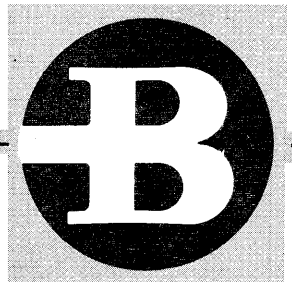


Burroughs E 8000

BASIC ASSEMBLER

REFERENCE MANUAL

Burroughs

E 8000
BASIC ASSEMBLER

REFERENCE MANUAL

$5.00

Burroughs Corporation
Detroit, Michigan 48232

AA 35,431

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

INTRODUCTION

An assembly program, such as the Burroughs E 8000 Basic Assembler, is a programing tool designed to alleviate part of the effort required in coding programs. In addition, it provides hard copy documentation for future reference.

The Assembler accepts symbolic language as input and automatically converts this input to internal machine language. The term symbolic refers to the use of non-machine language address and mnemonic operation codes in place of machine language codes. For example, total debits might be located in memory at the symbolic address 12P which might be equal to the actual machine address of 75. Three-character operation codes in symbolic language, such as RTN are converted by the Assembler to their machine language equivalents, in this case 8800. By using an assembly program, the programmer is not burdened with keeping track of the memory locations used, their addresses, or the actual machine language for the instructions being used. Since the assembly program will do this, and more, for the programmer, he is free to code programs at a greater speed.

Programs are coded with symbolic language on coding sheets which, in turn, serve as media for punching the symbolic input cards.

Three versions of the Basic Assembler are available. One uses the Burroughs E 8000 for assembly, one the Burroughs B 3500 and one the Burroughs B 300. All versions use punched cards for the symbolic input.

SECTION 1

CODING FORM

This section introduces the coding form. The examples that are provided are used to illustrate the use of of the coding form with specific instructions. Further explanation and clarification of their instructions and their use will follow in Section Two.

All symbolic programs for the E 8000 Assembler are written on the coding form (MKTG 2267). Customer, branch and programmer information is shown in the upper right-hand section of the form. A description of other sections follows.

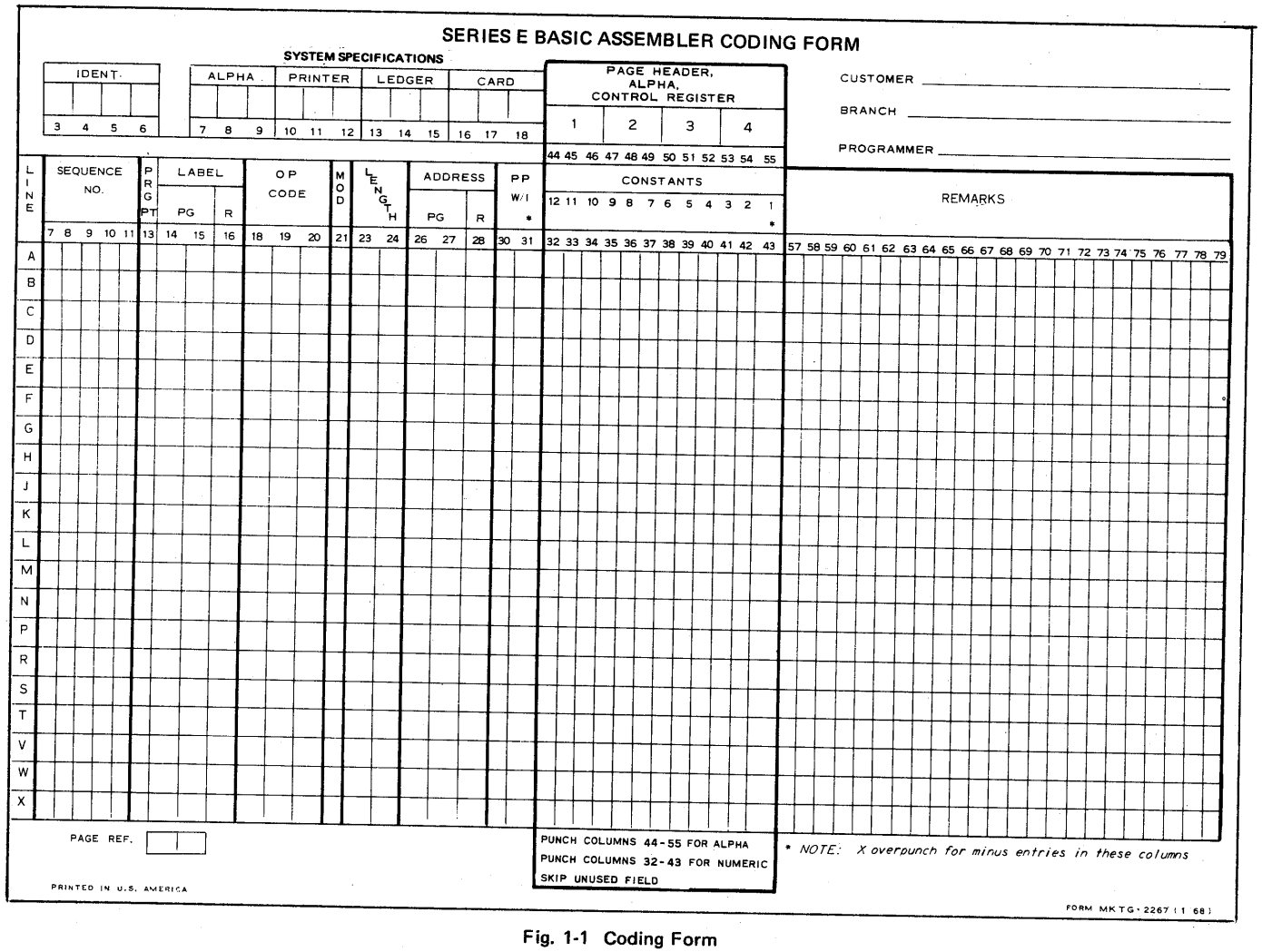
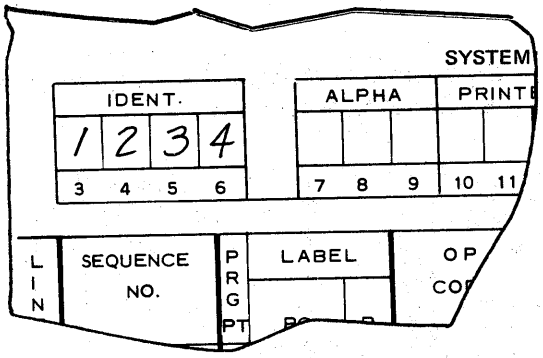


Fig. 1-1 Coding Form

USE OF THE CODING FORM

IDENTIFICATION

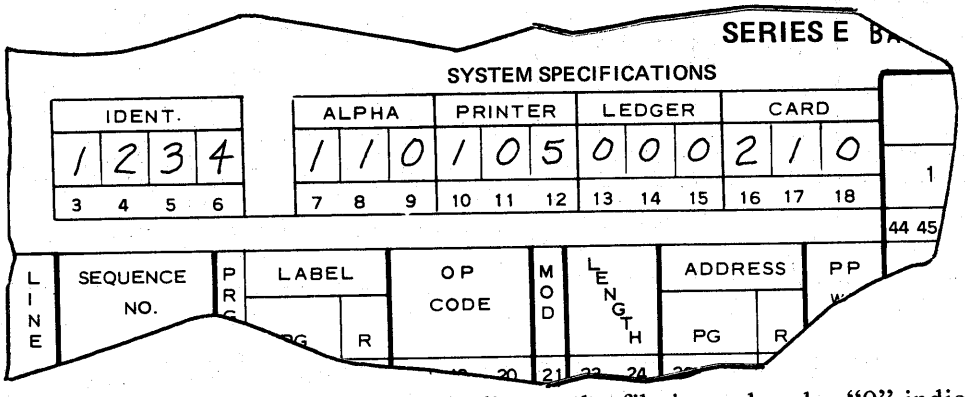
EXPLANATION – The identification field is used to identify a specific program. The identification may be alpha and/or numeric. It is entered on the first coding sheet of each symbolic program.



FIELD DEFINITION – The program identification is four (4) digits in length and is keypunched in card column 3-6 of the first card of each symbolic program. The standard drum card will automatically DUP the identification in all succeeding cards.

SYSTEMS SPECIFICATIONS

EXPLANATION – The Systems Specifications are entered on the first coding sheet of each program. These specifications are used to inform the Assembler of the input-output files to be used with this program. There are four fields in the specifications section, each consisting of 3 digits.



A "1" in the most significant position of the field indicates the file is used and a "0" indicates the file is not used. When the card file is to be used, a "1" specifies that numeric punched cards are to be read and a "2" specifies alpha punched cards are to be read. The adjacent two columns specify the number of words per decade to be reserved by the file. When any file is not used, three "0's" must be punched in the field.

FIELD DEFINITION

ALPHA – The alpha field specifies whether an alpha file is to be used and the number of words to be reserved in each of two decades for alpha constants. The maximum is ten (10) words in each decade. The alpha field uses card columns 7-9.

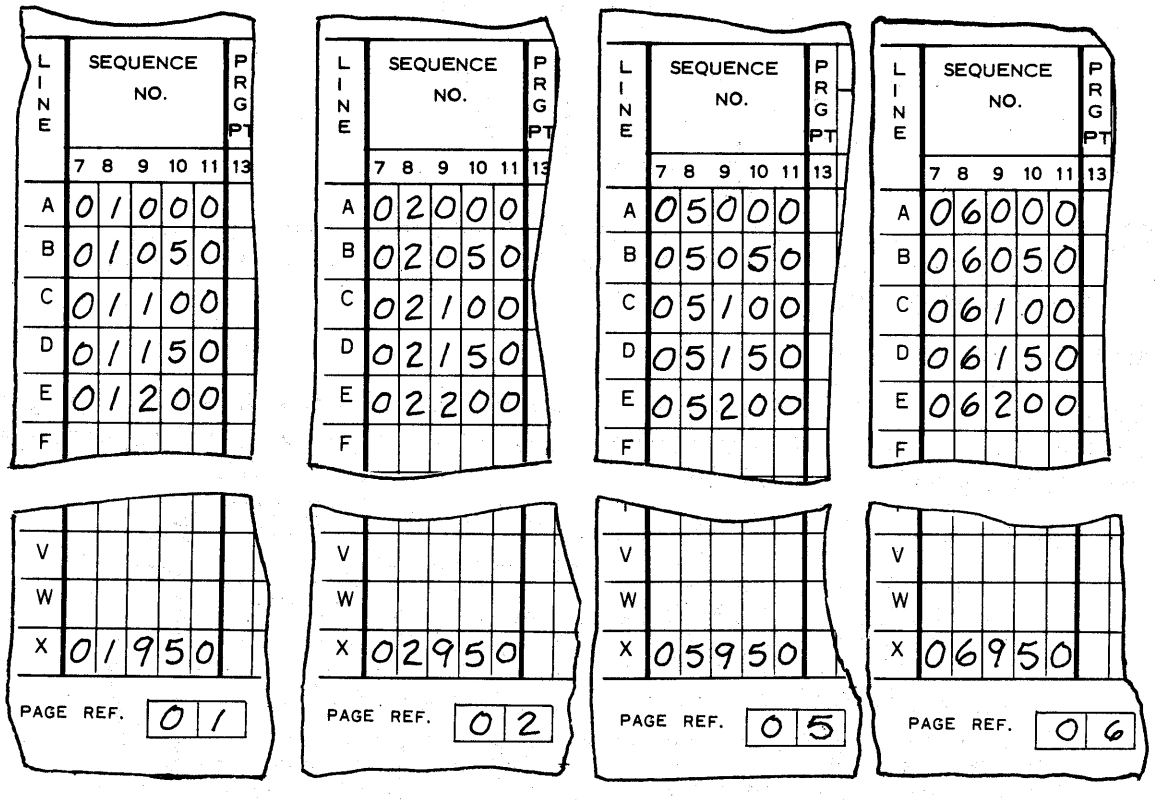
PRINTER – The line printer field informs the assembler whether the line printer is to be used and the number of words to be reserved for a print area. The maximum is ten (10) words in each of three consecutive decades. The printer field uses card columns 10-12.

LEDGER – The ledger field informs the assembler whether striped ledgers are to be used and the number of words to be reserved for the ledger area. The maximum is ten (10) words in each of two consecutive decades. The ledger field uses card columns 13-15.

CARD – The card field specifies whether the card reader is required and the number of words to be reserved. The maximum is ten (10) words when numeric input is used and 10 each of two consecutive decades when alphanumeric is used. The card field uses card columns 16-18.

SEQUENCE NUMBER

EXPLANATION – The sequence number field is used to number the symbolic program cards in ascending order. It is suggested that the following sequence numbering be used:



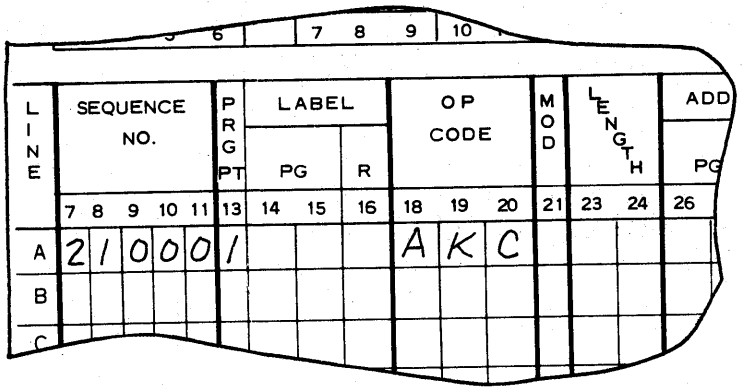
From the above examples it will be noted that the sequence number includes the page number in card columns 7 and 8. This method will assist the programmer in quickly locating specific sections of the program.

FIELD DEFINITION – The sequence numbers assigned to the symbolic program cards are punched in card columns 7-11.

PROGRAM POINT

DEFINITION – A Program Point is a digit, 1 through 9, used to reference an entry in the Symbolic Program.

EXPLANATION – Placement of a Program Point on a line of the coding sheet will force the instruction on that line to be the first syllable of the word. It is used to identify a place in memory to which the program will branch. A Program Point cannot be placed on a line containing a label, likewise a label cannot be placed on the same line containing a Program Point. A Program Point can be reused several times within a symbolic program, however, the total number of Program Points used within a Symbolic Program cannot exceed 99.



In the above illustration, a Program Point is entered in Column 13. The program can branch to this Program Point and act on the instruction AKC.

FIELD DEFINITION – Program Points 1-9 are punched in card column 13.

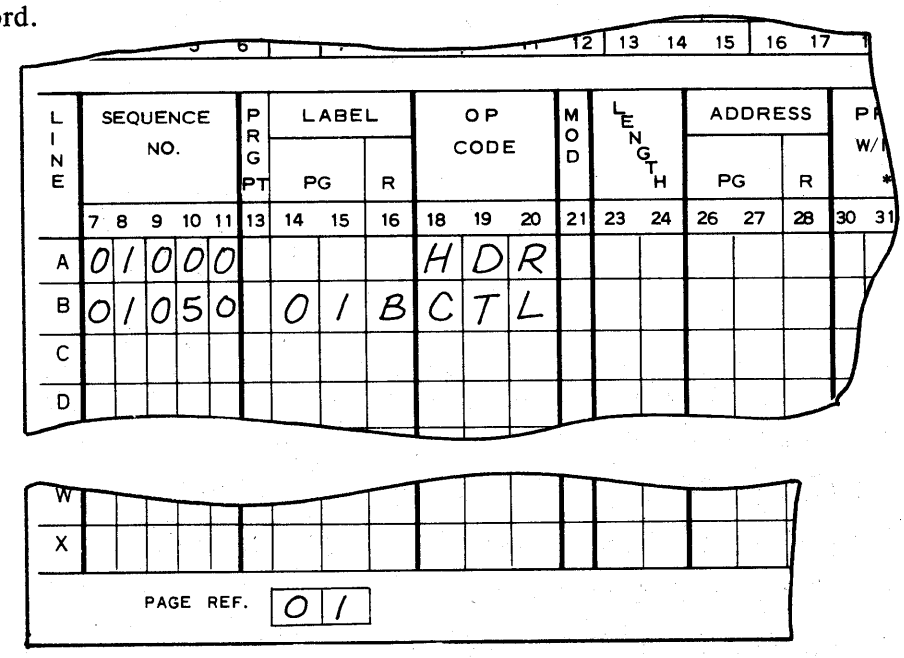
LABEL

DEFINITION – A label is an identifier used to reference a specific entry in the symbolic program.

EXPLANATION – The first two digits of the label are the coding sheet page number and the third character is the line letter.

Labels are used to identify symbolic entries used for: (A) accumulation, (B) constants, (C) control register constants and (D) branch locations.

A label, unlike a Program Point, cannot be reused within a Symbolic Program, however, there is no limit to the number of times it may be addressed. The total number of labels in any one Symbolic Program cannot exceed 96 including overlays. A label will force the instruction on that line to be the first syllable of the word.



In the above illustration, the label 01B identifies the location containing the CTL information. Any future reference to the location will be addressed by the label 01B.

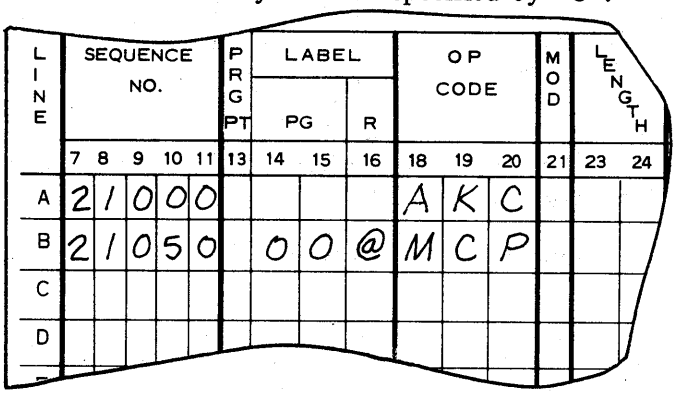
FIELD DEFINITION – The characters of a label are punched in card columns 14-16.

The special label 00@ forces the instruction on that line to be the first syllable of the word. The label 00@ is reusable and does not count towards the maximum of 96.

OPERATION CODE

DEFINITION – The Operation Code is the mnemonic representation for a machine instruction to be performed.

EXPLANATION – Operation Codes are instructions to perform arithmetic, control console, data movement, decisions and other code functions. The instruction is a 3-character alpha code. The first letter specifies the function (AXX = Add, MXX = Multiply). The last two characters specify locations – AKC = Add the contents of K to the memory location specified by "C".

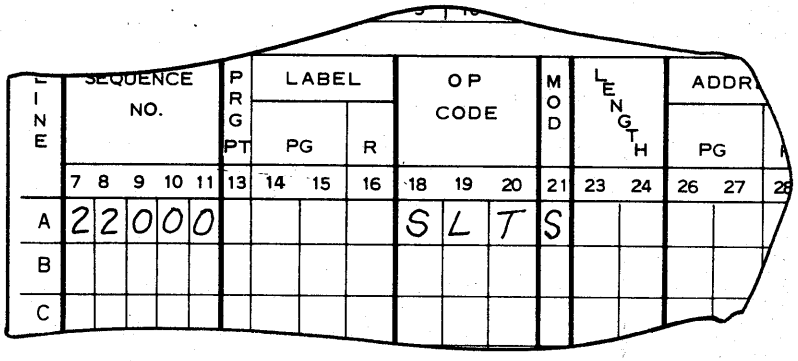


FIELD DEFINITION – The Operation Code is punched in card columns 18-20.

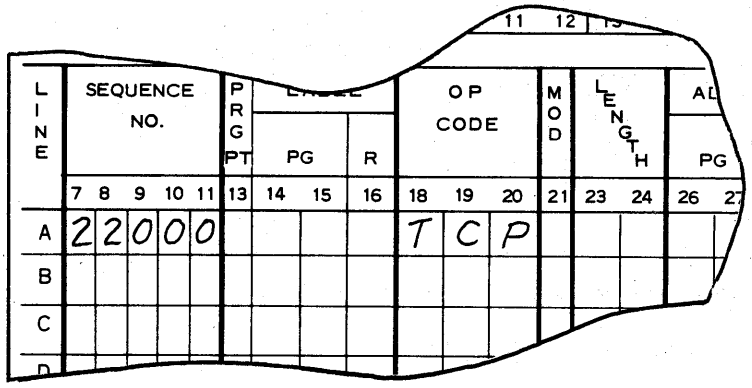
MODIFIER

DEFINITION – A Modifier is used to change a base instruction.

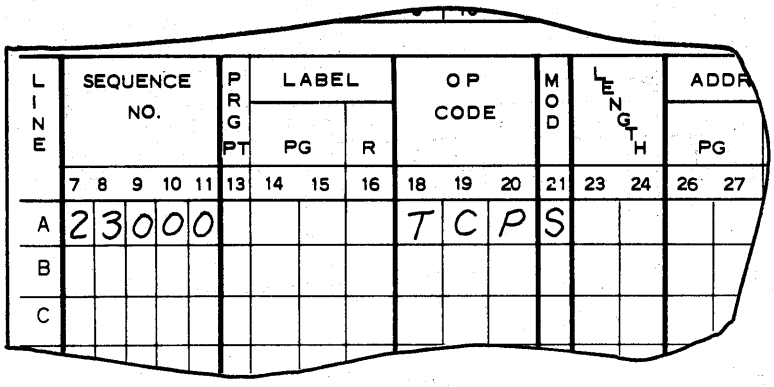
EXPLANATION – Some instructions require a modifier to complete the instruction.



The illustration above shows that a Modifier "S" is required to show a single word shift. A "D" Modifier is used for a double word shift. An instruction may or may not require a modifier.



The above instruction (TCP) is a complete instruction and transfers 12 columns.



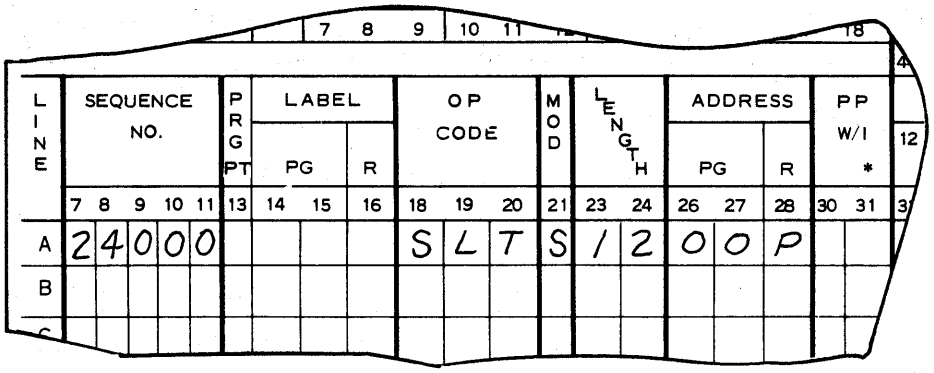
The Modifier "S" in this illustration changes the base instruction to transfer-only columns 1-6.

FIELD DEFINITION – The Modifier is punched in card column 21.

LENGTH

DEFINITION – The length will further define the instruction.

EXPLANATION – Some instructions require a specific length to complete the instruction. All shift instructions require a length of shift of from 1 to 12 places.



The above instruction will cause P to shift 12 places left.

FIELD DEFINITION – The length for an instruction is punched in card columns 23 and 24.

ADDRESS

DEFINITION – An address is a symbolic reference to a location in memory.

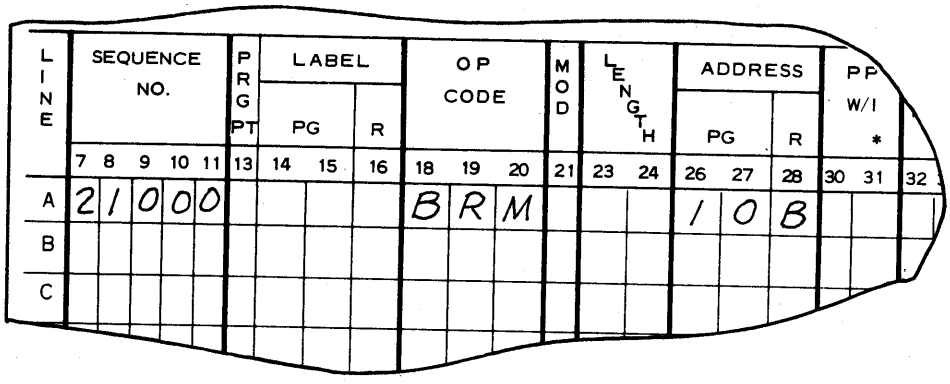
EXPLANATION – An address may be a label, or one of the four directly addressable memory locations, which are:

00P – (MA 000)

00W – (MA 009)

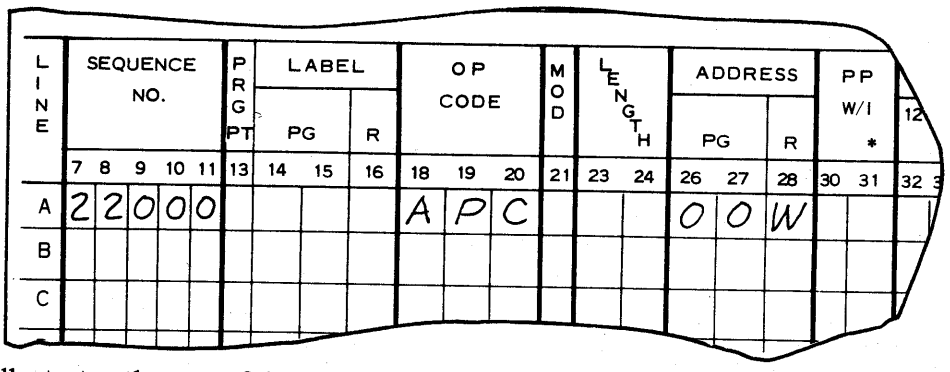
00B – (MA 010)

00K – (MA 011)



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31
A	2 1 0 0 0				B R M			1 0	B	
B										
C										

The above illustrates an instruction to branch on minus to label 10B. Label 10B was declared to be a specific entry in the symbolic program.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31
A	2 2 0 0 0				A P C			0 0	W	
B										
C										

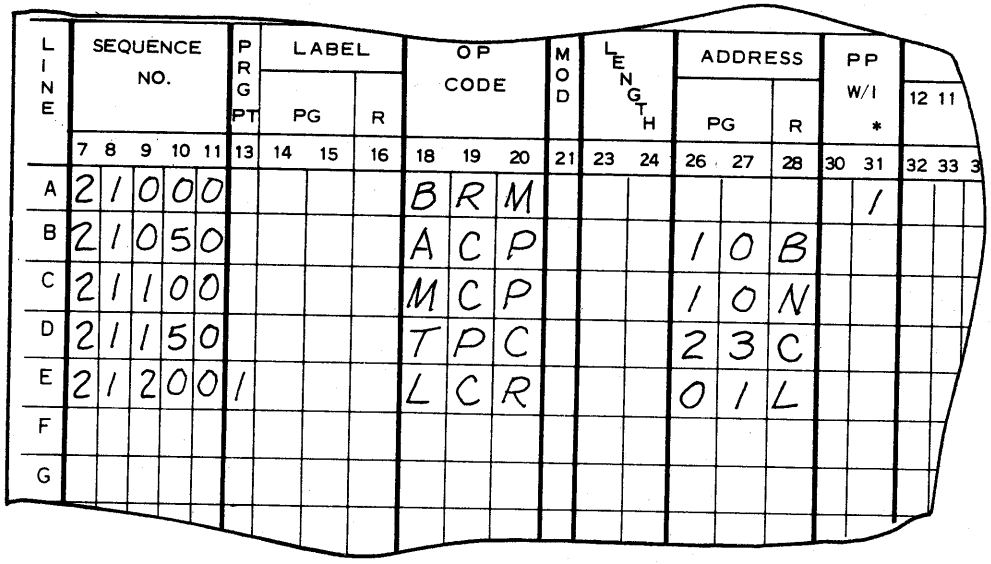
This instruction illustrates the use of 00W (MA 009) as an address.

FIELD DEFINITION – The address is punched in card columns 26-28.

PROGRAM POINT (ADDRESS)

DEFINITION – A Program Point is a digit, 1 thru 9, to define a location in the symbolic program.

EXPLANATION – The Program Point located in card column 31 is the branch out of the program. The Program Point located in card colunn 13 is the branch in of the program.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31
A	2 1 0 0 0				B R M					1
B	2 1 0 5 0				A C P			1 0	B	
C	2 1 1 0 0				M C P			1 0	N	
D	2 1 1 5 0				T P C			2 3	C	
E	2 1 2 0 0	1			L C R			0 1	L	
F										
G										

At line A, if the minus condition exists, the program will branch to line E where a program point of 1 is located.

EXPLANATION – When a Program Point is plus, as in the above illustration, the program will branch forward until the same digit is located. If the Program Point is minus (an X over-punch) the program will return until that same digit is located.

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31
A	21000				LCR			01	F	
B	21050	1			PAC			00	P	
C	21100				AKC			00	P	
D	21150				SKC			00	W	
E	21200				BRN					1X
F										
G										

The program will return to line B until a clear condition exists at line E.

EXPLANATION – A Program Point cannot be placed on a line containing an address nor can an address be placed on a line containing a Program Point.

FIELD DEFINITION – The Program Point is punched in card column 31.

WORD INCREMENT (DECREMENT)

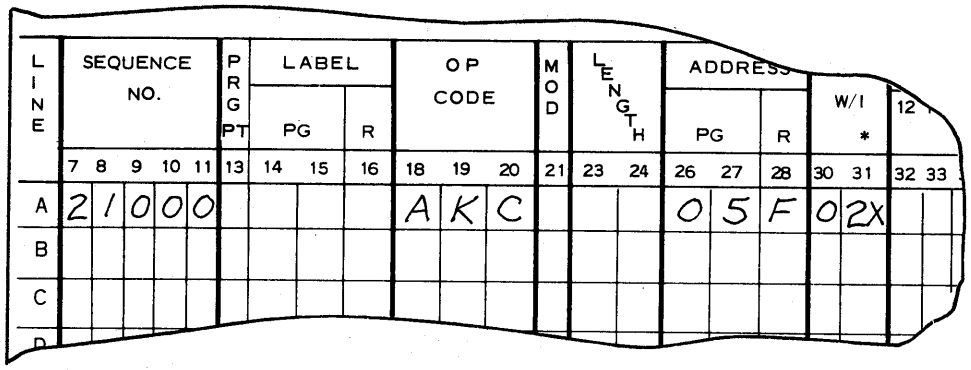
DEFINITION – A word Increment or Decrement is a two-digit number that changes the symbolic address.

EXPLANATION – A word Increment is always associated with an address. It will cause the address to be increased by the value of the Increment.

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31
A	21000				AKC			05	C	02
B										
C										

In the above illustration, should address 05C represent MA 020, the incremented "C" MA would be MA 022.

EXPLANATION – A word Decrement will cause the address to be decreased by the value of the decrement.



In the above illustration, if address 05F represents MA 025, the decrement 02 (minus) will cause the "C" MA to be MA 023.

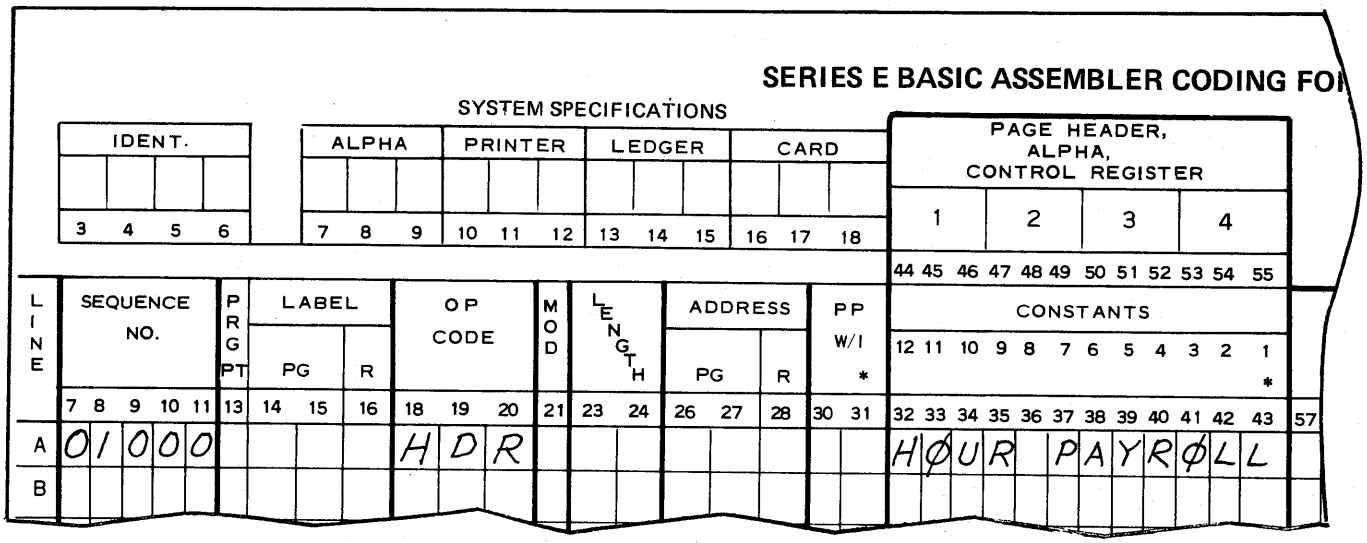
FIELD DEFINITION – Word Increments and Word Decrements are punched in card columns 30 and 31. Word decrements must have an X over-punch in card column 31. A "0" must be punched in card column 30 if the increment or decrement is a single digit.

ALPHA ENTRIES

PAGE HEADER

DEFINITION – The Page Header titles each page of the program print-out.

EXPLANATION – The first instruction of the Mainline Program is a header card (OP Code - HDR). The alpha description of the program is shown in the alpha section. The description can be a maximum of 12 characters in length and is printed in the upper left-hand corner of each Symbolic Program sheet.

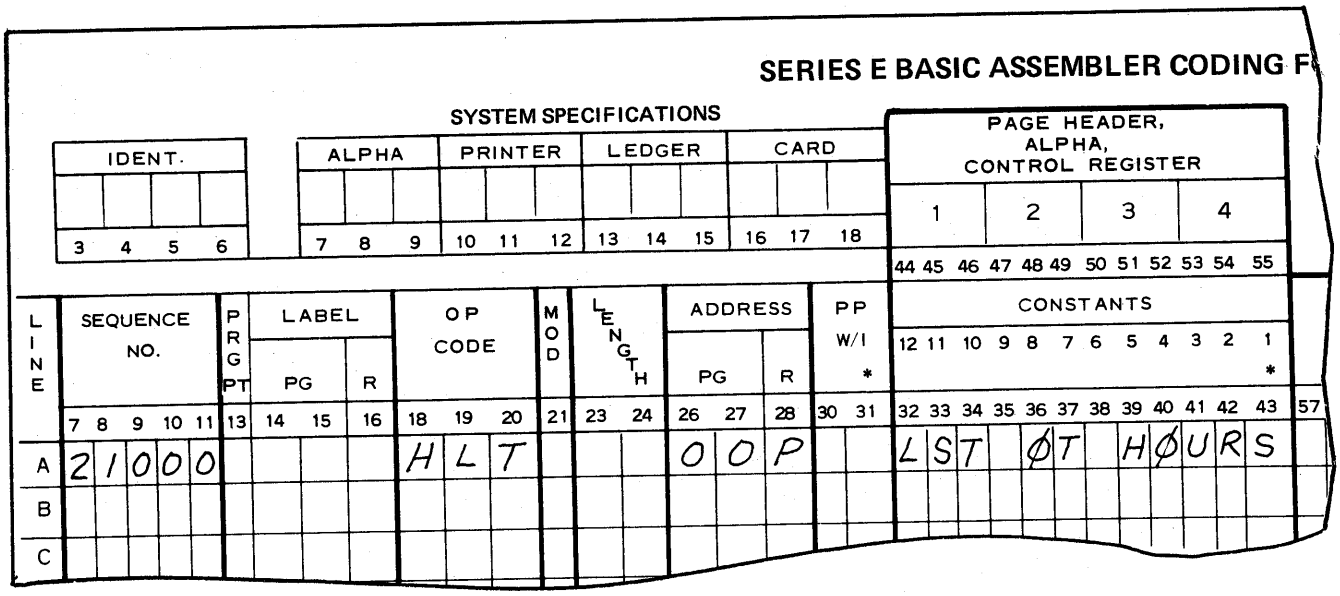


FIELD DEFINITION – The alpha description of the program is punched in card columns 44-55 of the header card.

ALPHA

DEFINITION – Special instructions to the operator to permit easy identification.

EXPLANATION – Operator or other instructions may be coded in the alpha field. These instructions are offset to the left for easy reference. Any special reference card must contain a sequence number.



FIELD DEFINITION – Alpha instructions are punched in card columns 44-55. If more than 12 digits of alpha are required, the remarks section may be used.

CONTROL REGISTER

DEFINITION – Control Register Constants are the instructions loaded into the Control Register prior to a halt instruction.

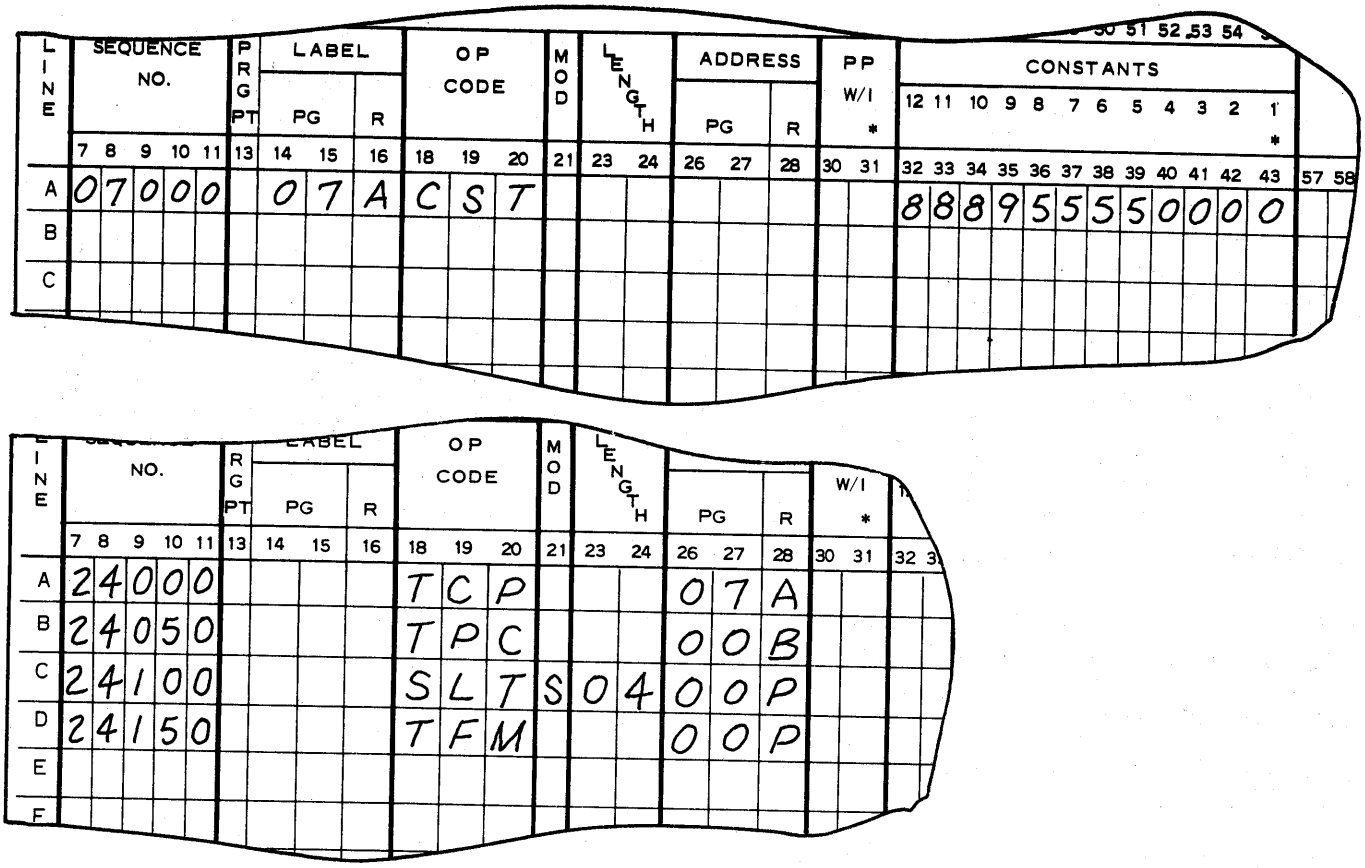
EXPLANATION – Control Register instructions are 3-digit instructions shown in the alpha section. A total of four instructions may be on a single line, however, if more are required, additional lines may be used. If a position check is used, it must be the last instruction.

CONSTANTS

DEFINITION – The constant field is used to specify numeric values associated with the defined constants.

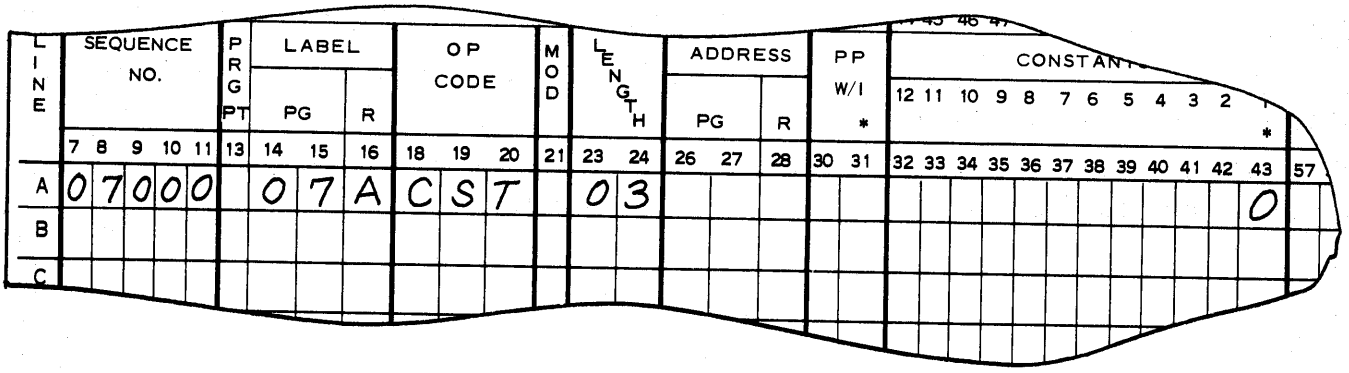
EXPLANATION – Numeric constants must be preceded by OP Codes CST or MSK. Constants are 12 digits in length and may be plus or minus.

Constants may be for accumulations, instructions, numeric expression of alpha, masks, tables, work areas, vertical spacing from type from memory, counters, limits, factors, etc.



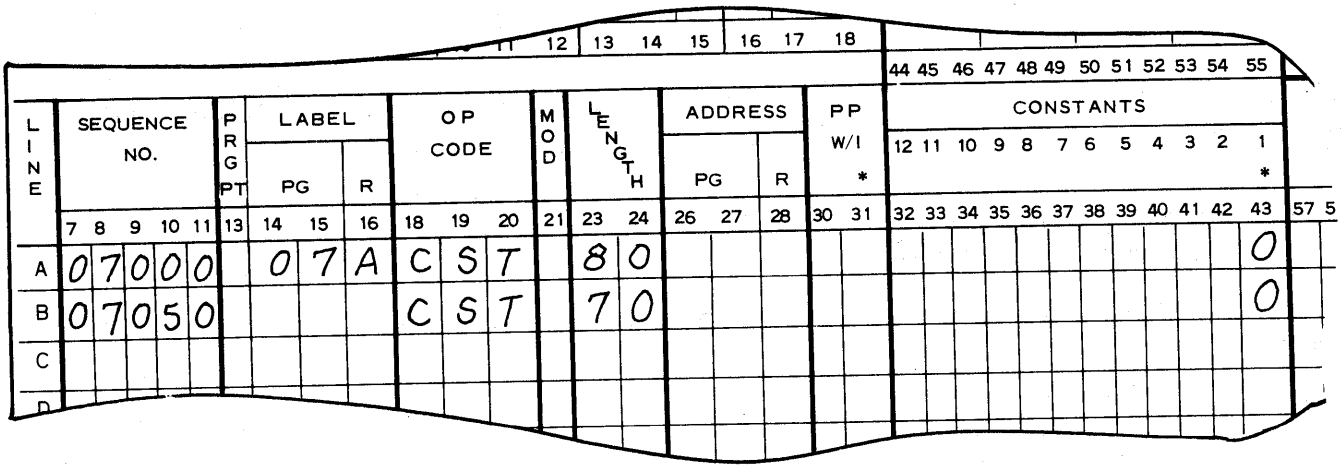
The above illustrates a constant for vertical spacing (4 spaces) such as used for filled sheet in run.

EXPLANATION – Numeric constants may be reserved as single constants or as sequential constants.



The above example shows reserving three MA's in sequence. To address the second and third constant the address would be 07A with word increment 01 and 02 respectively.

EXPLANATION – A maximum of 99 constants may be reserved with one instruction. Should more than 99 sequential constants be required, additional instructions are necessary.



The above illustrates reserving 150 sequential constants for distribution accumulations. These constants are addressed by correctly incrementing the base label 07A by using index register (TIX). (See Section 2 Page 35.)

FIELD DEFINITION – Numeric constants are punched in card columns 32-43. When a constant is used for accumulation, a "0" is punched in card column 43 to ensure that all MA's are clear.

REMARKS

EXPLANATION – The Remarks Section is used to insert remarks pertinent to the instruction. For example, it may give the scale number of a PAC (Print And Cycle) instruction or instructions to an operator for keyboard listing.

FIELD DEFINITION – Alpha remarks are punched in card columns 57-79.

SECTION 2

ASSEMBLER PROGRAMING

This section contains all of the instructions used when writing a symbolic program. These instructions are broken down into two general groups; E 8000 instructions and pseudo instructions.

The group dealing with E 8000 instructions is further divided into two subgroups; central processor instructions and control register commands. The commands and instructions are covered in an alphabetic sequence, by function, within its respective group.

Certain instructions discussed use symbols which represent special memory locations in the central processor, they are:

1. P or 00P represents word 000
2. K or 00K represents word 011
3. B or 00B represents word 010
4. W or 00W represents word 009

E 8000 INSTRUCTIONS

CENTRAL PROCESSOR COMMANDS

A. Arithmetic Commands

ADD – The function of this instruction is to perform a single-address algebraic addition of the first memory location specified in the command to the second memory location. The result will appear in the second location with no change to the first location.

The format of the add commands are:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	20000				ACK			05	B			ADD C TØ K
B	20050				ACP			05	C			ADD C TØ P
C	20100				AKC			05	D			ADD K TØ C
D	20150				APC			05	E			ADD P TØ C
E	20200				APC			05	E	01		ADD P TØ C PLUS ØNE
F												
G												
H												

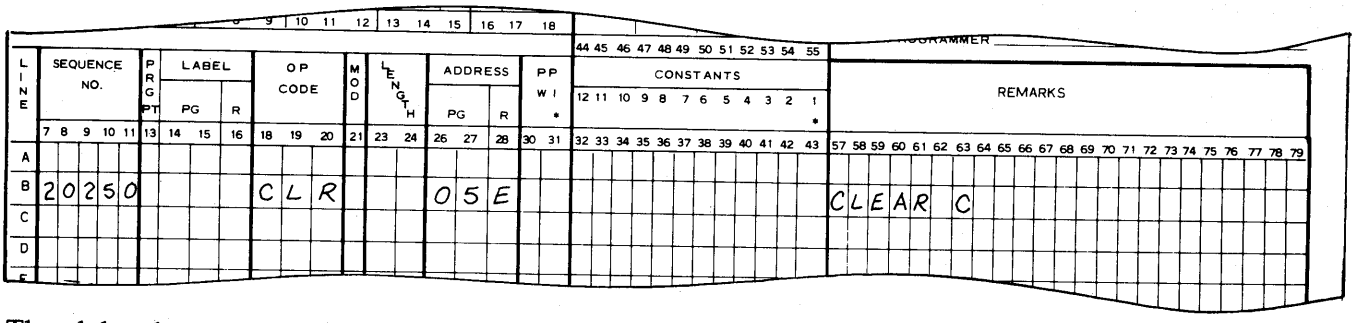
The alpha character (A) in column 18 indicates that this is an add command. The alpha character (C, K or P) in column 19 indicates the first memory location specified in the command. Therefore, it will be added to the second memory location, specified by the alpha character (K, P or C) in column 20. The sum or answer of the add instruction will appear in the second location and the factor in the first location will not be changed.

The two-digit numeric and one-digit alpha entry in the address field (cc 26-28) is the symbolic address, and will always identify the C memory location whether it is the first or second location in the instruction. The symbolic address must have been identified earlier as a label (see Page 1-4) and therefore will represent a specific memory location.

The two-digit number in the word increment field (cc 30-31) will modify the "C" address plus or minus by the number indicated in that field. As in above example, sequence No. 20200 above, "P" (word 000) would be added to 05E plus one memory location. The result of that addition would be in 05E plus one memory location and the contents of "P" would remain unchanged.

CLEAR – The function of this command is to clear the "C" location specified in the address field.

The format of this command is:



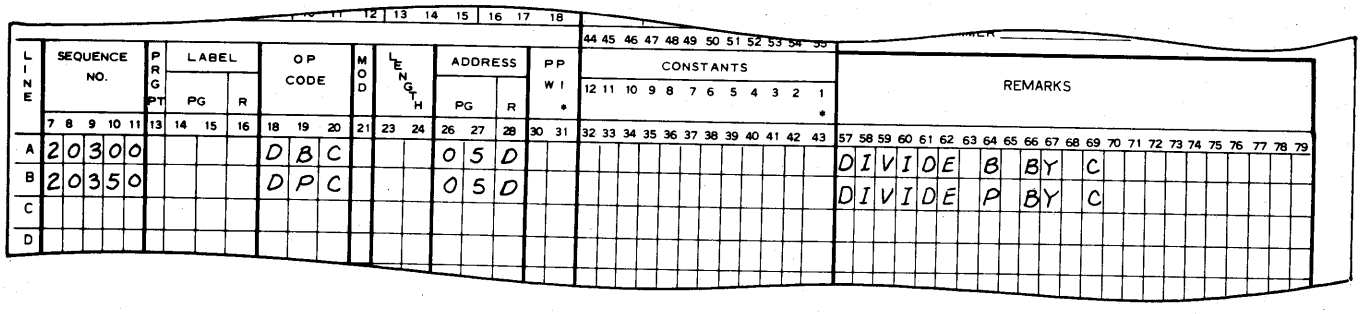
The alpha characters (CLR) in columns 18-20 indicate that this is the clear command. The symbolic address 05E is the "C" memory location which will be cleared by this command. The resulting contents of the "C" memory location after performing this command will always be .00+.

The symbolic address 05E in the example above must have been identified earlier as a label and therefore will represent a specific memory location.

The address may be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

DIVIDE – The function of this command is to perform a single-address algebraic division. That is, it will divide the first memory location specified in the command by the second memory location, and store the result in the first memory location.

The format of the divide commands are:



The alpha character ("D") in column 18 indicates that this is a divide command. The alpha character (B or P) in column 19 indicates the first memory location specified in the command and therefore the dividend for the divide operation. The alpha character ("C") in column 20 in conjunction with the specified symbolic address in columns 26-28 identify the second memory location of the instruction and therefore the divisor of the divide operation. The quotient or answer will appear in "B" or "P" depending upon which is active in the instruction. The remainder will appear in 00K as a plus amount. It is not necessary to clear any "C" address prior to the divide operation.

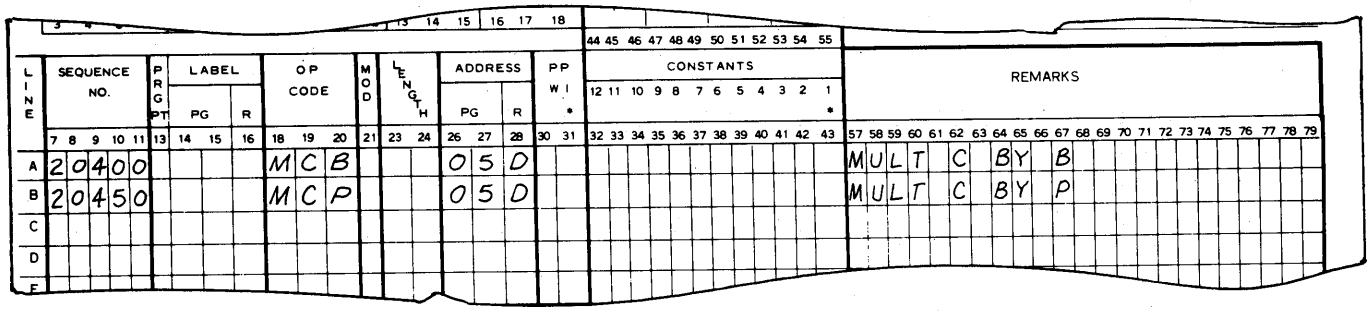
The symbolic address 05D in the example above must have been identified earlier as a label and therefore will represent a specific memory location.

The symbolic address may be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

MULTIPLY

MULTIPLY – The function of this command is to perform a single-address algebraic multiplication. That is, it will multiply the first memory location specified in the command by the second memory location, and store the result in the second memory location.

The format of the multiply commands are:



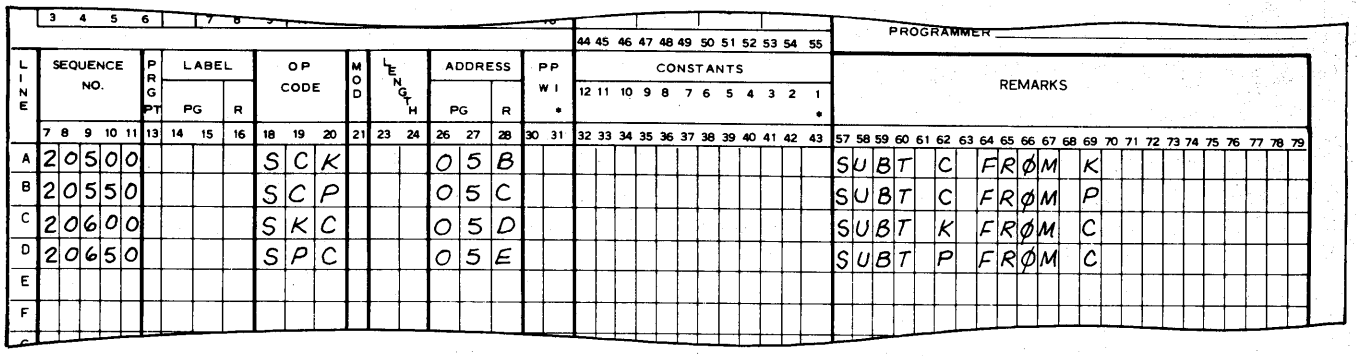
The alpha character ("M") in column 18 indicates that this is a multiply command. The alpha character ("C") in column 19 in conjunction with the specified symbolic address in columns 26-28 identify the first memory location of the instruction and therefore the multiplicand of the multiply operation. The alpha character (B or P) in column 20 indicates the second memory location of the instruction and therefore the multiplier of the multiply operation. The product or answer will appear in "B" or "P" depending upon which is active in the instruction. It is not necessary to clear any "C" address prior to the multiply operation.

The symbolic address 05D in the example above must have been identified earlier as a label and therefore will represent a specific memory location.

The symbolic address may be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

SUBTRACT – The function of this command is to perform a single-address algebraic subtraction. The first memory location specified in the command will be subtracted from the second memory location. The result will appear in the second memory location with no change to the first.

The format of the subtract commands are:



The alpha character ("S") in column 18 indicates that this is a subtract command. The alpha character (C, K or P) in column 19 indicates the first memory location specified in the command. Therefore it will be subtracted from the second memory location, specified by the alpha character (K, P or C) in column 20. The remainder or answer will appear in the second memory location; the first will not be changed by this command.

The two-digit numeric and one-digit alpha entry in the address field (cc 26-28) is the symbolic address and will always identify the C memory location whether it is the first or second location in the instruction.

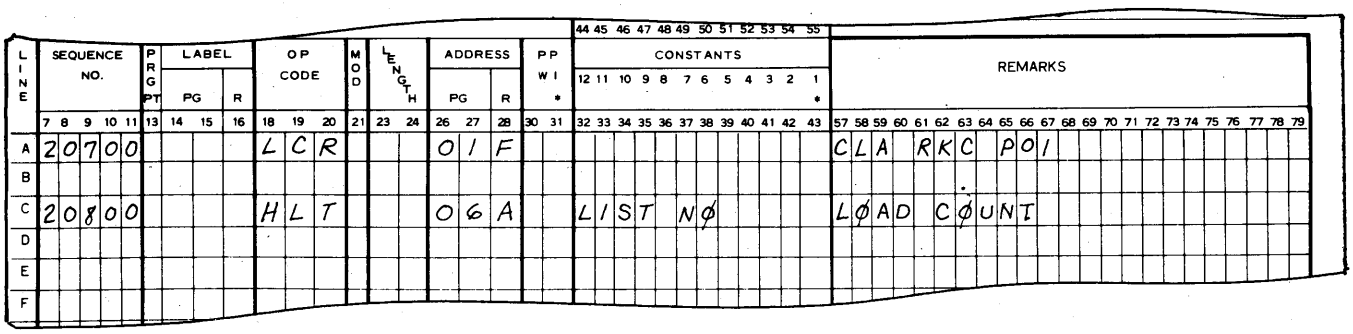
The symbolic address in columns 26-28 must have been identified earlier as a label and therefore will represent a specific memory location.

The symbolic address may also be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

B. Control Console Instructions

HALT – The function of this command is to halt, or stop, further processing by the E 8000 Central Processor. Processing will resume with a manual depression of one of the motor bars.

The format of the halt command is:



The alpha characters (HLT) in columns 18-20 indicate that this is a halt command. When this command is sensed by the central processor, control of the system is turned over to the operator for manual key-

board entry. This command must be used when it is desired to have the operator manually enter numeric data into the system.

The symbolic address (06A in the example above) will always identify the "C" memory location which is to be active with this command. The symbolic address must have been identified earlier as a label and therefore will represent a specific memory location.

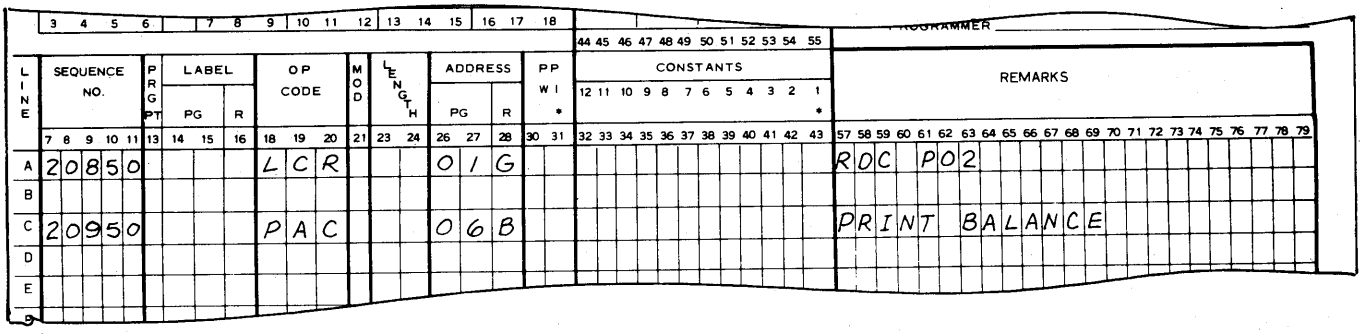
Before the HLT instruction may be executed, a load control register instruction must be programed as discussed in the other instructions section. (See Page 2-25.)

The symbolic address specified by the HLT command may also be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

Once the operator depresses a motor bar, control of the system automatically returns to the central processor which in turn will move to the next instruction in sequence.

PRINT AND CYCLE – The function of this command is to stop processing and cause the console to automatically cycle. Once an automatic machine cycle occurs, the central processor will resume processing with the next instruction in sequence.

The format of the print and cycle command is:



The alpha characters (PAC) in columns 18-20 indicate that this is a print and cycle command. When this command is sensed by the central processor, control of the system is given to the console for one automatic machine cycle. This command must be programed any time it is desired to have the console automatically print numeric data, or at any time you wish to cause an automatic cycle after the insertion of a striped ledger.

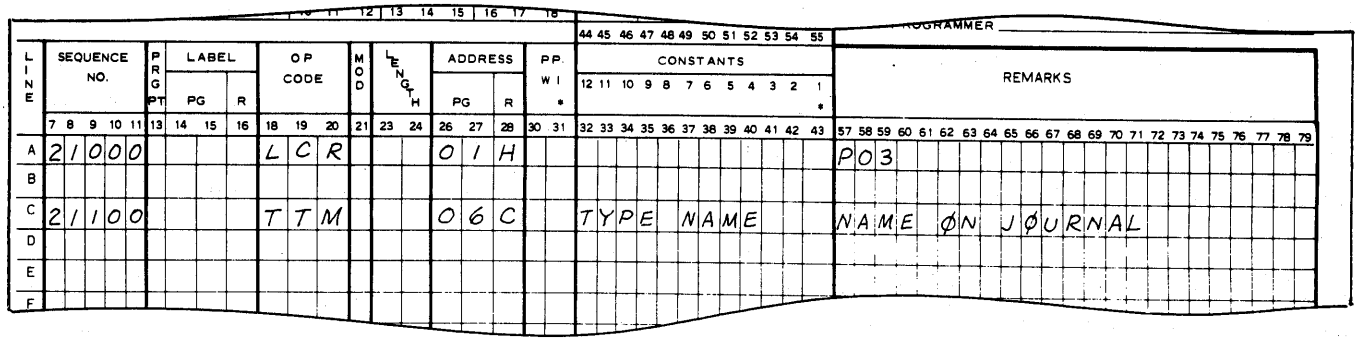
The symbolic address (06B in the example above) will always identify the "C" memory location which is to be active with this command. The symbolic address specified by the command must have been identified earlier as a label and therefore will represent a specific memory location.

Before the PAC instruction may be executed, a load control register instruction must be programed as discussed in the other instructions section. (See Page 2-25.)

The symbolic address specified by the PAC command may also be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

TYPE TO MEMORY – The function of this command is to stop processing and set up type to memory without cycling the console.

The format of the type to memory command is:



The alpha characters (TTM) in columns 18-20 indicate that this is a type to memory command. When this command is sensed by the central processor, control of the system is given to the operator to provide for manual typing. This instruction must be programed at any time it is desired to have the operator manually type data into the memory of the central processor. When this command is sensed, an automatic machine cycle will not occur. If it is desired to type without typing to memory, the HLT command would have been used. Internal processing will automatically resume upon the depression of any of the end alpha keys (K1, K2 or K4).

The symbolic address (06C in the example above) will always identify the "C" memory location which is to be active with this command. The symbolic address specified by the command must have been identified earlier as a label in a file declaration and therefore will represent a specific memory location.

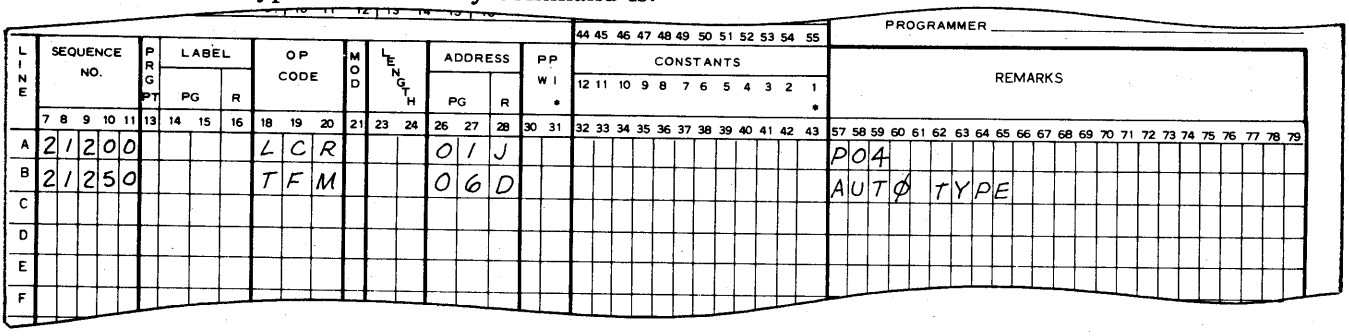
Since each alpha character is represented in memory as two numeric digits stored ten memory locations apart, a twelve-character alpha message under the above programing would have been stored in 06C and 06C plus ten. If the message being typed were more than twelve characters, the remainder (up to 12 characters) would have gone to 06C plus one and 06C plus eleven. The maximum alpha message may be 120 characters long.

Before the TTM instruction may be executed, a load control register instruction must be programed as discussed in the other instructions section. (See Page 2-25.)

The symbolic address specified by the TTM command may also be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

TYPE FROM MEMORY – The function of this command is to stop processing and automatically begin typing from memory without cycling the console. Internal processing will resume when the end alpha code is sensed.

The format of the type from memory command is:



TYPE FROM MEMORY
SHIFT

The alpha characters (TFM) in columns 18-20 indicate that this is a type from memory command. When this command is sensed by the central processor, control of the system is given to the console to automatically type from memory without cycling. This instruction must be programed at any time it is desired to automatically type with the console. Internal processing will resume upon sensing an end alpha code.

The symbolic address (06D in the example above) will always identify the "C" memory location which is to be active with this command. The symbolic address specified by the command must have been identified earlier as a label and therefore will represent a specific memory location.

Before the TFM instruction may be executed, a load control register instruction must be programed as discussed in the other instructions section. (See Page 2-25.)

The symbolic address specified by the TFM command may also be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

C. Data Movement Instructions

SHIFT – The function of this command is to move the contents of 00K and/or 00P or 00B to the left or right without changing the sign of the memory location or locations to be shifted.

The format of the shift commands are:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	21300				SRT	S	02	00	B			SHIFT RIGHT B
B	21350				SRT	D	02	00	B			SHIFT RIGHT K TØ B
C												
D	21450				SRT	S	03	00	P			SHIFT RIGHT P
E	21500				SRT	D	03	00	P			SHIFT RIGHT K TØ P
F												
G	21600				SLT	S	02	00	B			SHIFT LEFT B
H	21650				SLT	D	02	00	B			SHIFT LEFT B TØ K
J												
K	21750				SLT	S	03	00	P			SHIFT LEFT P
L	21800				SLT	D	03	00	P			SHIFT LEFT P TØ K
M												
N												

The alpha character ("S") in column 18 when combined with "RT" or "LT" in columns 19 and 20 indicates that this is a shift command. If the alpha characters "RT" appear in columns 19 and 20 then it is a shift right command. If "LT" is in columns 19 and 20 then it is a shift left command. A shift command will always affect 00B or 00P; the symbolic address in columns 26-28 indicate whether 00B or 00P is active with the shift command. The modifier column (cc 21) is used to indicate whether or not 00K is also active. If an "S" appears in column 21, then it is a single word shift and only the symbolic address will be affected. If a "D" is in column 21, then it is a dual word shift and 00K is active in addition to the symbolic address. The length or number of places that the data is to be shifted is indicated in columns 23 and 24. The minimum shift is one and the maximum is twelve.

Data that is shifted from one memory location to another will always take the sign of the location it was shifted into.

Note that 00B and 00P are the only symbolic addresses that may be specified by the shift command. The symbolic address may not be modified with a word increment in columns 30 and 31.

Any data shifted to the right of column one of 00P or 00B will be lost; data shifted to the left of column twelve of 00K will also be lost. On single word shifts, any data shifted to the left of column twelve of 00P or 00B will be lost.

SHIFT AND ROUND – The function of this command is to shift and round to the nearest cent. A shift right must be programed in order to round.

The format of the shift and round commands are:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	21850				SRR	S	02	00	B			SHIFT B RIGHT AND RØUND
B	21900				SRR	D	02	00	B			SHIFT RT K TØ B RØUND
C												
D	22000				SRR	S	02	00	P			SHIFT P RIGHT AND RØUND
E	22050				SRR	D	02	00	P			SHIFT RT K TØ P RØUND
F												
G												
H												
J												
K												

The alpha character ("S") in column 18 when combined with "RR" in columns 19 and 20 indicates that this is a shift right and round command. The symbolic address in columns 26-28 identifies whether 00P or 00B is active with this command. An "S" or "D" is entered in column 21 to indicate a single or dual word shift; if it is a dual word shift 00K will be shifted into the location specified by the symbolic address. If it is a single word shift 00K is not affected. The number of places to be shifted is indicated in the length field (cc 23-24).

All other conditions as explained under the shift command apply to the shift and round instruction.

TRANSFER – The function of this basic command is to move the data located in the first memory location specified in the command to the second memory location. The contents of the first memory location will remain unchanged while the prior content of the second location will be cleared and replaced with that of the first.

The format of the transfer commands are:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	22100				TCP			05	F			MØVE C TØ P
B	22150				TPC			05	G			MØVE P TØ C
C	22200				TCK			05	H			MØVE C TØ K
D	22250				TKC			05	J			MØVE K TØ C
E												
F												

TRANSFER

The alpha character ("T") in column 18 indicates that this is a transfer command. The alpha character (C, P or K) in column 19 indicates the first memory location specified in the command. Therefore it will be transferred to the second memory location, specified by the alpha character (P, C, or K) in column 20. The contents of the first memory location will remain unchanged; the prior contents of the second will be cleared and replaced with the contents of the first.

The symbolic address in columns 26-28 will always identify the "C" memory location whether it is the first or second location in the instruction.

The symbolic address in columns 26-28 must have been identified earlier as a label and therefore will represent a specific memory location. This symbolic address may be modified with a two-digit word increment (cc 30-31) as explained with the add instruction.

If TCP or TPC is programed with a symbolic address of 00P; or if TCK or TKC is programed with a symbolic address of 00K the respective contents of 00P or 00K will be changed to .00+.

TRANSFER SPECIFIED COLUMNS AND TRANSFER REVERSE – The function of these special commands is to move data as explained with the basic transfer commands. However, transfer specified columns will move only the six least significant digits to the second memory location specified by the command and transfer reverse will reverse the sign of the data as it is moved to the second location.

The format of these commands are:

LINE	SEQUENCE NO.	PRGPT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	22300				TCP	S		05	K			MØVE SPEC CØL ØF C TØ P
B	22350				TCP	R		05	L			MØVE REVERSE C TØ P
C	22400				TPC	S		05	M			MØVE SPEC CØL ØF P TØ C
D	22450				TPC	R		05	N			MØVE REVERSE P TØ C
E												
F												

The alpha character in the modifier column (cc 21) will indicate whether this is a transfer specified or transfer reverse instruction. If an "S" is coded in column 21 then it is a transfer specified command. As in the above example, sequence 22300, the six least significant digits of 05K will be transferred to 00P. If an "R" is coded in column 21, then it is a transfer reverse command. As in the example, sequence 22450, the contents of 00P will be transferred with a reverse sign to 05N. (For further understanding review the discussion on simple transfer.)

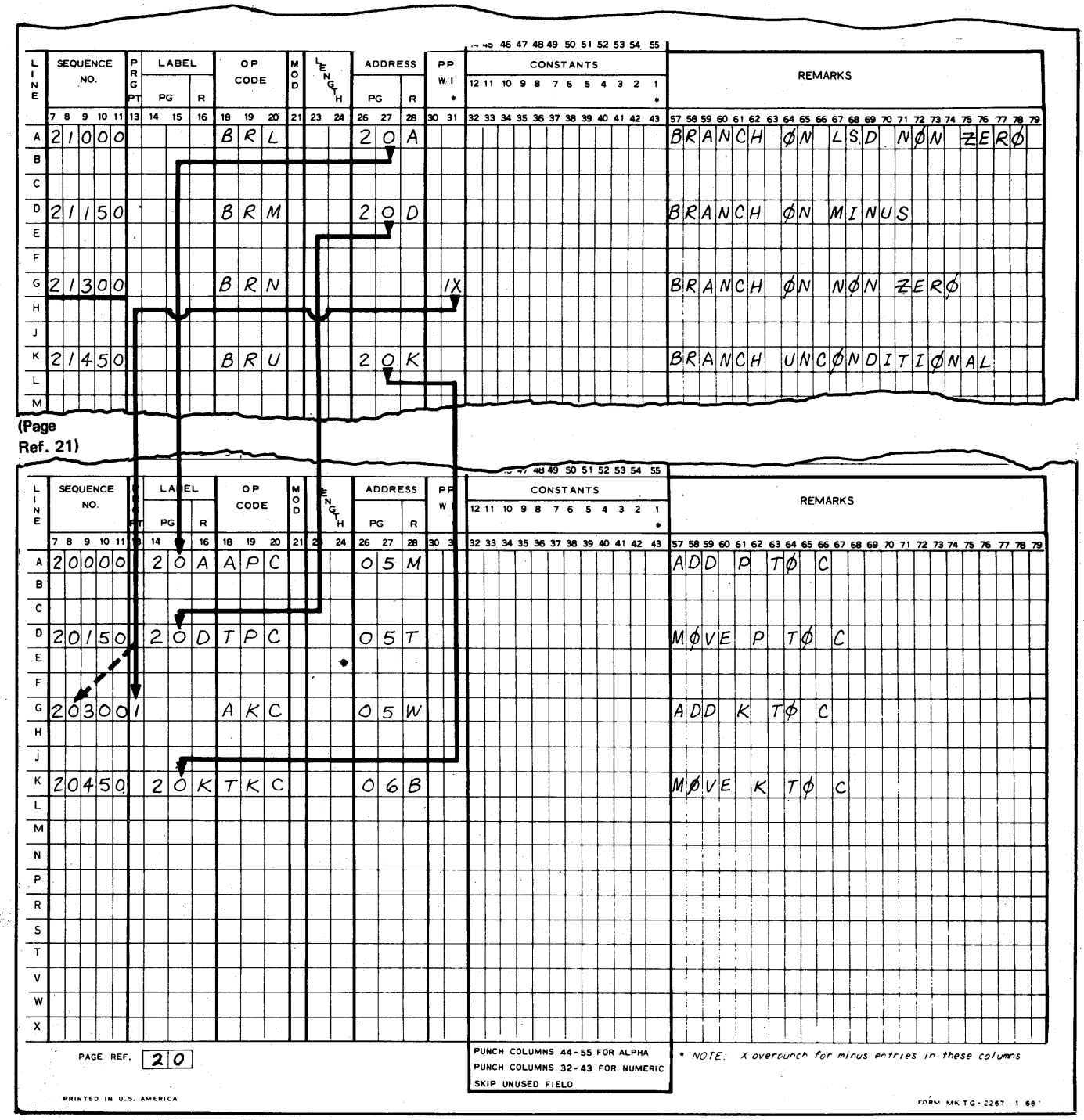
If TCP or TPC with reverse sign is programed with a symbolic address of 00P the contents of 00P will not be changed, but the sign will always be set to minus.

Notice that TCK or TKC with a modifier of specified columns or reverse sign may not be programed. However, "C" may have as its symbolic address 00K with either of the above commands.

D. Decision Commands

BRANCH – There are four branch commands; three of which are conditional. The remaining branch command is branch unconditional.

The format of the branch commands are:



BRANCH

The alpha characters ("BR") in columns 18 and 19 indicate that this is a branch command. The character (L, M, N or U) in column 20 indicates what type of branch it is. The command BRL will branch if there is anything other than zero in the least significant digit position of the memory location being tested. BRM will branch if the memory location being tested has a minus sign. BRN will branch if the location being tested has a significant value whether it be plus or minus. BRU will always, or unconditionally, branch. These branch instructions, when programed and active, are used to step the processor to an alternate program location rather than let it proceed in its normal sequence.

The alternate program position to be selected by the branch command may be coded in the symbolic address field, columns 26-28. The symbolic address shown, if any, must have been identified earlier as a label. In the example above, if the BRL instruction in sequence No. 21000 is taken the program would step to the command found on Page 20 Line A and execute it as its next instruction. Note: The BRL instruction may not be used to branch to any location above MA 199, therefore, it will not be used with a symbolic program.

The other way of indicating an alternate program position to be selected by any one of the branch commands is to code a "Program Point" of 1-9± in column 31. The program point entered in column 31 must have been assigned earlier as a plus figure in column 13. The program point is unlike the label when assigned in that it defines a position in the program but not a specific memory location. Therefore the same program point may be reused later in the program to define still another position in the program.

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	31000				BRN					1		BRANCH ØN NØN ZERØ
B	31050				BRU					3		BRANCH UNCØNDITIØNAL
C	31100	2			TCK			05	P			MØVE C TØ K
D	31150	1			AKC			05	W			ADD K TØ C
E	31200				BRM					1		BRANCH ØN MINUS
F	31250				BRU					2		BRANCH UNCØNDITIØNAL
G	31300	1			APC			05	W			ADD P TØ C
H	31350				BRM					2X		BRANCH ØN MINUS
J	31400	2			TCP	R		05	W			MØVE REVERSE C TØ P
K	31450	3			ACP			05	W	09		ADD C TØ P PLUS NINE
L												
M												

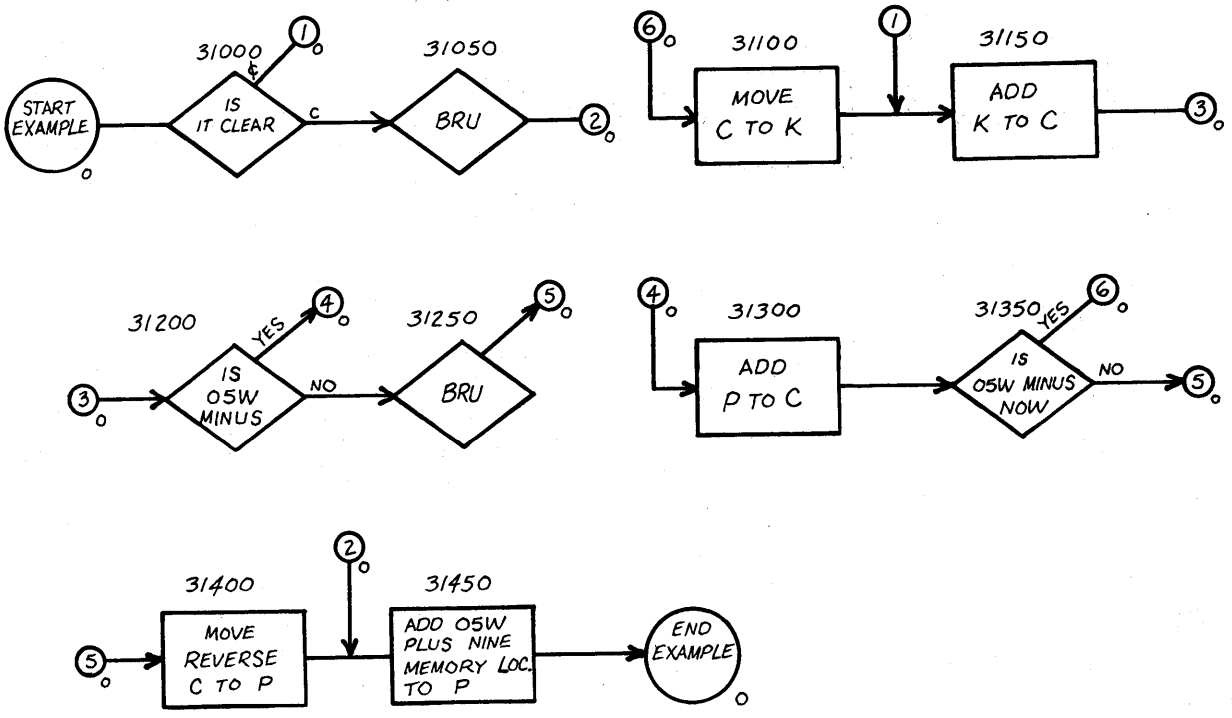


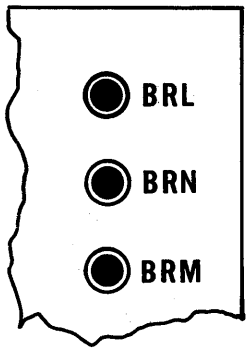
Fig. 2-1 Sample Flow Chart

From Fig. 2-1, it can be seen that the BRN instruction at sequence 31000 with a program point of plus one will branch to the command located at sequence 31150. In other words, when a plus program point is used with a branch command the processor will step to the next instruction in ascending order which has been assigned that program point. Notice that another plus one program point is utilized to step from sequence 31200 to 31300.

In the case where a minus program point is used, the processor will step to the next instruction in descending order which has that same program point assigned as a plus number. This technique was utilized to step from sequence 31350 to 31100.

If a program point is assigned to an instruction, a label may not be used! An assigned program point may only be used by the branch instruction, and if it is used a symbolic address in columns 26-28 may not be assigned! Also notice that the program points are not preceded with a zero in column 30. If it were coded with a preceding cipher, a possible error would occur when assembling. However, a word increment must have both columns coded as seen in sequence 31450.

There are four types of commands which may be used to set for status prior to a conditional branch command. They are the add, subtract, basic transfer and set status commands. Any time one of these commands is executed the status switches are set to on or off. The "on" setting is indicated in the processor by turning on the respective light or lights as pictured to the right. If the light is off, the status switch is not set and that particular conditional branch, if programed, would have no affect. The status switches are set by the sign and contents of the second memory location specified by an add, subtract or simple transfer command.



The following commands may not be programed just prior to a conditional branch.

A. CLEAR – CLR
B. DIVIDE – DBC, DPC
C. MULTIPLY – MCB, MCP
D. HALT – HLT
E. PRINT AND CYCLE – PAC
F. SHIFT – Both single and dual to the right or left.

The above commands will garble the status indicators and thereby render them useless until another add, subtract or simple transfer command is executed. All other commands not mentioned will not affect the status indicators; they will remain set as they were from the prior command.

SUBROUTINE JUMP AND SUBROUTINE RETURN – Are very much like the branch unconditional instruction. Both will cause the processor to step to an alternate program position rather than allow it to automatically perform the next instruction in sequence.

The format for subroutine jump and return is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	32000				JMP			32	J			SUBRØUTINE JUMP
B	32050				APC			05	D			ADD P TØ C
C												
D												
E	32200				JMP			32	J			SUBRØUTINE JUMP
F	32250				TPC			05	E			MØVE P TØ C
G												
H												
J	32400		32	J								
K	32450											
L	32500											
M	32550				RTN							RETURN FRØM SUBRØUTINE
N												
P												
R												
S												

The alpha characters ("JMP") in columns 18-20 indicate that this is a subroutine jump instruction. Its function, like the BRU command, is to cause the processor to step to an alternate program position which can be identified by either a symbolic address in columns 26-28 or a program point.

Unlike the BRU command, the JMP command will set the control register to the number of the memory location in which the JMP command was found. It will also cause the next command in sequence to be assembled as the first instruction in the next memory location allotted for program use.

The alpha characters ("RTN") in columns 18-20 indicate that this is a subroutine return instruction. It also has as its function to cause the processor to step to an alternate program position for its next instruction rather than let it execute the next command in sequence. The alternate program position selected by the "RTN" command is determined by the setting in the control register plus one memory location.

In the example above; the "JMP" instruction at sequence 32000 is executed. The processor would look for its next instruction at sequence 32400. It would also have set in the control register the number of the memory location which it was in when it executed the JMP command. It would then continue to process until it found the "RTN" command at sequence number 32550. Then it would step to the alternate program position determined by the setting in the control register plus one memory location; therefore it would return to sequence number 32050.

The advantage of this programing technique is that the programmer may take subroutine jumps from different areas of the program to a common subroutine and then return to the proper point in the mainline program without changing the "RTN" command in any way. This is shown in the example, when the JMP instruction at 32200 is executed the processor will step to sequence 32400 as in the prior example. However, when the "RTN" command is executed this time it will return to the instruction at 32250.

The last command executed in a subroutine entered with a "JMP" command will be "RTN"! Since the "RTN" instruction must use the control register to determine the point to which it will return, it is not possible to cycle the console while in a subroutine. This is because the control register must be loaded with commands prior to the cycle. Therefore, if it is loaded while in the subroutine the setting made by the "JMP" instruction would be destroyed and it would be impossible to return to the proper location. If it was loaded prior to the "JMP" command, the command structure set in the control register would be destroyed with the "JMP" instruction and the console could not cycle properly.

PROGRAM FLAGS – There are two commands used to test program flags. SFL will test and set any or all flags, and RFL will test and reset any or all flags. There are two basic functions of these commands. First, to determine which motor bar was active at the prior position so that the program may step to the proper routine. Second, to set up proper carriage travel by indicating the motor bar to be active at the next control console cycle.

The format of these commands are:

PROGRAMMER

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	33000				RFL		07					RESET FLAGS TØ ØFF
B	33050				HLT			00P			INDEX REF NØ	B2 FØR DR B1 FØR CR
C	33100				SFL		11					TEST FLAGS ØNE AND FØUR
D	33150				BRU			34F				BRANCH TØ CR RØUTINE
E	33200				LCR			01G				BEGIN DR RØUTINE
F												

Let us first examine sequence No. 33000 in the above example. The alpha characters ("RFL") in columns 18-20 indicate that this is a test and reset instruction. The flag or flags which are to be tested and reset is determined by the two-digit numeric entry in the length field. The flags to be tested are entered in column 23, the flags to be reset are entered in column 24. Since flag settings represent, in effect, motor bars to the central processor the command "RFL" is really testing what motor bar was active at a prior position and then resetting other motor bars to an off, or non-active, condition.

The numeric values entered in columns 23 or 24 represent the following motor bar, or motor bars, and conditions:

* 0 = No motor bar is tested or set/reset.

1 = Motor Bar 1 is tested or set/reset.

2 = Motor Bar 3 is tested or set/reset.

3 = Motor Bars 1 & 3 are tested or set/reset.

4 = Motor Bar 4 is tested or set/reset.

5 = Motor Bars 1 & 4 are tested or set/reset.

6 = Motor Bars 3 & 4 are tested or set/reset.

7 = Motor Bars 1, 3 & 4 are tested or set/reset.

8 = Can be used to set or reset lane three movement only except under special conditions as explained later.

9 = Can be used to set Motor Bar 1 with lane three movement.

* If no motor bar (Code 0) is set, Motor Bar two is automatically active.

In sequence number 33000 of the example, the RFL 07 instruction will test for no motor bar and then reset the flags for Bars 1, 3 and 4. This is normally programed at the beginning of a routine to reset (turn off) any program flags that may have been set as a result of executing that or another routine. In order to reset flag eight another command must be used.

In sequence number 33100 of the example, an operation code of "SFL" is entered in column 18-20. This indicates a test and set flag instruction. The flag or flags which are to be tested and set is determined by the entry in the length field. The flags to be tested, as with the "RFL" command, are entered in column 23, the flag or flags which are to be set are entered in column 24. Again, the central processor is really testing what motor bar was active at a prior position and then setting a motor bar to be active later in the program.

If a program flag has been set from a prior command and is now tested with either the "RFL" or "SFL" instruction, the next instruction following in sequence will be executed. If the flag being tested is not set, the processor will skip the next instruction. These commands then become a type of branch command. A practical example of the use of the "SFL" instruction is shown with sequence number 33050 of the example. In this case, assume Bar one had been depressed at the reference position and that flag 1 had been set, because of SMB in the control register. The instruction at sequence 33100 would first test for flag 1 and find that flag 1 was set; therefore the next instruction, which is a branch to the credit routine, would be executed. Had Bar two been depressed, flag 1 would not have been set and then the "BRU" command would have been ignored. The processor would have therefore gone into the debit routine. Whether or not flag 1 had been set, the instruction SFL 11 at sequence 33100 would have set flag 1 for future use. The depression of Bar one would have caused the carriage of the console to go to the credit position while Bar two would have gone to the debit position. If more than one flag is to be tested, as with SFL 50, both flags 1 and 4 must be set or the next instruction will be ignored.

SPECIAL FLAG

There is one additional test instruction which has special results and is programed only at certain times. The format of this instruction and its use is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	34000				LCR			01	H			RDL ALN P03
B	34050				PAC			00	P		INSERT LED	INSERT LEDGER AND CYCLE
C	34100				SFL		80					TEST FØR ERRØR
D	34150				BRU			35	P			BR TØ READ ERRØR
E	34200				BRU			35	V			BR TØ FILLED SHEET
F	34250											
G												
H	34350				LCR			01	J			WRL P04
J	34400				PAC			00	W			WRITE LEDGER
K	34450				SFL		80					TEST FØR ERRØR
L	34500				NØP							NØ ØPERATIØN
M	34550				BRU			36	C			BR TØ WRITE ERRØR
N	34600											
P												
R												
S												

The "SFL 80" instruction (Sequence 34100 and 34450 above) is programed just after the read and write ledger function takes place. When a ledger card is inserted into the system, an indicator would be set if a read error is present, another indicator would be set on a filled sheet condition. If the read error does occur, the "SFL 80" instruction at 34100 will cause the system to halt. By depressing the read light the error setting is turned off and the ledger is ejected for a second attempt. If the "PGA" (program advance) switch is depressed followed by a depression of the read light, the next instruction will be executed and the ledger will be ejected. Therefore, the command following the "SFL 80" instruction should branch to the read error routine. If the filled sheet condition is present and the PGA switch and then filled sheet light are depressed, causing the ledger to be ejected, the "SFL 80" command will skip one instruction and execute the next. That instruction should be a branch to the filled sheet routine.* If no error condition is set, the "SFL 80" will cause the processor to skip the next two instructions which would allow the mainline program to continue. As shown in the example starting with sequence 34000, the control register is loaded to permit insertion of a striped ledger. At 34050 the ledger is inserted. At 34100 our test for an error is made. At 34150, we would branch to a read error routine only if the read error condition was present and the PGA and read lights depressed. If the filled sheet condition was present and the PGA and filled sheet switches were depressed, we would branch from the instruction at 34200. If no error condition was present, the processor would automatically continue the mainline program beginning at 34250.

An "SFL 80" instruction following the write ledger operation will also cause a halt if the write error condition is present. If the "PGA" switch and then write light are depressed, the processor will skip one instruction and execute the next. The command following the "SFL 80" instruction, when used after write ledger, should be a "NOP" as it will never be executed. The "NOP" also causes the next command (branch to write error routine) to be properly located. If a write error condition is not present, the next two instructions would be skipped thus allowing the mainline program to continue.

When the "SFL 80" instruction is executed with the depression of the "PGA" switch, any error indicator which may be set is turned off or reset.

It should be noted that SFL 80 can not be used to test "Flag" 8; it can only be used to test for striped ledger error routines.

* With the depression of the filled sheet light Flag 2 will be set automatically. This may be used to assist in filled sheet programing and should be reset during the routine.

Other Programing Uses of Special Flag (SFL 80 and RFL 80).

Normal use of Flag 80 is in conjunction with Read Error, Write Error and Filled Sheet (for Filled Sheet see page 2-60) error conditions. Other programing of this instruction is possible as is illustrated in the following example:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	22000				LCR			01	H			
B	22050				SFL		41					
C	22100				RFL		84					
D	22150				RFL		01					
E	22200				HLT			00	P			TØ MANUALLY TYPE
F	22250				SFL		10					
G	22300				PAC			00	W			NØ TYPING
H												

The instruction on line "B" will set Flag 1 and if Flag 4 was previously set, will execute the next instruction, however, if not previously set, skip the next instruction. The RFL 84 instruction on line "C" will reset Flag 4, always skip the next two instructions and execute the instructions on lines "F" and "G". If Flag 4 was not previously set on line "B", the instructions on lines "D", "E" and "F" will be executed and the instruction on line "G" will be skipped.

NOTE: The Series E 8000 can utilize the flag associated with motor bar 4 for a programed console operation and override a carriage open instruction in the control unit. This will permit the use of a stop programed with carriage open for other purposes.

This function can be changed by programing an LN4(DS) or LN5(DSR) when operating at the carriage open stop.

SET STATUS – The function of this command is to set the Status Indicators on word 00P or 00K without affecting the contents of either.

The format of the two set status commands is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
A	35000				SSP							SET STATUS ØN WØRD 00P
B												
C	35100				SSK							SET STATUS ØN WØRD 00K
D												
E												
F												
G	35300				MCP			05K				MULT C BY P
H	35350				SRR	D	02	00P				POSITIØN PRØDUCT
J	35400				SSP							SET STATUS ØN WØRD 00P
K	35450				BRM			36F				BRANCH IF REV ENTRY
L												
M												
N												

The alpha characters "SSP" in columns 18-20 indicate this is a set status on 00P instruction. "SSK" is set status on 00K. Notice that there is no address associated with these instructions.

As mentioned earlier, a conditional branch command cannot be programed just after any clear, divide, multiply, halt, print and cycle, or shift command as these commands garble the status indicators. However, it is often desired to select alternate program routines depending upon the result of one of these commands. If this is the case, a set status instruction may be programed on word 00P or 00K without changing the sign or contents of that word.

An example of a use for this command is shown at sequence 35300. A multiplication is made and the product is positioned. Assume that it is now desired to branch to an alternate program position if the result of that multiplication was minus. Since the shift instruction garbles the status indicators, a set status instruction on word "00P" (the location of the product) must be programed before the branch can be taken. As indicated earlier, the sign and contents of the product in "00P" will not be affected.

PUNCHED CARD IN

E. Other Instructions

PUNCHED CARD IN – The function of this command is to feed one punched card through the card reader.

The format of this command is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	36000				CRD			08	B			FEED ONE PUNCHED CARD
B												

The central processor of the E 8000 can function with either the A 592 or A 594 card reader. Programing for punched card in, on the E 8000 will be discussed only as it is different from programing on the E 4000.

The alpha characters "CRD" in columns 18-20 indicate that this is a punched card in instruction. The symbolic address "08B" is coded to identify the location in memory that the card is to be read into.

The E 8000 has the ability to utilize two card readers. If a second reader is part of the system, programing will specify the units as Card Reader One and Card Reader Two.

With two card readers, the coding to indicate the alternate readers will be a "01" or "02" in columns 30 and 31. A symbolic address is coded in columns 26-28 to identify the location in memory that the cards are to be read into. Note the following.

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO.	OP CODE	ADDRESS PG	ADDRESS R	PP W/I *	REMARKS
	7 8 9 10 11	18 19 20	26 27	28	30 31	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
A	23000	CRD	08	B	01	READ A CARD FROM CARD
B	23050					READER ONE
C	23100	CRD	07	B	02	READ A CARD FROM CARD
D	23150					READER TWO
E						
F						
G						
H						

If the program calls for Card Reader Two, and the system has the capability of two card readers but only one is on the system, a block against further operation will result. This call for a card must be satisfied before the program will proceed. If the capability for two card readers is not present, the instruction is treated as if it were a Card Reader No. 1 instruction.

The second card reader must be of the same type as the first. A system cannot have both an A 592 and an A 594. If the system uses the A 594, each card reader will have its own format. They do not have to be the same.

The E 8000 is equipped to work with the A 594 card reader as standard. The E 8000 system can read cards in two modes:

1. Program Load Mode
2. Single Card Read, from Internal Instruction. (See example above.)

In order to read a card in the single card read mode, the "Format Register" of the A 594 reader must be programed as explained later. The function of the Format Register is to:

1. Provide field designation for cards read.
2. Provide start Alpha and end Alpha designation.
3. To designate those card columns to be ignored.

Under the program load mode, the card reader Format Register is ignored and all controls for card formatting are obtained from the card being read. However, the Format Register is set to a non-programed condition and must be reprogramed before a single card read from internal instruction can be performed. This mode permits the continuous feeding and reading of cards until an "end card in" code is read. Data read from the punched card, or cards, is stored in memory sequentially starting at the C Address "Selected". A "12" zone punch steps memory to the next higher location, a "12-0" punch or a "12" punch by itself will clear the active address and step memory to the next higher location. Alphabetic data cannot be read in under this mode, except by having punched the numeric digits in a manner that will store them into corresponding words of two decades. Also under this mode, data which is read in cannot be processed until the last card to be read has passed through the reader.

To Load Program or Data under the Program Load Mode:

1. Turn on system or
   (a) If system is on, depress any one of the manual control keys (*, •, +, or -) followed by the depression of the program advance switch light. This will cause the Program Counter to restore to 000 and internal processing will cease.
2. Select the memory location from the console keyboard to which the data is to be read.
3. Depress the Program Load Switch Light.
4. Depress a Motor Bar.

PUNCHED CARD IN

Cards will be fed and read until an end read code (1/3) followed by an end card in code (1/4) is read. This terminates card feeding and resets the program counter to zero. An end read punch (1/3) must be in all cards except the last card which requires an end read code followed by an end card in code (1/4) prior to card column 30.

The Format Register must be programed each time that power is turned on to the system, and/or following an E 8000 "Program Load Mode" operation. The Format Register is programed with a punched card. Therefore, the first card fed following either of these conditions must be a Format Register card. If the first card is not a Format Register card, the card read instruction is not completed, the card reader is blocked preventing further card feeding and the "prog" lamp on the reader is illuminated. This condition may be corrected by restacking the correct Format Register card and the Data card fed in error in the reader and depressing the switch associated with the program lamp.

The Card Format Register keeps in step, column by column, with the punched card in the read station of the A 594 card reader during the reading of all 80 columns of the punched cards. Each column of the punched card is read and transferred to memory in accordance with the functional codes stored in the Card Format Register. Consequently, a blank card column will be read and generate a code that is transferred to memory; in the numeric mode as a "0" and in the alpha mode as a space code.

The Card Format Register Functional Codes are:

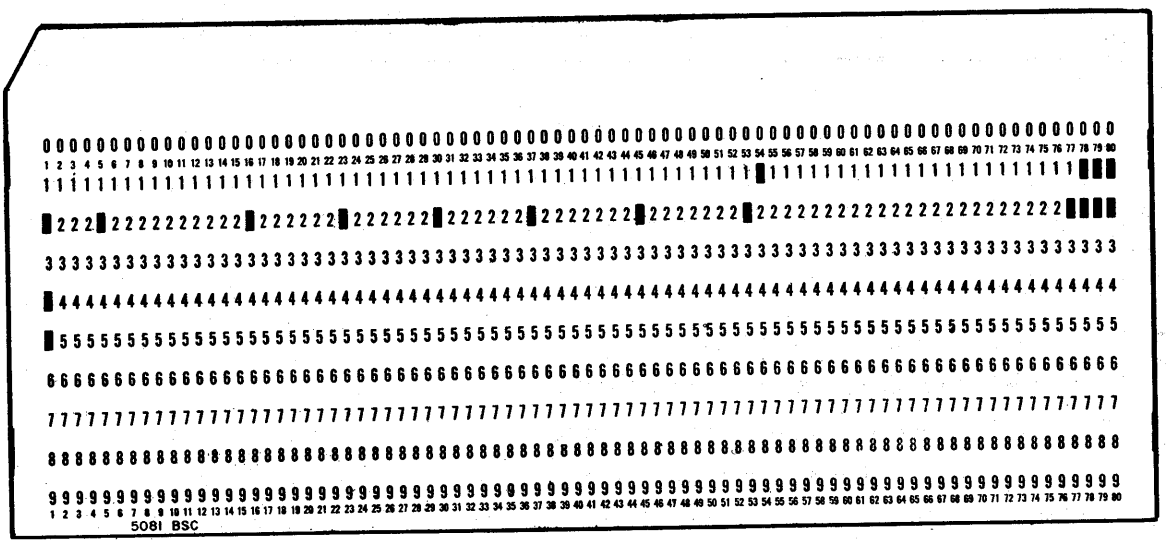


Fig. 2-2 Program Format Card

Card Column 1 – "2" punch indicates "End of Word"; "4/5" punches designate this card as a Format card.

Card Columns 5, 16, 23, 30, 37, 45, and 53 – "2" punches designate "End of Word".

Card Column 54 – "1" punch designates start alpha.

Card Column 77 – "2" punch indicates end of alpha and end of word.

Card Columns 78, 79, and 80 – "1/2" punches designate ignore these columns.

PUNCHED CARD IN

When this card is read from the single internal command (CRD) it will automatically load the Format Register because of the 4/5 codes in column one; the next card will automatically be fed and read. A sample Data Card that may be read appears as follows:

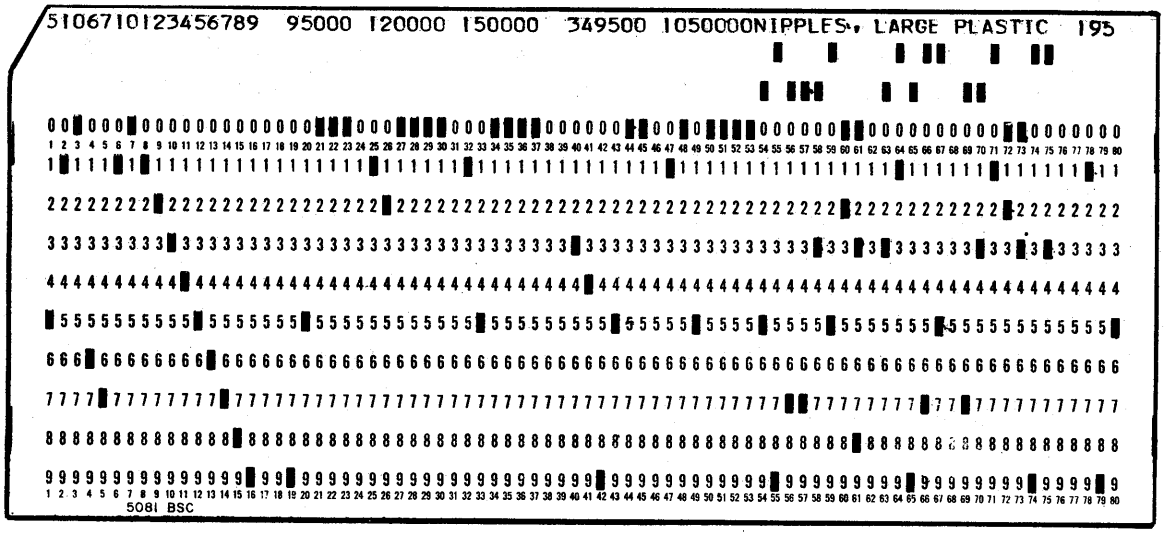


Fig. 2-3 Data Card

Assuming that the above Format and Data Card are fed and read by a command such as that shown in sequence No. 36000 on Page 2-20. The information on the Data Card would have gone to the following locations:

1. Card No. to 08B.
2. Date to 08B plus one memory location.
3. Product No. to 08B plus two memory locations.
4. Disc. No. 1 to 08B plus three memory locations.
5. Disc. No. 2 to 08B plus four memory locations.
6. Disc. No. 3 to 08B plus five memory locations.
7. Cost price unit to 08B plus six memory locations.
8. Sales prices unit to 08B plus seven memory locations.
9. Product Description to 08B plus eight and nine memory locations for low orders; 08B plus eighteen and nineteen for high order codes.

PUNCHED CARD IN

It may, at times, be desired to read punched cards with the A 594 reader that were originally designed to be used with the A 592 reader. Such a card is shown in Figure 2-4.

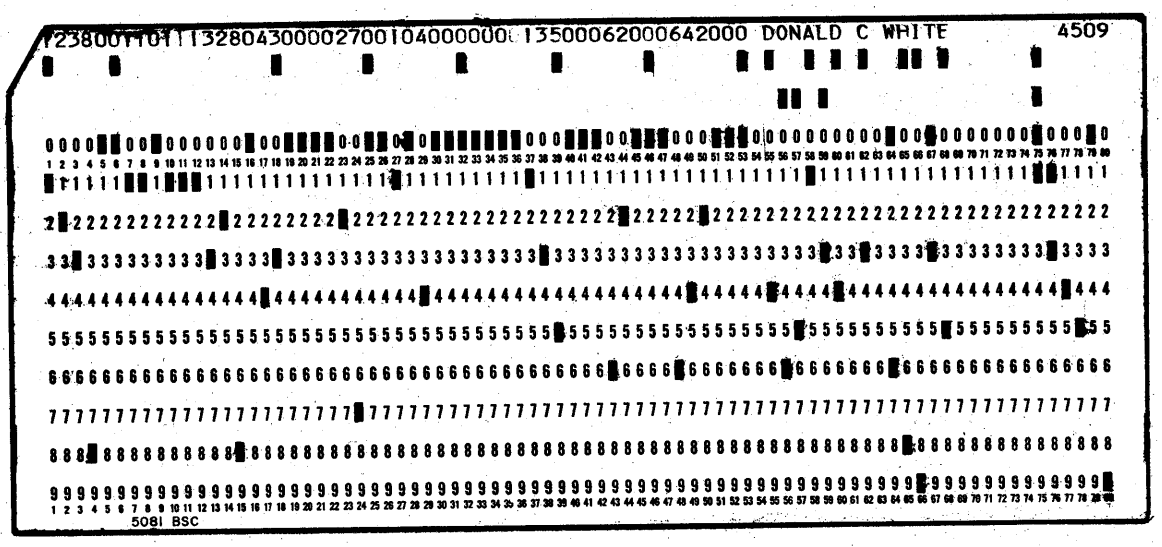
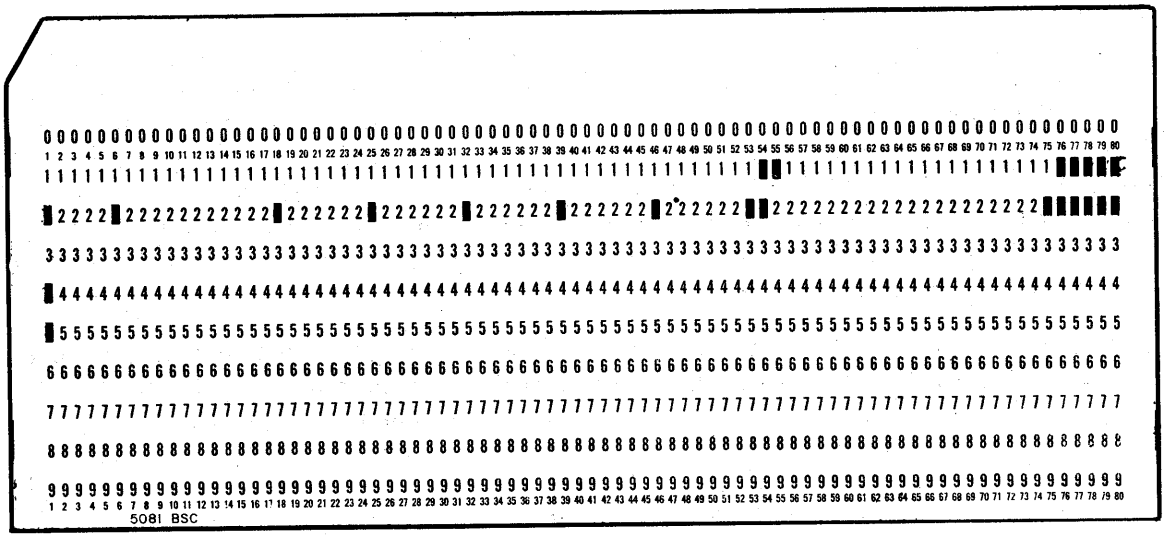


Fig. 2-4 Data Card Designed for A 592 Reader

Notice that this card has all of the necessary function codes that must be used with the A 592 reader. These codes can not be used with the A 594 reader on the E 8000! In order to read this card with the A 594, the following Format card would be used:



CARD COLUMN 1 – The "2" punch designates end of word, the same function as the "12" punch on the Data Card. The "4/5" punches designate this card as a Format.

CARD COLUMNS 6, 18, 25, 32, 39, 46 AND 53 – "2" punches designate end of word, the same as the "12" punches did for the A 592.

CARD COLUMN 54 – "1/2" designates ignore. (This column on the Data Card was used to indicate start Alpha for the A 592 and is not needed when using the A 594.)

CARD COLUMN 55 – "1" punch designates start Alpha. (Notice that it is in the same column as the first Alpha character.)

CARD COLUMN 75 – "2" punch designates end of word and end of Alpha. (The "12/11/0/2" punches in the Data Card, Typewriter Control Key One, will be read to memory but the end Alpha is sensed by the "2" in the Format Register of the A 594 Card Reader.)

CARD COLUMN 76 – "1/2" punches designate ignore this column. (This column on the Data Card was used to indicate end card read to the A 592, but is not needed when using the A 594.)

CARD COLUMNS 77, 78, 79, AND 80 – "1/2" punches designate ignore these columns of the Data Card. (Identification No.)

NOTE: Any field to be read as minus would have had an "11" zone punch in the Data Card.

As mentioned earlier, blank card columns on a punched card will, when read, transmit data to memory. If a card such as that shown in Figure 2-4 is read, the Alpha name would be followed by seven space codes prior to the end Alpha code. (Remember the A 592 does not recognize blank card columns) when that name is typed from memory by the console of the E 8000 it would not be desirable to type the name and then go through seven space operations prior to ending Alpha. To prevent this from occurring the type from memory mode of the E 8000 has been modified to permit only one spacing operation when a number of space codes are stored in adjacent positions of the memory location from which we are typing. (If it is necessary to recognize all space codes, DAP must be programed in the control register.) When a space code is sensed, one typewriter space cycle will occur, no other space cycles will occur until a code other than a space is sensed; in this case, end Alpha. When more than one space is required, "Space Escape" Codes (see DAP, Page 2-39) are required to permit escapement of the carriage of more than one tenth inch without printing. One code is needed for each space desired.

When a card is read from an internal command such as that shown in example on Page 2-20, the Data is loaded into memory starting in the zero word of the decade selected by the symbolic address. The entire contents of the card are read into memory in one pass of the card. In the internal mode, not the program load mode, read in from one card is limited to ten numeric words of memory. An end word code (a "2" in the Format Register) sensed when the units digit of the active memory address is "9", steps the memory back to the zero word of the same decade. Remember that in the Program Load Mode, the end of the word code ("12" punch in the card being read) will always step the memory address to the next higher word of memory.

Data is always read as numeric until a start Alpha code is sensed in the Format Register and reverts back to numeric when an end Alpha is sensed in the Format Register. Each numeric code requires one digit of memory, each Alpha code requires one digit in each of two companion words of memory. When reading numeric data, the first code or blank card column sensed after card read is initiated, or after an end word code, will clear the entire contents of the active memory address. When reading Alpha Data, codes are transferred to memory on a column-by-column basis. The Alpha code read replaces only the contents of the memory digit position into which it is transferred. When Alpha Data is read, the companion decade location is always one decade higher than the one selected by the symbolic address of the CRD instruction.

LOAD CONTROL REGISTER
NO OPERATION

LOAD CONTROL REGISTER – The function of this instruction is to load the Control Register with the commands needed by the E 8000 console upon cycling.

The format of the load control register instruction is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	36000				LCR			01	A			CLA RKC P01
B												
C												

The alpha characters (LCR) in columns 18-20 indicate this is a load control register command. The control register commands will be discussed later; it should be indicated at this point, however, that instructions to the console such as clear and add, read the keyboard to C and position check are stored in memory as numeric amounts. For the most part, there will be a numeric figure setup in memory for each operating position taken by the E 8000 Console. Likewise, each time the console operates, it will react to, and perform, the commands sensed in the Control Register. Therefore, it is necessary to transfer that numeric figure which contains the instructions for a particular operating position from its storage location to the Control Register prior to a given console cycle.

The "LCR" instruction will then transfer the data located in the memory location specified by the symbolic address (01A above) to the control register. This is a destructive command in that the prior contents of the control register will be cleared and replaced with the new data being moved in. The control register must be loaded with the "LCR" command at any point prior to the console cycle which is to react to, and perform, the commands in the control register. If two cycling positions require the same programing, the control register need only be loaded at some point prior to the first position.

NO OPERATION – The function of this instruction is to perform no operation and step automatically to the next instruction.

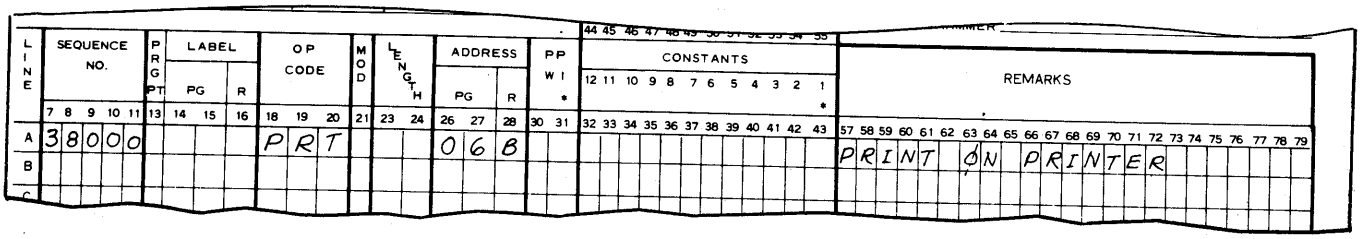
The format of this instruction is:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	37000				NØP							NØ ØPERATIØN
B												
C												
D	37150				SFL		80					TEST FØR ERRØR
E	37200				NØP							NØ ØPERATIØN
F	37250				BRU			36	C			BR TØ WRITE ERRØR
G												
H												

The alpha characters "NOP" in columns 18-20 indicate that this is a no operation instruction. There is no need to indicate a symbolic address with this command. The purpose of this instruction is to correctly position a succeeding command. An example of this is seen in the programing above beginning with sequence No. 37150. If a striped ledger had just been written, the "SFL 80" instruction would test for a write error condition. If such a condition was present, the central processor would skip one instruction and execute the next. (See discussion on program flags, Page 2-16.) In example above the "NOP" instruction was placed just after the "SFL 80" command to ensure that the "BRU" instruction would be located in the proper position.

PRINT ON LINE PRINTER – The function of this instruction is to cause the line printer to print. When programing in symbolic language it will cause the numeric data in the 40's and the alpha data in the 40's and 50's to print as directed by the mask values in the 60's.

The format of the print instruction is:



The alpha characters "PRT" in columns 18-20 indicate that this is a print instruction. The symbolic address in columns 26-28 designates the memory location from which printing is to begin. The symbolic address may be incremented by coding in columns 30 and 31, however the address must represent a memory location set up in the "Line Printer File" as discussed on Page 1-2.

As indicated in the first paragraph, the data located in the 40's and 50's will print as indicated by the mask values set up in the 60's. Therefore, before the print instruction may be programed it must first be determined:

1. What is to be printed.
2. Where it is now located.
3. How it is to be printed.
4. The proper mask values to cause printing as desired.
5. Move data and mask values to the proper locations.

To properly explain this command, let us follow the above procedures with a typical situation. Assume that a punched card is read, the data on this card is to print on the line printer. The card format is:

1. The first card field is the card code; it is one digit in length.
2. The second card field is the account number, it is a maximum of eight digits in length.
3. The third card field is the reference number and is also a maximum of eight digits in length.
4. The fourth and last card field is the amount and is a maximum of twelve digits in length.

Let us assume that the punched card was fed and read with a "CRD 08B" instruction. Therefore:

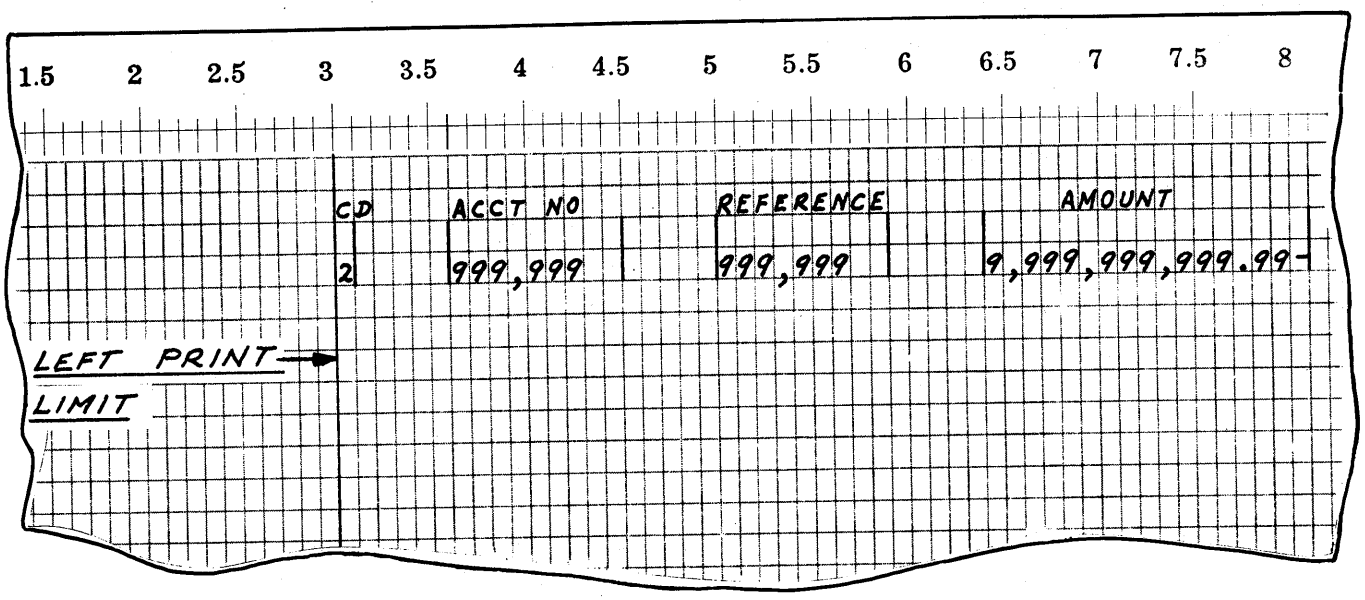
1. The card code is in 08B (column No. 1).
2. The account number is in 08B plus one memory location (columns 1-8).
3. The reference number is in 08B plus two memory locations (columns 1-8).
4. The amount is in 08B plus three memory locations (columns 1-12).

The above data is to print as follows:

1. Card code – To print from the left print limit on the line printer followed by 5 blank spaces. Then:
2. The account number followed by 5 blank spaces. Then:
3. The reference followed by 5 blank spaces. Then:
4. The amount with a sign if minus.

PRINT ON LINE PRINTER

This print format is now transferred to MKTG 2366 as shown in the following example. Notice that sample figures are used.



It has now been determined:

1. What is to be printed.
2. Where it is now located.
3. How it is to be printed.

At this point, the programmer must now determine the "mask" values necessary to cause printing as outlined. The purpose of mask values is to determine what characters will print and which will not, to insert blank spaces between columns, and to properly punctuate data being printed.

Prior to assigning the mask values for the above example, we shall discuss each mask value and its function in printing.

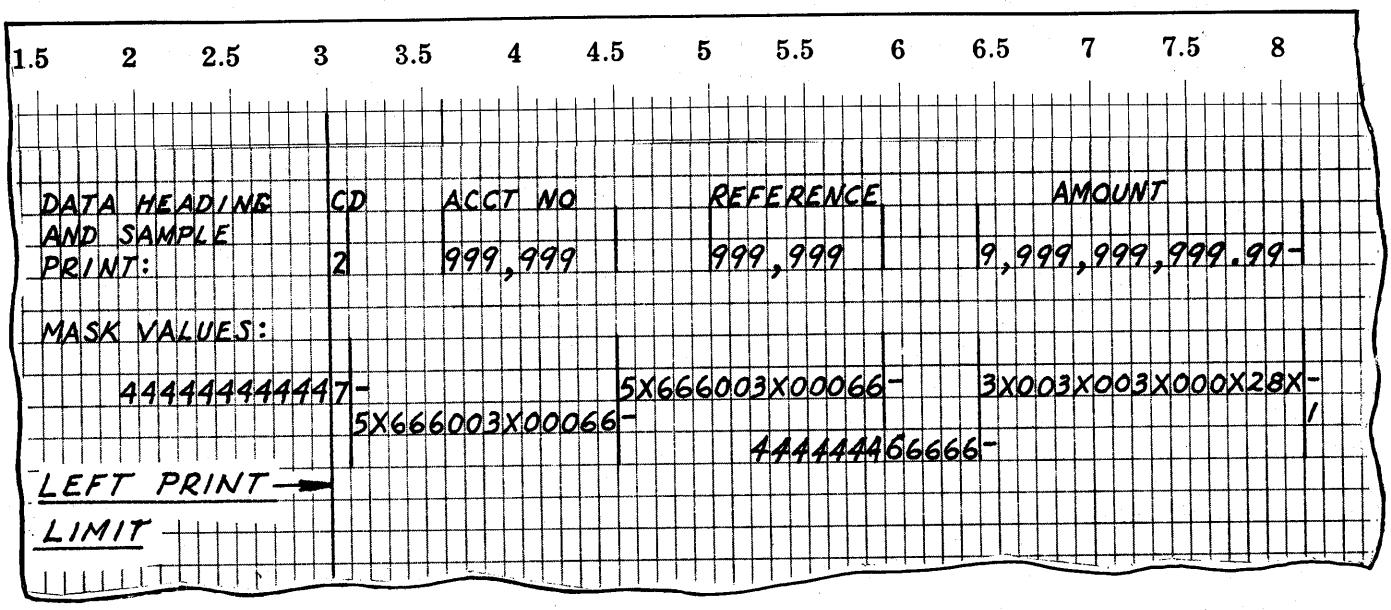
The mask constants in a mask word will be plus or minus and all mask constants within a mask word will have the sign of the mask word. In the following discussion the word significance is used. A significant digit is a digit which has value; for example, if a memory location is clear the cipher in digit position one is not significant and is usually not printed. However if a memory location has $100.00 in it, the ciphers in digit positions one through four are significant and have value, therefore they are normally printed. As a general rule, any digits 1-9 are always significant. A digit of zero is significant only if preceded by a digit of 1-9. To set significance is to indicate the beginning of a printed amount; to reset significance is to indicate the end of a printed amount.

Following is a list of mask values and their functions.

MASK SIGN AND VALUE	TO PRINT	FUNCTION OF MASK VALUE
+/-0	Digit of 0-9 or blank	Print digit and set significance if digit is significant or if significance was set on a previous digit, otherwise insert blank space.
-1	Character A-Z	Print alpha, but insert blanks when key code is reached and continue inserting blanks as long as this format mask code continues, reset significance.
+1	Blank	End of print line, print blank and reset significance.
-2	Decimal and digit 0-9	Print decimal and digit regardless of significance. Reset significance.
+2	Comma and digit 0-9	Print comma and digit regardless of significance. Reset significance.
-3	Digit of 0-9 and comma or blank, blank	Print digit and comma and set significance if digit is significant or if significance was set on previous digit; otherwise blank, blank.
+3	Digit of 0-9 and decimal or blank, blank	Print digit and decimal and set significance if digit is significant or if significance was set on previous digit; otherwise blank, blank.
+/-4		Ignore digit, do not load or shift line printer buffer. Reset significance.
-5	Digit of 0-9 and blank or blank, blank	Print digit and blank and set significance if digit is significant or if significance was set on previous digit; otherwise insert blank, blank.
+5	Digit of 0-9 or *	Print digit and set significance if digit is significant or if significance was set on previous digit, otherwise print asterisk (*).
-6	Blank	Insert blank regardless of significance, reset significance.
+6	Digit of 0-9 or $	Print $ or, if digit is significant or if significance was set on previous digit, print digit and set significance.
+/-7	Digit of 0-9	Print digit regardless of significance, reset significance.
+/-8	Digit of 0-9 and blank or digit of 0-9 and minus sign	Print digit and minus (-) sign if sign of amount is minus. Print digit and blank if sign of amount is plus. Reset significance.
+/-9	Digit of 0-9 and * or CR	Print digit and "CR" if sign of amount is minus. Print digit and "*" if sign of amount is plus. Reset significance.

PRINT ON LINE PRINTER

Once the print format has been indicated on MKTG Form 2366, as shown in the preceding example, the above mask values must be applied to permit such printing. This is done in the following example.



As stated earlier, the card code was read into memory and was located in column one of that memory location. The code is desired to print from the left print limit of the line printer, therefore the left 11 digits of that memory location must be ignored. The mask constant which will cause this is 444444444447-. The eleven left hand minus 4's loaded in the first word of the mask constants will cause the high order eleven digits in the first word of print to be ignored, those eleven digits in the first print word will not load or shift the line printer buffer and significance will be reset. The minus 7 loaded in column one of the first mask constant will cause the first digit of print word one to print whether or not it is significant. The 7- code would have caused the line printer buffer to have been loaded with one digit of 0-9.

When printing on the line printer, Style A 988-00, 120 print positions are available for printing. There are also 120 digit positions available to store mask values (10 twelve-digit mask words). Notice that the mask value used to print the code is twelve digits long (one mask word) however, the line printer buffer has only used, or been loaded with, one print character.

The next figure to be printed on the print line is the account number. It has been identified in memory as eight characters in length and is to print as a true number six characters in length with no decimal. It is also desired to have five blank spaces between the code and first possible digit of the account number.

Since the account number is eight digits in length, only one mask constant need be used to provide for both the blank spaces and the proper printing of the number. The mask constant is 5X666003X00066- (12 characters in length). Since the account No. was read from a punch card and is only 8 digits in length, the high order 4 digits in that memory location will always be zero. Therefore the minus 5 mask value in position twelve of mask constant two will cause two blanks to be shifted into the line printer buffer. Notice that "5X" has been coded on Todd Form 2142 to indicate that only one digit of mask value is used, but that two print positions are occupied. Each of the three following minus 6's will shift one blank into the print buffer, therefore the four left-hand mask values in the second mask constant will shift five blanks into the print buffer. The remaining eight mask values in the second mask word are now used to direct the printing of the account number. The two minus zeros in digit positions 7 and 8 of the mask constant will cause any significant digit in these respective positions of the print word to be shifted into the print buffer, otherwise they will cause blanks to be shifted into the print buffer. The minus three in digit position 6 of the mask word will cause a digit and a comma to print if that digit is significant, otherwise two blanks will be shifted in. Again notice that "3X" is coded to indicate that one mask value will cause two print positions to be loaded. The following zeros in digit positions 3-5 will cause printing of the significant digits remaining in the account number. The two minus 6's in digit positions one and two of the mask word will cause blanks to be inserted in place of the units and tens digit positions of the print word and significance will be reset. Notice that this mask constant is twelve characters in length, and that it will cause 14 print positions to be loaded.

The next figure to be printed is the reference number. It has the same requirements as did the account number and therefore will have the same representative mask value.

Since the next data to be printed is the amount and can be 12 characters in length, a full mask word will have to be reserved to control the printing of that amount. It is then necessary to use the fourth mask constant to provide the five blank spaces between the reference No. and the amount. The mask constant to provide these blank spaces is 444444466666- (12 characters in length). The left-hand seven minus 4's will ignore the seven most significant digit positions of print word four and the remaining five minus 6's will insert five blanks into the print buffer and reset significance. This mask constant will require 12 character positions and will load 5 print positions in the line printer buffer.

The fifth mask constant will control printing of the amount. The constant is 3X003X003X000X28X minus. The mask values "3X" will cause a digit and a comma to print if the digit is significant, otherwise two blanks will be loaded into the print buffer. The "X2" code will cause printing of the digit and a "-" sign if the value of print word five is minus.

Since this completes the print message, it is now necessary to indicate the end of the print line. This is done with a mask constant of 100000000000+. The mask value plus 1 is located at whatever point the programmer wishes to stop loading the print buffer. It will cause the print buffer to be loaded with a blank. Note: If the buffer is not full and end of line print mask is not programed, no printing will take place. Processing will continue as if printing had taken place, but the print buffer would not have been cleared and the next print instruction will continue to load the buffer. When all 120 positions of the buffer are loaded printing will take place.

When the "PRT" command is executed numeric characters, or codes in the case of alpha, which are printed from the 40's, and in the case of alpha, 50's, will be cleared up to and one digit beyond the end of print line mask code located in the sixties. Any other unused memory locations in the 40's, 50's or 60's may be used for working locations and reached by incrementing the proper label.

PRINT ON LINE PRINTER

When a line printer is to be used, and the printer file is established, as discussed on Page 1-2, the assembler will automatically select the 40's, 50's and 60's. Therefore, if a label is assigned with the first constant in the line printer file it will automatically identify the first print word location. A mask constant will direct printing of a figure 20 memory locations below it, or 10 and 20 memory locations below it in the case of alpha. Since the first memory location for printing always has a label and its mask value must be located twenty memory locations above it, the mask area can be reached by incrementing the print location label to properly load mask values, however to simplify programing a label will usually be assigned the mask area as well. Mask constants may be loaded directly into the 60's as discussed on Page 2-54. By doing this it is not necessary to move them into the 60's prior to each print instruction. In the programing example below, it is assumed that the mask constants were loaded someplace other than the 60's and therefore must be moved to the 60's prior to printing. We will assume that these constants may be reached with the label 15J as they were loaded as sequential constants.

It has now been determined:

1. What is to be printed.
2. Where it is now located.
3. How it is to be printed.
4. The proper mask values to cause printing as desired.

All that remains is to move the mask values and data to the proper location for printing. The following is an example of this.

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER _____

BRANCH _____

PROGRAMMER _____

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W'I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS PRØGRAM (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	38000				CRD			08	B			READ PUNCHED CARD
B	38050				TCP			08	B			MØVE CØDE TØ WØRK
C	38100				TPC			06	B			MØVE CØDE TØ PRINT TANK
D	38150				TCP			08	B	01		MØVE ACC NØ TØ WØRK
E	38200				TPC			06	B	01		MØVE ACC NØ TØ PRT TANK
F	38250				TCP			08	B	02		MØVE REF TØ WØRK
G	38300				TPC			06	B	02		MØVE REF TØ PRT TANK
H	38350				CLR			06	B	03		CLEAR FILLER LØCATIØN
J	38400				TCP			08	B	03		MØVE AMØUNT TØ WØRK
K	38450				TPC			06	B	04		MØVE AMØUNT TØ PRT TANK
L	38500				TCP			15	J			MØVE MSK CNST TØ WØRK
M	38550				TPC			06	C			LØAD MSK WØRD ØNE
N	38600				TCP			15	J	01		MØVE MSK CNST TØ WØRK
P	38650				TPC			06	C	01		LØAD MSK WØRD TWØ
R	38700				TPC			06	C	02		LØAD MSK WØRD THREE
S	38750				TCP			15	J	02		MØVE MSK CNST TØ WØRK
T	38800				TPC			06	C	03		LØAD MSK WØRD FØUR
V	38850				TCP			15	J	03		MØVE MSK CNST TØ WØRK
W	38900				TPC			06	C	04		LØAD MSK WØRD FIVE
X	38950				TCP			15	J	04		MØVE MSK CNST TØ WØRK

PAGE REF. 38

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1 68)

PRINT ON LINE PRINTER

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD	PAGE HEADER, ALPHA, CONTROL REGISTER
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18	1 2 3 4

CUSTOMER _____

BRANCH _____

PROGRAMMER _____

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS PROGRAM
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	39000				TPC			06C		05		LØAD MSK WØRD SIX
B	39050				PRT			06B				PRINT ØN LINE PRINTER
C												
D												

In the example shown a punched card was read with a "CRD 08B" instruction. Therefore the code was in 08B, account number in 08B plus 1, reference in 08B plus 2, and the amount in 08B plus 3. Lines B-K of Page 38 in the above example move the data read in from the punched card to the proper print locations as specified by the "PRT 06B" instruction. The filler location referred to at sequence 38350 is the memory location controlled by the mask constant which will be used to provide for the blank spaces between the reference and amount figures. From sequence 38500 on; the program moves the mask constants to the proper mask word locations (20 memory locations above the data words they are to control). Once this has been done the "PRT" command may be programed.

The data moved into the print locations (06B – 06B plus 4) will automatically be cleared with the print instruction. The mask values in 06C – 06C plus 5 will not be cleared and could be used again if the print format was not to change.

With each print instruction on the line printer the form must space. The vertical spacing on the printer is controlled by a four (4) channel tape. With each "PRT" instruction the control register will select a channel on the program tape. The tape and form in the line printer will then advance (or space) in the selected channel until a hole is sensed in that channel of the program tape. The control register commands which select a tape channel for tape advance are discussed on Page 2-39 under control register commands.

The suggested use for the various channels is:

Channel #1 – Punched to indicate selective or variable lines.

Channel #2 – To indicate 2nd to last line on form or page.

Channel #3 – To indicate first print line on form or page.

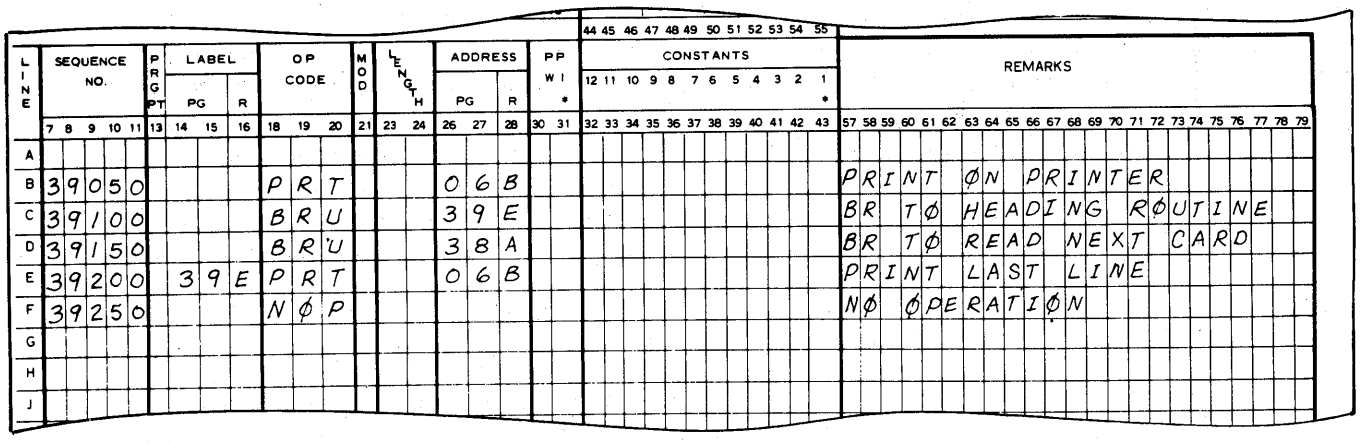
Channel #4 – To indicate normal line to line print locations.

When printing a report it is often desired to print form headings. This requires the processor to know when it is at the top of the next page. To indicate that the printer is going to print the last line on a page a program flag must have been set. One of the functions of a channel two punch on the spacing tape is to set this flag (bottom of form flag). However, to provide additional flexibility and allow variable advance not only in channel one, but also in channel two, a punch in channel two will only set the "bottom of form flag" if the print line on which the channel two punch appears was reached under channel four control.

The print instruction is actually a two-function command. It will first test for a bottom of form flag. If the bottom of form flag is not set, the print instruction will be executed and the next command will be ignored. If the bottom of form flag had been set by printing on a line which was reached under channel 4 advance and sensed a channel 2 punch, the next "PRT" instruction will be ignored. The following instruction will be executed and the bottom of form flag will be reset.

PRINT ON LINE PRINTER

Based on these facts, the following example shows typical programing which follows the "PRT" instruction.

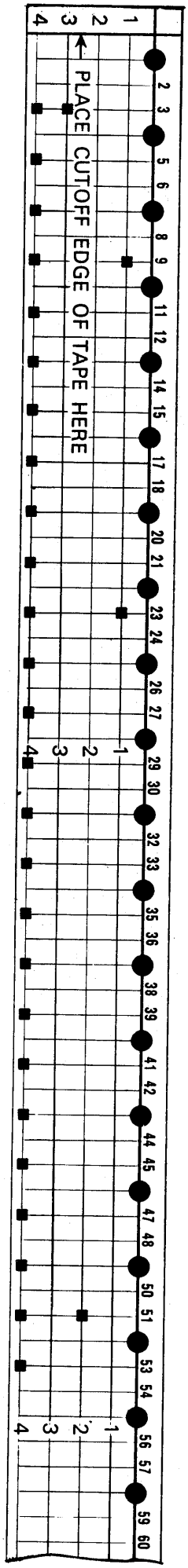


LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A												
B	39050				PRT			06	B			PRINT ØN PRINTER
C	39100				BRU			39	E			BR TØ HEADING RØUTINE
D	39150				BRU			38	A			BR TØ READ NEXT CARD
E	39200		39	E	PRT			06	B			PRINT LAST LINE
F	39250				NØP							NØ ØPERATIØN
G												
H												
J												

The normal operation of the print instruction in sequence 39050, of the example above, is to execute the "PRT" instruction and skip the next instruction at sequence 39100. Therefore, the first instruction normally executed after this "PRT" command is the "BRU 38A" instruction which, in this case, will branch back to feed the next card. The control register has not been loaded with any special spacing command, therefore channel four (normal spacing) is selected. However, when a hole is sensed in channel 2 (bottom of form indicator) of the tape loop while under channel 4 control by lack of programing in the control register, printing will occur on the line of the form corresponding to the hole in channel 2. Then the "bottom of form" flag will be set in the processor of the E 6000. When this flag is set, and the next time the print instruction at sequence 39050 is reached and the printer tries to print below the "bottom of form" hole in the tape, the print instruction is ignored and the instruction following the print instruction will be executed. Consequently, a "BRU" instruction should immediately follow each "PRT" instruction. At the time the print instruction is ignored, the "bottom of form" flag in the processor is reset (turned off). As in the example above, the branch instruction following "PRT" should branch to print the last line on this page after which channel four programing should skip to the top of the next page. The "PRT" instruction at sequence 39200 (which is reached only when the "BRU" instruction at 39100 is executed as a result of a bottom of form flag setting) should be followed by an "NOP" as the next instruction will always be ignored. The program may then provide to print page headings prior to branching back to feed another card.

If page headings are desired, the alpha and mask constants would have to be moved into the print locations followed by another "PRT", and "NOP" instruction prior to branching back to get the next punch card.

On the following page is an example of a sample program tape.



PRINT ON
LINE PRINTER

F. Sample Program Tape

The example on the right illustrates tape for a ten (10) inch form with three zone areas. Punch in channels 3 and 4 on line 3 enables locating the first printing line at the start of a run. Punches in channel 1 at line 9 and 23 allows for skipping intervening lines. Punches in channel 4 are for regular spacing (double in the example). Lack of punches in channel 4 from line 54 to 60 and 1 & 2 causes a slew to the first printing line of the next form, under channel 4 control. Channel 2 punch at line 51 will permit a programed skip to this line but will not sense "bottom of form". Consequently, the flag will NOT be set after printing on line 51 and the next PRT instruction, following the channel 2 slew, will be executed.

If form headings are not required, channel 2 need not be punched and an NOP could be substituted for the BRU instructions following PRT instructions.

If form headings are required and channel 2 is to set the bottom of form flag, line 51 must be reached with a channel 4 advance from line 49 in this example.

PUNCH FROM MEMORY

PUNCH FROM MEMORY – This instruction causes data to be punched from memory on an on-line A 149 Card Punch. Both alpha and numeric data may be punched from memory. Numeric punching from memory is done from the working address, 00P. Each numeric word must be individually moved to the punching location. Alpha punching from memory is done from the symbolic memory address where it is stored. Alpha punching must be from an area where the information is stored in tandem memory locations. In this respect it is similar to type from memory.

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PW (30)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 7)
A	32000		PFM	S	05	00	P		PUNCH FRØM MEMØRY
B	32050								NUMERIC -SINGLE-
C	32100		PFM	D	02	00	P		PUNCH FRØM MEMØRY
D	32150								NUMERIC -DUAL-
E									

NUMERIC PFM – The alpha characters "PFM" in columns 18-20 denote this as a punch from memory instruction. The address, columns 26-28, indicates the memory location from which punching takes place. The numeric PFM instruction is, in effect, a two stage instruction. First, since numeric punching begins at the MSD, the digit position in 00P, where actual punching is to begin, must be shifted to the leftmost or MSD position in 00P. Thus, the length, 05 above, columns 23 and 24, indicates the number of spaces before punching begins. Only after the contents have been shifted does punching take place.

The "S", column 21 of the numeric illustration, shows this is a punch from memory "Single" instruction, that is, only the contents of 00P will be affected. A "D" in column 21 indicates a punch from memory "Dual" instruction. This instruction causes the contents in the MSD position of 00P to be shifted into the LSD position of 00K the number of places indicated in columns 23 and 24. The numeric PFM instruction has the same Dual and Single characteristics as the Shift instruction. The Non-Clear status flag is reset at the beginning of each punch instruction. This will permit the use of the Non-Clear status flag for recovery from Position Check errors. (See page 2-43.)

INSTRUCTION	CONTENTS OF MEMORY		EXPLANATION
			PFM SINGLE:
TCP	000007211945	00P	Move data to be punched into 00P.
PFMS0500P	721194500000	00P	The 5 MDS (above left) are not needed. They are shifted left 5 places and lost. The 7 LSD are to be punched. The numbers 7211945 will be shifted to the MSD positions. They will remain in 00P left justified after punching.
			PFM DUAL:
TCP	246800135990	00P	Move data to be punched into 00P.
	00K	00P	
PFMD0600P	246800	135990	The 6 MSD positions in 00P (above left) were to be saved. They were shifted in the LSD positions of 00K before punching. After the shift the numbers 135990 will be in the MSD positions of 00P. They will be punched and remain left justified in 00P after punching.

If all twelve characters were desired in the above examples, the word length, columns 23 and 24, would have been left blank. Punching would have ended after the twelfth digit had been punched. Numeric punching ends when:

1. Terminated by a word length signal from the A 149 program drum card No. 2.
2. All twelve digits have been punched.

ALPHA PFM – The alpha characters PFM in columns 18-20 indicate this as a punch from memory instruction. The modifier "A", column 21 of the illustration, indicates this as an alpha punch from memory instruction. In the illustration, punching will begin at the MSD in symbolic address 08B 07. No shifting or transferring into position is required. The punching of alpha will continue until terminated by:

1. A typewriter TAB or RET key end alpha code.
2. LSD of an MA with units digit of 9 (this digit will not punch).
3. Word length in the A 149 program drum card No. 2.

If no termination factor is sensed, punching will continue – even on to the next card.

SERIES E BASIC ASSEMBLER CODING FORM

SEQUENCE NO.	OP CODE	MOD	ADDRESS PG	ADDRESS R	PP W/I *	REMARKS
7 8 9 10 11	18 19 20	21	26 27	28	30 31	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
3 2 3 2 5	P F M	A	0 8	B	0 7	PUNCH FRØM MEMØRY ALPHA

Punching from memory is simplified when the following are clearly understood:

1. What data is to be punched?
2. Where is the data located?
3. Does the data need to be transferred and shifted – if so, to where and how many places?
4. How is the data to be spaced on the card which is to be punched?

To illustrate and further explain how this command can be used, follow the above steps in a typical program.

Assume:

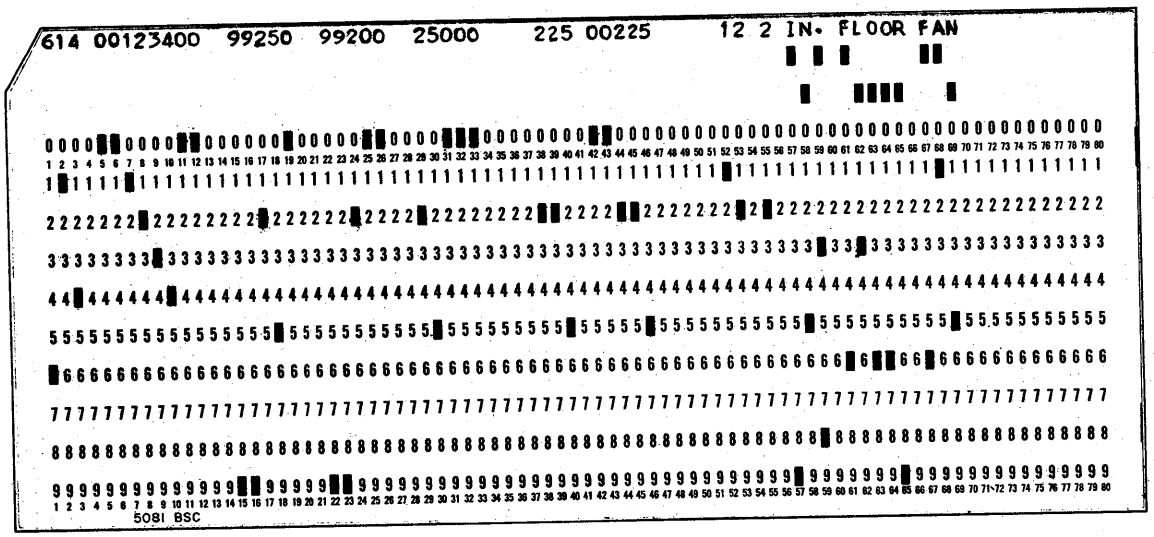
1. An inventory card has been read in with a "CRD 08B" instruction.
2. The information on this card has been updated in the program.

Card Information:

1. The format for the card is:

	ID	STOCK NUMBER	SELL PRICE	COST PRICE	REORDER POINT	ON ORDER	ON HAND	RESERVED	DES.
R									
X									
0	0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0
	1 2 3	4 5 6 7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24 25 26	27 28 29 30 31 32 33	34 35 36 37 38 39 40	41 42 43 44 45 46	47 48 49 50 51 52 53	54 55 56 57
1	1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1

2. Columns 4, 13, 20, 27, 34, 47, 54, 79 and 80 are to be skipped and will always be blank.
3. Preceding zeroes are desired in the stock number, (08 B 01) and ID number (08B).
4. Sell price, cost price, reorder point, on order, on hand and reserved, (08B 02 - 08B 06) never exceed five digits.
5. An updated inventory card is to be punched from memory.
6. The output card will look like this:



The outline suggested earlier may now be followed.

1. What is to be punched? The inventory data which has been read from a card and updated in the program.
2. Where is the data? The data to be punched is in the first nine words of the card-in file. (Label 08B)
3. What must be shifted? Numeric data which cannot be duplicated from the previous card must be transferred to 00P before it can be shifted and punched.
4. How much should the data be shifted? Punching starts at the MSD position. Data should be shifted left until the first digit to be punched is at the MSD or leftmost position in 00P.

The first field to be punched is a three digit I.D. number. In this example, all cards are to have the same I.D. number. Therefore, the I.D. number to be punched is the same as that in the first field on the previous card. A "DUP" instruction, sequence number 32150, causes one field, in this case the I.D. number, to be duplicated on the card now at the punching station. The I.D. number would have been set up in the first card.

Card column four is at the punching station. This column is to be skipped. The "SKP" instruction, sequence number 32200, causes one entire field to be skipped. In this example, Column 4 (and each of the other single columns which are to be skipped) has been defined as a field one column wide by placing 12 punch in Column 4 of the program drum card No. 1.

Column 5 now at the punching station, the punch is in position for the 8 digit stock number. The number is transferred to 00P in sequence number 32250. As 8 digits are to be punched, a 4 place left shift is all that is needed, to position the number before punching is executed, see sequence number 32300.

Column 13 is to be skipped, thus the skip instruction in sequence number 32350. Similar procedures are followed for Numeric Punching the contents of 08B 02 - 08B 06.

Note in sequence number 32900, the length columns 23 and 24 are blank. 00P will not be shifted and all twelve digits will be punched. If a 12 had been placed in the length columns, 00P would have been shifted 12 places to the left – in effect cleared, and zeros will punch.

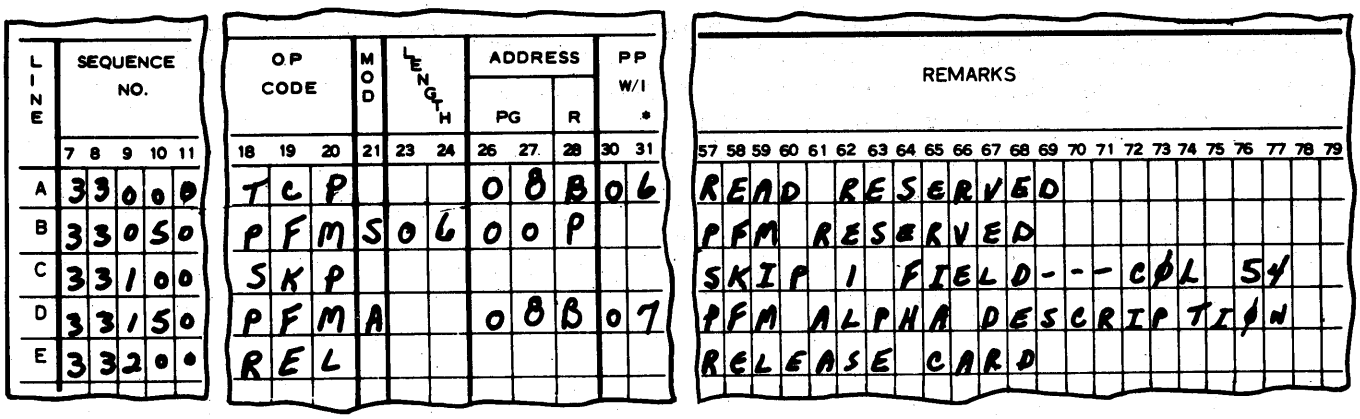
The instruction for punching all 24 characters of the alpha description is in sequence number 33150, as identified by the "A" in Column 21. No length for punching is needed. Alpha punching will begin at the symbolic address indicated, Columns 26-31, and will continue until a termination code is sensed in the drum card. In this case, an "O" punch was used in Column 78 of No. 2 drum card. If no termination punch is sensed, alpha punching will continue. In the example, had a termination punch in a drum card been omitted, punching would have continued not only in the last 2 columns of the present card, but also in the columns of the next card which was automatically fed in.

After the alpha has been punched, the card should be released and a new card read in (card releases automatically after Column 80 has been punched). The release instruction, "REL", sequence number 33200 provides for this.

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO. (7 8 9 10 11)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A								
B								
C								
D	32150	DUP						DUPLICATE 1 FIELD-ID ##0
E	32200	SKP						SKIP 1 FIELD---COL 4
F	32250	TCP			088	01		READ STOCK NO.
G	32300	PFM	S	04	00	P		PFM STOCK NO.
H	32350	SKP						SKIP 1 FIELD---COL 13
J	32400	TCP			088	02		READ SELL PRICE
K	32450	PFM	S	06	00	P		PFM SELL PRICE
L	32500	SKP						SKIP 1 FIELD---COL 20
M	32550	TCP			088	03		READ COST PRICE
N	32600	PFM	S	06	00	P		PFM COST PRICE
P	32650	SKP						SKIP 1 FIELD---COL 27
R	32700	TCP			088	04		READ REORDER POINT
S	32750	PFM	S	06	00	P		PFM REORDER POINT
T	32800	SKP						SKIP 1 FIELD---COL 34
V	32850	TCP			088	05		READ ON ORDER, ON HAND
W	32900	PFM	S		00	P		PFM ON ORDER, ON HAND
X	32950	SKP						SKIP 1 FIELD---COL 47

PUNCH FROM MEMORY



Character keys to the left of the numeric keyboard on the Control Console are not formatted or indexed to memory; therefore, they cannot be punched from memory.

Memory address keys 1 through 399 are not formatted or indexed to memory and cannot be punched from memory.

PROGRAM DRUMS NO. 1 AND NO. 2

Two Program Drums are provided on the Style A 149 Card Punch to control Punch From Memory (PFM) when the Card Punch is on-line to the Central Processor.

DRUM NO. 2 (left spindle) has two sections, a normal and alternate section, which are selected and set by separate internal instructions (NOR) (ALT) of the program. The number of columns in the field for numeric punching are controlled by Drum No. 2. The punching of alpha fields are controlled by both Drums No. 1 and No. 2.

DRUM NO. 1 (right spindle) includes a basic section, plus another section which uses four rows of control for end of field with any skip or duplication. This drum is used to program the number of columns for a field with a skip (SKP) or duplication (DUP) instruction; in addition, Drum No. 1 terminates a skip function initiated with an end of alpha code for a typewriter key code, or a not end alpha key code and no DL8 in the control register.

Program cards are punched and fitted onto the drums to provide the desired results for punching, skipping, or duplicating with Punch From Memory (PFM). Drum brushes sense the program card for the punched holes which are selectively programed to provide the punching control with the Card Punch and internal instructions.

ENDING ALPHA PUNCHING, TYPEWRITER KEYS, AND PROGRAMING DECIMAL LIGHT 8 (DL8)

Alpha fields must be a fixed length as programed on program Drum No. 2 and Drum No. 1. The alpha message in memory will not always be a fixed length. The termination of messages of various length is accomplished by utilizing word-length codes "0" and "6" on Drum No. 2, and end of field codes "6", "7", "8", "9", or "12" on Drum No. 1, to control the Skip associated with the Typewriter Keys.

Keys	1 2 3 4	Code	
High Order Code	9 9 8 9	"9"	= Always ends alpha and indexes skip of card columns.
Low Order Code	1 2 3 4	"8"	= Permits continued alpha punching with DL8.
		"8"	= Ends alpha and indexes skip of card columns without DL8.

The presence of DL8 in the Control Register enables the punching of the codes for the typewriter keys. A key code with a high order "8" will punch and alpha punching will continue until terminated. Without DL8 programed this key code ends alpha and skips the remaining columns of the field.

Programing of Dollar Amount Protection (DAP) in the control register determines the number of consecutive spaces on the card. With DAP the card will escape one column for each Space code read. Without DAP the card will escape only one column from consecutive space codes. The remaining area of the field is scanned until a code other than a space code is read.

1. Termination of alpha fields, general conditions:
   a. Word Length codes (Drum No. 2) terminate alpha punching after the alpha character for the last column is punched. A space code does not punch but is considered as a "blank punch" for alpha punching purposes. A punch or blank punch must occur for termination by word length. If memory includes characters in excess of the number allowed for the field, then these characters do not punch. Word length codes will not terminate a skip or duplication instruction.
   b. End of Field codes (Drum No. 1) terminate a skip which is indexed prior to the cards arrival at the last column of the field. With or without DL8 an end of alpha key code terminates punching of alpha characters and indexes a skip of the remaining columns of the field.
   c. Word Length and End of Field codes in the same card columns will insure the termination of punching and the termination of the field regardless of what memory codes are read for the last column of the field.
2. Termination by last digit of a decade:

   When the LSD of the last word of the decade is read, alpha punching is terminated and the card skips until terminated by an end of field code. The character for this last column does not punch, nor will an end of alpha code be punched.
3. Card column 80:

   In the absence of program drum code, column 80 will terminate a programed skip, duplication, or skip from an end of alpha code or end of alpha condition. Column 80 will not terminate numeric punching nor the punching of alpha characters. The numeric or alpha punching is temporarily held up until the following card is registered at column 1 and the punching can resume. This condition is usually not desirable.

COMMANDS FOR CARD PUNCH CONTROL

These commands are programed to select and set Program Drum No. 2 (left spindle) for the normal or the alternate program codes of the drum. The same instructions also select and set the four selectable end of field program codes of Drum No. 1. The normal section (upper section) of Drum No. 2 and the end of field code Row 6 of Drum No. 1 are automatically selected and set when power is turned ON to the system. When a section (NOR or ALT) of Drum No. 2 and an end of field code row (6, 7, 8 or 9) of Drum No. 1 are selected and set, the selected drum settings remain active until another internal instruction changes the selection and setting or the punch is off line. The settings may be changed prior to punching either numeric or alphanumeric data, or changed in between punching, skipping or duplication instructions. If the drum is set prior, you are assured that the proper controls are active.

INSTR

SKP Skip card columns until terminated by end of field code "6, 7, 8, 9, or 12" on Drum No. 1.

DUP Duplicate card columns until terminated by end of field code "6, 7, 8, 9, or 12" on Drum No. 1. Read Station reads card and punch station punches following card.

REJ Release card to Auxiliary Card Stacker.

REL Release card to Regular Card Stacker.

Select and Set:

Upper:

NOR 00 Normal of Drum No. 2 and end of field in Row 6, Drum No. 1.

NOR 01 Normal of Drum No. 2 and end of field in Row 7, Drum No. 1.

NOR 02 Normal of Drum No. 2 and end of field in Row 8, Drum No. 1.

NOR 03 Normal of Drum No. 2 and end of field in Row 9, Drum No. 1.

Lower:

ALT 00 Alternate of Drum No. 2 and end of field in Row 6, Drum No. 1.

ALT 01 Alternate of Drum No. 2 and end of field in Row 7, Drum No. 1.

ALT 02 Alternate of Drum No. 2 and end of field in Row 8, Drum No. 1.

ALT 03 Alternate of Drum No. 2 and end of field in Row 9, Drum No. 1.

Program Codes for Drum No. 2

The following punches in the normal and alternate sections of the Drum Card provide the described functions with numeric (PFM-S) (PFM-D) or alphanumeric (PFM-A) instructions. The program code must be punched in the card column where the result is required. Only one section of Drum No. 2 can be selected at any one time. The program codes for Drum No. 2 are ignored during a skip (SKP) or duplication (DUP).

SECTIONS

Normal		Alternate	Functions; Drum No. 2 Only
12	or	4	A "12" overpunch will occur in the corresponding card column if the sign of memory 00P is minus when a numeric punch instruction (PFM-S-xx-00P, or PFM-D-xx-00P) is executed.
11	or	5	An "11" overpunch instead of a "12" will occur with the above conditions.
0	or	6	The word length is programed to indicate the last column of a field for punch numeric, or punch alpha.
1	or	7	Position Check is programed in the first column of a field.
2	or	8	Disables Position Check.
3	or	9	Non-Punch Preceding Zeros is programed in the first column of a numeric field. Non-significant zeros to the left of first significant amount do not punch or print.

The two sections of Drum No. 2 provide the ability for the normal punching format to be completely different than the alternate format. For example, the Normal section the Program Drum No. 2 may indicate a field of 8 columns and Position Check in column 1. For the same application the Alternate section may indicate a field of 10 columns without a Position Check. Once a section is selected and set, the selected section remains active until the other section is selected or the punch if off-line.

More than one program code may be present in any one column. For example, codes "11" and "0" in one column would indicate an "11" overpunch (minus) and a word length code to terminate punching for the normal section, and codes "5" and "6" could be present for the alternate section.

Program Codes for Drum No. 1

The program codes for Drum No. 1 are used when the Card Punch is used on-line or off-line. With Punch From Memory active the Program Switch on the Card Punch Keyboard should be set at P1.

On-Line Code	Off-Line Code	Functions
12	12	End of Field, terminates skips or duplication of a field, or automatic operation of Card Punch off-line. This code takes precedence over any other.
	12/11	Automatic space, one-column skip.
	11	Automatic skip of a field, programed in first column of a field, terminated by an end of field code.
	0	Automatic Duplication of a field, programed in first column of a field, terminated by end of field code.
	1	Alpha shift, programed in each card column of a field, permits alpha punching without holding alpha shift key down, shifts back to numeric in the absence of a "1" code.
2	2	Blank Column Check, programed in each column, used to insure that punching occurs in the designated columns.

On-Line Code	Off-Line Code	Functions
	0/2	Blank Column Check and Automatic Duplication, the zero is programed in first column of a field, the "2" is programed in each column, duplication is terminated by end of field.
3	3	Print Suppress programed in each column to prevent printing.
11/3	11/3	Left zeros to print, programed in first card column of each field.
	4	Selective "12" overpunch, permits punching of "12" without space of card column and without use of Multi-Punch Key.
0/4	0/4	Automatic "12" Overpunch, punches a "12" in respective card column in addition to other punch.
	5	Selective "11" overpunch permits punching of "11" without space of card column and without use of Multi-Punch Key.
0/5	0/5	Automatic "11" Overpunch, punches an "11" in respective card column in addition to other punch.
6		Selectable End of Field, Row 6
7		Selectable End of Field, Row 7
8		Selectable End of Field, Row 8
9		Selectable End of Field, Row 9

POSITION CHECK OF CARD FIELDS

The Position Check function with Punch From Memory may be used to insure that the card field to be punched is in step with the program. As part of the programing for Punch From Memory, a Branch on Non-Clear (BRN) should be programed following a punch instruction. The Position Check function is programed to be related to the units digits of the position numbers (0-39). This capability permits a two-way check of the Position Check of the Program Control Center with the Position Check of a program code "1" or "7" on Drum No. 2. A Skip (SKP), Duplication (DUP), Release (REL), or Reject (REJ) instruction does not utilize the Position Check-Card feature.

Units Digit for Position Check in Control Register	Drum No. 2 Code in First Column of Field	
0	None	Checks
1, 3, 5, 7 or 9	No. 1 or No. 7	Checks
Any Number	No. 2 or No. 8	Check Disabled
0, 2, 4, 6, 8	No. 1 or No. 7	Check Error

With each punch numeric or alpha instruction the systems logic checks for position agreement unless disabled by a program code "2" or "8" (normal or alternate) on Drum No. 2.

Position check is satisfied with a units digit in the Control Register which includes a "1" bit of a binary combination (odd numbers 1, 3, 5, 7, 9) and a "1" or "7" punch in the respective normal or alternate section. Any different values for the units digit (0, 2, 4, 6, 8) in the Control Register create a Position Check error. The error condition will halt the execution of punching, and will halt the execution of the program. The Non-Clear Status Indicator is turned off to be reset by the Program Advance (PGA) switch light during the error release routine. The halt is accompanied by an illuminated PCK light at the right of the keyboard of the Control Console, plus the CARD PUNCH light and Decimal Digit 0 (DD0) light on the Central Processor are illuminated. These lights will be turned off after the Program Advance (PGA) switch light is depressed.

If the error is related to Position Check of the Program Control Center, the PCK light is the only one of the three illuminated.

RELEASE OF POSITION CHECK HALT

To correct the Position Check error the operator must:

1. Depress the PUNCH ON LINE switch light on the Card Punch Control Panel to put the system Off-Line.
2. Manually release the cards in the read and punch station if the error routine does not provide for the release of the cards. The cards may be manually released with two depressions of either the REL key on the Card Punch keyboard, or the RELEASE switch light on the Control Panel of the Card Punch.
3. The PGA switch light must be depressed. This light illuminates when the system is put off-line. The depression of the PGA control sets the Non-Clear Status Indicator and turns off the light. The program steps to the programed BRN instruction following the punch instruction. The Branch On Non-Clear (BRN) should be programed to execute the desired error recovery routine.
4. The PUNCH ON LINE switch light should be depressed to turn the system back to on-line. The program would continue executing as soon as the PGA switch light is depressed but instructions for the clear punch will not be executed until the system is turned on to punch.

POSITION CHECK AND SUBROUTINE JUMP (JMP)

If numeric or alpha punching are programed between instructions for a Subroutine Jump (JMP) and Subroutine Return, the Position Check feature should be disabled by a code "2" or "8" in the respective field on Drum No. 2. The memory address for the jump is stored in columns 1-3 of the Control Register and the units digit may create a Position Check error.

ERROR ROUTINE PROGRAMING FOR POSITION CHECK

The programing for the correction of a Position Check-Card error will depend upon the method used to format the data in memory, whether or not automatic duplication of the card is programed, the number of card formats being punched by the application, and whether or not the numeric data being punched is also used for accumulation.

Following the Branch On Non-Clear which is active after the Program Advance switch light is depressed, two internal instructions may be used to release the cards. One instruction to release the card to the regular stacker since this card is correct, the second instruction to release the card to the auxiliary stacker (REJ) to release the card which was in the Punch Station when the position check error occurred. The REL or REJ instructions will not execute until the Card Punch is turned on again by the PUNCH ON LINE switch light.

After the cards have been released, punching may be executed as determined by the requirements of the application.

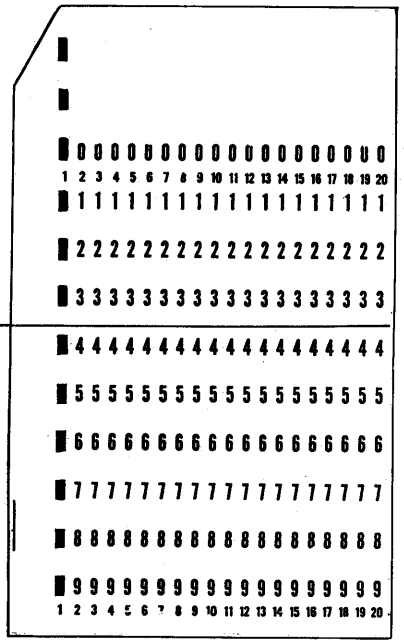
RELATED CONTROLS OF CARD PUNCH

The controls for the Card Punch are described in the A 149/A 150 EQUIPMENT REFERENCE MANUAL (1042207). The Card Punch PUNCH ON LINE switch is used to ready the system for on-line, plus the Power ON (green) switch of the Card Punch. The AUTO FEED switch light is used to keep a supply of cards at the Punch Station. Other than the Position Check light (POS CK) and Program Advance (PGA) light, the Control Console does not include any keys or lights which are related to Punch From Memory.

SUMMARY OF PROGRAM DRUM CONTROLS

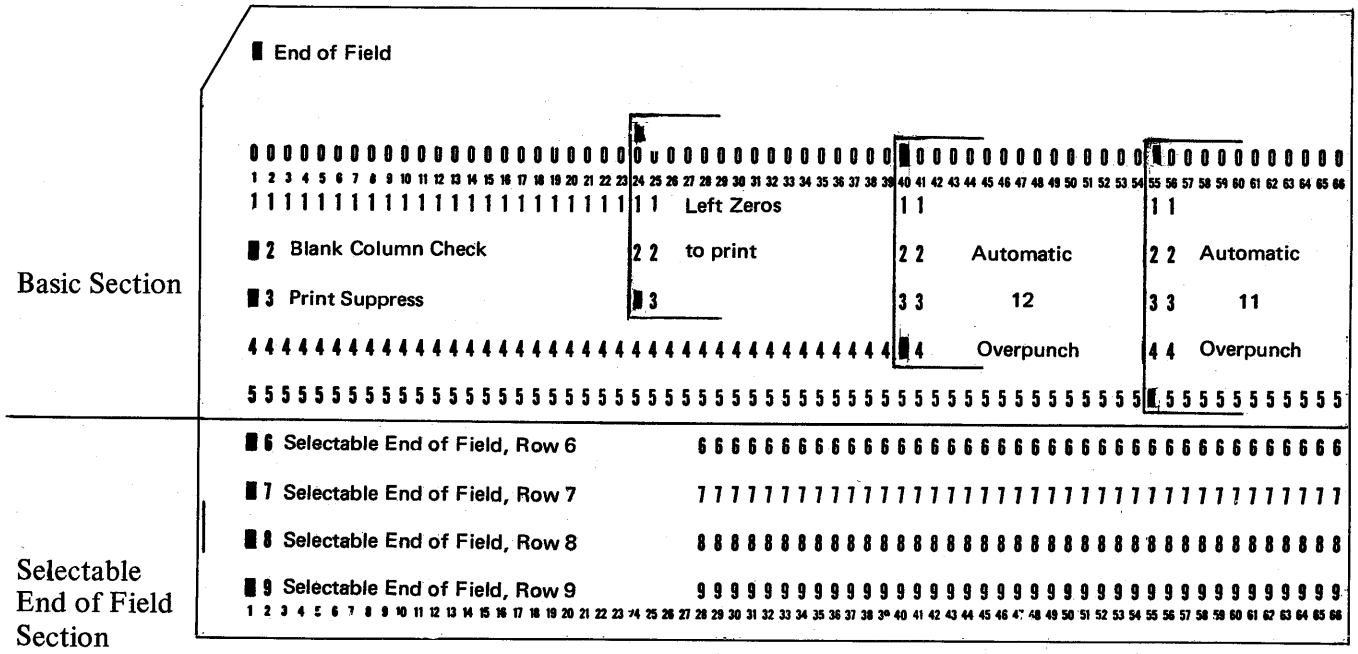
PROGRAM DRUM NO. 2, LEFT SPINDLE

Program Drum No. 2 has two sections which may be programed to function on-line with Punch From Memory. NORMAL SECTION: This is the upper section and includes Zones 12, 11, and 0, and Rows 1, 2, and 3 of the program card. ALTERNATE SECTION: The lower section of the program card includes Rows 4, 5, 6, 7, 8, and 9.



PROGRAM DRUM NO. 1, RIGHT SPINDLE

Program Drum No. 1 has a basic section plus another section which uses four numbered rows for end of field. BASIC SECTION: This is the upper area and includes Zones 12, 11, and 0, and Rows 1, 2, 3, 4, and 5 of the program card. SELECTABLE END OF FIELD SECTION: The lower section of the program card includes Rows 6, 7, 8 and 9.

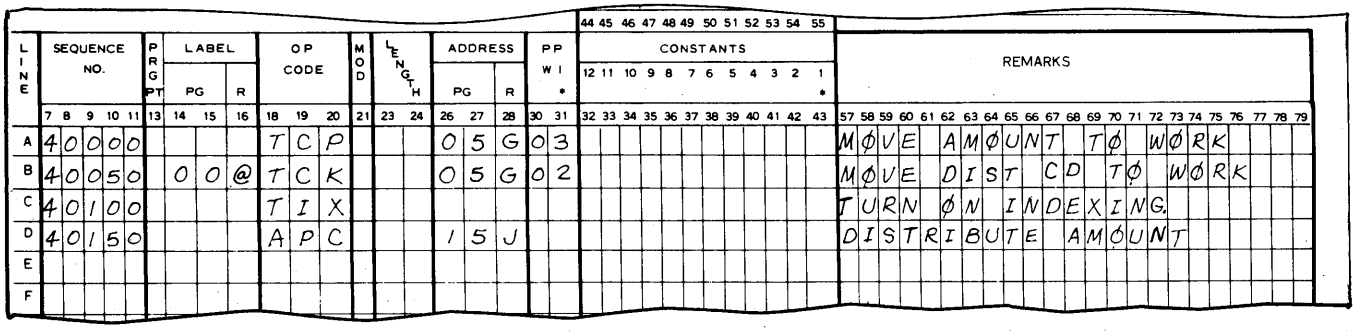


For the internal instructions which select and set each of the above sections – see Appendix E-2.

TURN ON INDEXING

TURN ON INDEXING (INDEX REGISTER) – The function of this command is to algebraically add the contents of 00K (MA 011) to the numeric value of the address of the next instruction and transfer the sum to the memory address register. This instruction (immediately following "TIX") will then be executed, as modified in the MAR, without changing the instruction in memory. (In the case of instructions having no real "address", such as shift, the whole instruction may be modified.)

The format of this instruction is:



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	40000				TCP			05G		03		MØVE AMØUNT TØ WØRK
B	40050		00@		TCK			05G		02		MØVE DIST CD TØ WØRK
C	40100				TIX							TURN ØN INDEXING
D	40150				APC			15J				DISTRIBUTE AMÓUNT
E												
F												

The alpha characters "TIX" in columns 18-20 of sequence 40100 indicate that this is a "Turn On Indexing" instruction. Notice that there is no address associated with this instruction.

The "TIX" instruction may be used to assist in an internal sort routine. For example, assume that punched cards are being read into 05G with the amount going into 05G plus three and the distribution code going into 05G plus two. Once a punched card in read, the amount to be distributed can be moved to a working location, ("P" in sequence 40000 of the example above) the distribution code is then moved to 00K just prior to the "TIX" instruction. This is done with the instruction located at sequence 40050 in the above example, notice that this instruction will be assembled as the first syllable in a word of memory due to the 00@ symbol (c.c. 14-16) which is coded with it. The following instruction is "TIX". It will combine the value (distribution code) now in 00K with the corresponding columns of the next instruction in the memory address register. From the example above, the next instruction is "APC – 15J". Therefore, if a distribution code of .25 were loaded into 00K the amount added at sequence 40150 with the APC – 15J instruction would have gone into 15J plus 25 memory locations. The numeric command at sequence 40150 would not have been altered in any way and need not be reset.

The contents of 00K, when using the "TIX" instruction, must be loaded in the same corresponding columns as the command to be modified otherwise it will have no affect. For example:

1. 000000000025 will modify the first instruction following "TIX" only if it is in the third syllable of an MA.
2. 000000250000 will modify the first instruction following "TIX" only if it is in the second syllable of an MA.
3. 002500000000 will modify the first instruction following "TIX" only if it is in the first syllable of an MA.
4. 002500250025 will modify the first instruction following "TIX" regardless of its location in a memory address.

The base instruction (APC – 15J in the above example) being modified will normally identify a memory location of less than MA 200 due to assembly procedures. The label indicated by this instruction may be incremented up to and including MA 399 with the "TIX" instruction, without regard to the boundary of MA 200.

TURN ON 200 AND TURN OFF 200 – The function of these instructions is to allow the E 8000 system to work with the upper plane of memory (MA's 200-399)

The format of these instructions are:

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	4 1 0 0 0				T Ø N							T U R N Ø N 2 0 0
B												
C												
D	4 1 1 5 0				T Ø F							T U R N Ø F F 2 0 0
E												
F												
G												

The instruction "TON" is necessary for machine language instructions that do not contain a modifier to designate an address above 200. For example, the machine language for APC – 025 (1025) is exactly the same as that for APC – 225 (1025). Programing "TON" prior to the instruction APC – 025 causes addition in MA 225 instead of MA 025.

Once "TON" has been executed all following instructions will work with the upper plane of memory until the instruction "TOF" (turn off 200) is executed. Therefore, great care should be used with these commands.

When programing in symbolic language, as you will be, the instructions "TON" and "TOF" will seldom, if ever, be necessary. This is because the "TIX" instruction, described previously, will automatically turn on and off the upper plane of memory when necessary.

In all cases the assembler will assign the files used (striped ledger, punched card, line printer and alpha files) in the lower 200 memory locations. Control register and non-sequential constants will also be reserved in the lower plane of memory. Sequential constants, which are defined last (see Page 2-50) will at least begin in the lower plane of memory. Since all files and constants are addressed with a defined label, it is not necessary to use "TON" or "TOF".

For example, assume that the programmer wanted to make a 210-way distribution of data from punched cards. He would therefore set up in the sequential constants section of his program (Page 1-2 of the coding sheets as described in Section 1) a label which would be used to address all 210 words of memory. Then by program the "TIX" instruction prior to the add or subtract command associated with that label "TON" and "TOF" would automatically be affected when necessary.

G. Control Register Commands

The function of control register commands is to assist in the program control of the console of the E 8000. Most console commands can come only from an internal source through the control register, some console instructions must be pinned in the control unit of the system and others may come from either source. This discussion will not include those commands which must come from an external source, and it is assumed that 1/10" carriage movement and functional motor bar programing is understood.

Procedurely, each control register command will be discussed after which it will be shown how they are programed. The various commands are split into five sections:

1. Carriage movement
2. Console print
3. Keyboard operation and printing
4. Line printer spacing/continuous forms spacing
5. Striped ledger

1. Carriage movement:
   a. "AMB" – Activate motor bar by the flag setting. As discussed earlier under program flags (Page 2-16) one function of a program flag was "to set up proper carriage travel by indicating the motor bar to be active at the next console cycle". With the command "AMB" active in the control register when the console operates either automatically from a "PAC" instruction (Page 2-6), or from a manual Bar 2 depression, carriage movement will result from the motor bar set by the program flag.

      If flag 1 is set, carriage movement would result with bar 1 programing. If flag 2 is set, carriage movement would result with bar 3 programing. If flag 4 is set, carriage movement would result with bar 4 programing. If no flag is set, carriage movement would result with bar 2 programing.

   b. "LN4" – Index lane 4 (DSR with bar 1 active). With this command active in the control register upon console cycle a lane 4 skip will result if bar 2 is active. If bar 1 is active the "LN4" command will act as a DSR and allow lane 1 movement. To move in lane 3 flag 8 must also be set prior to the console cycle. A bar 2 lane 4 skip or a bar 1 lane 1/3 skip and a DSR may be externally programed as with other equipment, however as much as possible should be done internally to increase program flexibility from a given series of print positions.
   c. "LN5" – Index lane 5 (DS with bar 1 active). With this command active in the control register upon console cycle a lane 5 return will result if bar 2 is active. If bar 1 is active, the "LN5" command will act as a "DS" and thereby cause a return in lane 2. When a skip (LN4) or return (LN5) is programed and active, a knock-off must be externally programed. The only exception to this is a bar 1 return in lane 3 which will automatically return one position. If it is desired to have a bar 1 function return in lane 3 from internal command, then flag 8 must be set prior to cycle. A bar 2 lane 5 return or a bar 1 lane 2/3 return and a DS may be externally programed as with other equipment.
   d. "PXX" – Position check XX against control unit. The position check instruction must be programed with each console cycle. The position called for by the "PXX" command will be illuminated on the console communication lights and must agree with the position pinned in the control unit. If the two are not in agreement, the "PCK" light illuminates and the console cycle is blocked. The lock and "PCK" light will be turned off when the carriage is moved to the position called for in the control register. There are 40 possible position checks, they are "P00" through "P39".

e. "SMB" – Set program flags via motor bar used. With this instruction active in the control register at a given console cycle, the motor bar used to activate the console will set the corresponding program flag for later use. If bar 1 were used, flag 1 would be set, bar 2 would not set any flags, bar 3 would set flag 2, and bar 4 would set flag 4.

The following page will serve as a review of carriage movement control register commands. It will indicate what must be programed to give certain carriage movement with a given motor bar.

E 8000 CARRIAGE MOVEMENT

MOVEMENT DESIRED:	MOTOR BAR A*	MOTOR BAR B**	LANE	PINS NEEDED	RELEASE REQUIRED	FLAGS SET	CONTROL REGISTER
NON-TAB	2	2	– –	NT	NO		
NON-TAB AND SPACE	2	2	– –	NT, SP	NO		
	2	3	– –		NO	2	AMB
	3		– –		NO		
TO RIGHT							
TO ADJACENT STOP	2	2	BT		NO		
TO SELECTED STOP	1		1	NONE	YES		LN4(DSR)
	1		3	NONE	YES	8	LN4(DSR), AMB
	2	2	4	NONE	YES		LN4
	2	1	3	NONE	YES	1 & 8	LN4(DSR), AMB
	2	1	1	NONE	YES	1	LN4(DSR), AMB
	2	4	1	NONE	YES	4	AMB
	4		1	NONE	YES		
TO RIGHT AND SPACE							
TO ADJACENT STOP	2	2	BT	SP	NO		
TO SELECTED STOP	2	2	4	SP	YES		LN4
TO LEFT							
TO ADJACENT STOP	1		RM	NONE	NO	8	LN5(DS), AMB
	2	1	RM	NONE	NO	1 & 8	LN5(DS), AMB
TO SELECTED STOP	1		2	NONE	YES		LN5(DS)
	2	1	2		YES	1	LN5(DS), AMB
	2	2	5		YES		LN5
TO LEFT AND SPACE							
TO ADJACENT STOP	1		RM	NONE	NO	8	AMB
	2	1	RM	NONE	NO	1 & 8	AMB
TO SELECTED STOP	1		2	NONE	YES		
	2	1	2	NONE	YES	1	AMB
	2	2	5	SP	YES		LN5

TYPE FROM MEMORY MOVEMENT

TO RIGHT – LANE 27
TO LEFT – LANE 28, 29, 30

*A – USE THIS COLUMN FOR OPERATOR ACTION OR REPEAT OF MOTOR BAR
**B – USE THIS COLUMN FOR AUTOMATIC CYCLES (UNDER PAC CONTROL)

2. Control console print:
   a. "APS" – Print a subtotal symbol. When this command is active in the control register and the print control allows, a subtotal symbol will print to the right of the amount being printed.
   b. "APT" – Print a total symbol. When this command is active in the control register and the print control allows, a total symbol will print to the right of the amount being printed. If it is desired to print .00 from a read command, APT or APS must be programed.
   c. "DAP" – Dollar amount protection. When this command is active in the control register a preceding $ will print just to the left of the most significant digit. If DAP is in the control register during TFM, successive space codes will be recognized. See Page 2-24. (See Page 2-41 for effect on PFM.)
3. Keyboard operation and printing:
   a. "CLA" – Clear the "C" address (must be programed with "RKC", read keyboard and add "C"). When this command is active in the control register the address specified by the "HLT" or "PAC" instruction will be cleared. The data cleared may not be printed or accumulated.
   b. "DL5" – Light decimal light 5. With this instruction active in the control register the decimal lamp located between columns 5 and 6 will illuminate.
   c. "DL8" – Light decimal light 8. With this instruction active in the control register the decimal lamp located between columns 8 and 9 will illuminate. If neither "DL5" or "DL8" is programed decimal light 2 will automatically be illuminated. (See Page 2-40 for effect on PFM.)
   d. "EKA" – Enforce the use of memory address keys. With this command active in the control register a console cycle will be prevented, until the operator indexes a memory location with the memory address keys of the console.
   e. "RDC" – Read the "C" address. With this command active in the control register, the address specified by the "HLT" or "PAC" instruction will be loaded into 00P and will be available for printing. The prior contents of 00P will be destroyed.
   f. "REV" – Reverse normal arithmetic. With this command in the control register the sign of 00P is changed from plus to minus or minus to plus. Since each console cycle activates 00P, all normal arithmetic associated with that cycle would be reversed.
   g. "RKA" – Read the memory address selected on the keyboard. With this command active in the control register a memory address selected from the keyboard will be read and transferred to the memory address register.
   h. "RKC" – Read keyboard and add "C". The function of this command is to add the amount indexed on the keyboard to the address specified by the "HLT" or "PAC" instruction. If "REV" is also programed, the amount will be subtracted since the sign of 00P will be changed to minus.
4. Line Printer and Continuous Forms Spacing:
   a. "VC 1", "VC 2" and "VC 3" – These commands when present in the control register during a line printer cycle will cause the forms to space in the selected channel (1, 2 or 3) until a hole is sensed in that respective channel. Spacing as controlled by channel 4 is automatic if none of the above three commands are programed. Note: VC 1 is the same command as EKA and if EKA is in the control register during a line printer print operation spacing will occur per channel 1 control. VC 2 is the same command as RKA and VC 3 is the same as RKA and EKA together.
   b. "HME" – Home Tractor. The presence of this instruction in the control register will cause the continuous forms carriage to space to the home position as controlled by its spacing control tape.

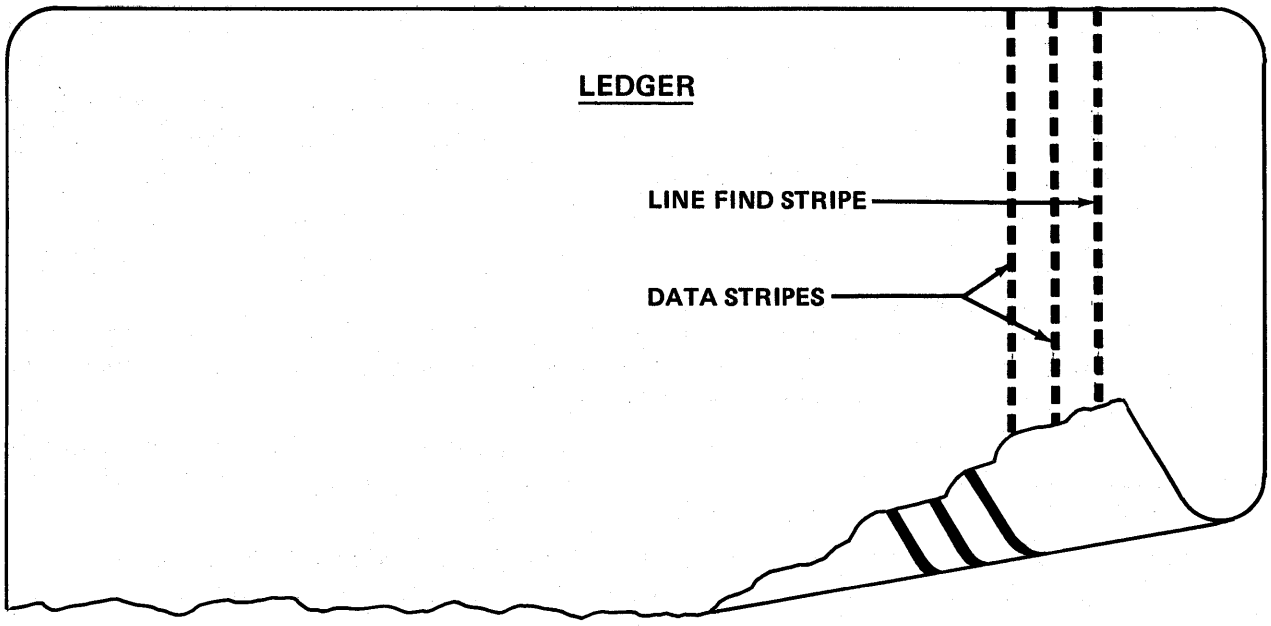
c. "ZNE" – Zone Tractor. The presence of this instruction in the control register will cause the continuous forms carriage to space to the next zone position. The zone position is used on the carriage to control variable line spacing.

5. Striped ledger:

Prior to discussing commands used by the control register to control striped ledgers, a few facts about the striped ledger itself should be understood.

The striped ledger forms specifications and printing requirement may be obtained from the equipment and reference manual or your Burroughs representative.

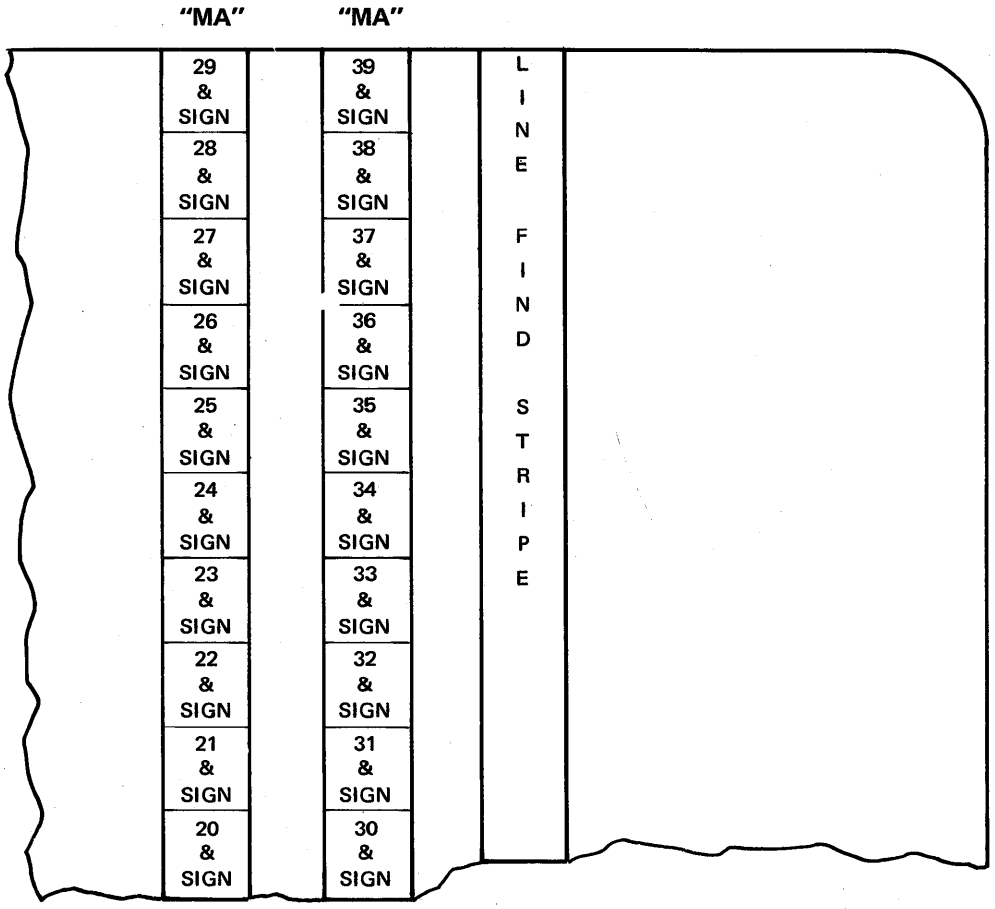
Each striped ledger has three magnetic encoded stripes on its rear side. As illustrated below, the stripe nearest the outside edge is the line find stripe and will control form alignment. The other two stripes house data associated with that ledger card.



The line find stripe will normally have one "pulse" encoded on it; that being the "line and pulse" which indicates to the carriage of the console the next posting line.

Each of the two stripes contain codes for 130 characters of information. Of these 130 characters, ten "words" of data are encoded each having 12 data characters and one sign character.

As indicated in the example below the left-hand stripe will represent the data of memory locations 20-29; the center stripe will represent the data of memory locations 30-39.



The total storage capacity of a single set of stripes on a striped ledger is 260 characters. Both numeric and alpha characters may be encoded on a striped ledger. One alpha character consists of two numeric characters both of which are in the same digit position ten memory locations apart.' With respect to numeric data, any number of codes or balances may be stored in one memory location, so long as they have the same sign. Any combination of alpha and numeric data may be stored on one set of stripes as long as the total does not exceed 260 characters.

The following commands are used in the control register to control striped ledgers:

a. "ALN" – Align striped ledger. The presence of this command in the control register will cause a striped ledger to be aligned automatically to the next posting line as indicated by the line find pulse in the right-hand stripe. The primary memory locations (MA's 20-39) will not be cleared and the data stripes will not be read. This command is used to align a ledger to the next posting line without attempting to read its data and is used with a striped ledger reconstruct routine.

b. "BAD" – Block add. The presence of this command in the control register will cause the data read into the primary memory locations (20-39) to be added algebraically to the striped ledger secondary memory locations (60-79). This command can be used only when a striped ledger is read into the system and is normally used with automatic non-print trial balance operations. This command will not be used when the line printer is used since the printer also uses the MA 60 decade for mask codes. If "BAD" is used in a program, a line printer file should be declared to reserve the MA 60 decade and a numeric card file declared to reserve the MA 70 decade.

c. "BCC" – Align blank card and clear. The presence of this command in the control register will cause a blank (new) striped ledger to be aligned. The primary memory locations (20-39) will be cleared of any prior data. (Note: "ALN" would not align a new or blank ledger as no line find pulse has yet been encoded on it.) Alignment, as controlled by "BCC", will be to one of two places. Either line 1 (the first posting line) or line minus 3 (three lines above the first posting line). The location to which alignment is made is determined by a switch setting on the carriage. With this switch in the forward position alignment would be to line 1; in the rear position to line minus 3. This instruction is normally used with new account or initial installation operations. If there are to be machine printed headings on a new ledger, the form would be inserted to line minus 3 and must then automatically be spaced up to line 1 prior to the ejecting and writing of that ledger so that data may be properly encoded.

d. "BLC" – Align blank card. The presence of this command in the control register will cause the same function and react in the same way as "BCC". The difference between the two commands (BCC & BLC) is that blank card (BLC) will not clear the primary memory locations of any prior data. The only way to introduce a new or blank ledger into the carriage of the E 6000 console is through the use of "BCC" or "BLC". If a blank card is inserted with any other command or commands, or is manually inserted, the three stripes will not be properly conditioned to accept new data. "BLC", like "BCC", is normally used for new account and initial installation routines.

e. "BTF" – Block transfer. The presence of this command in the control register will cause the secondary memory locations (60-79) to be cleared of prior data. Then the data read into the primary memory locations (20-39) will be added to the secondary locations. The "BTF" command can only be used when a striped ledger is read into the system and is sometimes used to load constants, to be used by the program, into memory. Note: See BAD with respect to MA's 60-70.

f. "FDL" – Feed ledger via the A 4004 Auto Reader. The presence of this command in the control register will cause a striped ledger to feed through the auto reader. A read ledger must also be programed to read the data on the ledger being passed. This command will not cause the console of the E 8000 system to cycle, although a HLT instruction must be programed in order to recognize the commands in the control register, once the commands are executed the processor will proceed to the next instruction. This command is normally used on automatic trial balance and other report routines or for automatically inputing variable data in an accounting operation.

g. "RDL" – Read a striped ledger. This command will read the data from a striped ledger and transfer it to the primary memory locations clearing any prior data already in memory locations 20-39. This command will not cause the striped ledger to be aligned. If the ledger is to be both read and aligned then both "RDL" and "ALN" must be programed. As with any command that calls for a ledger to be inserted into the console, the carriage must be open from a prior console cycle. This command will be used any time it is desired to update a ledger card with new information.

h. "VER" – Verify keyboard with memory address 20. This command, when present in the control register, will compare the entire keyboard against the contents of memory location 20, disregarding signs. If they agree normal operation will continue, if not the system will lock. One of two conditions must be present in order to make the "VER" command work. (1) A striped ledger must be in the carriage of the console or (2) a ledger must have just been passed through the auto reader.

i. "WRC" – Write correction. When a write error occurs the control register should be loaded with this command. It will cause, or allow, the striped ledger to be aligned to line one (1) without clearing the primary memory location (20-39) or any data on the stripe. (Note – We cannot

erase the line find stripe, which would be correctly encoded, or the next alignment would go to line two which may be invalid.) The console must not be allowed to print on the ledger with this command. Once this command has been executed the processor should step to a write ledger operation to again eject and write ledger.

j. "WRL" – Write and eject striped ledger. This command is also programed in the control register and will cause the striped ledger to be ejected. The data now in memory locations 20-39 will be written on the striped ledger. The carriage must be open from the prior console cycle in order to allow the form to be ejected. (Note: The ledger will not actually be written and ejected until the carriage has reached the next position). The data in the primary memory locations is not changed as a result of executing this command.

On the following page is a listing of each control register command. Review this list and refer to it as needed while studying the following programing discussion.

CONTROL REGISTER CONSTANTS

ALN %	ALIGN STRIPED LEDGER (USED WITH RDL TO READ AND ALIGN)
AMB	ACTIVATE MOTOR BAR BY THE FLAG SETTING
APS	PRINT THE SUBTOTAL SYMBOL
APT	PRINT TOTAL SYMBOL
BAD	BLOCK ADD STRIPED LEDGER DATA TO MA'S 60-79
BCC %	ENTER STRIPED LEDGER AND CLEAR STRIPED LEDGER AREA
BLC %	ENTER BLANK STRIPED LEDGER
BTF	CLEAR AND ADD MA'S 60-79
CLA	CLEAR THE "C" ADDRESS BEFORE RKC*
DAP	DOLLAR AMOUNT PROTECTION
DL5	LIGHT DECIMAL LIGHT 5
DL8	LIGHT DECIMAL LIGHT 8
EKA	ENFORCE THE USE OF THE MEMORY ADDRESS KEYS
FDL	FEED LEDGER VIA AUTO READER
HME	HOME TRACTOR
LN4	INDEX LANE 4 (DSR WITH BAR 1)
LN5	INDEX LANE 5 (DS WITH BAR 1)
PXX	POSITION CHECK XX AGAINST CONTROL UNIT
RDC	READ THE "C" ADDRESS
RDL %	READ STRIPED LEDGER DATA (READS DATA ONLY)
REV	REVERSE NORMAL ARITHMETIC
RKA	READ THE MEMORY ADDRESS SELECTED ON THE KEYBOARD
RKC	READ KEYBOARD AND ADD "C"
SMB	SET FLAGS VIA MOTOR BAR USED

VC1	VARIABLE CHANNEL
VC2	BOTTOM OF FORM
VC3	TOP OF FORM
VC4 **	SINGLE LINE
VER	VERIFY KEYBOARD WITH MEMORY ADDRESS 20
WRC %	WRITE CORRECTION
WRL %%	WRITE STRIPED LEDGER
ZNE	ZONE TRACTOR

* RKC MUST BE PROGRAMED ALSO
** NO CODING REQUIRED
% CONSOLE CARRIAGE MUST BE OPEN PRIOR TO THESE COMMANDS
%% CONSOLE CARRIAGE MUST BE OPENED AT THE OPERATING POSITION WHERE WRL IS PROGRAMED. WRL WILL BE PROGRAMED IN CONJUNCTION WITH A HLT 00W OR PAC 00W. (00W = MA 009)

Each of the control register commands discussed are represented in memory as numeric values. The numeric values which represent the various commands desired at a given console, or A 4004 Auto Reader, operating position are added together and put in memory as a "control register constant."

Below is a diagram of each control register command. From the block location of the command, the value and digit position which will represent that command may be determined.

SERIES E 8000 CONTROL REGISTER FORMAT

9	–	–	–	–	–	DL8 AMB	–	ZNE RDC	–	–	–	PCK 9
8	–	–	–	BCC	–	DL8	–	ZNE	–	–	–	PCK 8
7	–	–	–	–	–	–	LN5 REV CLA	–	RKC RKA EKA	–	LN4 PCK 30	PCK 7
6	–	SMB FDL	–	WRC	–	–	REV LN5	–	RKC RKA	DAP APT	LN4 PCK 20	PCK 6
5	–	SMB WRL	BTF RDL	BLC VER	–	DL5 AMB	LN5 CLA	HME RDC	RKC EKA	DAP APS	LN4 PCK 10	PCK 5
4	–	SMB	BTF	BLC	–	DL5	LN5	HME	RKC	DAP	LN4	PCK 4
3	–	–	BAD RDL	ALN VER	–	–	CLA REV	–	RKA/VC3 EKA/	–	PCK 30	PCK 3
2	–	FDL	BAD	ALN	–	–	REV	–	RKA/VC2	APT	PCK 20	PCK 2
1	–	WRL	RDL	VER	–	AMB	CLA	RDC	EKA/VC1	APS	PCK 1	PCK 1
	12	11	10	9	8	7	6	5	4	3	2	1

Digit Position

As indicated earlier, the numeric values of the functions desired at a given operating position are added together and loaded in memory as a control register constant. For example, to add an amount listed on the keyboard, at operating position number 6, into an enforced selected MA (memory address) would require RKC, EKA, RKA, and P06 numerically represented as a control register constant. From the above diagram it is seen that:

RKC = 40.00 (A value of 4 in digit position 4)

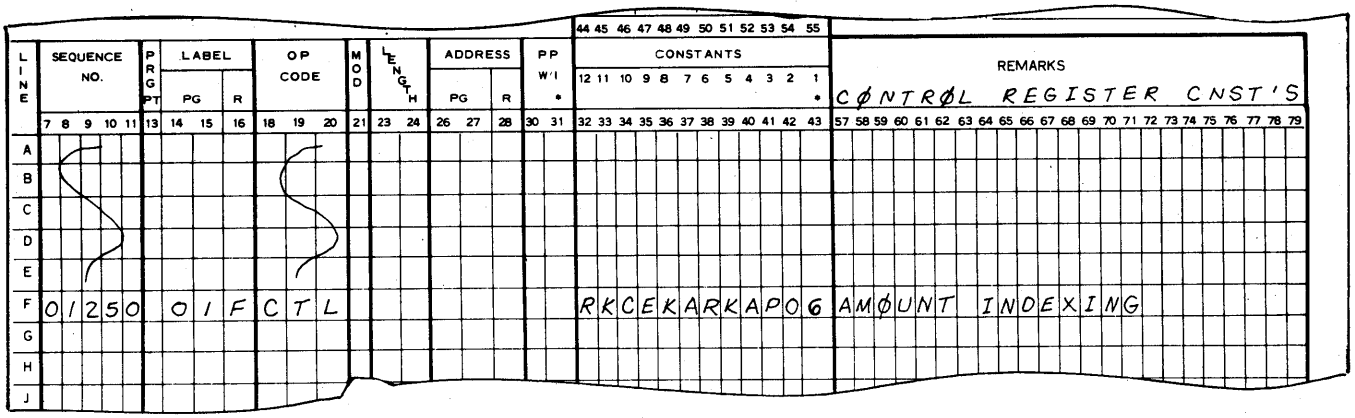
EKA = 10.00 (A value of 1 in digit position 4)

RKA = 20.00 (A value of 2 in digit position 4)

P06 = .06 (A value of 6 in digit position 1)

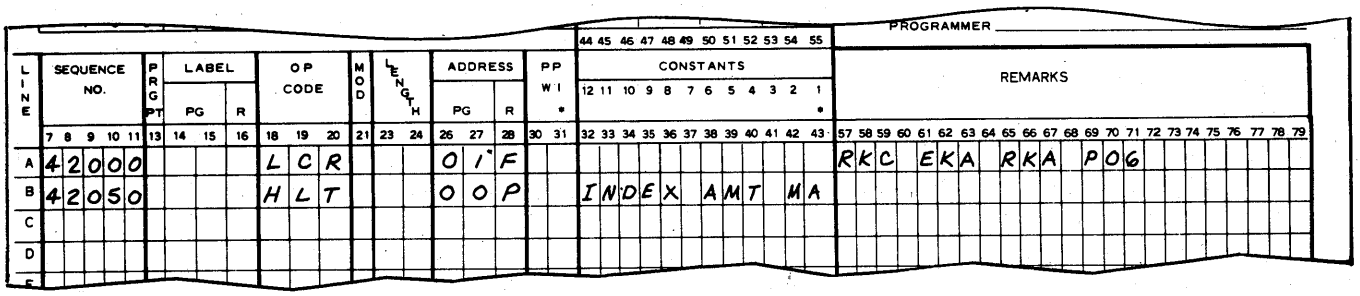
Thus "70.06" will represent RKC, EKA, RKA and P06 and will be loaded in memory as a "control register constant".

The example below indicates the proper coding for the above control register constant which would be coded on Page 01 of the coding sheets as discussed later under pseudo instructions.



The function of the pseudo instruction "CTL" will be discussed later. However, it is important to understand at this time that the assembler will automatically code "70.06" and assign it to memory location 01F as a result of this programing.

Now that a control register has been identified it can be seen how that constant can be used in a program to activate the console of the E 8000



At sequence number 42000, in the above example, the control register was loaded with the content of 01F which was defined as a control register constant in the previous example. (Note: "LCR" is discussed on Page 2-25.) The control register constant loaded in the control register was defined as having the numeric value of 70.06 which represents RKC, EKA, RKA and P06. Therefore prior to operating at position 6 the control register would be loaded with the contents of memory location 01F, then a "HLT"

instruction would be programed which will turn control of the system over to the operator to index an amount and memory address to be active. When the motor bar is depressed processing will resume with the next instruction as is indicated with the discussion of the "HLT" instruction on Page 2-5.

In the following example, assume it is desired to activate motor bar 1 and skip in lane three until knocked-off with a bar two depression. This will help keep system operation very easy from an operator standpoint.

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	43000				SFL		09					SET FLAGS 1 AND 8
B	43050				LCR			01	G			RKC AMB LN4 P07
C	43100				HLT			10	H		INDEX AMØUNT	
D	43150				RFL		09					RESET FLAGS 1 AND 8
E												
F												
G												

In this example the "SFL 09" instruction is used to set flags one and eight to later activate Bar 1 lane 3 movement. At sequence 43050 the control register was loaded with commands to activate the motor bar and lane travel as indicated by the program flags (AMB). Skip movement was indicated by the "LN4" instruction (a DSR with motor bar one active). At sequence 43100 processing was halted for a keyboard entry. When motor bar 2 was depressed a lane three (bar 1) action resulted. (Note: An external knock-off must be programed.) At sequence 43150, the "RFL 09" instruction reset (turned off) flags one and eight.

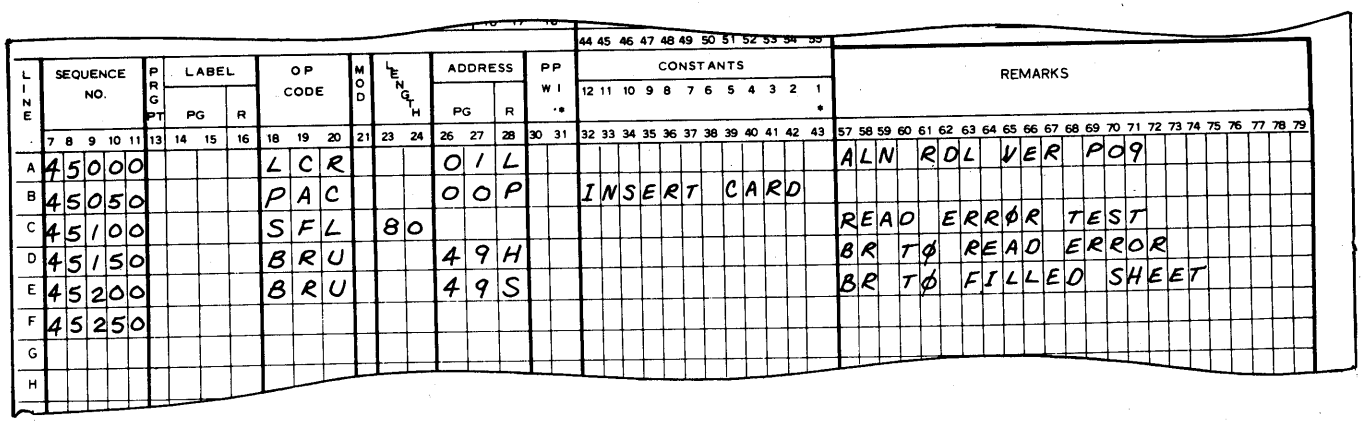
In the following example, assume that it is desired to operate at a position in which the operator is to have the option of using bar 1 or bar 2. Bar 1 and 2 are to return to different positions and automatically operate with their respective functions, but skipping in lanes one or four on the second operation.

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	44000		44	A	LCR			01	J			RKC LNS SMB P08
B	44050				HLT			00	P		INDEX AMØUNT	
C	44100				SFL		10					TEST FØR BAR 1
D	44150				BRU			44	H			BR TØ ALTERNATE STØP 05
E	44200				LCR			01	K			ROL LN4 P02
F	44250				PAC			00	P			PRINT TØTAL SKP TØ P08
G	44300				BRU			44	A			BR TØ NEXT ITEM
H	44350		44	H	LCR			01	J			RKC LN4 AMB P05
J	44400				PAC			00	P			PRINT TØTAL SKP TØ P08
K	44450				RFL		01					RESET FLAG ØNE
L	44500				BRU			44	A			BR TØ NEXT ITEM
M												

In the above example, at sequence 44000 the control register is loaded to read the amount indexed on the keyboard and add to 00P (this is done at position #8). If motor bar 1 is used flag 1 will be set, if bar 2 is used no flag will be set. Assume that bar 1 is externally knocked-off at position 5, bar 2 at position 2. The first instruction after the console cycle is "SFL 10" which will test for bar 1 use (flag 1). If bar 1 had been used, the program will operate on sequence 44150 and branch to load the control

register for sequence 44400. If bar 2 had been used, sequence 44150 would be skipped, the program will not branch and the control register will be loaded for sequence 44250. In either case the program will branch back to sequence 44000 and the carriage will skip in lane 1 or 4 to position 8 assuming proper location of external knock-off. A review of program flags (Page 2-16) or AMB, SMB, LN4 and LN5 may be necessary at this point, do not continue unless the above is understood.

In the following example, the programing necessary to align, read and verify a striped ledger is shown.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI	CONSTANTS	REMARKS
A	45000				LCR			01L				ALN RDL VER P09
B	45050				PAC			00P			INSERT CARD	
C	45100				SFL		80					READ ERROR TEST
D	45150				BRU			49H				BR TO READ ERROR
E	45200				BRU			49S				BR TO FILLED SHEET
F	45250											
G												
H												

Notice from the above example that both "ALN" and "RDL" are programed to both read and align a striped ledger. The insertion of a striped ledger will not cause an automatic machine cycle; this is to allow the insertion of a second form such as a check. If it is desired to insert a second form, a "HLT" instruction should be programed after loading the control register. If an automatic machine cycle is desired, a "PAC" instruction should be programed as in the above example. The special test flag instruction "SFL 80" is used following all read or write ledger operations. In this case it was used after inserting the ledger to check for both a read error or filled sheet condition. (Note: See discussion of the SFL 80 instruction on Page 2-17.)

If it is desired to print with the console instruction used to insert and read a striped ledger, the contents of memory location 20 may be printed if "VER" and "RDC" are loaded in the control register.

READ LEDGER OVERRIDE

The Read Ledger Override procedure provides a method of by-passing the Control Register when it requires a striped ledger. This permits interruption of a program without the use of special purpose ledgers. The depression of the Program Load (PGL) and Program Advance (PGA) switch lights, in that order, and the test of Flag 2 will provide the ability to bypass the necessity of reading a striped ledger. The override feature will be active only with a Read Ledger (RDL) command in the Control Register, with the carriage closed without a ledger present, and a Control Console halt (either Halt (HLT), or Print and Cycle (PAC). If the Control Register includes Feed Ledger (FDL) the Auto Reader must be turned off. Other commands for striped ledger feeding or alignment may be present in the Control Register along with RDL. However, the override procedure is not possible with a Blank Card instruction (BLC or BCC).

Programing for the override function is related to the programing for Filled Sheet. Flag 2 is set (turned on) by the detection of a Filled Sheet during reading of the striped ledger, but the Read Ledger Override programing is dependent upon Flag 2 being reset (turned off). Programing should provide that Flag 2 is reset when the program is at the read ledger sequence. The Activate Motor Bar (AMB) command at the halt should not utilize Flag 2.

A Read Ledger routine is outlined below to illustrate additions required for testing for the Read Ledger Override function.

The SFL 80 instruction will always skip the next two instructions unless a striped ledger error indicator is set, or unless the Read Ledger Override function is active. The override function is activated by a depression of the Program Load (PGL) and Program Advance (PGA) switch light.

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27 28		30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	20010		20A		RFL		02					CLEAR FLAG 2
B	20020				LCR			01C				RDL ALN 001
C	20030				PAC			00P			INSERT LEDG	
D	20040				SFL		80					LEDGER ERROR TEST
E	20050				BRU			50A				TO READ ERROR
F	20060				BRU			55A				TO FILLED SHEET OR
G												STRIPE LEDGER OVERRIDE

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27 28		30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A			55A								FILL SHEET--	LEDGER OVERRIDE ROUTINE
B	55010				RFL		22					TEST AND RESET FLAG 2
C	55020				BRU			60A				TO FILLED SHEET ROUT
D	55030										LEDGER OVER-	RIDE
E	55040										ROUTINE	

A. Filled Sheet

When a Filled Sheet condition exists in the Read Ledger and align position, the Filled Sheet and Program Advance (PGA) switch lights are illuminated. Depressing the PGA switch and then the Filled Sheet switch will cause the SFL 80 instruction to be executed and Flag 2 set. The next instruction will be skipped and the instruction following will be executed.

B. Read Ledger Override

When the Ledger Override is desired, the carriage, without a ledger present, must be closed, and the Auto Reader turned OFF if Feed Ledger (FDL) is in the Control Register. The PGL and PGA switch lights are depressed in that order. The SFL 80 instruction is executed and the following first instruction is skipped, but the next BRU instruction will be executed. The test and reset of Flag 2 (RFL22) should be the first instruction in the program to which the BRU to Filled Sheet Override was directed. The RFL22 instruction skips the next instruction (BRU to Filled Sheet) if the override function is active because Flag 2 is not set.

A Filled Sheet condition is not recognized by the Auto Reader.

FILLED SHEET ROUTINES

Filled Sheet In-run

When a Filled Sheet condition is detected during a posting run, including Type To or From Memory (TTM, TFM) Flag 2 is set and the Filled Sheet switch light is illuminated. The processing of instructions continues. In the absence of "one-line posting" applications, Flag 2 should be used in the program to test whether or not additional form spacing functions should be allowed or prevented for the form which is present. Spacing beyond the last line may endanger the proper writing of the data on the stripe. An in-run Filled Sheet condition will occur when a striped ledger form is spaced to the last useable line. The Filled Sheet switch light will not turn off Flag 2, nor will a reset instruction (RFL 02) turn off Flag 2.

Flag 2 may also be set when the Set Motor Bar (SMB) command is active in the Control Register and Bar 3 is depressed. The use of Flag 2 and Activate Motor Bar (AMB) to space and non-tabulate at a carriage position may conflict with a test for a Filled Sheet condition.

Test of Flag 2 for Filled Sheet Condition

There is no one rule for the test of Flag 2 for Filled Sheet of a striped ledger; there are many possibilities. The following are common conditions where Flag 2 should be tested.

A. Test Flag 2 for a Filled Sheet at a point in the program prior to a Control Console cycle (HLT, PAC) where a carriage controlled vertical form space may occur, or where a form space may be initiated by a Motor Bar of flag function (Bars 1 and 3, plus their flags may space).

B. Test Flag 2 before or after typing each line with Type To or From Memory (TTM or TFM) if the typed message is for more than one line and if there is a possibility that the ledger may be spaced to the last line of the ledger (Line 37 or 38) during any one posting. (Note: Typewriter keys do not set flags.)

After a form space a minimum of 110 ms is required to permit the movement of the platen before the setting of Flag 2 caused by a Filled Sheet. During this time the program continues and several instructions could be executed prior to the setting of Flag 2. If Flag 2 is tested immediately following a Control Console command (HLT or PAC) where a striped ledger may be spaced up, the flag may or may not yet be set.

The point in the program where Flag 2 is tested for Filled Sheet will depend on the application. For some applications it may be possible to program carriage movement to provide and to insure a last line routine.

For programs that require operator decisions for typing successive lines, or Control Console spaces the program may be different. It may be advantageous to test for the Flag 2 setting prior to starting the program for multiple lines of posting or typing, because data which is generated to be printed and written may have to be cleared, or corrected prior to writing the striped ledger.

SERIES E BASIC ASSEMBLER CODING FORM

LINE	SEQUENCE NO.	PRG FT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I	CONSTANTS	REMARKS
A	22000				LCR			01	C			RKC LN5 SMB P04
B	22010				HLT			09	C		LIST AMØUNT	BAR 2 BAR 4 FØR FINAL
C	22020				SFL		20					TEST TØ SEE IF THIS WAS
D	22030				BRU			30	A			LAST LINE RØUTINE
E	22040				RFL		44					TEST IF FINAL LINE
F	22050				BRU			32	A			TØ FINAL LINE RØUTINE
G												
H												

LINE	SEQUENCE NO.	PRG FT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I	CONSTANTS	REMARKS
A	23000		23	A	LCR			01	D			RKC LN4 SMB P06
B	23010				SFL		20					TEST TØ SEE IF THIS IS
C	23020				BRU			24	K			LAST LINE RØUTINE
D	23030				HLT			09	B		LIST AMØUNT	BAR 2 BAR 4 FINAL LINE
E	23040				RFL		44					TEST FØR FINAL LINE
F	23050				BRU							6Ø TØ FINAL LINE RØUT.
G												
H												

Filled Sheet with Manual Typing

Type To Memory (TTM) Programing may be used for typing of successive lines on the face of a striped ledger, which is not to be encoded. This programing permits the execution of the program and the test of Flag 2. Manual typing with only a halt (HLT) will not permit the test on each line. With a TTM routine, after the filled sheet light turns ON and flag is set, the selected (last) line routine may be used. The memory programed with TTM would be working memory and memory must be reserved for the longest message for any one line. The maximum message length is 120 characters.

PSEUDO INSTRUCTIONS

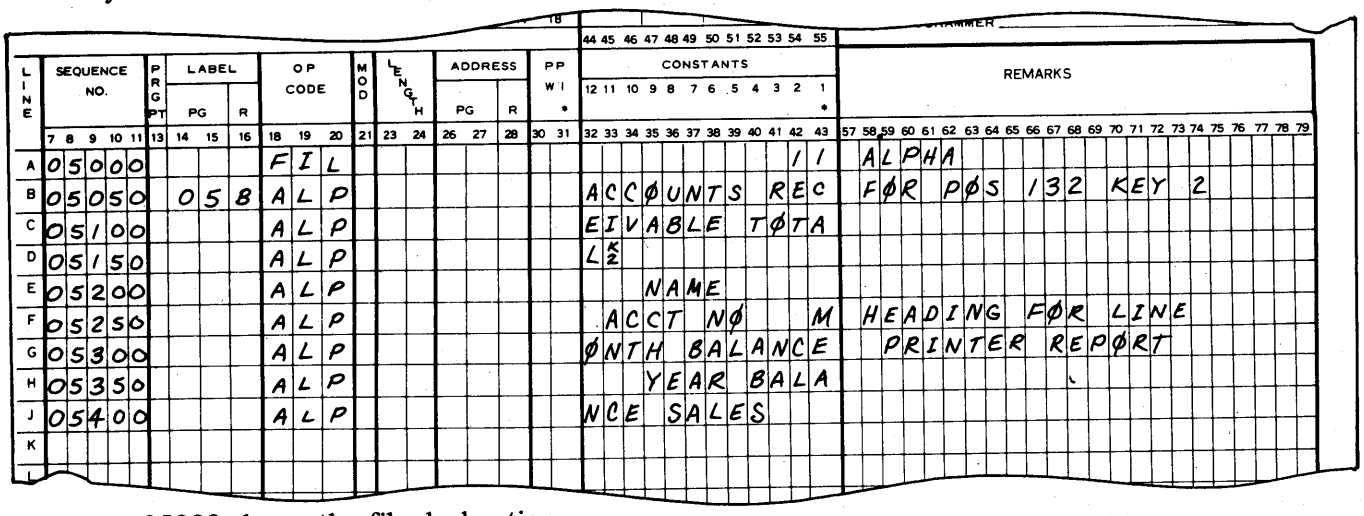
Pseudo instructions are operation codes punched in columns 18-20 that do not directly produce machine language instructions but rather control the manner of assembly and determine the interpretation of data fed to the assembler.

ALP – ALPHA CONSTANTS IN A FILE DECLARED AREA

DEFINITION – ALP is used to declare alpha constants as an input-output area by placing the low order digits in the even numbered decade and the high order digits in the next higher odd numbered decade.

EXPLANATION – Each constant is alphanumeric and may be up to twelve characters in length and will be entered left justified in columns 44-55. This pseudo instruction is usually entered following a FIL declaration "11" although it could be used to enter one-time data in any of the other file areas. Any alpha constants that will be printed on the console from a TFM instruction should be entered with the ALP pseudo so that memory is correctly aligned. Any alpha constant requiring more than twelve digits will be entered on subsequent lines along with the ALP pseudo for each line.

Normally a label will be used only on the first line and incremented to reach additional constants.



Sequence 05000 shows the file declaration.

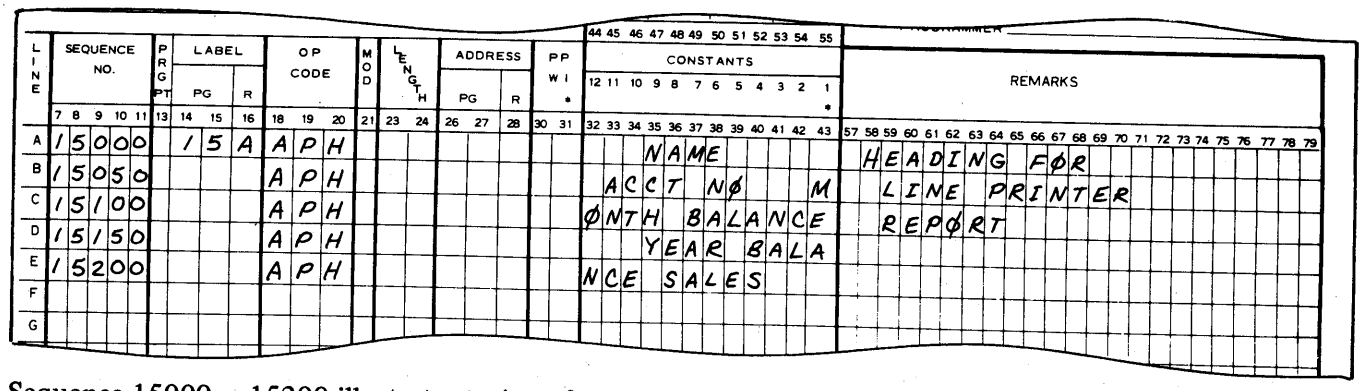
Sequence 05050 – 05150 illustrates an alpha constant stored for printing on the console with a key code to end-alpha. The instruction for this message would be a TFM 05B.

Sequence 05200 – 05400 illustrates a heading line for use on the line printer which does not require an end-alpha code. This data would be moved to the line printer area by addressing 05B plus 3 as the starting address.

APH – ALPHA CONSTANTS NOT DECLARED IN A FILE

DEFINITION – The APH instruction is used to define alpha constants by placing the low order digits in the labeled address and the high order digits in the next adjacent word (the labeled address plus one).

EXPLANATION – Each constant is alphanumeric and may be up to twelve characters in length and will be entered in columns 44-55. Any alpha constant requiring more than twelve digits will be entered on subsequent lines along with the APH pseudo for each line although a label will only be assigned to the first entry. No entry is to be entered in the length field.



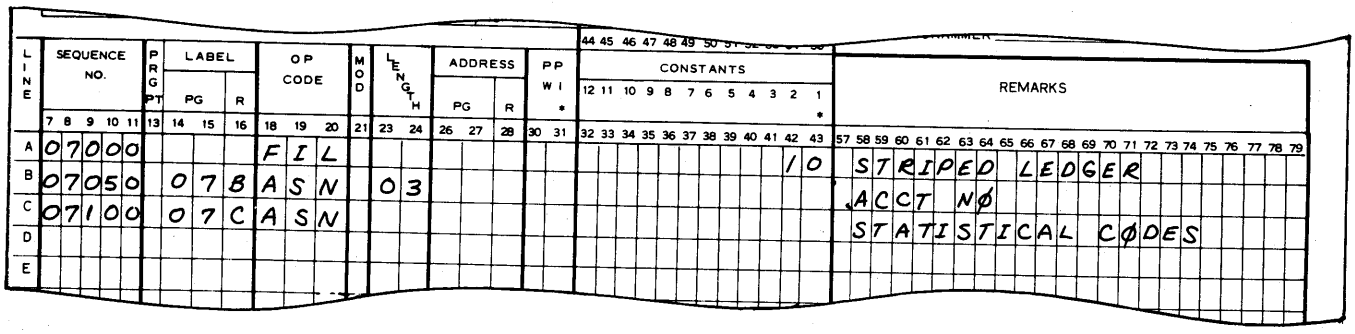
LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	15000		15	A	APH						NAME	HEADING FØR
B	15050				APH						ACCT NØ M	LINE PRINTER
C	15100				APH						ØNTH BALANCE	REPØRT
D	15150				APH						YEAR BALA	
E	15200				APH						NCE SALES	
F												
G												

Sequence 15000 – 15200 illustrate storing of an alpha constant, sixty digits long. Each portion of the message would be moved to the appropriate print area by addressing the label 15A and using word increments in increments of two, i.e., to address the data in sequence 15100 would require a label of 15A plus 4 for the low order portion and 15A plus 5 for the high order portion.

ASN – ASSIGN

DEFINITION – The ASN pseudo instruction is used to assign a label or labels to the input-output areas specified by a FIL declaration.

EXPLANATION – This instruction may be used to assign single or multiple labels within a file as specified in the length field. It should be used at least once in each file declaration except the Alpha File.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	07000				FIL						10	STRIPED LEDGER
B	07050		07	B	ASN		03					ACCT NØ
C	07100		07	C	ASN							STATISTICAL CØDES
D												
E												

Sequence 07050 assigns a label 07B to the first word in the striped ledger file (MA 20). By including a length (in this case 3) it is possible to assign another label 07C directly to MA 23. However, this would be done only for the convenience of the programmer since MA 23 could be identified with 07B plus 3 as well as 07C.

(BLANK) – REMARKS

DEFINITION – Blanks are used to insert additional documentation for the program in the Remarks field columns 57-79 by leaving columns 18-20 blank.

EXPLANATION – These entries are printed on the program listing but have no affect on the program.

Remarks may be used at any place. Some common points where this pseudo might be used are:

a. After the HDR card to further explain the contents of the program.

b. After the ASN in the striped ledger, card reader or line printer files to further identify the input or output detail.

c. After HLT instructions to further explain operator instructions.

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27 28)	ADDRESS R	PP W I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	07000				FIL						10	STRIPED LEDGER
B	07050				ASN		20					0 ACCT NØ
C	07100											1 ACCT BAL
D	07150											2 SALES TØ DATE
E	07200											3 MØNTH SALES
F	07250											4-5 NAME
G												
H												

Sequence 07100 – 07250 illustrates stripe location identification as remarks in the documentation.

CST – CONSTANT

DEFINITION – The CST instruction is used to define numeric constants, totals or working MA's either sequential or non-sequential dependent on a value being specified in the length field.

EXPLANATION – All constants are entered in columns 32-43 and may be plus or minus. A minus constant is indicated by an over-punch in column 43. All constants are to be right justified. Totaling and working areas are identified as constants and are usually initialized to a clear condition by placing a value of zero in column 43.

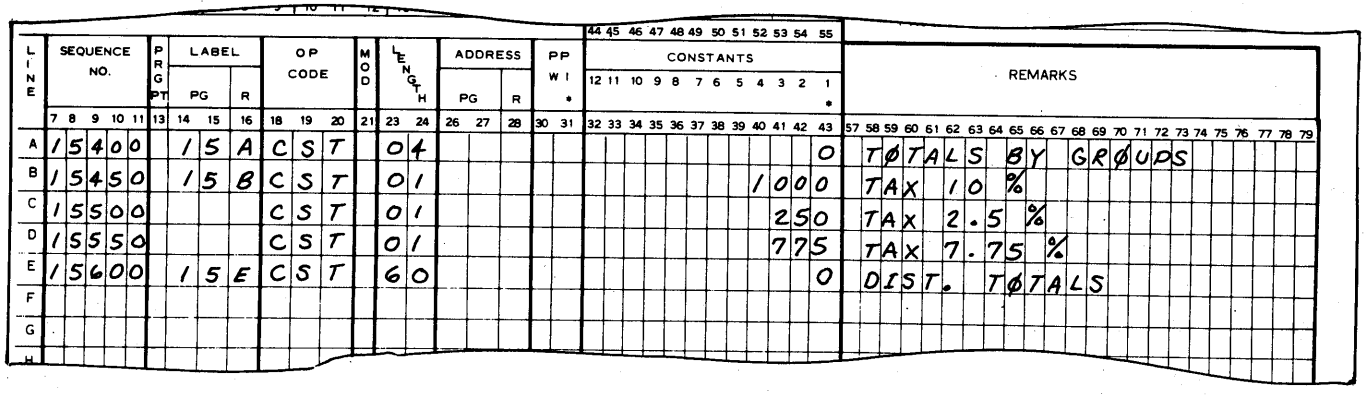
Non-sequential constants will have no value entered in the length field and will require a label for each constant. These constants will be assembled to fill in memory not used by the file declarations.

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27 28)	ADDRESS R	PP W I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	10400		10A		CST						912345123424	TK MØVEMENT
B	10450		10B		CST						111111111111	ALPHA MASK
C	10500		10C		CST						000100010001	INCREMENT
D	10550		10D		CST						002000500010	LIMITS
E												
F												
G												

The above sequences illustrate types of data that are commonly entered as non-sequential constants.

Sequential constants will be identified by a value of from 01 to 99 in the length field. A label is not required for each constant as these constants may be accessed by using a base label with a word increment. These constants must be used when using index register (TIX) to address them.

When coding, it is important that non-sequential constants are sequenced prior to sequential constants. Thus non-sequential constants will be located in memory under MA 200 and will not require TON and TOF. See Page 2-36.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15 16		18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	15400		15	A	CST		04				0	TOTALS BY GROUPS
B	15450		15	B	CST		01				1000	TAX 10 %
C	15500				CST		01				250	TAX 2.5 %
D	15550				CST		01				775	TAX 7.75 %
E	15600		15	E	CST		60				0	DIST. TOTALS
F												
G												
H												

Sequence 15400 shows initializing four consecutive locations for group totals to a zero value. These locations could be addressed individually by using a word increment in column 31 with the label 15A.

Sequence 15450 – 15550 illustrates three tax rates stored in memory which might be addressed by an index register (TIX) since they will be in three consecutive locations.

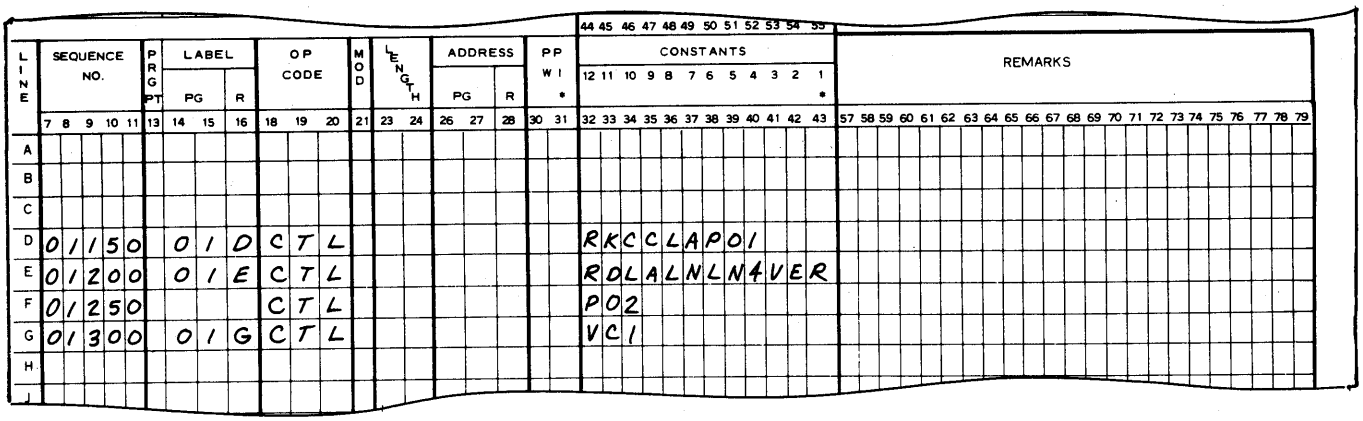
Sequence 15600 illustrates initializing 60 locations to a zero value for distribution totals which could be addressed by TIX (index register).

CTL – CONTROL REGISTER CONSTANT

DEFINITION – The control register constant (CTL) is used to convert alphanumeric instructions to their numerical equivalents which are entered in the object program as control register constants. These constants will be labeled to be addressed whenever an LCR operation code is programed.

EXPLANATION – Each control register function to be entered in a constant is listed as a three-character alphanumeric mnemonic in the control register field columns 44-55. These mnemonics must be left justified in the field with no intervening blanks. Once a CTL instruction has been labeled and declared, all subsequent entries are treated as additional functions for the same control register constant until terminated. When more than four functions are required in one constant, additional lines with CTL entries must be coded although these entries would not be labeled.

Termination is determined by a blank in the first column of any mnemonic sub-field or by a position check function. No function can be entered after a position check or blank. A blank will cause a zero position check. Each constant would be identified by a label.



Sequence 01150 illustrates a normal control register constant.

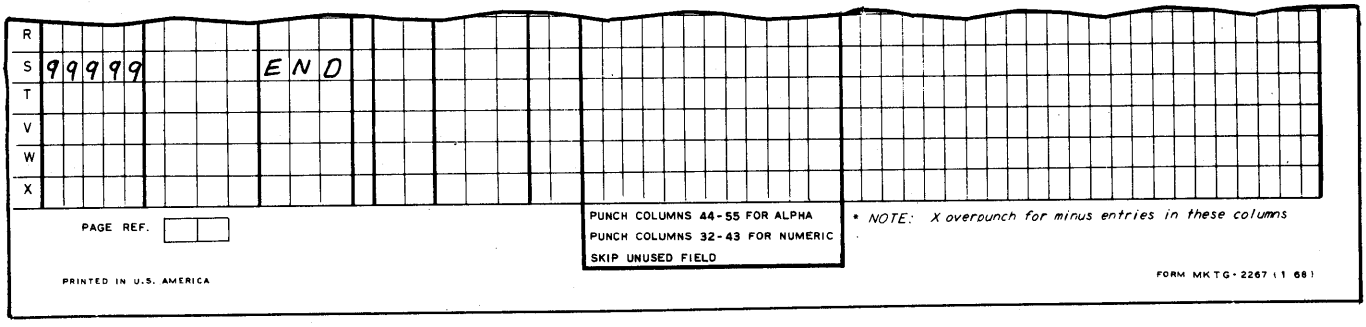
Sequence 01200 and 01250 illustrates a constant requiring more than one line since more than four functions were required. This also illustrates the absence of a label on the second line but the requirement of CTL as a pseudo.

Sequence 01300 illustrates a constant not requiring a position check.

END – END

DEFINITION – The END instruction is used to specify the end of the symbolic deck.

EXPLANATION – It must always be the last card in the symbolic deck, to indicate to the assembler that all symbolic cards have been processed.



FIL – FILE DECLARATION

DEFINITION – The FIL instruction is used to declare the labels and detail required for use in the input-output areas as defined by the specifications card (Section 1, Page 1-2 coding form).

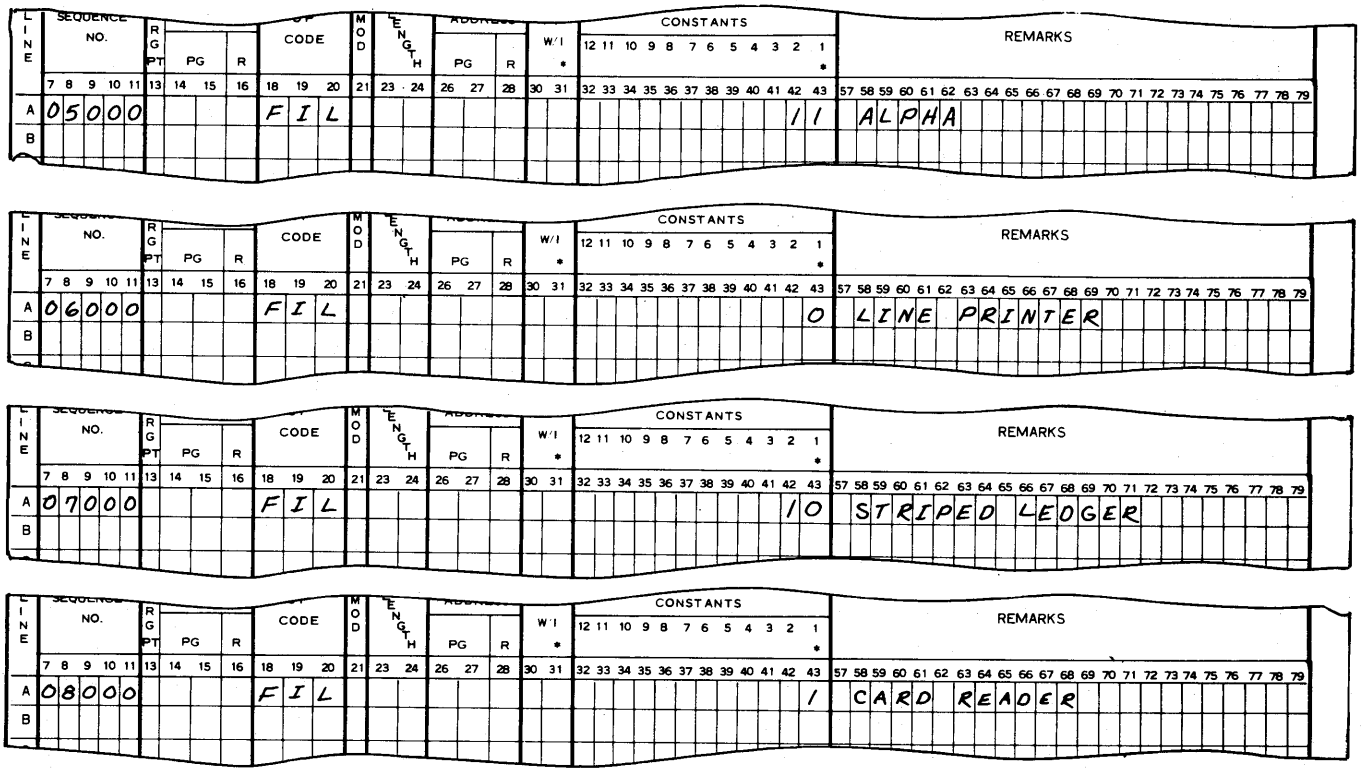
EXPLANATION – All input-output capabilities defined in the specification card will require a file declaration.

The specific file being defined is identified in columns 42-43 as follows:

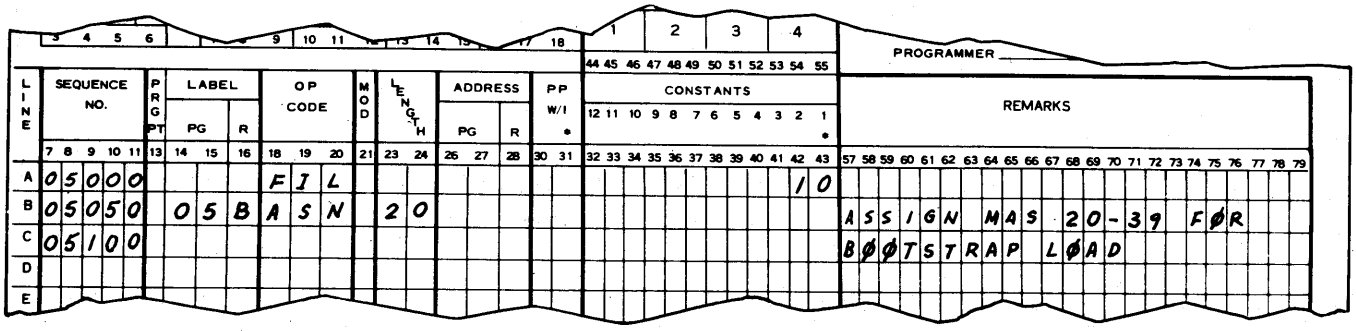
"11" – Alpha file
"0" – Line Printer file
"10" – Striped Ledger file
"1" – Card Reader File

File declarations may be used more than once, however, the same area in memory will be assigned to each subsequent declaration of the same file.

It should be remembered that the file declaration does not reserve memory. Memory is only reserved by the specifications card. The file declaration is only used to identify which area is referenced by subsequent ALP'S, etc. to assign labels and constants to that area previously reserved by the specifications card.



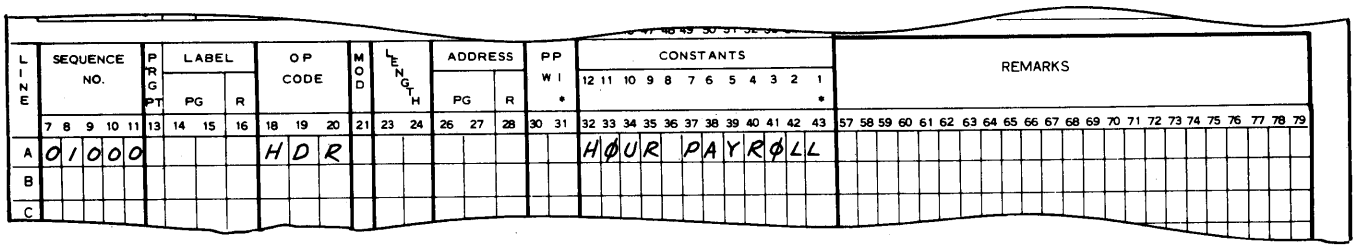
Caution: When the punch card bootstrap program uses MA's 20-39 a 20-word striped ledger file must be declared. See example below. After Assembly, MA's 20-39 which are referenced by the label 05B (incremented by 0-19) may be used for Card In, Total Area, Working, etc.



HDR – HEADER

DEFINITION: The pseudo instruction HDR is used to specify a heading to be printed at the top left side of each page of the assembly listing.

EXPLANATION – The heading is alphanumeric and may be up to twelve characters in length. It will be entered on line A of page 01 in columns 44-55 and will be the first symbolic card following the specifications card.



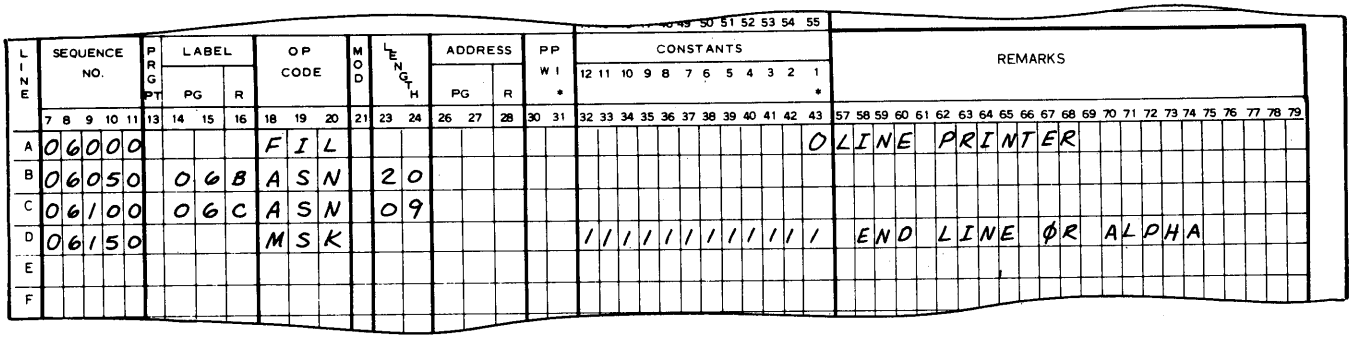
LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	01000				HDR						HØUR PAYRØLL	
B												
C												

MSK – MASK

DEFINITION – The instruction MSK is used to define initial constants in a file area. These constants are punched in the object deck and assigned to the file area, in the location defined by the relative position of the MSK.

EXPLANATION – Mask constants are entered in columns 32-43 and may be plus or minus. A minus constant is indicated by an over-punch in column 43. A mask may be assigned as single or multiple words as specified in the length field.

Mask constants are usually only assigned in the file declaration when they will not be changed or when an initial value is required. If more than one mask value is required for any mask word, it is necessary to store these values as normal constants either sequential or non-sequential (see explanation of CST) and place them in the mask area by program.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP WI *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	06000				FIL						0	LINE PRINTER
B	06050		06B		ASN		20					
C	06100		06C		ASN		09					
D	06150				MSK						1 1 1 1 1 1 1 1 1 1 1 1	END LINE ØR ALPHA
E												
F												

Sequence 06000 shows line printer file.

Sequence 06050 shows ASN with length of 20 in order to properly increment the counter so that the label assigned to the mask area will reference the correct memory location. The length of 20 is required regardless of the length of the line printer file as specified in the specifications card.

Sequence 06100 properly identifies the first mask location as 06C and the length of 09 would indicate that each of nine mask memories are used for more than one value.

Sequence 06150 defines a mask constant in the tenth position which will never be changed. This value is stored as a plus giving an end of line mask but can also be used as an alpha mask by transferring the value reverse to any other desired location.

OVR – OVERLAY

DEFINITION – OVR (Overlay) is used to permit assembly of additional programing over previously programed segments.

EXPLANATION – An overlay may be considered as an add-on or extension of program memory. With this pseudo it is possible to write additional program using the same file declarations, constants, and labels that were used in the primary program without redefining or reentering that data. Any number of overlay programs can be assembled with a primary program at one time. Even though one assembly is made for a primary and an overlay program, it should be recognized that two object decks (each to be loaded with a bootstrap) are in effect prepared. Additional labels or program points used with the overlay will be included in the total count against the allowable maximums.

Program will be loaded in memory beginning at the symbolic address identified in card columns 26-28 on the same line where the pseudo OVR is shown. The OVR must follow the primary program and will be concluded with an END pseudo. No END pseudo is to follow the primary program.

EXAMPLE – A totaling program illustrates a common usage of the overlay requirement. For the totals we, of course, need the accumulations developed from the primary program and we might also want to use the same file declarations and other constants.

LINE	SEQUENCE NO.	PRG PT	LABEL		OP CODE	MOD	LENGTH	ADDRESS		PP W'I *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS
			PG	R				PG	R			
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	42000				OVR			20	A			ØVERLAY FØR TØTALS
B	42050				CLR			00	K			
C	42100		42	C	TIX							
D	42150				TCP			15	J			TØTALS
E	42200				TIX							
F	42250				TPC			15	M			PRINT TANK
G	42300				TCP			10	B			LIMIT
H	42350				ACK			10	A			INCREMENT
J	42400				SKC			00	P			
K	42450				BRN			42	C			TEST FØR MØRE TØTALS
L												

Sequence 42000 shows an overlay to memory at the beginning of the primary program indicated by the symbolic address 20A. This would assure that all totals and constants, etc., would not be disturbed. The address of OVR is always the base label – incrementing or decrementing are not permitted.

Sequence 42050 – 42450 contain a loop program for moving the accumulated totals to working memory to print on the console. Notice that the label assigned with sequence 42100 would have to be one not used in the primary program since it is associated only with the overlay program. However the other labels given to the symbolic addresses for the totals, constants, and working areas are the same that were identified in the primary program. If special constants were needed for the overlay, they would still be assembled as required in the primary program.

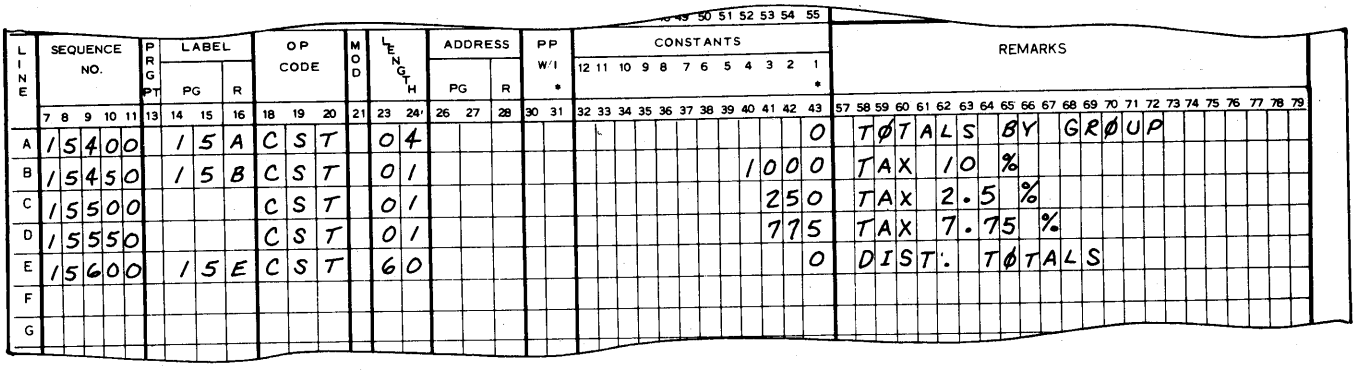
NOTE: See RSV (reserve) for additional consideration of overlay.

RSV – RESERVE

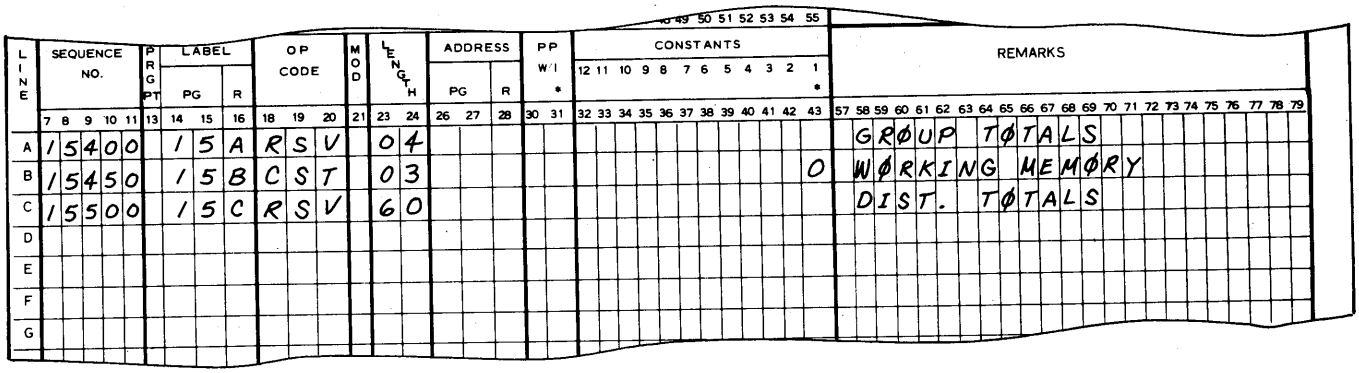
DEFINITION – The RSV instruction is used to set aside specific areas of memory without initializing that memory to generate a value in the object program.

EXPLANATION – To effectively use this pseudo, the programmer must be aware of the relative assignment of memory. RSV causes assembly exactly in the same manner as CST except that the memory assigned remains unchanged. This pseudo would be used when two programs are assembled separately but are to be combined in that data from the first program is to be saved for use in the second program. It would then be necessary to count words to determine where to locate the RSV instruction.

Because of the complexity of using this pseudo, it will not normally be used in application programs.



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	15400		15	A	CST		04				0	TØTALS BY GRØUP
B	15450		15	B	CST		01				1000	TAX 10 %
C	15500				CST		01				250	TAX 2.5 %
D	15550				CST		01				775	TAX 7.75 %
E	15600		15	E	CST		60				0	DIST. TØTALS
F												
G												



LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	15400		15	A	RSV		04					GRØUP TØTALS
B	15450		15	B	CST		03				0	WØRKING MEMØRY
C	15500		15	C	RSV		60					DIST. TØTALS
D												
E												
F												
G												

Sequence 15400 in the second program shows how we would save the group totals from the first program.

Sequence 15450 allows the second program to initialize what was the tax rates to a zero value to be used for working memory. Any values could have been placed in any of the three locations but no more than three constants could be entered since the relative relation of the distribution totals must be maintained.

Sequence 15500 allows us to save the distribution totals from the first program.

SECTION 3

SYMBOLIC PROGRAM WRITING PROCEDURE

To write a symbolic program for use with the assembly program, the programmer will normally follow this procedure:

1. Identify and number the coding sheets required for the program.
2. Make the entries on the first coding sheet that will:
    a. Provide the symbolic program card identification.
    b. Identify each page of the assembler print-out.
3. Make the entry on the first coding sheet that will be used to punch the specification card.
4. Declare the files used in the program.
    a. Alpha file
    b. Printer file
    c. Ledger file
    d. Card file
5. Write the "initialize" segment of the program.
    a. One time entry of data
    b. First time read of punched card (or) striped ledger
    c. Clear working locations
    d. Other
6. Write the "input" segment of the program.*
    a. End of run test
    b. Keyboard input
    c. Striped ledger input (Control Console)
    d. Striped ledger input (Ledger Reader)
    e. Punched card input
7. Write the "processing" segment of the program.*
    a. Arithmetic computations
    b. Accumulate data
    c. Move data to "output" areas
8. Write the "output" segment of the program.*
    a. Control console output
    b. Striped ledger output
    c. Line printer output
    d. Punched card (tape) output

9. Write the "end of run" segment
10. Write the "utility" routines required by the program.
    a. Read error
    b. Filled sheet at read ledger
    c. Filled sheet in run
    d. Write error
    e. Line printer bottom of form
11. Write routines associated with the main line program.
    a. Intermediate total routine
    b. Grand total routine
    c. New account routine
    d. End run filled sheet

*. Steps 6, 7, and 8 would normally be combined in an application program. They are considered separately here for ease of explanation.

SECTION 4

PROGRAM CODING EXAMPLE

PROBLEM:

Prepare salesman report of orders, update Accounts Receivable ledger, analyze order classification by fifty classes.

INPUT:

Punched Cards * - Sales and order detail from card reader one

Striped Ledger - Customer name and balances

OUTPUT:

Line Printer - Salesman activity report

Striped Ledger - Order detail and balance update

Console - Totals recap

Card Punch-New to-date summary cards

GENERAL

One punched card is read for each order and each customer ledger. Punched cards are sorted by salesman before posting with no more than 50 cards per salesman. Each line of detail is printed on the sales report along with the calculation of commission. Each salesman will start with a new line printer page. A posting will be made to an account ledger for each punched card. When the salesman number changes, totals for that salesman should print on the line printer report, a new summary card is punched with updated totals and a new report headed up.

When a total card (minus account number in salesman field) is read, the system should cause the salesman's totals to be taken on the report and then the daily totals should be printed on the console with description coming from memory for the major headings and then a print of 50 distribution totals (0-49).

SOLUTION INDEX:

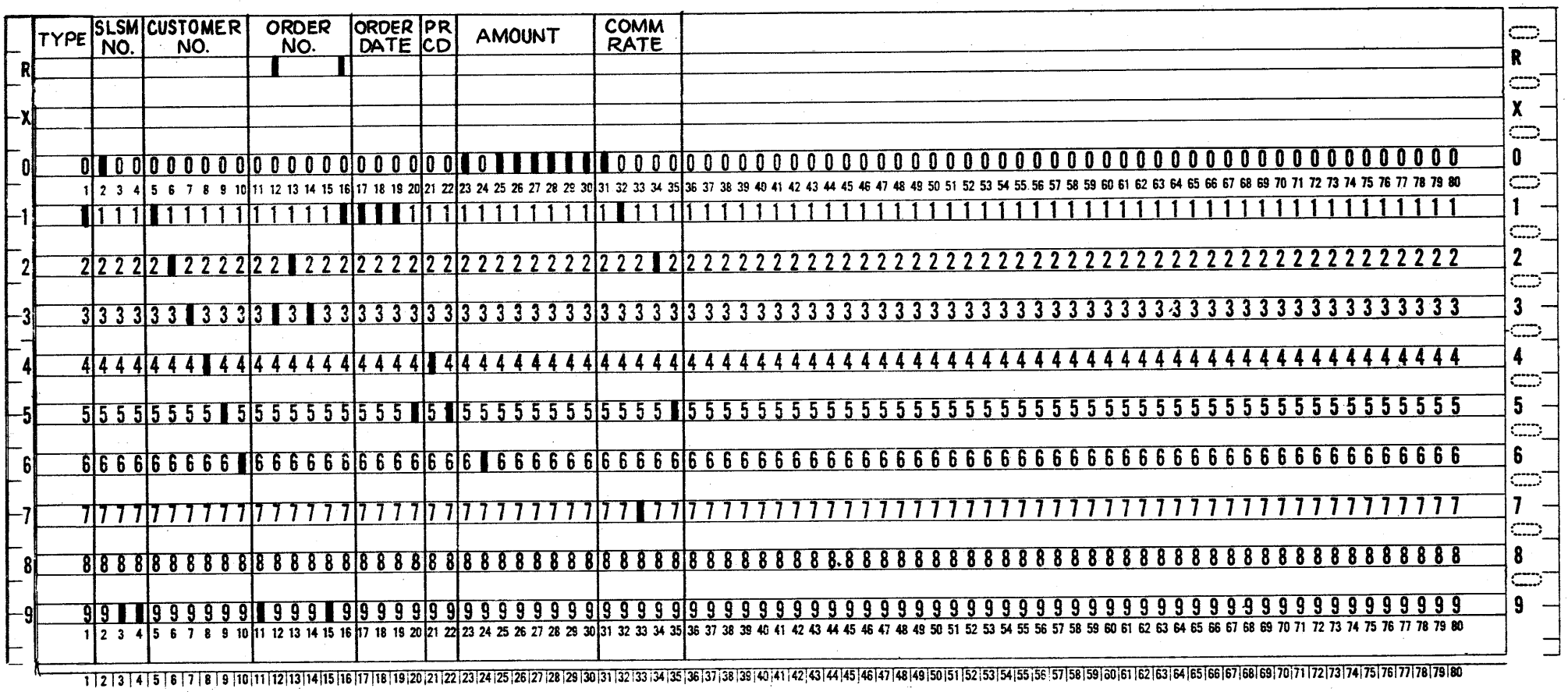
*To-date summary cards from card reader two.

LAYOUT FOR

80 COLUMN PUNCHED CARD

BURROUGHS DATA RECORDING EQUIPMENT

SAMPLE DETAIL CARD



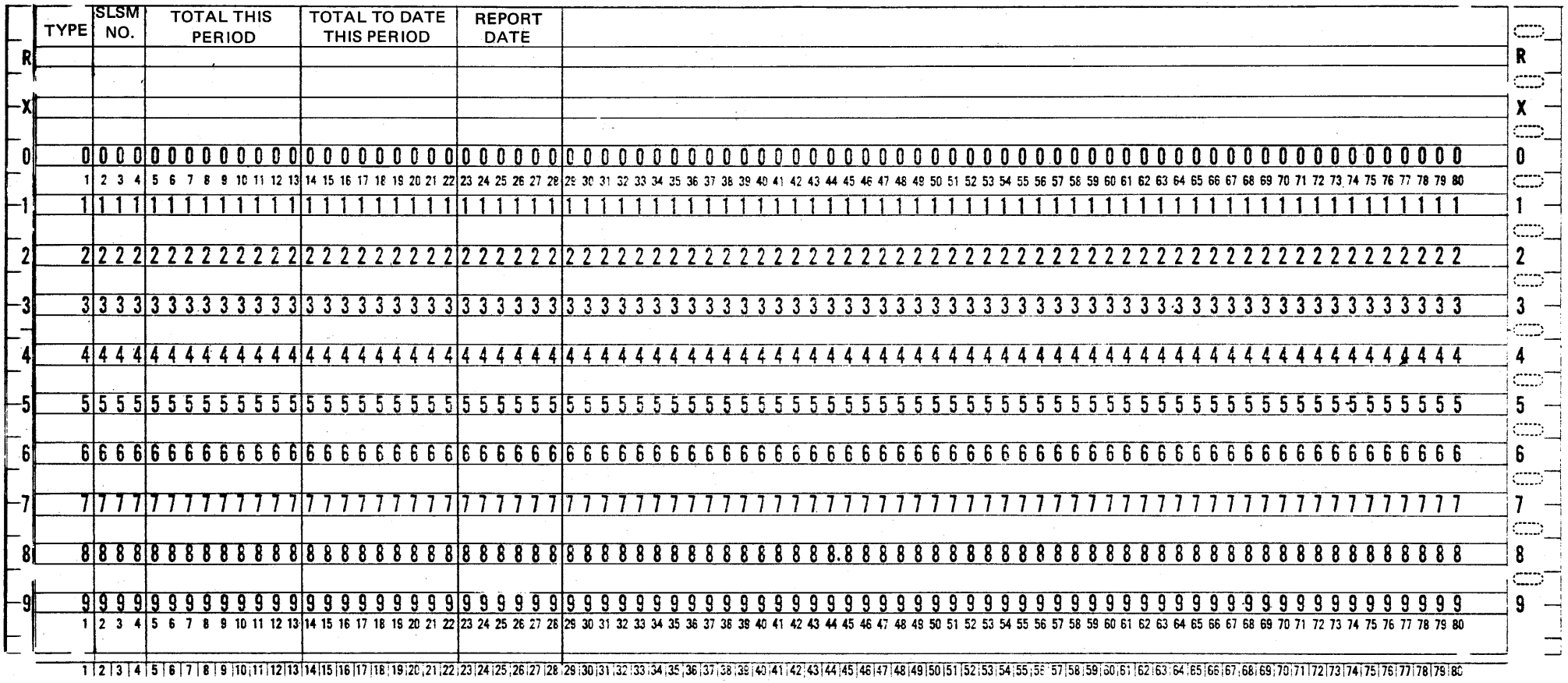
ONE PUNCHED CARD EACH CUSTOMER

LAYOUT FOR

80-COLUMN PUNCHED CARD

BURROUGHS DATA RECORDING EQUIPMENT

SAMPLE SUMMARY CARD



ONE PUNCHED CARD PER SALESMAN

STRIPED LEDGER CONTENT

0 = ACCOUNT NUMBER

1 = ACCOUNT BALANCE

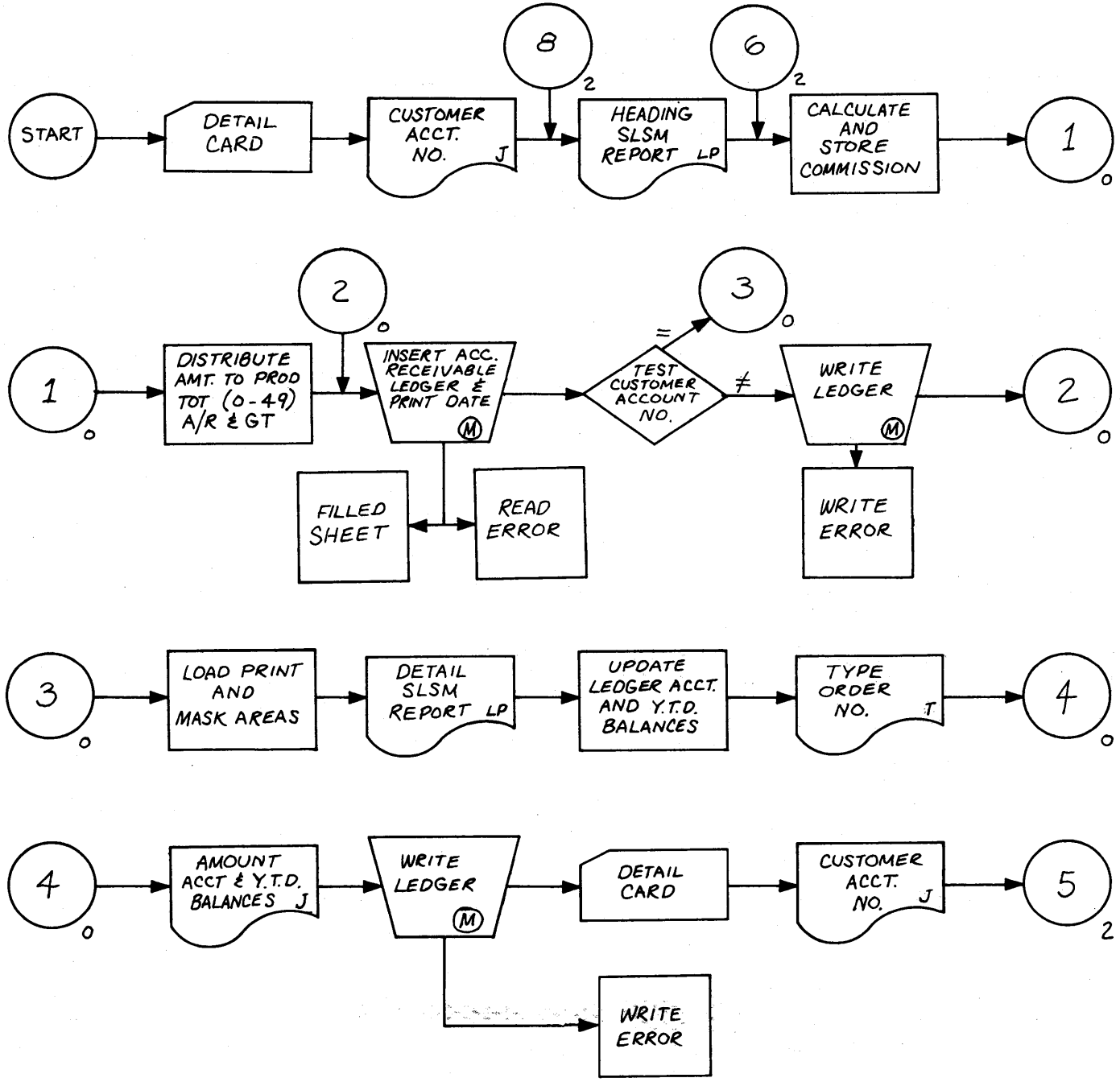
2 = PURCHASES Y.T.D.

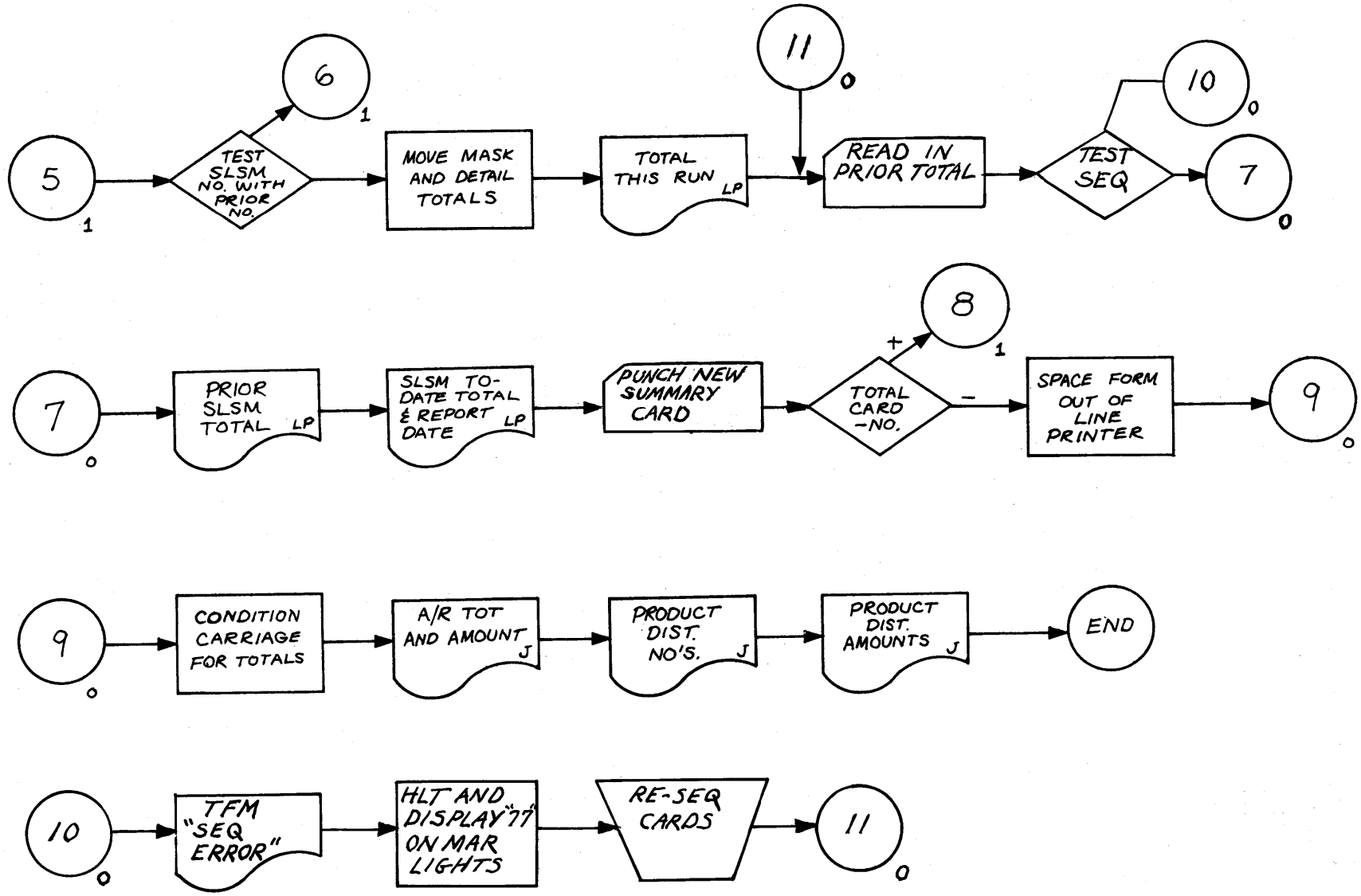
4–5, 14–15 = CUSTOMER NAME K 4

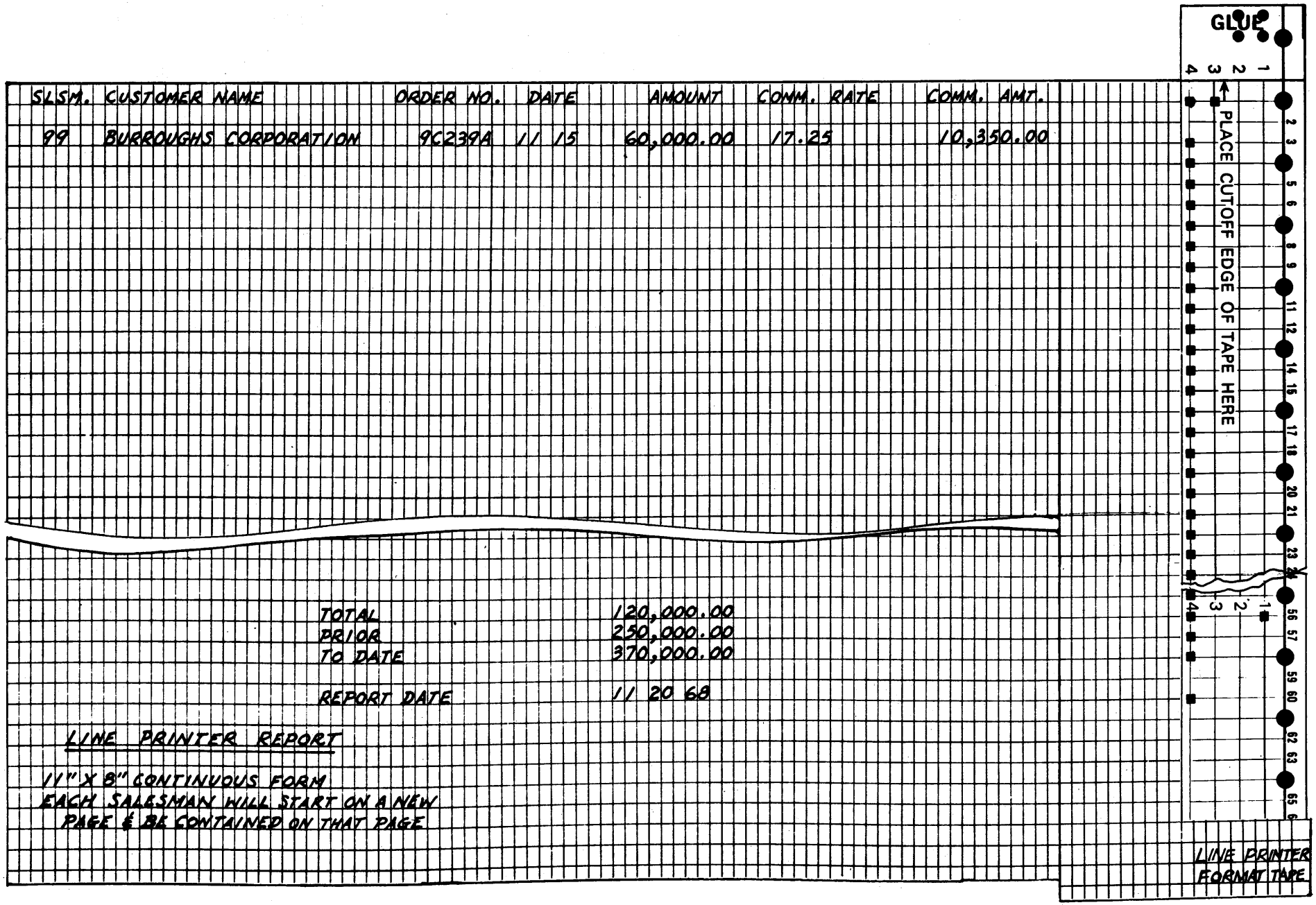
6–9, 16–19 = ADDRESS K 3 CITY & STATE K 2

JOURNAL REPORT TOTALS

(TFM)	"ACC REC TOT"	120,000.00	
(TFM)	"PROD TOTALS"	.00	500.00*
		.02	1000.00*
		.03	.00*
		.04	50.00*
		.49	400.00*


START
DETAIL CARD
CUSTOMER ACCT. NO. J
8 2
HEADING SLSM REPORT LP
6 2
CALCULATE AND STORE COMMISSION
1 0
1 0
DISTRIBUTE AMT. TO PROD TOT (0-49) A/R & GT
2 0
INSERT ACC. RECEIVABLE LEDGER & PRINT DATE M
FILLED SHEET
READ ERROR
TEST CUSTOMER ACCOUNT NO.
=
3 0
≠
WRITE LEDGER M
WRITE ERROR
2 0
3 0
LOAD PRINT AND MASK AREAS
DETAIL SLSM REPORT LP
UPDATE LEDGER ACCT. AND Y.T.D. BALANCES
TYPE ORDER NO. T
4 0
4 0
AMOUNT ACCT & Y.T.D. BALANCES J
WRITE LEDGER M
WRITE ERROR
DETAIL CARD
CUSTOMER ACCT. NO. J
5 2


5
1
TEST SLSM NO. WITH PRIOR NO.
6
1
MOVE MASK AND DETAIL TOTALS
TOTAL THIS RUN
LP
11
0
READ IN PRIOR TOTAL
TEST SEQ
10
0
7
0
7
0
PRIOR SLSM TOTAL
LP
SLSM TO-DATE TOTAL & REPORT DATE
LP
PUNCH NEW SUMMARY CARD
TOTAL CARD -NO.
+
8
1
-
SPACE FORM OUT OF LINE PRINTER
9
0
9
0
CONDITION CARRIAGE FOR TOTALS
A/R TOT AND AMOUNT
J
PRODUCT DIST. NO'S.
J
PRODUCT DIST. AMOUNTS
J
END
10
0
TFM "SEQ ERROR"
HLT AND DISPLAY "77" ON MAR LIGHTS
RE-SEQ CARDS
11
0


GLUE
PLACE CUTOFF EDGE OF TAPE HERE
SLSM. CUSTOMER NAME ORDER NO. DATE AMOUNT COMM. RATE COMM. AMT.
99 BURROUGHS CORPORATION 9C239A 11 15 60,000.00 17.25 10,350.00
TOTAL 120,000.00
PRIOR 250,000.00
TO DATE 370,000.00
REPORT DATE 11 20 68
LINE PRINTER REPORT
11" X 8" CONTINUOUS FORM
EACH SALESMAN WILL START ON A NEW PAGE & BE CONTAINED ON THAT PAGE
LINE PRINTER FORMAT TAPE

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
HØ 1	107	110	110	208
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
												"CØNTRØL REGISTER CØNSTANTS"
A	01000				HDR						SLSM REPØRT	
B	01050		01	B	CTL						RKCCLAP01	LØAD DATE CØLS 10 97643
C	01100		01	C	CTL						RDCP02	READ CUST ACC NØ
D	01150		01	D	CTL						RDLALNP03	READ LEDGER
E	01200		01	E	CTL						P04	KEY FØUR TFM
F	01250		01	F	CTL						P05	TYPE REFERENCE
G	01300		01	G	CTL						RDCLN4P06	PRINT AMØUNT
H	01350		01	H	CTL						RDCAPTP07	PRINT BALANCE
J	01400		01	J	CTL						WRLRDCAPTP0B	PRINT PURCH YTD
K	01450		01	K	CTL						WRLAMBP09	WRITE LEDGER
L	01500		01	L	CTL						P09	TFM TØ PØSITIØN CARR
M	01550		01	M	CTL						RKCAMBLN4P01	INDEX CØUNT
N	01600		01	N	CTL						VC1P03	SPACE PRINTER FØRM
P	01650		01	P	CTL						P03	PØSITIØN CARRIAGE
R	01700		01	R	CTL						RKCCLAAMBP04	READ PRIØR SALES TØ MEM
S	01750		01	S	CTL						VC3P03	SPACE FØRM PØSITIØN CRG
T	01800		01	T	CTL						AMBP03	CLØSE CARRIAGE
V	01850		01	V	CTL						RDCAMBP03	PRINT ACC REC TØT
W	01900		01	W	CTL						RDCAPTP05	PRINT PRØDUCT NØ
X	01950		01	X	CTL						RDCAMBAPTP07	PRINT PRØDUCT TØTAL

PAGE REF. 01

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4

CONSTANTS: 12 11 10 9 8 7 6 5 4 3 2 1 *

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18-19-20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
												"CØNTRØL REGISTER CØNSTANTS"
A	02000		02	A	CTL						ALNLN4AMBP03	INSERT LEDGER
B	02050		02	B	CTL						BLCAMBLN4P03	INSERT LEDGER
C	02100		02	C	CTL						WRCAMBP09	INSERT LEDGER
D	02150		02	D	CTL						WRLAMBP04	WRITE LEDGER
E	02200		02	E	CTL						WRLP08	WRITE CØRRECTIØN
F												
G												
H												
J												
K												
L												
M												
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 02

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	05000				FIL						/1	"FILE DECLARATIØN - ALPHA"
B	05050		05	B	ALP						SLSM CUSTØM	---RPT HEADINGS---
C	05100		05	C	ALP						ER NAME	---RPT HEADINGS---
D	05150		05	D	ALP						ØRDER	---RPT HEADINGS---
E	05200		05	E	ALP						NØ DATE	---RPT HEADINGS---
F	05250		05	F	ALP						AMØUNT	---RPT HEADINGS---
G	05300		05	G	ALP						CØM RATE	---RPT HEADINGS---
H	05350		05	H	ALP						CØM AMØUNT	---RPT HEADINGS---
J												
K												
L												
M												
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 05

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	06000				FIL						00	PRINTER FILE
B	06050		06	B	ASN		20					-ALPHA NUMERIC ØUTPUT
C	06100		06	C	MSK		10				0	--PRINT MSK
D												
E												
F												
G												
H												
J												
K												
L												
M												
N												
P												
R												
S												
T												
V												
W												
X												

REMARKS: "FILE DECLARATION- PRINTER"

PAGE REF. 06

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	07000				FIL						10	STRIPED LEDGER FILE "FILE DECLARATION - LEDGER"
B	07050		07	B	ASN		20					LØC 0 ACC NØ
C	07100											LØC 1 ACC BAL
D	07150											LØC 2 PURCH YTD
E	07200											LØC 3 BLANK
F	07250											LØC 4 14 NAME
G	07300											LØC 5 15 NAME K4
H	07350											LØC 6 16 ADDRESS
J	07400											LØC 7 17 ADDRESS K2 CIT
K	07450											LØC 8 18 ADDRESS K2 CIT
L	07500											LØC 9 19 ADDRESS K2 CIT
M												
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 07

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)
A	08000				FIL					
B	08050		08	B	MSK		08			
C	08100									
D	08150									
E	08200									
F	08250									
G	08300									
H	08350									
J	08400									
K	08450									
L	08500									
M	08550									
N	08600									
P	08650									
R										
S										
T										
V										
W										
X										

PAGE REF. 08

PRINTED IN U.S. AMERICA

PAGE HEADER, ALPHA, CONTROL REGISTER	1	2	3	4
44 45 46 47 48 49 50 51 52 53 54 55				

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *)	(32 33 34 35 36 37 38 39 40 41 42 43)
	1
	0
DETAIL CARD	
SUMMARY CARD	

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

REMARKS ALPHA NUM
"FILE DECLARATION-PUNCHED CARD"

(57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
PUNCH CARD ALPHA NUM
CARD CODE COL 1
SLSM NO 3 COL
CUSTOMER NO 6 COL
ORDER DATE 4 COL
PRODUCT CD 2 COL 0-49
AMOUNT 8 COL
COMM RATE 5 COL

CARD CODE COL 1
SLSM NO 3 COL
TOTAL THIS PERIOD 9 COL
TOTAL TO DATE 9 COL
REPORT DATE 6 COL

* NOTE: X overpunch for minus entries in these columns

FORM MKTG-2267 (1 68)

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	10000		10	A	CST						0	"NØN SEQUENTIAL CØNSTANTS"
B	10050		10	B	CST						0	REPØRT DATE
C	10100		10	C	CST						0	CØUNT
D	10150		10	D	CST						0	TØT SALES AMØUNT BY SLM
E	10200		10	E	CST						0	SALE AMØUNT GRAND TØT
F	10250		10	F	CST						0	CØMM WØRK LØCATIØN
G	10300		10	G	CST						0	SLSM NØ TEST LØCATIØN
H	10350		10	H	CST						0	PRØD NUMBER PRINT
J	10400		10	J	CST						50	NUMBER ØF PRØDUCTS 0-49
K	10450		10	K	CST						0	END RØUTINE TEST MA
L	10500		10	L	CST						912345000000	KEY CØDE CØNSTANTS
M											0	MØNTH CODE FØR TEST
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 10

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER			
1	2	3	4
44 45 46 47 48 49 50 51 52 53 54 55			

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS 12 11 10 9 8 7 6 5 4 3 2 1 *	REMARKS "SEQUENTIAL CONSTANTS"
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	15000		15	A	APH						TØTAL	---RPT SUB HEADINGS---
B	15050				APH						PRIØR	---RPT SUB HEADINGS---
C	15100				APH						TØ DATE	---RPT SUB HEADINGS---
D	15150				APH						REPØRT DATE	---RPT SUB HEADINGS---
E	15200				APH						ACC REC TØT	CØNSØLE HEADING
F	15250				APH						PRØD TØTALS	CØNSØLE HEADING
G	15300				APH						MANUAL ENT	RECØNSTRUCT
H	15350				APH						BAL FØRWARD	FILLED SHEET
J	15360				APH						SEQ ERRØR K1	
K	15400		15	J	CST		01				1111111111111X	ALPHA MASK
L	15450				CST		01				4444444444007X	NUMERIC
M	15500				CST		01				4444444444666X	MASK
N	15550				CST		01				1111111111116X	
P	15600				CST		01				4444466660577X	CØNSTANTS
R	15650				CST		01				4666003000028X	NUMERIC
S	15700				CST		01				4444460007277X	MASK
T	15750				CST		01				6666666666666X	
V	15800				CST		01				4444466666666X	CØNSTANTS
W	15850				CST		01				3003003000027X	NUMERIC
X	15900				CST		01				4477677677666X	MASK

PAGE REF. 15

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

IDENT. (3 4 5 6) | ALPHA (7 8 9) | PRINTER (10 11 12) | LEDGER (13 14 15) | CARD (16 17 18)

PAGE HEADER, ALPHA, CONTROL REGISTER (1 2 3 4) (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "SEQUENTIAL CONSTANTS" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	16000				CST		01				1000000000000	END OF LINE CONSTANT
B	16050				CST		01				1111114444444X	ALPHA MASK
C	16100		168		CST		50				0	PRODUCT DISTRIBUTION
D												
E												
F												
G												
H												
J												
K												
L												
M												
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 16

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	20000		20	A	RFL		07					RESET FLAGS
B	20050				LCR			01	B			CLA RKC P01
C	20100				HLT			10	A		INDEX DATE	
D	20150				SFL		01					SET BAR 1
E	20200				LCR			01	M			RKC AMB LN4 P01
F	20250				HLT			10	B		INDEX COUNT	
G	20300				RFL		01					RESET BAR 1
H	20350				TCP			10	K			MØVE KEY CØDE CST
J	20400				TPC			00	B			MØVE FØR HIGH ØRDER
K	20450				SLT	S	01	00	P			PØSITIØN FØR KEY ØNE
L	20500				SRT	S	06	00	P			PØSITIØN LØW ØRDER
M	20550				SRT	S	06	00	B			PØSITIØN HIGH ØRDER
N	20600				TPC			08	B	03		LØAD CARD FILE WITH CD
P	20650				TCP			00	B			MØVE HIGH ØRDER
R	20700				TPC			08	B	13		LØAD HIGH ØRDER
S	20750				TCP			15	A	12		MØVE DESC MANUAL ENTRY
T	20800				SRT	S	02	00	P			DRØP BLANKS
V	20850				SLT	D	12	00	P			PØSITIØN IN K
W	20900				TCP			10	K			MØVE KEY CØDE CØNSTANTS
X	20950				TPC			00	B			MØVE FØR HIGH ØRDER

PAGE REF. 20

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT. (3–6)	ALPHA (7–9)	PRINTER (10–12)	LEDGER (13–15)	CARD (16–18)

PAGE HEADER, ALPHA, CONTROL REGISTER (44–55)

1	2	3	4

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7–11)	PRG PT (13)	LABEL PG (14–15)	LABEL R (16)	OP CODE (18–20)	MOD (21)	LENGTH (23–24)	ADDRESS PG (26–27)	ADDRESS R (28)	PP W/I * (30–31)	CONSTANTS (32–43)	REMARKS "PRØGRAM" (57–79)
A	21000				SLT	S	04	00	P			PØSITIØN FØR 4
B	21050				SLT	D	02	00	P			LØW ØRDER TØ K
C	21100				TKC			15	A	12		LØAD IN MEM
D	21150				TCK			15	A	13		MØVE HIGH ØRDER DESC
E	21200				SRT	D	02	00	P			DRØP BLANKS
F	21250				SLT	D	02	00	B			LØAD KEY CØDE
G	21300				TKC			15	A	13		LØAD IN MEMØRY
H	21350				TCP			15	A	12		MØVE DESC & KEY CØDE
J	21400				SLT	S	10	00	P			ISØLATE KEY CØDE
K	21450				TCK			15	A	14		MØVE DESC BAL FØRWARD
L	21500				SRT	D	01	00	B			DRØP BLANK
M	21550				SLT	D	01	00	P			MØVE KEY CØDE
N	21600				TKC			15	A	14		LØAD IN MEMØRY
P	21650				TCP			15	A	13		MØVE DESC & KEY CØDE
R	21700				SLT	S	10	00	P			ISØLATE KEY CØDE
S	21750				TCK			15	A	15		MØVE DESC HIGH ØRDER
T	21800				SRT	D	01	00	B			DRØP BLANK
V	21850				SLT	D	01	00	P			MØVE KEY CØDE
W	21900				TKC			15	A	15		LØAD IN MEMØRY
X	21950				TCP			15	A	09		MØVE DESC HIGH ØRDER

PAGE REF. 21

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT. (3 4 5 6)	ALPHA (7 8 9)	PRINTER (10 11 12)	LEDGER (13 14 15)	CARD (16 17 18)

PAGE HEADER, ALPHA, CONTROL REGISTER	1	2	3	4
(44 45 46 47 48 49 50 51 52 53 54 55)				

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER _____

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP (30 31)	WI *	CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PRØGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	22000				SRT	S	01	00	P				DRØP BLANK
B	22050				SLT	S	01	00	P				REPØSITIØN DESC
C	22100				TPC			15	A	09			LØAD IN MEMØRY
D	22150				TCP			15	A	11			MØVE DESC HIGH ØRDER
E	22200				SRT	S	01	00	P				DRØP BLANK
F	22250				SLT	S	01	00	P				REPØSITIØN DESC
G	22300				TPC			15	A	11			LØAD IN MEMØRY
H	22350				CRD			08	B				READ PUNCHED CARD
J	22400				LCR			01	C				RDC P02
K	22450				PAC			08	B	02			PRINT CUST ACC NØ
L	22500		22	L	TCP	S		05	B				MØVE PAGE HEADING
M	22550				TCK			05	B	10			MØVE HIGH ØRDER
N	22600				TPC			06	B				LØAD PRINT TANK
P	22650				TKC			06	B	10			LØAD PRINT HIGH ØRDER
R	22700				TCP			05	C				MØVE PAGE HEADING
S	22750				TCK			05	C	10			MØVE HIGH ØRDER
T	22800				TPC			06	B	01			LØAD PRINT TANK
V	22850				TKC			06	B	11			LØAD PRINT HIGH ØRDER
W	22900				TCP			05	D				MØVE PAGE HEADING
X	22950				TCK			05	D	10			MØVE HIGH ØRDER

PAGE REF. 22

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PRØGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	23000				TPC			06	B	02		LØAD PRINT TANK
B	23050				TKC			06	B	12		LØAD PRINT HIGH ØRDER
C	23100				TCP			05	E			MØVE PAGE HEADING
D	23150				TCK			05	E	10		MØVE HIGH ØRDER
E	23200				TPC			06	B	03		LØAD PRINT TANK
F	23250				TKC			06	B	13		LØAD PRINT HIGH ØRDER
G	23300				TCP			05	F			MØVE PAGE HEADING
H	23350				TCK			05	F	10		MØVE PAGE HIGH ØRDER
J	23400				TPC			06	B	04		LØAD PRINT TANK
K	23450				TKC			06	B	14		LØAD PRINT HIGH ØRDER
L	23500				TCP			05	G			MØVE PAGE HEADING
M	23550				TCK			05	G	10		MØVE PAGE HIGH ØRDER
N	23600				TPC			06	B	05		LØAD PRINT TANK
P	23650				TKC			06	B	15		LØAD PRINT HIGH ØRDER
R	23700				TCP			05	H			MØVE PAGE HEADING
S	23750				TCK			05	H	10		MØVE PAGE HIGH ØRDER
T	23800				TPC			06	B	06		LØAD PRINT TANK
V	23850				TKC			06	B	16		LØAD PRINT HIGH ØRDER
W	23900				TCP			15	J			MØVE MASK WØRD
X	23950				TPC			06	C			LØAD MASK WØRD ZERØ

PAGE REF. 23

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT. (3 4 5 6)	ALPHA (7 8 9)	PRINTER (10 11 12)	LEDGER (13 14 15)	CARD (16 17 18)

PAGE HEADER, ALPHA, CONTROL REGISTER (1 2 3 4)

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (44 45 46 47 48 49 50 51 52 53 54 55) (32 33 34 35 36 37 38 39 40 41 42 43)

REMARKS "PRØGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	REMARKS
A	24000				TPC			06	C	01	LØAD MASK WØRD ØNE
B	24050				TPC			06	C	02	LØAD MASK WØRD TWØ
C	24100				TPC			06	C	03	LØAD MASK WØRD THREE
D	24150				TPC			06	C	04	LØAD MASK WØRD FØUR
E	24200				TPC			06	C	05	LØAD MASK WØRD FIVE
F	24250				TPC			06	C	06	LØAD MASK WØRD SIX
G	24300				TCP			15	J	11	MØVE MASK WØRD E ØF L
H	24350				TPC			06	C	07	LØAO E ØF L MASK
J	24400				PRT			06	B		PRINT HEADINGS
K	24450				NØP						NØ ØPERATIØN
L	24500		24	L	TCP			08	B	06	MØVE AMØUNT TO WØRK
M	24550				MCP			08	B	07	CALC CØMM
N	24600				SRT	S	04	00	P		PØSITIØN RESULT
P	24650				TPC			10	E		STØRE CØMM AMØUNT
R	24700				TCP			08	B	06	MØVE AMØUNT TØ WØRK
S	24750		00	@	TCK			08	B	05	MØVE DIST CØDE TØ WØRK
T	24800				TIX						TURN ØN INDEXING
V	24850				APC			16	B		DISTRIBUTE AMØUNT
W	24900				APC			10	C		ACCUM ACC REC AMØUNT
X	24950				APC			10	D		ACCUM GRAND TØTAL

PAGE REF. 24

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TR

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT. (3 4 5 6)	ALPHA (7 8 9)	PRINTER (10 11 12)	LEDGER (13 14 15)	CARD (16 17 18)

PAGE HEADER, ALPHA, CONTROL REGISTER (44 45 46 47 48 49 50 51 52 53 54 55)

1	2	3	4

CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)

12	11	10	9	8	7	6	5	4	3	2	1 *

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	25000		25	A	LCR			01	D			RDL ALN P03
B	25050				RFL		02					RESET BAR 3
C	25100				CLR			00	P			CLEAR PRINT AREA
D	25150				PAC			00	P		INSERT CARD	PRINT DATE
E	25200				SFL		80					TEST ERROR INDICATORS
F	25250				BRU			37	F			BR TO READ ERROR ROUT
G	25300				BRU			39	F			BR TO FILLED SHEET ROUT
H	25350				TCP			08	B	02		MOVE CUST NO TO WORK
J	25400				SCP			07	B			SUBT CUST NO
K	25450				BRN			40	D			BR IF NOT EQUAL
L	25500				TCP			08	B	01		MOVE SLSM NO TO WORK
M	25550				TPC			06	B			LOAD PRINT TANK
N	25600				TCP			07	B	04		MOVE NAME TO WORK
P	25650				TCK			07	B	14		MOVE HIGH ORDER
R	25700				TPC			06	B	02		LOAD PRINT TANK
S	25750				TKC			06	B	12		LOAD HIGH ORDER
T	25800				TCP			07	B	05		MOVE NAME TO WORK
V	25850				TCK			07	B	15		MOVE HIGH ORDER
W	25900				TPC			06	B	03		LOAD PRINT TANK
X	25950				TKC			06	B	13		LOAD HIGH ORDER

PAGE REF. 25

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER			
1	2	3	4

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS	REMARKS "PRØGRAM"
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	26000				TCP			08B		03		MØVE ØRDER NØ LØW ØRDER
B	26050				TCK			08B		13		MØVE ØRDER NØ HGH ØRDER
C	26100				TPC			06B		04		LØAD PRINT TANK
D	26150				TKC			06B		14		LØAD HIGH ØRDER
E	26200				TCP			08B		04		MØVE DATE TØ WØRK
F	26250				TPC			06B		05		LØAD PRINT TANK
G	26300				TCP			08B		06		MØVE AMØUNT TØ WØRK
H	26350				TPC			06B		06		LØAD PRINT TANK
J	26400				TCP			08B		07		MØVE CØM RATE TØ WØRK
K	26450				TPC			06B		07		LØAD PRINT TANK
L	26500				TCP			10E				MØVE CØM AMT TØ WØRK
M	26550				TPC			06B		08		LØAD PRINT TANK
N	26600				TCP			15J		01		MØVE MASK CNST
P	26650				TPC			06C				LØAD MASK WØRD ZERØ
R	26700				TCP			15J		02		MØVE MASK CNST
S	26750				TPC			06C		01		LØAD MASK WØRD ØNE
T	26800				TCP			15J				MØVE MASK CNST
V	26850				TPC			06C		02		LØAD MASK WØRD TWØ
W	26900				TCP			15J		03		MØVE MASK CNST
X	26950				TPC			06C		03		LØAD MASK WØRD THREE

PAGE REF. 26

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT (3-6)	ALPHA (7-9)	PRINTER (10-12)	LEDGER (13-15)	CARD (16-18)

PAGE HEADER, ALPHA, CONTROL REGISTER (44-55)

1	2	3	4

CONSTANTS (32-43)

12	11	10	9	8	7	6	5	4	3	2	1 *

LINE	SEQUENCE NO. (7-11)	PRG PT (13)	LABEL PG (14-15)	LABEL R (16)	OP CODE (18-20)	MOD (21)	LENGTH (23-24)	ADDRESS PG (26-27)	ADDRESS R (28)	PP W/I * (30-31)	REMARKS (57-79)
											"PROGRAM"
A	27000				TCP			15	J	12	MOVE MSK CST
B	27050				TPC			06	C	04	LOAD MASK WORD FOUR
C	27100				TCP			15	J	04	MOVE MASK CNST
D	27150				TPC			06	C	05	LOAD MASK WORD FIVE
E	27200				TCP			15	J	05	MOVE MASK CNST
F	27250				TPC			06	C	06	LOAD MASK WORD SIX
G	27300				TPC			06	C	08	LOAD MASK WORD EIGHT
H	27350				TCP			15	J	06	MOVE MASK CNST
J	27400				TPC			06	C	07	LOAD MASK WORD SEVEN
K	27450				TCP			15	J	11	MOVE MASK CNST EDL
L	27500				TPC			06	C	09	LOAD MASK WORD NINE
M	27550				PRT			06	B		PRINT ON PRINTER
N	27600				NOP						NO OPERATION NO BOF TST
P	27650				TCP			08	B	06	MOVE AMOUNT TO WORK
R	27700				APC			07	B	01	UPDATE ACCOUNT BAL
S	27750				APC			07	B	02	UPDATE PURCH YTD
T	27800				LCR			01	E		P04
V	27850				TCP			10	K		MOVE KEY CODE CST
W	27900				TPC			00	B		MOVE FOR HIGH ORDER
X	27950				SLT	S	04	00	P		POSITION FOR KEY A

PAGE REF. 27

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	28000				TFM			00	P		POSITION CARRIAGE
B	28050				LCR			01	F		P05
C	28100				TFM			08	B	03	TYPE ORDER NO
D	28150				LCR			01	G		RDC LNA P06
E	28200				PAC			08	B	06	PRINT DEBIT
F	28250				LCR			01	H		RDC APT P07
G	28300				PAC			07	B	01	PRINT BALANCE
H	28350				LCR			01	J		RDC WRL APT P08
J	28400				TCP			07	B	02	MOVE PURCH YTD TO WORK
K	28450				TPC			00	W		LOAD PURCH YTD IN W
L	28500		28	L	PAC			00	W		PRINT PURCH YTD
M	28550				SFL		80				TEST FOR WRITE ERROR
N	28600				NOP						NO OPERATION
X	28650				BRU			39	S		BR TO WRITE CORRECTION
R	28700				LCR			01	L		P09
S	28750				TCP			10	K		MOVE KEY CODE CST
T	28800				TPC			00	B		MOVE FOR HIGH ORDER
V	28850				SLT	S	02	00	P		POSITION FOR KEY 2
W	28900				TFM			00	P		POSITION CARR WITH K2
X	28950				TCP			08	B	01	MOVE SLSM NO

PAGE REF. 28

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PRØGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	29000				TPC			10	F			STØRE SLSM NØ
B	29050				CRD			08	B			READ PUNCHED CARD
C	29100				LCR			01	C			RDC P02
D	29150				PAC			08	B	02		PRINT CUSTØMER NØ
E	29200				TCP			10	F			MØVE SLSM NØ LAST CARD
F	29250				SCP			08	B	01		SUBT SLSM NØ THIS CARD
G	29300				BRN					1		BR IF NEW SALESMAN
H	29350				BRU			24	L			BR TØ CAL CØMM
J	29400	1			TCP			08	B			MØVE CARD CØDE
K	29450				PFM	S	11	00	P			PFM CARD TYPE
L	29500				TCP			08	B	01		MØVE SLSM NUMBER
M	29550				PFM	S	09	00	P		PFM CRD CØDE	AND SLSM NØ
N	29600				TCP			15	J	11		MØVE MASK CST
P	29650				TPC			06	C			LØAD MASK WØRD
R	29700				LCR			01	N			VCI P03
S	29750				PRT			06	B			SPACE FØRM NØ PRINT
T	29800				NØP							NØ ØPERATIØN
V	29850				TCP			15	A			MØVE DESC TØTAL
W	29900				TPC			06	B	02		LØAD PRINT TANK
X	29950				TCP			15	A	01		MØVE DESC TØTAL

PAGE REF. 29

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE. X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	30000				TPC			06	B	12		LØAD PRINT TANK
B	30050				TCP			10	C			MØVE SALM TØTAL
C	30100				TPC			06	B	04		LØAD PRINT TANK
D	30150				PFM	S	03	00	P		PFM TØTAL TH	IS PERIØD
E	30200				TCP			15	J	07		MØVE MASK CNST
F	30250				TPC			06	C			LØAD MASK WØRD ZERØ
G	30300				TPC			06	C	01		LØAD MASK WØRD ZERØ
H	30350				TCP			15	J			MØVE MASK CNST
J	30400				TPC			06	C	02		LØAD MASK WØRD TWØ
K	30450				TCP			15	J	08		MØVE MASK CNST
L	30500				TPC			06	C	03		LØAD MASK WØRD THREE
M	30550				TCP			15	J	09		MØVE MASK CØNSTANT
N	30600				TPC			06	C	04		LØAD MASK WØRD FØUR
P	30650				TCP			15	J	11		MØVE E ØF L MSK CØNST
R	30700				TPC			06	C	05		LØAD E ØF L MSK WØRD
S	30750				LCR			01	P			PØ3
T	30800				PRT			06	B			PRINT ØN PRINTER
V	30850				NØP							NØ ØPERATIØN
W	30900				SFL		09					SET BAR 1 LANE 3
X	30950				CLR			00	P			CLEAR PRINT AREA

PAGE REF. 30

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	31000				PAC			00	P			PØSITIØN CARRIAGE
B	31050				RFL		09					RESET FLAGS
C	31100				TCP			10	A			MØVE REPØRT DATE
D	31150				SRT	S	04	00	P			PØSITIØN MØ BY ITSELF
E	31200				SCP			10	L			TEST FØR MØNTH 1
F	31250				BRN			31	L			BRANCH TØ READ PRIØR
G	31300											SALES TØ DATE FRØM CARD
H	31350											READER NØ-2
J	31400				BRU					2		BRANCH TØ PRT NØ PRIØR
K	31450											SALES TØ DATE
L	31500		31	L	TCK			10	F			STØRE ØLD SLSM NØ IN K
M	31550				TCP			08	B	1		MØVE ØLD SLSM NØ
N	31600				TPC			10	F			STØRE AS CST
P	31650				CRD			08	B	02	FEED ØNE CRD	FRØM CRD READER NØ 2
R	31700				TCP			08	B	1		MØVE SLSM NØ
S	31750				SCP			00	K			TEST TØ SEE IF SLSM
T	31800											DETAIL AND PRIØR CARDS
V	31850											ARE MATCHED IF THE SAME
W	31900				BRN					1		BR TØ SEQUENCE ERRØR
X	31950				BRU					3		BR TØ CØNTINUE

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

PAGE REF. 31

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	32000	1			TCP			15	A	17		MØVE HIGH ØRDER
B	32050				TPC			00	B			STØRE IN OOB
C	32100				TCP			15	A	16		MØVE LØW ØRDER
D	32150				TFM			00	P		SEQUENCE ERR	MAR LIGHTS-INDICATØR
E	32200				NØP							
F	32250				LCR			01	W			PO5
G	32300				HLT			00	P	77	HANDLE SEQ	ERRØR-RELØAD READER
H	32350				BRU			31	L			READ ANØTHER PRIØR CARD
J	32400	2			CLR			08	B	03		CLEAR THIS AREA IF NØ
K	32450											PRIØR MONTHS SUMMARY CD
L	32500	3			TCP			08	B	03		MØVE PRIØR SALES-TØ-DAT
M	32550				TPC			06	B	04		LØAD PRINT TANK
N	32600				APC			10	C			ACCUMULATE TØ DATE
P	32650				RFL		09					RESET FLAGS
R	32700				TCP			15	A	02		MØVE DESC PRIØR
S	32750				TCK			15	A	03		MØVE DESC PRIØR
T	32800				TPC			06	B	02		LØAD PRINT TANK
V	32850				TKC			06	B	12		LØAD PRINT HIGH ØRDER
W	32900				PRT			06	B			PRINT PRIØR TØTAL
X	32950				NØP							NØ ØPERATIØN

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

PAGE REF. 32

* NOTE: X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER			
1	2	3	4
44 45 46 47 48 49 50 51 52 53 54 55			

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO.	PRG PT	LABEL PG	LABEL R	OP CODE	MOD	LENGTH	ADDRESS PG	ADDRESS R	PP W/I *	CONSTANTS	REMARKS
	7 8 9 10 11	13	14 15	16	18 19 20	21	23 24	26 27	28	30 31	32 33 34 35 36 37 38 39 40 41 42 43	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
A	33000				TCP			15	A	04		MØVE DESC TØ DATE
B	33050				TCK			15	A	05		MØVE DESC TØ DATE
C	33100				TPC			06	B	2		LØAD PRINT TANK
D	33150				TKC			06	B	12		LØAD PRINT HIGH ØRDER
E	33200				TPC			10	C			MØVE TØ DATE TØTAL
F	33250				CLR			10	C			CLEAR WØRK AREA
G	33300				TPC			06	B	04		LØAD PRINT TANK
H	33350				PFM	S	03	00	P		PFM TØTAL TØ	DATE
J	33400				PRT			06	B			PRINT TØ DATE
K	33450				NØP							NØ ØPERATIØN
L	33500				TCP			15	J	07		MØVE MASK CST
M	33550				TPC			06	C	03		LØAD MASK WØRD THREE
N	33600				TCP			15	J	10		MØVE MASK CST
P	33650				TPC			06	C	04		LØAD MASK WØRD FØUR
R	33700				TCP			15	A	06		MØVE DESC REPØRT DATE
S	33750				TCK			15	A	07		MØVE DESC REPØRT DATE
T	33800				TPC			06	B	02		LØAD PRINT TANK
V	33850				TKC			06	B	12		LØAD PRINT HIGH ØRDER
W	33900				TCP			10	A			MØVE DATE TØ WØRK
X	33950				TPC			06	B	04		LØAD PRINT TANK

PAGE REF. 33

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

PRINTED IN U.S. AMERICA

FORM MKTG-2267 (1/68)

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	34000				PFM	S	06	00	P		PFM REPORT	DATE
B	34050				REL							RELEASE NEW PRIØR CARD
C	34100				PRT			06	B			PRINT REPØRT DATE
D	34150				NØP							NØ ØPERATIØN
E	34200				TCP			08	B	01		MØVE SLSM NØ
F	34250				BRM					1		IF MINUS END ØF RUN
G	34300				BRU			22	L			BR TØ HEAD FØRM NEW MAN
H	34350				TCP			15	J	11		MØVE MSK CNST
J	34400				TPC			06	C			LØAD MSK WØRD
K	34450				LCR			01	S			VC3 P03
L	34500				PRT			06	B			SPACE FØRM NØ PRINT
M	34550				NØP							NØ ØPERATIØN
N	34600				PRT			06	B			SPACE FØRM NØ PRINT
P	34650				NØP							NØ ØPERATIØN
R	34700				PRT			06	B			SPACE FØRM NØ PRINT
S	34750				NØP							NØ ØPERATIØN
T	34800				PRT			06	B			SPACE FØRM NØ PRINT
V	34850				NØP							NØ ØPERATIØN
W	34900				PRT			06	B			SPACE FØRM NØ PRINT
X	34950				NØP							NØ ØPERATIØN

PAGE REF 3 4

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	CONSTANTS (12 11 10 9 8 7 6 5 4 3 2 1 *) (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	35000				SFL		02					SET FLAG FOR BAR 3
B	35050				LCR			01	T			AMB P03
C	35100				CLR			00	P			CLEAR PRINT AREA
D	35150				PAC			00	P			CLOSE CARRIAGE
E	35200				TCP			10	K			MOVE KEY CODE CNST
F	35250				TPC			00	B			MOVE FOR HIGH ORDER
G	35300				SLT	S	01	00	P			POSITION FOR KEY 1
H	35350				SRT	S	11	00	P			POSITION KEY CODE
J	35400				SRT	S	11	00	B			POSITION KEY CODE
K	35450				ACP			15	A	08		MOVE LOW ORDER DESC
L	35500				TCK			15	A	09		MOVE HIGH ORDER DESC
M	35550				AKC			00	B			SET FOR TYPE
N	35600				TFM			00	P			TYPE ACC REC TOT
P	35650				LCR			01	V			RDC AMB P03
R	35700				PAC			10	D			PRINT ACC REC TOTAL
S	35750				TCP			10	K			MOVE KEY CODE CNST
T	35800				TPC			00	B			MOVE FOR HIGH ORDER
V	35850				SLT	S	04	00	P			POSITION FOR KEY 4
W	35900				SRT	S	11	00	P			POSITION KEY CODE
X	35950				SRT	S	11	00	B			POSITION KEY CODE

PAGE REF. 35

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER

1	2	3	4

44 45 46 47 48 49 50 51 52 53 54 55

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
												"PROGRAM"
A	36000				ACP			15	A	10		MOVE LOW ORDER DESC
B	36050				TCK			15	A	11		MOVE HIGH ORDER DESC
C	36100				AKC			00	B			SET FOR TYPE
D	36150				TFM			00	P			TYPE PROD TOTALS
E	36200				CLR			00	K			CLEAR K
F	36250		36	F	TKC			10	G			LOAD COUNT IN PRINT TK
G	36300				TCP			10	G			MOVE PROD NO WORK
H	36350				TPC			10	J			MOVE PROD NO TO TEST MA
J	36400				LCR			01	W			RDC APT POS
K	36450				RFL		02					RESET FLAG 2
L	36500				PAC			10	G			PRINT PROD NO 0-49
M	36550				LCR			01	X			RDC AMB APT P07
N	36600				SFL		09					SET FOR BAR 1 LANE 3
P	36650		00	@	TCK			10	G			MOVE PROD NO TO K
R	36700				TIX							TURN ON INDEXING
S	36750				PAC			16	B			PRINT PRODUCT TOTAL
T	36800				RFL		09					RESET PROGRAM FLAGS
V	36850				TCP			10	J			PROD NO TO P
W	36900				ACP			10	B			ADD ONE
X	36950				TPC			00	K			MOVE NXT PROD NO TO K

PAGE REF. 36

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	37000				SCP			10	H			SUBT LIMIT ØF 50
B	37050				BRN			36	F			BR FØR NEXT PRØDUCT
C	37100				CLR			00	P			CLEAR
D	37150				LCR			00	P			CLEAR CØNTRØL REGISTER
E	37200				HLT			00	P			END RUN
F	37250		37	F	LCR			01	P		READ ERRØR	PØ3
G	37300				TCP			10	K			MØVE KEY CØDE CST
H	37350				TPC			00	B			MØVE FØR HIGH ØRDER
J	37400				SLT	S	01	00	P			PØSITIØN FØR KEY 1
K	37450				TFM			00	P			KEY ØNE PØSITIØN CARR
L	37500				LCR			01	B			RKC CLA PØ1
M	37550				HLT			07	B		INDEX ACC NØ	
N	37600				HLT			07	B	01	ACCØUNT BAL	
P	37650				HLT			07	B	02	PURCH YTD	
R	37700				TTM			07	B	04	TYPE NAME K4	
S	37750				TTM			07	B	06	ADD K3 CITY K2	
T	37800				CLR			00	P			CLEAR
V	37850				LCR			01	C			RDC PØ2
W	37900				PAC			00	P			ØPEN CARRIAGE
X	37950				LCR			02	A			ALN LN4 AMB PØ3

PAGE REF. 37

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT. (3 4 5 6)	ALPHA (7 8 9)	PRINTER (10 11 12)	LEDGER (13 14 15)	CARD (16 17 18)

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 | 2 | 3 | 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP (30)	WI * (31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	38000				SFL		09						SET BAR 1 LANE 3
B	38050				PAC			00	P			INSERT CARD	
C	38100				RFL		09						RESET FLAGS
D	38150				TCP			15	A	12			MØVE DESC MANUAL ENT
E	38200				TCK			15	A	13			MØVE DESC HIGH ØRDER
F	38250				SRT	D	12	00	B				PØSITION HIGH ØRDER
G	38300		38	G	LCR			01	F				P05
H	38350				TFM			00	P				TYPE MANUAL ENT
J	38400				LCR			01	H				RDC APT P07
K	38450				PAC			07	B	01			PRINT BALANCE
L	38500				LCR			01	J				RDC WRL APT P08
M	38550				TCP			07	B	02			MØVE PURCH YTD TØ WØRK
N	38600				TPC			00	W				LØAD PURCH YTD IN W
P	38650				PAC			00	W				PRINT PURCH YTD
R	38700				SFL		80						TEST FØR WRITE ERRØR
S	38750				NØP								NØ ØPERATIØN
T	38800				BRU			39	S				BR TØ WRITE CØRRECTIØN
V	38850				LCR			01	L				P09
W	38900				TCP			10	K				MØVE KEY CØDE CNST
X	38950				TPC			00	B				MØVE FØR HIGH ØRDER

PAGE REF. 38

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

PROGRAMMER

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER

1	2	3	4
44 45 46 47 48 49	50 51 52 53 54	55	

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP W/I * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	39000				SLT	S	01	00	P			PØSITIØN FØR KEY 1
B	39050				TFM			00	P			PØSITIØN CARRIAGE KEY 1
C	39100				LCR			01	G			RDC LN4 P06
D	39150				TFM			00	P			PØSITIØN CARRIAGE KEY 1
E	39200				BRU			25	A			BR TØ INSERT LEDGER
F	39250		39F		LCR			02	B		FILLED SHEET	BLC AMB LN4 P03
G	39300				RFL		02					RESET FLAG TWØ
H	39350				SFL		09					SET BAR 1 LANE 3
J	39400				CLR			00	P			CLEAR PRINT AREA
K	39450				PAC			00	P		INSERT CARD	
L	39500				RFL		09					RESET FLAGS
M	39550				TCP			15	A	14		MØVE DESC BAL FØRWARD
N	39600				TCK			15	A	15		MØVE HIGH ØRDER
P	39650				TKC			00	B			PØSITIØN HIGH ØRDER
R	39700				BRU			38	6			BRANCH TØ PRINT ØUT
S	39750		39S		LCR			02	C		WRITE CØRR	WRC AMB P09
T	39800				SFL		09					SET BAR 1 LANE 3
V	39850				CLR			00	P			CLEAR PRINT AREA
W	39900				PAC			00	P		INSERT CARD	
X	39950				LCR			02	E			WRL P08

PAGE REF. 39

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns

SERIES E BASIC ASSEMBLER CODING FORM

SYSTEM SPECIFICATIONS

IDENT.	ALPHA	PRINTER	LEDGER	CARD
3 4 5 6	7 8 9	10 11 12	13 14 15	16 17 18

PAGE HEADER, ALPHA, CONTROL REGISTER: 1 2 3 4 (44 45 46 47 48 49 50 51 52 53 54 55)

CUSTOMER ABC COMPANY

BRANCH HOME OFFICE TRAINING

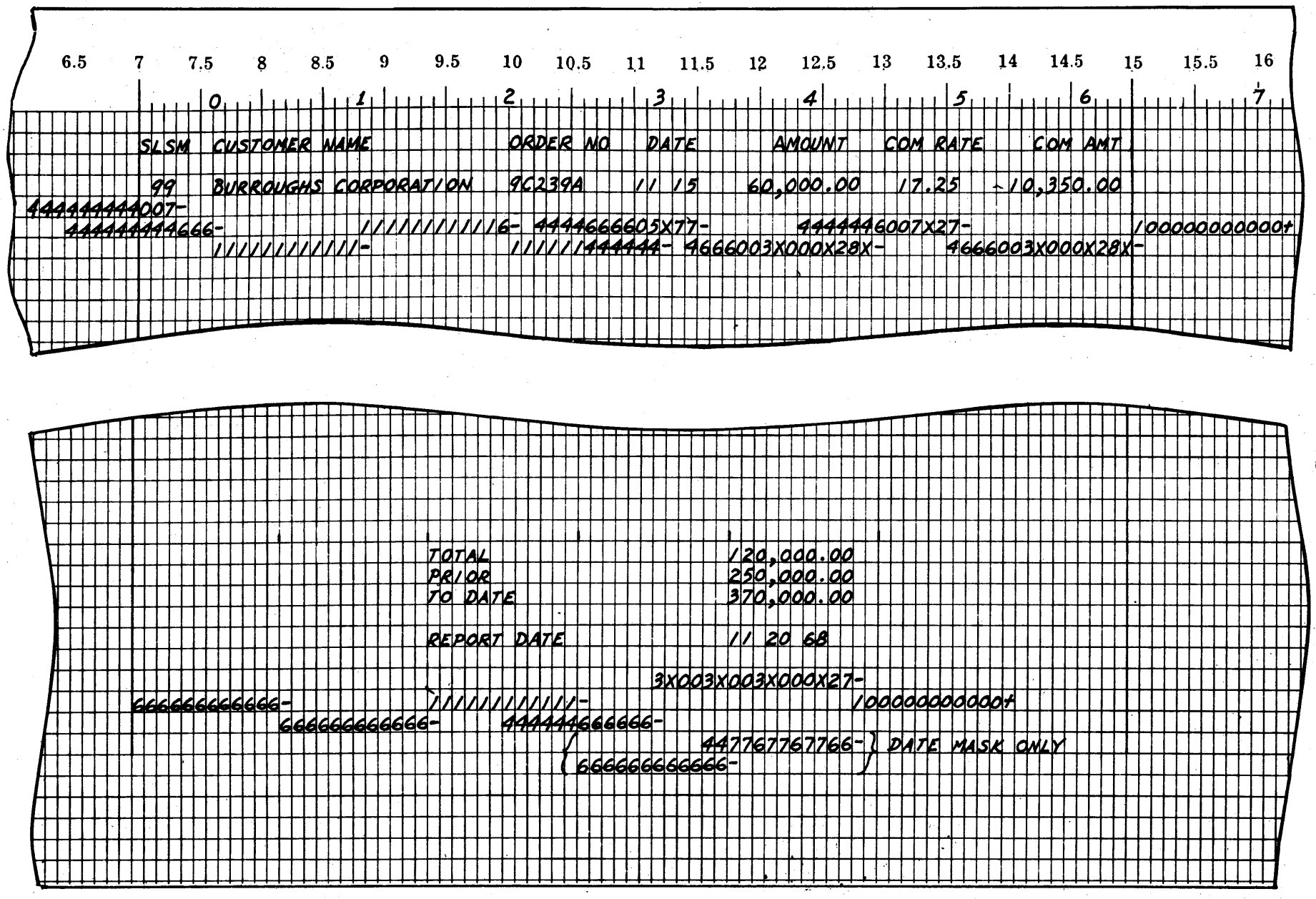
PROGRAMMER

LINE	SEQUENCE NO. (7 8 9 10 11)	PRG PT (13)	LABEL PG (14 15)	LABEL R (16)	OP CODE (18 19 20)	MOD (21)	LENGTH (23 24)	ADDRESS PG (26 27)	ADDRESS R (28)	PP WI * (30 31)	CONSTANTS (32 33 34 35 36 37 38 39 40 41 42 43)	REMARKS "PROGRAM" (57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79)
A	40000				CLR			00	W			CLEAR PRINT AREA
B	40050				RFL		09					RESET FLAGS
C	40100				BRU			28	L			BR TO WRITE LEDGER
D	40150		40	D	LCR			02	D		VER ERROR	WRL AMB PO4
E	40200				SFL		09					SET BAR 1 LANE 3
F	40250				CLR			00	P			CLEAR PRINT AREA
G	40300				PAC			00	P			WRITE LEDGER
H	40350				SFL		80					TEST FOR WRITE ERROR
J	40400				NOP							NO OPERATION
K	40450				BRU			39	S		MOVE CARR TO	BR TO WRITE CORRECTION
L	40500				BRU			25	A			BR FOR CORRECT LEDGER
M	40550				END							
N												
P												
R												
S												
T												
V												
W												
X												

PAGE REF. 40

PUNCH COLUMNS 44-55 FOR ALPHA
PUNCH COLUMNS 32-43 FOR NUMERIC
SKIP UNUSED FIELD

* NOTE: X overpunch for minus entries in these columns


6.5 7 7.5 8 8.5 9 9.5 10 10.5 11 11.5 12 12.5 13 13.5 14 14.5 15 15.5 16
0 1 2 3 4 5 6 7
SLSM CUSTOMER NAME ORDER NO DATE AMOUNT COM RATE COM AMT
99 BURROUGHS CORPORATION 9C239A 11 15 60,000.00 17.25 10,350.00
444444444007-
4444444444666-
1111111111116- 444466666605X77- 444444607X27- 10000000000f
11111111111111- 111111444444- 4666003X000X28X- 4666003X000X28X-
TOTAL 120,000.00
PRIOR 250,000.00
TO DATE 370,000.00
REPORT DATE 11 20 68
3X003X003X000X27-
66666666666666-
1111111111111- 10000000000f
666666666666- 444444666666-
{ 447767767766- } DATE MASK ONLY
{ 666666666666- }

```
ERRORS   LOC S --AUTOLOAD--- SEQ.# PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS----------
         001   000000104001 01050   01B
         002   000000010002 01100   01C
         003   001200000003 01150   01D
         004   000000000004 01200   01E
         005   000000000005 01250   01F
         006   000000010046 01300   01G
         007   000000010207 01350   01H
         008   010000010208 01400   01J
         012   010001000009 01450   01K
         013   000000000009 01500   01L
         014   000001004041 01550   01M
         015   000000001003 01600   01N
         016   000000000003 01650   01P
         017   000001104004 01700   01R
         018   000000003003 01750   01S
         019   000001000003 01800   01T
         070   000001010003 01850   01V
         071   000000010005 01900   01W
         072   000001010207 01950   01X
         073   000201000043 02000   02A
         074   000401000043 02050   02B
         075   000601000009 02100   02C
         076   010001000004 02150   02D
         077   010000000008 02200   02E
         100   242400042164 05050   05B
         110   464188644611
         101   550514500000 05100   05C
```

SLSM REPORT
ERRORS

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL	REG	CST	-----REMARKS----------
111		278121288888											
102		000000654550	05150		05D								
112		888888172278											
103		560004115000	05200		05E								
113		118382262888											
104		000146451000	05250		05F								
114		888211416888											
105		064051150000	05300		05G								
115		611872628888											
106		064014645100	05350		05H								
116		611821141688											
060		000000000000	06100		06C								
061		000000000000											
062		000000000000											
063		000000000000											
064		000000000000											
065		000000000000											
066		000000000000											
067		000000000000											
068		000000000000											
069		000000000000											
080		000000000000	08050		08B								
081		000000000000											
082		000000000000											
083		000000000000											
084		000000000000											
085		000000000000											

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL	REG	CST	-----REMARKS----------
	086		000000000000											
	087		000000000000											
	078		000000000000	10000						10A				
	079		000000000000	10050						10B				
	008		000000000000	10100						10C				
	089		000000000000	10150						10D				
	008		000000000000	10200						10E				
	099		000000000000	10250						10F				
	120		000000000000	10300						10G				
	121		000000000050	10350						10H				
	122		000000000000	10400						10J				
	123		912345000000	10450						10K				
	124		000000000001	10500						10L				
	125		161140000000	15000						15A				
	126		616268888888											
	127		756650000000	15050										
	128		177178888888											
	129		160411500000	15100										
	130		618226288888											
	131		557651041150	15150										
	132		721176822628											
	133		100055001610	15200										
	134		266872686168											
	135		756401611420	15250										
	136		171286162648											
	137		415414055100	15300										
	138		121426821688											

```
ERRORS  LOC S --AUTOLOAD--- SEQ.# PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS
          139   214066561540 15350
          140   226621742728
          141   254055565400 15360
          142   427827717988
          143   111111111111-15400   15J
          144   444444444007-15450
          145   444444444666-15500
          146   111111111116-15550
          147   444466660577-15600
          148   466600300028-15650
          149   444444600727-15700
          150   666666666666-15750
          151   444444666666-15800
          152   300300300027-15850
          153   447767767766-15900
          154   100000000000 15950
          155   111111444444-16000
          156   000000000000 16050   16B
          157   000000000000
          158   000000000000
          159   000000000000
          160   000000000000
          161   000000000000
          162   000000000000
          163   000000000000
          164   000000000000
          165   000000000000
```

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL	REG	CST	-----REMARKS-----------
	166		000000000000											
	167		000000000000											
	168		000000000000											
	169		000000000000											
	170		000000000000											
	171		000000000000											
	172		000000000000											
	173		000000000000											
	174		000000000000											
	175		000000000000											
	176		000000000000											
	177		000000000000											
	178		000000000000											
	179		000000000000											
	180		000000000000											
	181		000000000000											
	182		000000000000											
	183		000000000000											
	184		000000000000											
	185		000000000000											
	186		000000000000											
	187		000000000000											
	188		000000000000											
	189		000000000000											
	190		000000000000											
	191		000000000000											
	192		000000000000											

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL	REG	CST	-----REMARKS----------
	193		000000000000											
	194		000000000000											
	195		000000000000											
	196		000000000000											
	197		000000000000											
	198		000000000000											
	199		000000000000											
	200		000000000000											
	201		000000000000											
	202		000000000000											
	203		000000000000											
	204		000000000000											
	205		000000000000											

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
			01000		HDR					SLSM REPORT	
			01050	01B	CTL					RKCCLAP01	LOAD DATE CULS 1097643
			01100	01C	CTL					RDCP02	READ CUST ACCT NO
			01150	01D	CTL					RDLALNP03	READ LEDGER
			01200	01E	CTL					P04	KEY FOUR TFM
			01250	01F	CTL					P05	TYPE REFR
			01300	01G	CTL					RDCLN4P06	PRINT AMOUNT
			01350	01H	CTL					RDCAPTP07	PRINT BALB
			01400	01J	CTL					WRLRDCAPTP08	PRINT PURCH YTD
			01450	01K	CTL					WRLAMBP09	WRITE LEDFER
			01500	01L	CTL					P09	TFM TO POSITION CARR
			01550	01M	CTL					RKCAMBLN4P01	
			01600	01N	CTL					VC1P03	SPACE PRINTER FORM
			01650	01P	CTL					P03	POSITIOM CARRIAGE
			01700	01R	CTL					RKCCLAAMBP04	READ PRIOR SALES TO MEM
			01750	01S	CTL					VC3P03	SPACE FORM POSITION CAR
			01800	01T	CTL					AMBP03	CLOSE CARRIAGE
			01850	01V	CTL					RDCAMBP03	PRINT ACC REC TOT
			01900	01W	CTL					RDCP05	PRINT PRODUCT NO
			01950	01X	CTL					RDCAMBAPTP07	
			02000	02A	CTL					ALNLN4AMBP03	INSERT LEDGER
			02050	02B	CTL					BLCAMBLN4P03	INSERT LEDGER
			02100	02C	CTL					WRCAMBP09	INSERT LEDGER
			02150	02D	CTL					WRLAMBP04	WRITE LEDGER
			02200	02E	CTL					WRLP08	WRITE CORRECTION
			05000		FIL					11	ALPHA FILE
			05050	05B	ALP					SLSM CUSTOM	---REPT HEADINGS

```
SLSM REPORT                                                                    PAGE NO. 008
ERRORS   LOC S --AUTOLOAD--- SEQ.# PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS
                             05100  05C ALP                        ER NAME      ---RPT HEADINGS
                             05150  05D ALP                              ORDER  ---RPT HEADINGS
                             05200  05E ALP                        NO    DATE   ---RPT HEADINGS
                             05250  05F ALP                           AMOUNT   ---RPT HEADINGS
                             05300  05G ALP                        COM RATE     ---RPT HEADINGS
                             05350  05H ALP                        COM AMOUNT   ---RPT HEADINGS
                             06000      FIL                    -00              PRINTER FILE
                             06050  06B ASN  20                                  ALPHA NUMERIC OUTPUT
                             06100  06C MSK  10                     0            PRINT MASK
                             07000      FIL                    10               STRIPE LEDGER FILE
                             07050  07B ASN  20                                  LOC 0 ACC NU
                             07100                                               LOC 1 ACC BAL
                             07150                                               LOC 2 PURCH YTD
                             07200                                               LOC 3 BLANK
                             07250                                               LOC 4 14 NAME
                             07300                                               LOC 5  15 NAME K4
                             07350                                               LOC 6 16 ADDRESS K2 CIT
                             07450                                               LOC 8 18 ADDRESS K2 CIT
                             07500                                               LOC 9 19 ADDRESS K2 CIT
                             08000      FIL                     1               PUNCH CARD ALPHA NUM
                             08050  08B MSK  08                 0               CARD CODE COL 1
                             08100                                               SALESMAN NUMBER 3 COL
                             08150                                               CUST NUMBER 6 COL
                             08200                                               ORDER NUMBER 6 COL ALP
                             08250                                               ORDER DATE 4 COL
                             08300                                               PRODUCT CODE 2 COL
                             08350                                               AMOUNT 8 COL
```

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS-----------
				08400								COMM RATE 5 COL
				10000	10A	CST				0		REPORT DATE
				10050	10B	CST				0		COUNT
				10100	10C	CST				0		TOT SALE AMOUNT BY SLSM
				10150	10D	CST				0		SALE AMOUNT GRAND TOT
				10200	10E	CST				0		COMM WORK LOCATION
				10250	10F	CST				0		SLSM NO TEST LOCATION
				10300	10G	CST				0		PROD NUMBER PRINT
				10350	10H	CST				50		NUMBER OF PRODUCTS
				10400	10J	CST				0		END ROUTINE TEST MA
				10450	10K	CST				912345000000		KEY CODE CONSTANTS
				10500	10L	CST				1		MONTH CODE FOR TEST
				15000	15A	APH					TOTAL	---RPT SUB HEADINGS---
				15050		APH					PRIOR	---RPT SUB HEADINGS---
				15100		APH					TO DATE	---RPT SUB HEADINGS---
				15150		APH					REPORT DATE	---RPT SUB HEADINGS---
				15200		APH					ACC REC TOT	CONSOLE HEADING
				15250		APH					PROD TOTALS	CONSOLE HEADING
				15300		APH					MANUAL ENT	RECONSTRUCT
				15350		APH					BAL FORWARD	FILLED SHEET
				15360		APH					SEQ ERROR=	
				15400	15J	CST	01			111111111111-		ALPHA MASK
				15450		CST	01			444444444007-		NUMERIC
				15500		CST	01			444444444666-		MASK
				15550		CST	01			111111111116-		
				15600		CST	01			444466660577-		CONSTANTS
				15650		CST	01			166600300028-		NUMERIC

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS-----------
				15700		CST	01			444444600727-		MASK
				15750		CST	01			666666666666-		
				15800		CST	01			444444666666-		CONSTANTS
				15850		CST	01			300300300027-		NUMERIC
				15900		CST	01			447767767766-		MASK
				15950		CST	01			100000000000		CONSTANTS
				16000		CST	01			111111444444-		ALPHA MASK
				16050	16B	CST	50			0		PRODUCT DISTRIBUTION
	206	1	1907	20000	20A	RFL	07					RESET FLAGS
	206	2	3801	20050		LCR		01B				CLA RKC P01
	206	3	0078	20100		HLT		10A			INDEX DATE	
	207	1	1801	20150		SFL	01					SET FLAG 1
	207	2	3814	20200		LCR		01M				RKC AMB LN4 P01
	207	3	0079	20250		HLT		10B			INDEX COUNT	
	208	1	1901	20300		RFL	01					RESET FLAG 1
	208	2	3323	20350		TCP		10K				MOVE KEY CODE CST
	208	3	3010	20400		TPC		00B				MOVE FOR HIGH ORDER
	209	1	7641	20450		SLTS	01	00P				POSITION FOR KEY ONE
	209	2	7446	20500		SRTS	06	00P				POSITION LOW ORDER
	209	3	7046	20550		SRTS	06	00B				POSITION HIGH ORDER
	210	1	3083	20600		TPC		08B	03			LOAD CARD FILE WITH CD
	210	2	3210	20650		TCP		00B				MOVE HIGH ORDER
	210	3	3093	20700		TPC		08B	13			LOAD HIGH ORDER
	211	1	3337	20750		TCP		15A	12			MOVE DESC MANUAL ENT
	211	2	7442	20800		SRTS	02	00P				DROP BLANKS
	211	3	7639	20850		SLTD	12	00P				POSITION IN K
	212	1	3323	20900		TCP		10K				MOVE KEY CODE CONSTANTS

ERRORS

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP,M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
212	2	3010	20950		TPC		00B				MOVE FOR HUGH ORDER
212	3	7644	21000		SLTS	04	00P				POSITION FOR KEY 4
213	1	7602	21050		SLTD	02	00P				LOW ORDER TO K
213	2	3537	21100		TKC		15A	12			LOAD UN MEM
213	3	3738	21150		TCK		15A	13			MOVE HIGH ORDER DESC
214	1	7402	21200		SRTD	02	00P				DROP BLANKS
214	2	7202	21250		SLTD	02	00B				LOAD KEY CODE
214	3	3538	21300		TKC		15A	13			LOAD IN MEMORY
215	1	3337	21350		TCP		15A	12			MOVE DESC & KEY CODE
215	2	7659	21400		SLTS	10	00P				ISOLATE KEY CODE
215	3	3739	21450		TCK		15A	14			MOVE DESC BAL FORWARD
216	1	7001	21500		SRTD	01	00B				DROP BLANK
216	2	7601	21550		SLTD	01	00P				MOVE KEY CODE
216	3	3539	21600		TKC		15A	14			LOAD IN MEMORY
217	1	3338	21650		TCP		15A	13			MOVE DESC & KEY CODE
217	2	7659	21700		SLTS	10	00P				ISOLATE KEY CODE
217	3	3740	21750		TCK		15A	15			MOVE DESC HIGH ORDER
218	1	7001	21800		SRTD	01	00B				DROP BLANK
218	2	7601	21850		SLTD	01	00P				MOVE KEY CODE
218	3	3540	21900		TKC		15A	15			LOAD IN MEMORY
219	1	3334	21950		TCP		15A	09			MOVE DESC HIGH ORDER
219	2	7441	22000		SRTS	01	00P				DROP BLANK
219	3	7641	22050		SLTS	01	00P				POSITION DESC
220	1	3134	22100		TPC		15A	09			LOAD IN MEMORY
220	2	3336	22150		TCP		15A	11			MOVE DESC HIGH ORDER
220	3	7441	22200		SRTS	01	00P				DROP BLANK
221	1	7641	22250		SLTS	01	00P				REPOSITION DESC

ERRORS REPORT

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
221	2	3136	22300		TPC		15A	11			LOAD IN MEMORY
221	3	4880	22350		CRD		08B				READ PUNCH CARD
222	1	3802	22400		LCR		01C				RDC P02
222	2	0882	22450		PAC		08B	02			PRINT CUST ACC NO
222	3	1800									
223	1	3300	22500	22L	TCP		05B				MOVE PAGE HEADING
223	2	3710	22550		TCK		05B	10			MOVE HIGH ORDER
223	3	3040	22600		TPC		06B				LOAD PRINT TANK
224	1	3450	22650		TKC		06B	10			LOAD PRINT HIGH ORDER
224	2	3301	22700		TCP		05C				MOVE PAGE HEADING
224	3	3711	22750		TCK		05C	10			MOVE HIGH ORDER
225	1	3041	22800		TPC		06B	01			LOAD PRINT TANK
225	2	3451	22850		TKC		06B	11			LOAD PRINT HIGH ORDER
225	3	3302	22900		TCP		05D				MOVE PAGE HEADING
226	1	3712	22950		TCK		05D	10			MOVE HIGH ORDER
226	2	3042	23000		TPC		06B	02			LOAD PRINT TANK
226	3	3452	23050		TKC		06B	12			LOAD PRINT HIGH ORDER
227	1	3303	23100		TCP		05E				MOVE PAGE HEADING
227	2	3713	23150		TCK		05E	10			MOVE HIGH ORDER
227	3	3043	23200		TPC		06B	03			LOAD PRINT TANK
228	1	3453	23250		TKC		06B	13			LOAD PRINT HIGH ORDER
228	2	3304	23300		TCP		05F				MOVE PAGE HEADING
228	3	3714	23350		TCK		05F	10			MOVE PAGE HIGH ORDER
229	1	3044	23400		TPC		06B	04			LOAD PRINT TANK
229	2	3454	23450		TKC		06B	14			LOAD PRINT HIGH ORDER
229	3	3305	23500		TCP		05G				MOVE PAGE HEADING
230	1	3715	23550		TCK		05G	10			MOVE PAGE HIGH ORDER

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
	230	2	3045	23600		TPC		06B	05			LOAD PRINT TANK
	230	3	3455	23650		TKC		06B	15			LOAD PRINT HIGH ORDER
	231	1	3306	23700		TCP		05H				MOVE PAGE HEADING
	231	2	3716	23750		TCK		05H	10			MOVE PAGE HIGH ORDER
	231	3	3046	23800		TPC		06B	06			LOAD PRINT TANK
	232	1	3456	23850		TKC		06B	16			LOAD PRINT HIGH ORDER
	232	2	3343	23900		TCP		15J				MOVE MASK WORD
	232	3	3060	23950		TPC		06C				LOAD MASK WORD ZERO
	233	1	3061	24000		TPC		06C	01			LOAD MASK WORD ONE
	233	2	3062	24050		TPC		06C	02			LOAD MASK WORD TWO
	233	3	3063	24100		TPC		06C	03			LOAD MASK WORD THREE
	234	1	3064	24150		TPC		06C	04			LOAD MASK WORD FOUR
	234	2	3065	24200		TPC		06C	05			LOAD MASK WORD FIVE
	234	3	3066	24250		TPC		06C	06			LOAD MASK WORD SIX
	235	1	3354	24300		TCP		15J	11			MOVE MASK WORD E OF L
	235	2	3067	24350		TPC		06C	07			LOAD E OF L MASK
	235	3	2840	24400		PRT		06B				PRINT HEADINGS
	236	1	1800	24450		NOP						NO OPERATION
	236	2	1800									
	236	3	1800									
	237	1	3286	24500	24L	TCP		08B	06			MOVE AMOUNT TO WORK
	237	2	5487	24550		MCP		08B	07			CALC COMM
	237	3	7544	24600		SRRS	04	00P				POSITION RESULT
	238	1	3008	24650		TPC		10E				STORE COMM AMOUNT
	238	2	3286	24700		TCP		08B	06			MOVE AMOUNT TO WORK
	238	3	1800									
	239	1	3685	24750	00P	TCK		08B	05			MOVE DIST CODE TO WORK

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
	239	2	4000	24800		TIX						TURN ON INDEXING
	239	3	1156	24850		APC		16B				DISTRIBUTE AMOUNT
	240	1	1008	24900		APC		10C				ACCUMACC REC AMOUNT
	240	2	1089	24950		APC		10D				ACCUM GRAND TOTAL
	240	3	1800									
	241	1	3803	25000	25A	LCR		01D				RDL ALN P03
	241	2	1902	25050		RFL	02					RESET BAR 3
	241	3	5800	25100		CLR		00P				CLEAR PRINT AREA
	242	1	0800	25150		PAC		00P			INSERT CARD	PRINT DATE
	242	2	1880	25200		SFL	80					TEST ERROR INDICATORS
	242	3	8324	25250		BRU		37F				BR TO READ ERROR ROUT
	243	1	8338	25300		BRU		39F				BR TO FILLED SHEET ROUT
	243	2	3282	25350		TCP		08B	02			MOVE CUST NO TO WORK
	243	3	2220	25400		SCP		07B				SUBT CUST NO
	244	1	9745	25450		BRN		40D				BR IF NOT EQUAL
	244	2	3281	25500		TCP		08B	01			MOVE SALS NO TO WORK
	244	3	3040	25550		TPC		06B				LOAD PRINT TANK
	245	1	3224	25600		TCP		07B	04			MOVE NAME TO WORK
	245	2	3634	25650		TCK		07B	14			MOVE HIGH ORDER
	245	3	3042	25700		TPC		06B	02			LOAD PRINT TANK
	246	1	3452	25750		TKC		06B	12			LOAD HIGH ORDER
	246	2	3225	25800		TCP		07B	05			MOVE NAME TO WORK
	246	3	3635	25850		TCK		07B	15			MOVE HIGH ORDER
	247	1	3043	25900		TPC		06B	03			LOAD PRINT TANK
	247	2	3453	25950		TKC		06B	13			LOAD HIGH ORDER
	247	3	3283	26000		TCP		08B	03			MOVE ORDER NO LOW ORDER
	248	1	3693	26050		TCK		08B	13			MOVE ORDER NO HGH ORDER

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
248	2	3044	26100		TPC		06B	04			LOAD PRINT TANK
248	3	3454	26150		TKC		06B	14			LOAD HIGH ORDER
249	1	3284	26200		TCP		08B	04			MOVE DATE TO WORK
249	2	3045	26250		TPC		06B	05			LOAD PRINT TANK
249	3	3286	26300		TCP		08B	06			MOVE AMOUNT TO WORK
250	1	3046	26350		TPC		06B	06			L/AD PRINT TANK
250	2	3287	26400		TCP		08B	07			MOVE COMM RATE TO WORK
250	3	3047	26450		TPC		06B	07			LOAD PRINT TANK
251	1	3208	26500		TCP		10E				MOVE COM AMOUNT TO WORK
251	2	3048	26550		TPC		06B	08			LOAD PRINT TANK
251	3	3344	26600		TCP		15J	01			MOVE MSK CST
252	1	3060	26650		TPC		06C				LOAD MASK WORD ZERO
252	2	3345	26700		TCP		15J	02			MOVE MASK CNST
252	3	3061	26750		TPC		06C	01			LOAD MASK WORD ONE
253	1	3343	26800		TCP		15J				MOVE MASK CNST
253	2	3062	26850		TPC		06C	02			LOAD MASK WORD TWO
253	3	3346	26900		TCP		15J	03			MOVE MASK CNST
254	1	3063	26950		TPC		06C	03			LOAD MASK WORD THREE
254	2	3355	27000		TCP		15J	12			MOVE MASK CNST
254	3	3064	27050		TPC		06C	04			LOAD MASK WORD FOUR
255	1	3347	27100		TCP		15J	04			MOVE MASK CNST
255	2	3065	27150		TPC		06C	05			LOAD MASK WORD FIVE
255	3	3348	27200		TCP		15J	05			MOVE MASK CNST
256	1	3066	27250		TPC		06C	06			LOAD MASK WORD SIX
256	2	3068	27300		TPC		06C	08			LOAD MASK WORD EIGHT
256	3	3349	27350		TCP		15J	06			MOVE MASK CNST
257	1	3067	27400		TPC		06C	07			LOAD MASK WORD SEVEN

ERRORS

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
257	2	3354	27450		TCP		15J	11			MOVE MASK CNST EOL
257	3	3069	27500		TPC		06C	09			LOAD MASK WORD NINE
258	1	2840	27550		PRT		06B				PRINT ON PRINTER
258	2	1800	27600		NOP						NO OPERATION NO BOF TST
258	3	3286	27650		TCP		08B	06			MOVE AMOUNT TO WORK
259	1	1021	27700		APC		07B	01			UPDATE ACCOUNT BAL
259	2	1022	27750		APC		07B	02			UPDATE PURCH YTD
259	3	3804	27800		LCR		01E				P04
260	1	3323	27850		TCP		10K				MOVE KEY CODE CST
260	2	3010	27900		TPC		00B				MOVE FOR HIGH ORDER
260	3	7644	27950		SLTS	04	00P				POSITION FOR KEY 4
261	1	0400	28000		TFM		00P				POSITION CARRIAGE
261	2	3805	28050		LCR		01F				P05
261	3	0483	28100		TFM		08B	03			TYPE ORDER NO
262	1	3806	28150		LCR		01G				RDC LN4 P06
262	2	0886	28200		PAC		08B	06			PRINT DEBIT
262	3	3807	28250		LCR		01H				RDC APT P07
263	1	0821	28300		PAC		07B	01			PRINT BALANCE
263	2	3808	28350		LCR		01J				RDC WRL APT P08
263	3	3222	28400		TCP		07B	02			MOVE PURCH YTD TO WORK
264	1	3009	28450		TPC		00W				LOAD PURCH YTD IN W
264	2	1800									
264	3	1800									
265	1	0809	28500	28L	PAC		00W				PRINT PURCH YTD
265	2	1880	28550		SFL	80					TEST FOR WRITE ERROR
265	3	1800	28600		NOP						NO OPERATION
266	1	8342	28650		BRU		39S				BR TO WRITE CORRECTION

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
	266	2	3813	28700		LCR		01L				P09
	266	3	3323	28750		TCP		10K				MOVE KEY CODE CST
	267	1	3010	28800		TPC		00B				MOVE FOR HIGH ORDER
	267	2	7642	28850		SLTS	02	00P				POSITION FOR KEY 2
	267	3	0400	28900		TFM		00P				POSITION CARR WITH K2
	268	1	3281	28950		TCP		08B	01			MOVE SLSM NO
	268	2	3099	29000		TPC		10F				STORE SLSM NO
	268	3	4860	29050		CRD		08B				READ PUNCH CARD
	269	1	3802	29100		LCR		01C				ROC P02
	269	2	0882	29150		PAC		08B	02			PRINT CUSTOMER NO
	269	3	3299	29200		TCP		10F				MOVE SLSM NO LAST CARD
	270	1	2281	29250		SCP		08B	01			SUBT SLSM NO THIS CARD
	270	2	9671	29300		BRN			1			BR IF NEW SLSM
	270	3	8237	29350		BRU		24L				BR TO CALC COMM
	271	1	3280	29400	1	TCP		08B				MOVE CARD CODE
	271	2	7769	29450		PFMS	11	00P				PFM CARD TYPE
	271	3	3281	29500		TCP		08B	01			MOVE SLSM NUMBER
	272	1	7749	29550		PFMS	09	00P			PFM CRD CODE	AND SLSM NO
	272	2	3354	29600		TCP		15J	11			MOVE MASK CST
	272	3	3060	29650		TPC		06C				LOAD MASK WORD
	273	1	3815	29700		LCR		01N				VC1 P03
	273	2	2840	29750		PRT		06B				SPACE FORM NO PRINT
	273	3	1800	29800		NOP						NO OPERATION
	274	1	3325	29850		TCP		15A				MOVE DESC TOTAL
	274	2	3042	29900		TPC		06B	02			LOAD PRINT TANK
	274	3	3326	29950		TCP		15A	01			MOVE DESC TOTAL
	275	1	3052	30000		TPC		06B	12			LOAD PRINT TANK

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP,M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
	275	2	3208	30050		TCP		10C				MOVE SLSM TOTAL
	275	3	3044	30100		TPC		06B	04			LOAD PRINT TANK
	276	1	7743	30150		PFMS	03	00P			PFM TOTAL TH	IS PERIOD
	276	2	3350	30200		TCP		15J	07			MOVE MASK CST
	276	3	3060	30250		TPC		06C				LOAD MASK WORD ZERO
	277	1	3061	30300		TPC		06C	01			LOAD MASK WORD ONR
	277	2	3343	30350		TCP		15J				MOVE MASK CNST
	277	3	3062	30400		TPC		06C	02			LOAD MASK WORD TWO
	278	1	3351	30450		TCP		15J	08			MOVE MASK CONSTANT
	278	2	3063	30500		TPC		06C	03			LOAD MASK WORD THREE
	278	3	3352	30550		TCP		15J	09			MOVE MASK CONSTANT
	279	1	3064	30600		TPC		06C	04			LOAD MASK WORD FOUR
	279	2	3354	30650		TCP		15J	11			MOVE E OF L MSK CONST
	279	3	3065	30700		TPC		06C	05			LOAD E OF L MSK WORD
	280	1	3816	30750		LCR		01P				P03
	280	2	2840	30800		PRT		06B				PRINT ON PRINTER
	280	3	1800	30850		NOP						NO OPERATION
	281	1	1809	30900		SFL	09					SET BAR 1 LANE 3
	281	2	5800	30950		CLR		00P				CLEAR PRINT AREA
	281	3	0800	31000		PAC		00P				POSITION CARRIAGE
	282	1	1909	31050		RFL	09					RESET FLAGS
	282	2	3278	31100		TCP		10A				MOVE REPORT DATE
	282	3	7444	31150		SRTS	04	00P				POSITION MO BY ITSELF
	283	1	2324	31200		SCP		10L				TEST FOR MO 1
	283	2	9684	31250		BRN		31L				BRANCH TO READ PRIOR
				31300								SALES TO-DATE FROM CARD
				31350								READER NO-2

```
LOC S --AUTOLOAD--- SEQ.# PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS----------
283 3 8290          31400      BRU          2                          BRANCH TO PRT NO PRIOR
                    31450                                              SALES TO DATE
284 1 3699          31500  31L TCK      10F                            STORE OLD SLSM NO IN K
284 2 3281          31550      TCP      08B 01                         MOVE OLD SLSM NO
284 3 3099          31600      TPC      10F                            STORE AS CST
285 1 4882          31650      CRD      08B 02      FEED ONE CRD       FROM CRD READER NO 2
285 2 3281          31700      TCP      08B 01                         MOVE SLSM NO
285 3 2211          31750      SCP      00K                            TEST TO SEE IF SLSM
                    31800                                              DETAIL AND PRIOR CARDS
                    31850                                              ARE MATCHED IE THE SAME
286 1 9687          31900      BRN          1                          BR TO SEQUENCE ERROR
286 2 8291          31950      BRU          3                          BR TO CONTINUE
286 3 1800
287 1 3342          32000 1    TCP      15A 17                         MOVE HIGH ORDER
287 2 3010          32050      TPC      00B                            STORE IN 00B
287 3 3341          32100      TCP      15A 16                         MOVE LOW ORDER
288 1 0400          32150      TFM      00P         SEQUENCE ERR       MAR LIGHTS=INDICATOR
288 2 1800          32200      NOP
288 3 3871          32250      LCR      01W                            P05
289 1 0077          32300      HLT      00P 77      HANDLE SEQ         ERROR=RELOAD READER
289 2 8284          32350      BRU      31L                            READ ANOTHER PRIOR CARD
289 3 1800
290 1 5883          32400 2    CLR      08B 03                         CLR THIS AREA IF NO
                    32450                                              PRIOR MONTHS SUMMARY CD
290 2 1800
290 3 1800
291 1 3283          32500 3    TCP      08B 03                         MOVE PRIOR SALES=TO=DAT
```

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS-----------
	291	2	3044	32550		TPC		06B	04			LOAD PRINT TANK
	291	3	1008	32600		APC		10C				ACCUMULATE TO DATE
	292	1	1909	32650		RFL	09					RESET FLAGS
	292	2	3327	32700		TCP		15A	02			MOVE DESC PRIOR
	292	3	3728	32750		TCK		15A	03			MOVE DESC PRIOR
	293	1	3042	32800		TPC		06B	02			LOAD PRINT TANK
	293	2	3452	32850		TKC		06B	12			LOAD PRINT HIGH ORDER
	293	3	2840	32900		PRT		06B				PRINT PRIOR TOTAL
	294	1	1800	32950		NOP						NO OPERATION
	294	2	3329	33000		TCP		15A	04			MOVE DESC TO DATE
	294	3	3730	33050		TCK		15A	05			MOVE DESC TO DATE
	295	1	3042	33100		TPC		06B	02			LOAD PRINT TANK
	295	2	3452	33150		TKC		06B	12			LOAD PRINT HIGH ORDER
	295	3	3208	33200		TCP		10C				MOVE TO DATE TOTAL
	296	1	5808	33250		CLR		10C				CLEAR WORK AREA
	296	2	3044	33300		TPC		06B	04			LOAD PRINT TANK
	296	3	7743	33350		PFMS	03	00P			PFM TOTAL TO DATE	
	297	1	2840	33400		PRT		06B				PRINT TO DATE
	297	2	1800	33450		NOP						NO OPERATION
	297	3	3350	33500		TCP		15J	07			MOVE MASK CST
	298	1	3063	33550		TPC		06C	03			LOAD MASK WORD THREE
	298	2	3353	33600		TCP		15J	10			MOVE MASK CNST
	298	3	3064	33650		TPC		06C	04			LOAD MASK WORD FOUR
	299	1	3331	33700		TCP		15A	06			MOVE DESC REPORT DATE
	299	2	3732	33750		TCK		15A	07			MOVE DESC REPORT DATE
	299	3	3042	33800		TPC		06B	02			LOAD PRINT TANK
	300	1	3452	33850		TKC		06B	12			LOAD PRINT HIGH ORDER

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
300	2	3278	33900		TCP		10A				MOVE DATE TO WORK
300	3	3044	33950		TPC		06B	04			LOAD PRINT TANK
301	1	7746	34000		PFMS	06	00P			PFM REPORT	DATE
301	2	7788	34050		REL						RELEASE NEW PRIOR CARD
301	3	2840	34100		PRT		06B				PRINT REPORT DATE
302	1	1800	34150		NOP						NO OPERATION
302	2	3281	34200		TCP		08B	01			MOVE SLSM NO
302	3	9304	34250		BRM				1		IF MINUS END OF RUN
303	1	8223	34300		BRU		22L				BR TO HEAD FORM NEW MAN
303	2	1800									
303	3	1800									
304	1	3354	34350	1	TCP		15J	11			MOVE MSK CNST
304	2	3060	34400		TPC		06C				LOAD MASK WORD
304	3	3818	34450		LCR		01S				VC3 P03
305	1	2840	34500		PRT		06B				SPACE FORM NO PRINT
305	2	1800	34550		NOP						NO OPERATION
305	3	2840	34600		PRT		06B				SPACE FORM NO PRINT
306	1	1800	34650		NOP						NO OPERATION
306	2	2840	34700		PRT		06B				SPACE FORM NO PRINT
306	3	1800	34750		NOP						NO OPERATION
307	1	2840	34800		PRT		06B				SPACE FORM NO PRINT
307	2	1800	34850		NOP						NO OPERATION
307	3	2840	34900		PRT		06B				SPACE FORM NO PRINT
308	1	1800	34950		NOP						NO OPERATION
308	2	1802	35000		SFL	02					SET FLAG FOR BAR 3
308	3	3819	35050		LCR		01T				AMB P03
309	1	5800	35100		CLR		00P				CLEAR PRINT AREA

SLSM REPORT

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WT	CONSTANT	CTL REG CST	-----REMARKS----------
	309	2	0800	35150		PAC		00P				CLOSE CARRIAGE
	309	3	3323	35200		TCP		10K				MOVE KEY CODE CNST
	310	1	3010	35250		TPC		00B				MOVE FOR HIGH ORDER
	310	2	7641	35300		SLTS	01	00P				POSITION FOR KEY 1
	310	3	7469	35350		SRTS	11	00P				POSITION KEY CODE
	311	1	7069	35400		SRTS	11	00B				POSITION KEY CODE
	311	2	1333	35450		ACP		15A	08			MOVE LOW ORDER DESC
	311	3	3734	35500		TCK		15A	09			MOVE HIGH ORDER DESC
	312	1	1410	35550		AKC		00B				SET FOR TYPE
	312	2	0400	35600		TFM		00P				TYPE ACC REC TOT
	312	3	3870	35650		LCR		01V				RDC AMB P03
	313	1	0889	35700		PAC		10D				PRINT ACC REC TOTAL
	313	2	3323	35750		TCP		10K				MOVE KEY CODE CNST
	313	3	3010	35800		TPC		00B				MOVE FOR HIGH ORDER
	314	1	7644	35850		SLTS	04	00P				POSITION FOR KEY 4
	314	2	7469	35900		SRTS	11	00P				POSITION KEY CODE
	314	3	7069	35950		SRTS	11	00B				POSITION KEY CODE
	315	1	1335	36000		ACP		15A	10			MOVE LOW ORDER DESC
	315	2	3736	36050		TCK		15A	11			MOVE HIGHORDER DESC
	315	3	1410	36100		AKC		00B				SET FOR TYPE
	316	1	0400	36150		TFM		00P				TYPE PROD TOTALS
	316	2	5811	36200		CLR		00K				CLEAR K
	316	3	1800									
	317	1	3520	36250	36F	TKC		10G				LOAD COUNT IN PRT TANK
	317	2	3320	36300		TCP		10G				MOVE PROD NO TO WORK
	317	3	3122	36350		TPC		10J				MOVE PROD NO TO TEST MA
	318	1	3871	36400		LCR		01W				RDC APT P05

SLSM REPORT
ERRORS

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS	PAGE NO. 023
318	2	1902	36450		RFL	02					RESET FLAG 2	
318	3	0920	36500		PAC		10G				PRINT PROD NO 0-49	
319	1	3872	36550		LCR		01X				RDC AMB APT P07	
319	2	1809	36600		SFL	09					SET FOR BAR 1 LANE 3	
319	3	1800										
320	1	3720	36650	00@	TCK		10G				MOVE PROD NO TO K	
320	2	4000	36700		TIX						TURN ON INDEXING	
320	3	0956	36750		PAC		16B				PRINT PROD TOTAL	
321	1	1909	36800		RFL	09					RESET PROGRAM FLAGS	
321	2	3322	36850		TCP		10J				PROD NO TO P	
321	3	1279	36900		ACP		10B				ADD ONE	
322	1	3011	36950		TPC		00K				MOVE NXT PROD NO TO K	
322	2	2321	37000		SCP		10H				SUBT LIMIT OF 50	
322	3	9717	37050		BRN		36F				BR FOR NEXT PRODUCT	
323	1	5800	37100		CLR		00P				CLEAR	
323	2	3800	37150		LCR		00P				CLEAR CONTROL REGISTER	
323	3	0000	37200		HLT		00P				END RUN	
324	1	3816	37250	37F	LCR		01P			READ ERROR	P03	
324	2	3323	37300		TCP		10K				MOVE KEY CODE CNST	
324	3	3010	37350		TPC		00B				MOVE FOR HIGH ORDER	
325	1	7641	37400		SLTS	01	00P				POSITION FOR KEY 1	
325	2	0400	37450		TFM		00P				KEY ONE POSITION CARR	
325	3	3801	37500		LCR		01B				RKC CLA P01	
326	1	0020	37550		HLT		07B			INDEX ACC NO		
326	2	0021	37600		HLT		07B	01		ACCOUNT BAL		
326	3	0022	37650		HLT		07B	02		PURCH YTD		
327	1	0224	37700		TTM		07B	04		TYPE NAME K4		

SLSM REPORT ERRORS

LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS
327	2	0226	37750		TTM		07B	06		ADD K3 CIT 4	
327	3	5800	37800		CLR		00P				CLEAR
328	1	3802	37850		LCR		01C				RDC P02
328	2	0800	37900		PAC		00P				OPEN CARRIAGE
328	3	3873	37950		LCR		02A				ALN LN4 AMB P03
329	1	1809	38000		SFL	09					SET BAR 1 LANE 3
329	2	0800	38050		PAC		00P			INSERT CARD	
329	3	1909	38100		RFL	09					RESET FLAGS
330	1	3337	38150		TCP		15A	12			MOVE DESC MANUAL ENT
330	2	3738	38200		TCK		15A	13			MOVE DESC HIGH ORDER
330	3	7039	38250		SRTD	12	00B				POSITION HIGH ORDER
331	1	3805	38300	38G	LCR		01F				P05
331	2	0400	38350		TFM		00P				TYPE MANUAL ENT
331	3	3807	38400		LCR		01H				RDC APT P07
332	1	0821	38450		PAC		07B	01			PRINT BALANCE
332	2	3808	38500		LCR		01J				RDC WRL APT P08
332	3	3222	38550		TCP		07B	02			MOVE PURCH YTD TO WORK
333	1	3009	38600		TPC		00W				LOAD PURCH YTD IN W
333	2	0809	38650		PAC		00W				PRINT PURCH YTD
333	3	1880	38700		SFL	80					TEST FOR WRITE ERROR
334	1	1800	38750		NOP						NO OPERATION
334	2	8342	38800		BRU		39S				BR TO WRITE CORRECTION
334	3	3813	38850		LCR		01L				P09
335	1	3323	38900		TCP		10K				MOVE KEY CODE CNST
335	2	3010	38950		TPC		00B				MOVE FOR HIGH ORDER
335	3	7641	39000		SLTS	01	00P				POSITION FOR KEY 1
336	1	0400	39050		TFM		00P				POSITION CARRIAGE KEY 1

ERRORS	LOC	S	--AUTOLOAD---	SEQ.#	PLAB	OP.M	LN	ADR	WI	CONSTANT	CTL REG CST	-----REMARKS----------
	336	2	3806	39100		LCR		01G				RDC LN4 P06
	336	3	0400	39150		TFM		00P				POSITION CARRIAGE KEY 1
	337	1	8241	39200		BRU		25A				BR TO INSERT LEDGER
	337	2	1800									
	337	3	1800									
	338	1	3874	39250	39F	LCR		02B			FILLED SHEET	BLC AMB LN4 P03
	338	2	1902	39300		RFL	02					RESET FLAG TWO
	338	3	1809	39350		SFL	09					SET BAR 1 LANE 3
	339	1	5800	39400		CLR		00P				CLEAR PRINT AREA
	339	2	0800	39450		PAC		00P			INSERT CARD	
	339	3	1909	39500		RFL	09					RESET FLAGS
	340	1	3339	39550		TCP		15A	14			MOVE DESC BAL FORWARD
	340	2	3740	39600		TCK		15A	15			MOVE HIGH ORDER
	340	3	3410	39650		TKC		00B				POSITION HIGH ORDER
	341	1	8331	39700		BRU		38G				BRANCH TO PRINT OUT
	341	2	1800									
	341	3	1800									
	342	1	3875	39750	39S	LCR		02C			WRITE CORR	WRC AMB P09
	342	2	1809	39800		SFL	09					SET BAR 1 LANE 3
	342	3	5800	39850		CLR		00P				CLEAR PRINT AREA
	343	1	0800	39900		PAC		00P			INSERT CARD	
	343	2	3877	39950		LCR		02E				WRL P08
	343	3	5809	40000		CLR		00W				CLEAR PRINT AREA
	344	1	1909	40050		RFL	09					RESET FLAGS
	344	2	8265	40100		BRU		28L				BR TO WRITE LEDGER
	344	3	1800									
	345	1	3876	40150	40D	LCR		02D			VER ERROR	WRL AMB P04

SLSM REPORT

```
ERRORS   LOC S --AUTOLOAD---  SEQ.#  PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS----------
         345 2 1809           40200       SFL  09                                  SET BAR 1 LANE 3
         345 3 5800           40250       CLR      00P                               CLEAR PRINT AREA
         346 1 0800           40300       PAC      00P                               WRITE LEDGER
         346 2 1880           40350       SFL  80                                    TEST FOR WRITE ERROR
         346 3 1800           40400       NOP                                        NO OPERATION
         347 1 8342           40450       BRU      39S              MOVE CARR TO BR TO WRITE CORRECTION
         347 2 8241           40500       BRU      25A                               BR FOR CORRECT LEDGER
         347 3 1800
         401 1 8206           40550       END
         401 2 1800
         401 3 1800
```

LABELS USED= +65,LABELS ALLOWED ON THE E6000 BASIC ASSEMBLER= 100 INCLUDING 00P,00B,00K,AND 00W.

PROGRAM POINTS USED= +05. PROGRAM POINTS ALLOWED ON THE E6000 BASIC ASSEMBLER= 099.

DATE = 05/27/70
TIME = 09:04
VERSION = 08/01/69

FORM LAYOUT -- ALL BURROUGHS SERIES WITH 1/10" TA

RELEASE DISKS MUST BE INDICATED IN EXACT POSITION ON .1" LINES. TYPEWRITER RELEASE DISKS SHOULD BE IN LINE WITH SELECTED STOP.

TO RELEASE A SKIP OR RETURN INDEX IN LANES 1 THROUGH 9, THE RELEASE DISK MUST BE:

AT LEAST .9" BEYOND A BY-PASSED STOP ADJACENT TO THE SELECTED POSITION.

NOT MORE THAN .5" PAST THE POSITION AT WHICH THE CARRIAGE IS TO STOP.

ON REAR INSERTED PANELS, PLACED FROM .4" ON THE LEFT SIDE OF SCALE (OR .6" IF THE DISK IS REVERSED, HUB TO THE LEFT).

ON SIDE INSERTED PANELS, PLACED FROM .3" ON THE LEFT SIDE OF SCALE (OR .9" IF THE DISK IS REVERSED) AND PLACED FROM 21.9" ON THE RIGHT SIDE OF SCALE (OR 22.0" IF THE DISK IS REVERSED). A STOP PLACED AT 22.0" MUST ALSO BE REVERSED.

BRACE LOCATIONS (SEE SHADED AREA IN MAGAZINE SECTION AT BOTTOM OF FORM FOR POSSIBLE LOCATIONS):

FOR REAR INSERTED PANELS, BRACE WIDTH IS .9" SPACED NOT MORE THAN 9" APART ON EACH SHAFT WITH A MINIMUM OF THREE ON AN 18" CGE. AND FOUR ON A 22" CGE.

FOR SIDE INSERTED PANELS, BRACE WIDTH IS .9" SPACED NOT MORE THAN 7"APART ON EACH SHAFT WITH A MINIMUM OF FIVE LOCATIONS, ONE EACH AS CLOSE TO SCALE POSITIONS 2" AND 20" AS POSSIBLE.

A MINIMUM OF .2" MUST BE PROVIDED ... JACENT BY-PASSED STOP WHEN CARR... THROUGH 9, OR THE TYPEWRITER LAN...

REVERSE MOVEMENT IN LANE 3 IS A MI...

OVERLAPPING SKIP INDEXES AND RE... STOPS THAT ARE LOCATED AT LEAST ...

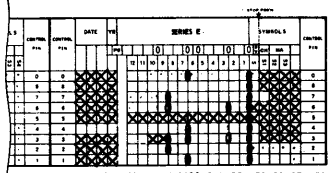
THE USE OF PINS NUMBERED 1 THROU... FROM LEFT EDGE OF PLATEN (NOT AP...

FORMS AROUND PLATEN (JOURNALS):

FORMS FRONT INSERTED: FIRST LINE I... LAST LINE I... WHEN USI... MAXIMUM PR... LIMITS, ...

USE X IN EACH HAMMERBLOCK POSITI... POSITIONS. USE Ø BETWEEN COLUMN... Ø INDICATES STANDARD SPLIT LOCATI...

LAYOUT MUST BE ACCOMPANIED BY ... FORM NN TQ-2578.



N OF CHECK LIMITS - SERIES E 2100 STRIPE LEDGER STYLES
PORTED BY CHECK LIMITS IS .25" ON THE SCALE.
ES - 4 SETS

SPECIAL FEATURES:

F.O. Number	Invoice Number	Style of Machine	Unit Number
		E 84-	1
Application			Bus. Class
SALESMAN REPORT AND LEDGER UPDATE			
Customer			
HOME OFFICE TRAINING			
Branch	Prepared Date	Approved	Date
	W.G.W. 6/24/68		

1ST STOP ALLOWED AT .5"

PRE-SET LEFT FORM GUIDE (ONE ONLY)

MAGNETIC STRIPE LOCATIONS

FIXED RIGHT FORM GUIDE

STOP SCALE

TYPEWRITER — 1ST TYPING STOP POSITION ALLOWED AT 4.1" — 1ST TYPING LIMIT — 18" CARRIAGE — 22" CARRIAGE

FORMS: AROUND PLATEN / FRONT FEED

1 ACCOUNTS RECEIVABLE LEDGER POSTING AND SALESMAN REPORT ON LINE PRINTER

(LOAD)	CUST. NO.	DATE	REFERENCE	DEBIT	CREDIT	BALANCE	SALES Y.T.D.

POSITION CHECK

1 ►
2 ◄
3 ◄►R/S
4 ►
5 ◄
26 SELECT
27 →K4
28 ←K3
29 ←K2
30 ←K1,K4

SPECIAL INSTRUCTIONS: Z SP; Z SP; 8 SP NT; 8 OP; 4; 5 OP; 8; 8; 8; 8 OP; 5

STOP SCALE

FORMS: AROUND PLATEN / FRONT FEED

2

POSITION NO.

1 ►
2 ◄
3 ◄►R/S
4 ►
5 ◄
26 SELECT
27 →K4
28 ←K3
29 ←K2
30 ←K1,K4
SPECIAL INSTRUCTIONS

STOP SCALE

FORMS: AROUND PLATEN / FRONT FEED

3

POSITION NO.

1 ►
2 ◄
3 ◄►R/S
4 ►
5 ◄
26 SELECT
27 →K4
28 ←K3
29 ←K2
30 ←K1,K4
SPECIAL

	ACCT. NO.	CUSTOMER	BALANCES FORWARD	
			BALANCE	SALES Y.T.D.
	12.42	H COMPANY 1234 MAIN USA	3,300.00	60,000.00

DATE	REFERENCE	DEBIT	CREDIT	BALANCE	SALES Y.T.D.
JUL 31'--	6L654J	400.50		3,700.50*	60,400.50*
JUL 31'--	4J442L	258.00		3,958.50*	60,658.50*

SLSM	CUSTOMER NAME	ORDER NO	DATE	AMOUNT	COM RATE	COM AMOUNT
1	B COMPANY	2U212U	7 23	2,563.50	10.00	256.35
1	B COMPANY	1L123J	7 01	500.00	10.00	50.00
1	B COMPANY	5L426P	7 12	2,654.20	10.25	272.06
1	B COMPANY	5M752U	7 14	2,642.00	10.25	270.81
1	B COMPANY	3L653L	7 14	25.00	15.00	3.75
1	B COMPANY	5K555J	7 14	2,413.00	12.25	295.59
1	B COMPANY	2K539J	7 20	268.50	12.25	32.89
1	B COMPANY	2U258M	7 22	2,132.00	10.25	218.53
1	B COMPANY	2J786J	7 23	265.00	5.00	13.25
1	B COMPANY	5K478M	7 23	2,512.00	12.25	307.72
1	B COMPANY	1K542M	7 24	2,531.00	12.00	303.72
1	B COMPANY	6L654J	7 24	2,546.30	12.00	305.56
1	B COMPANY	5L987M	7 28	2,589.60	10.25	265.43
1	D COMPANY	1H456M	7 02	25,000.00	5.15	1,287.50
1	E COMPANY	5F159L	7 03	5,682.00	10.25	582.41
1	F COMPANY	6M865F	7 05	56.32	15.65	8.76
1	G COMPANY	5K545K	7 28	2,587.52	13.00	336.38
1	H COMPANY	6L554J	7 29	400.50	10.50	42.05
1	H COMPANY	4J442L	7 12	258.00	5.25	13.55
1	H COMPANY	5J425U	7 13	2,585.00	12.50	323.13
1	J COMPANY	2K789G	7 19	258.96	12.25	31.72
1	K COMPANY	2T456F	7 21	568.00	1.22	6.93
1	K COMPANY	2K147R	7 23	2,500.00	5.00	125.00
1	L COMPANY	9K456L	7 12	830.00	10.25	85.08
1	M COMPANY	5R458F	7 29	5,324.00	10.25	545.71
1	M COMPANY	5J147F	7 01	2,583.00	10.00	258.30
1	N COMPANY	9G426T	7 24	2,635.00	13.25	349.14
1	O COMPANY	4M852D	7 25	564.00	12.25	69.09
1	P COMPANY	3P369N	7 26	258.00	15.55	40.12
1	R COMPANY	4H012Y	7 25	125.00	10.25	12.81
1	S COMPANY	8P147H	7 14	5,689.00	10.25	583.12
1	T COMPANY	5P863W	7 26	2,500.00	5.00	125.00
1	U COMPANY	6M478J	7 27	50.25	15.00	7.54

TOTAL	90,084.75
PRIOR	565,678.67
TO DATE	655,763.42
REPORT DATE	07 31 --

SLSM	CUSTOMER NAME	ORDER NO	DATE	AMOUNT	COM RATE	COM AMOUNT
2	A A COMPANY	2K555L	7 12	256.00	10.25	26.24
2	B B COMPANY	6P369K	7 27	254.00	12.25	31.12
2	C C COMPANY	5M753L	7 16	265.30	12.25	32.50
2	D D COMPANY	3L654K	7 10	258.00	10.25	26.45
2	E E COMPANY	5K541U	7 25	25.00	15.25	3.81
2	F F COMPANY	6L987M	7 15	2,530.00	10.25	259.33
2	G G COMPANY	3L357M	7 14	254.00	12.52	31.80

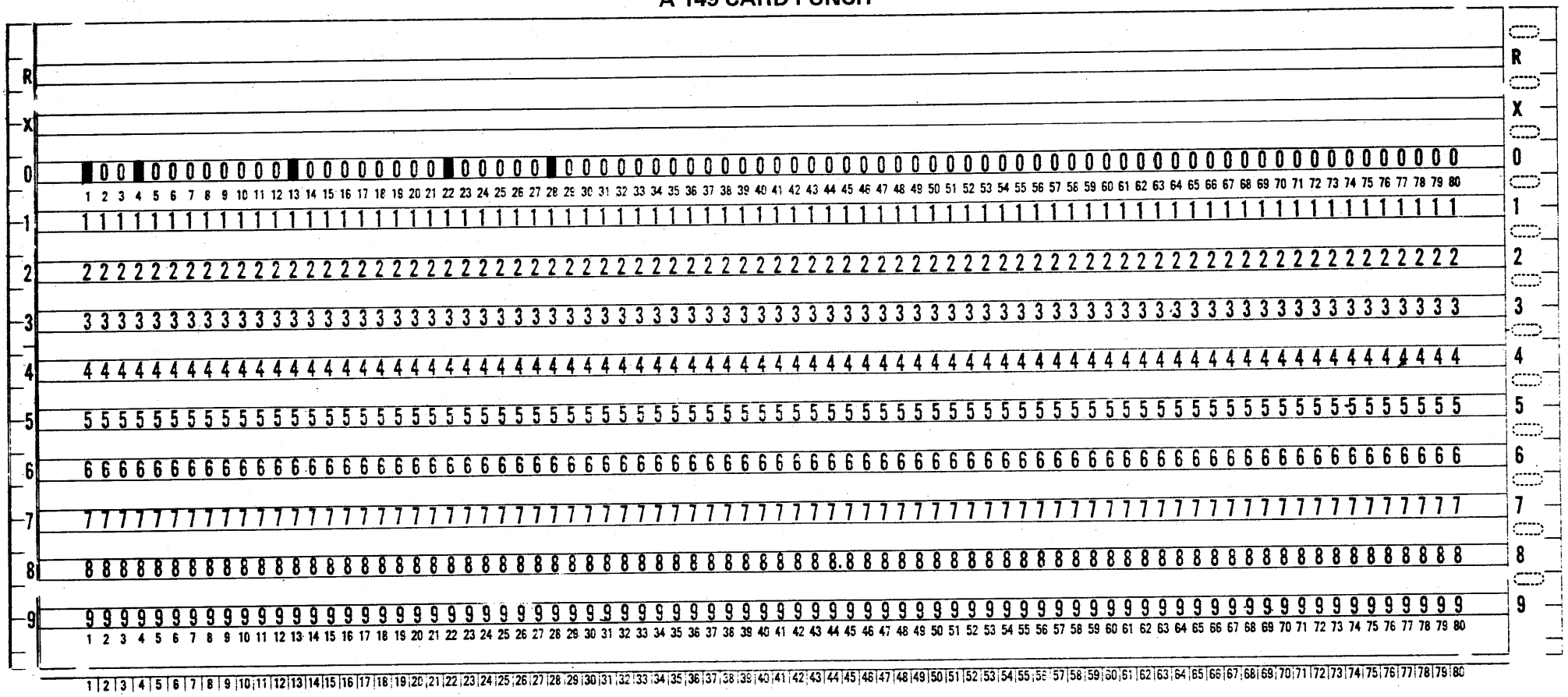
TOTAL	3,842.30
PRIOR	27,667.45
TO DATE	31,509.75
REPORT DATE	07 31--

13.55
13.56
13.57

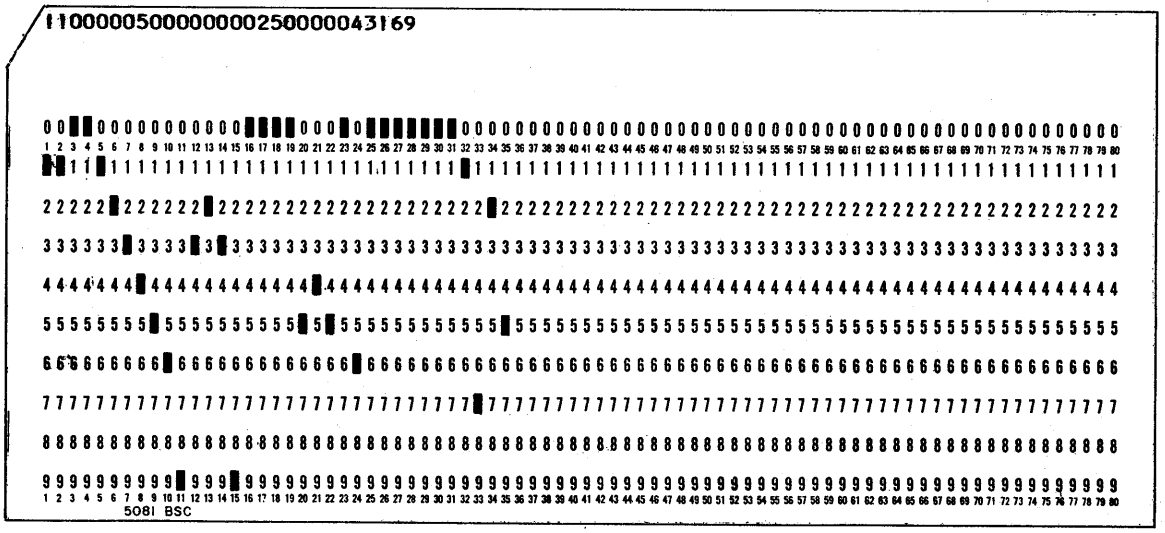
JUL 31'--
JUL 31'--
ACC REC TOT 93,927.05
PROD TOTALS

	2,563.50*
.01	500.00*
.02	27,785.00*
.03	5,682.00*
.04	56.32*
.05	265.00*
.06	.00*
.07	356.10*
.08	2,896.00*
.09	784.50*
.10	5,324.00*
.11	568.00*
.12	564.00*
.13	2,583.00*
.14	1,088.00*
.15	400.50*
.16	2,500.00*
.17	.00*
.18	.00*
.19	2,714.60*
.20	2,500.00*
.21	50.25*
.22	2,132.00*
.23	.00*
.24	.00*
.25	2,530.00*
.26	2,866.00*
.27	2,512.00*
.28	2,546.30*
.29	.00*
.30	.00*
.31	25.00*
.32	2,852.82*
.33	.00*
.34	.00*
.35	5,067.20*
.36	5,689.00*
.37	.00*
.38	258.96*
.39	.00*
.40	.00*
.41	.00*
.42	.00*
.43	.00*
.44	.00*
.45	5,632.00*
.46	.00*
.47	.00*
.48	.00*
.49	2,635.00*

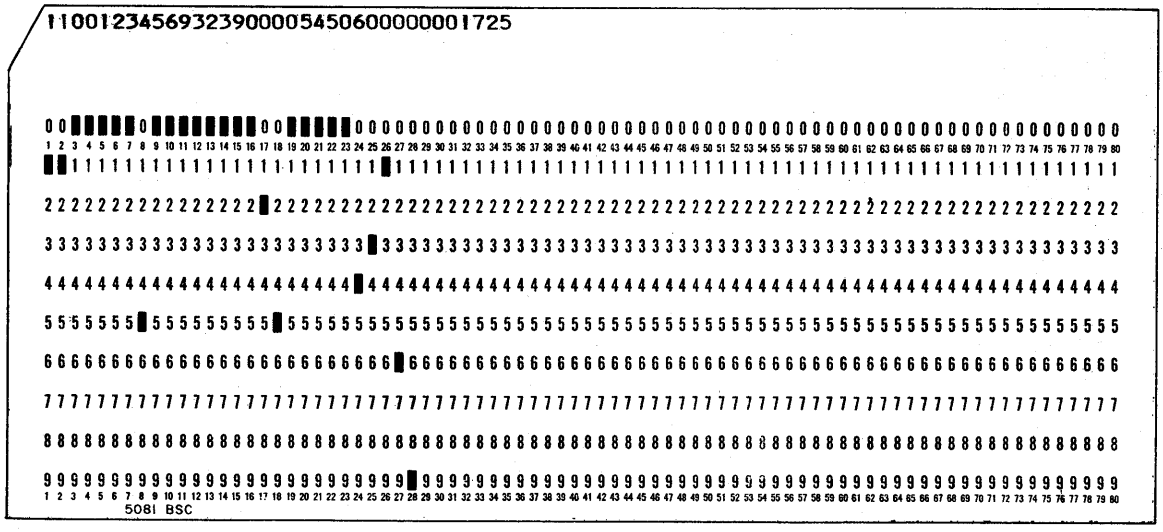
LAYOUT FOR
80-COLUMN PUNCHED CARD
BURROUGHS DATA RECORDING EQUIPMENT
SAMPLE PROGRAM – DRUM CARD – DRUM 2
A 149 CARD PUNCH



NOTE-DRUM 1 WILL USE A BLANK CARD IN THIS EXAMPLE



Sample Detail Card



Sample Summary Card

SECTION 5

FUNCTIONAL DESCRIPTION OF BASIC ASSEMBLERS

E 6000/E 8000 BASIC ASSEMBLER, PROGRAM NO. 1020100100/200/300

GENERAL

The E 6000/E 8000 Basic Assembler is a two-phase program which prepares a machine language program from a symbolic language program by substituting absolute operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

PHASE I

Phase I of the Basic Assembler assigns in memory and provides a print-out of the object program constants by the line printer and the error words, where applicable, on the control console.

PHASE II

Phase II of the Basic Assembler provides for printing by the line printer of each symbolic entry and generates the object program. Error words, where applicable, are printed on the line printer. The object program as generated in Phase II is punched into cards in machine language.

EQUIPMENT REQUIRED

The following system hardware is required for the Basic Assembler Program:

E 6000 system with 400 words of core memory, type to and from memory, card input, card output, index register and subroutine jump and return.

Style A 594 Card Reader

Style A 545 or A 505 Card Punch Control Unit with removable plugboard

Style 026B Card Punch with R.P.Q. 81041 or A 149 Card Punch

Style A 988-00 Line Printer

OR

E 8000 system with 400 words of core memory, type to and from memory, card input, punch from memory, index register, and subroutine jump and return.

Style A 594 Card Reader

Style A 149 Card Punch

Style A 988-00 Line Printer

This system utilizes Program No. 1030100400/500/600.

Note: Any control console, serial number F 191121 and below, requires the installation of RIN 3737-051 dated 3/29/68 to use with A 505 and A 149.

OPERATING INSTRUCTIONS

GENERAL

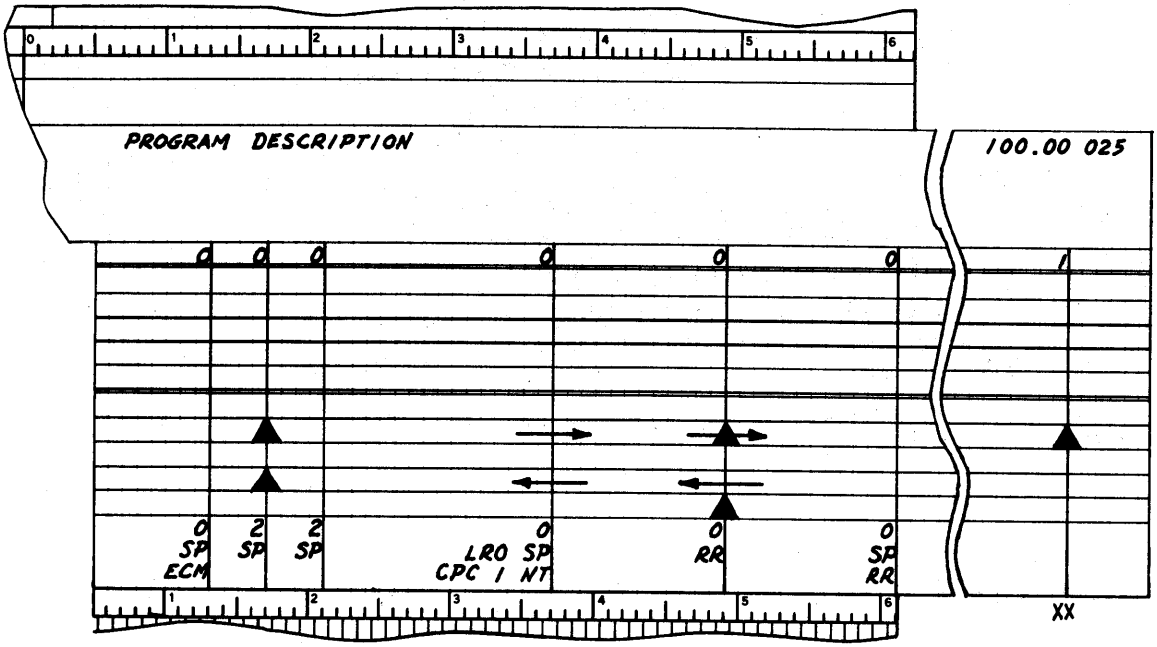
The following instructions cover the organization of the adjuncts and the card arrangements for the Basic Assembler.

LINE PRINTER

The line printer vertical format tape provides for printing on lines 7 through 60. A channel 4 hole is punched on each of these lines, a channel 2 hole is punched on line 59 and a channel 3 hole is punched on line 7.

CONTROL CONSOLE

On one schedule, the control unit must be programed as shown below:



Note: Change to CPC 0 for A 505 and A 149.

A 545 CARD PUNCH CONTROL

The wiring diagram for the A 545 Card Punch Control is shown in Appendix D-4.

026B CARD PUNCH

The front and rear drum cards are shown in Appendix D-2 and D-3.

A 505 CARD PUNCH CONTROL

The wiring diagram for the A 505 Card Punch Control is shown in App. D-7.

A 149 CARD PUNCH

The drum cards shown in Appendix D-5 and D-6 are to be used with the A 505. The drum cards shown in Appendix D-8 are to be used with E 8000 Punch From Memory.

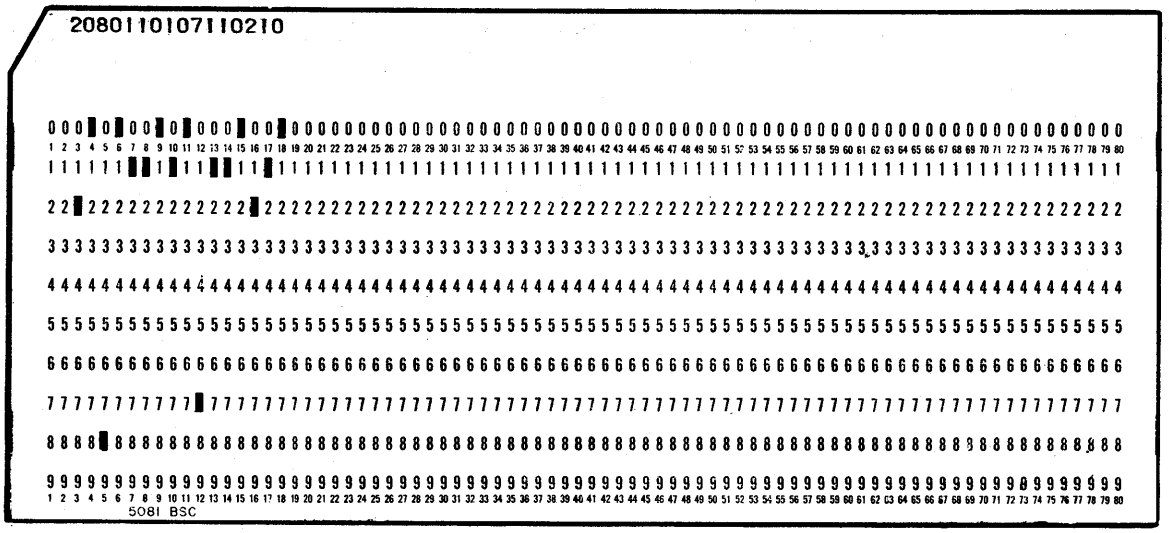
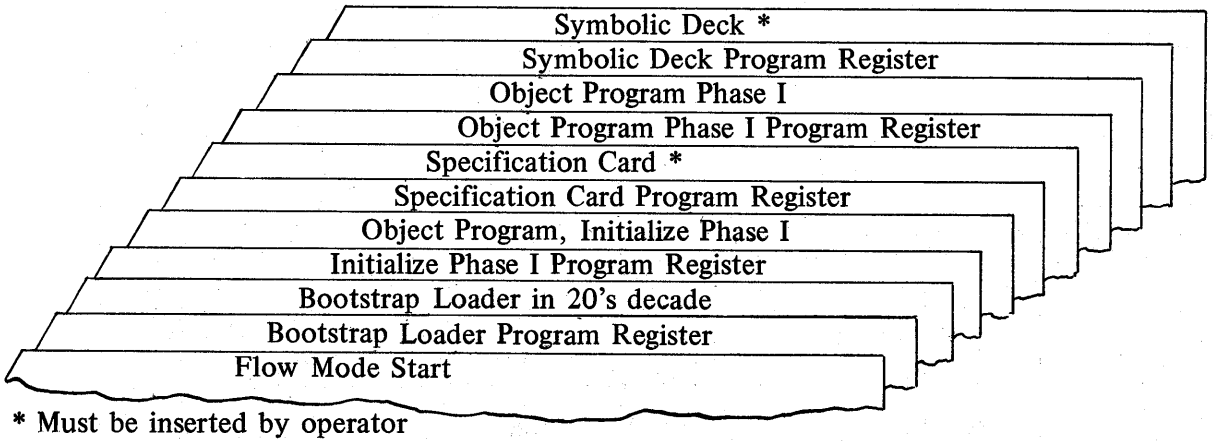


Fig. 5-1 Basic Assembler Specification Card

PHASE I CARD SEQUENCE

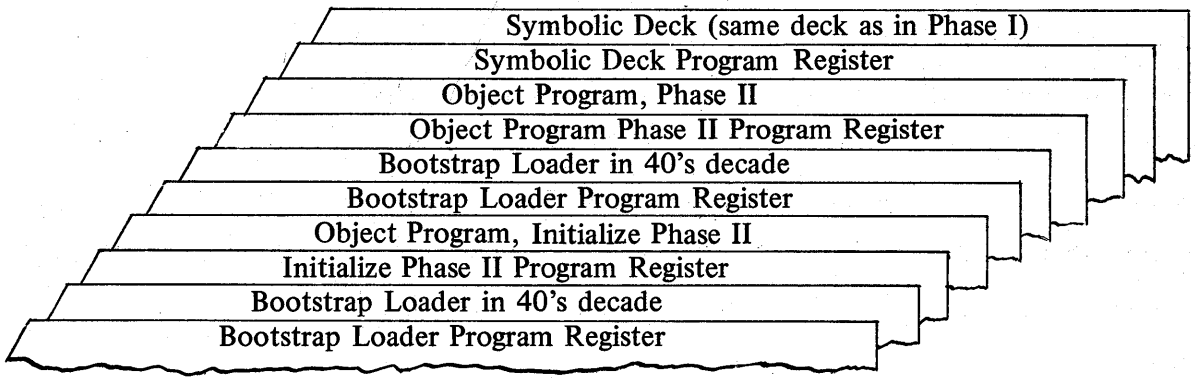
The input cards for Phase I are arranged in the following sequence



* Must be inserted by operator

PHASE II CARD SEQUENCE

The input cards for Phase II are arranged in the following sequence:



The following two examples are sample print-outs of Phase I and Phase II of the E 8000 Assembler:

```
014-1 NO 204


001           10001
002            3000
003      1200000008
004     10000000001
005           10042
006           10004
007           10005
008           10006
012           10010
013       600000001
014     10001410203
015         1010044
016         1010045
017       401000048
018          1410010
019       401400001
100     10000504001
101     10001410003
102               8
103     40008104047
104           10007
105       200000008
106           10044
107       600000048
080    557651066056
090    721176812871
081    640021575200
091    118672122488
082      7116551056
092    881267216811
083    711655105145
093    126721681212
084    556056640045
094    124871118622
085    556056640027
095    124871118672
086         5541522
```

Fig. 5-2 E 8000 Basic Assembler Print-out Phase I

```
253 2 4820          CRD      5C                                READ DATE CARD                1120
253 3 3220          TCP      5C                                READ L O DATE                 1130
254 1 3077          TPC      5T  7                             STORE                         1140
254 2 3230          TCP      5C 10                             READ H O DATE                 1150
254 3 3078          TPC      5T  8                             STORE                         1160
255 1 3221          TCP      5C  1                             READ CURRENT DATE             1170
255 2 3079          TPC      5T  9                             MOVE DATE TO WORK             1180
255 3 4717          TCPR     7J                                MOVE EOL                      1190
256 1 3060          TPC      4D                                MOVE EOL                      1200
256 2 3802          LCR      1B                                VC3                           1210
256 3 2840          PRT      4B                                SP TO TOP OF FORM             1220
257 1 1800          NOP                                                                      1230
257 2 3801          LCR      1A                                CLEAR CONT REG                1240
257 3 8716          JMP     35B                                GO TO HEADING ROUTINE         1250
258 1 3801          LCR      1A                                RDC P01                       1260
258 2 0879          PAC      5T  9                PRT CUR DATE POSI 080 TO 084 BT            1270
258 3 4870          CRD      5T                                READ 1ST CHARGE CARD          1280
259 1 3271    23N   TCP      5T  1                             TEST FOR TOTALS               1290
259 2 9392          BRM     44B                                TO TOTAL OVERLAY              1300
259 3 3803          LCR      1C                                RDL ALN P08                   1310
260 1 0800          PAC       P                 INSERT LED    POSI 084 TO 092 BT            1320
260 2 1880          SFL   80                                  TEST FOR FS OR RD ERROR       1330
260 3 8375          BRU     42B                                TO READ ERROR                 1340
261 1 8339          BRU     37B                                TO FILLED SHEET               1350
261 2 1800
261 3 1800
262 1 3805    23V   LCR      1E                                RDC LN4 P02                   1360
262 2 0820          PAC      5C                PRT PAT NO      POSI 092 TO 115 LN4           1370
262 3 1907          RFL    7                                   RESET FLAGS                   1380
                                                               ---TEST THIS PATIENT---       1390
263 1 3270          TCP      5T                                RD PAT NO CARD                1400
263 2 2220          SCP      5C                                COMPARE PAT NO LED            1410
263 3 9349          BRM     39E                                CARDS OR LED OUT OF SEQ       1420
264 1 3270    24A   TCP      5T                                READ PAT NO  CARD             1430
264 2 2220          SCP      5C                                COMPARE PAT NO LED            1440
264 3 9693          BRN     30B                                NOT-THIS PATIENT              1450
265 1 3273          TCP      5T  3                             READ PROF FEES                1460
```

Fig. 5-3 E 8000 Basic Assembler Print-out Phase II

ERROR CODE PRINT-OUT

GENERAL – The Assembler will detect certain types of errors and print an indication of these errors. Errors may occur in the Specification Card, Phase I or Phase II of the Assembler program.

SPECIFICATION CARD ERRORS – When an error is detected in the Specification Card further loading of the Phase I program is prevented. The message " SPEC CARD ERROR RESTART" is typed and the contents of card columns 7-18 of the Specification Card are printed out by the control console. The following errors in the Specification Card are detected by the Assembler:

1. The number of words are specified but the device is not specified.
2. The device is specified but the number of words are not specified.
3. A device is specified with more than 10 words.
4. The Specification Card is missing from the deck.

PHASE I ERRORS

GENERAL – In the Phase I program each error is assigned a number. When an error is detected, its corresponding number and the sequence number of the card on which the error occurred is printed on the control console. The error number prints from the corresponding digit position in memory. For example a type 4 error prints as 40.00. The following are the errors that may occur in Phase I of the program:

ERROR NO.	ERROR
1	No file declaration on Specification Card.
2	Overlay address not defined.
3	Invalid control register constant.
4	Number of program points exceeds 99
5	Number of labels exceeds 100

More than one invalid Control Register Constant (Error Code 3) is possible for a symbolic entry and if more than one the error print is cumulative. For example, if four errors occur for one symbolic entry, the error message would print 12.00 - 3+3+3+3.

At the conclusion of the Phase I program and if any errors have been detected, the Assembler executes a HLT 99 instruction and loading of Phase II is prevented.

PHASE II ERRORS

There are two types of Phase II errors detected by the Assembler. (1) A multiple defined label from Phase I which is detected during the initial part of the Phase II Assembler Program. (2) An error which occurs during the assembly of symbolic entry cards.

MULTIPLE DEFINED LABELS

When a multiple defined label error is detected, the message "MULT DEF XXX" (XXX is the label) is printed by the line printer following the listing of the constants. A separate message will be printed for each multiple definition detected. If multiple definitions are detected, the Assembler executes a HLT 99 and the loading of the remaining Phase II program is prevented.

PHASE II SYMBOLIC ENTRY ERRORS

When a Phase II Symbolic Entry Error is detected, it is assigned a reference number which is printed out by the line printer between the object program and the symbolic entry to which it pertains. The error number prints from the corresponding memory digit position. An "*" prints in all digit positions of the error word to the left of the digit position containing an error number. The following are the Phase II Symbolic Entry Errors:

ERROR NO.	ERROR
1	Sequence number of 00 or not in ascending sequence
2	Invalid operation code
3	Program Point undefined
4	Invalid BRL (to greater than a MA 200 address)
5	Program exceeds 400 words
6	Program Point greater than 9
7	Label undefined
8	Shift greater than 12 digits
9	Address greater than 200, non-branch instruction

If NOP's are generated by the Assembler when the program is over 400 (Error Code 5), the error is cumulative. For example, if two NOP's are generated with an address over 400, it would print out the error message 100000 (5+5 in position 5) with the next symbolic entry.

No special halt (HLT) is provided in Phase II; the Assembler always executes a HLT 00 at address 011.

UNDETECTED ERRORS

There are certain types of errors that will not be detected. In the analysis by the Assembler of the OP code and modifier, and the CTL constant mnemonic, a value is computed from the digits of each character. This will allow some invalid combinations of mnemonics to be interpreted as a valid code which may not be the code intended.

As an example, no error message will result from these misinterpreted codes:

1. A Control Register Constant of "RCL" is interpreted as "RDL".
2. An OP code of "TCE" would be interpreted as "TTM".

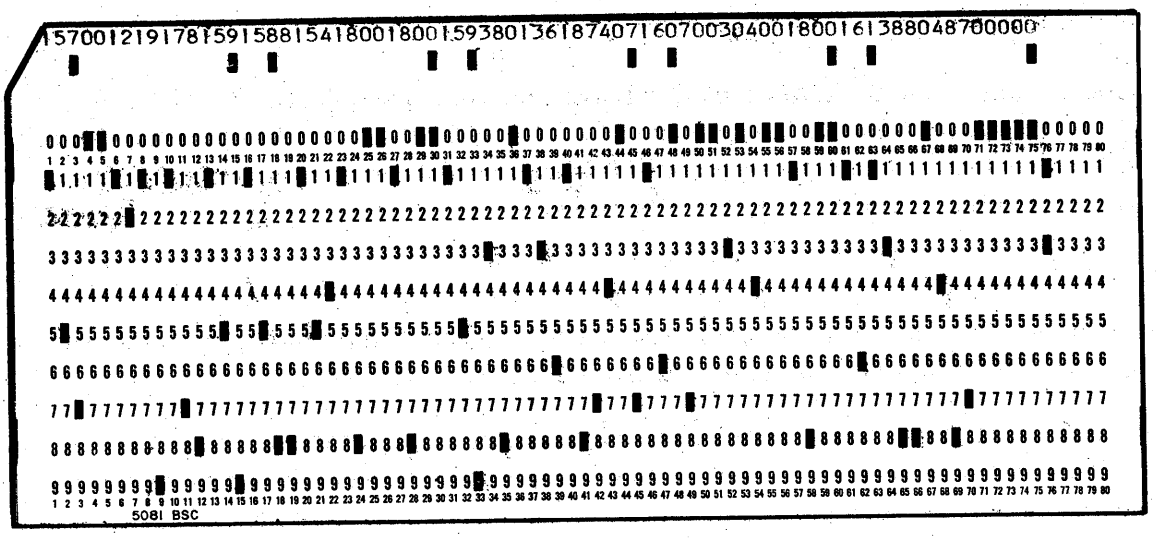


Fig. 5-4 Sample A 592 Object Program Card

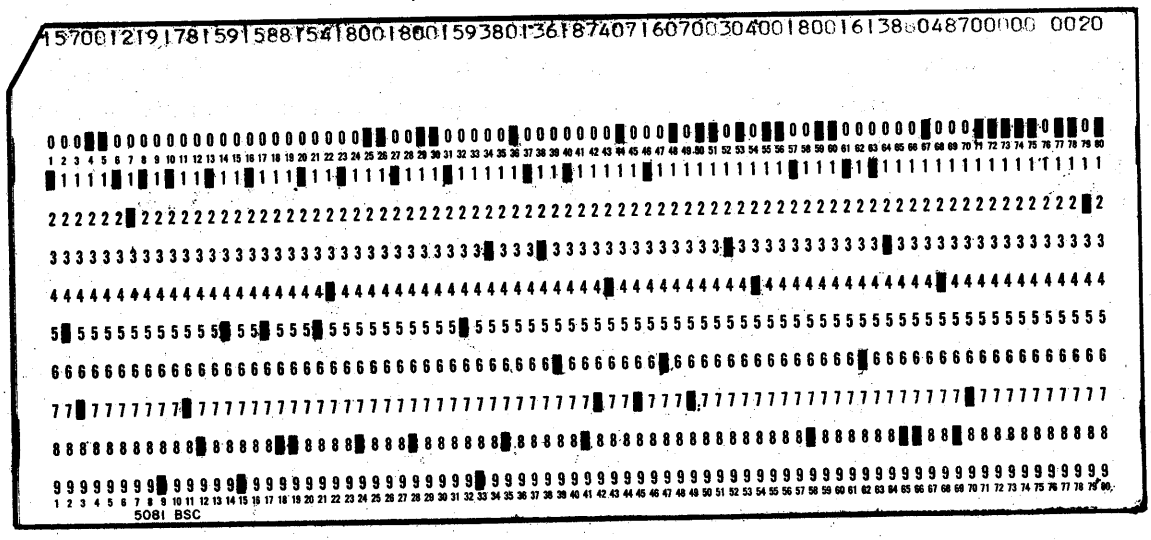


Fig. 5-5 Sample A 594 Object Program Card

E 6000/E 8000 BASIC ASSEMBLER NO. 1020102800 B 3500 VERSION

GENERAL

The B 3500 version of the Basic Assembler, will assemble Basic Assembler symbolic cards and provide an object program in A 592 or A 594 bootstrap format and an assembly listing on the line printer. The B 3500 version of the assembler provides for 200 labels and 300 program points instead of 100 and 99 respectively as furnished on the E 6000/E 8000 assembler.

PHASE I

Phase I of the B 3500 version of the Basic Assembler assigns in memory and provides a print-out of the object program constants and error words, where applicable, by the line printer.

When an error is detected, the associated error message is printed on the line printer and assembly is aborted. The error condition is corrected and assembly is started again. This process is continued until all Phase I errors are located and corrected.

After Phase I is assembled the assembler will cause a print-out of any duplication of labels. Following the correction of any duplicate labels, the assembler will proceed with Phase II.

PHASE II

Phase II of the B 3500 version of the Basic Assembler, which is from a disk file, thus eliminating reloading the symbolic program deck, provides for printing by the line printer each symbolic entry and the object program generated. Error codes, where applicable are printed by the line printer in the error column. The object program as generated is punched into cards in machine language.

ENVIRONMENT

The following system hardware and software is required for the B 3500 version of the E 6000/E 8000 Assembler:

HARDWARE

B 3501 Processor with 60K memory

Card Reader - 200, 800 or 1400 CPM

Card Punch - 100 or 300 CPM

Console Typewriter

Line Printer - 120-print position minimum

Disk - 1 module

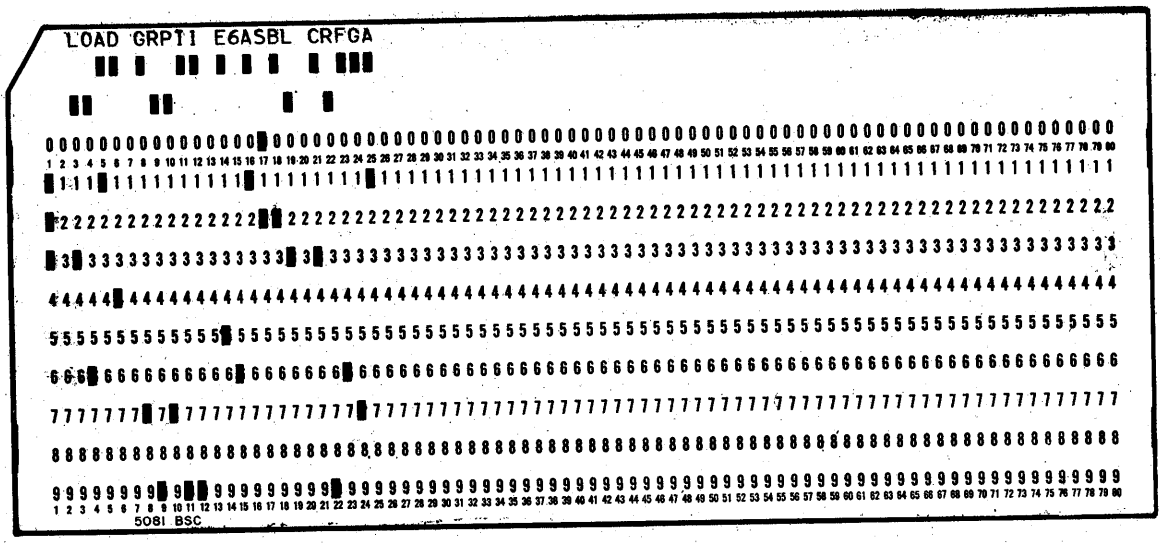
Magnetic Tape - 1 unit for loading program

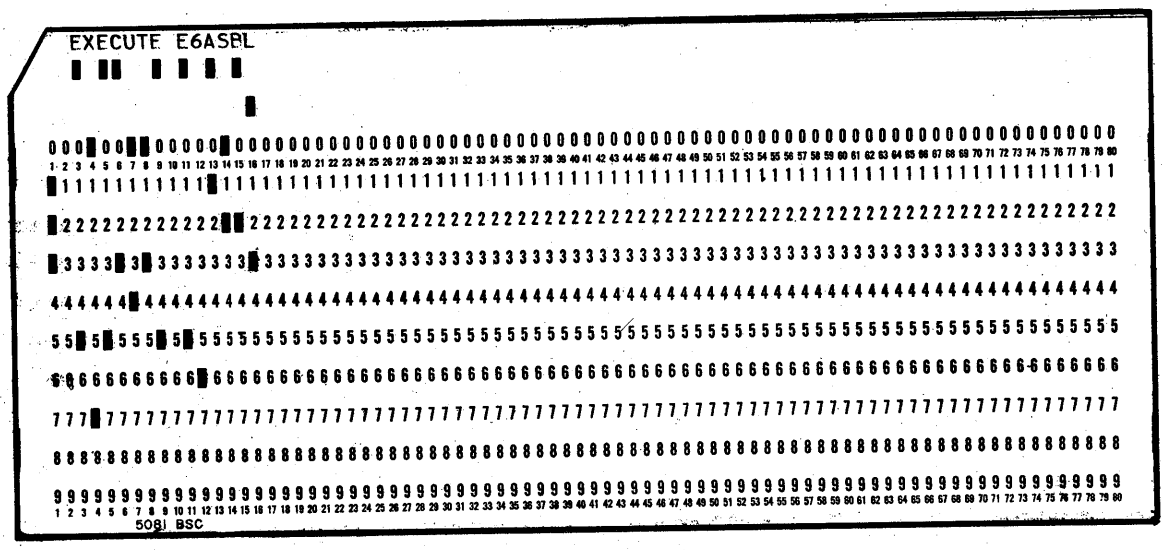
SOFTWARE

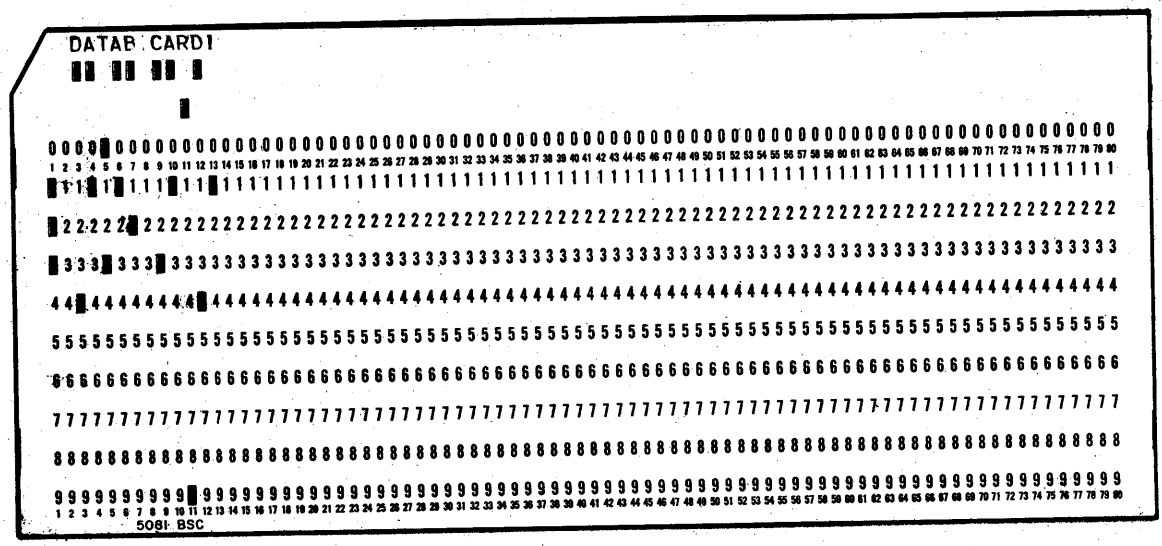
Master Control Program (MCP)

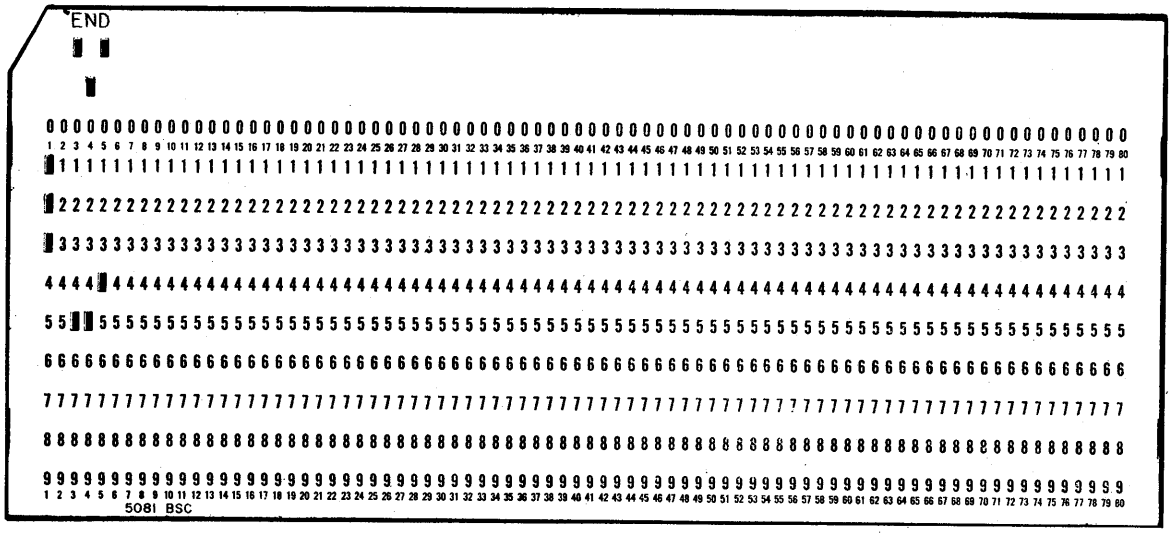
MCP CONTROL CARDS

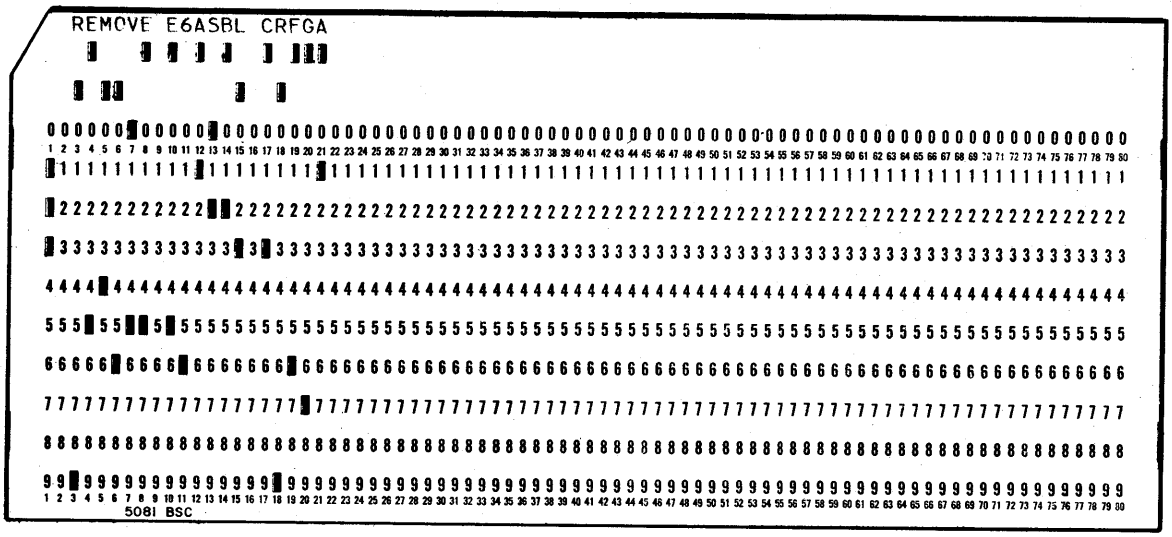
The following five (5) Control Cards are required for the B 3500 version of the E 6000/E 8000 Basic Assembler:


LOAD GRPT1 E6ASBL CRFGA
5081 BSC


EXECUTE E6ASBL
5081 BSC


DATAB CARD1
5081 BSC





INPUT – OUTPUT

INPUT

The input is an E 8000 Basic Assembler symbolic card deck as described in Appendix C-1. The card deck is punched in Burroughs Common Language (BCL) or in Extended Binary Coded Decimal Interchange Code (EBCDIC) card codes. Standard coding is BCL.

Certain changes are required in the specification card and the coding of alpha characters in the symbolic deck.

THE SPECIFICATION CARD is the same as with the E 6000/E 8000 Basic Assembler displayed in Figure 5-1, with the following additions:

CARD COL.	CODE	FUNCTION
19	P	To allow punch from memory commands to be valid.
20-24	RESEQ	If resequenced symbolic deck is desired, otherwise blank.
25	E	If EBCDIC codes are desired in resequenced symbolic deck, otherwise blank.
26-28	Blank	
29	N	For A 592 cards, otherwise blank.
30-35	Syntax	Provides assembly for diagnostic check without punching object deck. All other output options, if specified, are cancelled.

THE SYMBOLIC DECK input to the Assembler is in the source language format as described in Appendix C-1 except for the E 8000 Tab and Ret Key codes. In order to program these codes into an "ALP" or "APH" message, the following special characters and related codes must be substituted.

	NOT END ALPHA			END ALPHA		
		Card Code			Card Code	
Key No.	Character	BCL	EBCDIC	Character	BCL	EBCDIC
1	(	12-8-5	12-8-5	)	11-8-5	11-8-5
2	<	12-8-6	12-8-4	>	8-6	0-8-6
3	&	12	12	%	0-8-4	0-8-4
4	$	11-8-3	11-8-3	=	0-8-5	8-6
Vert. Sp.	:	8-5	8-2	#	8-3	8-3

OUTPUT

The following output is provided on the B 3500 version of the E 6000/E 8000 Assembler.

1. Line Printer Listing
2. Object Program Deck
3. Resequenced Symbolic Deck - if specified.

LINE PRINTER LISTING – Each page contains two heading lines. The first line has the "HDR" information on the left and "PAGE NO. XXX" on the right. The second line contains the following line printer assembly listing, printed from left to right:

ERRORS	–	Error Code Print
LOC	–	Machine Location
S	–	Syllable instruction
AUTOLOAD	–	Constants or machine language
SEQ. #	–	Symbolic card sequence
PLAB	–	P= Program Point: LAB = Label
OP.M	–	OP = OP code; M = Modifier
LN	–	Length
ADR	–	Address
WI	–	Word Increment or Program Point
CONSTANT	–	Constants
CTL REG CST	–	Control Register Constants
REMARKS	–	Remarks

At the end of the assembly listing, the number of Labels and Program Points used will be printed on the next line printer page.

RESEQUENCE SYMBOLIC DECK – The coding for a resequenced symbolic deck, if specified, is the same as for the Basic Assembler. The Ident., c.c 1-6, will dup, in each card. Standard card codes are BCL unless EBCDIC codes are specified on the specification card.

OPERATING INSTRUCTIONS

The operation to load the assembler program and to assemble is as described in the following sections:

Load Assembler Program

1. Mount the GRPII library tape on the tape drive.
2. Feed the card coded as follows through the Card Reader:

   ? LOAD GRPII E6ASBL CRFGA
   (? = 1-2-3 punch)
3. The B 3500 Supervisory Printer (SPO) prints out "E6ASBL LOADED" and "CRFGA LOADED", which indicates assembly may be begun.

To Assemble

1. Ready Card Punch
2. Load the following cards in the Card Reader in the order indicated:

   A. ? EXECUTE E6ASBL or ? EXECUTE E6ASBL value 2 = 0000nn

      (1) ? EXECUTE E6ASBL if standard printing format of 8 lines per inch on 11" high paper.

      (2) ? EXECUTE E6ASBL value 2 = 0000nn if it is desired to have the number of print lines vary from the standard format. "nn" is the number of print lines required and must be greater than 10.

B. ? DATAB Cardl (for "BCL" coded input) or
? DATA Cardl (for "EBCDIC" coded input)

C. Symbolic Deck

First Card = Specification Card

Last Card = End

D. ? End

(? = 1-2-3 punch)

3. Depress "Reset" and "Start" on Card Reader
4. After "EOJ E6ASBL" prints on SPO, remove print-out and object program cards.
5. Repeat above for each additional assembly desired. Assembler program does not have to be reloaded.
6. To remove program from the disk when assembly is completed, feed the following card through the Card Reader:

? REMOVE E6ASBL CRFGA

(? = 1-2-3 punch)

```
014-1 NO 204                                                                    PAGE NO. 001
ERRORS   LOC S --AUTOLOAD--- SEQ.# PLAB OP.M LN ADR WI CONSTANT     CTL REG CST  -----REMARKS----------

         001   000000010001          01A
         002   000000003000          01B
         003   001200000008          01C
         004   010000000001          01D
         005   000000010042          01E
         006   000000010004          01F
         007   000000010005          01G
         008   000000010006          01H
         012   000000010010          01J
         013   000600000001          01K
         014   010001410203          01L
         015   000001010044          01M
         016   000001010045          01N
         017   000401000048          01P
         018   000001410010          01R
         019   000401400001          01S
         100   010000504001          01T
         101   010001410003          01V
         102   000000000008          02A
         103   040008104047          02B
         104   000000010007          02C
         105   000200000008          02D
         106   000000010044          02E
         107   000600000048          02F
         080   557651066056          06D
         090   721176812871
         081   640021575200
         091   118672122488
         082   007116551056
         092   881267216811
         083   711655105145
         093   126721681212
         084   556056640045
         094   124871118622
         085   556056640027
         095   124871118672
         086   000005541522
         096   888887212714
         087   416400555554
```

Fig. 5-6 E 8000 Basic Assembler B 3500 Version Print-out Phase I

```
014-1 NO 204
ERRORS   LOC  S  --AUTOLOAD---  SEQ.#  PLAB  OP.M  LN  ADR  WI  CONSTANT     CTL REG CST   -----REMARKS----------
                                0118          TPC       05T  09              MOVE DATE TO WORK
         255  2  3079           0119          TCPR      07J                  MOVE EOL
         255  3  4717           0120          TPC       04D                  MOVE EOL
         256  1  3060           0121          LCR       01B                  VC3
         256  2  3802           0122          PRT       04B                  SP TO TOP OF FORM
         256  3  2840           0123          NOP
         257  1  1800           0124          LCR       01A                  CLEAR CONT REG
         257  2  3801           0125          JMP       35B                  GO TO HEADING ROUTINE
         257  3  8716           0126          LCR       01A                  RDC P01
         258  1  3801           0127          PAC       05T  09   PRT CUR DATE POSI 080 TO 084 BT
         258  2  0879           0128          CRD       05T                  READ 1ST CHARGE CARD
         258  3  4870           0129    23N   TCP       05T  01              TEST FOR TOTALS
         259  1  3271           0130          BRM       44B                  TO TOTAL OVERLAY
         259  2  9392           0131          LCR       01C                  RDL ALN P08
         259  3  3803           0132          PAC       00P       INSERT LED POSI 084 TO 092 BT
         260  1  0800           0133          SFL  80                        TEST FOR FS OR RD ERROR
         260  2  1880           0134          BRU       42B                  TO READ ERROR
         260  3  8375           0135          BRU       37B                  TO FILLED SHEET
         261  1  8339
         261  2  1800
         261  3  1800                                                       RDC LN4 P02
         262  1  3805           0136    23V   LCR       01E
         262  2  0820           0137          PAC       05C       PRT PAT NO POSI 092 TO 115 LN4
         262  3  1907           0138          RFL  07                        RESET FLAGS
                                0139                                         ---TEST THIS PATIENT---
         263  1  3270           0140          TCP       05T                  RD PAT NO CARD
         263  2  2220           0141          SCP       05C                  COMPARE PAT NO LED
         263  3  9349           0142          BRM       39E                  CARDS OR LED OUT OF SEQ
         264  1  3270           0143    24A   TCP       05T                  READ PAT NO  CARD
         264  2  2220           0144          SCP       05C                  COMPARE PAT NO LED
         264  3  9693           0145          BRN       30B                  NOT THIS PATIENT
         265  1  3273           0146          TCP       05T  03              READ PROF FEES
         265  2  1034           0147          APC       05C  14              STORE STRIPE
         265  3  1127           0148          APC       08V                  STORE TOTAL
         266  1  3272           0149    24H   TCP       05T   2              READ CHG CODE AND CHG
         266  2  3076           0150          TPC       05T  06              STORE TO PRT
         266  3  4010           0151          TPCS      00B                  STORE CHG WORKING
         267  1  5811           0152          CLR       00K                  CLEAR K
         267  2  7606           0153          SLTD 06   00P                  POSI CODE
         267  3  3409           0154          TKC       00W                  STORE CODE WORKING
```

Fig. 5-7 E 8000 Basic Assembler B 3500 Version Print-out Phase II

ERROR CODE PRINT-OUT

The errors that are detected with the B 3500 version of the E 8000 Assembler are described in the following sections:

PHASE I

Two types of errors may be detected during the first pass of the symbolic cards; those that provide both a printed indication on the Line Printer and abort the assembly, and those that provided a printed indication only.

ASSEMBLY ABORT ERRORS

CONDITION	SEQUENCE NUMBER	MESSAGE
Invalid Specification Card	NONE	Invalid Specification Card
Alpha file programed but not on Specification Card	xxxxx	Spec. Card Reqmt - Alpha
Printer file programed but not on Specification Card	xxxxx	Spec. Card Reqmt - Printer
Card file programed but not on Specification Card	xxxxx	Spec. Card Reqmt - Card
Ledger file programed but not on Specification Card	xxxxx	Spec. Card Reqmt - Ledger
Number of program points exceeds 300	xxxxx	Program Point table exceeded
Number of labels exceeds 200	xxxxx	Label table exceeded
Invalid Control Register constant coded	xxxxx	Invalid Control Register Constant
Invalid Overlay Address Coded	xxxxx	Overlay Address Error
Exceeded Memory with Constants, files, control registers & reserve memory	xxxxx	Exceeded memory with constants
Duplicate use of Labels	NONE	xxx* Label definition errors (* name of duplicate label)

All of the above error conditions except "Duplicate Use of Labels", abort the assembly as soon as detected. All duplicate labels are printed out at the end of Phase I and the assembly is then aborted.

NON-ABORT ERRORS

CONDITION	SEQUENCE NUMBER	MESSAGE
Invalid Alpha Characters when decoding "ALP" or "APH" constants	xxxxx	"A" in the Error column and the sequence number of the symbolic card in the SEQ.# column.

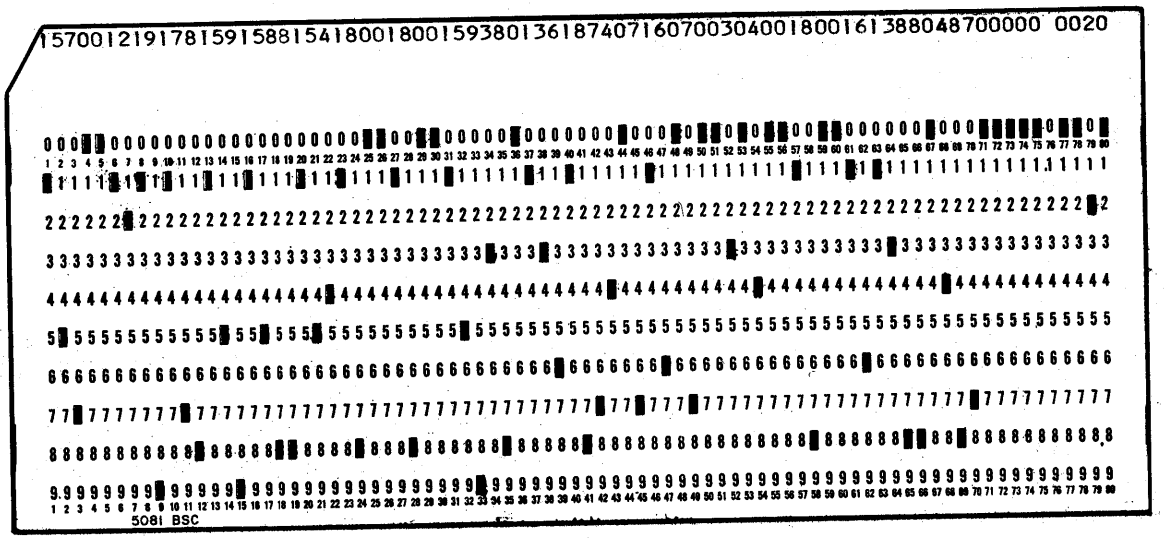
PHASE II

During the second pass of the symbolic cards (from the disk file) the following error codes will print out in the ERROR column if the indicated corresponding error condition is detected.

ERROR CODE	CONDITION
1	Sequence Error
2	Illegal OP Code
3	Program Point undefined
5	Address greater than 399
6	Invalid Program Point
7	Invalid Address
8	Shift length greater than 12
9	Address greater than 199
F	Flags incorrectly keypunched (Card columns 22 or 25 punched)
OVR ADR ERR	Invalid overlay address

OBJECT PROGRAM DECK

The following is an illustration of an object program card which is formated for the A 594 Card Reader:



The format is as follows:

CARD COL.	CODING
1-75	3/12 Bootstrap format
76	Blank for A 594 Card Reader -1-3 punch for A 592 Card Reader
77-80	Sequence Number

E 6000/E 8000 BASIC ASSEMBLER NO. 1020102900 – B 300 VERSION

GENERAL

The B 300 version of the Basic Assembler, will assemble E 6000/E 8000 Basic Assembler symbolic cards and provide an object program in A 594 bootstrap format and an assembly listing on the Line Printer.

The B 300 version is functionally identical to the E 6000/E 8000 Basic Assembler.

PHASE I of the Basic Assembler, B 300 version, assigns in memory and provides a print-out of the object program constants by the line printer and the error words, where applicable, on the line printer.

PHASE II of the Assembler provides for printing by the line printer of each symbolic entry and generates the object program. Error words, where applicable, are printed on the line printer. The object program as generated in Phase II is punched into cards in machine language.

ENVIRONMENT

The following system hardware and software is required for the B 300 version of the Assembler.

HARDWARE

B 300 Processor with 9.6 memory

Card Reader - 200, 800 or 1400 CPM

Card Punch - 100 or 300 CPM

Line Printer - 120-print position a minimum

Transfer and Branch Address modification

INPUT – OUTPUT

INPUT

The input is an E 8000 Basic Assembler symbolic card deck punched in Burroughs Common Language (BCL) card codes.

Certain changes are required in the specification card and the coding of alpha characters in the symbolic deck.

THE SPECIFICATION CARD is the same as with the E 6000/E 8000 Basic Assembler with the following exception:

CARD COL.	CODE	FUNCTION
19	P	To allow punch from memory commands to be valid.
30-35	Syntax	Provides assembly for diagnostic check without punching object deck.

THE SYMBOLIC DECK input to the Assembler is in the source language format described in Appendix C-1 except for the E 8000 tab and return key codes. In order to program these codes into an "ALP" or "APH" message, the following special characters and related codes must be substituted:

		CARD CODE		
Key No.	Char.	Not End Alpha	Char.	End Alpha
1	(	12-8-5	)	11-8-5
2	<	12-8-6	>	8-6
3	&	12	%	0-8-4
4	$	11-8-3	=	0-8-5
VS	:	8-5	#	8-3

OUTPUT

The following output is provided on the B 300 version of the Assembler:

1. Line Printer Listing
2. Object Program Deck

LINE PRINTER LISTING

Each page contains two heading lines. The first line has the "HDR" information on the left and "PAGE NO. xxx" on the right. The second line contains the following line printer assembly listing, printed from left to right:

ERRORS	–	Error Code Print
LOC	–	Machine Location
S	–	Syllable instruction
AUTOLOAD	–	Constants or machine language
SEQ. #	–	Symbolic card sequence
PLAB	–	P = Program Point
	–	LAB = Label
OP.M	–	OP = OP Code
	–	M = Modifer
LN	–	Length
ADR	–	Address
WI	–	Word increment or Program Point
CONSTANT	–	Constants
CTL REG CST	–	Control Register Constants
REMARKS	–	Remarks

At the end of the assembly listing, the number of Labels and Program Points used will be printed on the next line printer page.

OBJECT PROGRAM DECK

The format for the object program deck is as follows: (See Page 5-18)

CARD COL.	CODING
1-75	3/12 Bootstrap format
76	Blank
77-80	Sequence number

OPERATING INSTRUCTIONS

STEP	OPERATION
1	READY CARD PUNCH, TURN VALIDITY OFF ON CARD READER
2	LOAD PHASE I PROGRAM DECK INTO THE CARD READER (160 CARDS)
3	DEPRESS "CLEAR" AND "LOAD" ON CONSOLE
4	DEPRESS "RESET" AND "START" ON CARD READER
5	AFTER PHASE I LOADS: DEPRESS "CLEAR" AND "CONTINUE" ON CONSOLE
6	LOAD SYMBOLIC CARDS INCLUDING "SPEC. CARD" AND "END CARD" INTO CARD READER
7	DEPRESS "RESET" AND "START" ON CARD READER TO INITIATE FIRST PASS OF SYMBOLIC CARDS
8	AFTER SYMBOLIC CARDS HAVE BEEN PROCESSED, LOAD PHASE II PROGRAM INTO CARD READER
9	DEPRESS "RESET" AND "START" ON CARD READER
10	AFTER PHASE II IS LOADED: RELOAD CARD READER WITH SAME SYMBOLIC CARD DECK AS PROCESSED IN PHASE I
11	DEPRESS "RESET" AND "START" ON CARD READER
12	REMOVE LISTING, OBJECT PROGRAM, AND POWER DOWN CARD PUNCH

Note: Upon detection of duplicate use of labels (Halt 998), a depression of the "CONTINUE" switch will let the Assembler continue and print out an Assembly Listing.

END OF ASSEMBLY INDICATED BY A HALT 999.

ERROR CODE PRINT-OUT

The errors that are detected with the B 300 version of the Assembler are described in the following sections:

PHASE I

Two types of errors may be detected during the first pass of the symbolic cards; those that provide both a printed indication on the Line Printer and abort the assembly, and those that provide a printed indication only.

ASSEMBLY ABORT ERRORS

MACHINE HALT	CONDITION
997	Exceeded Memory with Constants, files, control registers and reserve memory

NON-ABORT ERRORS

MESSAGE	CONDITION
"A" in the Error column and the sequence number of the symbolic card in the SEQ # column.	Invalid Alpha Characters when decoding "ALP" or "APH" constants
SPEC CRD ERR	Invalid Specification Card
INV FILE-ALP	Alpha File programed but not on Specification Card
INV FILE-PRT	Printer File programed but not on Specification Card
INV FILE-CRD	Card File programed but not on Specification Card
INV FILE-LDGR	Ledger File programed but not on Specification Card
PP LIMIT EX	Number of Program Points exceeds 300
LAB LIMIT EX	Number of Labels exceeds 200
INV CTL REG	Invalid Control Register Constant Coded
OVR ADR ERR	Invalid Overlay Address Coded
LABEL DEFINE ERROR	Duplicate Use of Labels

PHASE II

During the second pass of the symbolic cards the following error codes will print out in the Error column if the indicated corresponding error condition is detected.

ERROR CODE	CONDITION
1	Sequence Error
2	Illegal OP code
3	Program Point undefined
5	Address greater than 399
6	Invalid Program Point
7	Invalid Address
8	Shift length greater than 12
9	Address greater than 199
F	Flags incorrectly keypunched (card columns 22 or 25 punched)
L	Shift length incorrectly keypunched (card columns 22 or 25 punched)
W	WI/PP Field incorrectly keypunched or no Label or Program Point. cc 29 should be blank, cc 30 blank with Program Point. (WI) should be right justified.
OVR ADR ERR	Invalid overlay address

SECTION 6

OBJECT PROGRAM LOADING

This section contains the procedures, punch card and striped ledger formats involved when loading a program into the E 8000 system with either an A 594 Card Reader, A 592 Card Reader or A 4004 Automatic Striped Ledger Reader.

External print locations and programing are also indicated and should be used as a guide.

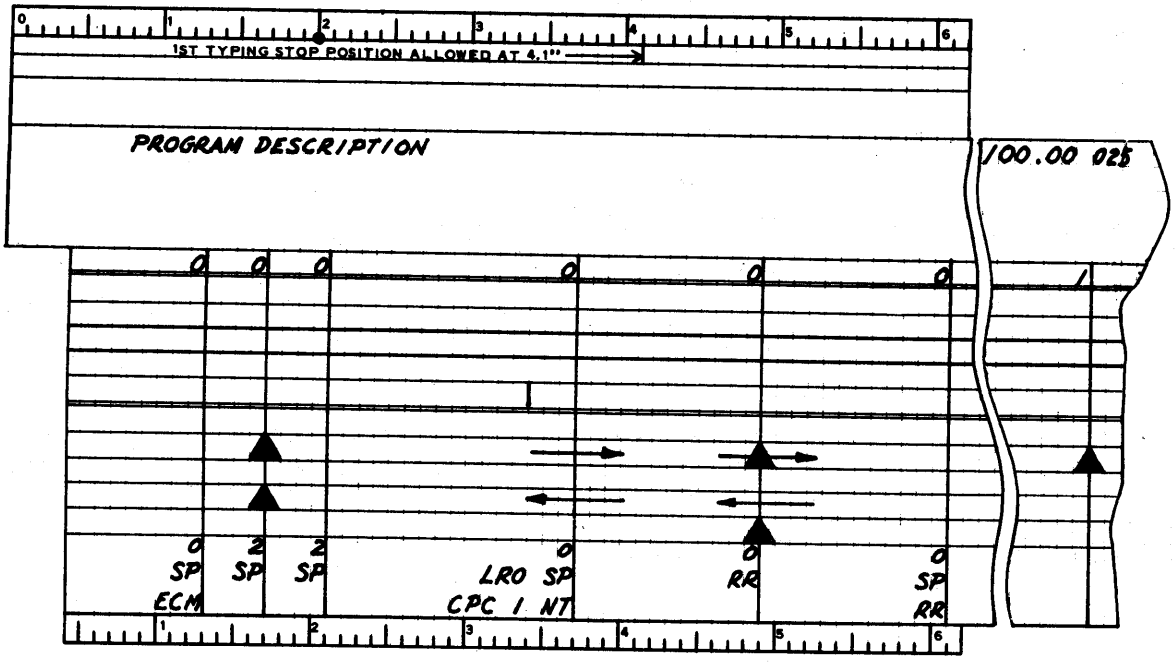
The programing and operating procedures used to load a program with either punched card reader is the same. The only real difference comes in the punch card formats and the reading of format register cards. These differences will be indicated as discussed.

A 594 CARD READER

To load a program with the A 594 Card Reader, certain cards are necessary and must be in the following sequence:

1. Bootstrap loader
2. Bootstrap format register card
3. Bootstrap
4. Object program format register card
5. Object program
6. Proof format register card
7. Proof card

To load the above program cards will normally utilize three operating positions. They are shown in the following example. Notice that the position check no. is indicated on the position no. line; and that the print control number is added in the special instructions section.



The operating procedures used to load a program with the above cycling positions is:

1. Use key 4 (type tab key) to go to scale position 13.
2. Depress "Program Load Switch" (PGL) and Motor Bar 2.
3. Depress "Program Start Switch" (STR) and Motor Bar 2.
4. First position of program.

When the above procedures are followed this is the results if the A 594 Card Reader is used.

When the "PGL" switch is depressed, and Motor Bar 2 used, the bootstrap loader card will be read automatically in program load mode. Read-in will start at memory location 00P. (You may wish to review the punched card in discussion, Section 2 - Page 2-20.)

The format of the bootstrap loader card is:

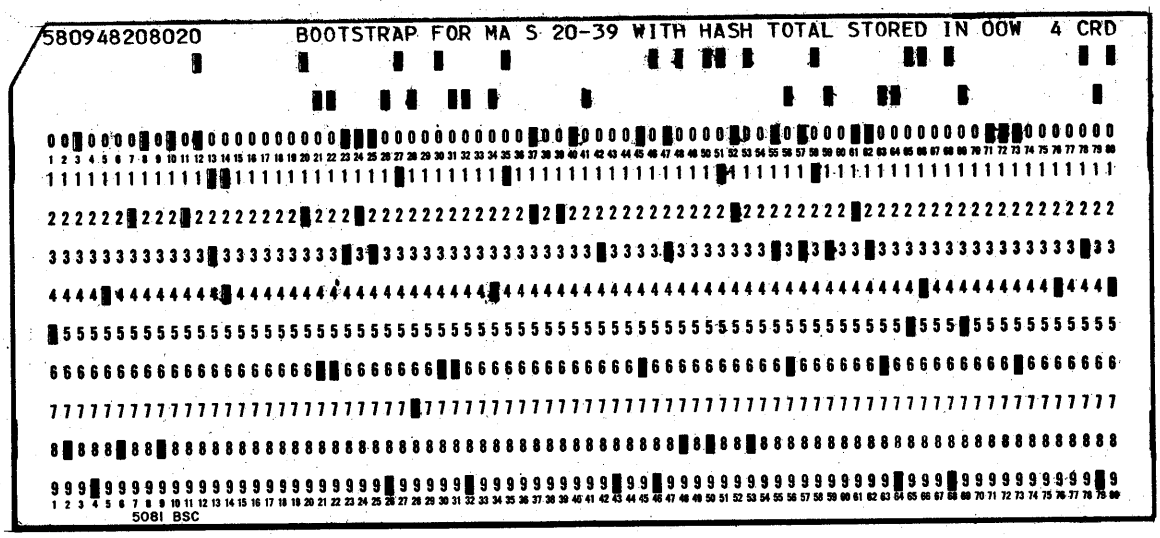


Fig. 6-1 First Card - Bootstrap Loader

This card will load three commands into 00P. They are:

5809 CLR-009
4820 CRD-020
8020 BRU-020

Remember, this card was loaded under program load mode and therefore utilizes a "12" punch in c.c. 12 to step memory and a "1/3" punch in c.c. 13 to end read in and a "1/4" punch in c.c. 14 to end card in.

When the program start switch (STR) is depressed with Bar 2 at scale position 17, the central processor will begin internal operation at the memory location indicated on the keyboard of the console. Since no memory location was selected internal processing will begin at 00P. Memory location 00P contains the program loaded from the bootstrap loader to clear MA 009 which is the location used to develop proof of the program load operation, to feed a punched card into the 20's decade, and then to branch to memory location 020.

The CRD-020 instruction, the second instruction in 00P, is a single card read instruction. From the sequence of punched cards to be loaded indicated earlier, it is seen that the next punched card to be read is the "bootstrap format register" card followed by the bootstrap card. As indicated in the punched card discussion the internal "CRD" instruction will read one punch card unless that card is a format register card, in which case it will read that card and the next one following. Therefore, the CRD-20 instruction loaded into 00P from the bootstrap loader card will load the format register to read the bootstrap card and then read the bootstrap card into the 20's decade.

The card formats for the bootstrap format register card and the bootstrap itself is:

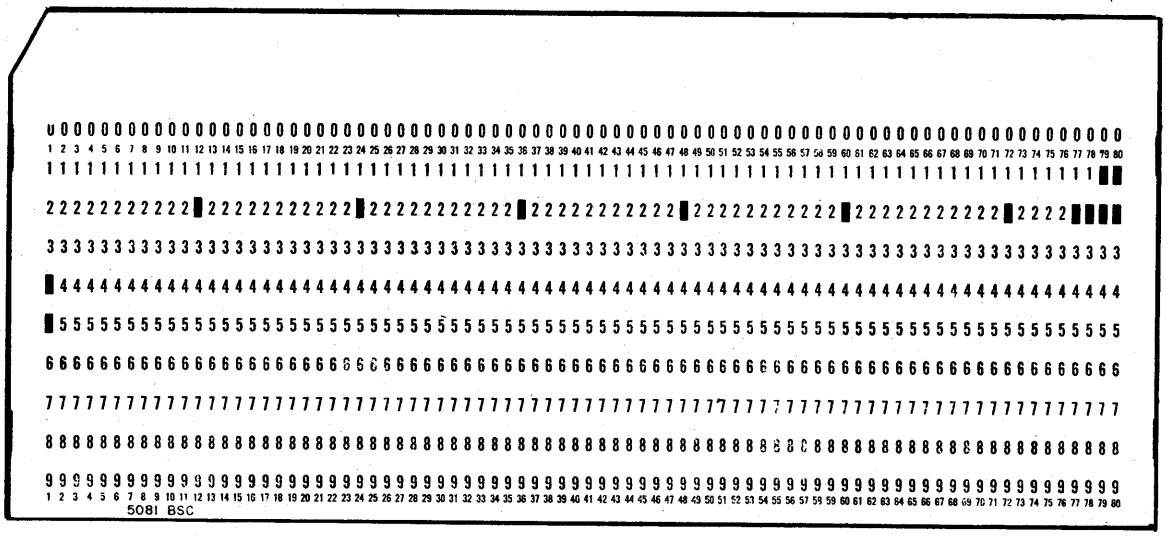


Fig. 6-2 Second Card - Bootstrap Format

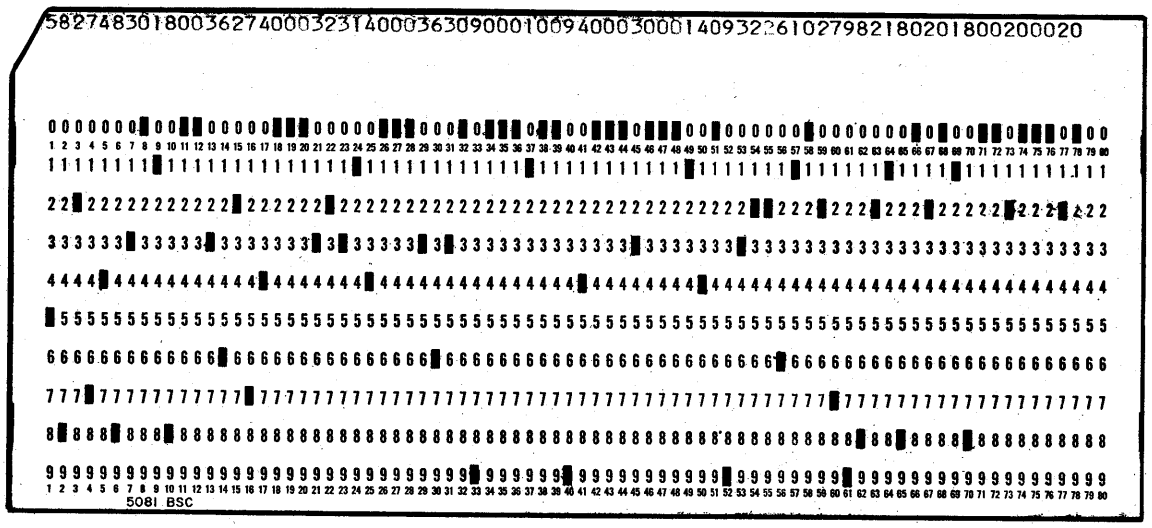


Fig. 6-3 Third Card - Bootstrap

Once these two cards have been read the program will advance to the third instruction in memory location 00P which is "BRU-020". The processor will then branch to the bootstrap program just loaded from the above card.

The bootstrap program is now executed; it is a program to load "any" detail program. Therefore, it will load the object program format register card, the object program, the proof format register card and the proof card.

The card formats for:

1. The object program format register card
2. The object program cards
3. The proof format register card
4. The proof card

are as indicated below.

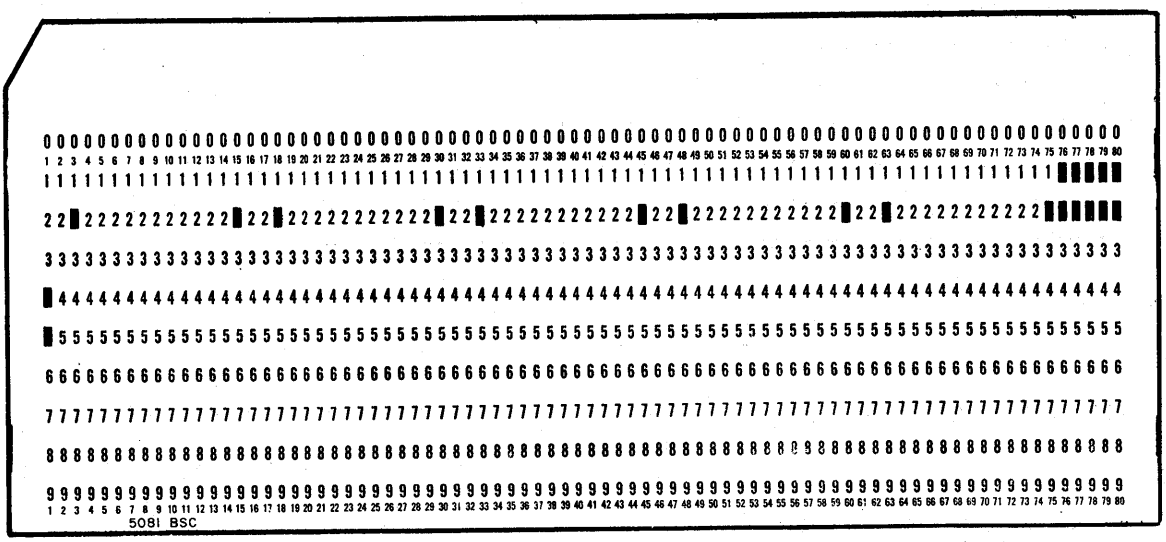


Fig. 6-4 Fourth Card - Object Program Format

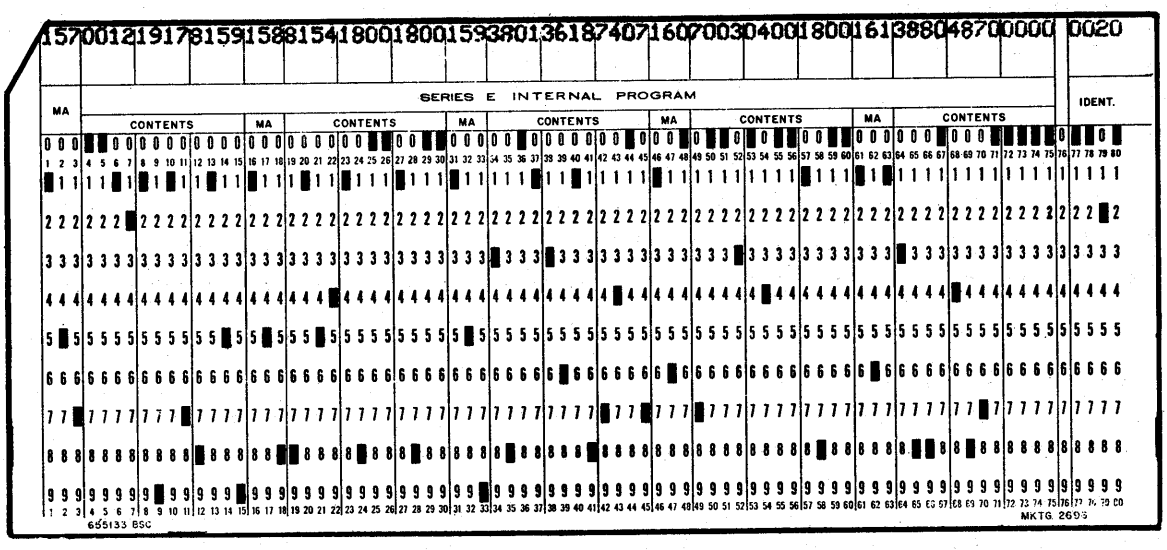


Fig. 6-5 Object Program Cards*

* The card containing the minus MA number (starting MA) must be the last object program card.

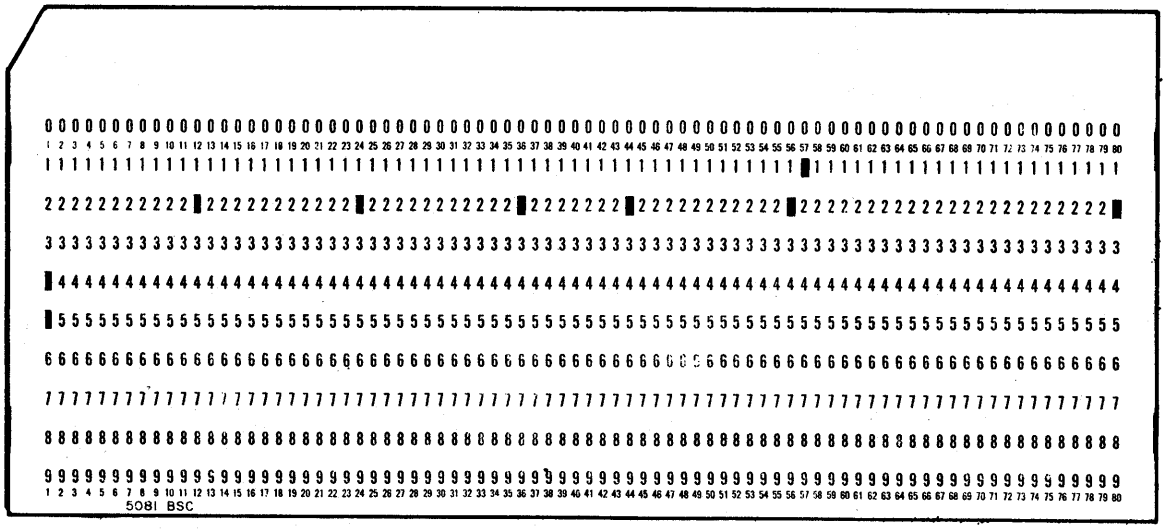


Fig. 6-6 Next to Last Card - Proof Format

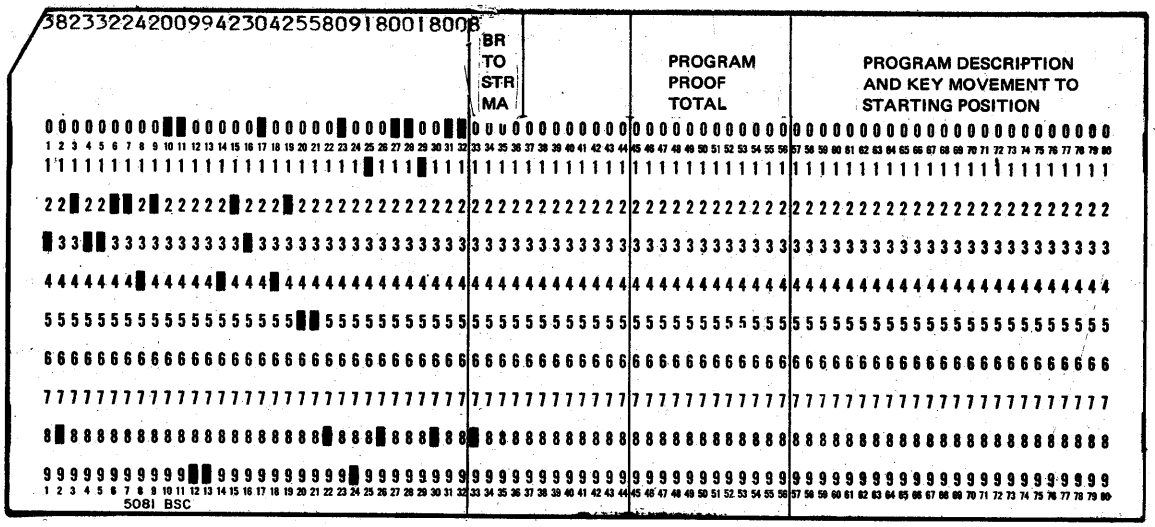


Fig. 6-7 Last card - Proof *

* NOTE: Starting MA (columns 34-36), program proof total (columns 45-56), and program description and key movement to starting position (columns 57-80) must be key punched on this card. If A 592 card reader is used a 0 is punched in cc 44 and a 1/2 punch in cc 57.

Let us now direct our attention to the object program cards. Because of the bootstrap program, the object program cards may be loaded in any sequence. The program data will be transferred to the appropriate memory locations and accumulated for proof. (MA numbers and program words are accumulated.)

The last detail program card must contain a minus MA number and, in the second field of the card, two commands which load the proof format card and the proof card as well as a branch command. The minus memory location number will cause the processor to branch out of the bootstrap program into the proof routine loaded from the proof card.

At this time the carriage has moved to scale position 49.

The proof routine will compare the proof total read from the proof card with the proof accumulation of the object program cards. If they agree, the system will type from memory at scale position 49 after which it will come to rest at position 60 where variable constants such as a date, etc., may be loaded. If the proof numbers do not agree no typing will occur. The operator will use K2 returning to scale position 13 where she will abort the program and reload. (To abort program, depress any console control key *, ', +, -, and the PGA light.)

Following is a list of instructions found on the bootstrap loader, bootstrap, the last detail and the proof cards. Also indicated is the MA these instructions are read into.

MEMORY LOCATION	INSTRUCTION		FUNCTION
BOOTSTRAP LOADER			
00P	5809	CLR-009	Clear proof location
	4820	CRD-020	Feed card (bootstrap format and bootstrap)
	8020	BRU-020	Branch to bootstrap program
BOOTSTRAP			
020	5827	CLR-027	Clear counter
	4830	CRD-030	Feed card into 30's decade (detail pro)
	1800	NOP	No operation
021	3627	TCK-027	Move counter setting to MA 011
	4000	TIX	Index register
	3231	TCP-031	Amount in 31 to P(33 to P etc.)
022	4000	TIX	Index Register
	3630	TCK-030	MA# in 30 to K (32 to K etc.)
	9000	BRM-000	BR. to MA 000 to activate program on last detail
023	1009	APC-009	Add amount to proof
	4000	TIX	Index register
	3000	TPC-XXX	Amount MA# specified in 00K
024	1409	AKC-009	Add MA# to proof
	3226	TCP-026	Move increment to work (20002)
	1027	APC-027	Increment counter
025	9821	BRL-021	BR. to MA 21 for next command group
	8020	BRU-021	BR. for next card
	1800	NOP	No operation
026		CNST	Increment CNST. of 20002
027		CNST	Working CNST. .00

MEMORY LOCATION	INSTRUCTION		FUNCTION
LAST DETAIL CARD			
030 (any even MA-30-38)	–	–	Minus MA number
031 (any odd MA 31-39)	4820	CRD-020	Feed card (proof card)
	8020	BRU-020	Branch to proof routine
PROOF CARD	1800	NOP	No operation
020	3823	LCR-023	Load control register (normally blank)
	3224	TCP-024	Move detail proof total to work
	2009	SPC-009	Subt proof total developed in program load
021	9423	BRN-023	If not equal branch to halt (0000)
	0425	TFM-025	Type program description
	5809	CLR-009	Clear proof total used
022	1800	NOP	No operation
	1800	NOP	No operation
	8XXX	BRU-XXX	Branch to starting MA in Program

When a program is written on coding sheets the first thing that is done is to keypunch the symbolic program per the instructions indicated in the appendix. When the symbolic deck is completed the detail program deck is created by the assembler in bootstrap format. The assembler will automatically insert on the last detail program card a minus MA number and a branch to the starting memory location of the program assembled. (Note: No proof routine is considered at this point.)

The pre-punched bootstrap loader, bootstrap format register, bootstrap and object program format register cards are now added to the front of the detail program cards assembled.

The program is then loaded and debugged. When the program is correct the last detail card is removed and re-punched. The commands following the minus memory location should be changed to "CRD-020" and "BRU-020" as shown on Fig. 6-4. A pre-punched proof card is then selected and to it is added:

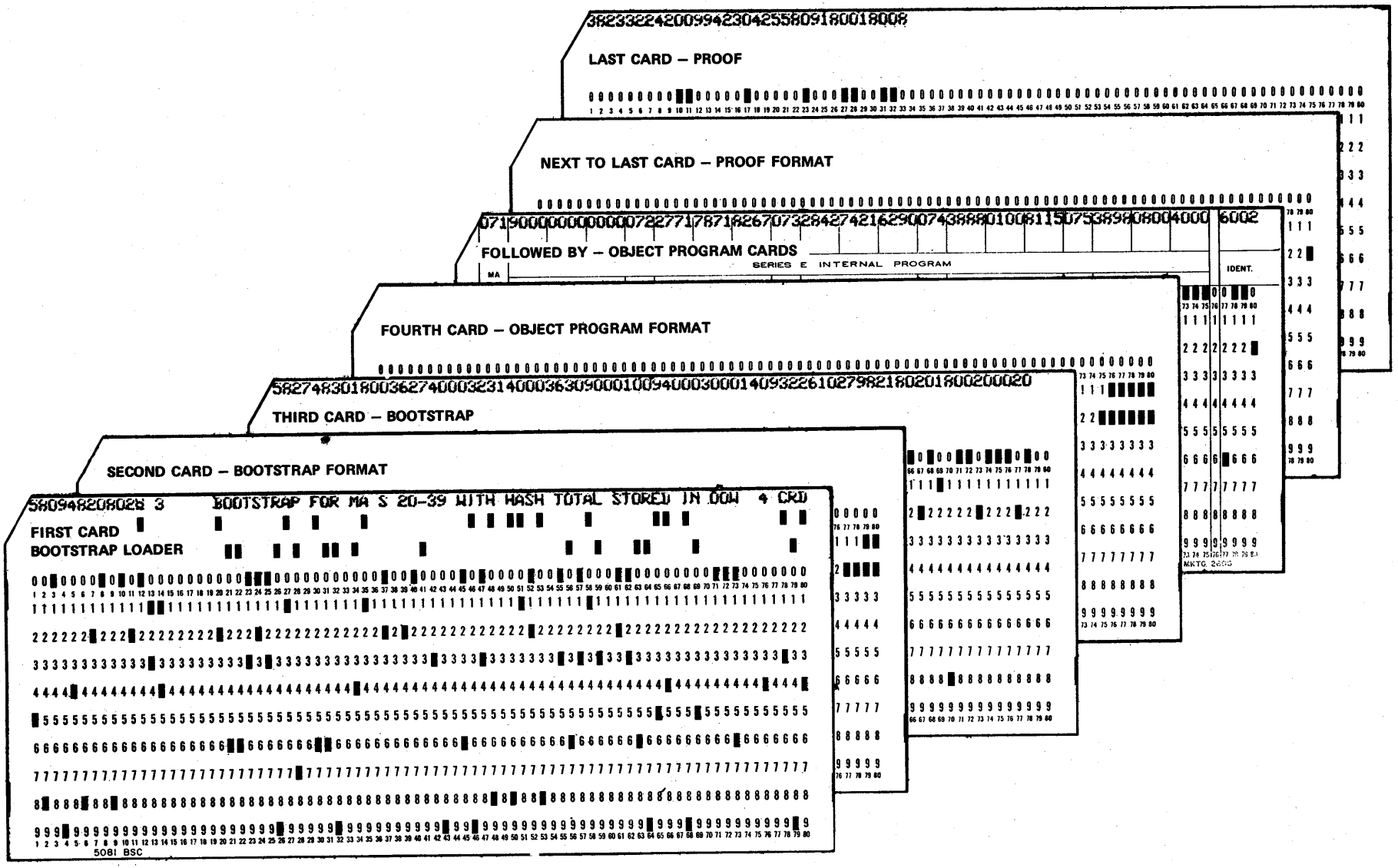
A. Starting MA in columns 34-36.

B. Program proof total in columns 45-56. The proof total is obtained by loading the debugged detail program deck in the A 594 card reader and manually reading MA 009.

C. The desired program description and key movement to the starting position in columns 57-80.

This card as shown on Fig. 6-5 is then added with the last detail card just corrected, to the proof format register card and then combined with the rest of the program deck.

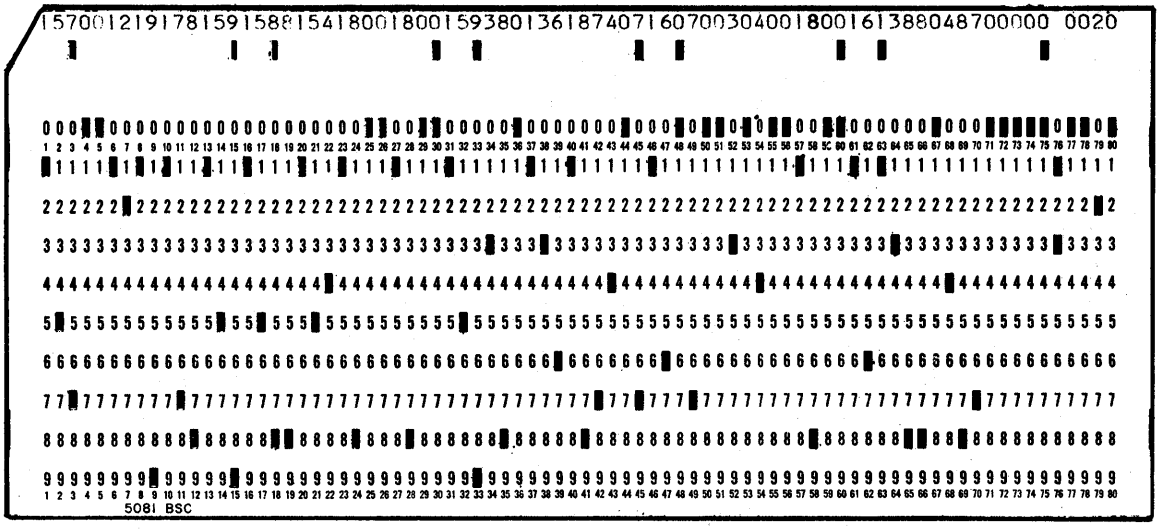
The program is now ready for use. The following example shows a complete program deck which indicates the proper sequence of all cards.


LAST CARD – PROOF
NEXT TO LAST CARD – PROOF FORMAT
FOLLOWED BY – OBJECT PROGRAM CARDS
SERIES E INTERNAL PROGRAM
IDENT.
FOURTH CARD – OBJECT PROGRAM FORMAT
THIRD CARD – BOOTSTRAP
SECOND CARD – BOOTSTRAP FORMAT
FIRST CARD
BOOTSTRAP LOADER
BOOTSTRAP FOR MA S 20-39 WITH HASH TOTAL STORED IN 00W 4 CRD
5081 BSC

A 592 CARD READER

The programing and operating procedure for loading a program with the A 592 Card Reader is identical to that described above.

The only difference is in the card design.



Since the A 592 does not have a card format register it would not be necessary to use any format register cards. All cards used would need "12" zone punches for word marks and "1/3" punches prior to any data not being read as shown in the above example. The Proof Card would require the necessary 12 punches, a "0" punch in cc 44 and a 1/2 punch in cc 57.

Other than the format register cards, the sequence of cards would remain the same.

That is:

A. Bootstrap loader
B. Bootstrap
C. Object program cards
D. Proof card

At this point, please refer to Page 6-1 showing the suggested external programing procedures.

If it is necessary to place a print position between scales 17 and 49, all that must be done to get proper print-out of the program description is to place a K4 movement prior to the program description on the proof card and a lane 27 knock-off at scale 49. See Page 5-2.

Notice that at no point was memory cleared. This is because all total and working areas will be declared as constants and initialized to zero as the program is loaded. See the discussion of the "CST" pseudo instruction in Section 2.

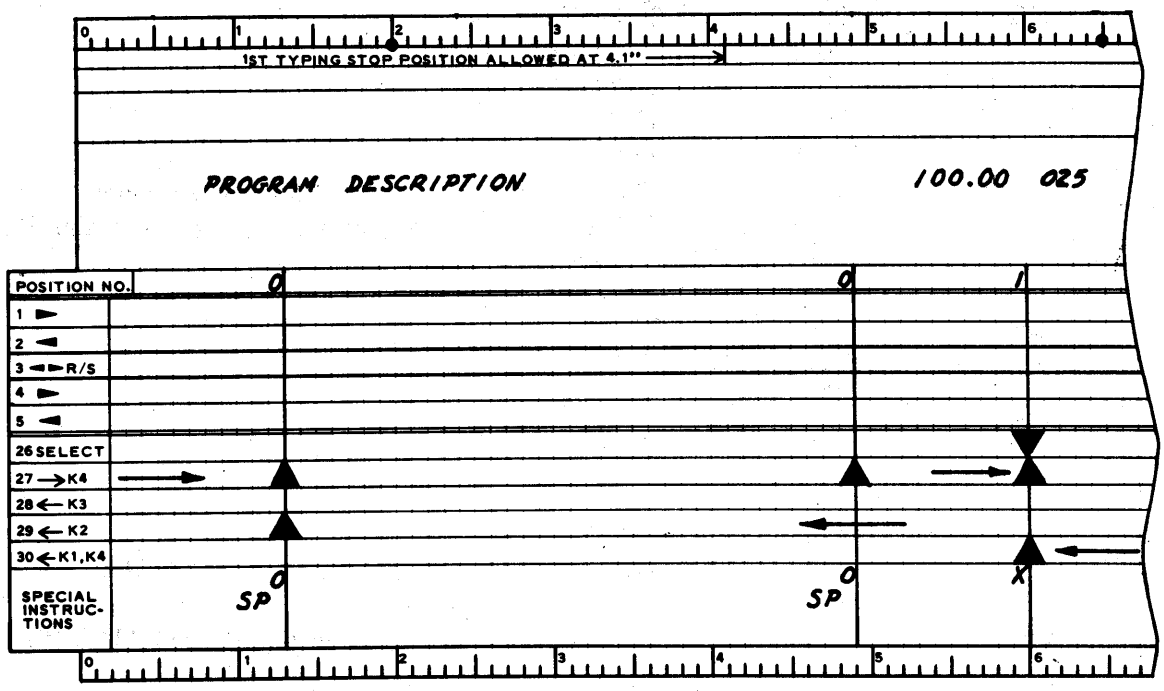
A 4004 STRIPED LEDGER READER

The operating procedure to load a program with the A 4004 Automatic Striped Ledger Reader is somewhat different than when using the punch card reader.

To load a program with the A 4004 will require certain striped ledgers in the following sequence :

A. Bootstrap loader and bootstrap (one ledger).
B. Detail program ledgers.
C. Proof ledger.

To load the above program ledgers will normally utilize two operation positions. They are shown in the following example. Notice that the position check no. is indicated on the position no. line; and that the print control number is added in the upper right-hand corner of the special instructions section.



The operating procedures used to load a program with the above cycling positions are:

A. Use key 4 (type tab key) to go to scale position 13.

B. Manually clear and load memory locations 010 and 011.
   1. MA 010 = 3,811,080,080.20
   2. MA 011 = 0,210,000,000.00

C. Use key 4 (type tab key) to go to scale position 49.

D. Select memory location 010 on the keyboard of the console and depress the "program start switch" (STR) with Motor Bar 2.

When the above procedures are followed and the A 4004 Ledger Reader is used, these are the results.

A. When MA 011 is loaded with 0,210,000,000.00, the numeric representation of feed ledger (FDL) and read ledger (RDL) have been loaded. The value 3,811,080,080.20 loaded in MA 010 represents three commands. They are LCR-011, PAC-00P, and BRU-020.

B. When memory location 010 is selected and the "STR" switch depressed with Bar 2, the above three commands will be executed. That is, the control register will be loaded with FDL and RDL, a ledger will be fed through the ledger reader without cycling the console, and the program will advance to MA 020.

The first ledger read (Step B above) contains the bootstrap ledger and bootstrap programs. The programs contained on this ledger are read into memory locations 020-037 and will cause all other ledgers to be processed.

Following is a list of instructions found on the bootstrap loader and bootstrap program ledger.

MEMORY LOCATION	INSTRUCTION		FUNCTION
020	3228	TCP-028	
	3040	TPC-040	
	3229	TCP-029	
021	3041	TPC-041	
	3230	TCP-030	
	3042	TPC-042	"Bootstrap Loader"
022	3231	TCP-031	Move Bootstrap Program to the 040 decade.
	3043	TPC-043	
	3232	TCP-032	
023	3044	TPC-044	
	3233	TCP-033	
	3045	TPC-045	
024	3234	TCP-034	
	3046	TPC-046	
	3235	TCP-035	
025	3047	TPC-047	
	3236	TCP-036	Counter of 0002 0002 0002
	3048	TPC-048	
026	3237	TCP-037	Constant of 0,210,000,000.00 (FDL, RDL)
	3049	TPC-049	
	5810	CLR-010	Clear Manual load
027	5809	CLR-009	Clear MA used for proof
	8040	BRU-040	Branch to Bootstrap Program
	1800	NOP	No operation
028	5810	CLR-010	
	3849	LCR-049	
	0800	PAC-00P	
029	1880	SFL-080	
	8040	BRU-040	"Bootstrap Program"
	1800	NOP	
030	3610	TCK-010	
	4000	TIX	
	3221	TCP-021	
031	4000	TIX	
	3620	TCK-020	
	9047	BRM-047	
032	1009	APC-009	
	4000	TIX	
	3000	TPC-000	
033	1409	AKC-009	
	3248	TCP-048	"Bootstrap Program"
	1010	APC-010	
034	7641	SLTS-1-00P	
	2210	SCP-010	Last ledger will have a minus
	9442	BRN 042	amount in an even numbered
035	1042	APC-042	MA. Place this constant
	8040	BRU-040	440960096779
	1800	NOP	in the next higher MA. Proof
036	0002 0002 0002	CONSTANT	ledger may be made up with the same
037	0210 0000 0000	CONSTANT	MA location as punch card location.

The function of "Bootstrap Loader" Program is to move the "Bootstrap" Program to the 040 decade so that the detail program cards may then be read into the 20's and 30's decades.

After moving the Bootstrap Program to the 40's the processor branches to MA 040 to begin the Bootstrap Program. This program will then load the detail program in the proper memory locations as did the punched card Bootstrap Program. The Bootstrap may be used to load any detail object program.

The last detail program ledger must contain a minus MA number in one of the even memory locations 20-38. The next higher odd memory location must contain the following commands:

An odd MA
21-39
1 - 3849 - LCR-049
2 - 0800 - PAC-00P
3 - 8020 - BRU-020

The minus MA number will cause the processor to branch out of the Bootstrap Program and to execute the above three commands. These instructions will read the last card, the proof card, and branch to the proof routine loaded by the proof ledger.

The proof card has the following format:

MEMORY LOCATION	INSTRUCTION		FUNCTION
020	3823	LCR-023	Clear control register
	0800	PAC-00P	Close carriage if necessary
	3224	TCP-024	Move proof total to work
021	2009	SPC-009	Subt. proof total developed in pro. load
	9423	BRN-023	If not equal branch to halt
	0425	TFM-025	Type program description
022	5809	CLR-009	Clear proof area
	1800	NOP	No operation
	8XXX	BRU-XXX	Branch to starting MA in program
023		BLANK	
024		CONSTANT	Proof total
025-029		CONSTANT	Program description
035-039		CONSTANT	

The format of the proof ledger is the same for all programs with the following exceptions:

A. The third instruction in memory location 022 must be supplied by the programmer and may be obtained from the symbolic print-out as a result of assembling.

B. The proof total must be supplied by the programmer.

C. The program description must be supplied by the programmer.

The proof routine will compare the proof total read from the proof ledger with the proof accumulation of the object program ledgers. If they agree, the system will type from memory at scale position 49 after which it will come to rest at position 60 where variable constants such as a date, etc., may be loaded. If the proof numbers do not agree, no typing will occur. The operator will use K2 returning to scale position 13 where she will abort the program and reload.

It should be understood that the striped ledger bootstrap uses the 020, 030 and 040 decades. Therefore, when writing a symbolic program that will be used to create a detail program which will later be loaded from striped ledger, a printer file or card file may be declared to reserve the 40 decade.

If the program is to be loaded through the carriage, rather than automatically with the A 4004, all that need be changed is MA 037 of the Bootstrap Loader and Bootstrap Program Ledger. Change the contents of memory location 037 from 0,210,000,000.00 to 0,200,000,000.00

APPENDIX A

E 8000 INSTRUCTIONS & MACHINE LANGUAGE CODES

ARITHMETIC

INST.	MOD.	LEN.	ADDR.	MACH. LANG
* ACK			xxx	16xx/17xx
* ACP			xxx	12xx/13xx
* AKC			xxx	14xx/15xx
* APC			xxx	10xx/11xx
** CLR			xxx	58xx/59xx
** DBC			xxx	52xx/53xx
** DPC			xxx	56xx/57xx
** MCB			xxx	50xx/51xx
** MCP			xxx	54xx/55xx
* SCK			xxx	26xx/27xx
* SCP			xxx	22xx/23xx
* SKC			xxx	24xx/25xx
* SPC			xxx	20xx/21xx

CONTROL CONSOLE

INST.	MOD.	LEN.	ADDR.	MACH. LANG.
** HLT			xxx	00xx/01xx
** PAC			xxx	08xx/09xx
TTM			xxx	02xx/03xx
TFM			xxx	04xx/05xx

DATA MOVEMENT

INST.	MOD.	LEN.	ADDR.	MACH. LANG.
** SLT	S	xx	00B	72wz-w=4-7, z=1-9
** SLT	D	xx	00B	72wz-w=0-3, z=1-9
** SLT	S	xx	00P	76wz-w=4-7, z=1-9
** SLT	D	xx	00P	76wz-w=0-3, z=1-9
** SRR	S	xx	00B	71wz-w=4-7, z=1-9
** SRR	D	xx	00B	71wz-w=0-3, z=1-9
** SRR	S	xx	00P	75wz-w=4-7, z=1-9
** SRR	D	xx	00P	75xz-w=0-3, z=1-9
** SRT	S	xx	00B	70wz-w=4-7, z=1-9
** SRT	D	xx	00B	70wz-w=0-3, z=1-9
** SRT	S	xx	00P	74wz-w=4-7, z=1-9
** SRT	D	xx	00P	74wz-w=0-3, z=1-9

SINGLE			DOUBLE		
LENGTH	W	Z	LENGTH	W	Z
1-9	4	1-9	1-9	0	1-9
10	5	9	10	1	9
11	6	9	11	2	9
12	7	9	12	3	9

INST.	MOD.	LEN.	ADDR.	[1] MACH. LANG.
* TCP			xxx	32xx/33xx
* TPC			xxx	30xx/31xx
TCP	R		xxx	46xx/47xx
TPC	R		xxx	44xx/45xx
TCP	S		xxx	42xx/43xx
TPC	S		xxx	40xx/41xx
* TCK			xxx	36xx/37xx
* TKC			xxx	34xx/35xx

DECISIONS

INST.	MOD.	LEN.	ADDR.	[2] pp	MACH. LANG.
BRL			xxx	Y	9Cxx-C=8-9
BRM			xxx	Y	9Cxx-C=0-3
BRN			xxx	Y	9Cxx-C=4-7
BRU			xxx	Y	8Cxx-C=0-3
JMP			xxx	Y	8Cxx-C=4-7
RTN					8800
SFL	TS				18TS-T=0-7, S=0-9
RFL	TR				19TR-T=0-7, R=0-9
SSP			000		1000
SSK			011		1411

PUNCH

INST.	MOD	LEN	ADDR.	MACH. LANG.	
ALT				779X 4-7	Drum 2, Row 4-9
					Drum 1, Row 6-9
DUP				7782	
NOR				779X 0-7	Drum 2, Row 12-3
					Drum 1, Row 6-9
PFM	A		XXX	06XX/07XX	
PFM	S	XX	00P	77wz	w=4-7, z=1-9
PFM	D	XX	00P	77wz	w=0-3, z=1-9
REJ				7784	
REL				7788	
SKP				7781	

OTHER

INST.	MOD.	LEN.	ADDR.	MACH. LANG.
CRD			xxx	48xx
LCR			xxx	38xx
NOP				1800
PRT			xxx	28xx
TIX				4000
TOF				2800
TON				2802

1 - Symbolic Address
2 - Symbolic Addresses Shown
(xxx) May be Replaced by Program Points

(*) Sets Status
(**) Garbles Status

APPENDIX B

E 8000 ASSEMBLER ERROR CODES

DIAGNOSTIC FACILITIES

Provisions are made for the detection and printed indication of certain types of errors that may occur in the Specification Card and in Phase I and Phase II of the Assembler program.

SPECIFICATION CARD ERRORS

The following errors may occur in the Specification Card. When detected, further loading of the Phase I program is prevented, the message "SPEC CARD ERROR–RESTART" is typed out and the contents of card columns 7-18 of the Specification Card are printed out on the Console Printer.

1. Number of words specified but device not specified
2. Device specified but number of words not specified
3. Device specified with more than ten words
4. Specification Card missing from deck

PHASE I ERRORS

The following errors may occur in the Phase I program. Each error is assigned a number. When an error is detected its corresponding number and the sequence number of the card on which the error occurred is printed from the Control Console. The error number prints from the corresponding digit position in memory. For example a type 4 error prints a 40.00

ERROR NO.	ERROR
1	No file declaration on Specification Card
2	Overlay address not defined
3	Invalid Control Register Constant
4	Number of program points exceeds 99
5	Number of labels exceeds 100

More than one invalid Control Register Constant (error number 3) is possible for a symbolic entry; they are cumulative.

For example if four such errors occur for one entry, the error message would print out as 12.00 (3+3+3+3 in col. 3)

At the conclusion of the Phase I program, if any errors have been detected, the Assembler executes a HLT 99 and loading of Phase II is prevented.

PHASE II ERRORS

Phase II errors may be either a multiple defined label from Phase I which is detected during the initial part of the Phase II Assembler Program, or an error which occurs during the assembly of symbolic entry cards.

MULTIPLE DEFINED LABELS

When a multiple defined label error is detected, the message "MULT DEF XXX" (XXX = 3 character label) is printed on the Line Printer immediately below the listing of the constants. A message will be printed for each multiple definition detected.

If any multiple definitions are detected, the Assembler executes a HLT 99 and the loading of the remaining Phase II program is prevented.

PHASE II SYMBOLIC ENTRY ERRORS

The following Phase II Symbolic Entry errors are detected. Each error is assigned a reference number which is printed out on the Line Printer, between the object program and symbolic entry to which it pertains. The error number prints from the corresponding memory digit position. An "*" prints in all digit positions of the error word to the left of the digit position containing an error number.

ERROR NO.	ERROR
1	Sequence number of 00 or not in ascending sequence.
2	Invalid operation code
3	Program Point undefined
4	Invalid BRL (to greater than a 200 address)
5	Program exceeds 400 words
6	Program Point greater than 9
7	Label undefined
8	Shift greater than 12 digits
9	Address greater than 200, Non-branch instruction

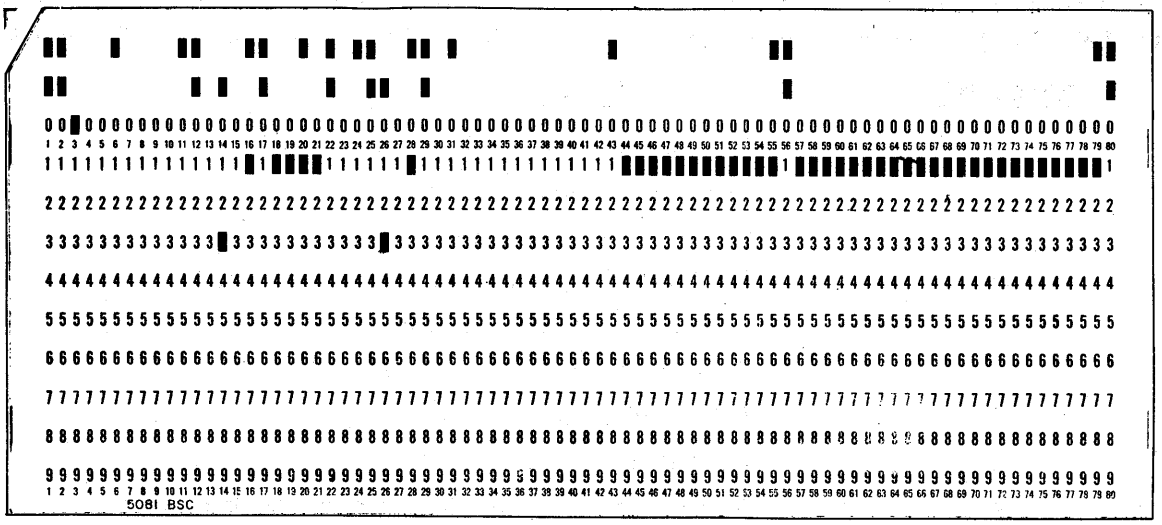
In the event NOP's are generated by the Assembler when the program is over 400 (error 5), the error is cumulative. For example if two NOP's are generated with an address over 400, it would print out the error message 100000 (5+5 in position 5) with the next symbolic entry.

No special halt (HLT) is provided in Phase II; the Assembler always executes a HLT 00 at address 011.

APPENDIX C

DRUM CARD A 149/A 150 KEYPUNCH

The card illustrated below is the drum card to be used in punching the symbolic program cards:



SYMBOLIC CARD FORMAT

The format of the Symbolic Program Cards is as follows:

CARD COLUMNS	DESCRIPTION
3-6	Identification
7-11	Sequence Number
13	Program Point
14-16	Label
18-20	OP Code
21	Modifier
23-24	Length
26-28	Address
30-31	Program Point or Word Increment
32-43	Numeric Constants
44-55	Page Header, Alpha Constants or Control Register Constants
57-79	Remarks

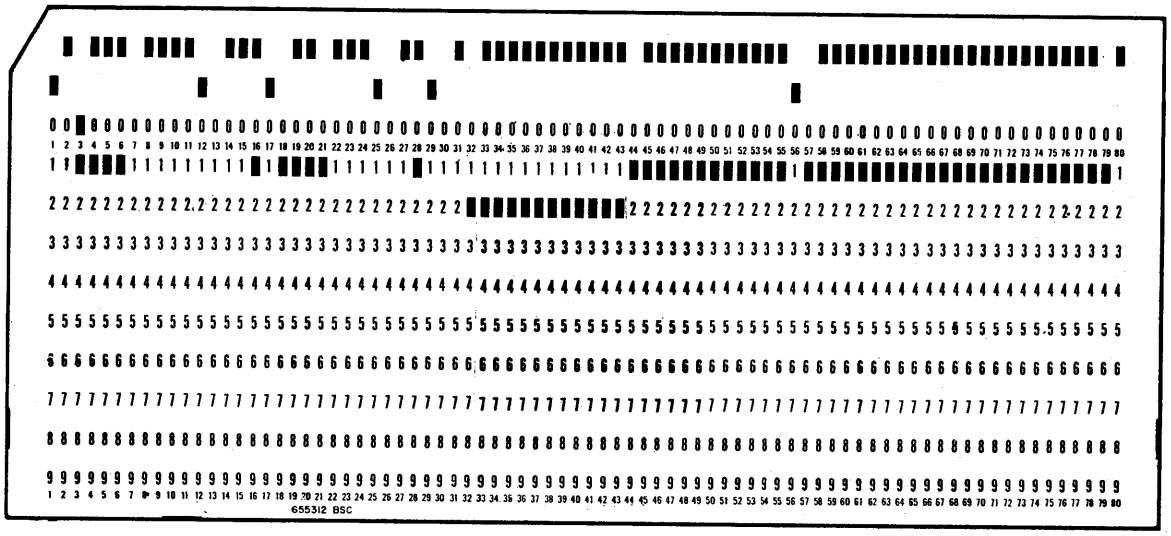
KEYPUNCHING INSTRUCTIONS

1. Turn punch on.
2. Turn on: Auto Feed, Printer.
3. Place Program Drum Card on Program Drum 1.
4. Set Program Drum Selection to 0.
5. Space to card column 3.
6. Punch Identification in card column 3-6.
7. Set Program Drum Selection to 1.
8. Punch Sequence Number in cc 7-11, skip to cc 13.
9. In cc 13, punch Program Point. If no program point, space to cc 14. Skip key to cc 18.
10. Punch Label in cc 14-16. Drum Card is set for Alpha Shift in cc 16. Skip key to cc 18.
11. Punch OP code in cc 18-20. Drum Card is set for Alpha Shift. If no OP code, Skip key to cc 21.
12. Punch Modifier in cc 21. Alpha Shift is on. If no Modifier, Skip key to cc 26.
13. Punch Address in cc 26-28, skips to cc 30. Alpha Shift is on at cc 28. If no Address, Skip key to cc 30.
14. Punch Program Point or Word Increment in cc 30-31, skip to 32. If no punching, Skip key to cc 32.
15. Punch numeric Constants in cc 32-43. If no punching, Skip key to cc 44.
16. Punch Page Header, Alpha Constant or Control Register Constant in cc 44-55. If no punching, Skip key to cc 57.
17. Punch Remarks in cc 57-79. If no Remarks, Skip key or Release will release the card.

APPENDIX C

DRUM CARD 026 KEYPUNCH

The card illustrated below is the drum card to be used in punching the symbolic program cards:



SYMBOLIC CARD FORMAT

The format of the Symbolic Program Cards is as follows:

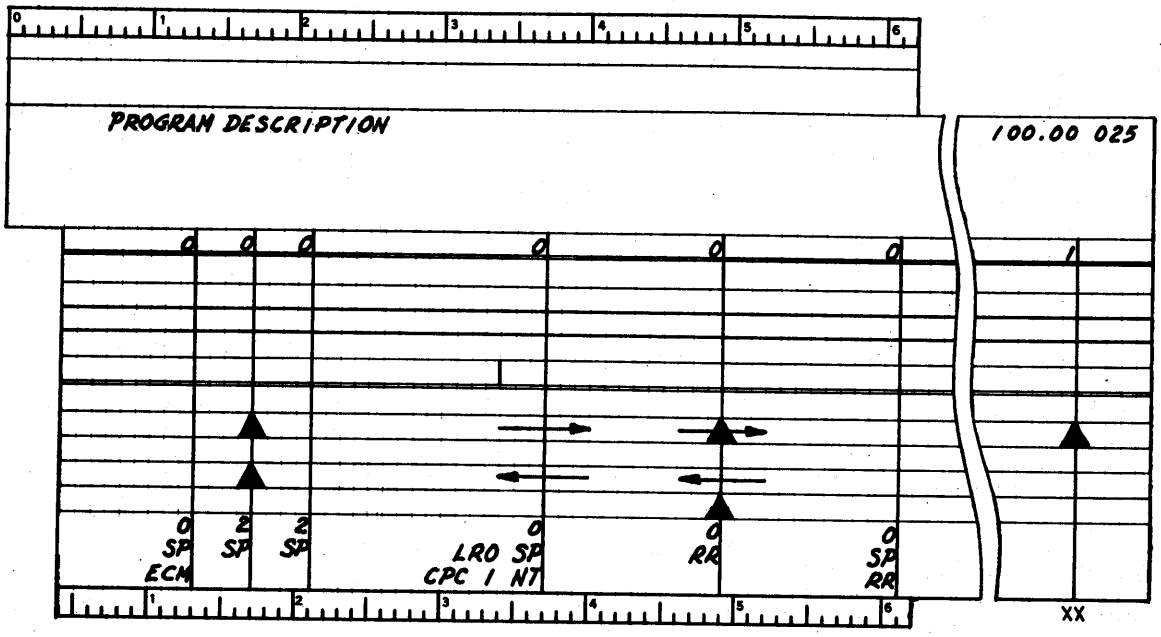
CARD COLUMNS	DESCRIPTION
3-6	Identification
7-11	Sequence Number
13	Program Point
14-16	Label
18-20	OP Code
21	Modifier
23-24	Length
26-28	Address
30-31	Program Point or Word Increment
32-43	Numeric Constants
44-55	Page Header, Alpha Constants or Control Register Constants
57-79	Remarks

KEYPUNCHING INSTRUCTIONS

1. The Print Switch is set at ON. All other switches are set at OFF.
2. Feed the first card and punch the Identification Number in cc 3-6. Move the Auto Skip, Auto-Dup Switch to ON.
3. Punch Sequence Number in cc 7-11–skips to cc 13.
4. In cc 13 punch Program Point. If no Program Point, use Space Bar to cc 14. Skip key to cc 18.
5. Punch Label in cc 14-16. Drum Card is punched for Alpha Shift in cc 16. Skips to cc 18.
6. Punch OP Code in cc 18-20. Drum Card is punched for Alpha Shift. If no OP Code, Skip Key to cc 21.
7. Punch Modifier in cc 21. Alpha Shift is on. If no Modifier, Skip Key to cc 26.
8. Punch Address in cc 26-28–auto skip to cc 30. Alpha Shift is on at cc 28. If no Address, Skip Key to cc 30.
9. Punch Program Point or Word Increment in cc 30-31–auto skip to cc 32. If no punching, Skip Key to cc 32.
10. Punch Numeric Constants in cc 32-43. If no punching, Skip Key to cc 44.
11. Punch Page Header, Alpha Constant or Control Register Constant in cc 44-55. If no punching, Skip Key to cc 57. Alpha Shift is on in cc 44-55.
12. Punch Remarks in cc 57-79. Drum Card is punched for Alpha Shift.
13. REL, FEED and REG Keys to next card.

APPENDIX D

INSTRUCTIONS FOR E 6000 ASSEMBLER

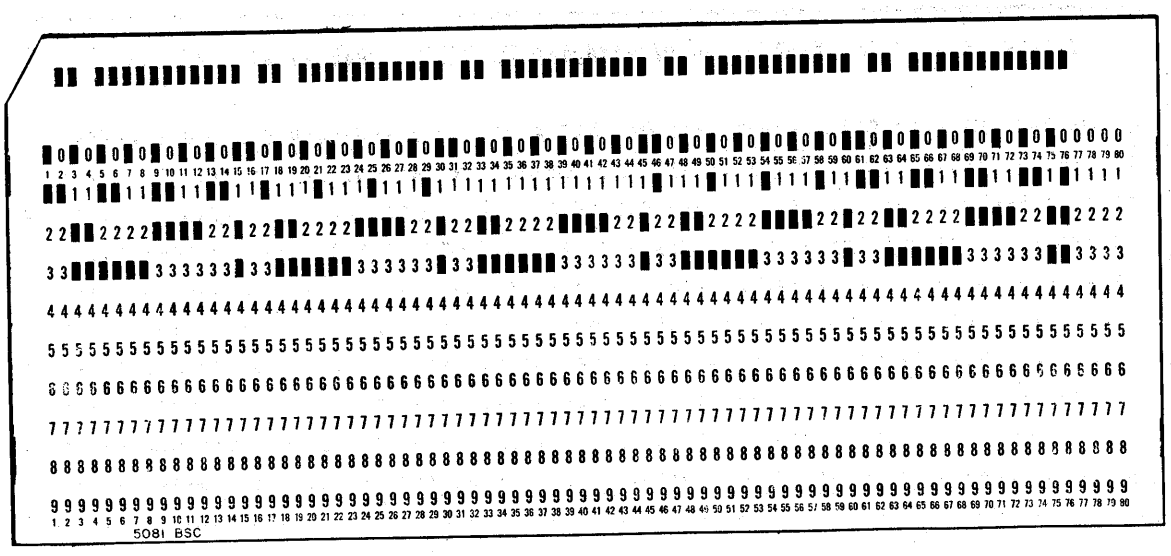


Note: Change to CPC 0 for A 505 and A 149.

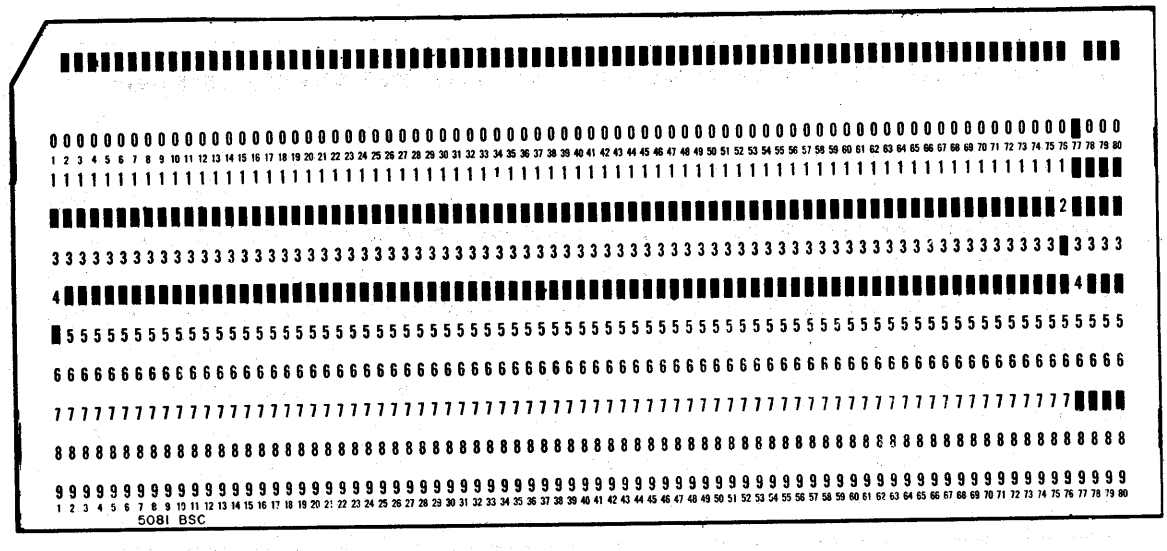
OPERATOR INSTRUCTIONS

Assuming that the System is in a "ready" condition, as described in Section 5, with Console Printer, Card Punch, Card Punch Control Unit and Line Printer properly programed, proceed with the following:

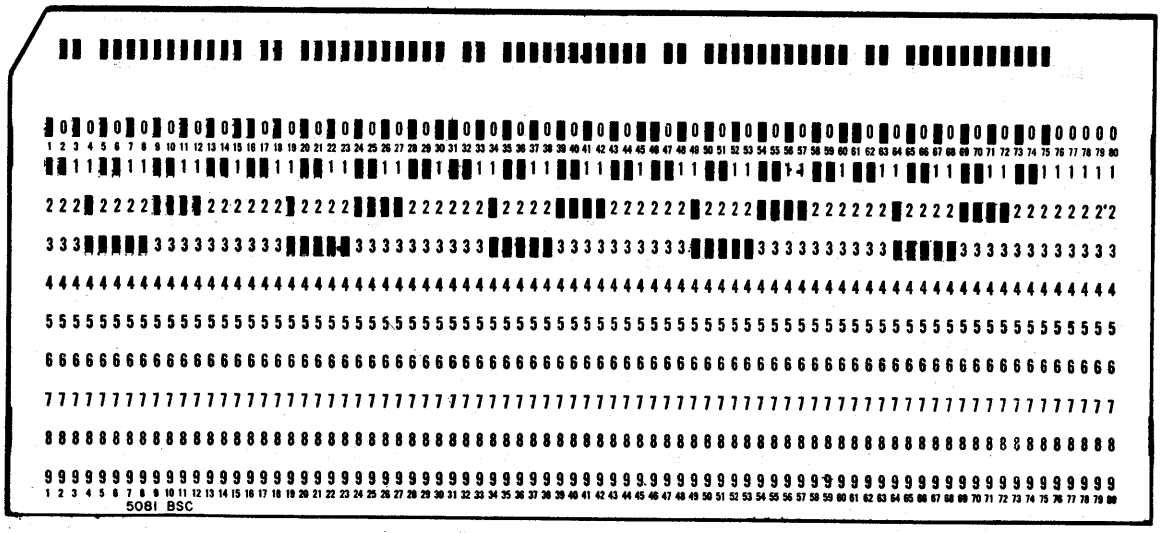
1. Load the A 594 Card Reader with Phase I input cards arranged in the sequence described in Section 5.
2. Use K4 to scale 17 the starting position for the Assembler Program.
3. Depress PGL switch/light and Motor Bar 2. The Control Console cycles and tabulates to the next stop position at scale 21.
4. Depress the STR switch/light and Motor Bar 2.
5. At the completion of the Phase I operations, turn off the Card Reader, remove Phase I cards and place the Symbolic Deck with the Phase II cards which are to be arranged as described in Section 5. Restart the Card Reader.



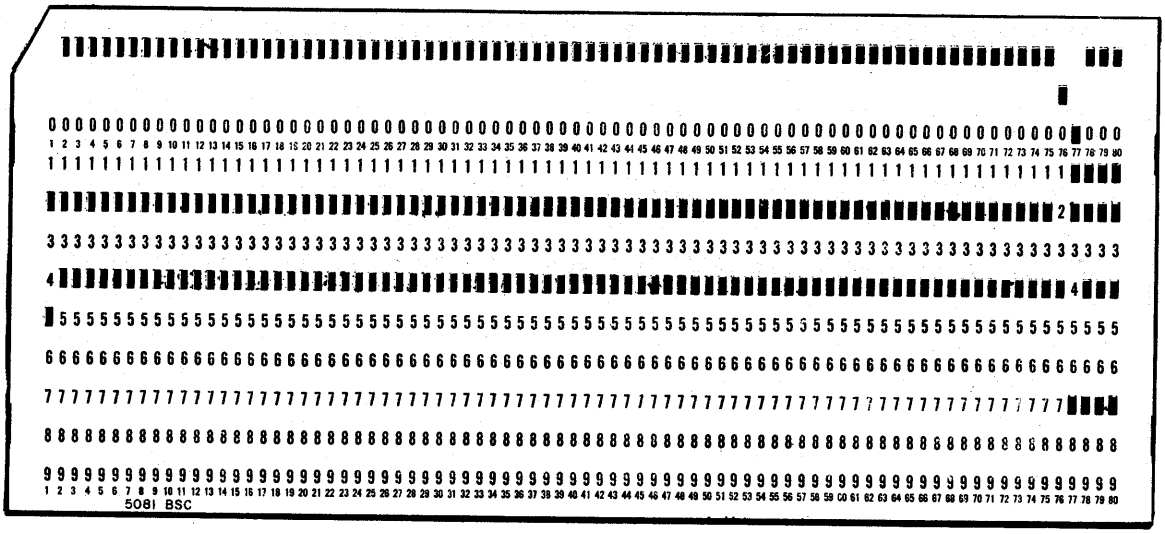
REAR DRUM CARD - A 592



FRONT DRUM CARD - A 592

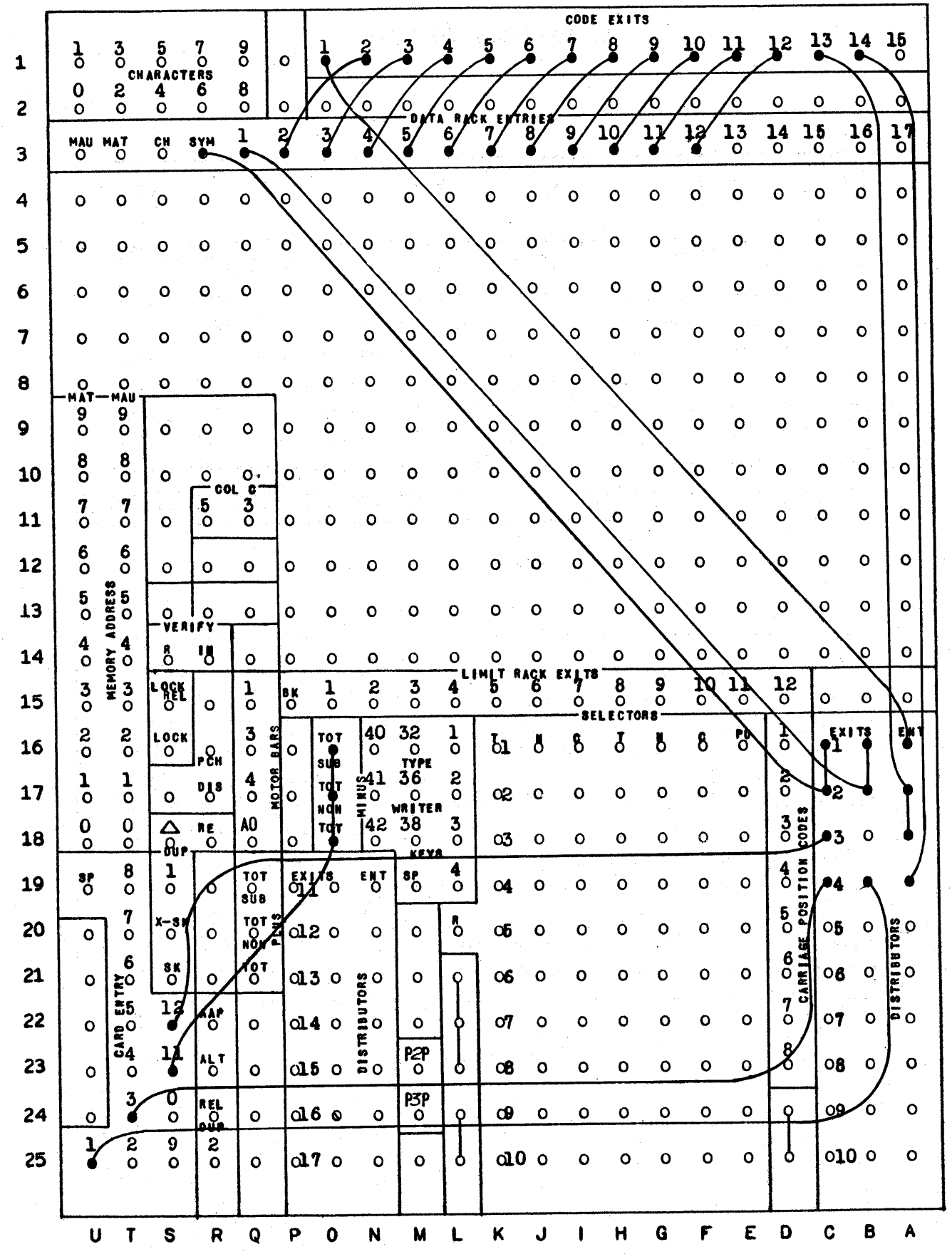


REAR DRUM CARD - A 594

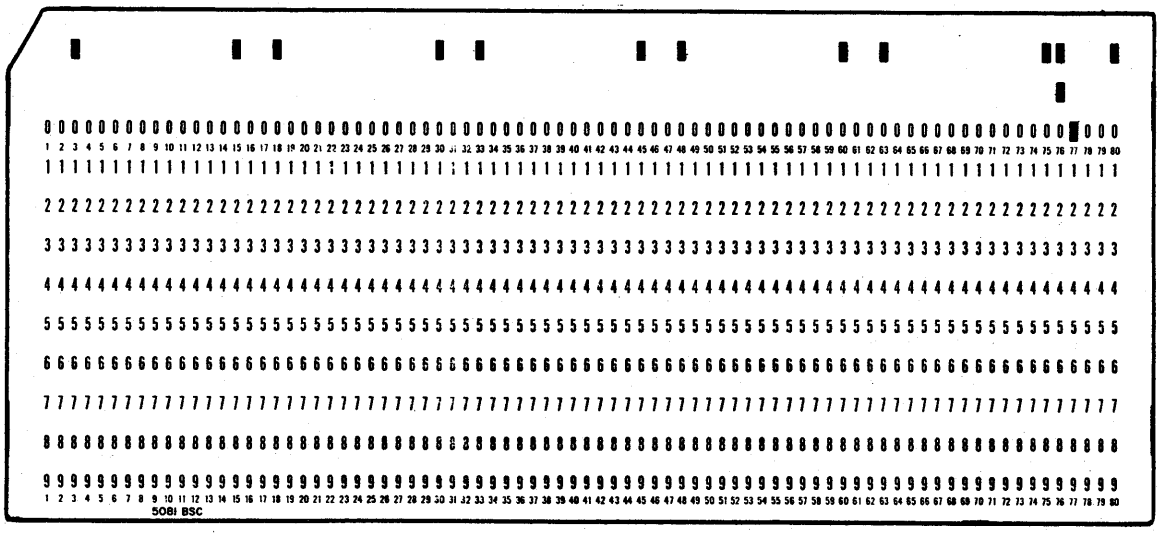


FRONT DRUM CARD - A 594

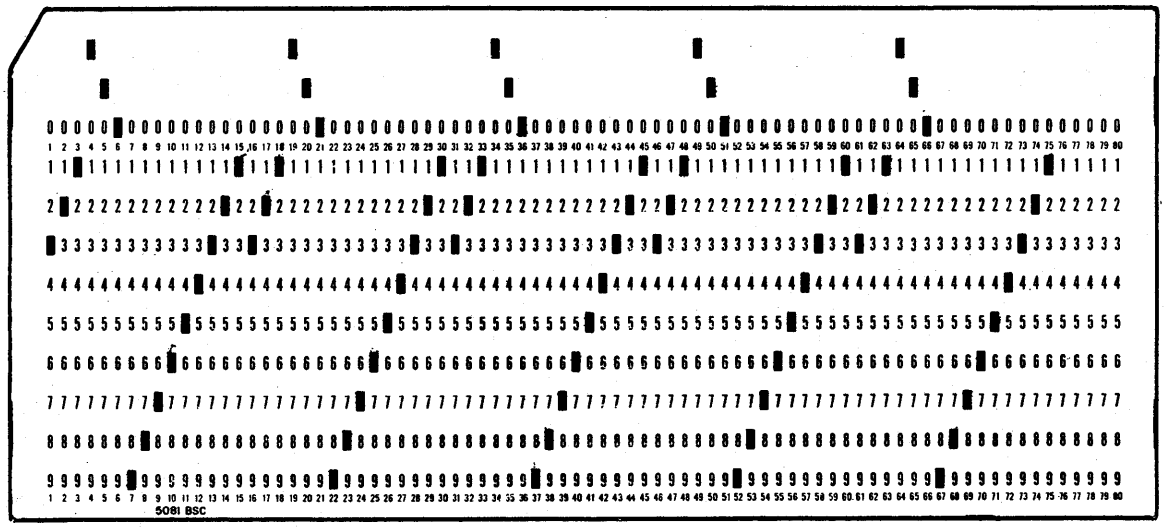
MACHINE STYLE E 6000 DATE PREPARED _____ BRANCH _____
CUSTOMER E 6000 ASSEMBLER BY HOME OFFICE INVOICE _____
F O NO _____



DRUM CARDS FOR A 149

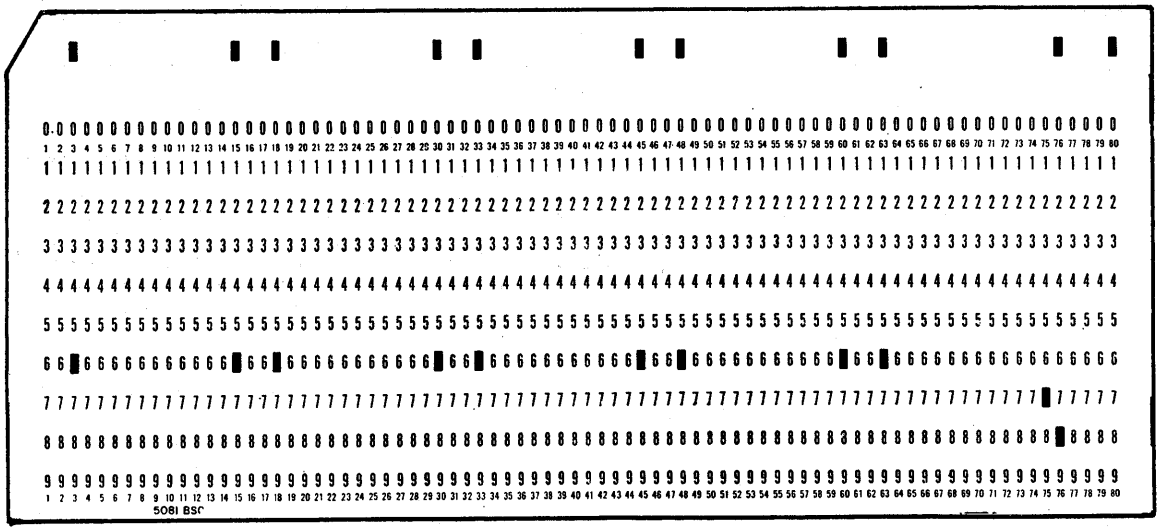


A 594 DRUM 1

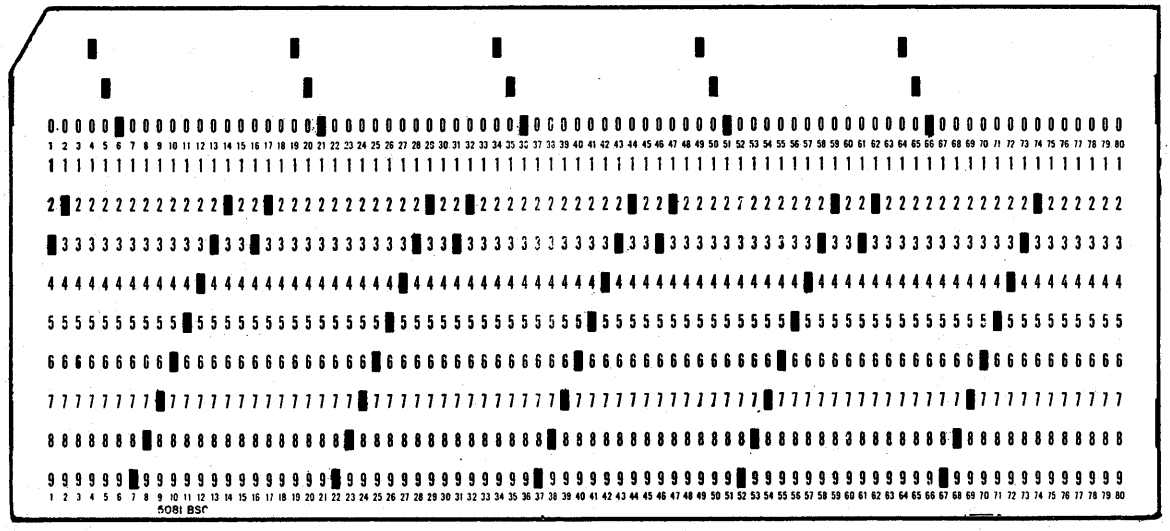


A 594 DRUM 2

DRUM CARDS FOR A 149



A 592 DRUM 1



A 592 DRUM 2

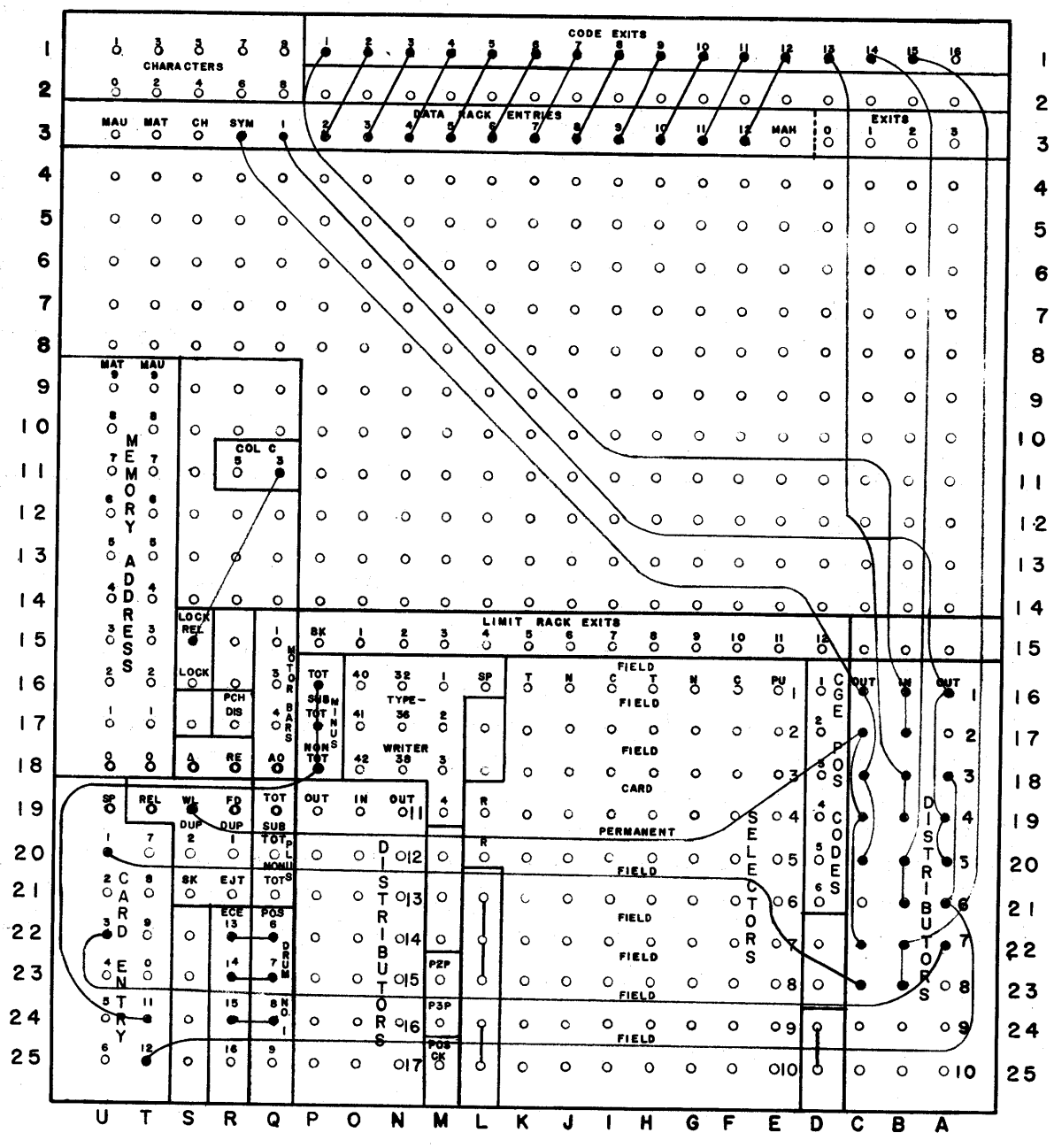
WIRING DIAGRAM FOR PUNCH CARD

STYLE A 505

MACHINE STYLE E 6000 PLUGBOARD NO. _____ BRANCH _____

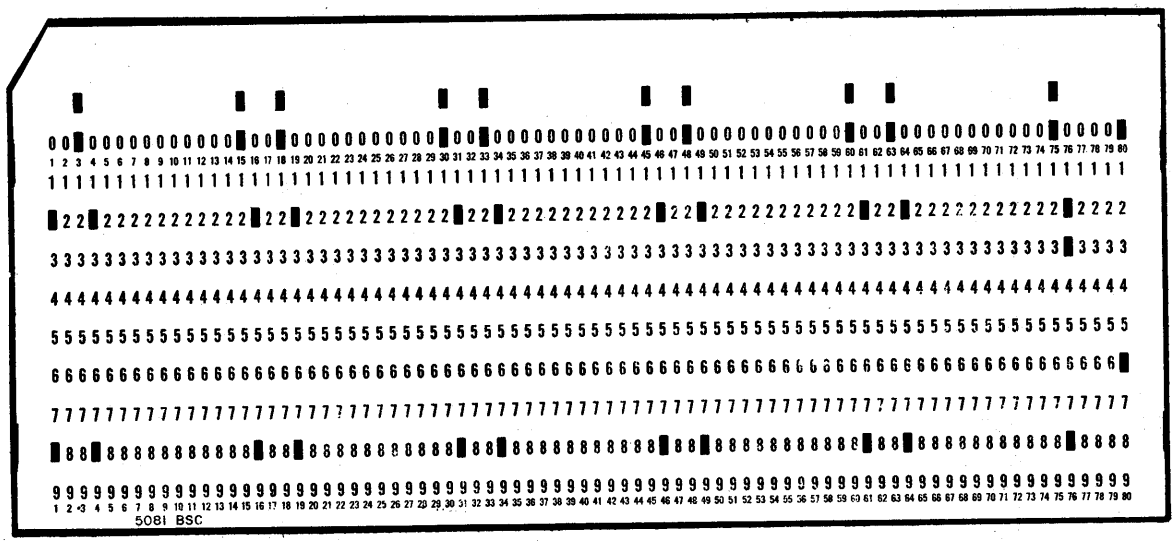
CUSTOMER H. O. DATE PREPARED 9/69 INVOICE NO. _____

_____ BY J. W. F.O. NO. _____

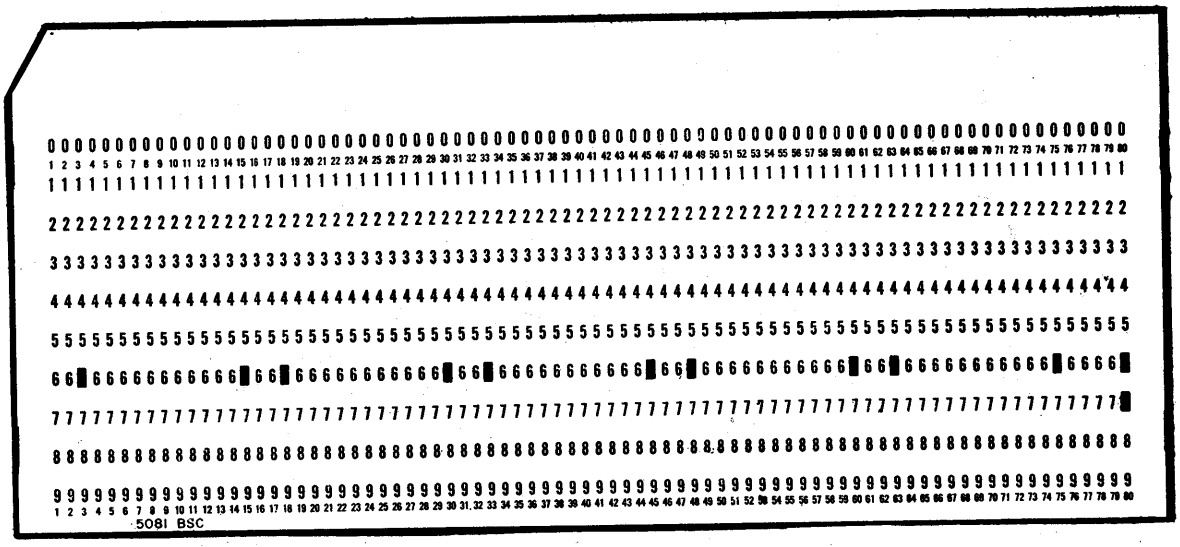


FORM MKTG-2389 (3/69)

DRUM CARDS FOR A 149
E 8000 With Punch From Memory



DRUM 1



DRUM 2

APPENDIX E

E 8000 CENTRAL PROCESSOR INSTRUCTIONS

	INST.	MOD.	LEN.	ADDR.[1]	
ARITHMETIC	ACK			xxx	Add C to K
	ACP			xxx	Add C to P
	AKC			xxx	Add K to C
	APC			xxx	Add P to C
	CLR			xxx	Clear C
	DBC			xxx	Divide B by C
	DPC			xxx	Divide P by C
	MCB			xxx	Multiply C by B
	MCP			xxx	Multiply C by P
	SCK			xxx	Subtract C from K
	SCP			xxx	Subtract C from P
	SKC			xxx	Subtract K from C
	SPC			xxx	Subtract P from C
CONTROL CONSOLE	HLT			xxx	Halt
	PAC			xxx	Print and cycle
	TTM			xxx	Type to memory
	TFM			xxx	Type from memory
DATA MOVEMENT	SLT	S	xx	00B	Shift left B
	SLT	D	xx	00B	Shift left B to K
	SLT	S	xx	00P	Shift left P
	SLT	D	xx	00P	Shift left P to K
	SRR	S	xx	00B	Shift B right and round
	SRR	D	xx	00B	Shift right K to B and round
	SRR	S	xx	00P	Shift P right and round
	SRR	D	xx	00P	Shift right K to P and round
	SRT	S	xx	00B	Shift right B
	SRT	D	xx	00B	Shift right K to B
	SRT	S	xx	00P	Shift right P
	SRT	D	xx	00P	Shift right K to P
	TCP	(S/R)*		xxx	Transfer C to P
	TPC	(S/R)*		xxx	Transfer P to C
		*Use S in the modifier column to program SPECIFIED COLUMNS or R to program REVERSE.			
	TCK			xxx	Transfer C to K
	TKC			xxx	Transfer K to C

E 8000 CENTRAL PROCESSOR INSTRUCTIONS (CONT'D.)

	INST.	MOD.	LEN.	ADDR.[1]	PP[2]	
DECISIONS	BRL			xxx	y	Branch on LSD non-zero
	BRM			xxx	y	Branch on minus
	BRN			xxx	y	Branch on non-zero
	BRU			xxx	y	Branch unconditional
	JMP			xxx	y	Subroutine jump
	RTN					Subroutine return
	SFL	ts				Test and set flag
	RFL	tr				Test and reset flag
	SSP					Set status on word 00P
	SSK					Set status on word 00K
PUNCH	ALT		xx			Select and set alternate area of drum 1 and end of field in drum 2
	DUP					Duplicate card columns
	NOR		xx			Select and set normal area of drum 1 and end of field in drum 2
	PFM	A		xxx		Punch alpha characters from memory
	PFM	D	xx	00P		Shift left P to K and punch contents of P
	PFM	S	xx	00P		Shift left P and punch contents of P
	REJ					Release card to auxiliary stacker
	REJ					Release card to regular stacker
	SKP					Skip card columns until terminated by end of field code
OTHER	CRD			xxx		Read punched card
	LCR			xxx		Load control register
	NOP					No operation
	PRT			xxx		Print on line printer
	TIX					Turn on indexing
	TOF					Turn off 200
	TON					Turn on 200

[1] Symbolic Address

[2] Symbolic Addresses shown (xxx) may be replaced by program points (y).

APPENDIX F

CONTROL REGISTER FORMAT

The following chart illustrates the controls that can be loaded in the control register. These controls each have a binary value and that value is expressed numerically in the assigned MA. Since more than one control is in the same digit position, the numeric value would be expressed as the total of the binary values. For example, if the operation should require the following instructions, the selected MA would contain the values shown:

WRL, LN4, RDC, APT and P39

010000010279

This is determined as follows:

WRL has a value of	010000000000
LN4 has a value of	000000000040
RDC has a value of	000000010000
APT has a value of	000000000200
P39 has a value of	000000000039
Sum to be loaded in Control Register	010000010279

BINARY													
	8				BCC		DL8		ZNE				PO8
	4		SMB	BTF	BLC		DL5	LN5# (DS)	HME	RKC	DAP%	LN4@ (DSR)	PO4
	2		FDL	BAD	ALN			REV		RKA VC2	APT%	P20	PO2
	1		WRL*	RDL	VER		AMB	CLA	RDC	EKA VC1	APS%	P10	PO1
Dig. Pos.		12	11	10	9	8	7	6	5	4	3	2	1

WRC = BLC & ALN.

* Occurs at next control console position prior to machine cycle - carriage open must be pinned and MA address will usually contain unit digit of 9 when WRL is in the Control Register.

@ LN4 with motor bar 1 = DSR will skip to lane 1; with flag 8 also set, will skip in lane 3. Lane 3 movement can occur only from MB 1, activated internally or manually with AMB programed.

\# LN5 with motor bar 1 = DS - will return in lane 2 and disable space. Will index lane 5 on basic machine operation.

% The Subtotal or Total Symbol and/or the Dollar sign may be printed with keyboard entered amounts.

NOTE: 1. With RDL and FDL in the Control Register, only the A 4004 will cycle, the console printer will not cycle. With RDL, FDL and BLC in the Control Register a printer cycle will occur, if programed when blank ledger is used.

NOTE: 2. Contents of MA 20 can be printed by programing RDC with RDL & ALN in the Control Register. MAR will be set at 20 or 30 after this operation.

APPENDIX G

E 8000 CONTROL REGISTER CONSTANTS

ALN	ALIGN STRIPED LEDGER (USED WITH RDL TO READ AND ALIGN)
AMB	ACTIVATE MOTOR BAR BY THE FLAG SETTING
APS	PRINT THE SUBTOTAL SYMBOL
APT	PRINT TOTAL SYMBOL
BAD	BLOCK ADD STRIPED LEDGER DATA TO MA'S 60-79
BCC	ENTER STRIPED LEDGER AND CLEAR STRIPED LEDGER AREA
BLC	ENTER BLANK STRIPED LEDGER
BTF	CLEAR AND ADD MA'S 60-79
CLA	CLEAR THE "C" ADDRESS BEFORE THE PROGRAMED RKC
DAP	DOLLAR AMOUNT PROTECTION
DL5	LIGHT DECIMAL LIGHT 5
DL8	LIGHT DECIMAL LIGHT 8
EKA	ENFORCE THE USE OF THE MEMORY ADDRESS KEYS
FDL	FEED LEDGER VIA AUTO READER
HME	HOME TRACTOR
LN4	INDEX LANE 4 (DSR WITH BAR 1)
LN5	INDEX LANE 5 (DS WITH BAR 1)
PXX	POSITION CHECK XX AGAINST CONTROL UNIT
RDC	READ THE "C" ADDRESS
RDL	READ STRIPED LEDGER DATA (READS DATA ONLY)
REV	REVERSE NORMAL ARITHMETIC
RKA	READ THE MEMORY ADDRESS SELECTED ON THE KEYBOARD
RKC	READ KEYBOARD AND ADD "C"
SMB	SET FLAGS VIA MOTOR BAR USED
VC1	VARIABLE CHANNEL
VC2	BOTTOM OF FORM
VC3	TOP OF FORM
VC4	SINGLE LINE (NO CODING REQUIRED)
VER	VERIFY KEYBOARD WITH MEMORY ADDRESS 20
WRC	WRITE CORRECTION
WRL	WRITE STRIPED LEDGER
ZNE	ZONE TRACTOR

APPENDIX H

E 8000 MASK VALUES

(Starting with A 988 No. A 720058 use these values)

PRINTING COMBINATION DESIRED:	TO PRINT	USE MASK VALUE	MASK VALUE WILL ALLOW PRINT AS SHOWN AND:
ALPHA			
ALPHA DIGITS	X		
KEY CODE OR SPACE CODES	bbb	-1	INSERT BLANKS AFTER KEY CODE TO END OF ALPHA. RESET SIGNIFICANCE
BLANK IF:			
REPLACES DIGIT	b	-6	RESET SIGNIFICANCE
COMMA AND DIGIT IF:			
SIGNIFICANT DIGIT	,d		
NOT SIGNIFICANT DIGIT	,0	+2	RESET SIGNIFICANCE
DECIMAL AND DIGIT IF:			
SIGNIFICANT DIGIT	.d		
NOT SIGNIFICANT DIGIT	.0	-2	RESET SIGNIFICANCE
DIGIT AND DECIMAL IF:			
SIGNIFICANT DIGIT	d.		
NOT SIGNIFICANT DIGIT	bb	+3	
DIGIT IF:			
SIGNIFICANT DIGIT	d		
NOT SIGNIFICANT DIGIT	0	±7	RESET SIGNIFICANCE
DIGIT IF:			
SIGNIFICANT DIGIT	d		
NOT SIGNIFICANT DIGIT	b	±0	
DIGIT IF:			
SIGNIFICANT DIGIT	d		
NOT SIGNIFICANT DIGIT	*	+5	
DIGIT AND COMMA IF:			
SIGNIFICANT DIGIT	d,		
NOT SIGNIFICANT DIGIT	bb	-3	
DIGIT AND BLANK IF:			
SIGNIFICANT DIGIT	db		
NOT SIGNIFICANT DIGIT	bb	-5	
DIGIT AND SIGN IF:			
PLUS	0b		
MINUS	0-	±8	RESET SIGNIFICANCE
DIGIT AND SYMBOL IF:			
PLUS	0*		
MINUS	0CR	±9	RESET SIGNIFICANCE
DOLLAR SIGN IF:			
SIGNIFICANT DIGIT	d	+6	
NOT SIGNIFICANT DIGIT	$		
END OF LINE	b	+1	RESET SIGNIFICANCE
IGNORE DIGIT	IGNORED	±4	RESET SIGNIFICANCE

b INDICATES PRINTING A BLANK WHICH TAKES 1/10 INCH OF PRINTING SPACE.

d AS SHOWN IN ALL EXAMPLES INDICATES PRESENCE OF A SIGNIFICANT DIGIT IN PRINTING LOCATION OF MEMORY.

0 AS SHOWN IN ALL EXAMPLES INDICATES PRESENCE OF NON-SIGNIFICANT ZERO IN PRINTING LOCATION OF MEMORY.

APPENDIX I

E 8000 SUMMARY OF CARRIAGE MOVEMENT CONSIDERATIONS

MOVEMENT DESIRED:	MOTOR A*	BAR B**	LANE	PINS NEEDED	RELEASE REQUIRED	FLAGS SET	CONTROL REGISTER
NON-TAB	2	2	- -	NT	NO		
NON-TAB AND SPACE	2	2	- -	NT, SP	NO		
	2	3	- -		NO	2	AMB
	3		- -		NO		
TO RIGHT							
To Adjacent Stop	2	2	BT #		NO		
To Selected Stop	1		1	NONE	YES		LN4(DSR)
	1		3	NONE	YES	8	LN4(DSR), AMB
	2	2	4	NONE	YES		LN4
	2	1	3	NONE	YES	1 & 8	LN4(DSR), AMB
	2	1	1	NONE	YES	1	LN4(DSR), AMB
	2	4	1	NONE	YES	4	AMB
	4		1	NONE	YES		
TO RIGHT AND SPACE							
To Adjacent Stop	2	2	BT	SP	NO		
To Selected Stop	2	2	4	SP	YES		LN4
TO LEFT							
To Adjacent Stop	1		RM%	NONE	NO	8	LN5(DS), AMB
	2	1	RM	NONE	NO	1 & 8	LN5(DS), AMB
To Selected Stop	1		2	NONE	YES		LN5(DS)
	2	1	2		YES	1	LN5(DS), AMB
	2	2	5		YES		LN5
TO LEFT AND SPACE							
To Adjacent Stop	1		RM	NONE	NO	8	AMB
	2	1	RM	NONE	NO	1 & 8	AMB
To Selected Stop	1		2	NONE	YES		
	2	1	2	NONE	YES	1	AMB
	2	2	5	SP	YES		LN5

TYPE FROM MEMORY MOVEMENT

TO RIGHT - LANE 27
TO LEFT - LANE 28, 29, 30

*A - USE THIS COLUMN FOR OPERATOR ACTION OR REPEAT OF MOTOR BAR
**B - USE THIS COLUMN FOR AUTOMATIC CYCLES
- BT Basic Movement
% - RM Reverse Movement

APPENDIX J

E 8000 TIMINGS

	MNEMONIC CODE	EXECUTION TIME
ARITHMETIC	ACK xxx	1.60 ms maximum
	ACP xxx	1.60 ms maximum
	AKC xxx	1.60 ms maximum
	APC xxx	1.60 ms maximum
	CLR xxx	.97 ms
	DBC xxx	16.11 ms + 6.6 ms per dividend digit
	DPC xxx	16.11 ms + 6.6 ms per dividend digit
	MCB xxx	12.0 ms + 5.7 ms per multiplier digit
	MCP xxx	12.0 ms + 5.7 ms per multiplier digit
	SCK xxx	1.60 ms maximum
	SCP xxx	1.60 ms maximum
	SKC xxx	1.60 ms maximum
	SPC xxx	1.60 ms maximum
CONTROL CONSOLE	HLT xxx	.35 ms
	PAC xxx	60.35 ms plus 145 ms for console cycle (Balance of cycle and carriage movement time for internal processing)
	TTM xxx	.35 ms
	TFM xxx	20.35 ms plus 133 ms each character
SHIFT	SLT S xx 00B	.40 ms
	SLT D xx 00B	.40 ms
	SLT S xx 00P	.40 ms
	SLT D xx 00P	.40 ms
	SRR S xx 00B	.40 ms
	SRR D xx 00B	.40 ms
	SRR S xx 00P	.40 ms
	SRR D xx 00P	.40 ms
	SRT S xx 00B	.40 ms
	SRT D xx 00B	.40 ms
	SRT S xx 00P	.40 ms
	SRT D xx 00P	.40 ms

0 place = .40 ms
1 place = 1.33 ms
2 places = 1.97 ms
3 places = 2.61 ms
4 places = 3.24 ms
5 places = 3.88 ms
6 places = 4.51 ms
7 places = 5.15 ms
8 places = 5.79 ms
9 places = 6.42 ms
10 places = 7.14 ms
11 places = 7.78 ms
12 places = 8.41 ms

TRANSFER	TCP xxx	1.60 ms maximum
	TPC xxx	1.60 ms maximum
	TCP R xxx	.972 ms maximum
	TPC R xxx	.972 ms maximum
	TCP S xxx	.972 ms maximum
	TPC S xxx	.972 ms maximum
	TCK xxx	1.60 ms maximum
	TKC xxx	1.60 ms maximum
DECISIONS	BRL xxx	.34 ms
	BRM xxx	.34 ms
	BRN xxx	.34 ms
	BRU xxx	.36 ms
	JMP xxx	.36 ms
	RTN	.36 ms
	SFL TS	.36 ms
	RFL TR	.36 ms
	SSP	1.60 ms
	SSK	1.60 ms

	MNEMONIC CODE	EXECUTION TIME
PUNCH	ALT xx	.528 ms
	DUP	.528 ms plus .40 ms for each column duplicated – this time available for internal processing
	NOR xx	.528 ms
	PFM-A xxx	.45 ms plus 40.0 ms for each card column punched. 12.5 ms for each card column skipped after punching terminates – this time is available for internal processing
	PFM S/D xx 00P	.48 ms plus .40 ms for each place shifted and 40.0 ms for each column punched
	REJ	.528 ms
	REL	.528 ms plus 12.5 ms for each column skipped during release. This time is available for internal processing.
	SKP	.528 ms plus 12.5 ms for each column skipped. This time is available for internal processing.
OTHER	CRD xxx	.60 ms
	LCR xxx	.97 ms
	NOP	.35 ms
	PRT	.70 ms plus 358 ms for line printer to print - this time is available for internal processing
	TIX	1.284 ms
	TOF	.34 ms
	TON	.34 ms

APPENDIX K

E 8000 BASIC ASSEMBLER
PSEUDO INSTRUCTIONS

ALP	LOAD ALPHA CONSTANTS IN FILE
APH	LOAD ALPHA CONSTANT IN SUCCESSIVE WORDS
ASN	ASSIGN
CST	LOAD CONSTANT
CTL	LOAD CONTROL REGISTER CONSTANTS
END	END
FIL	FILE DECLARATION
HDR	HEADER
MSK	LOAD CONSTANT IN FILE
OVR	OVERLAY
RSV	RESERVE

APPENDIX L

PUNCH CARD CODES

The internal memory codes and the required punched card codes are tabulated below:

Numeric

Character	Memory Code	Card Code
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
0	0	0

Alphanumeric

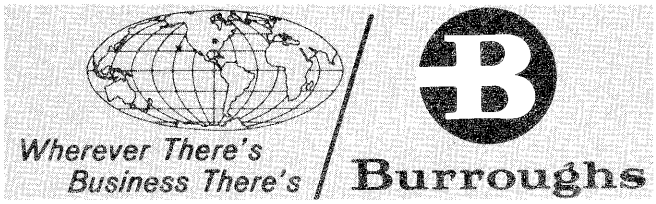
Character	Memory Code		Card Code	
	High	Low	Zone	Digit
A	2	1	12	1
B	2	2	12	2
C	6	0	12	3
D	2	4	12	4
E	2	5	12	5
F	2	6	12	6
G	2	7	12	7
H	7	2	12	8
I	7	6	12	9
J	1	1	11	1
K	1	2	11	2
L	6	4	11	3
M	1	4	11	4
N	1	5	11	5
O	1	6	11	6
P	1	7	11	7
Q	7	4	11	8
R	7	5	11	9
S	4	2	0	2
T	6	1	0	3
U	4	4	0	4
V	4	5	0	5
W	4	6	0	6
X	4	7	0	7
Y	7	0	0	8
Z	7	7	0	9
1	0	1		1
2	0	2		2
3	6	3		3
4	0	4		4
5	0	5		5
6	0	6		6
7	0	7		7
8	2	0		8
9	7	1		9
0	4	0		0
,	6	7	0	3,8

Character	Memory Code High	Memory Code Low	Card Code Zone	Card Code Digit
.	6	6	12	3,8
/	4	1	0	1
;	6	5	11	8,6
**_	1	0	11	
SPACE	8	0		BLANK
@	0	0		8,4
ESCAPE	8	8	12	8,4
Key 1, not end alpha	8	1	12-11	1
Key 2, not end alpha	8	2	12-11	2
Key 3, not end alpha	8	3	12-11	3
Key 4, not end alpha	8	4	12-11	4
Key 1, end alpha	9	1	12-11-0	1
Key 2, end alpha	9	2	12-11-0	2
Key 3, end alpha	9	3	12-11-0	3
Key 4, end alpha	9	4	12-11-0	4

E 8000 BASIC ASSEMBLER REFERENCE MANUAL

ALPHABETIC INDEX

ALPHABETIC INDEX


Wherever There's
Business There's
Burroughs