restart address. If an output restart address is not specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes of the output tape.

If a writing error has occurred the message "902 R 02xx" is typed and three re-writes of the output tape are attempted. If it is impossible to successfully write the record and an output restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-writes of the output tape.

If the re-writing was successful the reading of the input tape is then checked. If a reading error has occurred the message "902 Y 02xx" is typed and three re-reads of the input tape are attempted. Again if it is impossible to successfully read the record and an input restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-reads of the input tape.

If an end-of-file condition has occurred on either the input or output tape the appropriate input-output indicator will be turned off and a transfer made to the address specified. If none is specified either the message "EOF Y 02xx" or "EOF R 02xx" will be typed and the 705 will come to Stop 0566. Pressing the console start key will cause successive re-reads or re-writes depending upon which tape had the end-of-file condition.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	NOP		T1 +45	
	SEL		02X1	
	RD		X2	
	DOA		TPERR	
□ m	LOD	14	□ m	
	TRA		TPERR	
	INCL		TPERR	
	RACON		X3	(input end-of-file address)
	RACON		X4	(input restart address)
	SGN	15	T1 -4	
	TR		T1 +95	
	SEL		02X1	
	RWW		X2	
	RACON		X3	(input end-of-file address)
	RACON		X4	(input restart address)
	INCL		XOFF	
T2	SEL		02X5	
	WR		X6	
	DOA		RWWER	
□ n	LOD	14	□ n	
	TRA		RWWER	
	INCL		RWWER	
	RACON		X7	(output end-of-file address)
	RACON		X8	(output restart address)
	INCL		TPERR	
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 on Program.
- (b) Auxiliary Storage Units - The ASU sign, comparison and zero triggers may be changed, the sign trigger being set to plus.
- (c) Memory - 95 characters each time the macro-instruction is written plus 783 characters for the subroutines RWWER, TPERR and XOFF and various literal constants. The subroutines and constants are included in the user's program only once.

Macro-Instruction: Print under Automatic Control Operation Code: PRNTA
 (717 Printer)

Function: To print a line from memory. Test for and take appropriate action upon detection of error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	PRNTA	X1	X2 □ X3 □	

where: X1 is the last two digits of the printer number, right justified. Zeros may be omitted.
 X2 is the writing address, either actual, descriptive, literal or blank.
 X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
 X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the printer number, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the printer error subroutine PRERB and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence in the user's program, the ANY indicator is found on it is turned off and a transfer is made to the subroutine PRERB. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 04xx" is typed and a transfer is made to the restart address. If none has been specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 04xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The printer remains selected. If it is impossible to write correctly and a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0902. Again, pressing the console start key will cause successive re-writes.

If a printing error has occurred the message "903 04xx" is typed and the line "PRINT ERROR 2 LINES UP" printed. Transfer is made back to the main program.

An error in typing or printing messages will not be corrected. The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WR		X2	
	DOA		PRERB	
$\bar{A}n$	LOD	14	$\bar{A}n$	
	TRA		PRERB	
	INCL		PRERB	
	RACON		X3	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 495 characters for the subroutines PRERB and XOFF and various literal constants which are included in the user's program only once.
 - (d) Printer Control Switch - On single or double space.

Macro-Instruction: Print under Program Control Operation Code: PRINT
(717 Printer)

Function: To print a line from memory. Test for and take appropriate action upon detection of end-of-page and error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	PRINT	X1	X2 ◊ X3 ◊ X4 ◊	

where: X1 is the last two digits of the printer number, right justified. Zeros may be omitted.
 X2 is the writing address, either actual, descriptive, literal or blank.
 X3 is the end-of-page transfer address, either actual, descriptive or blank (omitted).
 X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
 X2, X3 and X4 must be immediately followed by a lozenge (◊).

Description: The macro-instruction contains four operands which specify the printer number, the writing address, the end-of-page transfer address and the restart procedure transfer address. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the printer error subroutine PRERA and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine PRERA. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 04xx" is typed and a transfer is made to the restart address. If none has been specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 04xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The printer remains selected. If it is impossible to write correctly and a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0902. Again, pressing the console start key will cause successive re-writes.

If a printing error has occurred the message "903 04xx" is typed and the line "PRINT ERROR 2 LINES UP" printed. Transfer is made back to the main program.

If an end-of-page condition has occurred and an end-of-page transfer address has been specified, a transfer is made of it. If none is specified a blank line is printed causing a skip to the start of the next page (channel 1). Again transfer is made back to the main program.

An error in typing or printing messages will not be corrected. The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WR		X2	
	DOA		PRERA	
\bar{M}_n	LOD	14	\bar{M}_n	
	TRA		PRERA	
	INCL		PRERA	
	RACON		X3	(end-of-page address)
	RACON		X4	(restart address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 and 0903 on Program.
- (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
- (c) Memory - 30 characters each time the macro-instruction is written plus 716 characters for the subroutines PRERA and XOFF and various literal constants which are included in the user's program only once.

Macro-Instruction: Punch

Operation Code: PUNCH

Function: To write a record from memory, punching it on a card. Test for and take appropriate action upon detection of errors.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	PUNCH	X1	X2□ X3□	

where: X1 is the last two digits of the punch number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual descriptive or blank (omitted).
X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the punch number, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the punch error subroutine PNERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine PNERR. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 03xx" is typed and a finder card is punched with the message "ERROR PREV CD." If a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0901. Pressing the console start key will cause success re-writes.

If a writing error has occurred the message "902 03xx" is typed and three rewrites are attempted. If rewriting is successful a transfer is made back to the main program. If it is impossible to write correctly and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 902. Again, pressing the console start key will cause successive rewrites.

If a punching error has occurred the message "903 03xx" is typed and a finder card is punched with the message "ERROR 3RD CD FWD." Then transfer is made back to the main program.

An error in typing or punching finder cards will not be corrected. The ANY indicator and 0901, 0902 and 0903 check indicators will simply be turned off.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		03X1	
	WR		X2	
	DOA		PNERR	
\bar{A}_n	LOD	14	\bar{A}_n	
	TRA		PNERR	
	INCL		PNERR	
	RACON		X3	(restart transfer address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 503 characters for the subroutines PNERR and XOFF and various literal constants which are included in the user's program only once.

Macro-Instruction: Read Card

Operation Code: RDCD

Function: To select a designated card reader and read a card into memory.
Test for and take appropriate action upon detection of end-of-file and reading error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RDCD	X1	X2 ◻ X3 ◻	

where: X1 is the last 2 digits of the card reader number, right justified. Zeros may be omitted.
X2 is the reading address, either actual, descriptive, literal or blank.
X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted.)
X2 and X3 must be immediately followed by a lozenge (◻).

Description: The macro-instruction contains three operands which specify the card reader number, the reading address and the end-of-file transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the card reader error subroutine CDERR and the typewriter subroutine XOFF at the end of the program.

If during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine CDERR. Here a further analysis is made. If reading error has occurred the message "902 0lxx" is typed and the 705 will come to Stop 0902. The operator may then correct the card in error and reload the card reader starting with the corrected card. Pressing the console start key will cause the corrected card to be re-read.

If an end-of-file condition has occurred, the input-output indicator will be turned off and a transfer made to the address specified. If none is specified the message "EOF 0lxx" is typed and the 705 will come to Stop 0566. If the card reader is loaded with more cards, pressing the start key will cause the program to continue, reading the next card.

An error in typing messages will not be corrected. The 0901 and 0902 check indicators and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		01X1	
	RD		X2	
	DOA		CDERR	
\bar{A}_n	LOD	14	\bar{A}_n	
	TRA		CDERR	
	INCL		CDERR	
	RACON		X3	(end-of-file address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 on Program.
- (b) Auxiliary Storage Units - The ASU sign, comparison and zero triggers may be changed, the sign trigger being set to plus.
- (c) Memory - 25 characters each time the macro-instruction is written plus 264 characters for the subroutines CDERR and XOFF and various literal constants. The subroutines and literal constants are included in the user's program only once.

Macro-Instruction: Read Drum

Operation Code: RDDR

Function: To read a specified drum section into memory. Test for reading and end-of-file error conditions and attempt correction.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RDDR		X1 □ X2 □ X3 □	

where: X1 is the 4 digit number of the starting drum section.
X2 is the reading address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).

X1, X2, and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the drum section, the reading address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the drum error subroutine DRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine DRERR. Here a further analysis is made. If a reading error has occurred the message "902 Yxxxx" is typed and three re-reads are attempted. If re-reading is successful a transfer is made back to the main program. The specified drum section remains selected. If it is impossible to read from the drum correctly and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-reads.

If an erroneous end-of-file condition has occurred, the input-output indicator and the 0902 check indicator will be turned off and the message "EOF Y xxxx" typed. If a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0566. Pressing the console start key will cause successive re-reads.

An error in typing will not be corrected. The check indicators 0901 and the ANY indicator will simply be turned off by the subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		X1	
	RD		X2	
	DOA		DRERR	
□ n	LOD	14	□ n	
	TRA		DRERR	
	INCL		DRERR	
	RACON		X3	(restart transfer address)
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 on Program
- (b) Auxiliary Storage Units - The sign, comparison and zero triggers may be changed. The sign trigger is left plus.
- (c) Memory - 25 characters each time the macro-instruction is written plus 485 characters for the subroutines DRERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Read Tape (using 754 CU)

Operation Code: RDTP

Function: To select a designated tape unit and read a record into memory.
Test for and take appropriate action upon detection of end-of-file and reading error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RDTP	X1	X2 □ X3 □ X4 □	

where: X1 is the last 2 digits of the tape unit, right justified. Zeros may be omitted.
X2 is the reading address, either actual, descriptive, literal or blank.
X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted).
X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).

X2, X3 and X4 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains four operands which specify the tape unit, the reading address, the end-of-file transfer address and the restart procedure transfer address. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine TPERR. Here a further analysis is made. If a reading error has occurred the message "902 Y 02xx" is typed and three re-reads are attempted. If re-reading is successful a transfer is made back to the main program. The tape unit remains selected. If it is impossible to successfully read the record and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-reads.

If an end-of-file condition has occurred, the input-output indicator will be turned off and a transfer is made to the address specified. If none is specified the message "EOF Y 02xx" is typed and the 705 will come to Stop 0566. Pressing the console start key will cause successive re-reads.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	RD		X2	
	DOA		TPERR	
I _n	LOD	14	I _n	
	TRA		TPERR	
	INCL		TPERR	
	RACON		X3	
	RACON		X4	
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 Program.
 - (b) Auxiliary Storage Units - The ASU sign, comparison and zero triggers may be changed, the sign trigger being set to plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 532 characters for the subroutines TPERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Read-While-Write Tape (using 754 CU) Operation Code: RWWTP
and
WWRTP

Function: To select designated input and output tape units and simultaneously read a record into memory while writing a record on tape. Test for and take appropriate action upon detection of end-of-file and reading and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND			COMMENTS
T1	RWWTP	X1	X2 □	X3 □	X4 □	
T2	WWRTP	X5	X6 □	X7 □	X8 □	

- where:
- X1 is the last 2 digits of the input tape unit number, right justified. Zeros may be omitted.
 - X2 is the reading address, either actual, descriptive, literal or blank.
 - X3 is the input end-of-file transfer address, either actual, descriptive or blank (omitted).
 - X4 is the input restart procedure transfer address, either actual, descriptive or blank (omitted).
 - X5 is the last 2 digits of the output tape unit number, right justified. Zeros may be omitted.
 - X6 is the writing address, either actual, descriptive, literal or blank.
 - X7 is the output end-of-file transfer address, either actual descriptive or blank (omitted).
 - X8 is the output restart procedure transfer address, either actual, descriptive or blank (omitted).

The operation codes RWWTP and WWRTP must be used together and in that order with no intervening operations. X2, X3, X4, X6, X7 and X8 must be immediately followed by a lozenge (□).

Description: The macro-instruction consists of the two operation codes RWWTP (Read While Write Tape) and WWRTP (Write While Read Tape) which are used together. Each contains four operands, which for RWWTP specify the input tape unit, reading address, input end-of-file transfer address and input restart procedure transfer address. The operands for WWRTP similarly specify the output addresses. The Autocoder generates an in-line sequence of ten 705 instructions and includes the read while write error subroutine RWWER, the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine RWWER. Here the writing and reading operations are set up as two distinct sequences of instructions and the cause of the error analyzed for each operation in turn using the tape error subroutine TPERR.

If a redundant character has been detected in the output area of memory the message "901 R 02xx" is typed and a transfer is made to the output restart address. If an output restart address is not specified the 705 will come to Stop 0901. Pressing the console start key will cause successive rewrites of the output tape.

If a writing error has occurred the message "902 R 02xx" is typed and three rewrites of the output tape are attempted. If it is impossible to successfully write the record and an output restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive rewrites of the output tape.

If the rewriting was successful the reading of the input tape is then checked. If a reading error has occurred the message "902 Y 02xx" is typed and three rereads of the input tape are attempted. Again if it is impossible to successfully read the record and an input restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive rereads of the input tape.

If an end-of-file condition has occurred on either the input or output tape the appropriate input-output indicator will be turned off and a transfer made to the address specified. If none is specified either the message "EOF Y 02xx" or "EOF R 02xx" will be typed and the 705 will come to Stop 0566. Pressing the console start key will cause successive rereads or rewrites depending upon which tape has the end-of-file condition.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutines XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	RWW		X2	
	RACON		X3	(input end-of-file address)
	RACON		X4	(input restart address)
T2	SEL		02X5	
	WR		X6	
	DOA		RWWER	
⌈ n	LOD	14	⌈ n	
	TRA		RWWER	
	INCL		RWWER	
	RACON		X7	(output end-of-file address)
	RACON		X8	(output restart address)
	INCL		TPERR	
	INCL		XOFF	

Requirements:

- (a) Check Indicator Switches - 0902 on Program.
- (b) Auxiliary Storage Units - The ASU sign, comparison and zero triggers may be changed, the sign trigger being set to plus.
- (c) Memory - 50 characters each time the macro-instruction is written plus 783 characters for the subroutines RWWER, TPERR and XOFF and various literal constants. The subroutines and constants are included in the user's program only once.

Macro-Instruction: Rewind Tape (using 754 CU) Operation Code: RWDTP

Function: To rewind a designated tape unit and turn off its input-output indicator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RWDTP	X1		

where: X1 is the last 2 digits of the tape unit number, right justified.
Zeros may be omitted.

Description: The macro-instruction contains one operand which specifies the tape unit. The Autocoder generates an in-line sequence of three 705 instructions.

During the execution of this sequence of instructions in the user's program the designated tape unit is selected and rewinding is started. The 705 then proceeds to the next instruction written by the programmer.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	RWD			
	IOF			

Requirements: (a) Memory 15 characters.

Macro-Instruction: Type (without error correcting) Operation Code: TYPE

Function: To select a designated typewriter and write a record from memory.
Turn off the ANY indicator and check indicators 0901 and 0902, if they are turned on. Used primarily to type messages.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	TYPE	X1	X2	

where: X1 is the last 2 digits of the typewriter number, right justified.
Zeros may be omitted.

X2 is the writing address, either actual, descriptive, literal or blank.

Description: The macro-instruction contains two operands which specify the typewriter and the writing address. The Autocoder generates an in-line sequence of four 705 instructions and causes the inclusion of the typewriter subroutine XOFF at the end of the program.

If during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer made to the subroutine. The subroutine turns off the 0901 and 0902 check indicators and transfers back to the main program.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		05X1	
	WR		X2	
	DOA		XOFF	
n	LOD	14	n	
	TRA		XOFF	
	INCL		XOFF	

Requirements: (a) Check Indicators - 0902 on Program.
(b) Memory - 20 characters each time the macro-instruction is written plus 44 characters for the subroutine XOFF and a literal constant which are included only once.

Macro-Instruction: Type and Check

Operation Code: TYPCK

Function: To write a record on the typewriter. Test for and take appropriate action upon detection of errors.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TYPCK	X1	X2 ▯ X3 ▯	

where: X1 is the last 2 digits of the typewriter number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2 and X3 must be immediately followed by a lozenge (▯).

Description: The macro-instruction contains three operands which specify the typewriter number, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the typewriter error subroutines TYPER and XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine TYPER. Here a further analysis is made. If a redundancy error has been detected in the memory write-out area the message "901 05xx" is typed and a transfer is made to the restart address. If none has been specified the 705 will come to Stop 0901. Pressing the console start key will cause successive rewrites.

If a writing error has occurred the message "902 05xx" is typed and three rewrites are attempted. If rewriting is successful a transfer is made back to the main program. If it is impossible to write correctly and a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0902. Again, pressing the console start key will cause successive rewrites.

An error in typing error messages will not be corrected. The 0901 and 0902 check indicators and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		05X1	
	WR		X2	
	DOA		TYPER	
\bar{M}_n	LOD	14	\bar{M}_n	
	TRA		TYPER	
	INCL		TYPER	
	RACON		X3	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 358 characters for the subroutines TYPER and XOFF and various literal constants which are included only once.

Macro-Instruction: Write Check Tape (using 754 CU) Operation Code: WRCTP

Function: To select a designated tape unit, write a record from memory and re-read it to check that it has been written correctly. Test for and take appropriate action upon detection of end-of-file and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND		COMMENTS
T1	WRCTP	X1	X2	□	X3 □ X4 □

where:

- X1 is the last 2 digits of the tape unit number, right justified. Zeros may be omitted.
- X2 is the writing address, either actual, descriptive, literal or blank.
- X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted).
- X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).

X2, X3 and X4 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains four operands which specify the tape unit, the writing address, the end-of-file transfer address and the restart procedure transfer address. The Autocoder generates an in-line sequence of eight 705 instructions and causes the inclusion of the write check error subroutine CWRER, the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine CWRER. Here the writing and re-reading operations are set up as two distinct sequences of instructions and the cause of the error analyzed for each operation in turn using the tape error subroutine TPERR.

If a redundancy has been detected in the memory write-out area, the message "901 R 02xx" is typed and a transfer made to the restart address. If no restart address has been specified, the 705 will come to Stop 0901. Pressing the console start key will cause successive rewrites.

If a writing error has occurred the message "902 R 02xx" is typed and three re-writes are attempted. If it is impossible to successfully write the record and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-writes.

If the re-writing was successful the record is re-read to check that it is on tape correctly. If a reading error occurs the message "902 Y 02xx" is typed and three re-reads are attempted. If re-reading is successful a transfer is made back to the main program. The tape unit remains selected. Again, if it is impossible to successfully read the record and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-reads.

If an end-of-file condition has occurred, the input-output indicator will be turned off and a transfer made to the address specified. If none is specified the message "EOF R 02xx" is typed and the 705 will come to Stop 0566.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	WR		X2	
	BSP			
	RD	1	@ 0	
	DOA		CWERR	
A _n	LOD	14	A _n	
	TRA		CWRER	
	INCL		CWRER	
	RACON		X3	(end-of-file address)
	RACON		X4	(restart address)
	INCL		TPERR	
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program.
 - (b) Auxiliary Storage Units - The setting of the ASU sign, zero and comparison triggers may be changed, the sign trigger being set plus.
 - (c) Memory - 40 characters each time the macro-instruction is written plus 757 characters for the subroutines CWRER, TPERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Write Drum

Operation Code: WRDR

Function: To write a record from memory to a specified drum section or sections. Test for writing and end-of-file error conditions and attempt correction.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WRDR		X1 ▣ X2 ▣ X3 ▣	

where: X1 is the 4 digit number of the starting drum section.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address.
X1, X2 and X3 must be immediately followed by a lozenge (▣).

Description: The macro-instruction contains three operands which specify the drum section, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of five 705 instructions and causes the inclusion of the drum error subroutine DRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program the ANY indicator is found on, it is turned off and a transfer is made to the subroutine DRERR. Here a further analysis is made. If a redundancy error has been detected in the memory unit out area the message "901 R xxxx" is typed and a transfer is made to the restart address. If none is specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 R xxxx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The specified drum section remains selected. If it is impossible to write on the drum correctly and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Again, pressing the console start key will cause successive re-writes.

If an erroneous end-of-file condition has occurred, the input-output indicator and the 0902 check indicator will be turned off and the message "EOF R xxxx" typed. If a restart address has been specified a transfer is made to it. If none is specified the 705 will come to Stop 0566. Pressing the console start key will cause successive re-writes.

An error in typing will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by the subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		X1	
	WR		X2	
	DOA		DRERR	
\bar{A}_n	LOD	14	\bar{A}_n	
	TRA		DRERR	
	INCL		DRERR	
	RACON		X3	(restart transfer address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program
 - (b) Auxiliary Storage Units - The sign, comparison and zero triggers may be changed. The sign trigger is left plus.
 - (c) Memory - 25 characters each time the macro-instruction is written plus 485 characters for the subroutines DRERR and XOFF and various literal constants which are included in the user's program only once.

Macro-Instruction: Write Erase Printer
(717 Printer, Auto Ctl)

Operation Code: WREPA

Function: To write a record from memory onto a printer, setting the memory record area to blanks. Test for and take appropriate action upon detection of error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WREPA	X1	X2 \boxtimes X3 \boxtimes	

where: X1 is the last 2 digits of the printer number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2 and X3 must be immediately followed by a lozenge (\boxtimes).

Description: The macro-instruction contains three operands which specify the printer, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the write erase error subroutine WRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and a transfer is made to the subroutine WRERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 Z 04xx" is typed and transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901.

If a writing error has occurred the message "902 Z 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0902.

If a printing error has occurred the message "903 Z 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0903.

Pressing the console start key after any of the above stops will cause a return to the Stop. An error in typing messages will not be corrected. The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WRE		X2	
	DOA		WRERR	
I n	LOD	14	I n	
	TRA		WRERR	
	INCL		WRERR	
	RACON		T1 +30	
	RACON		X3	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed by the macro-instruction and subroutine. The sign trigger being left plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 528 characters for the subroutine and various literal constants are included in the program only once.
 - (d) Printer Control Switch - On single or double space.

Macro-Instruction: Write Erase Printer
(717 Printer under pgm ctl)

Operation Code: WREPR

Function: To write a record from memory onto a printer, setting the memory area to blanks. Test for and take appropriate action upon detection of end-of-page and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WREPR	X1	X2 ▣ X3 ▣ X4 ▣	

where: X1 is the last 2 digits of the printer number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the end-of-page transfer address, either actual, descriptive or blank (omitted).
X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2, X3 and X4 must be immediately followed by a lozenge (▣).

Description: The macro-instruction contains three operands which specify the printer, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the write erase error subroutine WRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and a transfer is made to the subroutine WRERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 Z 04xx" is typed and transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901.

If a writing error has occurred the message "902 Z 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0902.

If a printing error has occurred the message "903 Z 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0903.

Pressing the console start key after any of the above stops will cause a return to the Stop. An error in typing messages will not be corrected.

The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		04X1	
	WRE		X2	
	DOA		WRERR	
$\bar{M}n$	LOD	14	$\bar{M}n$	
	TRA		WRERR	
	INCL		WRERR	
	RACON		X3	(end-of-page address)
	RACON		X4	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed by the macro-instruction and subroutine. The sign trigger being left plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 528 characters for the subroutine WRERR and XOFF and various literal constants. The subroutines and constants are included in the program only once.
 - (d) Printer Control Switch - On Program.

Macro-Instruction: Write Erase Punch

Operation Code: WREPN

Function: To write a record from memory, punching it on a card and setting the memory record area to blanks. Test for and take appropriate action upon detection of errors.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WREPN	X1	X2 □ X3 □	

where: X1 is the last two digits of the punch number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the restart procedure transfer address, either actual, descriptive or blank (omitted).
X2 and X3 must be immediately followed by a lozenge (□).

Description: The macro-instruction contains three operands which specify the punch, the writing address and the restart procedure transfer address. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the write erase error subroutine WRERR and the typewriter subroutine XOFF at the end of the program.

If during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine WRERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 Z 04xx" is typed and transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901.

If a writing error has occurred the message "902 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0902.

If a printing error has occurred the message "903 Z 04xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0903.

Pressing the console start key after any of the above stops will cause a return to the Stop. An error in typing messages will not be corrected. The check indicators 0901, 0902 and 0903 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		03X1	
	WRE		X2	
	DOA		WRERR	
$\bar{A}n$	LOD	14	$\bar{A}n$	
	TRA		WRERR	
	INCL		WRERR	
	RACON		T1 +30	
	RACON		X3	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 and 0903 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed by the macro-instruction and subroutine. The sign trigger being left plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 528 characters for the subroutine WREPR and XOFF and various literal constants. The subroutines and constants are included in the program only once.

Macro-Instruction: Write Erase Tape (using 754 CU) Operation Code: WRETP

Function: To select a designated tape unit and write a record from memory, setting the memory record area to blanks. Test for and take appropriate action upon detection of end-of-file and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND				COMMENTS
T1	WRETP	X1	X2	□	X3	□	X4 □

where:

- X1 is the last 2 digits of the tape unit number, right justified. Zeros may be omitted.
- X2 is the writing address, either actual, descriptive, literal or blank,
- X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted).
- X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).

X2, X3 and X4 must be immediately followed by a lozenge. (□)

Description: The macro-instruction contains four operands which specify the tape unit, the writing address, the end-of-file transfer address and the restart procedure transfer address. The Autocoder generates an inline sequence of six 705 instructions and causes the inclusion of the write erase error subroutine WRERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine WRERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 Z 02xx" is typed and transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901.

If a writing error has occurred the message "902 Z 02xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0902.

If an end-of-file condition has been detected by sensing the reflective spot on the end of the tape, the input-output indicator is turned off and a transfer made to the address specified. The tape unit remains selected. If no address is specified the message "EOF Z 02xx" is typed and the 705 will come to Stop 0566.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	WRE		X2	
	DOA		WRERR	
$\bar{A}n$	LOD	14	$\bar{A}n$	
	TRA		WRERR	
	INCL		WRERR	
	RACON		X3	(end-of-file address)
	RACON		X4	(restart address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program.
 - (b) Auxiliary Storage Units - The sign, zero and comparison triggers may be changed by the macro-instruction and subroutine. The sign trigger being left plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 528 characters for the subroutine and various literal constants. The subroutines and constants are included in the program only once.

Macro-Instruction: Write Erase Typewriter
(without error correcting)

Operation Code: WRETY

Function: To write a record from memory onto a typewriter, setting the memory record area to blanks. Turn off the ANY indicator and check indicators 0901 and 0902, if they are turned on.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WRETY	X1	X2	

where: X1 is the last 2 digits of the typewriter number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.

Description: The macro-instruction contains two operands which specify the typewriter and the writing address. The Autocoder generates an in-line sequence of four 705 instructions and causes the inclusion of the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and a transfer made to the subroutine. The subroutine turns off the 0901 and 0902 check indicators and transfers back to the main program.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		05X1	
	WRE		X2	
	DOA		XOFF	
Mn	LOD	14	Mn	
	TRA		XOFF	
	INCL		XOFF	

Requirements: (a) Check Indicator Switches - 0902 on Program.
(b) Memory - 20 characters each time the the macro-instruction is written plus 44 characters for the subroutine XOFF and a literal constant which is included only once.

Macro-Instruction: Write Tape (using 754 CU)

Operation Code: WRTP

Function: To select a designated tape unit and write a record from memory onto tape. Test for and take appropriate action upon detection of end-of-file and writing error conditions.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WRTP	X1	X2 ▯ X3 ▯ X4 ▯	

where: X1 is the last 2 digits of the tape unit number, right justified. Zeros may be omitted.
X2 is the writing address, either actual, descriptive, literal or blank.
X3 is the end-of-file transfer address, either actual, descriptive or blank (omitted).
X4 is the restart procedure transfer address, either actual, descriptive or blank (omitted).

X2, X3 and X4 must be immediately followed by a lozenge (▯).

Description: The macro-instruction contains four operands which specify the tape unit, the writing address, the end-of-file transfer address and the restart procedure transfer address. The Autocoder generates an inline sequence of six 705 instructions and causes the inclusion of the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine TPERR. Here a further analysis is made. If a redundant character has been detected in the output area of memory the message "901 R 02xx" is typed and a transfer is made to the restart address. If a restart address is not specified the 705 will come to Stop 0901. Pressing the console start key will cause successive re-writes.

If a writing error has occurred the message "902 R 02xx" is typed and three re-writes are attempted. If re-writing is successful a transfer is made back to the main program. The tape unit remains selected, If it is impossible to successfully write the record and a restart address has been specified, a transfer is made to it. If none is specified the 705 will come to Stop 0902. Pressing the console start key will cause successive re-writes.

If an end-of-file condition has been detected by sensing the reflective spot on the end of the tape, the input-output indicator will be turned off and a transfer made to the address specified. If none is specified the message "EOF R 02xx" is typed and the 705 will come to Stop 0566. Again, pressing the console start key will cause successive re-writes.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	WR		X2	
	DOA		TPERR	
H n	LOD	14	H n	
	TRA		TPERR	
	INCL		TPERR	
	RACON		X3	(end-of-file transfer address)
	RACON		X4	(restart procedure address)
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches-0902 on Program.
 - (b) Auxiliary Storage Units -The ASU sign, comparison and zero triggers may be changed, the sign trigger being set to plus.
 - (c) Memory -30 characters each time the macro-instruction is written plus 532 characters for the subroutines TPERR and XOFF and various literal constants which are included only once.

Macro-Instruction: Write Tape Mark (using 754 CU) Operation Code: WRTM

Function: To write a tape mark on a designated tape; checking to see that it was written correctly.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	WRTM	X1		

where: X1 is the last 2 digits of the tape unit number, right justified. Zeros may be omitted.

Description: The macro-instruction contains one operand which specifies the tape unit. The Autocoder generates an in-line sequence of six 705 instructions and causes the inclusion of the tape error subroutine TPERR and the typewriter subroutine XOFF at the end of the program.

If, during the execution of this sequence of instructions in the user's program, the ANY indicator is found on, it is turned off and transfer is made to the subroutine TPERR. Here a further analysis is made. If the tape mark has been written incorrectly the message "902 3 02xx" is typed and three attempts are made to re-write it. If re-writing is successful a transfer is made back to the main program. The tape unit remains selected. If it is impossible to write the tape mark correctly the 705 will come to Stop 0902. Pressing the console start key will cause successive re-writes.

If the physical end of the tape has been sensed the input-output indicator is turned off and a transfer is made back to the main program.

An error in typing messages will not be corrected. The check indicators 0901 and 0902 and the ANY indicator will simply be turned off by subroutine XOFF.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		02X1	
	WTM			
	DOA		TPERR	
Hn	LOD	14	H n	
	TRA		TPERR	
	INCL		TPERR	
	RACON		T1 +30	
	RACON			
	INCL		XOFF	

- Requirements:
- (a) Check Indicator Switches - 0902 on Program.
 - (b) Auxiliary Storage Units - The setting of the ASU sign, comparison and zero triggers may be changed. The sign trigger will be set to plus.
 - (c) Memory - 30 characters each time the macro-instruction is written plus 532 characters for the subroutines TPERR and XOFF and various literal constants which are included only once.

SECTION II -

LOGICAL MACRO - INSTRUCTIONS

The macro-instructions in this section are of two general types: programming aids; and extensions of machine logic. They include the following topics:

Address Modification

Comparison, Sign and Zero Trigger Storage

Conditional Stops

Conditional Transfers

Data Transmission

Loop Control

Switch Setting

Total and Sequence Checking

Macro-Instruction: Alternate Transfer

Operation Code: ALTTR

Function: Alternate transfer and non-transfer.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	ALTTR		xxxx	

where: T1 is any tag, or may be left blank.

xxxx may be any descriptive or actual address of an instruction.

Description: Is a Transfer to designated address first time, NOP the second, TR the third, etc.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TR		T1 + 15	
	SGN	15	T1 - 4	
	TR		T1 + 30	
	SGN	15	T1 - 4	
	ADM	15	T1 - 4	
	TR		xxxx	

Restrictions:

- (a) Turns on the ASU zero trigger.

Macro-Instruction: Check Total

Operation Code: CHKT

Function: To take a check total of a specified field.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	CHKT	nn	xxxx ▯ TAG ▯	

where: T1 is any tag, or may be left blank.
nn is the length of the field to be checked.
xxxx is the descriptive or actual address of the field to be checked by check totaling.
TAG is the tag to be applied to the counter containing the check total.

Description: Loads the designated field in the accumulator and removes the zoning, then adds this quantity to a counter.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET		00nn	
	LOD		xxxx	
	SUB		(+0)	
	TR		▯ TAG	
TAG	DCON			
		05	00000	
		nn	+00000.....(51 zeros)	
▯ TAG	ADM		TAG	

Requirements:

- (a) 30 positions plus the length of the field to be checked.
- (b) Uses the literal (+0).
- (c) Uses the accumulator.
- (d) Maximum size of field to be checked 99 characters.
- (e) Maximum safe number of records to total 99999.

Macro-Instruction: Do Subroutine

Operation Code: DO { P
Z

Function: Creates linkage to subroutine which transfer when designated transfer conditions are met.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	DOxxx	nn	SUBRT	

where: nn is designated ASU
xxx is designated transfer condition.
SUBRT is name of subroutine
T1 is any tag, or may be left blank.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRxxx	nn	T1 + 10	
	TR		T1 + 20	
	LOD	14	T1 + 10	
	TR		SUBRT	
	INCL		SUBRT	

Macro-Instruction: End Loop

Operation Code: END

Function: To be used in conjunction with the LOOP macro-instruction to control the number of passages through a loop.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	END	nn	xxxx	

where: T1 is any tag, or may be left blank.

xxxx is the tag of the corresponding LOOP instruction.

nn is the accumulator or ASU designated as loop monitor in the LOOP instruction.

Description: Subtracts + 1 from the accumulator or ASU and transfers to the next instruction if 0; if not to the instruction following the LOOP instruction.

Coding Produced:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	SUB	nn	(+1)	
020		TRZ	nn	T1 + 15	
030		TR		xxxx +5	

Requirements and Restrictions:

- (a) Turns on the zero trigger of the ASU or accumulator upon leaving the loop.

Macro-Instruction: First Time NOP

Operation Code: FTNOP

Function: NOP first time, transfers afterward.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FTNOP		xxxx	

where: T1 is any tag, or may be left blank.

xxxx is any descriptive or actual address.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	NOP		xxxx	
	SGN	15	T1 - 4	

Macro-Instruction: First Time Transfer

Operation Code: FTTR

Function: Transfers first time, afterward NOP.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FTTR		xxxx	

where: T1 is any tag, or may be left blank.

xxxx is any descriptive or actual address.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	NOP		T1 + 15	
	SGN	15	T1 - 4	
	TR		xxxx	

Macro-Instruction: Halt Off

Operation Code: HLTOF

Function: Halts when alteration switch is off.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	HLTOF	nn		

where: T1 is any tag, or may be left blank.

nn designated alteration switch.

Description: Halts when designated the alteration switch if OFF. Pushing

Start causes a continuation of the program.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		09nn	
	TRS		T1 + 15	
	HLT		09nn	

Macro-Instruction: Halt On

Operation Code: HLT ON

Function: Halts with alteration switch on.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	HLTON	nn		

where: T1 is any tag, or may be left blank.

nn is designated alteration switch.

Description: Halts if designated alteration switch is on. Pushing Start causes a continuation of the program.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEL		09nn	
	TRS		T1 + 15	
	TR		T1 + 20	
	HLT		09nn	

Macro-Instruction: Halt Transfer

Operation Code: HLTTR

Function: Halts and transfers.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	HLTTR		xxxx ▯ nnnn ▯	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address of an instruction.
nnnn is the number of the Halt.

Description: Halts, pressing start causes transfer to designated address.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	HLT		nnnn	
	TR		xxxx	

Macro-Instruction: If xxx

Operation Code: IF

E
LOW
H
EH
EL
NE
Z
NZ

Function: Transfers on high, equal, etc.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	IFxxx	nn	FIRST ITEM SECOND ITEM ADDRESS	

where: T1 is any tag, or may be left blank.
nn is designated ASU.
xxx is designated transfer condition.

Description: Compares the items in two locations, and transfers to the address if the condition is satisfied, otherwise proceeds to next instruction. The condition may be those built into the machine or the additional condition EH, EL, LOW, NE; the appropriate condition is specified in place of the xxx, i. e. IFH, IFNZ, etc.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	nn	FIRST ITEM	
	LOD	nn	FIRST ITEM	
	CMP	nn	SECOND ITEM	
	TRxxx	nn	ADDRESS	

NOTE: The first six possible transfer conditions demand the instruction CMP in order to function.

The last three possible transfer conditions demand only the LOD instruction to function. If the first item consists of zeros, it will cause a transfer zero to ADDRESS assuming Z is the designated transfer condition.

Macro-Instruction: Load Left Location Operation Code: LLL

Function: The high-order location of the designated field is loaded in the designated ASU (or accumulator).

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LLL	nn	FIELD Q	

where: T1 is any tag, or may be left blank.
FIELD Q is any descriptive or actual address.

Description: The designated ASU is set to 4 and the high-order location of the designated field is loaded in that ASU.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	nn	0004	
	LOD	nn	T1 +10	
	LACON		FIELD Q	

Macro-Instruction: Load Left Location 14 Operation Code: LLL14

Function: Loads in ASU 14 the high-order location of the designated field,
zoned for the designated ASU.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LLL14	nn	FIELD Q	

where: T1 is any tag.
FIELD Q is any descriptive or actual location.
nn is the designation of any ASU or accumulator.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	14	T1 +5	
	LACON	nn	FIELD Q	

Macro-Instruction: Load Right Location

Operation Code: LRL

Function: The low-order location of the designated field is loaded in the designated ASU (or accumulator).

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LRL	nn	FIELD Q	

where: T1 is any tag, or may be left blank.
FIELD Q is a descriptive or actual address.

Description: The designated ASU is set to 4 and the low-order location of the designated field is loaded in that ASU.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	nn	0004	
	LOD	nn	T1 +10	
	RACON		FIELD Q	

Macro-Instruction: Load Right Location 14

Operation Code: LRL14

Function: Loads in ASU 14 the low-order location of the designated field,
zoned for the designated ASU.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LRL14	nn	FIELD Q	

where: T1 is any tag, or may be left blank.

FIELD Q is any descriptive or actual location.

nn is the designation of any ASU or accumulator.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	14	T1 +5	
	RACON	nn	FIELD Q	

Macro-Instruction: Loop

Operation Code: LOOP

Function: All of the following instructions until a completing end instruction, will be repeated M times, where the location of M is indicated in the operand.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
TAG	LOOP	nn	M	

where: Tag is any tag.
nn is any accumulator or ASU.
M is a literal, descriptive or actual location of the number of times this loop is to be passed through.

Description: Reset adds M in ASU nn. To be used with END macro-instruction.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
TAG	RAD	nn	M	

Requirements and Restrictions:

- (a) The loop instructions must be tagged.
- (b) The quantity in M must be signed (it may be a literal). If negative the sign trigger will be set negative.

Macro-Instruction: Move

Operation Code: MOVE

Function: Moves information to a specified place from a specified place in memory.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	MOVE	nn	FROM □ TO □	

where: nn is the designation of the accumulator or ASU's.
TO and FROM are descriptive or actual addresses.

Description: Produces a Receive and Transmit instruction. If the numeric column of the macro-instruction designates the accumulator, high-speed transmission will occur. If the numeric column of the macro-instruction designates an ASU, slow-speed transmission will occur. In this case the number of characters transmitted will depend upon the setting of the designated ASU.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RCV	nn	TO	
	TMT	nn	FROM	

Macro-Instruction: Move Characters

Operation Code: MOVEC

Function: Moves the number of characters specified by the digits in the numerical column from a specified address to a specified address.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	MOVEC	nn	FROM \boxtimes TO \boxtimes	

where: nn is any unsigned number from 1 to 99.
T1 is any tag, or may be left blank.
FROM and TO are descriptive or actual addresses.

Description: Sets ASU 15 to the length specified in the numeric column and produces a RCV and TMT via ASU 15.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	15	00nn	
	RCV	15	TO	
	TMT	15	FROM	

Requirements and Restrictions:

- (a) If nn is a number greater than 31 the contents of ASU 1 and other higher ASU's may be effected.
- (b) May change the setting of ASU 15, which in turn may effect the zero trigger of the ASU's.

Macro-Instruction: Move Instruction Address

Operation Code: MOVEI

Function: Moves the four character portion of an instruction.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	MOVEI		FROM \square TO \square	
020					

where: T1 is any tag, or may be left blank.
FROM and TO are descriptive address of instructions or address constants.

Description: Produces a Receive and a Transmit instruction via ASU 14 supplying the character adjustment necessary to move the four characters.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RCV	14	TO + 1	
	TMT	14	FROM + 1	

Requirements and Restrictions:

- (a) ASU 14 set to 4.
- (b) The TO and FROM address may be descriptive or literal, but may not be actual. If literal, must be 5 characters in length and the first character will be ignored, the last four only are moved.

Macro-Instruction: Order Check

Operation Code: ORDCH

Function: To check the order of items in a specific file and branch to different paths depending upon the high, equal or low (out of sequence) condition encountered.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	ORDCH	nn	xxxx H HIGH H EQUAL H ERROR H	

where: T1 is any tag, or may be left blank.

nn is the length of the field.

HIGH is the actual or descriptive address of the next instruction to be executed if items are strictly in sequence.

EQUAL is the actual or descriptive address of the next instruction to be executed if the items are equal.

ERROR is the actual or descriptive address of the next instruction to be executed if the items are out of sequence.

xxxx is the actual or descriptive location of the field being checked.

Description: Sets aside a working storage area and compares the specified field with this quantity. Unloads the new quantity in the working storage and transfers to the appropriate address.

Note: See also SEQCH which checks for strictly in sequence condition and Halts on error.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	15	00nn	
	LOD	15	xxxx	
	CMP	15	H TAG	
	UNL	15	H TAG	
	TRH		HIGH	
	TRE		EQUAL	
	TR		ERROR	
	DCON			
H TAG		nn		

Requirements and
Restrictions:

- (a) Require 35 positions plus the number of positions in the field to be checked if a multiple of S; if not, to the next larger multiple of 5. (i. e.) If the field is 5 a total of 45 positions; if the field is 15 a total of 50.
- (b) If the field is of greater length 31, ASU 1 and higher ASU may be affected.
- (c) Comparison is used and the setting will normally be left High.
- (d) Under certain situations this macro may give a false stop on the first item. This possibility will be eliminated if at least one of the following conditions is met:
 - (1) Clear memory button is pushed before loading program.
 - (2) The first field to be checked will contain at least 5 characters that are not blank.
 - (3) The field to be checked will be a multiple of 5 in length.

Macro-Instruction: Repeat, ASU Controlled

Operation Code: RPTA

Function: To traverse a loop a designated number of times, using an ASU as loop monitor.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RPTA	nn	xxxx. □ nnnn □	

where: T1 is any tag, or may be left blank.
 xxxx is the location of the first instructions of the loop.
 nnnn is the number of times the loop is to be repeated before exiting (maximum 9999).

Description: The macro-instruction contains two operands, first in the address of the first instruction in the loop and second is the number of times the loop is to be passed through before exiting.

The macro-instruction RPTA follows the loop it controls. After the specified number of times through the loop, the instruction immediately following the RPTA main instruction is executed.

The macro-instruction will not function properly if in traversing the loop the ASU sign trigger is set minus. If the loop instructions can create this condition, use macro-instruction RPTM.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	NOP		T1 +25	
	SGN	15	T1 -4	
	RAD	nn	T1 +20	
	TR		T1 +25	
	NOP		nnnn	
	SUB	nn	(+1)	
	TRZ	nn	T1 + 40	
	TR		xxxx	
	SGN	15	T1 -4	
	ADM	15	T1 -4	

Modification: To reset the RPTA instruction use the instruction

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	SWNOP		SWITCH	

where: SWITCH is the tag of the RPTA instruction.
The number of passages through the loop may be altered by adding or subtracting from the designated ASU.

- Restrictions:
- (a) Turns on the zero trigger of the ASU's.
 - (b) Uses an ASU for a monitor.
 - (c) Requires 51 character positions.

Macro-Instruction: Repeat, Memory Controlled

Operation Code: RPTM

Function: Provide ability to traverse a loop a designated number of times, using a memory counter as loop monitor.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RPTM		xxxx ▯ nnnn▯	

where: T1 is any tag, or may be left blank.
 xxxx is the descriptive or actual address of the first instruction of the loop.
 nnnn is the number of times the loop is to be traversed.

Description: The macro-instruction RPTM contains two operands, first is the address of the first instruction in the loop and the second is the number of times the loop is to be passed through before exiting.

The macro-instruction RPTM follows the loop it controls. After the specified number of times through the loop, the instruction immediately following RPTM is executed.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	NOP		T1 +30	
	SGN	15	T1 -4	
	RAD	15	T1 +20	
	TR		T1 +35	
	AACON	+	nnnn	Coding Produced
	RACON			
	RAD	15	T1 +25	
	SUB	15	(+1)	
	TRZ	15	T1 +55	
	ST	15	T1 +25	
	TR		xxxx	
	SGN	15	T1 -4	
	ADM	15	T1 -4	

Modification: To reset the loop issue the instruction

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	SWNOP		T1	

where: T1 is the tag of the RPTM instruction.

- Requirements and Restrictions:
- (a) Turns on the zero trigger of the ASU's.
 - (b) Maximum of 9999 times through the loop.
 - (c) Uses 61 character positions.

Macro-Instruction: Restore Comparison

Operation Code: RCMP

Function: To be used after SCMP macro-instruction (which see) to restore the condition of the comparison triggers.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RCMP	nn		

where: T1 is any tag or may be blank.
nn are any two 705 characters or may be blank.

Description: Reset Adds +2 in ASU 15, and compares to the literal (-nn) where nn is specified in the numeric column.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RAD	15	(+2)	
	CMP	15	(-nn)-1	

Requirements and Restrictions:

- (a) The programmer should specify SCMP with the same entry in the numeric column somewhere in his program.
- (b) SCMP should be executed before the corresponding RCMP.

Macro-Instruction: Restore Sign of ASU

Operation Code: RSGN

Function: To be used after SSGN macro-instruction (which see) to restore the condition of the sign and zero triggers.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	RSGN	nn		

Description: Reset Adds in ASU 15, the contents of the literal (+ nn) where nn is specified in the numeric column.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RAD	15	(+ nn)	

Requirements and Restrictions:

- (a) The programmer should specify SSGN with the same entry in the numeric column somewhere in his program.
- (b) SSGN should be executed before the corresponding RSGN.
- (c) Contents of ASU 15 are destroyed.

Macro-Instruction: Save Comparison

Operation Code: SCMP

Function: Saves the indication of the comparison triggers in the literal table in memory. Designed to be used with the RCMP (Restore Comparison) macro-instruction.

Any two 705 characters identifying the comparison (including blanks) may be placed in the numerical column. The RCMP macro-instruction will restore the comparison trigger to the condition existing prior to the SCMP with the same entry in the numerical column.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SCMP	nn		

where: T1 is any tag or may be left blank.
nn are any two 705 characters including blanks.

Description: Creates a literal of the form (-nn) when nn is the entry in the numeric field. A 2 is stored there if the equal trigger is on, a 1 if the high trigger is on and a 3 if neither is on.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRE		T1 + 30	
	TRH		T1 + 20	
	RAD	15	(+3)	
	TR		T1 + 35	
	RAD	15	(+1)	
	TR		T1 + 35	
	RAD	15	(+2)	
	UNL	15	(-nn)-1	

Requirements and Restrictions:

- (a) Creates a literal of the form (-nn) which is used as working storage altering its contents. This might conflict with other literals in the program.
- (b) Uses literals (+1), (+2), and (+3) which should not be modified by the main program.

Macro-Instruction: Save Sign of ASU

Operation Code: SSGN

Function: Saves the indication of the sign and zero triggers of the ASU's in memory. Designed to be used with the RSGN (Restore Sign) macro-instruction.

Any two 705 characters (including blanks) may be placed in the numeric column. The RSGN macro-instruction will restore the triggers to the condition existing prior to the SSGN instruction with the same entry in the numeric column.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SSGN	nn		

where: T1 is any tag, or may be left blank.
nn are any two 705 characters including blanks.

Description: Creates a literal of the form (+ nn) where nn is the entry in the numeric field. + 0 is stored in this literal if the zero trigger were on, and + 1 is stored there if the plus trigger were on, and - 1 is stored if neither were on.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRZ	15	T1 + 30	
	TRP	15	T1 + 20	
	RSU	15	(+ 1)	
	TR		T1 + 35	
	RAD	15	(+ 1)	
	TR		T1 + 35	
	RAD	15	(+ 0)	
	ST	15	(+ nn)	

Requirements and Restrictions: (a) Creates a literal of the form (+ nn) which is used as working storage altering its contents. This might conflict with other literals in the program.

- (b) Does not alter the condition of the sign and zero triggers except that zero is considered to be plus zero.
- (c) Zeros are assumed to be positive.
- (d) Uses literals (+1) and (+0) which should not be modified by the main program.

Macro-Instruction: Sequence Check

Operation Code: SEQCH

Function: Sequence check a specified field in a file and halt if the file is not in ascending sequence. Will not accept equals.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SEQCH	nn	xxxx \square mmmm \square	

where: nn is the length of the field to be checked.
xxxx is the actual or descriptive address of the field.
mmmm is the number of the Halt to be used if out of sequence occurs.

Description: Sets aside a working storage area and compares the specified field with this quantity. If high, unloads the new quantity, if equal or low, halts. Pushing start will cause continuation in spite of the out of sequence conditions.

Normally used immediately following a read instruction of a supposedly well ordered file, or immediately prior to writing a sorted file.

Note: See also macro ORDCH (order check), which provides different paths for High, Equal and Low conditions.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SET	15	00nn	
	LOD	15	xxxx	
	CMP	15	\square TAG -5	
	TRH		\square TAG +5	
	TR		\square TAG	
	DCON			
		nn		
\square TAG	HLT		mmmm	
	UNL		TAG -5	

- Requirements:
- (a) Requires 35 positions plus the number of digits in the field to be checked. May waste up to 4 additional positions if followed by an instruction.
 - (b) If the field is of greater length than 31, ASU 01 and higher ASU's may be affected.
 - (c) Comparison is used and the setting will vary.

Macro-Instruction: Set Switch to NOP

Operation Code: SWNOP

Function: Sets designated switch to NOP.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SWNOP		SWITCH	

where: T1 is any tag, or may be left blank.

SWITCH is the descriptive address of some switch.

Description: Signs in ASU 15 and ADM 15 the operation part of the designated instruction. Will set to NOP regardless of the previous setting.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SGN	15	SWITCH -4	
	ADM	15	SWITCH -4	

Requirements and Restrictions:

- (a) Turns the zero trigger of ASU's on.
- (b) The switch must be designated as a descriptive, may not be an actual address.

Macro-Instruction: Set Switch to TR

Operation Code: SWTR

Function: Sets designated switch to TR.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SWTR		SWITCH	

where: T1 is a tag.

SWITCH is the descriptive location of some switch.

Description: Signs in ASU 15 the operation part of the designated switch.

Will set switch to TR regardless of previous setting.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	SGN	15	SWITCH -4	

Requirements and Restrictions:

- (a) Turns the zero trigger of the ASU's on.
- (b) The switch must be designated as a descriptive, may not be an actual address.

Macro-Instruction: Set Up

Operation Code: SETUP

Function: Establishes certain recommended ASU settings.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	SETUP			

Description: Sets ASU 1 to length 1, ASU 2 to length 2, ASU 3 to length 3, ASU 4 to length 4, ASU 5 to length 5, ASU 13 to length 10, ASU 14 to length 4. These are the initial settings made by the loading program and 00 card produced by the Autocoder. Some of the routines and macro-instructions depend upon these settings.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	SET	1	1	
	SET	2	2	
	SET	3	3	
	SET	4	4	
	SET	5	5	
	SET	14	4	
	SET	13	10	

Restriction: (a) May turn on the ASU zero trigger.

Macro-Instruction: Transfer Equal or High

Operation Code: TREH

Function: Transfers equal or high.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TREH		xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address.

Description: Transfers to designated address if high or equal triggers are on.

Coding Produced:

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRH		xxxx	
	TRE		xxxx	

Macro-Instruction: Transfer Equal or Low

Operation Code: TREL

Function: Transfers on equal or low.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TREL		xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRH		T1 +15	
	TRE		xxxx	
	TR		xxxx	

Macro-Instructions: Transfer Low

Operation Code: TRLOW

Function: Transfers low.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRLOW		xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address of an instruction.

Description: Transfers to the designated address if neither the high nor equal triggers are on.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRH		T1 + 15	
	TRE		T1 + 15	
	TR		xxxx	

Macro-Instruction: Transfer Minus

Operation Code: TRMIN

Function: Transfers minus.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRMIN	nn	xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address of an instruction.

Description: Transfers to the designation if zero trigger is not on.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRP	nn	T1 + 10	
	TR		xxxx	

Macro-Instruction: Transfer Non-Zero

Operation Code: TRNZ

Function: Transfers non-zero.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRNZ	nn	xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address of an instruction.

Description: Transfers to the designated address if plus trigger is not on.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRZ	nn	T1 + 10	
	TR		xxxx	

Macro-Instruction: Transfer Not Equal

Operation Code: TRNE

Function: Transfers not equal.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRNE	nn	xxxx	

where: T1 is any tag, or may be left blank.
xxxx may be any descriptive or actual address of an instruction.
nn is designated ASU.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	TRE	nn	T1 + 10	
	TR		xxxx	

SECTION III

FLOATING DECIMAL ARITHMETIC MACRO-INSTRUCTIONS

A library of subroutines and macro-instructions for floating point operations, is available with the 705 Autocoder System. A mnemonic code, beginning with F requests each such operation.

LIST OF FLOATING POINT OPERATIONS

<u>MNEMONIC</u>	<u>OPERATION</u>	<u>TYPE</u>
FRA	Reset Add	M
FRS	Reset Subtract	M
FAD	Add	MS
FSU	Subtract	MS
FMP	Multiply	MS
FDV	Divide	MS
FAB	Absolute	M
FST	Store	M
FTP	Transfer Plus	M
FTZ	Transfer Zero	M
FTM	Transfer Minus	M
FLO	Float	MS
FIX	Fix	MS
DO FLN	Log	DMS
DO FEX	Exponential	DMS
DO FSQR	Square Root	DMS
DO FSIN	Sine	DMS
DO FATN	Arctan	DMS

Explanation of Types

- M - Macro-instruction.
 - MS - Macro-instruction linkage and linked subroutine, which may require a specified operand.
 - DMS - Macro-instruction linkage and linked subroutine, not requiring a specified operand.
- (See DO Subroutine, page 129)

Each floating point number consists of two parts: a mantissa, a number between 0 and 1; and an exponent, which is the power of 10 that restores the mantissa to its true form (fixed-point) by multiplication.

Examples:

Fixed-Point		Floating Point
-12.345678	=	-.12345678 x 10 ²
.12345678	=	.12345678 x 10 ⁰
.012345678	=	.12345678 x 10 ⁻¹

The floating point word-length for these operations described herein is 10 characters: 8 for the mantissa and 2 for the exponents. Both the mantissa and exponent are signed. In the 705 memory the above numbers would appear as:

$0\overset{+}{2}$ 1234567 $\bar{8}$
 00 1234567 $\overset{+}{8}$
 $0\bar{1}$ 1234567 $\overset{+}{8}$

The range of the mantissa M is

$$0 \leq M < 1.$$

The range of the exponent P is

$$-99 \leq P \leq 99.$$

Floating point operations which could possibly produce numbers whose exponents exceed this range have programmed halts.

A floating point zero is represented as

$9\bar{9}$ 0000000 $\overset{+}{0}$

A 10-character area in memory, which is designated by the signed literal operand

(+FLOATACCUM)

is used as a pseudo-accumulator for floating point operations, serving the same function as the accumulator for 705 instructions. Any Autocoder instruction, other than a floating point operation, will not disturb the contents of the pseudo-accumulator unless the instruction specifies the signed literal operand

(+FLOATACCUM)

Use of the floating point operations FLOAT and FIX is restricted to fixed point numbers of the form $\overset{+}{xxxxx.xxxxx}$

a 10-character field. The decimal point should not be written.

PROGRAMMED HALTS IN FLOATING POINT OPERATIONS.

HALT ADDRESS	OP	CAUSE	RESULTS ON PRESSING START KEY
8000	FAD/FSU	Underflow in exponent	Exponent reduced
8001	FAD/FSU	Overflow in exponent	Modulo 100
8002	FMP	Underflow in exponent	
8003	FMP	Overflow in exponent	
8004	FDV	Underflow in exponent	
8005	FDV	Overflow in exponent	
8006	FDV	Division by zero	Contents of Floating Decimal Accumulator is set 9999999999
8007	FIX	Number too large	9999999999 is stored at designated address
8008	FSQR	Square root of negative argument	Original contents of Floating Decimal Accumulator left intact. Transfer out of subroutine.
8009	FEX	Argument too large	Contents of Floating Decimal are set to 9999999999
8010	FSIN	Argument too large	Contents of Floating Decimal Accumulator are set to 0110000000
8011	FLN	Log of negative argument	Original contents of Floating Decimal Accumulator are left intact and a transfer out of the sub-routine is executed
8012	FLN	Log of zero argument	

Macro-Instruction: Absolute, Floating

Operation Code: FAB

Function: The sign of the mantissa in the Floating Decimal Accumulator is made plus.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FAB			

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator

Description: The sign of the mantissa is stripped off and replaced by a Plus sign.

Coding Produced:

T1	RAD	15	(+FLOATACCUM)	
	UNL	15	(+FLOATACCUM)	
	LOD	15	(+FLOATACCUM)	
	ST	15	(+FLOATACCUM)	

Requirements and Restrictions: (a) Contents of ASU 15 are destroyed.
(b) Memory: 20 characters.

Macro-Instruction: Add, Floating

Operation Code: FAD

Function: The contents of the Floating Decimal Accumulator are added algebraically to the operand, and this sum replaces the contents of the Floating Decimal Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FAD		OPERAND	

Description: The difference of the exponents is calculated and tested. If this difference is greater than seven the larger argument is placed in the Floating Decimal Accumulator. Otherwise the difference of the exponents (in absolute value) is used by a shift instruction to line up the decimal points of the two arguments for addition. After addition of the two arguments, the exponent of the sum is determined by the exponent of the larger argument. In the case of overflow in the addition of the mantissas, a one is added to the exponent of the sum.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	13	OPERAND	
H n	LOD	14	H n	
	TR		FAD	+00020
	INCL		FAD	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) ASU 14 must be set to 4 characters.
- (c) Contents of ASU's 13, 14, 15 are destroyed.
- (d) The sign of the ASU's is set to plus.
- (e) Contents of the accumulator are destroyed.
- (f) Programmed stops:

- 1. HLT 8000 - Underflow in exponent
- 2. HLT 8001 - Overflow in exponent

Upon pressing the start key the exponent is reduced mod 100 (by a SET 0002 instruction).

- (g) Number of instructions: 57.
- (h) Amount of space required: 327.
- (i) Execution time: 5.1 milliseconds.

Macro-Instruction: Divide, Floating

Operation Code: FDV

Function: The contents of the Floating Decimal Accumulator are divided by the operand, and this quotient replaces the contents of the Floating Decimal Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FDV		OPERAND	

Description: The divisor is tested for zero, in which case there is a programmed halt. Then the dividend is tested for zero to by-pass calculation of a zero quotient. If neither argument is zero, a division is executed and the leading digit of the quotient is tested for zero. In the case of a zero leading digit, the exponent of the quotient is the difference of the exponents of the arguments. In the case of a non-zero leading digit, one is added to the exponent of the quotient.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	13	OPERAND	
A _n	LOD	14	A _n	
	TR		FDV	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) ASU 14 must be set to 4 characters.
- (c) Contents of ASU's 13, 14, 15 are destroyed.
- (d) The sign of the ASU's is set to plus.
- (e) Contents of the accumulator are destroyed.
- (f) Programmed stops:
 1. HLT 8004 - Underflow in exponent
 2. HLT 8005 - Overflow in exponentUpon pressing the start key the exponent is reduced mod 100 (by a SET 0002 instruction).
 3. HLT 8006 - Division by zeroUpon pressing the start key the contents of the Floating Decimal Accumulator are set to a mantissa of 99999999 and an exponent of 99.
- (g) Number of instructions: 35.
- (h) Amount of space required: 211.
- (i) Execution time: 15.7 milliseconds.

Macro-Instruction: Fix, Floating

Operation Code: FIX

Function: The number in the Floating Decimal Accumulator is converted to a fixed decimal word

xxxxx.xxxxx⁺

and stored at the designated address.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T ₁	FIX		OPERAND	

Description: The exponent of the contents of the Floating Decimal Accumulator is compared (in absolute value) to five which is the maximum exponent possible. If the exponent is less than or equal to five, it used for a shift instruction. If the exponent is less than -5, the answer is set to zero.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T ₁	NOP		OPERAND	
	RCV	14	T ₁ +00026	
	TMT	14	T ₁ +00001	
A _n	LOD	14	A _n	
	TR		FIX	
	ST		OPERAND	

Requirements and Restrictions:

- (a) ASU 14 must be set to 4 characters.
- (b) Contents of ASU's 14, 15 are destroyed.
- (c) Contents of the accumulator are destroyed.
- (d) Programmed stops:
 - 1. HLT 8007 - Argument out of range
Upon pressing start key the number 99999.99999 is stored at the designated address.
- (e) Number of instructions: 20.
- (f) Amount of space required: 128.
- (g) Execution time: 1.36 milliseconds.

Macro-Instruction: Float

Operation Code: FLO

Function: The fixed decimal word

xxxxx. xxxxx⁺

at the designated address is converted to a floating decimal word and stored in the Floating Decimal Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FLO		OPERAND	

Description: An initial maximum exponent of 5 is assumed. By normalizing and shifting, leading zeros are stripped from the word. Each zero thus dropped accounts for a one subtracted from the working exponent.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	RAD		OPERAND	
n	LOD	14	n	
	TR		FLO	
	INCL		FLO	

Requirements and Restrictions:

- (a) ASU 14 must be set to 4 characters.
- (b) Contents of ASU's 14, 15 are destroyed.
- (c) Contents of the accumulator are destroyed.
- (d) Programmed stops: None.
- (e) Number of instructions: 13.
- (f) Amount of space required: 82.
- (g) Execution time: 2.48 milliseconds.

Macro-instruction: Multiply, Floating

Operation Code: FMP

Function: The contents of the Floating Decimal Accumulator are multiplied by the operand, and this product replaces the contents of the Floating Decimal Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FMP		OPERAND	

Description: Each number is tested for zero, in which case calculation is by-passed. For a non-zero product the leading digit is tested for zero. In the case of the non-zero leading digit the exponent of the answer is the sum of the exponents of the arguments. In the case of the zero leading digit, one is subtracted from the exponent of the product.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	13	OPERAND	
Mn	LOD	14	Mn	
	TR		FMP	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) ASU 14 must be set to 4 characters.
- (c) Contents of ASU's 13, 14, 15 are destroyed.
- (d) The sign of the ASU's is set to plus.
- (e) Contents of the accumulator are destroyed.
- (f) Programmed stops:
 - 1. HLT 8002 - Underflow in exponent
 - 2. HLT 8003 - Overflow in exponentUpon pressing the start key the exponent is reduced mod 100 (by a SET 0002 instruction).
- (g) Number of instructions: 29.
- (h) Amount of space required: 173.
- (i) Execution time: 3.9 milliseconds.

Macro-Instruction: Reset Add, Floating

Operation Code: FRA

Function: The contents of the Floating Decimal Accumulator are replaced by the operand.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FRA		OPERAND	

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The operand is loaded into ASU 13 which is then unloaded into a 10-character area of memory referred to as the Floating Decimal Accumulator.

Coding Produced:

T1	LOD	13	OPERAND	
	UNL	13	(+ FLOAT ACCUM)	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) Contents of ASU 13 are destroyed.
- (c) Address modification of the macro-instruction should refer to T1.
- (d) Memory: 10 characters.

Macro-Instruction: Reset Subtract, Floating

Operation Code: FRS

Function: The contents of the Floating Decimal Accumulator are replaced by the negative of the operand.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FRS		OPERAND	

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The operand is loaded into ASU 13 and ASU 13 is then unloaded into a 10-character area of memory referred to as the Floating Decimal Accumulator. The negative of the mantissa is placed in Accumulator and then this negative mantissa is stored in the Floating Decimal Accumulator.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	13	OPERAND	
	UNL	13	(+FLOATACCUM)	
	RSU		(+FLOATACCUM)	
	ST		(+FLOATACCUM)	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) The contents of ASU 13 are destroyed.
- (c) Address modification of the macro-instruction should refer to T1.
- (d) Memory: 20 characters.

Macro-Instruction: Store, Floating

Operation Code: FST

Function: The operand is replaced by the contents of the Floating Decimal Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FST		OPERAND	

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The contents of the Floating Decimal Accumulator is transmitted (character-by-character) to the location of the operand.

Coding Produced:

T1	RCV	13	OPERAND	
	TMT	13	(+FLOATACCUM)	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) Contents of ASU 13 are destroyed.
- (c) Address modification of the macro-instruction should refer to T1
- (d) Memory: 10 characters.

Macro-Instruction: Subtract, Floating

Operation Code: FSU

Function: The operand is subtracted from the contents of the Floating Decimal Accumulator, and this difference replaces the contents of the Floating Accumulator.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FSU		OPERAND	

Description: The sign of the operand is changed. Then the sub-routine Add, Floating is used.

Coding Produced:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	LOD	13	OPERAND	
\bar{A}_n	LOD	14	\bar{A}_n	
	TR		FAD	
	INCL		FAD	

Requirements and Restrictions:

- (a) ASU 13 must be set to 10 characters.
- (b) ASU 14 must be set to 4 characters.
- (c) Contents of ASU's 13, 14, 15 are destroyed.
- (d) The sign of the ASU's is set to plus.
- (e) Contents of the accumulator are destroyed.
- (f) Programmed stops:
 - 1. HLT 8000 - Underflow in exponent.
 - 2. HLT 8001 - Overflow in exponent.Upon pressing the start key the exponent is reduced mod 100 (by a SET 0002 instruction).
- (g) Number of instructions: 57
- (h) Amount of space required: 327
- (i) Execution time: 5.1 milliseconds

Macro-Instruction: Transfer Minus, Floating

Operation Code: FTM

Function: Control is transferred to the instruction at the designated address if the mantissa in the Floating Decimal Accumulator is minus.

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	FTM		OPERAND	

where: T1 is any tag or may be blank.
 (+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The mantissa is placed in the Accumulator and then the sign of the Accumulator is tested for plus. If the sign is not plus then a transfer is executed to the designated address.

Coding Produced:

T1	RAD		(+FLOATACCUM)	
	TRP		T1 + 15	
	TR		OPERAND	

Requirements and Restrictions:

- (a) The contents of the Accumulator are destroyed.
- (b) Address modification of the macro-instruction should refer to T1+ 10.
- (c) Memory: 15 characters.

Macro-Instruction: Transfer Plus, Floating

Operation Code: FTP

Function: Control is transferred to the instruction at the designated address if the sign of the mantissa in the Floating Decimal Accumulator is Plus.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FTP		OPERAND	

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The mantissa is placed in the Accumulator and then the sign of the Accumulator is tested for plus.

Coding Produced:

T1	RAD		(+FLOATACCUM)	
	TRP		OPERAND	

Requirements and Restrictions:

- (a) The contents of the Accumulator are destroyed.
- (b) Address modification of the macro-instruction should refer to T1 + 5.
- (c) Memory: 10 characters.

Macro-Instruction: Transfer Zero, Floating

Operation Code: FTZ

Function: Control is transferred to the instruction at the designated address if the mantissa in the Floating Decimal Accumulator is zero.

Instruction Format:

TAG	OPERATION	NUM.	OPERAND	COMMENTS
T1	FTZ		OPERAND	

where: T1 is any tag or may be blank.
(+FLOATACCUM) is the literal referring to the Floating Decimal Accumulator.

Description: The mantissa is placed in the Accumulator and then the contents of the Accumulator is tested for zero.

Coding Produced:

T1	RAD		(+FLOATACCUM)	
	TRZ		OPERAND	

Requirements and Restrictions:

- (a) The contents of the Accumulator are destroyed.
- (b) Address modification of the macro-instruction should refer to T1 + 5.
- (c) Memory: 10 characters.

Macro-Instruction: DO FATN
 FEX
 FLN
 FSIN
 FSQR

Operation Code: DO

Operand: Name of Subroutine

Function: The number in the Floating Decimal Accumulator is replaced by the functional value (i. e. FSQR produces the square root of the number).

Instruction Format:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010	T1	DO		xxxxxx	
020					

where: T1 is any tag, or may be left blank.
 xxxxx is the mnemonic code for the subroutine. (i. e. FSQR)

Description: An INCLude operation code for the specific subroutine and the link to the subroutine are produced.

Coding Produced:

LINE	TAG	OPERATION	NUM.	OPERAND	COMMENTS
010		INCL		FSQR	
020	T1	LOD	14	T1	
030		TR	14	FSQR	

Requirements and Restrictions:

- (a) ASU 14 must be set to 4 positions.
- (b) Contents of ASU 14 are destroyed.
- (c) See Requirements and Restrictions for specific subroutine.

Sub-routine: Arctan, Floating

Label: FATN

Function: The number in the Floating Decimal Accumulator is replaced by its arctan.

Method of Calculation: The argument is converted to fixed-point form. To reduce the range of the argument for the approximating polynomial the following inverse trigonometric relation is used:

$$\arctan x = \arctan x_m + \arctan z$$

where
$$z = \frac{x - x_m}{1 + x_m \cdot x}$$

which has the range $(-.13734, +.13734)$ required for the approximating polynomial.

A table of values of the arctan function over the quadrant $(0, \frac{\pi}{2})$, which is divided into twelve segments, provides the necessary accuracy for the approximating polynomial in the argument z .

A 5-th order approximating polynomial for $\arctan z$ is used:

$$\arctan z = a_1 z + a_3 z^3 + a_5 z^5$$

where
$$\begin{aligned} a_1 &= +0.999999922 \\ a_3 &= -0.3332964 \\ a_5 &= +0.1957 \end{aligned}$$

The final calculation: $\arctan x = \arctan x_m + \arctan z$

requires the value of $\arctan x_m$ extracted from the table of the arctan function and the evaluation of the approximating polynomial.

Requirements and Restrictions:

- (a) Programmed stops: None.
- (b) Number of instructions: 99.
- (c) Amount of space required: 801.
- (d) Execution time: 26.8 milliseconds.
- (e) Accuracy: Maximum absolute error of 5 in 9th decimal place.
- (f) Range of argument: $|x| < 10^9$.

Sub-routine: Exponential, Floating

Label: FEX

Function: The number in the Floating Decimal Accumulator is replaced by its exponential.

Method of Calculation:

Let

$$e^x = 10^{Mx} \text{ where } M = \log_{10} e.$$

Since

$$10^{Mx} = 10^{W+D} = 10^W \cdot 10^D, \quad 0 \leq D < 1$$

then

$$e^x = 10^W \cdot 10^D.$$

To reduce the range of the argument D for a polynomial approximation of range $(-.054, .054)$, the following transformations are employed:

- A. The first relation applied reduces the range of the argument D to a range of $(0, \log_{10} 2)$:

$$\begin{aligned} \text{for } 0 \leq D < \log_{10} 2 \quad \text{we have } 10^D &= 1 \cdot 10^D \\ \log_{10} 2 \leq D < \log_{10} 4 &= 2 \cdot 10^{D-\log_{10} 2} \\ \log_{10} 4 \leq D < \log_{10} 8 &= 4 \cdot 10^{D-\log_{10} 4} \\ \log_{10} 8 \leq D < 1 &= 8 \cdot 10^{D-\log_{10} 8} \end{aligned}$$

- B. To employ a polynomial approximation for $e^{\bar{D}}$ we apply the following relation to the new arg \bar{D}

$$10^{\bar{D}} = e^{\frac{1}{M} \cdot \bar{D}}$$

If we let $Y = \frac{1}{M} \cdot \bar{D}$

then the range of the new argument Y is $0 < Y < .7$

- C. To further reduce the range of the argument a table of the exponential function supplies the value x_0 such that

$$|Y - x_0| < .054$$

Let $Y - x_0 = Z$

then $e^Y = e^{x_0} \cdot e^Z$

Corresponding to x_0 , the table supplies e^{x_0} .

Sub-routine: Exponential, Floating -2-

Label: FEX

To evaluate e^z we apply the polynomial

$$e^z = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4$$

where $a_0 = 1.00000000$

$$a_1 = 0.99999999$$

$$a_2 = 0.50000000$$

$$a_3 = 0.166688$$

$$a_4 = 0.04167$$

The range of z is $(-.054, +.054)$.

Finally we find the product

$$10^W \cdot 2^i \cdot e^{x_0} \cdot e^z$$

which is $10^W \cdot 2^i (e^Y)$

$$= 10^W \cdot 10^D$$

$$= e^x$$

Requirements and Restrictions:

(a) Programmed stops:

1. HLT 8009 - Argument out of range (too large).

Upon pressing start key the contents of the Floating Decimal Accumulator are set to $(.99999999 \times 10^{99})$.

(b) Number of instructions: 101

(c) Amount of space required: 799

(d) Execution time: 20.74 milliseconds

(e) Accuracy: Maximum absolute error of 5 in 9th decimal place.

(f) Range of argument: $|x| < 10^2$.

Sub-routine: Natural Logarithm, Floating

Label: FLN

Function: The number in the Floating Decimal Accumulator is replaced by its natural logarithm.

Method of Calculation: Let $N = M \cdot 10^P$, where $.1 \leq M < 1$ and P is an integer, then
 $\log_{10} N = P + \log_{10} M$
Given a table of multipliers

$$A_1, A_2, \dots, A_9$$

a multiplier is chosen on the basis of the first significant digit of the argument M . The product of M and the multiplier is formed to obtain the next multiplier, on the same basis as the first multiplier. This process continues until the final product

$$M' \equiv M(A_1 A_2 A_3 \dots A_k) = M \prod_{i=1}^k A_i$$

falls in the range $.1 \leq M' < 1.1$.

The calculation of $\log_{10} M$ is based on

$$\log_{10} M = \log_{10} M' - \sum_{i=1}^k \log_{10} A_i$$

To evaluate $\log_{10} M'$ ($.1 \leq M' < 1.1$), the following approximating polynomial is used

$$\log_{10} (1+x) = a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5$$

$$\text{where } a_1 = + 0.4342944616$$

$$a_2 = -0.2171448$$

$$a_3 = + 0.14466$$

$$a_4 = -0.10653$$

$$a_5 = + 0.068$$

which has a maximum absolute error of 68 in the 12th decimal place.

The expression $\sum_{i=1}^k \log_{10} A_i$ is computed while the multipliers are chosen, having stored $\log A_i$ in the table with their corresponding A_i .

Finally $\log_e N$ is obtained by multiplying $\log_{10} N$ by $\log_e 10$.

Sub-routine: Natural Logarithm, Floating -2-

Label: FLN

Requirements and
Restrictions:

(a) Programmed stops:

1. HLT 8011 - Negative Argument
2. HLT 8012 - Zero Argument

Upon pressing start key, the original contents of the Floating Decimal Accumulator are left intact, and a transfer out of the sub-routine is executed.

(b) Number of instructions: 69

(c) Amount of space required: 572.

(d) Execution time: 21.05 milliseconds.

(e) Accuracy: Maximum absolute error of 5 in 9th decimal place.

(f) Range of argument: $0 < x < 10^{99}$.

Sub-routine: Sine (radian), Floating

Label: FSIN

Function: The number in the Floating Decimal Accumulator is replaced by its sine.

Method of Calculation: The argument is converted to fixed-point form and multiplied by $\frac{1}{2\pi}$. The fractional part of this product represents the angle as a fraction of 2π radians. The integral part is discarded. By use of trigonometric identities the angle is reduced to an angle in the first quadrant as follows:

$$\begin{array}{ll} \text{for } 0 \leq x < \frac{\pi}{2} \text{ radians} & \sin x = \sin x \\ \frac{\pi}{2} \leq x < \pi & \sin x = \sin(\pi - x) \\ \pi \leq x < 3\frac{\pi}{2} & \sin x = \sin(x - \pi) \\ 3\frac{\pi}{2} \leq x < 2\pi & \sin x = \sin(2\pi - x) \end{array}$$

The angle, now in the first quadrant, is changed to a fraction of $\frac{\pi}{2}$ radians for use in the approximating polynomials for $\sin \frac{\pi}{2} z$ and $\cos \frac{\pi}{2} z$. The range of the argument, x , which is a fraction of $\frac{\pi}{2}$ radians, is $(-1, +1)$

A table of sine and cosine values for angles which are fractions of $\frac{\pi}{2}$ in the range $(-1, +1)$ produces a new argument

$$z = x - y \quad |z| < .11934$$

where y is the nearest table value to x . The sine and cosine values of y are applied in the trigonometric relation

$$\sin(z + y) = \sin z \cos y + \sin y \cos z$$

The approximating polynomials for $\sin \frac{\pi}{2} z$ and $\cos \frac{\pi}{2} z$, respectively, are:

$$\sin \frac{\pi}{2} z = a_1 z + a_3 z^3 + a_5 z^5$$

$$\begin{array}{ll} \text{where} & a_1 = +1.57079637 \\ & a_3 = -0.64597 \\ & a_5 = +0.08 \end{array}$$

$$\cos \frac{\pi}{2} z = a_0 + a_2 z^2 + a_4 z^4$$

$$\begin{array}{ll} \text{where} & a_0 = 1. \\ & a_2 = -1.233698 \\ & a_4 = +0.2532 \end{array}$$

Requirements and Restrictions: (a) Programmed Stops:

1. HLT 8010 - Argument out of range (too large). Upon pressing start key the contents of the Floating Decimal Accumulator are set to one.

Sub-routine: Sine (radian), Floating -2-

Label: FSIN

- (b) Number of instructions: 122.
- (c) Amount of space required: 946.
- (d) Execution time: 25.5 milliseconds.
- (e) Accuracy: Maximum absolute error of 5 in 9th decimal place.
- (f) Range of argument: $|x| < 10^3$.

Sub-routine: Square Root, Floating

Label: FSQR

Function: The number in the Floating Decimal Accumulator is replaced by its square root.

Method of Calculation: Let $N = M \cdot 10^P$, where $.1 \leq M < 1$
 $|P| \leq 99$

To establish a common program for odd and even powers of 10.
 $N = M' \cdot 10^{P'}$, where $.01 \leq M' < 1$
 P' is even.

then $N = M' \times 10.5^{P'}$

For even powers we have

$$P' = P$$
$$M' = M$$

but for odd powers

$$P' = P + 1$$
$$M' = M \cdot 10^{-1}$$

Two Newton's iterations are required after a first approximation A_1 is calculated. This first approximation is derived by using a table of linear segments which approximate M' within a tolerance of 5 in the 5th decimal position. The range of the argument is (.01, 1).

Using Newton's Iteration Formula we find

$$A_2 = \frac{1}{2} \left(A_1 + \frac{M'}{A_1} \right)$$

$$M' = \frac{1}{2} \left(A_2 + \frac{M'}{A_2} \right).$$

Requirements and Restrictions:

- (a) Programmed stops:
 1. HLT 8008 - Negative ArgumentUpon pressing start key, the original contents of the Floating Decimal Accumulator are left intact, and a transfer out of the sub-routine is executed.
- (b) Number of instructions: 76.
- (c) Amount of space required: 497.
- (d) Execution time: 17.54 milliseconds.
- (e) Accuracy: Maximum absolute error of 5 in 9th decimal place.
- (f) Range of argument: $0 < x < 10^{99}$.

SECTION IV - WRITING NEW MACRO-INSTRUCTIONS

The following section describes the process of preparing new macro-instructions which may be added to the Autocoder. An understanding of this process is not a requirement of program writing, assembly, or utilization of the Autocoder system. Reference to this section need only be made when it becomes necessary to prepare supplementary instructions to meet the requirements of an individual installation.

Form

The form in which the macro-instruction will be written is determined by the need it must fulfill. A macro-instruction may have a single or multiple address. Its numerical field may be used for an ASU designation or a tape unit specification or it may have other significance convenient to its function. The format is variable depending upon the use for which the instruction is designed. Each specification of a macro describes how it is to be written and what data are to be supplied.

A macro-instruction represents a sequence of other Autocoder operations which are grouped together for the convenience of the programmer. Any given programmed function can be also written using a number of conventional 705 instructions and other Autocoder operations. It is usually more economical of programming effort, however, to write a single macro-instruction instead.

Component Instructions

In the following explanation, the term macro-instruction refers to the written program entry. This entry causes the Autocoder assembly to refer to the Library for further information. Instructions taken from the library and included in the output program are referred to as component instructions.

The components in the Library are all Autocoder operations, i. e. 705 instructions, definitions, address constants, and DO operations. The last component of the series may be another macro-instruction. (For this purpose, DO, DOA, DOH, DOS, DOE are not considered macro-instructions. They may be placed in the series, while all other operations must be in the last line, if they are used. The component tag, if any, must be supplied by the Autocoder.)

The component instructions, as they are stored in the library, are only partially completed. The Autocoder assembly uses the information supplied by the macro-instruction entry on the program sheet to fill in the missing parts. For example, the macro-instruction MOVE may be used to transfer information from a designated location in memory to some other location. (Refer to specification of MOVE.)

Several items of information are furnished by the macro-instruction:

1. The tag of the area to be received;
2. The tag of the area to be transmitted;
3. The ASU designation, if serial transmission is to be used.

The two components of the MOVE macro-instruction contained in the library are the instructions RCV and TMT without addresses or ASU designation. The tags of the areas to be moved are translated by the assembly to actual addresses, the proper ASU is specified and the completed instructions placed in the main program.

Control Matrix

Each partial instruction in the library has a five-character code field associated with it. This code tells the assembly how each field of the component is to be treated. The five-character code is called a control matrix and each position of the code is referred to as a cell within the matrix. Reading from left to right, the cells correspond to the columns of the Autocoder program sheet, omitting the line number (Figure 1).

A character entry in the first cell causes the insertion of up to 10 characters in the tag field of the component. An entry in the second cell causes the insertion of up to five characters in the operation field of the component. An entry in the third cell causes the insertion of up to 2 characters in the numerical field; an entry in the fourth cell up to 16 characters in the operand field; an entry in the fifth cell causes the insertion of up to 52 characters in the operand field of the component. When a component is complete, it is further processed by the Autocoder in the same manner as other entries from the main program.

Should a component in a macro-instruction refer to another component in the series, the Autocoder assembly supplies one tag per macro-instruction. Such tags are always a full ten characters in length and begin with the character N (lozenge).

Figure 2 is a list of permissible character entries into the control matrix. The control functions of each character, through which the Autocoder is told how to complete each component, are described. After a component instruction has been processed under control of the matrix, it is turned over to the main processing part of the Autocoder for normal incorporation into the output program.

TAG	OPERATION	NUM	OPERAND	COMMENTS

Figure 1

Figure 2

Cell Entries in Control Matrix:

blank	No effect.
A	Insert tag found in macro-instruction.
B	Insert operation found in macro-instruction.
C	Insert numeric found in macro-instruction.
D	Insert operand (10) characters found in macro-instructions. No test is made for character adjustment.
E	Insert π quantity as generated by the Autocoder.
F	Examine tag found in macro-instruction. If blank, consider as E; if not blank, consider as A.
J	Last 705 instruction. May be in any column.
K	Insert numeric found in macro-instruction into positions 3 and 4 of operand. May be in any column of the Matrix.
L	Insert numeric found in macro-instruction into positions 3 and 4 of operand. SUB (0) to add leading zero if necessary.
M	Insert operand. Checks for character adjustment in macro-instruction and 705 instruction, using the algebraic sum as the character adjustment in the resulting 705 instruction.
P	Move operand (maximum of 52 characters) from macro-instruction to operand field. <u>Note</u> : To move 52 characters must be in fifth column.
W	Insert $\pi\pi\pi$ quantity as generated by the Autocoder.
X	Examine tag found in macro-instruction. If blank, consider as W; if not blank consider as A.
Y	Insert $\pi\pi\pi\pi$ quantity as generated by the Autocoder.
Z	Examine tag found in macro-instruction. If blank, consider as Y; if not blank consider as A.
1	Search for the first lozenge in the macro-instruction and inserts in the component all the characters preceding it in the operand.
2	Search for the second lozenge in the macro-instruction and inserts in the component the characters between the 1st and 2nd lozenge.

Figure 2

3	Same - between 2nd and 3rd.
4	Same - between 3rd and 4th.
5	Same - between 4th and 5th.
6	Same - between 5th and 6th.
7	Same - between 6th and 7th.
8	Same - between 7th and 8th.
9	Same - between 8th and 9th.

The character in the matrix determines the place from which the information is to be obtained. An A causes information to be brought from the tag of the macro-instruction. A B causes information to be brought from the operation field of the macro-instruction. A C causes information to be brought from the numeric field of the macro-instruction. A D causes information to be brought from the operand field of the macro-instruction. In the case of multiple operand macro-instructions, a 1 causes information before the first lozenge be brought down. A 2 causes information between the first and second lozenge and so forth. Any information written after the terminal lozenge designated by the matrix is descriptive information only appearing in the output - the same function as comments.

The number of characters to be moved is determined as the lesser of two limits. Each place from which information may come has as a limit the number of characters in the field, similarly each place to which information may be sent has a limit equal to the number of characters in its field.

For instance the character 1 will cause the Autocoder to search for the 1st lozenge and insert a maximum of ten, five, two, sixteen or fifty-two characters in the component depending on whether it is placed in the 1st, 2nd, 3rd, 4th or 5th column of the matrix. The characters D, and P have identical effect, except that D establishes a limit of 10 and P a limit of 52 characters. Other exceptions to the general rule are J which flags the last entry of this macro-instruction; K and L which move two characters from the numeric column to the 3rd and 4th character positions of the operand field.

Macro-Instruction Preparation

In the preparation of a new macro-instruction which is to be added to the Library, its functions must be properly defined. For the macro-instruction MOVE, it is known that two instructions RCV and TMT, will always serve to transfer a memory area from one location to another. Therefore, the component instructions will consist of these two instructions only. It is also known that three separate pieces of information must be supplied from the macro-instruction. These are the addresses for the receive and transmit instructions and the ASU designation to specify serial or five-character transmission. The specification for MOVE will call for the macro-instruction to be written as shown in Figure 3. Tag 1 represents the location of the "from" area; while Tag 2 represents the location of the "to" area; nn represents the ASU designation from 00 to 15. The code MOVE has been decided upon as the mnemonic of the macro-instruction.

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	MOVE	nn	TAG 1 ▣ TAG 2 ▣	

Figure 3

The second consideration concerns the component instructions themselves and the arrangement of their associated matrices. The components are written on the Autocoder Program sheet in normal fashion as shown in Figure 4. Note that the instructions RCV and TMT are written in the operation column. The control matrix is written in the first five positions of the Comments column. The first two cell entries are blank because neither the tag nor operation are needed from the macro-instruction. The third character is C which tells the assembly to insert the numerical information into the instruction. The fourth character of the RCV instruction is a 2 which searches for the second lozenge in the macro-instruction operand and inserts the characters preceding it into the component operand. This is the tag of the "to" area. The fourth character of the TMT component is a 1 which inserts the characters preceding the first lozenge of the macro-instruction operand into the operand of the component instruction. This is the tag of the "from" area.

TAG	OPERATION	NUM.	OPERAND	COMMENTS
	RCV			AbC2b
	TMT			bbC1J

Figure 4

General Consideration

The following general considerations should be noted in the preparation of macro-instruction components, and in using macro-instructions.

1. Tagging of macro-instructions is optional.
2. Component instructions do not have tags; tags, as required, must be supplied through the use of the control matrix.
3. Macro-instructions either will leave the sign trigger of the ASU's undisturbed or set plus. Exceptions will be noted.
4. Macro-instructions either will leave the zero trigger of the ASU's undisturbed or turned off. Exceptions will be noted.
5. Macro-instructions will not alter the condition of the High and Equal triggers, unless otherwise specified.
6. Macro-instructions and the sub-routines they include will make wide use of ASU's 13, 14 and 15 subject to 3, 4 and 5 above and also subject to the provision that they assume to find and will leave 13 set to length 10 and 14 set to length 4. The floating arithmetic routines will in addition make free use of the accumulator. Any additional ASU requirements will be noted.
7. Character adjustment must be placed in the 11th through 16th column of the components.
8. Figures for character positions required are maximum. As literals which may be shared with the rest of the program are used, and as sub-routines are included only once no matter how often they are called for, the actual requirement will have to be determined by Autocoding or else by careful study of the entire program.
9. Literals for working storage should contain at least one non-numerical character to eliminate the possibility of conflict with literals generated by component instructions.

INDEX

<u>Macro-Instruction</u>	<u>Operation Code</u>	<u>Page</u>
Absolute, Floating	FAB	115
Add, Floating	FAD	116
Alternate Tape Units	ALTP	5
Alternate Transfer	ALTTR	69
Arctan, Floating	FATN	131
Backspace Tape (using 754 CU)	BSTP	6
Check Total	CHKT	70
Divide, Floating	FDV	117
Do Floating Point Mathematical Subroutine		129
Do Subroutine	DO P	71
	Z	
	NZ	
	MIN	
Dump on Drum	DPDR	7
Dump on Printer (Auto Control, 717 Printer)	DPPRA	9
Dump on Printer (Pgm ctl, 717 Printer)	DPPRT	11
Dump on Punch	DPPCH	13
Dump on Tape (using 754 CU)	DPTP	15
Dump on Typewriter (without checking)	DPTYP	17
End Loop	END	72
Exponential, Floating	FEX	133
First Time NOP	FTNOP	73
First Time Transfer	FTTR	74
Fix, Floating	FIX	118
Float	FLO	119
Floating Decimal Arithmetic Macro-Instructions		111
Forward Space Tape (using 754 CU)	FSTP	19
Forward Wind Tape (using 754 CU)	FWDTP	21
Halt Off	HLTOF	75
Halt On	HLTON	76
Halt Transfer	HLTTR	77
If xxx	IF E	78
	LOW	
	H	
	EH	
	EL	
	NE	
	Z	
Input-Output Macro-Instructions	NZ	

INDEX
(cont'd)

<u>Macro-Instruction</u>	<u>Operation Code</u>	<u>Page</u>
Load Left Location	LLL	79
Load Left Location 14	LLL 14	80
Load Right Location	LRL	81
Load Right Location 14	LRL 14	82
Logical Macro-Instructions		67
Logical Read While Write Tape	RWWLG WWRTP	23
Loop	LOOP	83
Move	MOVE	84
Move Characters	MOVEC	85
Move Instruction Address	MOVEI	86
Multiply, Floating	FMP	120
Natural Logarithm, Floating	FLN	135
Order Check	ORDCH	87
Print Under Automatic Control (717 Printer)	PRNTA	27
Print Under Program Control (717 Printer)	PRINT	29
Programmed Halts in Floating Point Operations		113
Punch	PUNCH	31
Read Card	RDCD	33
Read Drum	RDDR	35
Read Tape (using 754 CU)	RDTP	37
Read While Write Tape (using 754 CU)	RWWTP	39
Read While Write Tape, Logical	RWWLG WWRTP	22
Repeat, ASU Controlled	RPTA	89
Repeat, Memory Controlled	RPTM	91
Reset Add, Floating	FRA	121
Reset Subtract, Floating	FRS	122
Restore Comparison	RCMP	93
Restore Sign of ASU	RSGN	94
Rewind Tape (using 754 CU)	RWDTP	42
Save Comparison	SCMP	95
Save Sign of ASU	SSGN	97
Sequence Check	SEQCH	99
Set Switch to NOP	SWNOP	101
Set Switch to TR	SWTR	102
Set Up	SETUP	103
Sine, Floating	FSIN	137
Square Root, Floating	FSQR	139
Store, Floating	FST	123

INDEX
(cont'd)

<u>Macro-Instruction</u>	<u>Operation Code</u>	<u>Page</u>
Subtract, Floating	FSU	124
Transfer Equal or High	TREH	104
Transfer Equal or Low	TREL	105
Transfer Low	TRLOW	106
Transfer Minus	TRMIN	107
Transfer Minus, Floating	FTM	125
Transfer Non-Zero	TRNZ	108
Transfer Not Equal	TRNE	109
Transfer Plus, Floating	FTP	126
Transfer Zero, Floating	FTZ	127
Type (without error correcting)	TYPE	43
Type and Check	TYPCK	45
Write Check Tape (using 754 CU)	WRCTP	47
Write Drum	WRDR	51
Write Erase Printer (717 Printer, Auto Ctl)	WREPA	53
Write Erase Printer (717 Printer under pgm ctl)	WREPR	55
Write Erase Punch	WREPN	57
Write Erase Tape (using 754 CU)	WRETP	59
Write Erase Typewriter (without error correcting)	WRETY	61
Write Tape (using 754 CU)	WRTP	63
Write Tape Mark (using 754 CU)	WRTM	65

It is suggested that this manual be maintained in a loose-leaf form in order that new material may be easily added.

Macro-instructions and subroutines for the TRC and 760 are being prepared and will be distributed when completed.

A listing of the macro-instructions and subroutines as they appear on the Autocoder System tape is available upon request from the Program Librarian, Customer Assistance Department, IBM, 590 Madison Avenue, New York 22, New York. One master, suitable for reproduction, will be provided each account requesting this listing.

IBM[®]
*International
Business Machines
Corporation*

590 MADISON AVENUE
NEW YORK 22, N. Y.

32-7479

PRINTED IN U. S. A.