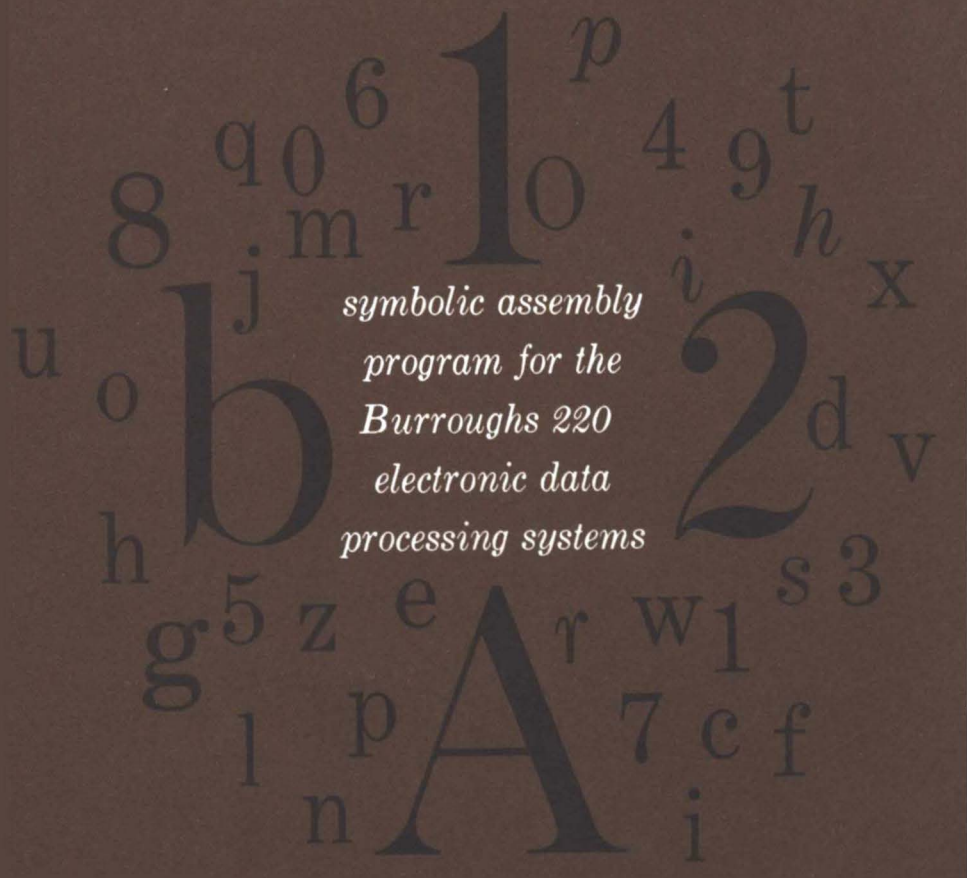
STAR II



symbolic assembly program for the Burroughs 220 electronic data processing systems

Bulletin No. 220-21012-D
Revised OCT. 1961

STAR II
SYMBOLIC ASSEMBLY PROGRAM
FOR THE
Burroughs 220
ELECTRONIC DATA-PROCESSING SYSTEMS

This publication supersedes all others
which pertain to this subject.

Equipment and Systems Marketing Division
Sales Technical Services

Burroughs Corporation
Detroit 32, Michigan

ORDERING PROCEDURE

The various routines mentioned in this manual are available upon request from your Burroughs Representative.

Each user may order the Basic Package of routines and, upon special request, the Reassembly Package of Routines.

The Basic Package includes:

The Star II Machine-Language Program Tape.* (Magnetic Tape)	6.00.072
The Macro Incorporate Routine (Paper Tape)	6.00.073
The Macro Delete Routine (Paper Tape)	6.00.074
The General Purpose Loader (Paper Tape)	6.00.075
The Program Library Creation Routine (Paper Tape)	6.00.076
The BMTR Break-In and Restart Routine (Paper Tape)	6.00.077

The Reassembly Package includes:

The Star II Symbolic Language Tape* (Magnetic Tape)	6.00.078
The Star II Blocking and BMTR Copy Routine (Paper Tape)	6.00.079
The Macro Copy Routine (Paper Tape)	6.00.080
The Macro Table Copy and Develop New Hash Total Routine (Paper Tape)	6.00.081

Also available upon request are the flow charts and listings of all the routines mentioned.

A small reel (Datafile reel) is adequate for a copy of the Star II Machine-Language Tape but a large reel of tape is required for the copy of the Star II Symbolic Language Tape.

*In order to obtain copies of the Star II Machine-Language (program) Tape or the Star II Symbolic Language Tape a reel of magnetic tape upon which the routine will be copied must accompany the request.

TABLE OF CONTENTS

APPENDIXES

FIGURES

TABLES

ACKNOWLEDGEMENTS

The Sales Technical Services Department of Burroughs Corporation, which has had the responsibility of implementing STAR II, is grateful for the assistance it has received from many interested users, notably from the Colgate-Palmolive Company, 300 Park Avenue, New York, New York; and Smith, Kline and French Laboratories, 15th and Spring Garden Streets, Philadelphia, Pennsylvania.

SECTION I

INTRODUCTION

STAR II is an automatic coding technique used to translate programs written in symbolic language into machine coding for use by the BURROUGHS 220.

AUTOMATIC CODING

Automatic coding is a phrase used to describe a number of special programs designed to aid the programer. These special programs have one thing in common: they make use of the computer to do part of the programer's work for him. Without such machine assistance, a programer must prepare a complete and detailed list of instructions ready for direct acceptance by the computer. The use of these programing aids enables the computer to take care of many details, such as translating easily remembered letter codes into digits, choosing the most suitable areas of storage and supplying required subroutines. Not every automatic coding system is capable of providing all of these aids, but some do provide them and have even greater capabilities.

According to their major characteristics, several types have been assigned names and, although there is much overlapping, each type can be fairly well described.

Compilers

The compiler is one of the most ambitious of these automatic coding routines. It offers the greatest relief to the programer from the routine details of his job by allowing him to write in a notation that more closely resembles the language in which the problem is stated.

A compiler is a routine which produces many real instructions from a single pseudo instruction. It produces a specific program for a particular problem by:

1. Determining the intended meaning of an element of information expressed in pseudo code (pseudo instructions).
2. Selecting and generating the required subroutine.
3. Transforming the subroutine into specific coding for the specific problem.
4. Assigning specific storage locations for the subroutine and then entering it as an element of the problem-solving program.
5. Maintaining a record of the subroutines used and their placement in the problem-solving program.

Interpreters

An interpretive routine differs from other automatic coding schemes in that it translates a pseudo instruction, executes the resulting actual instruction, and then processes the next pseudo instruction. It does not produce a record of the executed program which could be run at another time.

Assemblers

Assembly routines produce a machine-language program from pseudo-language expressions. There are three types of pseudo-language expressions:

1. Control Instructions — Control Instructions are defined as instructions to which the assembler responds but which, in most cases, produce no actual machine-language instruction. They are control instructions to the assembler itself.
2. Pseudo Instructions — Pseudo Instructions can be defined as instructions or constants that cannot be used directly by the computer but, instead, must be translated into *real* (machine-language) instructions or constants by a special routine. Each pseudo instruction will result in one real instruction.

3. Macro Instructions — Macro Instructions are symbolic instructions which, upon assembly, result in several machine-language instructions.

There are basically two types of assembly programs: the Numeric Assembler like STAR I (see BURROUGHS Bulletin 5054) and the Mnemonic Assembler like STAR II, which will be described in detail presently. These two types of assemblers differ from one another in their addressing scheme. The mnemonic system uses alphanumeric names for the addressing of memory and the numeric system uses numeric (symbolic) names for referring to storage locations. In either case, the pseudo addresses are converted to actual machine addresses by the assembly program before the program is ready for computer use.

STAR II, a Mnemonic (alphanumeric) Assembler, is a routine which translates a program written in symbolic language into a machine-language program, which can be executed by the BURROUGHS 220.

Some advantages to be gained through its use are:

1. A reduction in the time and effort required to code and debug programs by performing some of the routine clerical and editing operations which would otherwise have to be undertaken by the programer himself.
2. STAR II has an alphanumeric addressing scheme. If this facility is used properly, the alphanumeric addresses will express logical references to the functions which they represent. This feature plus the use of adequate remarks will make the program listings easier to read and more meaningful.
3. STAR II uses easily remembered alphabetic operation codes.
4. The use of STAR II makes it easy for a programer to alter the program. STAR II has a reassembly feature which allows the programer to *change* instructions, *delete* instructions and *insert* instructions rapidly and without difficulty anywhere within the program.
5. STAR II has a Macro facility which enables a programer, by means of a symbolic reference, to incorporate standard or to generate specific subroutines and sections of coding for incorporation into the assembled program.
6. If desired, one of four versions of the BURROUGHS Magnetic-Tape Routine may be incorporated into the machine-language program by the use of a macro instruction. This routine will take care of all programed controls required when using magnetic tapes. It will accomplish the following for the programer automatically:
   a. The checking of input tape labels.
   b. The checking and writing of output tape labels.
   c. The necessary actions when end-of-lane conditions are encountered for both single- or multiple-block operations.
   d. The necessary action when end-of-file conditions occur.
   e. The ping-ponging of tape units.
   f. The procedures required for initial write operations.
   g. The procedures demanded by the automatic Break-In, Break-Out, and Restart features.
   h. The procedure for lane-parallel operations.
   i. The provision for control totals on the magnetic tapes used.
7. The machine-language program produced by STAR II is placed on a reel of magnetic tape. A program has been developed (the Program Library Creation Routine) that will place this machine-language program onto a program library tape (when the program is considered debugged) in a form suitable for use by an executive routine or the General Purpose Program Load Routine.

SECTION II

STAR II ASSEMBLY ROUTINE

The STAR II Assembly Routine is a three-pass assembly program written for the BURROUGHS 220. The routine requires one paper tape or card reader, at least 5000 words of memory and four magnetic-tape units. It is possible to get along with only three tape units but this is not recommended (See Alternate Operating Instructions, Section 8).

As far as output from the assembly routine is concerned, the machine-language program developed by the assembler will be placed on magnetic tape. By setting Program Control Switches, which will be explained in the Operating Instructions, Section 5, it is also possible to list the symbolic- and machine-language program on either the High Speed Printer or the on-line 407 Printer and, if desired, to punch the program into paper tape. However, no output media is absolutely necessary except for magnetic tape.

PASS I

During the first pass of the STAR II Routine the program written in symbolic language is read into the computer from either cards or paper tape.

As the (symbolically) named entries are read, a table is created in memory which relates these symbolic names to the memory locations which the assembler will assign to them. The assembler determines the machine locations it will assign by means of a Location Counter.

The Location Counter is a tally which the assembly program maintains. It is initially given a particular value by a **SET** control instruction, which will be explained later.

Each time a named pseudo instruction is encountered the symbolic name will be placed in the table with the corresponding Location-Counter setting. This setting will be the assigned machine-language location for that name. Then, the Location Counter is increased by one. For each unnamed pseudo instruction the Location Counter will also be increased by one but no entry will be made in Symbolic-Name and Machine-Location Table.

Also, during Pass I, the operation codes of the pseudo instructions are translated into numeric machine-language operation codes.

When a macro instruction is encountered, a list of valid macros is investigated to determine whether or not it is in the macro library. If the macro is found in this table, it will determine from the table how many coding instructions will be generated by the macro. The Location Counter will then be increased by this number to allow for the insertion (later) of the generated coding.

Some STAR II control instructions like **SET** and **BLK** (to be explained later) will also cause the Location Counter setting to be altered.

The assembly program will also detect various coding errors such as illegal operation codes and references to locations beyond the size of memory. (See Error Flags, Section 5.)

As STAR II reads each symbolic entry, performs the required Location-Counter operation, and places each name it encounters into a table, it writes out each symbolic entry onto a 21-word preblocked scratch tape (Symbolic Output Tape). This operation continues until an **END** entry is encountered, which signifies the end of the pass. At this time Pass I ends, the tapes are rewound and the Pass II routine of the assembly program is called in from the STAR II Program Tape.

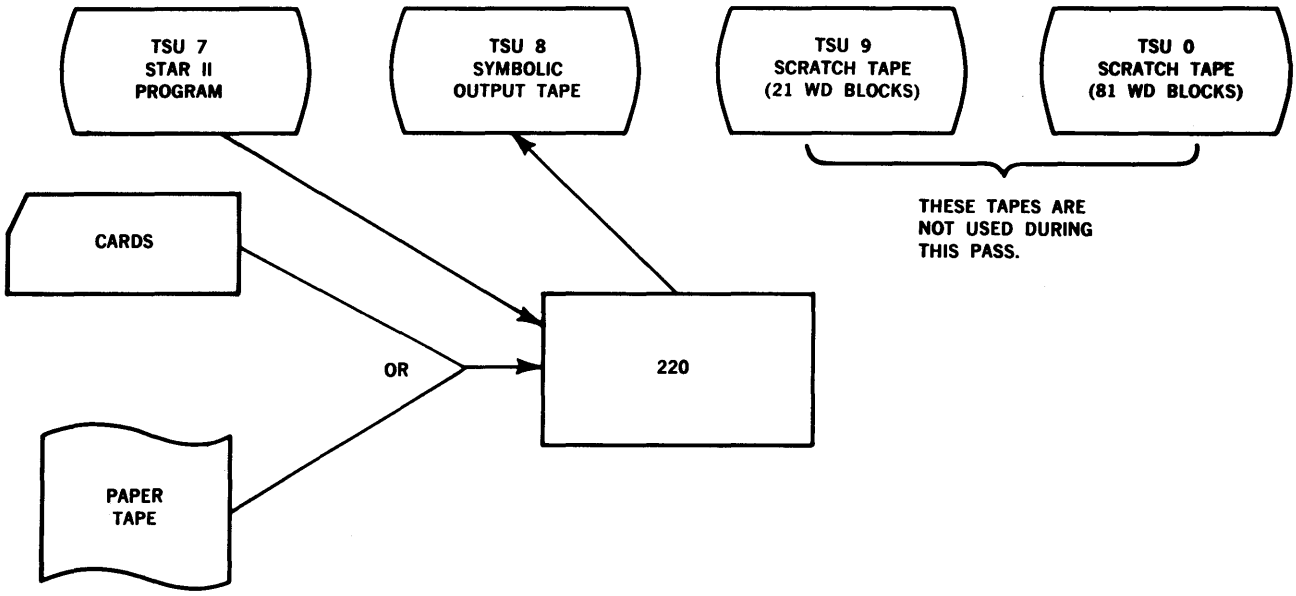


Figure 2-1. General Flow Chart of Pass I

WORD NO.	WORD DESCRIPTION	WORD LAYOUT										
		±	1	2	3	4	5	6	7	8	9	0
PREFACE	NUMBER OF WORDS PER BLOCK, 00=100		2	1								
1	Identification Number (Cols. 1-9)	0	I	I	I	I	I	I	I	I	I	0
2	Name (First Half) (Cols. 25-29)	2	N	N	N	N	N	N	N	N	N	N
3	Name (Second Half) (Cols. 30-34)	2	N	N	N	N	N	N	N	N	N	N
4	S, Control (Cols. 35-39)	0	0	0	0	0	0	S	C	C	C	C
5	Oper, S (Cols. 40-44)	2	0	P	0	P	0	P	0	P	S	S
6	Address/Constant (First Half) (Cols. 45-49)	2	A	A	A	A	A	A	A	A	A	A
7	Address/Constant (Second Half) (Cols. 50-54)	2	A	A	A	A	A	A	A	A	A	A
8*	±, Actual/Increment (Cols. 55-59)	0	0	0	0	0	0	±	I	I	I	I
9	Remarks (Cols. 60-64)	2	R	R	R	R	R	R	R	R	R	R
10	Remarks (Cols. 65-69)	2	R	R	R	R	R	R	R	R	R	R
11	Remarks (Cols. 70-73)	0	0	2	R	R	R	R	R	R	R	R
12	Page, Line (74-80)	0	0	0	0	L	L	L	P	P	P	P
13	STAR II Assigned Address	0	0	0	0	0	0	0	N	N	N	N
14	Partially Assembled Instruction	0	N	N	N	N	N	N	N	N	N	N
15	Error Flags	2	A	A	A	A						
16	Error Flags	2	A	A	A	A						
17	Error Flags	2	A	A	A	A						
18	Error Flags	2	A	A	A	A						
19	Error Flags	2	A	A	A	A						
20	Error Flags	2	A	A	A	A						
21	Special Flags Used By STAR II	0	N	N								

Figure 2-2. Layout of Symbolic Output Tape Produced by Pass I

*This field is Alphabetic when there is a Remarks entry.

PASS II

During Pass II the symbolic entries written on the 21-word preblocked tape during Pass I are read into the 220, one at a time. All the symbolic references are looked up in the table created in memory during Pass I and are assigned their proper machine-language address. As macro instructions are encountered a macro generator is called into memory from the STAR II Program Tape and control is turned over to the generator after STAR II has translated all of its parameters. The macro generator then produces the required coding steps and places them into an area known as the Graveyard. After all coding steps have been generated, the macro generator will return control to STAR II which then removes the generated coding from the graveyard and places it in the program.

Halt instructions will be generated in place of the macro coding if any of the following conditions occur:

1. An incorrect number of parameters has been given
2. Any parameter cannot be translated properly
3. The parameters are obviously incorrect.

The second pass of STAR II will detect and flag certain programing errors just as it does in Pass I.

As each symbolic entry is translated during Pass II into one or more machine-language instructions, the entire symbolic entry including the machine-language equivalent and the error flags, if any, is written out onto a special 21-word preblocked output tape, called the Symbolic and Machine-Language Output Tape.

It is also possible to produce a listing on the 407 Printer or the High-Speed Printer (see Figure 2-4) of the complete entry, just as it appears on the Symbolic and Machine-Language Output Tape. This is accomplished by selection of the appropriate Program Control Switches, which are explained in the Operating Instructions, Section 5.

Pass II then continues until a symbolic entry **END** is read. After the tapes are rewound and the Pass III routine of the assembly program is called in from the STAR II program tape, Pass II is completed.

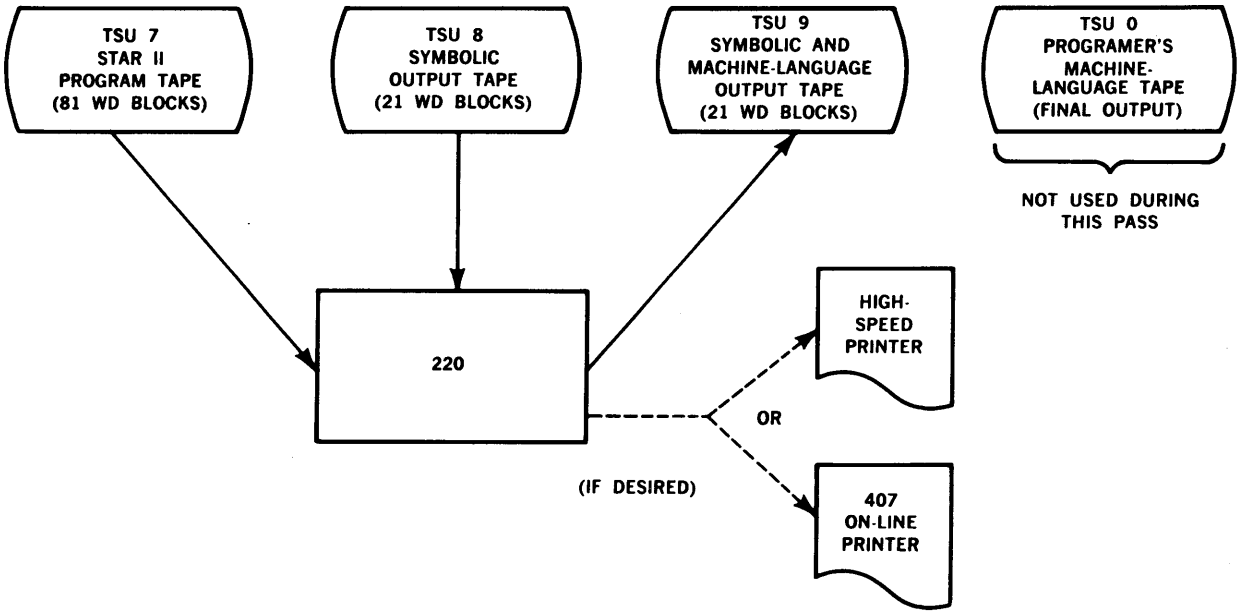


Figure 2-3. General Flow Chart of Pass II

ERROR FLAGS	PAGE & LINE #	NAME	OPR	ADDRESS/CONSTANT	A/I	LOCATION COUNTER	MACHINE LANG.	REMARKS
LF	01 003		0000 BCE	CHECK TRAN	0000	0487	0000 35 0550	
LF	02 003		3112 MOW	MASTER	0000	0488	3112 56 0270	
LF	03 003	READNXMAS	4100 MRD	MASTER	0000	0489	1100 52 0270	
LF	04 003		0000 CAD	MASTER	0000	0490	0000 10 0270	
LF UA	05 003		0000 STA	WA4	0000	0491	0000 40 0000	
LF	06 003		0000 BUN	START	0000	0492	0000 30 0469	
LF	07 003	ADDTO	0000 CAD	CARD	0000	0493	0000 10 0330	
LF	08 003		0103 BFA	*	0004	0494	0103 36 0498	
LF	09 003		0010 SPO	CARD	0002	0495	0010 09 0332	
LF	10 003		0050 SPO	ERROR	0004	0496	0050 09 0673	
LF	11 003		0000 BUN	READNXCARD	0000	0497	0000 30 0523	
LF	12 003		0000 CAD	CARD	0005	0498	0000 10 0335	CON
LF	13 003		0000 STA	WRITE	0000	0499	0000 40 0410	
	14 003		0000 CAD	CARD	0002	0500	0000 10 0332	AS
	15 003		0000 STA	WRITE	0001	0501	0000 40 0411	
	16 003		0000 CAD	CARD	0001	0502	0000 10 0331	
	17 003		0000 STA	WRITE	0003	0503	0000 40 0413	ID
	18 003		0000 CAD	CARD	0004	0504	0000 10 0334	
	19 003		0000 STA	WRITE	0002	0505	0000 40 0412	NO
	00 004		0000 LDB	WRITE	0004	0506	0000 42 0414	
	01 004		0050 RTF	CARD	0007	0507	0050 29 0337	DE
	02 004		0000 CAD	CARD	0014	0508	0000 10 0344	
	03 004		0000 STA	WRITE	0009	0509	0000 40 0419	PC
	04 004		0000 CAD	CARD	0015	0510	0000 10 0345	
	05 004		0000 SLA		0004	0511	0000 49 0004	
	06 004		0000 STA	WRITE	0010	0512	0000 40 0420	SEC
	07 004		0000 CAD	CARD	0012	0513	0000 10 0342	PLE
	08 004		0000 SLA		0001	0514	0000 49 0001	
	09 004		0000 ADD	CARD	0013	0515	0000 12 0343	NON
	10 004		0000 STA	WRITE	0011	0516	0000 40 0421	
	11 004		1100 MOW	MASTER	0000	0517	1100 56 0270	
	12 004	SXITCH1	0000 NOP	*	0002	0518	0000 01 0520	
	13 004		0000 BUN	READNXCARD	0000	0519	0000 30 0523	
	14 004		7229 DFL	SWITCH1	0000	0520	6229 27 0518	
	15 004		1100 CWR	WRITE	0000	0521	1100 61 0410	
	16 004		0000 BUN	INQUIRY	0000	0522	0000 30 0616	
	17 004	READNXCARD	1000 CRD	CARD	0020	0523	1000 60 0350	
	18 004		0000 LDB	CARD	0000	0524	0000 42 0330	
	19 004		0160 RTF	CARD	0005	0525	0160 29 0335	
	00 005		0000 CAD	CARD	0002	0526	0000 10 0332	
	01 005		0000 ADD	CARD	0003	0527	0000 12 0333	

Figure 2-4. Sample Listing

WORD NO.	WORD DESCRIPTION	WORD LAYOUT										
		±	1	2	3	4	5	6	7	8	9	0
PREFACE	NUMBER OF WORDS PER BLOCK, 00=100		2	1								
1	Identification Number (Cols. 1-9)	0	I	I	I	I	I	I	I	I	I	0
2	Name (First Half) (Cols. 25-29)	2	N	N	N	N	N	N	N	N	N	N
3	Name (Second Half) (Cols. 30-34)	2	N	N	N	N	N	N	N	N	N	N
4	S, Control (Cols. 35-39)	0	0	0	0	0	0	S	C	C	C	C
5	Oper, S (Cols. 40-44)	2	0	P	0	P	0	P	0	P	S	S
6	Address/Constant (First Half) (Cols. 45-49)	2	A	A	A	A	A	A	A	A	A	A
7	Address/Constant (Second Half) (Cols. 50-54)	2	A	A	A	A	A	A	A	A	A	A
8*	±, Actual/Increment (Cols. 55-59)	0	0	0	0	0	0	±	I	I	I	I
9	Remarks (Cols. 60-64)	2	R	R	R	R	R	R	R	R	R	R
10	Remarks (Cols. 65-69)	2	R	R	R	R	R	R	R	R	R	R
11	Remarks (Cols. 70-73)	0	0	2	R	R	R	R	R	R	R	R
12	Page, Line (74-80)	0	0	0	0	L	L	L	P	P	P	P
13	STAR II Assigned Address	0	0	0	0	0	0	0	N	N	N	N
14	Assembled Instruction	0	N	N	N	N	N	N	N	N	N	N
15	Error Flags	2	A	A	A	A						
16	Error Flags	2	A	A	A	A						
17	Error Flags	2	A	A	A	A						
18	Error Flags	2	A	A	A	A						
19	Error Flags	2	A	A	A	A						
20	Error Flags	2	A	A	A	A						
21	Special Flags Used By STAR II	0	N	N								

Figure 2-5. Layout of the Symbolic and Machine-Language Output Tape Produced by Pass II

PASS III

The input to Pass III is the symbolic entries and machine coding found on each block of the Symbolic and Machine-Language Output Tape. The blocks are read one at a time and the machine-language instruction which was produced during Pass II is extracted from the symbolic instruction and error flags, etc., and is then placed in its proper location in core storage. This continues until a **DUM** pair is read. The **DUM's** cause the core-stored machine-language program to be dumped on a special tape. This tape is known as the Programer's Machine-Language Tape. It is an 81-word preblocked tape.

If overlays are used, additional entries will be read and stored in core memory until the next **DUM** pair is encountered, then the specified area of memory is placed on the Programer's Machine-Language Tape. This continues until an **END** entry is reached. The assembly is now completed.

As just stated, this tape is an 81-word preblocked tape. Beginning each lane of tape is a 10-word **BMTR** tape-label block. Following the tape label are 81-word blocks each containing up to 80 words of the Machine-Language Program and a one-word search key found in word number 1. This search key has the following format:

±	1	2	3	4	5	6	7	8	9	0
±	0	0	0	P	G	M	N	N	N	N

Where,

PGM is a three-digit program number assigned to the program by the programer and entered into the assembler via the **NAM** entry (a STAR II control instruction to be explained later).

*** This field is Alphabetic when there is a Remarks entry.**

NNNN is the block address of each block on the Programer's Machine-Language Tape. These numbers are assigned consecutively to each block by STAR II during Pass III. The first block of the program itself has **NNNN** = 0001. The **LOD** block has **NNNN** = 0000.

Following the last data block on the Programer's Machine-Language Tape, STAR II will place an end-of-program control block. Since the **LOD** Pseudo and the **OVLY** macro separately control all program loading, a programing error will occur if the control block is ever encountered. The layout of the Programer's Machine-Language Tape is pictured below: (81-words/block)

Table 2-1. Layout of Programer's Machine-Language Tape

TYPE OF BLOCK	BLOCK NUMBER	BLOCK SIZE
BMTR LABEL		10 words
LOD Block-Program Self Loader	PGM 0000	81 words
Program (80 instructions per block)	PGM 0001 to End of Program	81 words
Control Block	PGM 9999	81 words

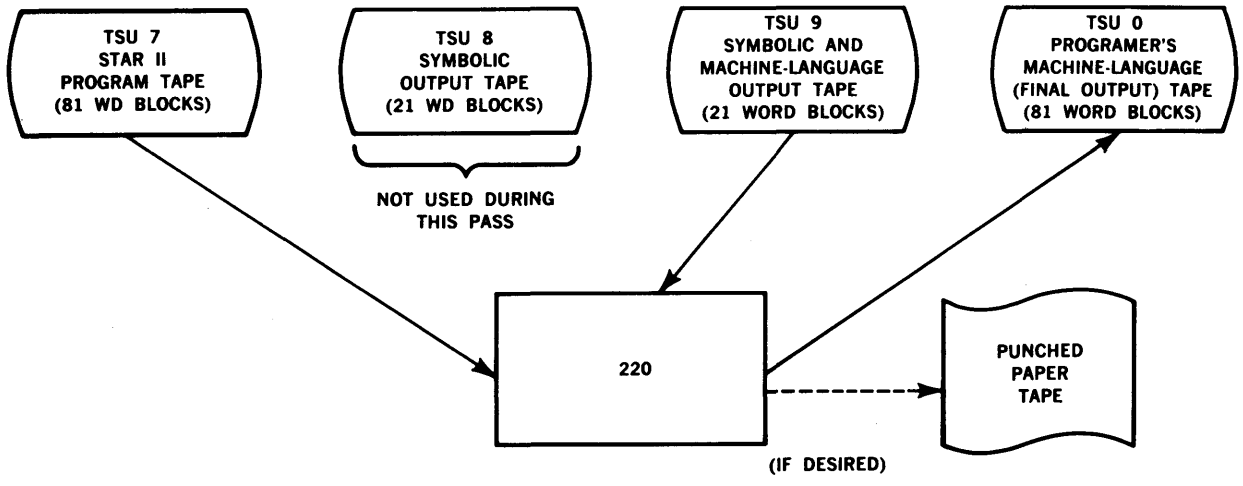


Figure 2-6. General Flow Chart of Pass III

REASSEMBLY

The Reassembly feature of STAR II, selected by a Program Control switch, makes the correction of programs fast and efficient. To use Reassembly the 21-word output tape (the Symbolic Output Tape) developed by Pass I of the last assembly or reassembly must be saved. This tape will thus provide, at high speed, all of the symbolic entries of the program. Only the correction cards or paper-tape messages must be prepared in the form of symbolic Changes, Deletions or Insertions.

Although the cards (or paper tape) and the magnetic tape are used as input to the Reassembly pass of STAR II, the Reassembly routine acts just like Pass I with one exception. The symbolic names on the change cards or paper tape are matched against the symbolic entries on the magnetic tape. Whenever they are found to be equal, the Reassembly routine will take the corrective action as indicated by the correction card or paper-tape message. Once this has been done the Reassembly pass proceeds exactly like Pass I.

Passes II and III will follow and perform in the same manner as they would during an assembly. For further information regarding Reassembly and its use refer to Use of Reassembly, Section 6.

SECTION 3

STAR II CODING FORMS

Before a discussion of the use and content of the STAR II control instructions, pseudo instructions and macro instructions, it is appropriate to review the STAR II paper-tape and card forms which will be used to explain these pseudo codes. An example of the forms used to code a program in STAR II notation can be found in Figures 3-1, 3-2, and 3-3.

Some basic rules for writing on the forms must be observed so that the coding may be punched properly.

Write:
- Letter O as Ø.
- Digit 0 as 0.
- Letter i as I.
- Digit one as 1.
- Letter z as Ƶ.
- Number 2 as 2.

All symbolic names and addresses *must* be written left justified. Since a blank column is regarded as zero, all zeros may be omitted or written as desired.

On the paper-tape form, the input data as stated corresponds to the word layout on the paper tape.

On the card form the columns correspond to card columns and the information written in each column is punched into the corresponding column of the card. Each line on the coding form, therefore, represents one 80-column punched card.

WORD NO.	WORD DESCRIPTION	WORD LAYOUT										
		±	1	2	3	4	5	6	7	8	9	0
1	Identification Number	0	1	1	1	1	1	1	1	1	1	0
2	Name (First Half)	2	N	N	N	N	N	N	N	N	N	N
3	Name (Second Half)	2	N	N	N	N	N	N	N	N	N	N
4	S, Control	0	0	0	0	0	0	S	C	C	C	C
5	Oper, S	2	0	P	0	P	0	P	0	P	S	S
6	Address/Constant (First Half)	2	A	A	A	A	A	A	A	A	A	A
7	Address/Constant (Second Half)	2	A	A	A	A	A	A	A	A	A	A
8	±, Actual/Increment	2	S	S	A	A	A	A	A	A	A	A
9	Remarks	2	R	R	R	R	R	R	R	R	R	R
10	Remarks	2	R	R	R	R	R	R	R	R	R	R
11	Remarks*	0	0	2	R	R	R	R	R	R	R	R
12	Page, Line	0	0	0	0	L	L	L	P	P	P	P

Figure 3-1. Paper-Tape Layout for an Entry During Assembly

*Leading zeros should be suppressed.

WORD NO.	WORD DESCRIPTION	WORD LAYOUT										
		±	1	2	3	4	5	6	7	8	9	0
1	Identification Number	0	1	1	1	1	1	1	1	1	1	0
2	CID*	0	0	0	0	0	0	0	0	2	C	C
3	C Name (First Half)	2	N	N	N	N	N	N	N	N	N	N
4	C Name (Second Half)	2	N	N	N	N	N	N	N	N	N	N
5	C Increment	0	0	0	0	0	0	0	I	I	I	I
6	Name (First Half)	2	N	N	N	N	N	N	N	N	N	N
7	Name (Second Half)	2	N	N	N	N	N	N	N	N	N	N
8	S, Control	0	0	0	0	0	0	S	C	C	C	C
9	Oper, S	2	0	P	0	P	0	P	0	P	S	S
10	Address/Constant (First Half)	2	A	A	A	A	A	A	A	A	A	A
11	Address/Constant (Second Half)	2	A	A	A	A	A	A	A	A	A	A
12	±, Actual/Increment	2	S	S	A	A	A	A	A	A	A	A
13	Remarks	2	R	R	R	R	R	R	R	R	R	R
14	Remarks	2	R	R	R	R	R	R	R	R	R	R
15	Remarks*	0	0	2	R	R	R	R	R	R	R	R
16	Page, Line	0	0	0	0	L	L	L	P	P	P	P

Figure 3-2. Paper-Tape Layout for an Entry During Reassembly

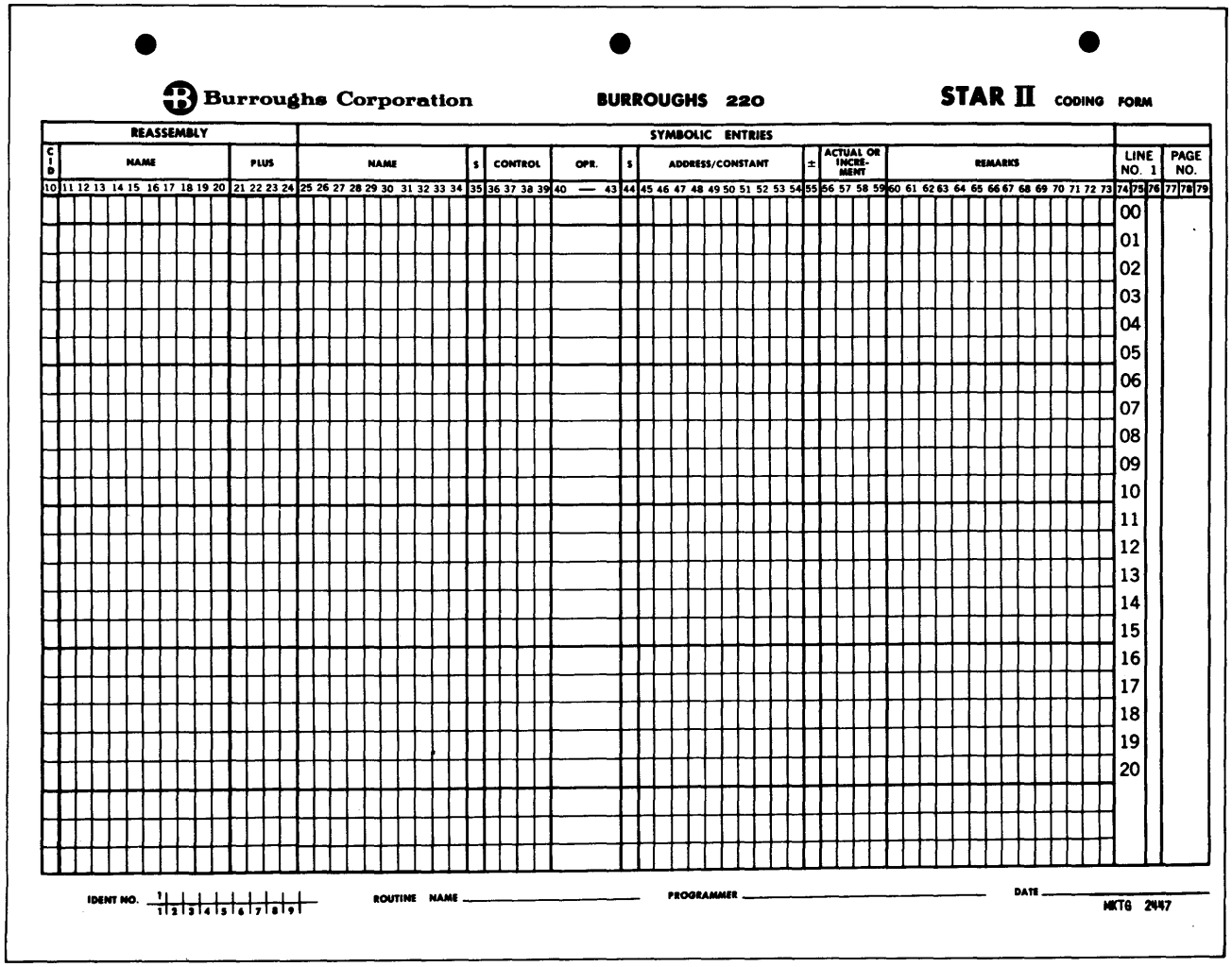


Figure 3-3. Star II Coding Form

*Leading zeros should be suppressed.

CODING FORM COLUMN IDENTIFICATION

COLUMNS

1-9 The **IDENTIFICATION NUMBER** of the program. Column one will select format band 1.

10-24 These columns are used for symbolic reassembly and are discussed in Section 6 — Use of Reassembly.

25-34 The **NAME** is a ten-character alphanumeric field. Any combination of digits, alphabetic characters, or special characters may be used in this field to represent the name (memory location) of data, instructions, or constants. Any field may be left unnamed. If the first five columns (25-29) are blank, the entire **NAME** field is considered blank. A total of *one thousand names* may be used in any one assembly.

35 The **SIGN** field is a one-digit numeric field which can contain any digit (0-9). Sign control is not exercised during assembly. Any entry with a 6 or a 7 in this column will be flagged as a potential programing error.

36-39 The **CONTROL DIGITS** are the four numeric digits which are part of every 220 operation code. These need not be written unless they differ from zero and then only when they are not part of the actual numeric statement of the instruction, itself (**CLA, CLR, SLT, CFR,** etc.). In other words, the program takes care of all variant digits.

40-43 The **OPERATION CODE** is a four-character alphabetic field which contains specific computer, macro, or control instructions (**CAD, MRD, CBSR,** etc.). Specific computer and control instructions are expressed by three alphabetic characters, left justified, while macros are stated by four-character operation codes. No numeric entries may be used.

44 **SIGN** is a one-digit numeric field. The sign of the constant which is written in columns 45-54 may be any digit (0-9). Sign control is not exercised during assembly, and any constant with a 6 or 7 in this position is flagged as a potential error.

45-54 The **ADDRESS/CONSTANT** field is a ten-digit alphanumeric field and is used for the following purposes:

Symbolic Address. When this field is used as a symbolic address, the symbolic address must refer to a named entry (columns 25-34). Care must be taken to be certain that this address is spelled in exactly the same manner as the name to which it refers. If a matching name is not found for every symbolic operand, the entry will be flagged on the listing as an error.

Blank. A blank operand will be assembled as address 0000.

Numerical Constants. The **NUM** pseudo instruction indicates that a numeric constant has been written in the Address/Constant field. The sign of the constant is in column 44. Zeros need not be written.

Alphanumeric Constants. The pseudo instruction **ALF** uses the first five columns of the Address/Constant field. These will be translated into the BURROUGHS 220 two-digit alphanumeric code with the sign digit as a 2. The last five columns of the field are ignored.

Number (#) or Asterisk (*). A # or * symbol in column 45 with columns 46-49 *blank* means that the symbolic address of the instruction is identical to the symbolic name of the instruction (its location counter setting). For example:

Location Counter Setting or Machine Location After Assembly	Name	OPR	S	Address/Constant	±	Actual or Increment
1000 1001 1002 1003	Modify	CAD ADD STA BUN		Net Pay Adjustment Adj. Net #	 1	 0003

Here, the **BUN** instruction's address has a # symbol in this field, hence it will be assembled as 1003. To this is added the increment —0003 leaving a final address of 1000.

CAUTION

Actual machine locations or numeric operands are not placed in the Address/Constant field.

55 A one-digit **INCREMENT SIGN** field (±) applies to the adjacent increment field. This field is entered as a zero or blank for plus, and a one for minus.

56-59 **ACTUAL/INCREMENT** is a four-character numeric field. The contents of this field (zeros need not be written) are added to or subtracted from the Address/Constant field, depending upon whether or not the sign of the increment (column 55) is zero or one. If the Address/Constant field is blank then column 55 is treated as plus and the Actual/Increment field becomes an *actual machine address.*

60-73 **REMARKS** is a 14-digit alphanumeric field. The programer may use this field to write descriptive information. This does not affect the assembly process in any way.

74-75 **LINE** is a two-digit numeric field which is preprinted on the form and does not affect assembly.

76 **LINE INSERT DIGIT** is a one-digit numeric field allowing up to nine program additions between each line. This digit does not affect the assembly.

77-79 **PAGE** is a three-digit numeric field. All pages must be numbered. This information does not affect the assembly.

80 **PAGE INSERT DIGIT** is a one-digit numeric field allowing up to nine pages of programing additions between each coding page. Space is provided for this purpose at the right-hand side of the coding form. It does not affect the assembly.

Columns 74-80 (page and line number) are used to facilitate manual or mechanical updating of the original symbolic program deck. It is very important that this deck be kept up to date at all times.

SECTION 4

CODING FOR ASSEMBLY

Certain rules must be followed when coding in STAR II. These rules are:

1. Instructions and constants cannot use locations 0000 to 0500, initially, because the Pass III program itself, which places the assembled coding into storage before writing it on the Programer's Machine-Language tape, is located in this area of memory. So its self-destruction cannot be allowed.

   The area from 0000 to 0500 can be used for the following purposes:

   a. As an input/output area — These memory locations can be named by a **BLK** pseudo (individually, if desired), a **SYN** pseudo or an **EQU** pseudo (which will be discussed later) but by no other pseudo codes. Any other entry in this area will inhibit Pass III.

   b. If a version of the BURROUGHS Magnetic Tape Routine is used and is called into the program by the macro **BMTR,** this routine will occupy all or part of this reserved area, depending upon the version used.

2. Do not **NAME** more than 1000 symbolic entries.

3. All coding must be left-justified within its respective field on the coding form, except for the Actual/Increment field, which is a fixed, four-digit, plus-sign field.

4. Use only alphabetical operation codes. Do not supply any variant digits for the instructions.

5. Zeros need not be written. Blank columns will be interpreted as zeros.

6. Make sure all spelling is exact. The misspelling of a symbolic reference will create a new reference and a program bug.

7. The use of the Actual or Increment field as a part of an address reference and the use of the # or * symbol in column 45 of the Address/Constant field should be used only when necessary, since the insertion of an instruction or constant in the wrong place during another assembly or reassembly will cause these references to become incorrect and will result in a number of program bugs.

8. A definite pattern or framework must be followed when coding in STAR II. The framework is comprised of a group of STAR II control instructions. STAR II control instructions, as earlier defined, are instructions to which the assembler itself responds, but which, in most cases, produce no actual machine-language coding. They are control instructions to the assembler.

   The required framework (format) for every program is:

   a. One **NAM** entry

   b. Five **LOD** entries

   c. One **SET** entry

   d. At the end of the program a pair of **DUM's** must be present.

   e. These **DUM's** must be followed by an **END** card which tells the assembler that the pass is complete and ultimately that the assembly is complete.

An example of the typical framework for a program might be:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT	
20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63
		JØHN DØE			NAM		021			
					LØD		P1			
					LØD		P2			
					LØD		P3			
					LØD		P4			
					LØD		P5			
					SET				0500	
		P1			—					
					—					
					—					
					—					
		P2			—					
		END ØF PGM			DUM		P1			
					DUM		P2			
					END					

Any other sequence of STAR II control instructions is illegal and will be rejected by the STAR II program. Note that if the initial **SET** entry or, for that matter, if any subsequent **SET** entries put the Location Counter below location 0500, the next entry will only be acceptable if it is a **BLK, SYN, EQU,** or another **SET** pseudo instruction. In any event, the Location Counter must be set to 0500 or greater, prior to the use of a pseudo instruction.

Since Pass III involves an extra tape pass, STAR II will check to see if the area of memory from 0000 to 0500 has been violated. If this area has been violated or a serious programing error has been committed, Pass III will be automatically inhibited.

Pass III may also be manually inhibited by using a Console switch.

STAR II CONTROL INSTRUCTIONS

Now that an example of the STAR II control instruction framework has been given, a detailed explanation of the function of each STAR II control instruction involved is in order.

NAM

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	REMAI
22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 6
	JØHN DØE			NAM		021		0001	

Description: **NAM** is used once and only once for every program and it must be the first entry. **NAM** identifies the program for later debugging and Master Program Library control. **NAM** does not appear in the assembled program.

The Name Field — contains the name of the programer. The Sign and Control fields are left blank.

The Operation Code — contains the STAR II control instruction.

The Address/Constant Field — This field contains the three-digit program number which the installation has assigned for identification purposes. The program number is left-justified.

The Actual or Increment Field — This field contains the assembly number. If a program's symbolic entries (cards or paper tape) have been updated and it is desired to assemble the program again, a one will be placed in this field to indicate it was the second time the program had been assembled. When Reassembly is used, this field will be updated automatically on the card image which appears on magnetic tape. This procedure facilitates locating the latest assembled program.

LOD (A series of five **LOD's** are required.)

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
					LOD		P1			
					LOD		P2			
					LOD		P3			
					LOD		P4			
					LOD		P5			

Description: The five **LOD** (STAR II) control instructions should be thought of as one pseudo instruction. The set of five **LOD's** causes a program-loading routine to be generated. This specialized program loader will enable the operator to call the program into memory from the Programer's Machine-Language Tape which was produced by Pass III of the assembly routine. It will also verify the Program Loading by means of a Hash Total.

The 81-word block of coding which these **LOD** control instructions have developed will immediately follow the Tape Label on the Programer's Machine-Language Tape and will be written on tape as the first block of the machine-language program.

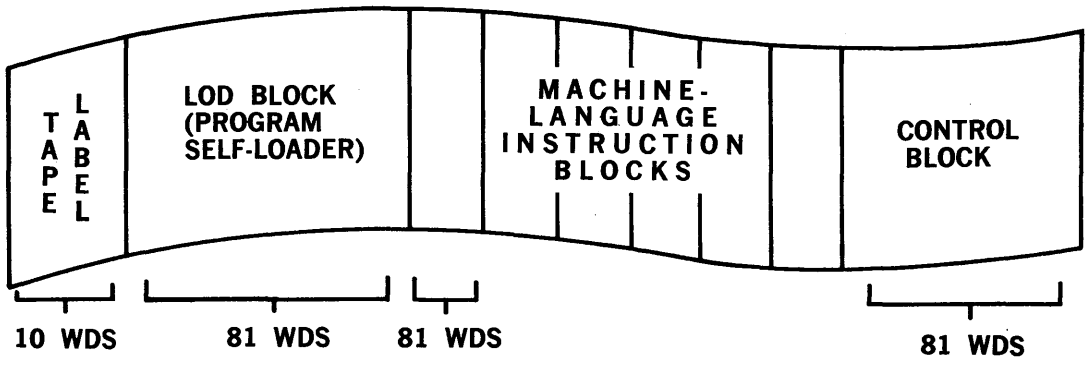


Figure 4-1. Layout of Programer's Machine-Language Tape

The specifications which must be provided to the Load Block Generator are expressed as P1, P2, P3, P4, and P5.

P1 — is the symbolic name of the first pseudo or macro instruction in the symbolic program.

P2 — is the symbolic name of the last named entry in the symbolic program.

P3 — is the symbolic or actual location of the first word of an 81-word temporary storage area in which the program's self-loading routine will be stored until the program has been loaded into memory. At that time, the loader is no longer needed and this portion of memory can be used by the program.

P4 – is the symbolic or actual location to which control will be transferred after the **LOD** Program has been executed.

P5 — is normally set to zero. However, by providing a value other than zero the programer can cause the location of the input tank used by the Program Load Routine, which is located in the area of memory specified by P3, to be changed from locations 0000 to 0080, to whatever 81-word area of memory is desired.

SET

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
8 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63
					SET				1550	
					SET		NET PAY			

Description: **SET** the Location Counter to the actual address found in the Actual/Increment field (for example LC = 1550).

Or, **SET** the location counter to the machine-language address which has been assigned by STAR II to the symbolic entry appearing in the Address/Constant field. *The symbolic address must have been previously named.*

The **SET** instruction itself cannot be named and, like the **NAM** and the **LOD** mentioned previously, does not appear in the assembled program.

A **SET** STAR II control instruction must immediately follow the five **LOD** control instructions in every STAR II program.

DUM

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62
		END MAIN			DUM		NAME 1			
					DUM		NAME 2			

Description: This instruction is used by STAR II during Pass III when core storage contains the machine-language program produced by the assembly routine. It causes the program located in core storage between the locations Name 1 and Name 2 to be placed on a fixed 81-word blocked tape, the Programer's Machine-Language Tape.

Note that a pair of **DUM's** are always used. The first **DUM** by means of the Symbolic Reference in its Address/Constant field designates the first symbolic location to be placed on the program tape. The address of the second **DUM** designates the last memory location to be dumped on the program tape. The **NAMES** appearing in the Address/Constant field of the coding form, which establishes the limits for writing on the

program tape, will have been assigned machine-language addresses by the Location Counter during Pass I; (like Name 1 = 1250 and Name 2 = 4300). Thus, when these **DUM's** are encountered by Pass III, all of the machine-language instructions and constants from memory location 1250 to location 4300 are placed on the Programer's Machine-Language Tape. If the total number of program steps to be dumped is not evenly divisible by 80, the unused portion of the last 81-word block taken from memory will be filled with zeros.

The **DUM** control instruction is used principally at the end of a program to dump the program on the program tape. However, if overlays are used, the first pair of **DUM's** will refer to the basic program. Other sets of **DUM's** will pertain to overlays. Overlays are blocks of instructions which can be called in from the program tape at any desired time to replace sections of the basic program. For a further discussion of overlays and how **DUM's** can be used for this purpose, see the **OVLY** (Overlay) macro, Section 4.

Regardless of the number of **DUM** pairs used in a program, the first set of **DUM's** must specify the symbolic addresses of the first and last entries of the basic program, and thus must be the same addresses designated by the first two **LOD's** (P1 and P2).

Due to the size of storage, 25 pairs of **DUM's** are the maximum number which may be used during any assembly.

The control instruction **DUM** does not alter the Location Counter and does not appear in the assembled program.

The first **DUM** entry of every **DUM** pair *must* be named.

END

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	
9 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 6
					END					

Description: **END** is used once and only once for every program. It identifies the end of the program for STAR II during assembly and is the last entry of every program. **END** does not appear in the assembled program.

Aside from those control instructions which form the framework or skeleton for every STAR II program and which are absolutely required in order for it to function, there are six other control instructions in the STAR II Assembly Routine which are very helpful to the programer but which are not necessary in order for the program to operate.

These useful control instructions which are instructions to the assembly routine itself are:

BLK

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
		CARD INPUT			BLK				0040	

Description: **BLK** reserves areas of memory for input, output, and intermediate storage. The name written in the Name field is assigned to the first location of the storage area (the value of the Location Counter) and then the Location Counter is increased by xxxx as supplied in the Increment field. This entry *must* be named. No other location in this storage area can be named (except by an **SYN** or **EQU** pseudo instruction), although each location in the area can be referenced by the use of increments.

The instruction **BLK** does not appear in the assembled program.

REM

IE	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	REMARKS	LINE NO.
0 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73	74 75
		REM			PASS 1 ØF		STAR	2 ASSEM. PRØG.	00
									01

Description: **REM** allows the remaining columns (45-73) on the line to be used for remarks. These remarks will appear in the STAR II listing but will not appear in the assembled program.

HED

IAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61
					HED		ABC	¤		

Description: The **HED** control instruction is normally used when two or more individuals are coding portions of the same program. It is designed to assist the programers in avoiding the complications and bugs arising from two people accidently using the same symbolic reference for different purposes, in two or more sections of the same program.

Beginning with the entry following **HED**, make the character found in column 54 of the **HED** entry the last character of all symbolic Names and Addresses, provided that the last character is not used by the Name or Address itself. If it is used, do not insert the character. Continue this procedure until another **HED** instruction changes the character or deletes it (column 54 blank).

Due to the method by which the assembler recognizes symbolic references, it is necessary to place a Name in columns 45-53, as well as the **HED** symbol in column 54. The Name which appears in columns 45-53 will have no other effect upon the program.

Thus, by using control instruction **HED**, even though two programers have used the same symbolic reference in their respective portions of the program, the assembler will not assign the same machine-language address to them. For example:

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT	
) 21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 6
	PROGRAMER	NO.	1:						
				STA		JØHN DØE			
	PROGRAMER	NO.	2:						
				HED		ABC	¤		
				ADL		JØHN DØE			

Because of the **HED** instruction, Programer No. 2's instruction will actually be recognized by STAR II as the following:

NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
24 25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67 68 69
			ADL		JØHN DØE ¤			

As far as the assembler is concerned, this address is completely different from plain John Doe.

The **HED** instruction will not appear in the assembled program.

EQU

NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
24 25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67 68 69
CUST. NØ.			EQU				1000	

Description: Assign the machine address found in the Actual/Increment field to the Name found in the Name field. This instruction allows more than one Name to reference the same location, but does not allow the same Name to be assigned to two or more locations. This instruction might be used where a storage location (for example location 1000) was used during the beginning of a problem to store "Net Pay," and during the latter part of the program to store "Personnel Statistics." When this instruction is given, the Address/Constant field *must be blank.*

The **EQU** instruction will not appear in the assembled program.

SYN

NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
24 25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67 68 69
JØHN DØE			SYN		JØHNDØE			EXAMPLE 1
NET PAY			SYN		GRØSS PAY		0003	EXAMPLE 2

Description: **SYN** makes synonymous the Name punched in the Name field and the Name in the Address/Constant field plus its increment, if any. The symbolic address appearing in the Address/Constant field must be an entry that was previously named. This allows a location to be called by more than one Name. (For example — a spelling error can be corrected by the use of **SYN.**)

This entry does not appear in the assembled program. All **SYN** cards must be placed just before the **END** card of the symbolic deck.

MOD

US	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67 68 6
				MØD				0010	

Description: The control instruction **MOD** is used to increase the setting of the Location Counter to the next higher number which is a multiple of xxxx found in the Increment field, where xxxx can be any number up to 9999. If the current Location Counter setting is an exact multiple of xxxx, do not alter it. If xxx = 0001, increase the Location Counter by 0001. xxxx = 0000 is an error, as is any xxxx which causes the

Location Counter to be increased beyond storage capacity. The instruction **MOD** will not appear in the assembled program.

After assembly, reassembly, or even production runs, all programs will normally be changed to some degree. If the alterations are minor, it may be desired to correct the machine-language coding rather than reassemble. The use of **MOD** will give the programer room wherever he desires within his program to make these corrections.

For instance, in the above example **MOD** was 0010. If the Location Counter were set to 1223 when this pseudo code was encountered, the Location Counter would be advanced to 1230, leaving seven unused memory locations for later program alterations.

However, be certain to keep symbolic corrections up to date, since the time will come when enough machine-language corrections have been made to warrant a reassembly.

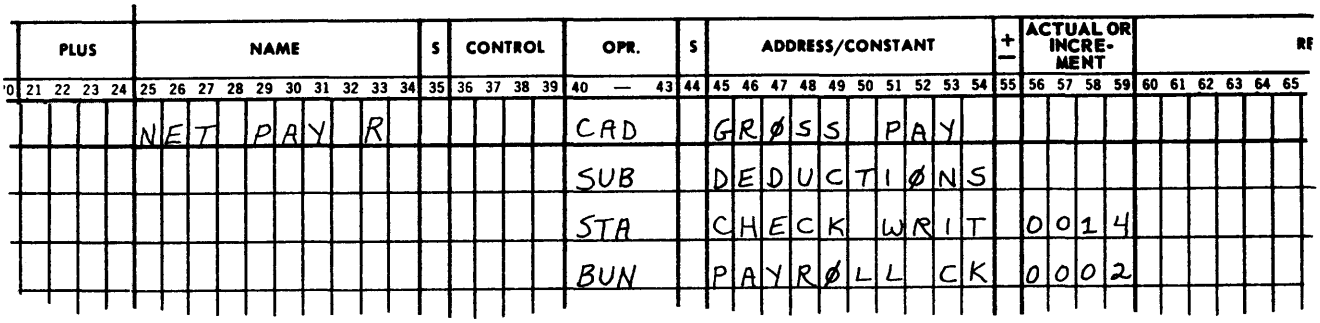
PSEUDO INSTRUCTIONS

Pseudo Instructions are instructions or constants that cannot be used directly by the computer but instead must be translated into "real" (machine-language) instructions or constants by a special routine. Each Pseudo Instruction will result in one real instruction.

All 97 BURROUGHS 220 instructions are included in this category. Any instruction can be named by a ten-character alphanumeric name and have a ten-character alphanumeric address. All 97 instructions will be represented by a three-character alphabetic code (for example **CAD, MDA, ADA,** etc.).

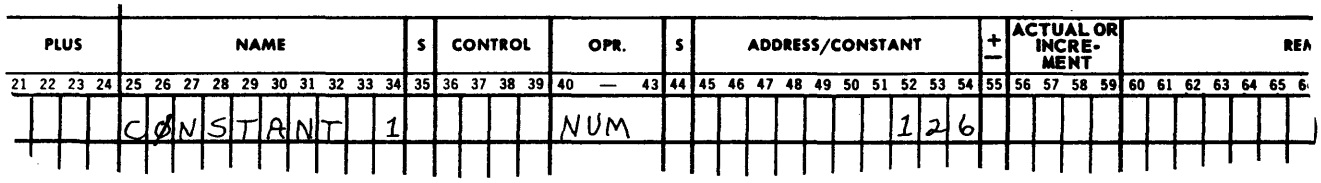
All digits in the control field of each instruction must be numerically coded as in machine-language coding (SL positions, etc.) except for the variant digits which distinguish between certain 220 instructions. The variant digit will be taken care of automatically by the assembler when it recognizes the three-character alphabetic operation code.

Example:



PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	RE
0 21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65
	NET PAY R			CAD		GRØSS PAY			
				SUB		DEDUCTIØNS			
				STA		CHECK WRIT		0014	
				BUN		PAYRØLL CK		0002	

There are two special pseudo instructions for which STAR II will produce machine-language coding. These instructions designate alphabetic and numeric constants. They are:
NUM



PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	REA
21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 6
	CØNSTANT 1			NUM		126			

Description: The **NUM** pseudo instruction indicates to the assembler that a *numeric* constant has been written in the Address/Constant field. The sign of the constant is in column 44. Zeros need not be written. A numeric constant can either be named or unnamed.

ALF

NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
CØNSTANT 2			ALF		A2B3			EXAMPLE 1
CØNSTANT 3			ALF		ABCDE			EXAMPLE 2

Description: The **ALF** pseudo instruction indicates to the assembler that an alphanumeric constant has been written in the first five columns of the Address/Constant field. The characters found in this field will be translated into the BURROUGHS 220 two-digit alphanumeric code with the sign position of the word in memory equal to two. The last five columns of the field are ignored. An alphanumeric constant can be named if desired.

For instance, EXAMPLE 1 (above) will be assembled and appear in memory as:

$$\pm$$
2 41 82 42 83 00

EXAMPLE 2 will be assembled and appear in memory as:

$$\pm$$
2 41 42 43 44 45

NOTE

A Lozenge symbol (¤) in an alphanumeric constant will produce a carriage return on the **SPO** printout wherever the symbol appears.

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	
	CØNSTANT 4			ALF		END¤			
	CØNSTANT 5			ALF		UN¤GØ			

MACRO INSTRUCTIONS

The third and last type of pseudo code is the macro. As already defined, a macro is an instruction which is coded in symbolic language and which upon assembly will result in several machine-language instructions.

The macro instructions in STAR II are represented by a four-character operation code. Each macro, when encountered, will call a generator routine into memory. This routine, based upon the parameters supplied to it by the macro instruction, will produce the desired coding. All the parameters associated with a macro must be stated in the exact manner prescribed by the particular macro or it will not be executed.

Each macro, regardless of the parameters supplied to it, will produce a fixed number of program steps.

MACRO INSTRUCTIONS IN THE STAR II ASSEMBLY ROUTINE

During Pass I, when STAR II encounters a macro, it locates this macro on a special table which is a part of the assembly program. The table contains specific information about each macro, namely:

1. The number of machine-language instructions that will be produced by the macro.

2. The number of parameters associated with the macro.
3. The search key for locating the proper macro generator on the STAR II Program Tape.

When a particular macro has been found on the Macro Table, the Location Counter setting is increased by the number of coding steps which the macro generator will produce. This permits the instructions, following the macro and its parameters, to be assigned to their proper memory locations during Pass II.

STAR II next refers to the Macro Table to determine the number of parameters to be read. It reads these parameters and records them for use during the second pass.

During Pass II, when the macro instruction is read from the Symbolic Output Tape, produced by Pass I, the following takes place:

1. STAR II locates the macro's generator on the STAR II Program Tape and places it in core storage, beginning with location 4000.
   a. Location 4000 will contain the search key of the generator.
   b. Location 4001 is the subroutine exit for the generator and contains a **BUN 0000** instruction.
   c. Location 4002 contains the parameter flag word. Since macro parameters may use either symbolic addresses or numeric constants, this flag word will tell STAR II whether or not each parameter will be translated from the symbolic to the actual. Each digit position of the flag word corresponds to one of the ten possible parameter words and will have either a zero or a one in it.
      A one will cause the parameter to be translated.
      A zero will leave the parameter in its original form.
      The SL:11 position of the flag word corresponds to the first parameter. The SL:21 position of the flag word corresponds to the second parameter and so forth. (See the Transfer Equal example below.)
   d. Location 4003 is the first instruction of the macro generator.
2. STAR II reads each of the macro parameters, translating them or not as indicated by the flag in Location 4002. The translated addresses and the constants (parameters) are placed in consecutive locations starting with Location 3990.
3. STAR II places the machine address, which will be the location of the first instruction of the generated coding (Location Counter setting), in location 3989.
4. STAR II places a flag of all nines in Locations 3988, if any parameter is incorrect or missing. This signals the macro generator so that it will not attempt to produce coding but rather *will place zeros in the area of memory which the generated coding would normally occupy.*
5. Next, control is transferred to the macro generator and the machine coding is produced. Then, the generated machine-language instructions are placed in consecutive locations starting with location 4500.
6. STAR II finally transfers this machine-language coding from location 4500 and following to its proper place in the program.

An example of a simple macro and its generator follows:
Macro: **TRANSFER EQUAL** **TREQ**

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	REMARK
22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67
	BAL. RØUT.			TREQ		GRØSS PAY			
						NET & DED.			
						CK WRITE			

* The macro need not be named.

Description: Compare the contents of the location specified by the first parameter to the contents of the location specified by the second parameter. If these locations are equal to each other transfer control to the cell indicated by the third parameter.

In the example above:

Gross Pay = the first location to be compared. (The first parameter.)

Net and Ded. = the second location to be compared. (The second parameter.)

Ck Write = the location to which control is to be transferred, if Gross Pay = Net and Ded. (The third parameter).

When this macro is encountered during Pass II the parameters, Gross Pay, Net and Ded., and Ck Write, will be translated by STAR II into machine-language addresses. For example:

Gross Pay = 0962
Net and Ded. = 1002
Ck Write = 1096

Further, if the memory location assigned by the Location Counter to the first step of the generated coding was 0516, the coding produced by the generator would be:

MACHINE LANGUAGE

LOCATION	±	CONTROL	OPR	ADDRESS
0516	0	0000	10 (CAD)	0962
0517	0	0000	18 (CFA)	1002
0518	0	0000	35 (BCE)	1096

During Pass II when the assembler turns control over to the generator for the development of the machine-language coding as shown above, the generator and its associated information (including the parameters) would appear as follows:

LOCATION	CONTENTS	
3988	0 0000 00 0000	(Error Flag)
3989	0 0000 00 0516	(Location Counter)
3990	0 0000 00 0962	(Gross Pay) Parameter 1
3991	0 0000 00 1002	(Net & Ded.) Parameter 2
3992	0 0000 00 1096	(Ck Write) Parameter 3
—		
—		Room for more Parameters
—		
3999		
4000	Search Key of the Generator	
4001	*BUN (rP)	(Exit from the generator Routine)
4002	0 1110 00 0000	(Parameter Flag word)
4003	0 0000 CAD 3988	(First Instruction of the generator)
4004	0 0099 BFA 4014	
4005	0 0000 CAD 3990	
4006	0 0410 STA 4018	
4007	0 0000 CAD 3991	
4008	0 0410 STA 4019	
4009	0 4500 CAD 3992	
4010	0 0410 STA 4020	
4011	0 4400 DLB 4009	
4012	0 0030 RTF 4018	

*The operation codes are expressed alphabetically for ease of reading. These codes would have their actual numeric values if they appeared in memory.

```
4013        0  0000 BUN 4001
4014        0  0000 CLL 4500
4015        0  0000 CLL 4501
4016        0  0000 CLL 4502
4017        0  0000 BUN 4001
4018        0  0000 CAD 0000
4019        0  0000 CFA 0000
4020        0  0000 BCE 0000
```

This will be the generated routine which is placed in the program, when the addresses are completed. In its ungenerated form this routine is normally coded relative to location 0000.

As already stated, for each macro instruction the programer must list the pseudo code and as many parameters as the macro requires. The parameters must appear in the Address/Constant field and/or the Actual/Increment field in a definite sequence. They may be stated in any of the following ways:

Address/Constant	± Actual/Increment
1. Symbolic Name	± Increment
2. Blank	Actual
3. Symbolic Name	
4. Constant (usually left-justified)	

Each macro will specify how the parameters are to be stated.

If desired, the macro instruction may be named. This name will be assigned to the first machine-language instruction produced by the macro.

How to Write, Incorporate and Delete Macros

The STAR II Assembly Routine allows each programer or installation to add macros to the macro library or to delete them from it. This helps an installation to implement standard practices by allowing for the use of specialized or general subroutines more easily.

To incorporate a generator routine for a macro instruction into STAR II, the following steps should be followed:

1. Code the macro generator in STAR II notation as if it were a regular program. The coding should be relative to location 4001. Hence, the initial **SET** pseudo code would set the Location Counter to 4001.
   Location 4000, the search key for this macro generator is left blank. This search key is assigned by the Macro Incorporate Routine when the macro is incorporated into the STAR II Macro Library.
2. The generator must be designed so that it will produce a fixed number of coding steps regardless of the parameters used. The maximum number of instructions which can be generated is 500.
3. No more than ten (10) parameter lines may be used for any one macro instruction. This includes the one parameter that is listed on the same line as the macro instruction itself. Parameters may be expressed as either symbolic references, actual addresses, or numeric constants. The flag word in location 4002 must indicate to STAR II which parameters require translation.
4. No macro generator may occupy more than 960 locations starting with location 4000. These 960 locations include the number of locations required for the generated coding.
5. The maximum number of macro instructions that can be incorporated into the STAR II Macro Library is 60.

6. After the generator has been coded it should be assembled as if it were a regular program. Be certain that the generator is written so that the generated coding which it produces will be placed in memory locations 4500 and following.
7. All macros have been written for programs using the Overflow Remember mode of operation. If programs are written for the Overflow Halt mode of operation and it is desired to use the existing macros, each macro instruction should be preceded by an **SOR** (Set Overflow Remember) and followed by **SOH** (Set Overflow Halt) instruction.
8. The routine used to incorporate a new macro instruction into STAR II is called the Macro Incorporate Routine.

MACRO INCORPORATE ROUTINE

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader
2. One Supervisory Printer
3. Two Magnetic Tape Storage Units

BILL OF MATERIALS

1. Plugboards — None
2. Input Forms — The Macro Incorporate Routine in paper-tape form
3. Output Forms — Paper for the Supervisory Printer
4. Magnetic Tapes
   a. The STAR II Machine-Language Program Tape.
   b. The Programer's Machine-Language Tape that STAR II produced when it assembled the generator.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the Macro Incorporate Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer
   a. Load the Supervisory Printer with the paper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO.**
   d. For further details, refer to the standard installation procedures for setting up the **SPO.**
3. Magnetic Tapes
   a. Mount the STAR II Machine-Language Program Tape on a tape unit and designate it as TSU 7.
   b. Mount the Programer's Machine-Language Tape that STAR II produced when it assembled the generator on a tape unit and designate the unit as TSU 0.
   c. Follow the standard installation procedures for the handling and mounting of the magnetic tapes.
   d. Make certain that all other Tape Storage Units and Datafiles are designated as Local.
4. Program Control Switches

PCS	ON	OFF
1	The table item is incorrect. Depress Start. (See **SPO** message on next page for more information.)	The table item is correct. Depress Start.

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into register (r)D. Transfer rD to rC.
3. Depress the Start button. The Macro Incorporate Routine will now run automatically until completion or interruption by a supervisory printout.

   The operator will respond to all messages and programed halts, taking the proper action as indicated below in the section on Supervisory Messages and Programed Halts.
4. Upon completion of the run, remove it from the computer, following the installation's standard procedures.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE	EXPLANATION
Incorporate routine for STAR II.	For identification purposes.
Enter macro op code — right-justified.	Enter the eight numeric digits which represent the alphabetic macro op code (i.e. BMTR = 0042546359).
Enter No. of parameters. If zero or one, Enter 1.	Right-justified.
Enter No. of coding steps produced	Right-justified.
Macro op code already used. Enter new op code.	Self-explanatory.
The following macro has been incorporated into STAR II —	This will state the alphabetic operation code for the macro.
The search key is_____.	Self-explanatory.
MT-7 — not ready.	Check TSU No. 7 and depress Start.
Can not locate Pass I.	Pass I cannot be found on the STAR II program tape. Change program tape and restart.
Can not locate nines flag on MT-7.	At the end of the program on the STAR II Machine-Language Tape, the following blocks should have search keys of 9___9.
Too many nines on MT-7.	The search has failed. Depress Start and the search will be retried.
Tape too full.	Self-explanatory.
Macro table is full.	60 macros are already placed on the Macro Table.
MT-10 not ready.	Check TSU No. 0 and depress Start.
The table item is_____. If good turn PCS-1 Off, if bad turn PCS-1 On.	The operator will verify the table item number and take the appropriate action as indicated by the SPO.
Macro is too big.	The macro generator is more than 960 instructions or the generated coding is more than 500 instructions.

MACRO DELETE ROUTINE

This routine allows an installation to delete an old macro from the STAR II Macro Library. At the completion of this routine, the designated macro will have been deleted both from the Macro Table and from the STAR II program tape.

To use this routine, follow the Operating Instructions listed below.

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader
2. One Supervisory Printer
3. One Magnetic Tape Storage Unit

BILL OF MATERIALS

1. Plugboards
   None.
2. Input Forms
   The Delete Macro Routine is in paper-tape form.
3. Output Forms
   Paper for the Supervisory Printer.
4. Magnetic Tapes
   The STAR II Machine-Language Program Tape.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the Delete Macro Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO.**
   d. For further details, refer to the standard installation procedures for setting up the **SPO.**
3. Magnetic Tapes
   a. Mount the STAR II Machine-Language Program Tape on a tape unit and designate it as TSU 7.
   b. Follow the standard installation procedures for mounting the tape.
   c. Make certain that all other Tape Storage Units and Datafiles are designated as Local.
4. Program Control Switches

PCS	ON	OFF
1	The macro identified is incorrect. Depress Start and the program will begin again.	The macro identified is correct. Depress the Start button.
2	Retry. Depress Start.	

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Depress the Start button. The Delete Macro Routine will now run automatically until completion or interruption by a supervisory printout.

   The operator will respond to all messages and programed halts, taking the proper action as indicated below in the section on Supervisory Messages and Programed Halts.
4. Upon completion of the run, remove it from the computer, following the installation's standard procedures.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE	EXPLANATION
SQUOOSH Routine for STAR II	For program identification purposes.
Enter macro op code – right-justified.	This is the identification number which will be assigned to this macro generator.
The macro to be deleted is_____.	For operator verification purposes.
If correct, Start. If not, turn PCS 1 on, then Start.	If the number of the macro is correct depress Start. If the number of the macro is incorrect, turn PCS 1 On and depress Start.
The following macro has been deleted from STAR II – _____.	This is provided for further identification and as a record of the macro which was deleted.
MT-7 not ready.	Check TSU 7 and depress Start.
Cannot locate Pass I.	Change the STAR II Program Tape and start over.
Macro cannot be found in Macro Table. To stop the routine, depress Start. To retry turn PCS 2 On, then depress Start.	Self-explanatory.

THE STAR II MACRO LIBRARY

The macro instructions which are presently included in the STAR II Macro Library, listed in alphabetical order, are:

MACRO	OP CODE
ALTERNATE	ALTA
BURROUGHS MAGNETIC TAPE ROUTINE	BMTR
CLEAR CORE STORAGE	CLST
CLEAR CORE STORAGE SELECTIVELY	CLSS
CREATE BINARY SEARCH ROUTINE	CBSR
DATE DIFFERENCE	DADI
DATE TRANSLATE DAY	DADA
DATE TRANSLATE MONTH	DAMØ
DELETE ITEM	DELE
FIELD SUM	FLDS
FIRST TIME NOP	FNØP
FIRST TIME TRANSFER	FTTR
GROUP CHANGE	GPCH
HALT AND TRANSFER	HATR
HASH TOTAL	HASH

INSERT ITEM	INIT
LOOP	LØØP
LOOP RESET	LØPR
MOVE	MØVE
OVERLAY	ØVLY
READ-WRITE CHECK	RWCK
RESTORE INDICATORS	REIN
RESTORE REGISTERS	RERG
SET SWITCH	SSWH
SEQUENCE CHECK	SEQC
SEQUENCE TABLE	SETA
STORE INDICATORS	STIN
STORE REGISTERS	STRG
SWITCH	SWCH
SWITCH RESET	SWRS
TEST AND PRINT PROGRAM SWITCHES	TPPS
VARIABLE HASH	VAHA

As previously mentioned, a Macro Table containing various pieces of information pertaining to each macro is maintained by the Star II Program. If it is desired to find out what macros are contained on the Macro Table of the Star II Program Tape, the following procedure can be followed:

1. Rewind the Star II Program Tape.
2. Position forward two blocks (past the Tape Label and the Program Load Block).
3. Read the next two blocks into memory.
4. Print the memory locations into which the blocks were read.
5. The printout will look like the following:

MACRO OPERATING CODE	NO. OF PARAMETERS	NO. OF INSTRUCTIONS PRODUCED	MACRO GENERATOR SEARCH KEY
SWCH	1	001	0011000
TPPS	1	010	0011010
STIN	1	015	0011020

This printout will give a complete index to all the macros in the Macro Library. By using this index, the search key which corresponds to the desired macro code can be determined. It is then possible to field search (SL:07) for a given Macro Generator. The address word of the block which is found by the search will contain the search key of the macro (SL:07) plus the number of blocks which the macro occupies (SL:33). By completing the reading of the number of blocks specified, the particular Macro Generator can be read into memory. If the generator is more than one block in length, be sure to ignore the search keys which are the first word of each block, since these search keys are ignored (stripped off) by the routine which loads the Macro Generator during Pass II. Only the search key of the first block remains a part of the generator.

ALTERNATE – **ALTA**

Description: This macro will alternately cause a transfer of control to the symbolic locations specified by the parameters (P1 and P2). It will alternately perform either a **BUN** to P1 or a **BUN** to P2, starting first with a **BUN** to P1.

The two parameters are symbolic addresses appearing in the Address/Constant field.

This macro may be named, if desired.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT	
20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 6
		ANY NAME			ALTA		P1			
							P2			

Restrictions: rA is altered.

Number of instructions generated: 5

Example:

LUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT	REMARKS
23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67
	ALTERNATE			ALTA		RD MASTER			
						RD EMPLØYE			

BURROUGHS MAGNETIC TAPE (HANDLING) ROUTINE – BMTR

Description: This macro will cause one of four versions (A, B, C or D) of **BMTR** to be included as part of the object program. The tape-handling routine will be placed into memory starting with location 0000.

1. The A version of **BMTR** provides the programer with a routine which will take care of all the programed controls normally required with the use of magnetic tape. It includes:
   a. The verification of Input tape labels.
   b. The writing of Output tape labels.
   c. The taking of storage dumps upon an end-of-lane condition, if desired.
   d. The counting and verification of the number of input blocks.
   e. The facility for providing program controlled restarts.
   f. A provision for manually controlled Break-In and Break-Out procedures.
   g. Automatic direction of unit and/or lane flip-flops (ping-pongs), if this is desired.
   h. The writing of control blocks when necessary.
   i. The writing of end-of-tape blocks when necessary.
   j. The handling of multiple block read and write conditions.
   k. The development of a log-of-tape operations via the **SPO.**
   l. A provision for lane-parallel tape operations.
   m. The initial writing on edited tapes.

   This routine occupies the first 597 words of memory. (**SET** $\geq$ 0597, initially.)
2. The B version provides all of the above tape-handling procedures except the following:
   a. The initial writing on edited tape.
   b. The use of lane-parallel operations.

   This version occupies fewer memory locations than version A. It requires the first 542 words of memory. (**SET** $\geq$ 0542, initially.)
3. The C version provides all of the tape-handling procedures discussed in version A except the following:
   a. The initial writing on edited tape.
   b. The use of lane-parallel operations.
   c. Storage dumps to be used for Restarts and Break-In.

This version occupies the smallest number of memory locations. It requires the first 415 words of memory. (**SET** $\geq$ 0415, initially.)

4. The D version provides all of the tape-handling procedures discussed in version A except the following:
   a. The use of lane-parallel operations.
   b. Storage dumps to be used for Restarts and Break-In.

   This version requires the first 449 words of memory. (**SET** $\geq$ 0449, initially.)

NOTE

If for any reason the **BMTR** Load Routine generated by the **BMTR** macro cannot find the correct version of **BMTR** which should have been included in the assembled program, a halt 1248 00 8421 will occur at object program load time.

A further attempt to find the **BMTR** subroutine can be accomplished by depressing the Start key.

If the program is punched onto paper tape during Pass III, no version of **BMTR** will be punched.

How To Use The BMTR Macro

1. If **BMTR** is used, **LOD** P4 of the basic STAR II control instruction framework must have **BMTR** in its Address/Constant field. For example:

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT	REMARKS
22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66 67
	JØHN DØE			NAM		131			
				LØD		START			
				LØD		END 2			
				LØD		INPUT TANK			
				LØD		BMTR			
				LØD		0000000000			
				SET				0542	
				↓					

2. The Location Counter must be set *initially* to a specific value if **BMTR** is used.
3. If version A is used, the Location Counter must be **SET** to 0597 before any instructions or reservations of memory can be written. This will, in effect, reserve the first 597 locations of memory which **BMTR** version A will occupy.
4. If version B is used, the Location Counter must be **SET** to 0542 prior to any instructions or reservations of memory. This will reserve the first 0542 locations of memory, which **BMTR** version B will occupy.
5. If versions C or D are used, normal procedures should be followed. However, since version C occupies the first 415 words of memory and version D occupies the first 449 words of memory, no reservation (**BLK**) of these areas should be made.
6. When version C is used, locations 0415 and following can be used as input/output areas.
7. When version D is used, locations 449 and following can be used for input/output tanks.

The macro's parameters are:

V = The version to be incorporated.
An *A* for version A,
a *B* for version B,
a *C* for version C,
or a *D* for version D must be placed in the Address/Constant field of this parameter, left-justified.

P2 = A storage area to be temporarily used by the **BMTR** Load Routine. This area must be 81 words in length. After the program is loaded, this area of memory can be used for any purpose. Its use is similar in concept to the parameter P3 of **LOD** P3 and can be the same area as that specified by **LOD** P3.

P3 = The location to which control is to be transferred after the object program and **BMTR** have been loaded.

The parameter P3 in the **BMTR** macro corresponds to the parameter P4 of **LOD** P4 when a version of **BMTR** is not used.

Form:

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	REA
22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 65 66
	BMTR			BMTR		V			
						P2			
						P3			

Restrictions: None

Number of Instructions Generated: 17

Example:

PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCRE-MENT	
20 21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 6
	BMTR			BMTR		B			
						CARD INPUT			
						START			

For further information regarding the **BMTR** Routine, see Appendix B.

CLEAR CORE STORAGE – CLST

Description: Clear the consecutive storage locations starting with the location specified by the first symbolic address (P1) and ending with the location specified by the second symbolic address (P2), inclusively.

The macro parameters are:

P1 = The first location to be cleared.
P2 = The last location to be cleared.

The two parameters are symbolic addresses appearing in the Address/Constant field.

This macro may be named if desired.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
8 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		ANY NAME			CLST		P1		
							P2		

Restrictions: The B register and the Repeat indicator are altered. The Overflow indicator is left on.

Number of Instructions Generated: 7

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		CLEAR TABL			CLST		TABLE 4		
							TABLE 5	−	0001

CLEAR CORE STORAGE SELECTIVELY – CLSS

Description: Clear a specific number (NNNN) of locations, starting with the symbolic location specified by the first parameter.

The macro parameters are:

P1 = The first location to be cleared (symbolic address).
NNNN (left-justified) = the number of consecutive locations starting at P1 which are to be cleared, including P1.

This macro may be named, if desired.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62
		ANY NAME			CLSS		P1		
							NNNN		

Restrictions: The B register and the Repeat indicator are altered. The Overflow indicator is left on.

Number of Instructions Generated: 7

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		CLR CØRE S			CLSS		TABLE 6		
							1465		

CREATE BINARY SEARCH ROUTINE – CBSR

Description: This macro utilizes the Binary Search Generator. For further information

about this subroutine refer to Technical Bulletin Number 25.

The **CBSR** macro will produce a Binary Search Routine for a table, the characteristics of which are defined by two parameter words. The coding produced will be placed in memory starting at the location specified by the symbolic entry in the Name field (Any Name). Upon execution of the generated coding a Search routine will be created; it will not search the table.

In order to Search a table for a particular item, execute this macro at a prior point in time, then:

1. Load the R register with the argument (left-justified).
2. Store the P register in location aaaa. (aaaa is the first location of the generated routine — in this case Any Name and is the exit for the subroutine.)
3. **BUN** to Any Name + 2. For example:

```
LDR     Argument
STP     Any Name
BUN     Any Name + 2
```

After the Search is completed, the routine will return control back to the program on the line following the **BUN** instruction mentioned in 3 above. A and R registers will remain undisturbed. The B register will contain the location of the argument which was found. The Comparison indicator will be set to either Equal or Low.

1. If Equal, the argument was found.
2. If Low, it signifies that the B register contains the address of the next sequentially higher argument (assuming an ascending order Search).

If there are two or more equal arguments in the table, the B register will contain the address of the first one. When arguments are greater than one word, the routine is adjusted according to the specifications. See pages 28-30 of Technical Bulletin No. 25.

If the signs of all arguments are plus, an ascending order Search will be executed; if minus, there will be a descending order Search. In any event, signs must be either all plus or all minus.

The two parameter words for this macro are:

```
O  WW  YYY  XXXX
D  K   TTTT OOOO
```

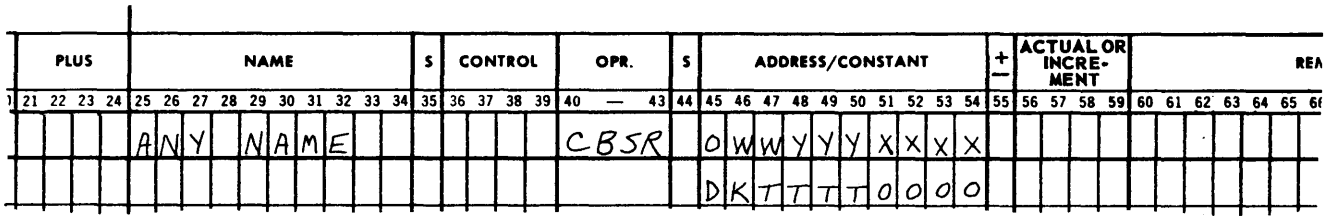
where:

WW = Number of words in each argument. 01 = 1 wd. 00 = 100
YYY = Number of words in each item. 001 = 1 wd. 000 = 1000
XXXX = Number of items in the table, including sentinels.
XXXX must be greater than 0001
D = Number of digits in the argument field of the last argument word.
Do not count signs. The count starts in position SL:11. 0 = 10.
K = 0 or 1. 0 means sign digits of all arguments are included in the Search.
1 means sign digits of all arguments are omitted from the Search.

NOTE
Signs of all arguments must be the same.

TTTT = Address of the first location following the end of the table.

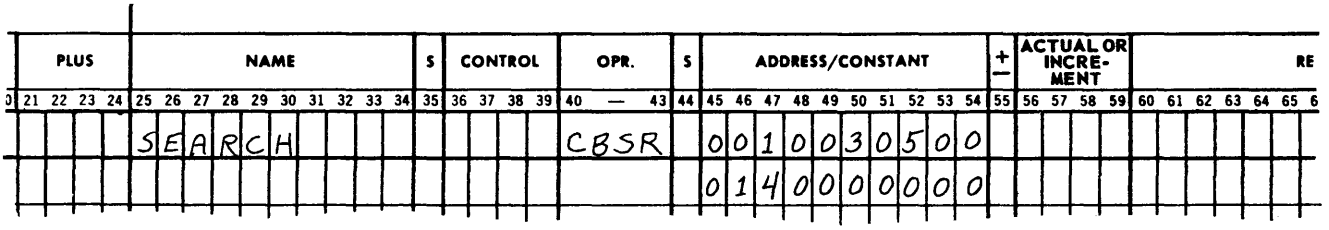
Form:



Restrictions: During execution of the search, the B register and the Comparison indicators are adjusted.

Number of Instructions Generated: 44

Example:



Programer Modification

The programer is able to search more than one table with a given generated routine, provided that the only difference among the like tables is the location reference TTTT.

To accomplish this: Store the address of the last argument in digit positions SL:04 of Search + 1 before executing a search.

For example — Assumptions: That tables A and B are alike; that the last argument of table A is in location 0400; that the last argument of table B is in location 0600; and that the search routine begins in the symbolic location Search.

CONSTANT

CASE 1.
```
     CAD   (0 — 0400)
041  STA   Search + 1
     LDR   aaaa
     STP   Search
     BUN   Search + 2
```
Searching Table A with one routine.

CONSTANT

CASE 2.
```
     CAD   (0 — 0600)
041  STA   Search + 1
     LDR   aaaa
     STP   Search
     BUN   Search + 2
```
Searching Table B with the same generated routine.

DATE DIFFERENCE — **DADI**

Description: The number of days between the date, located in the Partial field of the symbolic location specified by the parameter P1, and the date, located in the Partial field of the symbolic location specified by the parameter P2, will be placed in the Partial field SL:04 of rA.

The macro parameters are:

P1 — The symbolic location of the earlier date in time.
P2 — The symbolic location of the later date in time.
SLSL — The partial-word designators of P1 and P2. SLSL is left-justified.

P1 must be the earlier date of the two. Both dates must be of the form YYDDD.

This macro may be named if desired.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		ANY NAME			DADI		P1		
							P2		
							SLSL		

Restrictions: rA, rR and the Comparison indicator are altered.

Number of Instructions Generated: 49

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		DIFF IN DA			DADI		CREATION D		
							PRESENT D		
							6505		

DATE TRANSLATE DAY — **DADA**

Description: Translate the date stored in the Partial-Word field of the symbolic location specified by the parameter P1, which is stated in the form YYDDD (year-day), to a date stated in the form of MMDDYY (month-day-year). Store the translated date in the Partial-Word field of the symbolic location specified by the parameter P2.

The macro parameters are:

P1 — The symbolic location of the date stated in the form YYDDD [where DDD (day of the year) = 001 to 366].

P2 — The symbolic location in which the translated date of the form MMDDYY (month-day-year) is to be stored.

SLSL — The partial-word designators of P1 and P2. SLSL is left-justified.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62
		ANY NAME			DADA		P1		
							P2		
							SLSL		

Restrictions: rA, rB, rR, Comparison indicators and Overflow indicators are altered.

Number of Instructions Generated: 53

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		MØ DA YR			DADA		YEAR DAY		
							MØ DAY YR		
							5506		

DATE TRANSLATE MONTH – DAMØ

Description: Translate the date located in the Partial-Word field of the symbolic location specified by the parameter P1, which is stated in the form MMDDYY (month-day-year), to a date stated in the form of YYDDD (year-day, where DDD = 001 to 366). Place the translated date in the Partial-Word field of the symbolic location specified by the parameter P2. For programer modification purposes, the exit from this routine is located at Any Name + 14.

The macro parameters are:

P1 = Symbolic address of the date in MMDDYY form.
P2 = Symbolic address where the new date in YYDDD form is to be stored.
SLSL = The partial-word designation of P1 and P2 respectively. SLSL is left-justified.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		ANY NAME			DAMØ		P1		
							P2		
							SLSL		

Restrictions: rA, rB, rR and the Comparison indicator are altered.

Number of Instructions Generated: 35

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT
20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		TØ YR DAY			DAMØ		DATE YYDM		
							DATE YRDA		
							6605		

DELETE ITEM – DELE

Description: This marco will delete an item from a table. The location of the item to be deleted must be placed in the B register. Its size must be stated by the parameter NNN. The upper limit of the table which is the location of the first argument must be zeros. This upper limit is designated by the symbolic address of the parameter P1. The lower limit of the table which is the location of the last argument must be nines. The lower limit is specified by the symbolic address of the parameter P2.

A Halt will be executed if there is an attempt to delete the first or last item of the table.

After an item is deleted all subsequent items are moved toward the beginning of the table. This will add another sentinel containing an argument of nines to the bottom of the table.

This macro may be named, if desired.

The macro's parameters are:

P1 = Symbolic address of the argument (first word) of the first item in the table.
P2 = Symbolic address of the argument (first word) of the last item in the table.
NNN = Number of words per item. NNN must be *right*-justified and can range from 001 to 100.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
		ANY NAME			DELE		P1		
							P2		
							NNN		

Additional Requirements:

1. Register B must contain the location of the argument (or first word) of the item to be deleted.
2. Do not attempt to delete the first item (P1) ; this item must be the zero argument.
3. Do not attempt to delete the last item (P2) ; this item must be the nines argument.

Restrictions: rB, and rA, and the Comparison indicator are altered.

Number of Instructions Generated: 15

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
		DELET ITEM			DELE		TABLE 5		
							TABLE 5		0883
							0000000098		

FIELD SUM – **FLDS**

Description: Sum the defined Partial-Word fields of the memory locations beginning with the symbolic address of the first parameter (P1) and ending with the symbolic address of the second parameter (P2). Store the sum in the corresponding Partial-Word field of the symbolic address of the fourth parameter (P4). Ignore Overflow. The macro may be named, if desired.

The macro's parameters are:

P1 = The symbolic address of the first location to be summed.
P2 = The symbolic address of the last location to be summed.
SL = The field to be summed. SL is left-justified.
P4 = The location in which the sum is to be stored. The sum will be stored in the Partial-Word field specified by SL.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
		ANY NAME			FLDS		P1		
							P2		
							SL		
							P4		

Restrictions: rA, rB and the Overflow indicator are altered.

Number of Instructions Generated: 9

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 6
		FIELD SUM			FLDS		PAYRØLL TBL		
							TABLE END		
							07		
							DEPART TØT		

FIRST TIME NOP – **FNØP**

Description: The first time this area of the program is used, execute an NOP. Thereafter, every time this point is reached, transfer control to P1.

The macro's parameter is:

P1 – The symbolic address to which control is to be transferred on all occasions except the first.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 6
		ANY NAME			FNØP		P1		

Restrictions: None

Number of Instructions Generated: 2

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		INITIAL SW			FNØP		PGM START		

FIRST TIME TRANSFER – **FTTR**

Description: Execute a **BUN** and change control to the symbolic location P1, when this portion of the program is encountered for the first time. Thereafter, when this section of the program is used, execute a **NOP** and continue in sequence.

The macro's parameter is:

P1 – The symbolic address to which control is transferred the first time this set of coding is used.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		ANY NAME			FTTR		P1		

Restrictions: None

Number of Instructions Generated: 3

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT
9 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		INITIALIZE			FTTR		CLEAR TØTL		

GROUP CHANGE – **GPCH**

Description: The first time that the routine generated by this macro is encountered, the partial field of the word found in the memory location indicated by the first parameter will be stored. Each succeeding time that this coding is used, the partial field of the word specified by the first parameter (P1) will be compared to the argument which was initially stored. If the contents of the location represented by the partial field of parameter one (P1) are found to be different from those of the stored argument, control will be transferred to the symbolic location specified by the third parameter (P3). Otherwise, control will continue in sequence.

When a change in the argument has been detected, the partial-word contents of the new item are stored in the Partial-Word field of the symbolic address specified by the first parameter (P1).

This macro facilitates the processing of definable groups of data such as Sales Statistics By Day, etc.

This macro's parameters are:

P1 — The location to be checked for continuity of the group.
SL — Designates the Partial-Word field of parameter 1 to be used. SL is left-justified.
P3 — The location to which control is to be transferred if there is a change in the group.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63
		ANY NAME			GPCH		P1		
							SL		
							P3		

Restrictions: rA and the Comparison indicator are altered.

Number of Instructions Generated: 10

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59 60 61 62 63 64
		GRØUP RØUT			GPCH		ARGUMENT		
							08		
							PRINT TØTL		

HALT AND TRANSFER – **HATR**

Description: Print a message on the Supervisory Printer and then Halt. The Halt will be identified by an index number. If the Start button is depressed, control will be transferred to the symbolic location specified by the last parameter (P2).

This macro's parameters are:

P1 = The symbolic address of the first word of the message to be printed on the **SPO.**
NN = The number of words in the message, 01 to 100 (00).
IIII = The Halt index.
P2 = The location to which control is to be transferred if the Start button is depressed. If P2 is blank, control will be transferred back to the Halt instruction.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64 6
		ANY NAME			HATR		P1			
							NNIIII			
							P2			

Restrictions: None

Number of Instructions Generated: 3

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 6
		END ØF JØB			HATR		EØJ MESSAG			
							069669			
							EXECUTIVR			

HASH TOTAL – **HASH**

Description: Take a Hash Total of all locations from the symbolic location designated by the first parameter (P1) to the symbolic location specified by the second parameter (P2). The results of the Hash Total are stored in the symbolic location indicated by the third parameter (P3).

At the completion of the routine, the coding will be reset to allow for its execution an indefinite number of times.

This macro's parameters are:

P1 = The symbolic address of the first location to be hashed.
P2 = The symbolic address of the last location to be hashed.
P3 = The symbolic address of the location in which the Hash Total is to be stored. This address cannot lie between P1 and P2 inclusively.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT
		ANY NAME			HASH		P1		
							P2		
							P3		

Restrictions: rA, rB, the Repeat indicator and the Overflow indicator are altered.

Number of Instructions Generated: 13

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT
		CUST. HASH			HASH		TAPE INPUT		
							END ØF TNK		
							VALID CK		

INSERT ITEM — **INIT**

Description: This macro deals with a table; the format is as follows.*

	SYMBOLIC ADDRESS	MEMORY LOCATION	ILLUSTRATION	ARGUMENT
First argument must be zero	P1	0100		0.....0
		0101		0.....04
		0102		0.....08
		0103		0.....09
		0104		0.....012
Arguments		0105		0.....014
		0106		0.....017
Insert rB = 0107		0107		0.....021
		0108		0.....0436
		0109		0.....0527
The end of the table must be filled with 9's.		0110		9.....9
		0111		9.....9
		0112		9.....9
		0113	a	9.....9
(undisturbed) Last argument must be 9's.	P2	0114	Same	9.....9

The Insert macro will move items toward the end of the table, starting with the third item from the bottom (See Illustration a, above).

All items are moved the space of one table entry, until the memory location or locations, into which the insert must be placed have been moved (e.g. location 0107 above). When this has been accomplished, the table insert, designated by the fifth parameter (P5), is moved into that area of the table.

The last item in the table, the 9's argument, must remain undisturbed.

The macro parameters are:

P1 = Symbolic address of the first location of the table — the zero argument. (Location 0100 in the example above.)

* This is the same type of table with which the Delete macro deals.

P2 = Symbolic address of the last table argument – the 9's argument. (Location 0114 in the example above.)

P3 = Symbolic address of the first location of a temporary storage area which must be NNN words long. This area cannot be in the table area P1 to P2 + NNN.

NNN = Number of words per item, right justified.

P5 = Symbolic Address of the first word of the item to be inserted in the table.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ –	ACTUAL OR INCRE-MENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
		AMY NAME			INIT		P1			
							P2			
							P3			
							0 0 0 0 0 0 0 N N N			
							P5			

Other Requirements:

1. The B register must contain the memory location of the first word (argument word) of the table area into which the insert will be placed.
2. Do not attempt to insert an item in the first item area (P1), this will produce a Halt. The first item must be the zero argument.
3. Do not attempt to insert an item in the next-to-last item area (location 0113), this will produce a Halt. The reason for this is to prevent any items from being moved into the 9's argument area at the end of the table.
4. The temporary storage area (P3) must not fall within the range of P1 to P2 + NNN. This is the table area. This rule also applies to P5.
5. P3 and P5 cannot be coincident with each other.

Restrictions: rB, rA and the Comparison indicator are altered.

Number of Instructions Generated: 36

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ –	ACTUAL OR INCRE-MENT	
3 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63
		INSERT NØ.			INIT		CUST TABLE			
							END TABLE			
							TEMPØRARY			
							0 0 0 0 0 0 0 0 0 1			
							INSERT A			

LOOP – **LØØP**

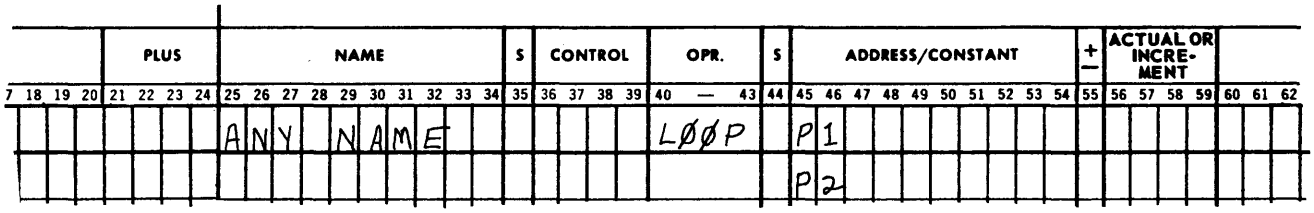
Description: This macro must be placed just after the last step of a section of coding which is to be repeated a number of times successively. It writes the required looping instructions and keeps the necessary tally of the number of times the routine was executed. When the required number of loops have been completed as indicated by the second parameter (P2), the control will continue in sequence.

This macro's parameters are:

P1 = Symbolic address of the first instruction of the **LØØP** to be formed.

P2 = The number of times the **LØØP** is to be executed. If P2 = 0, the **LØØP** will be formed, but no tally will be kept. In this case the programer must make his own exit. After the **LØØP** has been executed P2 times, control will continue in sequence. P2 is right-justified.

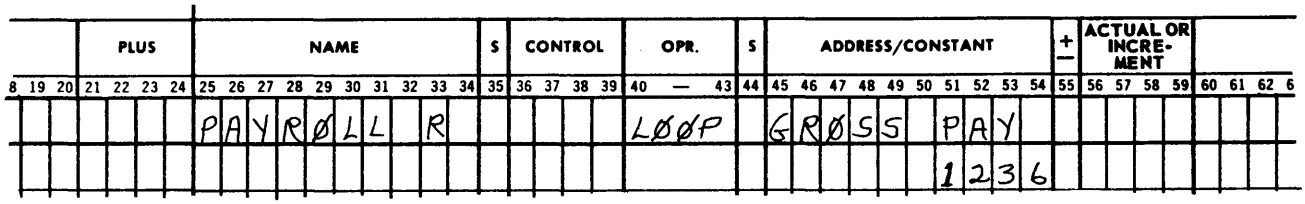
Form:



Restrictions: The Repeat indicator is set each time the **LØØP** is executed.

Number of Instructions Generated: 4

Example:



LOOP RESET – **LØPR**

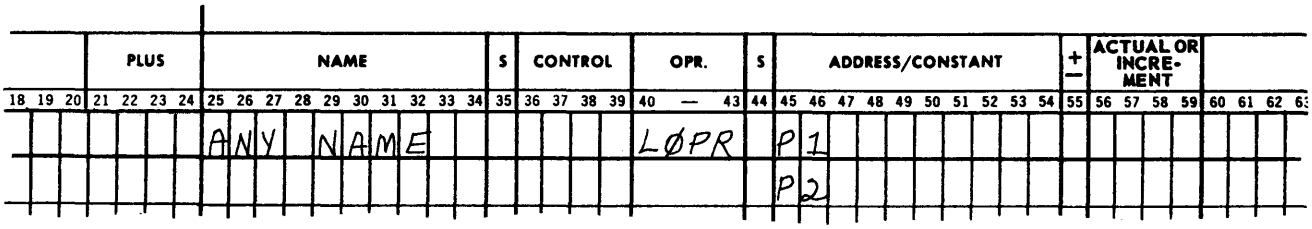
Description: This macro must be placed just after the last step of a section of coding which is to be repeated a number of times, successively. It writes the required looping instructions and keeps the necessary tally of the number of times the routine was executed. Upon completion of the required number of iterations as stated by the second parameter (P2), the looping tally (P2) will be reset to its original state and control will continue in sequence.

The looping tally will be reset to its original setting so that when control is again given to this set of generated coding, it will perform the looping operation the same number of times.

The macro's parameters are:

P1 = Symbolic address of the first instruction in the loop.
P2 = The number of times the loop is to be executed.

Form:



Restrictions: rA is altered and the Repeat indicator is turned off when control continues in sequence.

Number of Instructions Generated: 7

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
		DEDUCTIONS			LØPR		FICA TAX			
							2013			

MOVE – **MØVE**

Description: A specified number of memory locations will be moved from consecutive locations, starting with the cell specified by the symbolic address of parameter one (P1). These contents will be placed into consecutive locations beginning with the cell specified by the symbolic address of parameter two (P2).

The macro parameters are:

P1 = Symbolic address of the first word to be moved.
P2 = Symbolic address of the first word of the area to which the words are to be moved.
NNNN = Number of consecutive words, starting at P1, which are to be moved to the area starting at P2. NNNN is right justified.

Form:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
		ANY NAME			MØVE		P1			
							P2			
							0000000NNNN			

Restrictions: rA, rR and rB are altered, plus the Repeat indicator is off.

Number of Instructions Generated: 10

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
8 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63 64
		TØ ØUTPUTA			MØVE		WØRK AREA			
							ØUTPUT A			
							0000000073			

OVERLAY – **ØVLY**

Overlay is a coding technique normally employed when a program exceeds the size of a computer's memory. It allows a programer to use a particular area of memory in the same program for two or more different sets of instructions. When the program is running it will realize, at various times, that it must "call in," from magnetic tape, routines to perform particular functions. It will load these routines on top of instructions which were formerly a part of the basic program. This Overlay will later be executed and, if necessary, "overlayed" by another routine, or perhaps by a portion of the basic program which it previously replaced. A program (coded in STAR II) may Overlay many sections of its basic routine or one section of the basic routine many times.

In order to use Overlay techniques three things are necessary, namely:

1. The basic program, which will occupy all or most of memory.
2. A Load Routine located within the basic program, the function of which is to search the program tape for the proper Overlay and to load that Overlay into the designated section

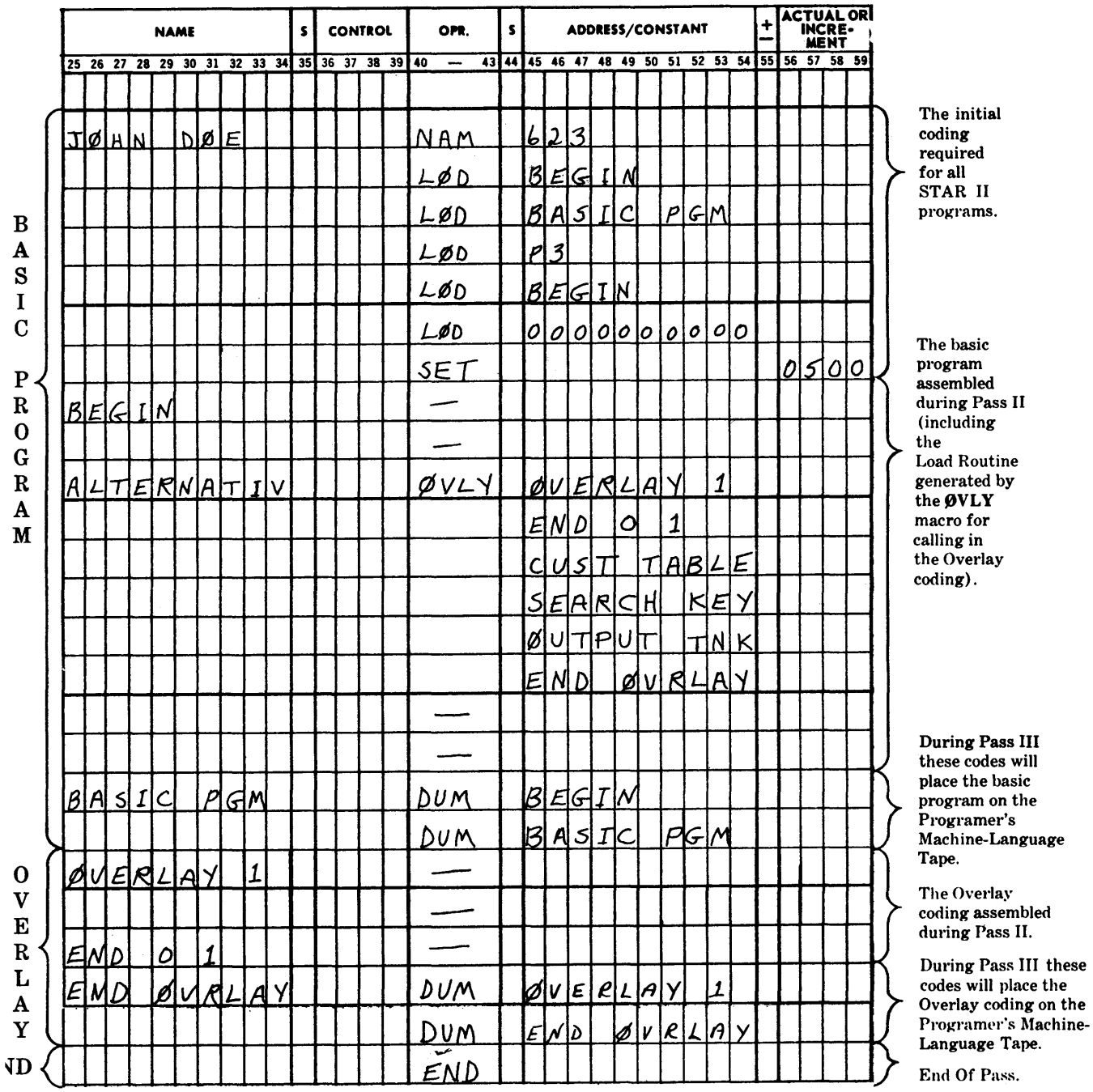
of memory, when the basic program requires it.

3. The coding of the Overlay, itself, which will appear on the program tape.

When the STAR II Assembler is used, a pair of **DUM's** will place the basic program on the Programer's Machine-Language Tape as mentioned earlier. This fulfills the first requirement for the use of Overlays.

Within this basic program, if the Overlay technique is used, a Load Routine must be incorporated for locating the Overlay coding on the program tape and calling it into memory, thus replacing the particular portion of the basic program specified. The Load Routine is provided automatically by STAR II through the use of the **ØVLY** macro. This macro will generate the coding for the required Load Routine and place this coding in the proper area of the basic program, fulfilling the second requirement.

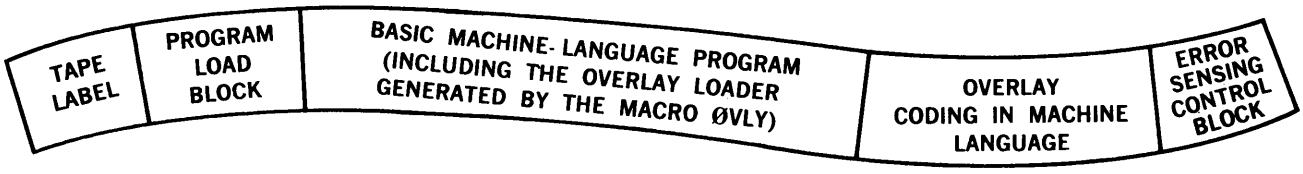
Before discussing the **ØVLY** macro in detail, it is necessary to show how the instructions comprising the Overlay itself are placed on the Programer's Machine-Language Tape in order to fulfill the third requirement. As in the case of the basic program, this is accomplished by the use of a pair of **DUM** instructions. An example will help to clarify this use.



NOTE

All of the parameters expressed by the **ØVLY** macro will not be found in the above illustration. For a complete example of how to code for an Overlay, see the example on page 4-37.

The Programer's Machine-Language Tape would look like the following as a result of the Overlay coded above.



Description: The Overlay macro generates the Load Routine included in the basic program that calls into memory, from magnetic tape, the Overlay coding which will replace a designated portion of the basic program.

The generated Load Routine will search the Programer's Machine-Language Tape for the Overlay coding, using the Search key found in the symbolic location designated by the fourth parameter (P4). When the proper Overlay coding is located, the Loader will read one block at a time into memory starting at the symbolic location specified by the fifth parameter (P5). The coding is then transferred to consecutive memory locations beginning with the location defined by the symbolic name of the third parameter (P3).

When the Overlay coding has been completely loaded, control will be returned to the instruction following the **ØVLY** macro.

The macro's parameters are:

P1 = The symbolic name of the first instruction in the Overlay coding.
P2 = The symbolic name of the last instruction in the Overlay coding.
P3 = The symbolic name of the first instruction which will be overlaid.
P4 = The symbolic name of a storage location where the Search key for locating the Overlay coding on the Programer's Machine-Language Tape is found.
P5 = The symbolic name of the first location of some area that can be used temporarily as an input area for the Overlay coding. This area must be at least 81 words in size.
P6 = The symbolic name of the first **DUM** instruction of the pair associated with this Overlay.

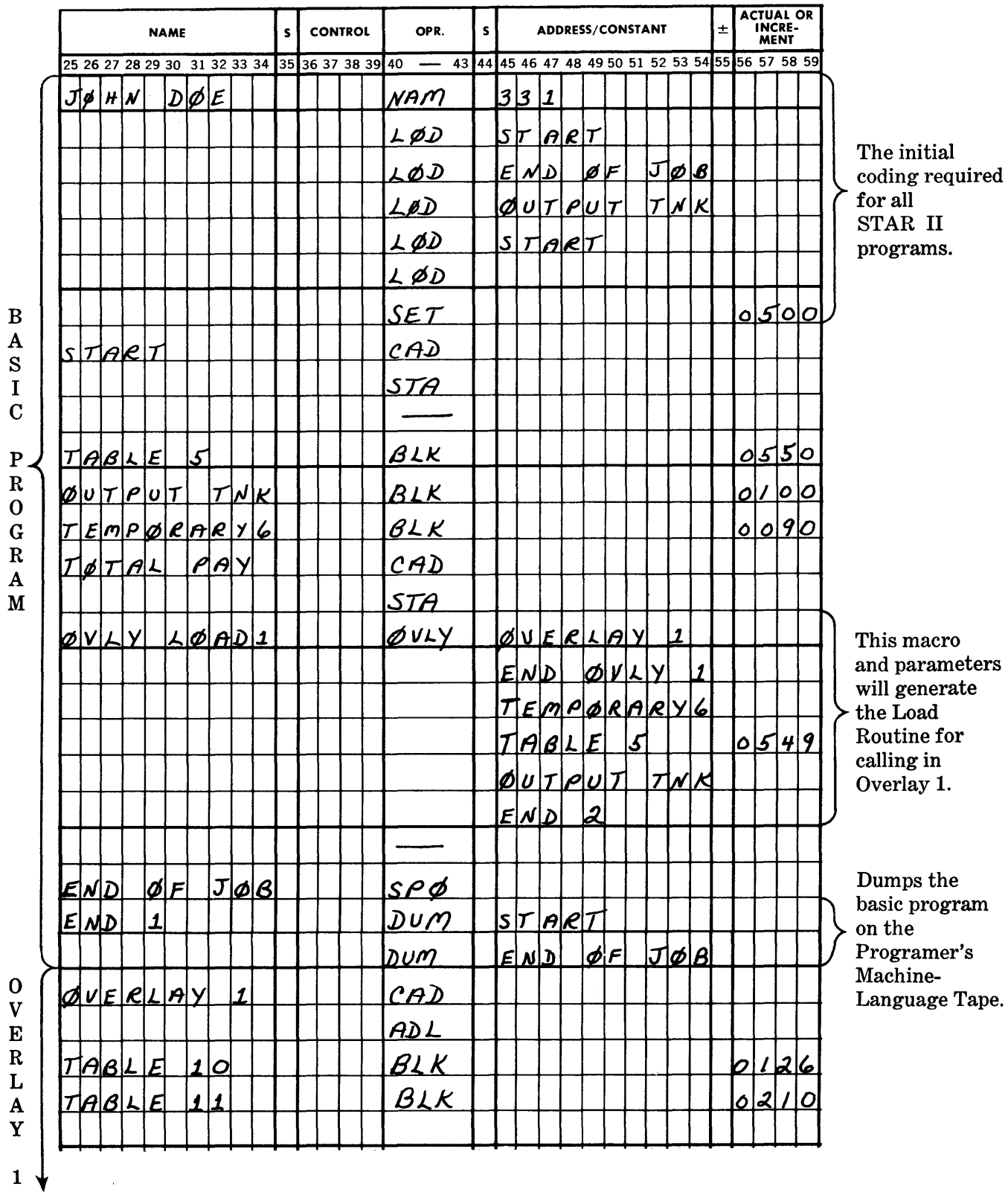
Form:

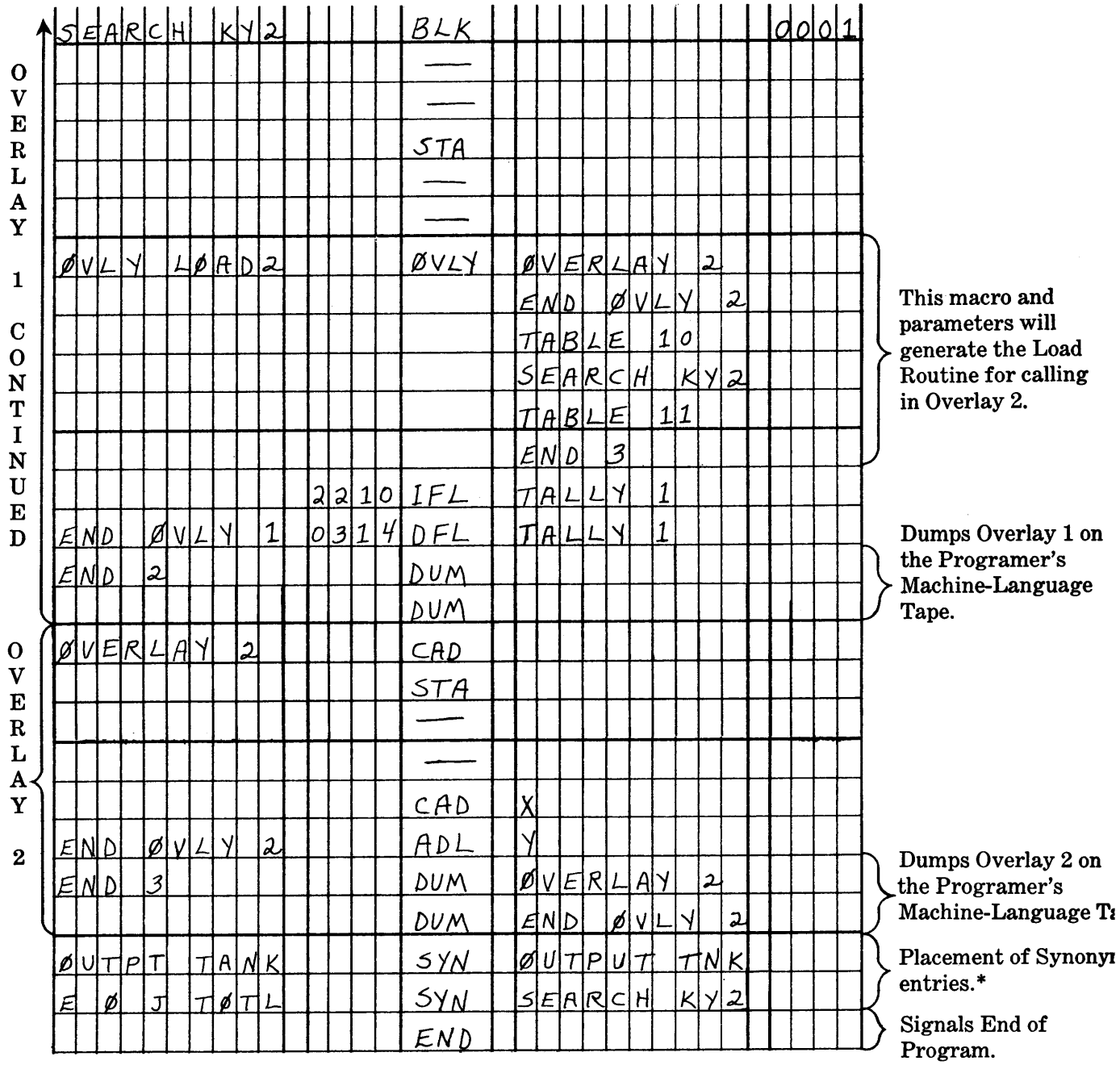
	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62
		ANY NAME			ØVLY		P1			
							P2			
							P3			
							P4			
							P5			
							P6			

Restrictions: rB and the Repeat indicator are altered. The 81 words starting at the location specified by the fifth parameter are altered.

Number of Instructions Generated: 15

Example: How to code for an OVERLAY.



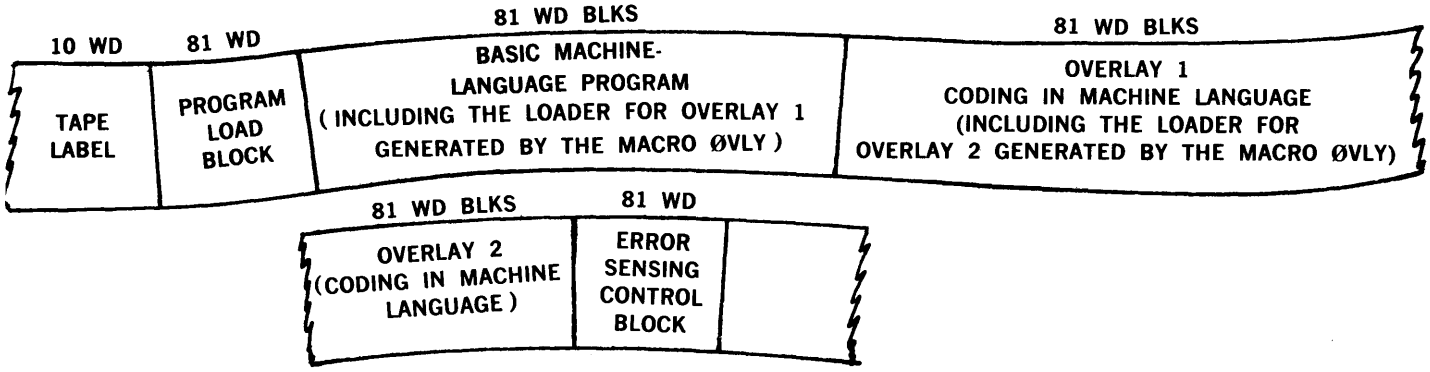


*These SYN's do not have anything to do with the Overlays as such. They are presented merely as an illustration and an attempt to present the example realistically.

A few rules which will assist the programer in the use of the Overlay macro are:

1. It is a good idea to leave all of the Search keys in memory at all times.
2. Memory locations must be reserved for the Search keys by a **BLK** control instruction. A constant *cannot* be used for this purpose as it will cause the Search keys to be lost during the third pass.
3. Since control is transferred to the location immediately following the coding generated by the macro (after the Overlay is called into memory) a **BUN** will normally be placed immediately following the generated coding to change control to the desired routine.
4. Each pair of **DUM** control instructions should be followed by a **SET** (Symbolic Address) control instruction.
5. The first and second parameters of the Overlay macro should agree with the names used in the Address/Constant field of the **DUM** pair. Also, the last parameter of the **ØVLY** should agree with the name of the **DUM** pair.
6. No more than 25 Overlays can be used in any one program.

The Programer's Machine-Language Tape will look like the following as a result of the above example:



READ-WRITE CHECK – RWCK

Description: This macro is used to read a block just written on magnetic tape and to check that block for complete accuracy. This checking is performed by means of a Hash Total, which is derived by adding all the words of a block together, while ignoring Overflow, except for the preface word and the last word of the block. This Hash Total, which was previously computed, is contained in the block itself. It normally appears in the last word of the block and is the reason why the last word is excluded from the Hash Total.

If the two Hash Totals agree, control of the program will continue in sequence. However, if these two Hash Totals do not agree, control will be transferred to the symbolic address specified by the fourth parameter (P4).

In either event, the tape file which was checked will be positioned to the same place after the coding generated by the macro has been executed, as it was prior to its execution. (The read-write head is positioned ready to write the next block.)

This macro must be executed immediately after the block which is to be checked has been written out on tape.

The macro parameters are:

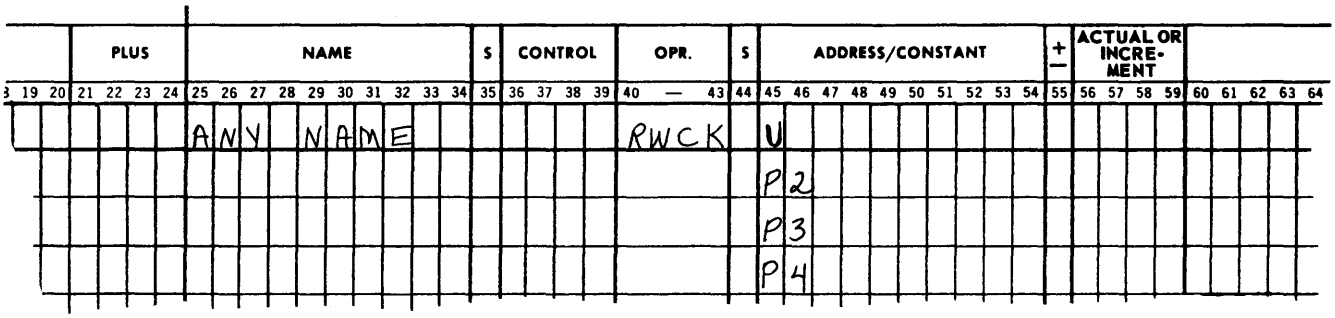
U = Unit designation (left-justified).
P2 = The symbolic address of the first location of an area into which the block just written out on magnetic tape can be read. This area must be large enough to

hold the block to be checked plus its preface word (e.g. block size + 1), since the checking routine uses an **MRR** instruction for reading from magnetic tape.

P3 = The symbolic address of the Hash Total against which the Hash Total developed from the block after it is read will be checked. This Hash Total should be computed and stored prior to the writing of the block on magnetic tape.

P4 = Symbolic address of the location to which control will be transferred if the Hash Totals are not equal.

Form:



Restrictions: rA, rB, Repeat indicator, Overflow indicator, and the Comparison indicators are altered.

Number of Instructions Generated: 19

RESTORE INDICATORS – **REIN**

This macro will be discussed immediately following the explanation of the Store Indicators **(STIN)** macro

RESTORE REGISTERS – **RERG**

This macro will be discussed immediately following the explanation of the Store Registers **(STRG)** macro.

SET SWITCH – **(SSWH)**

This macro will be discussed immediately following the explanation of the Switch **(SWCH)** macro.

SEQUENCE CHECK – **SEQC**

Description: The first time that the coding generated by this macro is executed, store the contents of the symbolic location defined by the first parameter (P1) in a temporary storage area. Each succeeding time that this generated routine is executed, the argument previously placed in temporary storage must be compared to the new argument now appearing in the location defined by the symbolic address of parameter one (P1).

If the new argument is equal to or greater than the previous argument which was placed in temporary storage, program control will continue in sequence and the new argument will then be stored in the temporary storage area.

If, however, the new argument is less than the previously stored argument, control will be transferred to the location specified by the symbolic address of the third parameter (P3) and the new argument will not be stored in the temporary storage area.

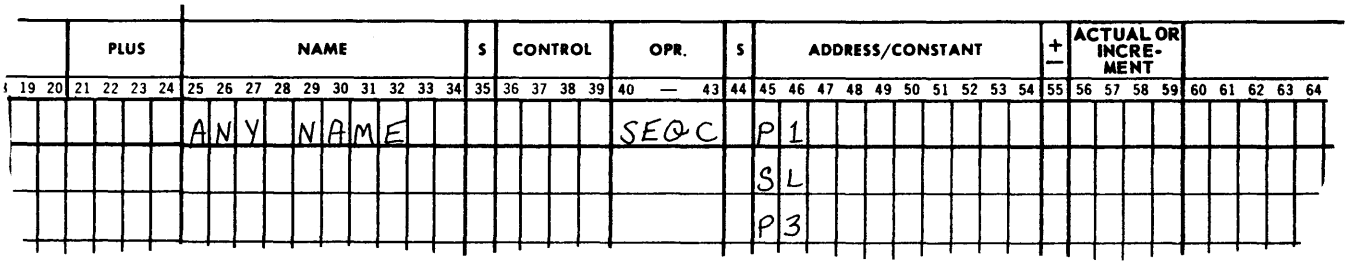
The macro parameters are:

P1 = The symbolic name of the location in which the new argument to be checked for sequence is located.

SL = The Partial field of each argument which is to be checked for proper sequence. (Left-justified.)

P3 = The symbolic location to which control will be transferred if the new argument is out of sequence.

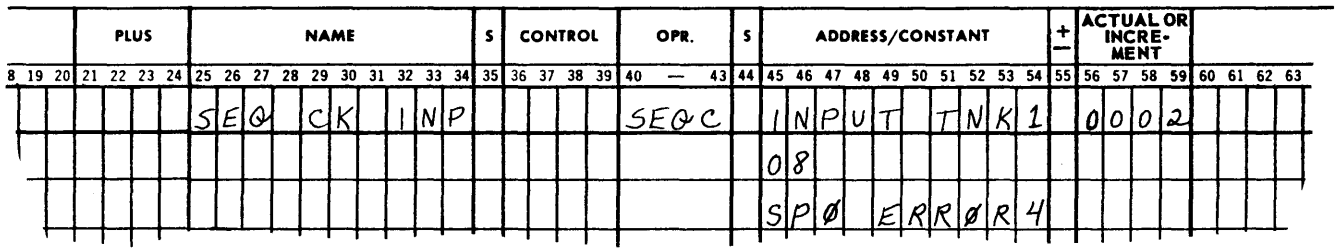
Form:



Restrictions: rA and the Comparison indicators are altered.

Number of Instructions Generated: 10

Example:



SEQUENCE TABLE – SETA

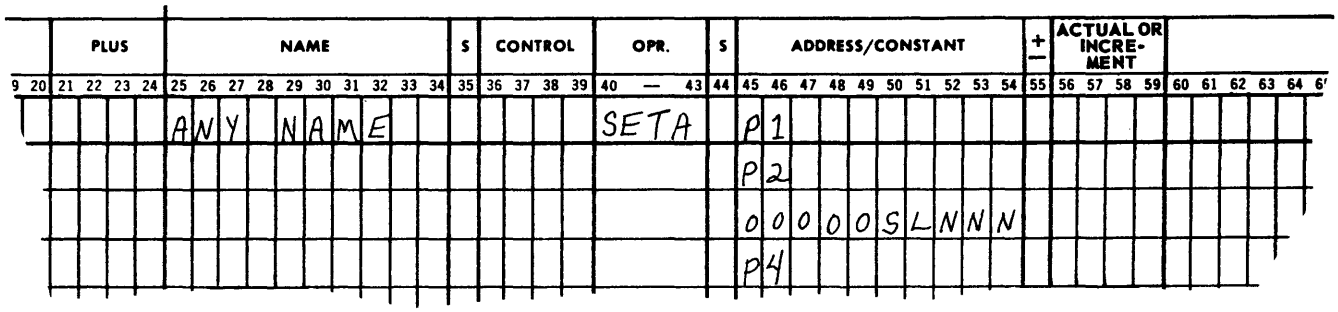
Description: This macro will check the first word of every item in a table (field specified by SL) for ascending sequence. Equal keys are assumed to be in order. If the table is found to be in sequential order, program control continues in sequence.

If the table is out of sequence, control will be transferred to the location specified by the symbolic address of the fourth parameter (P4). The four low-order digits (SL:04) of Any Name plus 2 will contain the address of the item encountered which was out of sequence.

The macro parameters are:

P1 = The symbolic address of the beginning of the table.
P2 = The symbolic address of the last location of the table.
SL = The field to be examined for sequence.
NNN = The number of words per item (001 = 1 word).
P4 = The symbolic address to which control will be transferred if the table is out of sequence.

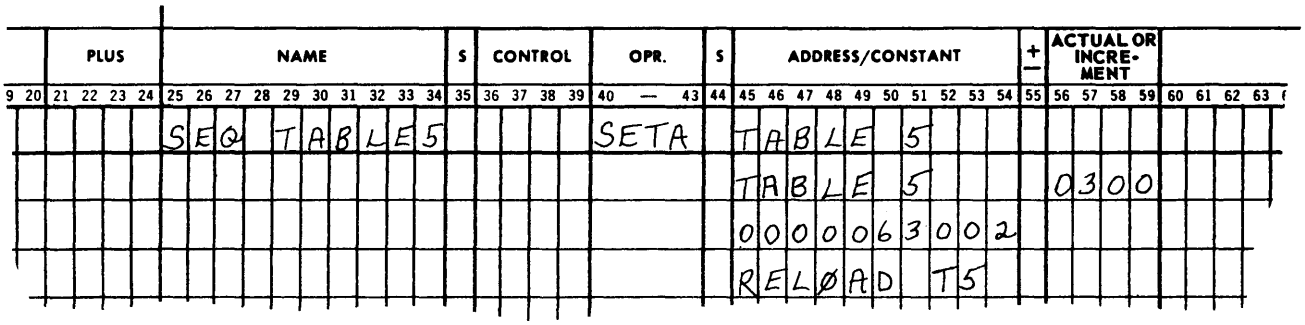
Form:



Restrictions: rA, rR, and the Comparison indicator are altered.

Number of Instructions Generated: 23

Example:



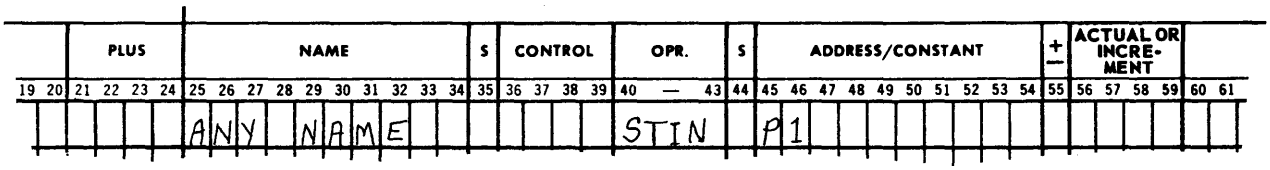
STORE INDICATORS – **STIN**

Description: Store the status of the Comparison indicator, the Overflow indicator and the Repeat indicator in the location specified by the symbolic address of the first parameter.

The macro parameter is:

P1 = The symbolic location where the status of the indicators is to be stored.

Form:



After the execution of the generated routine, the contents of location P1 will have the following appearance: 0000X00Y00Z

X = Comparison indicator status
- 0 = Low
- 1 = Equal
- 2 = High

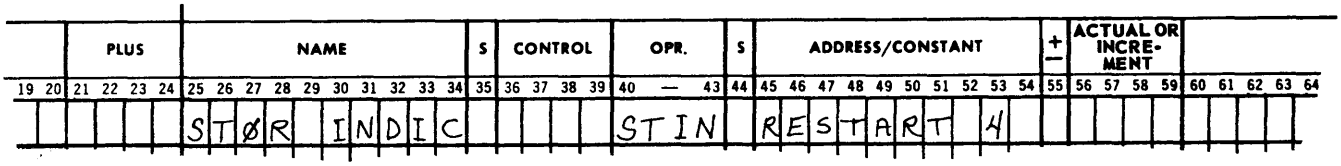
Y = Overflow indicator status
- 0 = Off
- 1 = On

Z = Repeat indicator status
- 0 = Off
- 1 = On

Restrictions: rA and all indicators are altered.

Number of Instructions Generated: 15

Example:



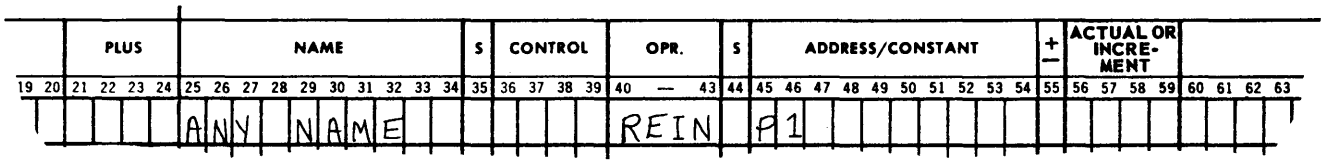
RESTORE INDICATORS – **REIN**

Description: Restore the Comparison indicator, the Overflow indicator and the Repeat indicator to the state to which they were set when they were previously stored by a Store Indicator macro **(STIN).**

The macro parameter is:

P1 = The symbolic location where the indicator's status was previously stored by a Store Indicator macro **(STIN).** The parameter P1 of this macro must equal the parameter P1 of some Store Indicator macro **(STIN).**

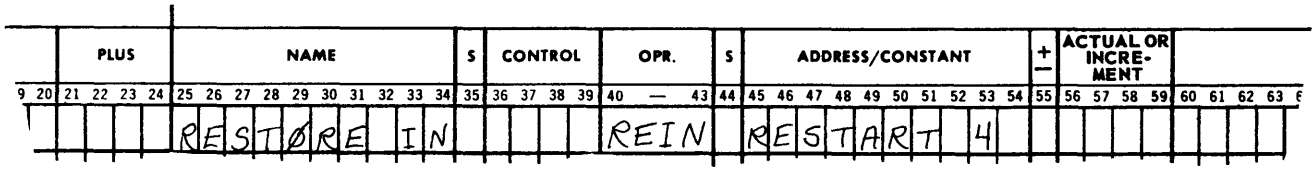
Form:



Restrictions: rR and all indicators are altered.

Number of Instructions Generated: 7

Example:



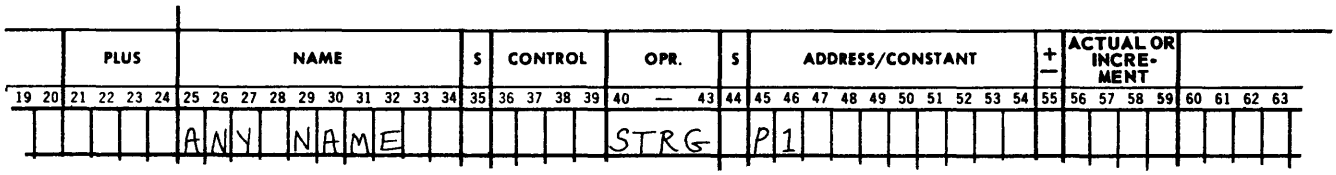
STORE REGISTERS – **STRG**

Description: Store in four consecutive locations beginning with the symbolic location specified by the parameter (P1): rA, rR, rB and rP.

The macro parameter is:

P1 = The symbolic address of the first of four locations in which these registers will be stored.

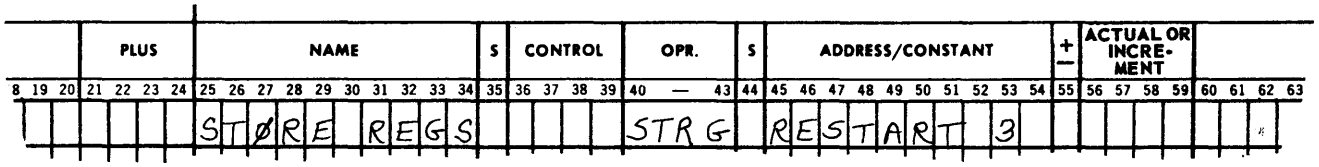
Form:



Restrictions: None

Number of Instructions Generated: 4

Example:



RESTORE REGISTERS – **RERG**

Description: Restore rA, rR and rB to the state to which they were set when they were previously stored by a Store Registers macro **(STRG).** Register P will also be restored if desired.

The macro parameter is:

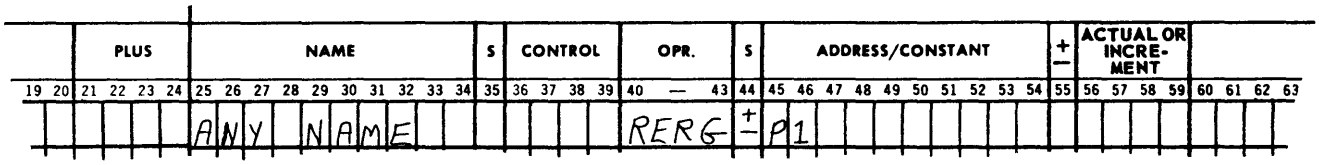
P1 = The symbolic location of the first of four locations in which the registers were

previously stored by a Store Registers macro **(STRG).**

If ± = 0, do not restore rP.

If ± = 1, restore rP.

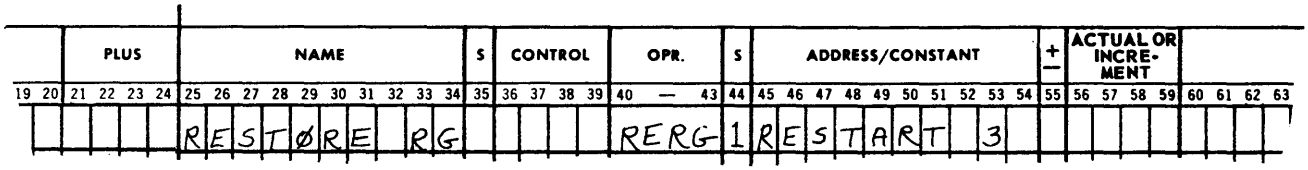
Form:



Restrictions: All registers are altered.

Number of Instructions Generated: 6

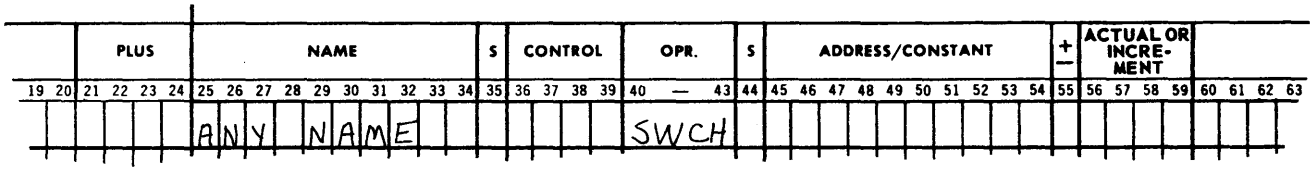
Example:



SWITCH – **SWCH**

Description: No parameters are required for this macro instruction. Its function is to permit the naming of a switch for purposes of identification. A **NOP** instruction is placed in this location. Upon execution control continues in sequence.

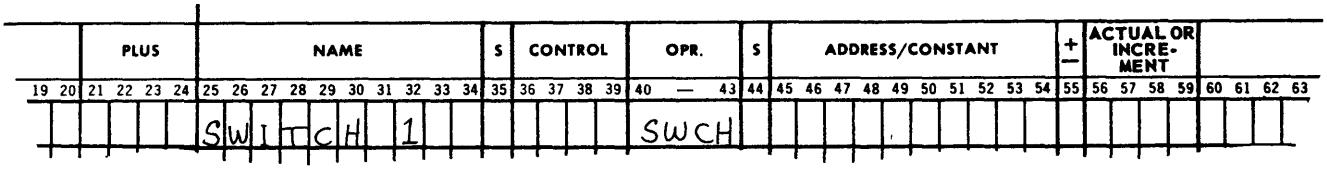
Form:



Restrictions: None

Number of Instructions Generated: 1

Example:



SET SWITCH – **SSWH**

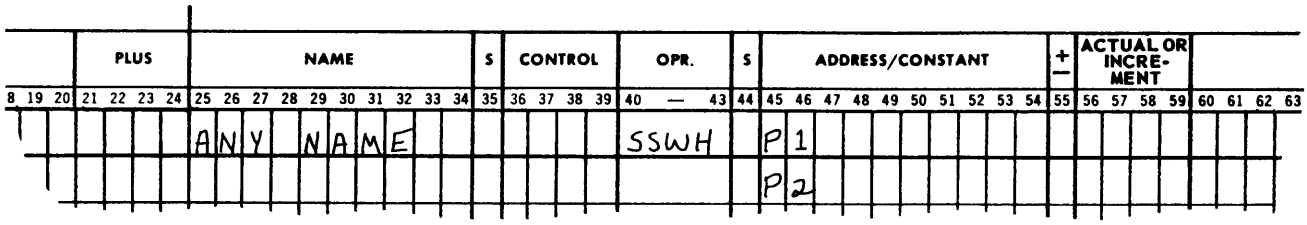
Description: This macro will cause the contents of a switch previously named by a Switch **(SWCH)** macro to be changed from a **NOP** to a **BUN.** The change of control **(BUN)** will be to the symbolic location expressed by the Set Switch **(SSWH)** macro's second parameter.

The macro parameters are:

P1 = This is the symbolic address of a switch which was previously defined by the Switch **(SWCH)** macro.

P2 = This is the symbolic address of the location to which control will be transferred the next time the specified switch is encountered.

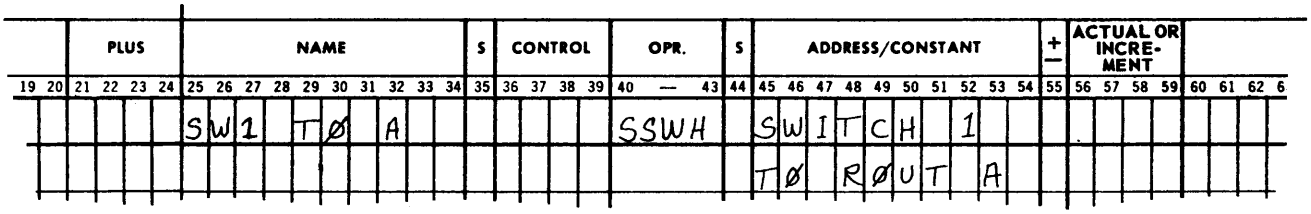
Form:



Restrictions: rA is altered.

Number of Instructions Generated: 4

Example:



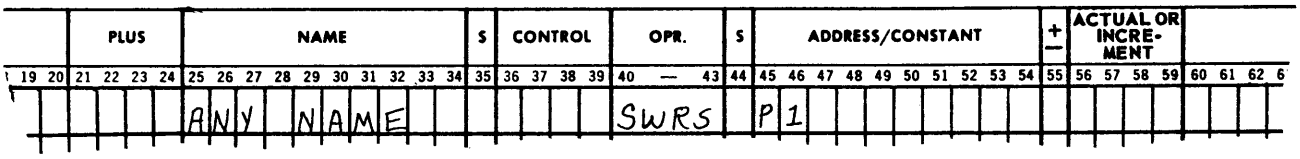
SWITCH RESET – SWRS

Description: Reset (restore to a **NOP**) the switch designated by the symbolic address in the macro's parameter (P1). The parameter will refer to a switch which was previously defined by a Switch macro **(SWCH).**

The macro parameter is:

P1 = The name of the Switch that will be reset by restoring the **NOP** instruction.

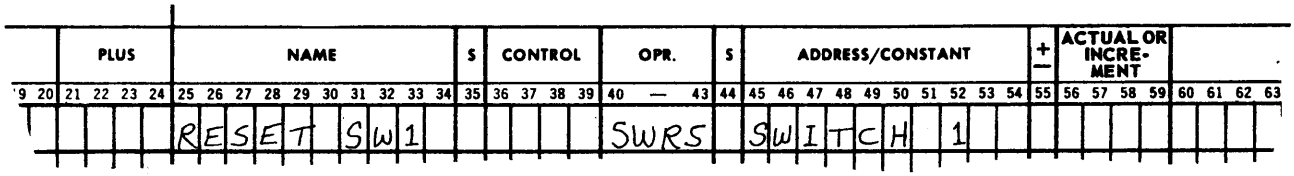
Form:



Restrictions: rA is altered.

Number of Instructions Generated: 3

Example:

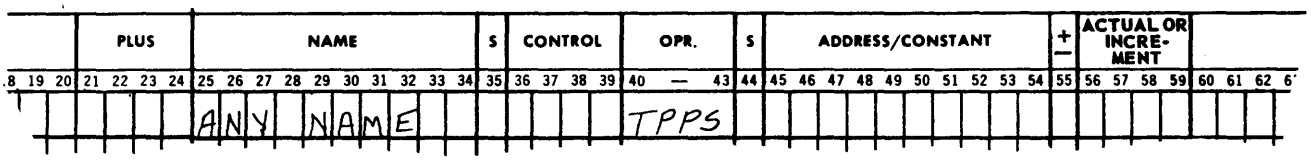


TEST AND PRINT PROGRAM SWITCHES – TPPS

Description: Test all Program Control switches and print the switch numbers which are on. The message printed will read: PCS 0 on, 1 on, etc.

No parameters are required by this macro.

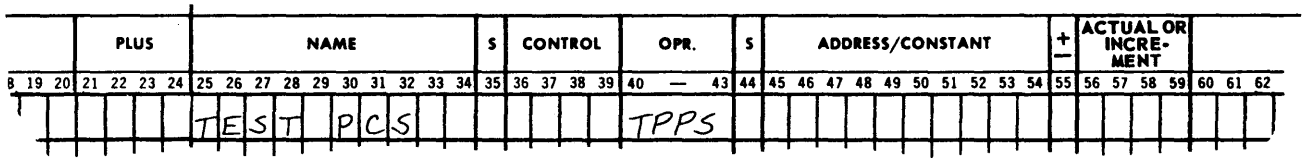
Form:



Restrictions: The Overflow indicator is altered. The Overflow indicator must be off before the generated coding can be used.

Number of Instructions Generated: 10

Example:



VARIABLE HASH – VAHA

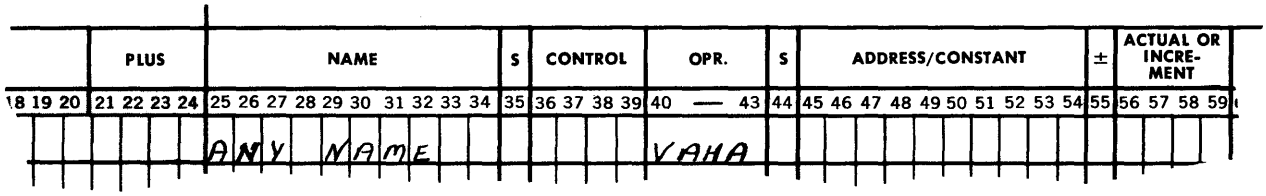
Description: This macro will Hash Total NNN locations, starting with location mmmm. These parameters are not a part of the macro itself, rather, they will be provided by the object program and placed in registers A and R at the proper time. The coding produced by the macro is a closed subroutine. This allows the programer to execute the generated set of coding any number of times and also to change the starting location and/or the number of locations to be hashed.

The Hash Total will *always* be stored in memory following the last location which was hashed.

Before entering this subroutine, place mmmm, the first location of the area to be hashed (absolute machine address) in rA, SL:04 and the number of locations to be hashed NNN, in rR, SL:03.

The programer must also specify the exit from the subroutine. This can be accomplished by an STP Any Name + 10, before a change of control to the Variable Hash subroutine is executed (**BUN** Any Name).

Form:



Restrictions: rA, rB, Overflow, and Repeat indicators are altered.

Number of Instructions Generated: 16

Example:

	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCRE-MENT	
18 19 20	21 22 23 24	25 26 27 28 29 30 31 32 33 34	35	36 37 38 39	40 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	60 61 62 63
		HASH TØTAL			VAHA					

SECTION 5

OPERATING INSTRUCTIONS FOR ASSEMBLY

EQUIPMENT REQUIREMENTS

There are a number of equipment configurations possible. The particular configuration to be used will be determined by the setting of Program Control switches. The use of these switches for the STAR II Program will be explained later.

Configuration A — Card Input, 407 Listing Output.

1. One 293, 087, or 089 Card Reader
2. One 407 Line Printer
3. Four Magnetic Tape Storage units*
4. Supervisory Printer.

Configuration B — Card Input, 407 Listing Output, Paper-Tape Output.

Same requirements as Configuration A plus a Paper-Tape Punch.

Configuration C — Card Input, High Speed Printer Output.

1. One 293, 087, or 089 Card Reader
2. One High Speed Printer
3. Four Magnetic Tape Storage units*
4. Supervisory Printer.

Configuration D — Card Input, High Speed Printer Output, Paper-Tape Output.

Same requirements as Configuration C plus a Paper-Tape Punch.

Configuration E — Card Input, Punched Paper-Tape Output.

1. One 293, 087, or 089 Card Reader
2. Supervisory Printer
3. Four Magnetic Tape Storage units*
4. Paper-Tape Punch.

Configuration F — Paper-Tape Input, 407 Listing Output.

1. One Photoelectric Reader
2. One 407 Line Printer
3. Four Magnetic Tape Storage units*
4. Supervisory Printer.

Configuration G — Paper-Tape Input, 407 Line Printer Listing Output, Paper-Tape Output.

Same requirements as Configuration F plus a Paper-Tape Punch.

Configuration H — Paper-Tape Input, High Speed Printer Output.

1. One Photoelectric Reader
2. One High Speed Printer
3. Four Magnetic Tape Storage units*
4. Supervisory Printer

Configuration I — Paper-Tape Input, High Speed Printer Output, Paper-Tape Output.

Same requirements as Configuration H plus a Paper-Tape Punch.

Configuration J — Paper-Tape Input, Paper-Tape Output.

1. One Photoelectric Reader
2. Supervisory Printer
3. Four Magnetic Tape Storage units*
4. Paper-Tape Punch.

*Three tape units can be used, if necessary.

BILL OF MATERIALS

This is a list of all the items, except for equipment, needed to run the job.

1. PLUGBOARDS
   a. If card input is used, one 293, 087, or 089 (80-80) plugboard is required, selecting format from column one with digit select 1 wired to format select 1 and with digit select 6 wired to format select 6.
   b. If printed output on the 407 Line Printer is used, a 120-120 on-line 407 board is necessary.
   c. If the High Speed Printer is used, a special panel must be used. See Appendix A.
2. INPUT FORMS
   a. If cards are used, the Symbolic-Language Program deck and the STAR II General Purpose Program Load Routine, in card form, are required.
   b. If paper tape is used, the Symbolic-Language Program and the STAR II General Purpose Program Load Routine, in paper-tape form, are required.
3. OUTPUT FORMS
   a. A reel of unpunched paper tape — if the Machine-Language Program is going to be punched into paper tape.
   b. Paper for the 407 Line Printer (at least 12 inches wide), if used.
   c. Paper for the High Speed Printer, if used.
   d. Paper for the Supervisory Printer.
4. MAGNETIC TAPES
   Four reels of tape are required.
   a. The STAR II Program Tape.
   b. Two 21-word preblocked scratch tapes with a 10-word **BMTR** tape label as the first block on each tape.
   c. An 81-word, preblocked scratch tape with a 10-word **BMTR** tape label as the first block on the tape.

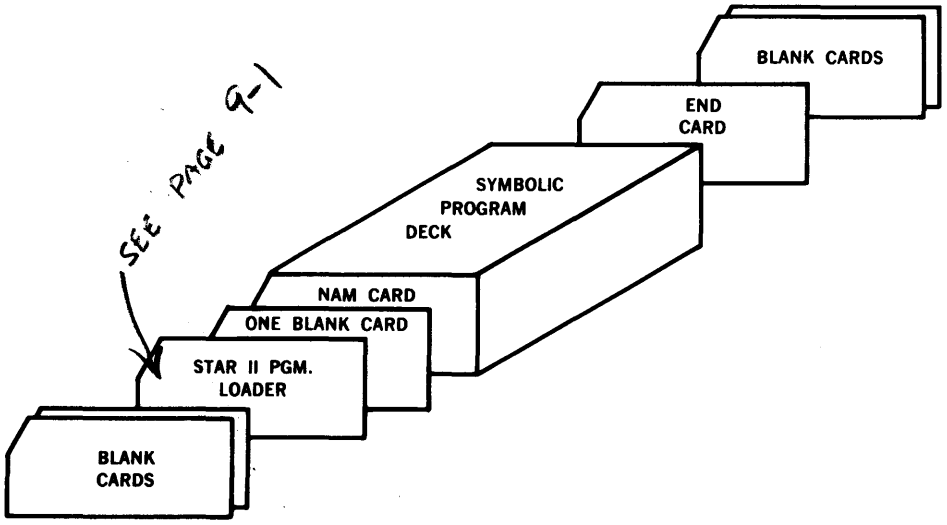
OPERATOR PROCEDURES

Equipment Setup

1. If Card Input is used:
   a. Place the 80-80 panel, described in the Bill of Materials, into the Card Reader.
   b. By following the standard installation procedures, load the symbolic deck into the reader. The sequence of the symbolic deck for assembly should be:

      Blank cards**
      The STAR II General Purpose Program Load Routine.
      The **NAM** card.
      The remainder of the symbolic deck with the **END** card last.
      Blank cards.

**The number of blank cards, if any, will depend upon the type of reader used.



2. If Punched Paper-Tape Input is used:
   a. Designate the Photoreader as Unit 1, set the reading speed to High, mount the STAR II General Purpose Program Load Routine and place the unit into the Remote mode of operation.
   b. After the loader has been executed, the computer will stop. A supervisory message will state KEY IN DATE. At this time, remove the STAR II General Purpose Program Loader from the Photoreader and mount the paper tape containing the symbolic instructions – before the date is entered.
3. If the High Speed Printer is used as output:
   a. Place the High Speed Printer board in the Printer. (See Appendix A.)
   b. Follow the standard installation procedures for setting up the High Speed Printer.
4. If the 407 Line Printer is used as output:
   a. Place the 120-120 board in the 407 Line Printer and set the proper alteration switches.
   b. Follow the standard installation procedures for setting up the Line Printer.
5. If the Paper-Tape Punch is used as output:
   a. Designate the Paper-Tape Punch as Unit 1.
   b. Mount the blank paper tape.
   c. Follow the standard installation procedures for setting up the Paper-Tape Punch.
6. The Supervisory Printer:
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO.**
   d. Follow the standard installation procedures for setting up the Supervisory Printer.
7. Magnetic Tape Storage Units:
   a. Mount the STAR II Program Tape (with Not Write ring) on a tape unit and

designate it as Unit 7.

b. Mount a 21-word blocked scratch tape on a tape unit and designate it as Unit 8.
c. Mount a 21-word blocked scratch tape on a tape unit and designate it as Unit 9.
d. Mount an 81-word blocked scratch tape on a tape unit and designate it as Unit 10. This tape will be the final output of STAR II and will contain the Machine-Language Program.
e. Follow the standard installation procedures for the handling and mounting of the magnetic tape.
f. Make certain that all other Tape Storage Units and Datafiles are designated as Local.

8. Program Control Switches.
STAR II allows considerable flexibility through the use of the Program Control switches located on the Console of the BURROUGHS 220. The Program Control switches and their functions in STAR II are shown in the table below.

TABLE 5-1. Program Control Switch Functions For Assembly

PCS NO.	ON	OFF
1.	Punched card input of the symbolic program.	Paper-tape input of the symbolic program.
2.	Allow the tape label mismatch on tape unit 10 to be bypassed if the programer's name is not identical to the name on the label of the Programer's Machine-Language Tape.	Do not bypass any tape label mismatches.
3.	Print the symbolic output from Pass II on a 407 Line Printer designated as Output Unit 2.	Do not print on the 407 Line Printer during Pass II.
4.	Print (on-line) the symbolic output from Pass II on the High Speed Printer.	Do not print on the High Speed Printer during Pass II.
5.	Do not go into Pass III under any circumstances. Turn PCS 0 on to prevent any references to TSU 0.	Unless errors automatically inhibit Pass III, proceed into Pass III upon completion of Pass II.
6.	During Pass III the assembler will produce a Machine-Language Program on both magnetic tape and punched paper tape. Punched paper tape *cannot* be used if Overlay techniques have been employed.	During Pass III, the assembler will produce a Machine-Language Program on magnetic tape only.
7.	Not used for assembly.	
8.	Not used for assembly.	
9.	Not used for assembly.	
10.	Less than the required number of magnetic-tape units will be used. Assembly—using 3 tape storage units. Unit 0 will not be mounted initially. Mount unit 0 as soon as a tape unit becomes free.	The required number of tape units will be used. Assembly—4 units.

Running the Program

1. Complete the Equipment Setup.
2. If the STAR II General Purpose Program Load Routine is kept on paper tape, enter the instruction (1000 04 0000) into rD, by means of the keyboard. Transfer rD into rC.

   On the other hand, if the STAR II General Purpose Program Load Routine is kept on cards enter the instruction (1000 60 0000) into rD, by means of the keyboard. Transfer rD into rC.
3. Depress the Start button. STAR II will now run automatically until completion or interruption by a supervisory printout. The operator will respond to all messages, taking the proper action as indicated below in Error Flags and Supervisory Messages.
4. When the run is completed remove it from the computer and follow the installation procedures. However, save the tapes on the units designated 8 and 10. Unit 8 contains the symbolic program which can later be used for Reassembly purposes. Unit 10 contains the Programer's Machine-Language tape, which is used for program testing and production runs.

ERROR FLAGS AND SUPERVISORY MESSAGES

Since most Error Flags and Supervisory Messages are identical for both Assembly and Reassembly, all are presented below and are referred to by the operating instructions for both Assembly and Reassembly.

Error Flags

The symbolic listing produced during Pass II of Assembly or Reassembly will contain a number of two-letter error flags, signalling errors or possible errors. This same listing can also be accomplished by listing the tape which is the output from Pass II of either an Assembly or Reassembly. This tape is the one mounted on the tape unit designated as Unit 9.

The flags appearing on the listing should be checked by the programer for possible errors before testing is begun.

The following error flags will be automatically inserted by STAR II during either Assembly or Reassembly when errors or doubtful coding is encountered.

FLAGS	EXPLANATION
AE	An address is specified which has been assembled as a number greater than the size of memory or as 0000.
BE	After the completion of a **BLK** control instruction, the Location Counter was set to a number greater than the size of memory or a **BLK** 0000 was given. In either case, no memory will be reserved.
C4	The programer has unnecessarily expressed the variant digit SL:41 of an instruction and the variant digit expressed is not equal to the one which would have been inserted by the STAR II Assembly Program, automatically. STAR II will assemble the variant digit as expressed by the programer.
DD	Double defined name. (Location)
DE	**DUM** Error. The Address of one or both **DUM's** is incorrect.
LE	Location Counter error. The Location Counter was set to a number greater than the size of memory.
LF	Location Counter error. The Location Counter was set illegally to some number less than 0500.
MD	A **MOD** control instruction has caused the Location Counter to be increased beyond the size of memory or a **MOD** 0000 was given. In either case, **MOD** is not executed and the Location Counter is unchanged.
MT	The macro used cannot be found on the STAR II Program Tape.
OP	Illegal operation code.
PB	Potential **IBB** or **DBB** error.
PS	Potential Shift instruction error.
RV	Violation in the use of a control instruction.
SD	A sign digit of some constant is a 6 or a 7.
SE	A **SET** has been given to an unnamed symbolic reference or a **SYN** refers to an address not previously named. The Location Counter is not set and assembly is terminated after Pass II.
SL	An SL error in partial-word designation.
UA	Undefined symbolic address. (No corresponding location named.)
UM	The macro given is not in the Macro Table.

Supervisory Messages

MESSAGE	EXPLANATION
BMTR NO LOAD	Something is wrong with the parameters of the **BMTR** macro, or the version of **BMTR** requested is not on the odd lane of the STAR II Program Tape. The assembly will continue but no version of **BMTR** will be included.
CB ERROR	An **END** entry was not detected—inhibit Pass III.
C,I,D IS MISSING ON CARD	During reassembly an input entry is not properly defined as an insertion (I), change (C), or deletion (D). The card contents are printed on the **SPO** below the error message.

MESSAGE (Cont.)	EXPLANATION
C,I,D NAME MISSING ON CARD (CARD CONTENTS)	During reassembly an insertion (I), change (C), or deletion (D) entry is not named. (Columns 11-24 are blank.) The card contents are printed on the **SPO** below the error message.
END OF PASS I, II, or III	The end of phase one, two, or three has occurred.
ERROR **END** ASSEMBLY	No Pass III—See listing or previous **SPO's** for specific errors which have caused Pass III to be inhibited.
GG DIGITS WRONG	Wrong tape has been mounted. The block size is incorrect. Change tape and start.
INCREMENT IS INCORRECT (Plus contents of the Card or Paper-Tape Read-in area.)	An attempt has been made to Insert, Delete, or Change an entry whose (reassembly) Name has been assigned a memory location which is less than the present Location Counter setting.
INPUT NAME ER	During reassembly, **END** entries were not found to be present at the end of both the symbolic modifications and the program's symbolic input found on the magnetic tape. Reassembly will stop, the program will be cleared and Pass I of the STAR II Program will be reloaded.
KEY IN DATE	Enter today's date via a **KAD.** Date should be of the form YYDDD. (SL:05)
LOAD ERROR. PGM#	A sum check error has occurred while trying to load the STAR II Program. Begin assembly again. If the same error occurs again use another program tape.
LOD OK	The program has been loaded successfully.
LOD NG (Halt 3333 00 3333)	Pass III has been loaded incorrectly. Depress Start and the tape units will be rewound.
MEM HIGH OR MEM LOW	Memory exceeded. (High—greater than memory size; Low—less than 0500). This is caused by either an LE or LF error. These error flags will appear on the assembly listing. Inhibit Pass III.
MOUNT NEW TAPE ON UNIT 0. (Halt—6641 00 1466)	The block size is incorrect or the **NAME** differs and PCS 2 was not turned on.
MT 7, 8, 9, 0 NOT READY	One or more tape units are not ready. Check all tapes and start.
MT (8, 9 or 0) TAPE LABEL ADDRESS— CHANGE TAPE	Tape label error detected—change tape then press start.
NO **DUM's**	No **DUM** instructions have been given—inhibit Pass III.
NO **NAM** CARD	A **NAM** entry must be inserted and assembly begun again.
NO **SET** CARD—RELOAD WITH **SET** CARD	No **SET** card was placed immediately after the five initial **LOD** cards. Reload the symbolic program including the **SET** card and start.
25 **ØVLY**	Too many **ØVLY** macros were used—inhibit Pass III.

MESSAGE (Cont.)	EXPLANATION
ØVLY-DUM ERROR	The Name of the **DUM** pair does not correspond to the parameter P6 of its **ØVLY** macro.
ØVLY P-4 ERROR	The Search key for this overlay would be incorrectly located in the reserved area of memory < 0500.
PASS LOST I, II, III	Pass X cannot be found on magnetic tape. Check the program tape and the tape system, then, press Start.
PROGRAMER'S NAME DIFFER _____ON TAPE _____ON CARD FORCING LABEL	The 81-word output tape has a Name in its label different from the Name in the **NAM** entry. Change the tape and start again. Or, this Name error may be bypassed by turning on PCS 2 and then pressing Start. "Forcing Label" will print out on the **SPO.**
REMOVE TAPE ON TSU No. 6	During reassembly, after the symbolic input has been read from this tape, remove the tape from the unit. This will make operations more efficient and will minimize the tear-down time at the end of the reassembly.
REMOVE TAPE ON TSU 7, 8, 9, or 0	This tells the operator at the earliest possible time (for efficient operating purposes) when he can remove the tapes from the tape units.
TOO MANY NAMES	More than 1000 names have been used. The computer will Halt.
TSU No. 6 BC ERROR	During Pass I of a reassembly, all of the blocks on the input tape have not been read. The reassembly will then continue but it might be wise to halt the run and restart.
TSU No. 8 BC ERROR	During Pass II of an assembly, all of the blocks on the input tape have not been read. Restart.
26 **DUM's**	Too many **DUM's** have been given—inhibit Pass III.
XS **LOD**	Too many **LOD** entries—inhibit Pass III.

SECTION 6

USE OF REASSEMBLY

Reassembly, as already mentioned, is a fast and convenient method for correcting and updating programs. STAR II's Reassembly Routine makes the changing, insertion, and deletion of instructions faster and easier. The routine reads the bulk of the symbolic program from high-speed magnetic tapes and only uses the slower input mediums of cards or paper tape for the changes, insertions, and deletions themselves.

To use Reassembly two requirements are necessary. One is to save the 21-word Symbolic Output Tape produced by Pass I of the previous Assembly or Reassembly. This tape will contain all of the symbolic entries of the program and thus provide most of the input data. The second requirement is to create the Change, Insert, and Delete cards or paper-tape messages. These changes, insertions, and deletions will be matched against the symbolic entries found on the magnetic tape. When equality is found, appropriate action will be taken.

The format and function of the three types of Reassembly entries are:

CHANGE

A C is placed in column 10 of the card to identify this item as a change. When a change card is encountered, the entry appearing on magnetic tape, which is identified by the name and increment found in columns 11 through 24, will be replaced by the entry found in columns 25 through 80 of the same card.

REASSEMBLY				SYMBOLIC ENTRIES								
CID	NAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT		
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59		6
C	EDIT RØUT		EDIT RØUT			CAD		CUST CØDE		0001		
C	NET PAY	0014	E Ø J RØUT			LDR		E Ø J				

DELETE

A D is placed in column 10 of the card to indicate that the entries found on the symbolic tape, starting with the entry identified by the Name and Increment field (columns 11 through 24) and ending with the entry identified by the Address/Constant field including any increment (columns 45 through 59), will be removed from the program.

REASSEMBLY			SYMBOLIC ENTRIES								
CID	NAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	+ —	ACTUAL OR INCREMENT	
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59	
D	FICA DEDUC							FICA DEDUC		0026	
D	SWITCH 1	0005						SWITCH 2			

It is possible to delete one entry at a time by placing a D in column 10 on the card to indicate that the entry found on the symbolic tape (identified by the Name and Increment field, columns 11 through 24) be eliminated.

REASSEMBLY			SYMBOLIC ENTRIES							
C I D	NAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58
	DFICA DEDUC	0004								

A programer may also delete a specific number of items (columns 56-59), starting with the entry identified by the Name and Increment field (columns 11-24). The number in the Actual/Increment field is a count of the total number of entries to be deleted including the first entry.

REASSEMBLY			SYMBOLIC ENTRIES							
C I D	NAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58
	DFICA DEDUC	0002								012

INSERT

An I is placed in column 10 of the card to identify the entry as one to be inserted immediately following the symbolic location indicated by columns 11 through 24. The item itself, which is to be inserted, will be found in columns 25 through 80. Continue inserting symbolic entries until another C, I, or D entry is recognized. Note that when a continuous insertion of more than one entry is made, only the first entry to be inserted need have an I in column 10 and an entry in columns 11 through 24.

REASSEMBLY			SYMBOLIC ENTRIES							
C I D	NAME	PLUS	NAME	S	CONTROL	OPR.	S	ADDRESS/CONSTANT	±	ACTUAL OR INCREMENT
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 — 43	44	45 46 47 48 49 50 51 52 53 54	55	56 57 58 59
	INY STAT TX	0018				CAD		TAX INCØME		
						STA		ØUTPUT TNK		0012
						STA		ØUTPUT TNK		0004
	DSWITCH 6									

An Insert (I) may not follow a Change (C) or Delete (D), if the Insert, Change, or Delete refer to the same name with the same increment number. This situation can be avoided by placing the Insert first, thus causing the Insert to have a different increment number.

REASSEMBLY NOTES

Program modifications must be in either punched-card or punched paper-tape form. (For the required coding forms and formats, see Section 3.) All modifications should be in the same sequence as they appear on the symbolic listing. Any entry may be referenced by either its Name, or the Name of the previously named entry, plus some positive increment. No new entry may be referenced until the next reassembly.

No **NAM** card is required as the first card during a reassembly. Following the last program modification, a regular **END** entry is required.

SECTION 7

OPERATING INSTRUCTIONS FOR REASSEMBLY

EQUIPMENT REQUIREMENTS

There are a number of equipment configurations possible. The particular configuration to be used will be determined by the setting of Program Control switches.

Configuration A — Card Input, 407 Listing Output

1. One 293, 087, or 089 Card Reader
2. One 407 Line Printer
3. Five Magnetic Tape Storage units*
4. Supervisory Printer.

Configuration B — Card Input, 407 Listing Output, Paper-Tape Output

Same requirements as Configuration A plus a Paper-Tape Punch.

Configuration C — Card Input, High Speed Printer Output

1. One 293, 087, or 089 Card Reader
2. One High Speed Printer
3. Five Magnetic Tape Storage units
4. Supervisory Printer.

Configuration D — Card Input, High Speed Printer Output, Paper-Tape Output

Same requirements as Configuration C plus a Paper-Tape Punch.

Configuration E — Card Input, Punched Paper-Tape Output

1. One 293, 087, or 089 Card Reader
2. Supervisory Printer
3. Five Magnetic Tape Storage units
4. Paper-Tape Punch.

Configuration F — Paper-Tape Input, 407 Listing Output

1. One Photoelectric Reader
2. One 407 Line Printer
3. Five Magnetic Tape Storage units
4. Supervisory Printer.

Configuration G — Paper-Tape Input, 407 Listing Output, Paper-Tape Output

Same requirements as Configuration F plus a Paper-Tape Punch.

Configuration H — Paper-Tape Input, High Speed Printer Output

1. One Photoelectric Reader
2. One High Speed Printer
3. Five Magnetic Tape Storage units
4. Supervisory Printer.

Configuration I — Paper-Tape Input, High Speed Printer Output, Paper-Tape Output

Same requirements as Configuration H plus a Paper-Tape Punch.

*The only difference between Assembly and Reassembly from an equipment standpoint is the use of five Magnetic Tape Storage units rather than four. The use of five tape units, one for each type of tape on Reassembly, eliminates the need for changing tapes and unit designations during the Reassembly run. A Reassembly can, however, be successfully accomplished by using only four tape units and even by using only three units, if absolutely necessary. For an explanation of how this can be accomplished, see Alternate Operator Instructions, page 8-1.

Configuration J — Paper-Tape Input, Paper-Tape Output

1. One Photoelectric Reader
2. Supervisory Printer
3. Five Magnetic Tape Storage units
4. Paper-Tape Punch.

BILL OF MATERIALS

This is a list of all the items, except for equipment, needed to run the job.

1. PLUGBOARDS
   a. If card input is used, one 293, 087, or 089 (80-80) plugboard is required. Format is selected from column 1 with digit select 1 wired to format select 1, and digit select 6 wired to format select 6.
   b. If printed output on the 407 Line Printer is used, a 120-120 on-line 407 board is necessary.
   c. If the High Speed Printer is used, the panel described in Appendix A must be available.
2. INPUT FORMS
   a. If cards are used, the symbolic-language correction or modification cards and the STAR II General Purpose Load Routine (in card form) are required.
   b. If paper tape is used, the symbolic-language correction or modification paper tape and the STAR II General Purpose Load Routine (in paper-tape form) are required.
3. OUTPUT FORMS
   a. A reel of unpunched paper tape should be used if the Machine-Language Program is going to be punched into paper tape.
   b. Paper for the 407 Line Printer (at least 12 inches wide), if used.
   c. Paper for the High Speed Printer, if used.
   d. Paper for the Supervisory Printer.
4. MAGNETIC TAPES

   Five reels of tape are required:
   The STAR II Program Tape.
   Two 21-word, preblocked scratch tapes with **BMTR** tape labels as the first block on each tape.
   An 81-word, preblocked scratch tape with a **BMTR** tape label as the first block on the tape.
   The Symbolic Output Tape from Pass I of the previous assembly or reassembly.
   Note that there are five *different* tapes used by reassembly.

OPERATOR PROCEDURES

Equipment Setup — Reassembly

1. If Card Input is Used:
   a. Place the 80-80 panel, described under Bill of Materials, into the Card Reader.
   b. By following the standard installation procedures, load the changes and modifications in symbolic form into the Reader. The sequence of the symbolic deck for reassembly should be:

      Blank cards*

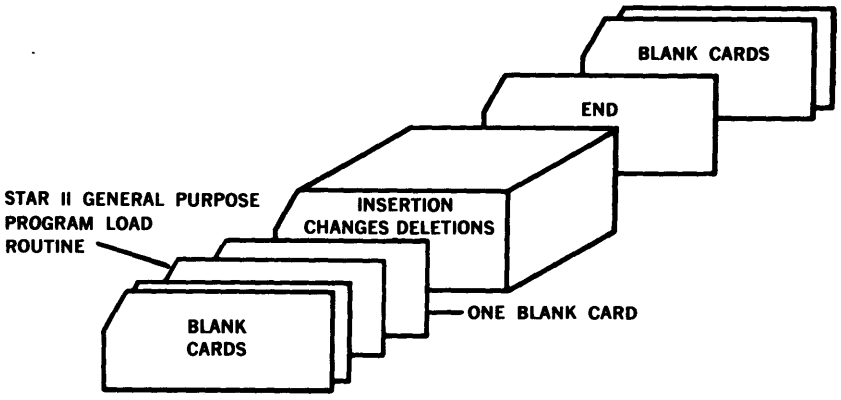
      The STAR II General Purpose Program Load Routine

      The Insertion (I), Change (C), and Deletion (D) cards

      An **END** card will be the last card in the deck.

      Blank cards.

*The number of blank cards will depend upon the type of Reader used.



2. If Punched Paper-Tape Input is used:
   a. Designate the Photoreader as Unit 1. Set the reading speed to High, mount the STAR II General Purpose Program Load Routine and place the unit into the Remote mode of operation.
   b. After the Loader has been executed, the computer will stop. A supervisory message will state KEY IN DATE. At this time, remove the STAR II General Purpose Program Load Routine from the Photoreader and mount the tape containing the Insertions, Changes, and Deletions in symbolic form, before the date is entered.
3. If the High Speed Printer is used as output:
   a. Place the board (described in Appendix A) in the High Speed Printer.
   b. Follow the standard installation procedures for setting up the High Speed Printer.
4. If the 407 Line Printer is used as output:
   a. Place the 120-120 board in the 407 Line Printer and set the proper alternation switches.
   b. Follow the standard installation procedures for setting up the 407 Line Printer.
5. If the Paper-Tape Punch is used for output:
   a. Designate the Paper-Tape Punch as Unit 1.
   b. Mount the blank paper tape.
   c. Follow the standard installation procedures for setting up the Paper-Tape Punch.
6. The Supervisory Printer:
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the SPO
   d. Follow the standard installation procedures for setting up the Supervisory Printer.

7. Magnetic Tape Storage Units:
   a. Mount the STAR II Program Tape (with **NOT** Write Ring) on a tape unit and designate it as Unit 7.
   b. Mount a 21-word blocked scratch tape on a tape unit and designate it as Unit 8.
   c. Mount a 21-word blocked scratch tape on a tape unit and designate it as Unit 9.
   d. Mount an 81-word blocked scratch tape on a tape unit and designate it as Unit 10. This tape will be the final output of the reassembly and will contain the Machine-Language Program.
   e. Mount the program's Symbolic Input Tape on a tape unit and designate it as Unit 6. This tape is the Symbolic Output tape produced by Pass I of the previous assembly or reassembly on Tape Storage Unit 8 and saved for reassembly purposes.
   f. Follow the standard installation procedures for the handling and mounting of magnetic tapes. Make certain that all other Tape Storage Units and Datafiles are designated as Local.
8. Program Control Switches. The Program Control Switches and their functions in STAR II are shown in the table below.

TABLE 7-1. Program Control Switch Functions for Reassembly

PCS NO.	ON	OFF
1.	Punched card input of the symbolic program.	Paper-tape input of the symbolic program.
2.	Allow the tape label mismatch on tape unit 10 to be bypassed if the programer's name is not identical to the name on the label of the Programer's Machine-Language Tape.	Do not bypass any tape label mismatches.
3.	Print the symbolic output from Pass II on a 407 Line Printer designated as Output Unit 2.	Do not print on the 407 Line Printer during Pass II.
4.	Print (on-line) the symbolic output from Pass II on the High Speed Printer.	Do not print on the High Speed Printer during Pass II.
5.	Do not go into Pass III under any circumstances. Turn on PCS 0 to prevent any references to TSU 0.	Unless errors automatically inhibit Pass III, proceed into Pass III upon completion of Pass II.
6.	During Pass III, the assembler will produce a Machine-Language Program on both magnetic tape and punched paper tape. Punched paper tape cannot be used if overlay techniques have been employed.	During Pass III, the assembler will produce a Machine-Language Program on magnetic tape only.
7.	Reassembly.	Regular assembly.
8.	During reassembly, page and line numbers will be renumbered.	During reassembly, no automatic renumbering of page and line numbers will take place.
9.	Some Reassembly Halts will be overridden by a depression of the Start key.	No Halts will be overriden.

TABLE 7-1. (Continued)

PCS NO.	ON	OFF
0.	Less than the required number of magnetic tape units will be used. (Reassembly using 3 or 4 Tape Storage Units.) If PCS 0 is on during reassembly, a **KAD** (keyboard add) instruction (8880 08 0888) will occur and the number of TSU's to be used (either 3 or 4) will be entered in the SL:01 position of the D register. When four Tape Storage Units are used, a Halt (7270 00 2270) will occur at the end of Pass II to provide for the mounting of the Programer's Machine-Language Tape on TSU 0. In the event that three Tape Storage Units are being used, a Halt (6990 00 0699) will occur at the end of Pass I to allow the operator to remove the tape on TSU 6 and to mount a 21-word preblocked tape on the unit. The TSU will be designated as TSU 9. A Halt (7270 00 2270) will also occur at the end of Pass II to allow the operator to mount the Programer's Machine-Language Tape.	The recommended number of tape units will be used. Reassembly—5 units.

Running the Program

1. Complete the Equipment Setup.
2. If the STAR II General Purpose Program Load Routine is kept on paper tape, by means of the keyboard enter the instruction (1000 04 0000) into rD. Transfer rD into rC.

   On the other hand, if the STAR II General Purpose Program Load Routine is kept on cards, by means of the keyboard enter the instruction (1000 60 0000) into rD. Transfer rD to rC.
3. Make certain that PCS 7 is turned on and depress the Start button. STAR II (reassembly) will now run automatically until completion or interruption by a supervisory printout. The operator will respond to all messages and will take the proper action. (See Error Flags and Messages, Section 5.)
4. When the run is completed remove it from the computer and follow the installation procedures. However, save the tapes on the units designated as 8 and 10.

   Unit 8 contains the symbolic program which can later be used for reassembly purposes. Unit 10 contains the Programer's Machine-Language tape, which is used for program testing and production runs.

SECTION 8

ALTERNATE OPERATING INSTRUCTIONS

As stated before, it is recommended that four Tape Storage Units be used for each assembly run and that five Tape Storage Units be used for each reassembly run. However, it was also noted that assemblies could be accomplished using only three storage units, while reassemblies can be achieved by the use of four tape storage units or even three storage units, if absolutely necessary. This section will explain how to run assemblies and reassemblies with fewer Tape Storage Units than the number recommended.

ASSEMBLY (Using Three Tape Storage Units)

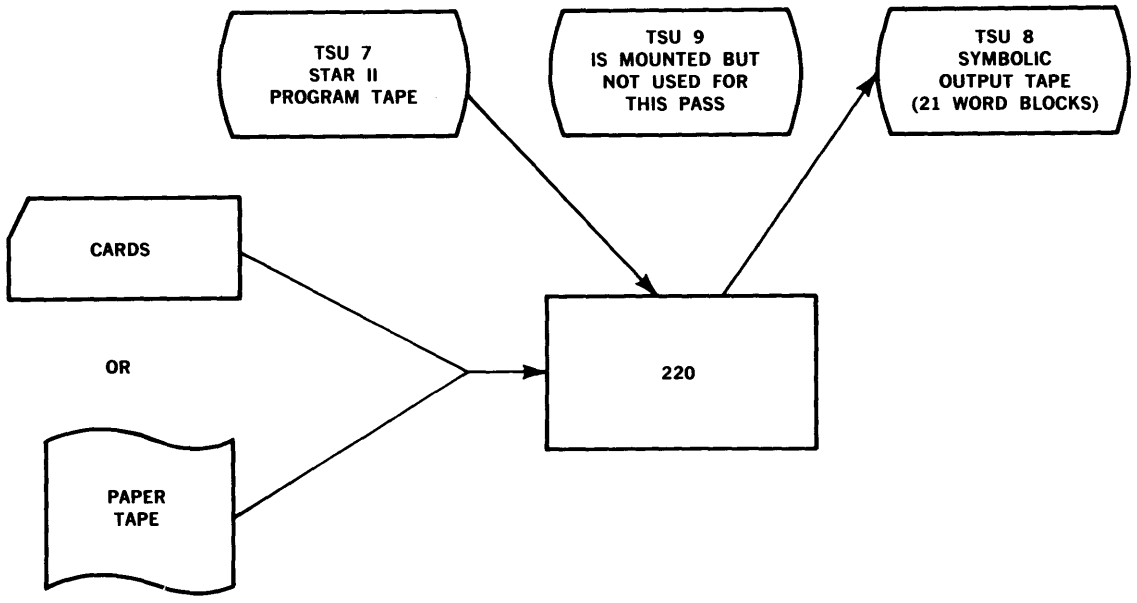
For Passes I and II, it is necessary to have Tape Storage Units designated as units 7, 8, and 9. Program Control Switch 0 must be turned on to inhibit references to TSU 0 and to cause a Halt after Pass II (7270 00 2270) in order to complete reel changing.

TSU 7 will contain the STAR II Program Tape.

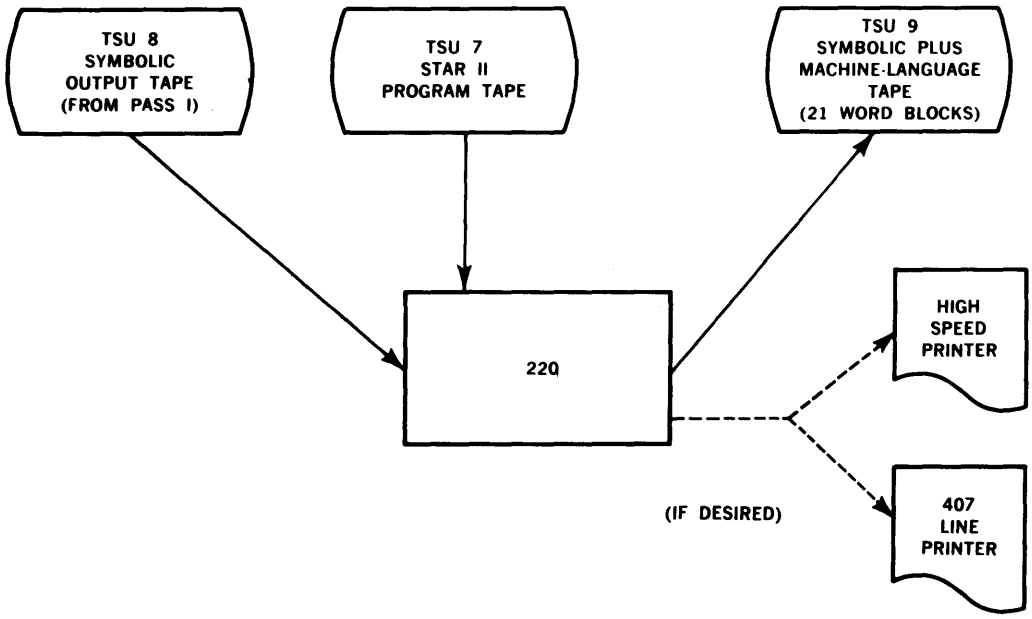
TSU 8 will be a 21-word blocked tape used for the symbolic output from Pass I.

TSU 9 is the Symbolic plus Machine-Language Tape and is the output from Pass II.

Pass I

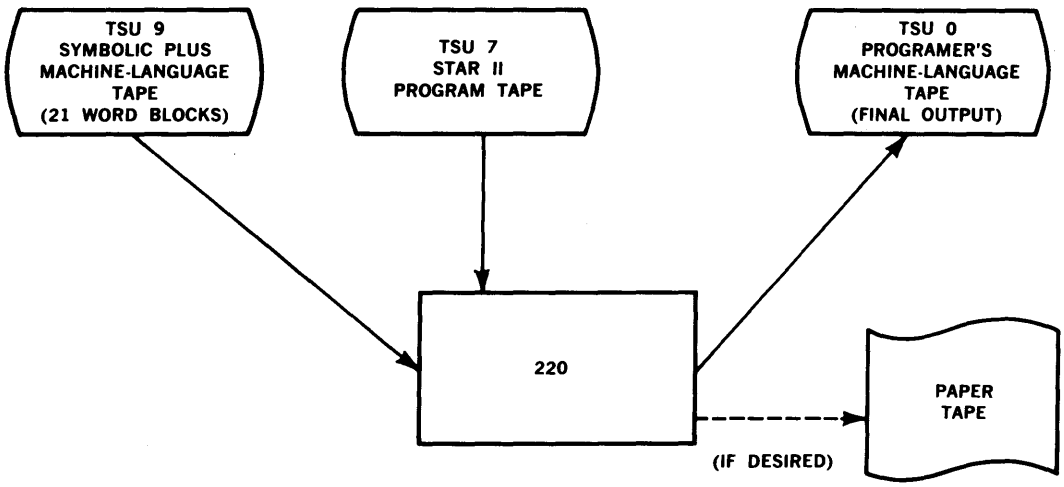


Pass II



For Pass III it is necessary to have Tape Storage Units designated as TSU 7, TSU 9, and TSU 0. TSU's 7 and 9 will contain the same tapes as mentioned above. TSU 0 is an 81-word preblocked tape which will contain the Machine-Language Program and is the final output of the assembly. Hence, at the end of Pass II it is necessary to stop the computer and remove the tape on TSU 8. Then mount the 81-word Programer's Machine-Language Tape and redesignate the TSU as 0. To cause the computer to halt at the end of Pass II and bypass the instructions referring to TSU 0 during earlier passes, Program Control Switch No. 0 will be turned on. An **SPO** will print "Remove Tape on TSU 8."

Pass III



REASSEMBLY (Using Four Tape Storage Units)

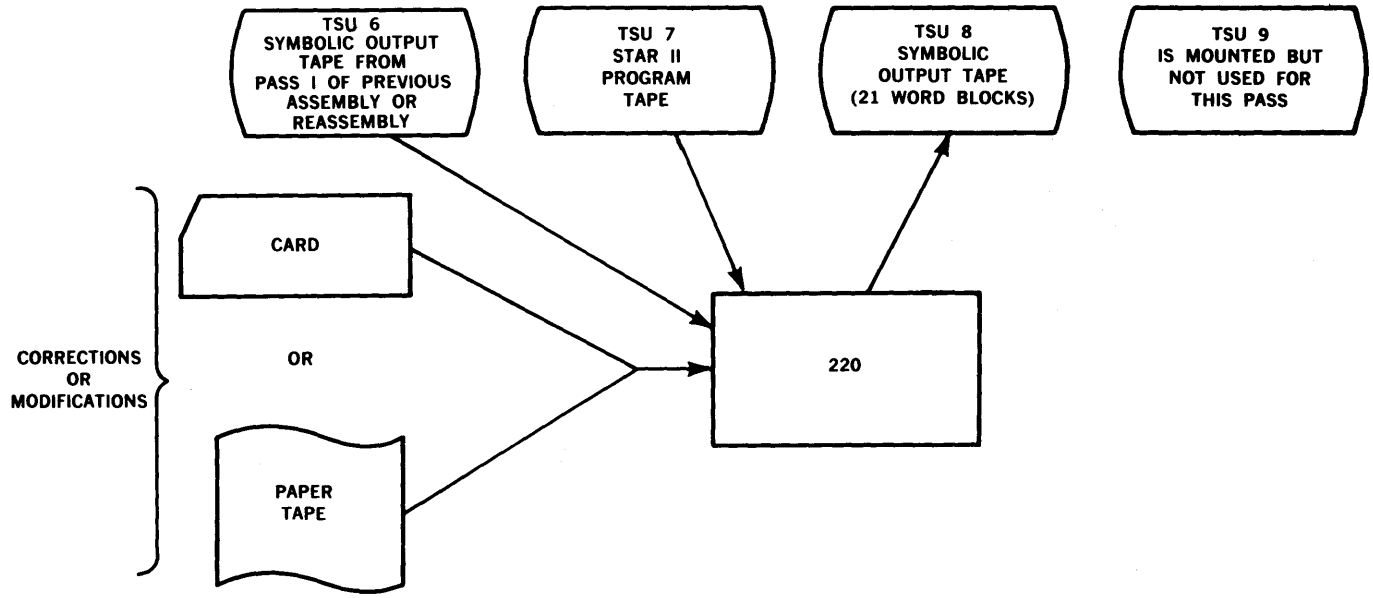
For Pass I, it is necessary to have Tape Storage Units designated as Units 6, 7, 8, and 9. Program Control Switch 0 must be turned on to inhibit references to TSU 0 and to cause a Halt after Pass II (7270 00 2270) to allow for reel changing. Further, it will cause **KAD** (keyboard add) instruction to be executed. The operator will key into rD SL:01 the number of tape units to be used during the reassembly — either three or four.

TSU 6 is the Symbolic Output Tape developed by Pass I of the previous assembly or reassembly.

TSU 7 is the STAR II Program Tape.

TSU's 8 and 9 are 21-word blocked scratch tapes.

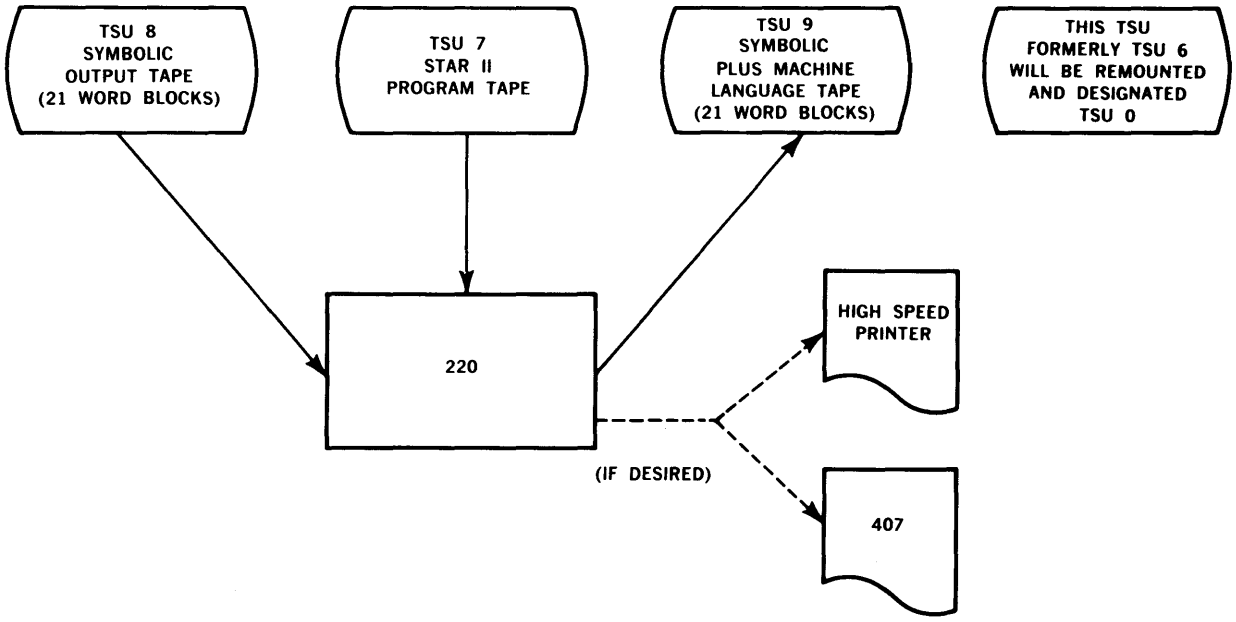
Pass I



For Pass II, it is necessary to have Tape Storage Units designated as Units 7, 8, and 9. At the beginning of this pass a message on the SPO will state "Remove Tape on TSU No. 6." While Pass II continues to process, remove this tape, mount an 81-word blocked tape to be used as the Programer's Machine-Language Tape and designate this Tape Storage Unit as 0.

In most cases, this reel can be changed while Pass II is in progress and no time will be lost due to stopping the computer for reel changes prior to continuing with Pass III.

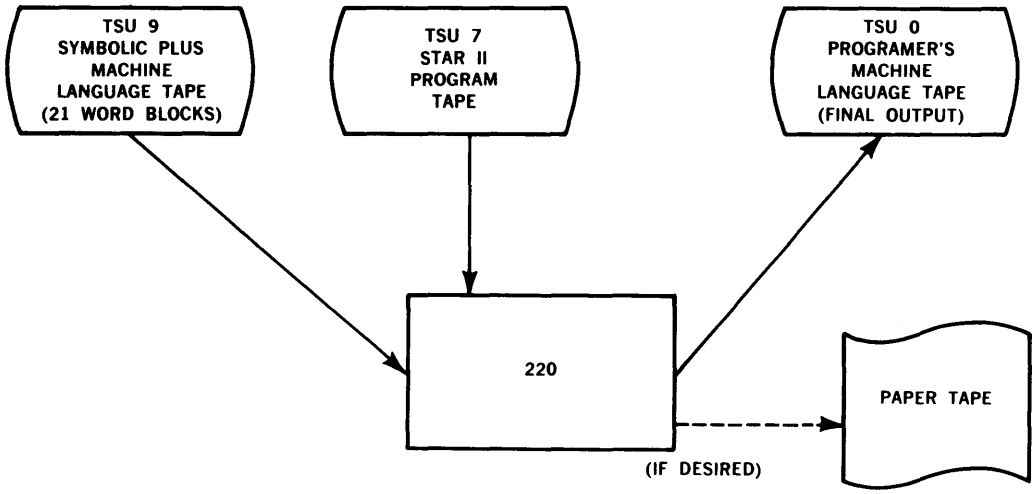
Pass II



For Pass III, it is necessary to have Tape Storage Units designated as 7, 9, and 0.

TSU 0 is an 81-word blocked tape which will be used as the Programer's Machine-Language Tape and is the final output from the run.

Pass III



REASSEMBLY (Using Three Tape Storage Units)

This is the least desirable equipment configuration for the use of reassembly. However, if it is necessary to use only three tape units, the following procedure can be followed:

Turn PCS 0 On.

This inhibits references to TSU 0 and will cause a Halt after Pass II (7270 00 2270), to allow for reel changing. Further, since PCS 7 is on, it will cause a **KAD** (keyboard add) instruction to be executed (8880 08 0888). The operator will key into rD SL:01 the number of tape units to be used during the reassembly — either four or three. If three units are used, this entry will prevent any references to Tape Storage Unit 9, during the first pass.

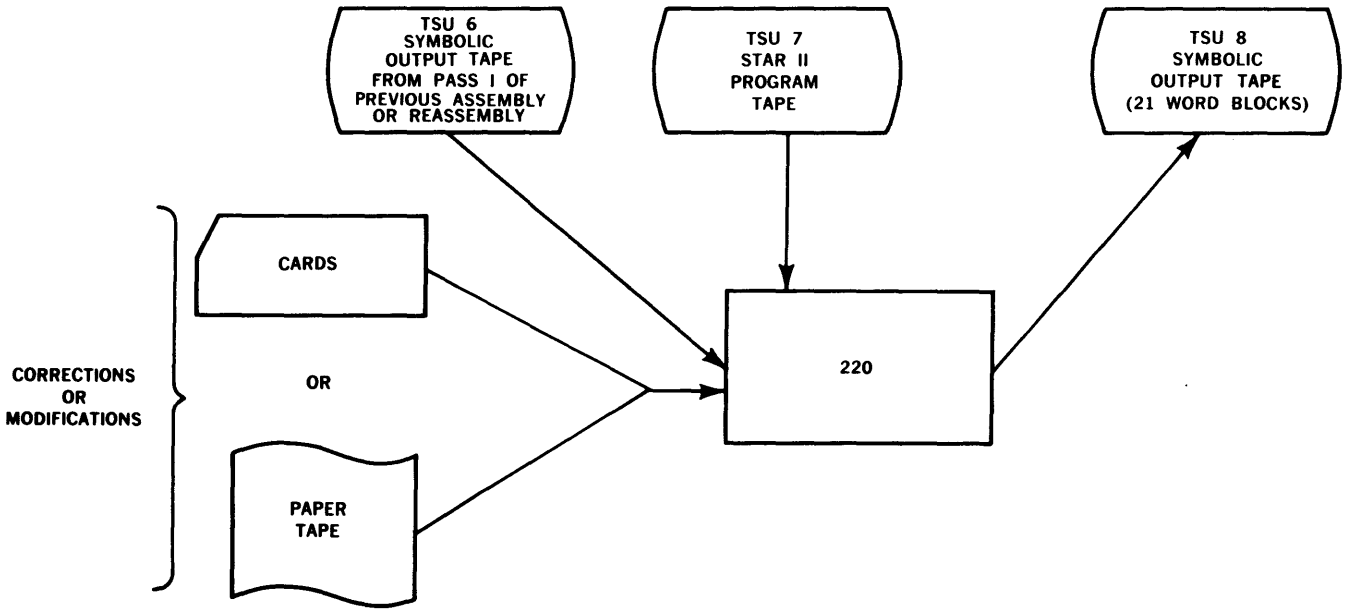
For Pass I, it is necessary to have three storage units designated as Units 6, 7, and 8.

TSU 6 is the Symbolic Output Tape developed by Pass I of the previous assembly or reassembly.

TSU 7 is the STAR II Program Tape.

TSU 8 is a 21-word blocked scratch tape.

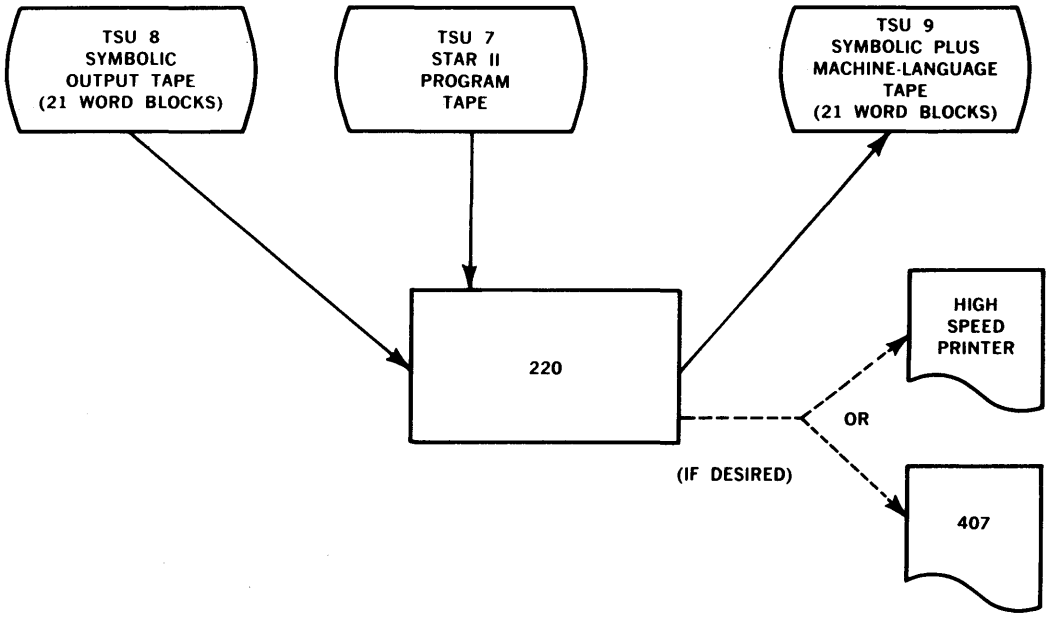
Pass I



At the end of Pass I a Halt will occur (6990 00 0699) which will enable the operator to remove the tape from the Tape Storage Unit designated as Unit 6 and to replace it with a 21-word blocked scratch tape. This unit will be redesignated as TSU 9.* The operator will then depress Start and Pass II will be executed.

*TSU 9 will be the Symbolic Plus Machine-Language Tape produced by the next pass.

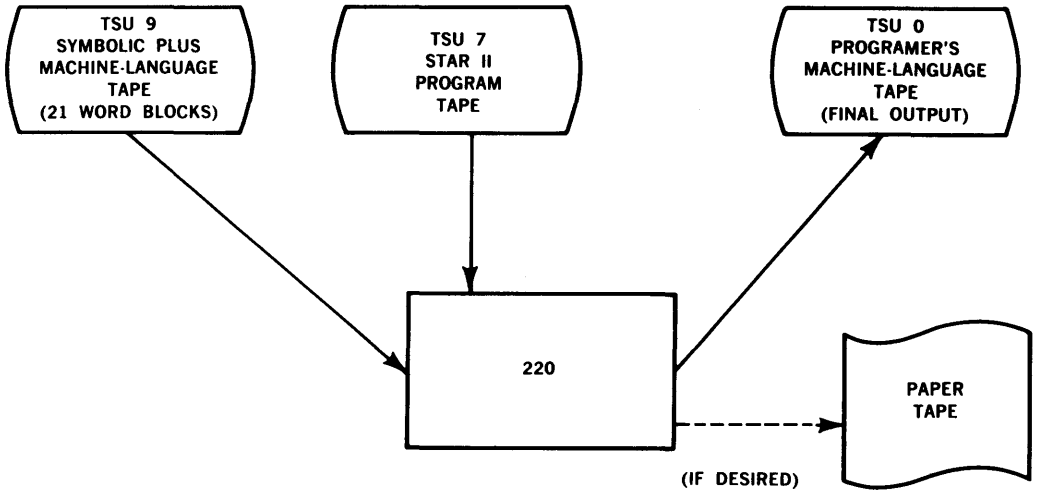
Pass II



At the end of Pass II, the computer will halt (7270 00 2270) because PCS 0 is on. This allows the operator to remove the tape from the unit designated as TSU 8 and replace it with an 81-word blocked tape, which will be the Programer's Machine-Language Tape. This unit will then be designated as Unit 0.

For Pass III, it is necessary to have Tape Storage Units designated as 7, 9, and 0. TSU 0 is an 81-word blocked tape which will be used as the Programer's Machine-Language Tape and is the final output from the run.

Pass III



SECTION 9

METHOD FOR CALLING A PROGRAM INTO MEMORY FROM THE PROGRAMER'S MACHINE-LANGUAGE TAPE

Pass III places a Machine-Language Program on the Programer's Machine-Language Tape (81-word blocks). Two alternatives now present themselves: (1) To leave the program on this tape all by itself, which is normally what is done during testing and debugging; or (2) To place this program on another tape as part of a program library. (See Library Creation Routine.) This is the normal procedure for production runs. In either case it is necessary to load memory with the program in order to process.

CASE NUMBER I

In case number one, where there is only *one* program on the tape, it is possible to load this program either manually or automatically.

Manually

1. The operator, after mounting the program tape and designating it as Unit 7, can skip the tape label by rewinding the tape **(MRW)** and positioning the tape forward **(MPF)** one block.
2. Next, the operator will read one block (the Load Block) into location 0000.
3. He will then transfer control to location 0001.
4. The Load Block then moves itself to the location specified by the programer in **LOD** P3.
5. The Load Block (created by the 5 **LOD's** required at the beginning of each program written in STAR II) then calls in the program, one block at a time, places it in the proper storage locations and transfers control to the location specified by **LOD** P4.

Automatically

The above Program Load Routine might best be placed on a short paper tape or on load cards. The instructions that would be required are:

Card

Card 1	6 1000 60 0212	**CRD**
Card 2	0 7008 50 0000	**MRW**
~~Card 3~~	0 7100 58 0000	**MPF**
~~Card 4~~	0 7101 52 0000	**MRD**
~~Card 5~~	0 0000 30 0001	**BUN**
~~Card 6~~	6 0000 30 0209	**BUN**

To load the program it would merely be necessary for the operator to cause a (1000 60 0000) Card Read instruction to be executed.

Paper Tape

Word 1	6 1000 04 0200	**PRD**
Word 2	0 7008 50 0000	**MRW**
Word 3	0 7100 58 0000	**MPF**
Word 4	0 7101 52 0000	**MRD**
Word 5	0 0000 30 0001	**BUN**
Word 6	6 0000 30 0200	**BUN**

To load the program it would merely be necessary for the operator to cause a 1000 04 0000 Paper-Tape Read instruction to be executed.

CASE NUMBER II

For case number two, automatically loading programs when a program library tape is used, a General Purpose Program Load Routine is recommended. This routine is available in paper-tape form. See Section 10.

SECTION 10

THE GENERAL PURPOSE PROGRAM LOAD ROUTINE

This routine is used to locate a program on the Master Program Tape, created by the Program Library Creation Routine, and to load it into memory.

To use this routine, the General Purpose Program Load Routine will be placed on the Photoreader and a Paper-Tape Read instruction will be executed. The program will stop on a **KAD** (keyboard add) instruction. At this time, the number of the program desired will be entered.

The General Purpose Program Load Routine will locate the desired program on the Master Program Tape and read the first block of the program, the **LOD** block, into locations 0000 and following. Control will be transferred to location 0001 and the **LOD** block will proceed to load its program in the usual manner.

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader
2. One Supervisory Printer
3. One Magnetic Tape Storage Unit.

BILL OF MATERIALS

1. INPUT FORMS.
   The General Purpose Program Load Routine on Paper Tape.
2. OUTPUT FORMS.
   Paper for the Supervisory Printer.
3. MAGNETIC TAPES.
   The Master Program Tape.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape

   Designate the Photoreader as Unit 1. Set the reading speed to High. Mount the General Purpose Program Load Routine and place the unit into the Remote mode of operation.

2. Supervisory Printer
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO.**
   d. For further details, refer to the standard installation procedures for setting up the **SPO.**
3. Magnetic Tapes
   a. Mount the Master Program Tape on a tape unit and designate it as TSU 7.
   b. Follow the standard installation procedures for the handling and mounting of magnetic tapes.
   c. Make certain that all other Tape Storage Units and Datafiles are in Local.

4. Program Control Switches.

	ON	OFF
PCS NO. 0	Program number keyed in is incorrect. Press Start and the Loader will start from the beginning.	Program number keyed in is correct. Press Start and continue.

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD into rC.
3. Make sure that Program Control Switch 0 is set properly. Depress the Start button. The General Purpose Program Loader will now run automatically until completion or interruption by a supervisory printout.

   The operator will respond to all messages and programed halts, taking the proper action as indicated in the Supervisory Messages and Programed Halts chart below.

Supervisory Messages and Programed Halts

MESSAGE OR HALT	EXPLANATION
PGM Load Routine	For identification purposes.
*Enter PGM No. (KAD 8421 08 4210)	The program will stop on a **KAD** (keyboard add) instruction (8421 08 4210). Enter via the keyboard the number of the program to be loaded into memory.
PGM No. (HLT 0660 00 6600)	This is provided for operator verification purposes. The computer will Halt (0660 00 6600). If the program number is correct, depress Start and continue normally. If the program number is incorrect, turn Program Control Switch 0 on and depress Start.
MT-7 Not Ready (HLT 1991 00 1991)	Check tape Unit 7. If necessary change the tape. Depress Start.
Ready (HLT 7997 00 7997)	The program has been found on the program tape. Depress Start and the program will be loaded.
Can't find Program No._____ (HLT 2992 00 2992)	Change the Program tape or depress Start and try again.

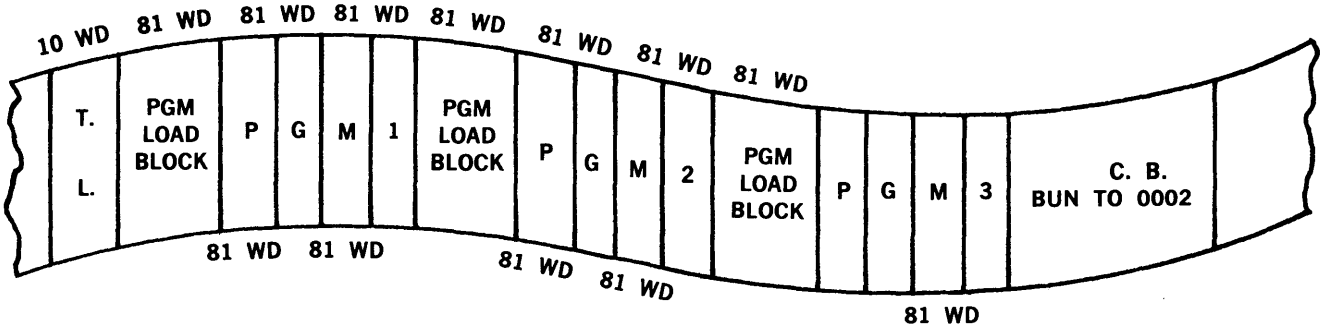
*The program number for finding STAR II is 0010000 (SL:07).

SECTION 11

CREATION OF A PROGRAM LIBRARY TAPE

This routine places programs on a Master Program or Program Library Tape. The programs on the Master Program Tape can be called into memory by means of the General Purpose Program Load Routine or an executive type of routine developed by an installation.

The programs will appear on the Master Program Tape in the following way:



All programs will be arranged in searchable sequence by Program Number.

In order to load memory with a program which is on the Master Program Tape, place the General Purpose Program Load Routine on the Photoreader. Execute a Paper-Tape Read instruction. The program will stop on a **KAD** (keyboard add) instruction. At this time, the number of the program requested will be entered.

The General Purpose Program Load Routine will locate the desired program on the Master Program Tape and then read the first block of the program, the **LOD** block, into locations 0000 and following. Control will be transferred to location 0001 and the **LOD** block will proceed to load its program in the normal manner.

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader
2. One Supervisory Printer
3. Three Magnetic Tape Storage Units.

BILL OF MATERIALS

1. PLUGBOARDS.
   None
2. INPUT FORMS.
   The Program Library Creation Routine in paper-tape form.
3. OUTPUT FORMS.
   Paper for the Supervisory Printer.
4. MAGNETIC TAPES.
   a. One 81-word scratch tape with a 10-word **BMTR** tape label.
   b. Master Program Tape. If there is no existing Master Program Tape, an 81-word blocked scratch tape with a 10-word **BMTR** tape label must be created. The block following the tape label must have a search key of 9980001. The next block must be a control block with a search key of 79999999999 and a change control to 0002.

c. The program to be added. This magnetic tape is the Programer's Machine-Language Tape produced by Pass III of the STAR II Assembly Routine.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape.
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the Program Library Creation Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer.
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO**.
   d. For further details, refer to the standard installation procedures for setting up the **SPO**.
3. Magnetic Tapes.
   a. Mount the Master Program Tape on a tape unit and designate it as TSU 6.
   b. Mount the program to be added on a tape unit and designate it as TSU 0. The program on the tape must appear in the form produced by the STAR II Assembly Routine.
   c. An 81-word blocked scratch tape with a **BMTR** tape label will be mounted on a tape unit and this unit will be designated as TSU 5. This tape will contain the updated Master Program Tape.
   d. Follow the standard installation procedures for the handling and mounting of magnetic tapes.
   e. Make certain that all other Tape Storage Units and Datafiles are in Local.
4. Program Control Switches.

PCS	ON	OFF
0	Do not **SPO** program numbers except for the program to be added.	**SPO** the program number of all programs.
1	A program with the same number has been found on the Master Program Tape. Do not replace the program on the Master Program Tape. Terminate the creations of the new Master Program Tape.	Replace the program on the Master Program Tape with the one found on the Programer's Machine-Language Tape (TSU 0) and continue normally.

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Make sure that all Program Control Switches are set properly. Depress the Start button. The Library Creation Routine will now run automatically until completion or interruption by a supervisory printout.

   The operator will respond to all messages and programed halts, taking the proper action as indicated in the Supervisory Messages and Programed Halts chart below.

4. Upon completion of the run, remove it from the computer, following the installation's procedures. The tape on TSU 5 will contain the updated Master Program Tape.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE OR HALT	EXPLANATION
Program to Master Tape Routine for use with STAR II. Use TSU's 0, 5, 6.	Program Identification.
TSU 0, 5, 6 N/R (HLT—7777)	Check the tape unit in question. Take remedial action and depress Start.
PGM No.— _____ (HLT—9669)	The program number of the program to be added. Verify its correctness and depress Start.
PGM No.— _____	The program number of the program being written onto the updated Master Program Tape.
Program with the same number on Master Tape. If it is to be updated turn PCS No. 1 off. Turn PCS No. 1 on, there will be no update. (HLT—7777)	Follow instructions expressed by the SPO.
End of Program (HLT—9999)	This is the end of the program. Depress Start and the program will reset itself for immediate use.

SECTION 12

THE USE OF STAR II TO REASSEMBLE ITSELF

STAR II is capable of reassembling itself if an installation wishes to alter the STAR II program.

However, when STAR II reassembles itself, the macro generators, Macro Table, and the versions of **BMTR** are not placed on the Programer's Machine-Language Tape, produced by the reassembly. It is necessary, therefore, to use three routines to place the Macro Table, macro generators and **BMTR** on the new STAR II Machine-Language Tape.

These routines are:

1. The STAR II Blocking and **BMTR** Copy Routine.
2. The Copy Macro Routine.
3. The Copy Macros Table and Create New Hash Total Routine.

The STAR II Blocking and **BMTR** Copy Routine

Since STAR II searches its program tape for the various passes and macro generators, the new STAR II program tape must be preblocked on the even lane with the blocks appearing in a searchable sequence.

In addition to this preblocking on the even lane, the various versions of **BMTR** must be copied onto the odd lane of what will become the new STAR II Program Tape. This routine will copy the odd lane of the old STAR II Program onto the odd lane of the new STAR II Program Tape.

The Copy Macros Routine

After reassembly, this routine will locate the macro generators on the old STAR II Machine-Language Program Tape and copy them onto the new STAR II Machine-Language Program Tape.

The Copy Macros Table and Create New Hash Total Routine

After reassembly this routine will copy the Macro Table found on the old STAR II Program Tape onto the new STAR II Machine-Language Program Tape and will generate a new Hash Total, incorporating it on the new program tape.

THE STAR II BLOCKING AND BMTR COPY ROUTINE

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader
2. One Supervisory Printer
3. Two Magnetic Tape Storage Units.

BILL OF MATERIALS

1. PLUGBOARDS.
   None
2. INPUT FORMS.
   The STAR II Blocking and **BMTR** Copy Routine paper-tape form.
3. OUTPUT FORMS.
   Paper for the Supervisory Printer.
4. MAGNETIC TAPES.
   Two reels of magnetic tape are required:
   a. The old STAR II Machine-Language Program Tape.

b. A newly edited tape.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape.
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the STAR II Blocking and **BMTR** Copy Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer.
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO**.
   d. For further details, refer to the standard installation procedures for setting up the **SPO**.
3. Magnetic Tapes.
   a. Mount the old STAR II Machine-Language Program Tape on a Tape Storage Unit and designate it as TSU 7.
   b. Mount an edited tape on a Tape Storage Unit and designate it as TSU 0.
   c. Follow the standard installation procedures for the handling and mounting of magnetic tapes.
   d. Make certain that all other Tape Storage Units and Datafiles are designated as Local.
4. Program Control Switches.
   a. None.

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Depress the Start button.
4. The blocking of the even lane will take place after which the computer will Halt (6996 00 6996).
5. Depress the Start key if **BMTR** is to be copied onto the odd lane of the new tape. The computer will Halt again (3333 00 3333) after the **BMTR** Copy Routine is loaded. To continue depress Start again and **BMTR** will be copied.
   The end-of-job Halt is 6996 00 6996.
6. Upon completion of the run, remove it from the computer, following the installation's standard procedures.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE OR HALT	EXPLANATION
STAR II Tape Ready (6996 00 6996)	The even lane of the tape has been blocked in a manner acceptable for reassembly of STAR II by itself. To copy **BMTR** on the Odd lane of the tape, depress Start.
HLT (3333 00 3333)	The **BMTR** copy portion of the routine has been loaded into memory. To continue depress Start.
BMTR on Tape (6996 00 6996)	**BMTR** has been copied onto the Odd lane of the tape and the program has been completed.

MESSAGE (Cont.)	EXPLANATION
Tape too short HLT (7227 00 7227)	The tape being blocked does not contain enough blocks to be used as the STAR II Program tape.

THE MACROS COPY ROUTINE

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader.
2. One Supervisory Printer.
3. Two Magnetic Tape Storage Units.

BILL OF MATERIALS

1. PLUGBOARDS.

None.

2. INPUT FORMS.

The Macros Copy Routine in paper-tape form.

3. OUTPUT FORMS.

Paper for the Supervisory Printer.

4. Magnetic Tapes.

Two reels of Magnetic Tape are required:

a. The old STAR II Machine-Language Program tape.

b. The new STAR II Machine-Language Program tape.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape.
   a. Designate the photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the Macros Copy Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer.
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the SPO.
   d. For further details, refer to the standard installation procedures for setting up the SPO.
3. Magnetic Tapes.
   a. Mount the old STAR II Machine-Language Program Tape on a Tape Storage Unit and designate it as TSU 7.
   b. Mount the new STAR II Machine-Language Program on a Magnetic Tape Storage Unit and designate it as TSU 0.
   c. Follow the standard installation procedures for the handling and mounting of magnetic tapes.
   d. Make certain that all other Tape Storage Units and Datafiles are designated as Local.
4. Program Control Switches.
   a. None.

Running the Program

1. Complete the Equipment Setup.

2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Depress the Start button. The Macros Copy Routine will run automatically until completion or interruption by a supervisory printout.

   The operator will respond to all messages and programed halts, taking the proper action as indicated in the Supervisory Messages and Programed Halts chart below.
4. Upon completion of the run, remove it from the computer, following the installation's standard procedures.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE OR HALT	EXPLANATION
(Search key of the Block.)	This provides a list of the Search keys of the macro generators placed on tape.
END (HLT—6996 00 6996)	End-of-Job. The Macros listed on the **SPO** have been copied on the new program tape.

COPY MACRO TABLE AND CREATE NEW HASH TOTAL ROUTINE

Operating Instructions

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader.
2. One Supervisory Printer.
3. Two Magnetic Tape Storage Units.

BILL OF MATERIALS

1. PLUGBOARD.
   None.
2. INPUT FORMS.
   The Copy Macro Table and Create New Hash Total Routine.
3. OUTPUT FORMS.
   Paper for the Supervisory Printer.
4. MAGNETIC TAPES.
   Two reels of magnetic tape are required:
   a. The old STAR II Machine-Language Program tape.
   b. The new STAR II Machine-Language Program tape.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape.
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the Copy Macro Table and Create New Hash Total Routine and place the unit into the Remote mode of operation.
2. Supervisory Printer.
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation and designate it as the **SPO**.
   c. For further details refer to the standard installation procedures for setting up the **SPO**.
3. Magnetic Tapes.
   a. Mount the old STAR II Machine-Language Program Tape on a Tape Storage Unit

and designate it as TSU 0.

b. Mount the new STAR II Machine-Language Program on a Magnetic Tape Storage Unit and designate it as TSU 7.

c. Follow the standard installation procedures for the handling and mounting of magnetic tapes.

d. Make certain that all other Tape Storage Units and Datafiles are designated as Local.

4. Program Control Switches.

   a. None.

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Depress the Start button. The routine will run automatically until the End-of-Program Halt is encountered.
4. Upon completion of the run, remove it from the computer following the installation's standard procedures.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

There is one Programed Halt. This is the End-of-Job (program) Halt. The Halt identification is 0000 00 1212.

There is also one Supervisory Message. This is PGM No. -00000 10000, which should be ignored. It occurs as a result of the use of the Load block at the beginning of the program. It has no significance as far as this program is concerned.

APPENDIX A

HIGH SPEED PRINTER PANEL

The following is the plugboard program and operating instructions required to list a program assembled by Star II on the High Speed Printer.

This plugboard is the board commonly called the "Matts" panel. It has the ability to perform various other functions aside from listing a program produced by Star II. However, only that information which is pertinent to Star II has been mentioned.

OPERATING INSTRUCTIONS

ON-LINE PRINTING

1. For STAR II On-line, both Alteration Switch 1 and Alteration Switch 2 must be depressed.
2. Depress the Start key on the High Speed Printer Control Panel.
3. The printing is now under control of the 220 Console.

OFF-LINE PRINTING

1. Mount the plugboard in the Printer Control Unit. Alteration Switch 5 must be off.
2. Load a 21-word preblocked print tape, containing data to be printed on the High Speed Printer Tape unit, and designate it Unit 1.
3. Clear the Control Panel and depress the Start key on the Printer Control Unit.
4. For listing STAR II tapes, depress Alteration Switch 2. Also set the Field Interrogate switches 1 through 6 to 455544. (No control block follows STAR II, so the High Speed Printer tests for **END**.)

STAR II TAPE LAYOUT

50	51	52	53	54
Id. Number	Name 1st Half	Name 2nd Half	Sign Control	Operand

55	56	57	58	59
Address Constant	Address Constant	Actual Increment	Remarks	Remarks

60	61	62	63	64
Remarks	Page and Line Number	STAR II Assigned Address	Assembled Instruction	Error Flags

65	66	67	68	69
Error Flags	Error Flags	Error Flags	Error Flags	Error Flags

Flags Used by STAR II 70	

APPENDIX B

BURROUGHS MAGNETIC TAPE ROUTINE BMTR

INTRODUCTION

Data-Processing or scientific installations which make extensive use of magnetic tape as intermediate storage, face certain problems which are unique to this medium. Since it is not reasonable to visually check what has been recorded on tape, it becomes the responsibility of the programer and the console operator to ensure that the correct tape files are being processed during any run.

One method of making certain that the correct file is being updated is to externally identify each reel with a gummed label. A second method is to write an identifying block as the first block of a reel which can be checked either under program control or by a type-out on the Supervisory Printer. The most commonly used procedure is a combination of the methods described.

Presuming then that both an external and an internal check are being used, another problem presents itself; exactly what should this label look like? Left to their own devices, every programer in an installation will design a different label format and a different method of checking the label. It is obvious, therefore, that an installation must standardize certain tape-handling procedures in order to promote efficient computer operation, the first of these being the design and handling of magnetic-tape labels.

The programer must also provide controls for going from lane to lane and/or from one tape to another of a multiple-reel file. These routines should provide the ability to switch from one transport to another, if the file is being processed on two units, and the type-out of information, such as block counts and other controls which are of vital interest to the installation.

From a completely realistic standpoint, it is also necessary to plan for such things as computer malfunction, power failures, and other eventualities which could cause an unscheduled interruption of a program run.

A computer installation must, therefore, design programs which not only detect errors but also provide the facility to correct these errors with a minimum loss of time and effort. One commonly used method of verifying results is to balance developed program totals against pre-established totals at regular intervals during processing. This can be done after a reel or number of reels of tape have been processed, after a certain number of cards have been read, or could very well depend upon a given amount of elapsed running time.

If an error is discovered it will normally be necessary to start the job over again from the very beginning. However, if the run happens to take an hour or several hours it is obviously both expensive and impractical to do so.

In order to avoid complete reprocessing, intermediate "return points" must be set up. These return points usually are constructed by writing the entire contents of computer storage onto magnetic tape storage dumps, thereby preserving the status of the program as of that stage of the run. These storage dumps must also be written at logical stages so that if an error is detected at the next verification point it will only be necessary to process from the last storage dump instead of from the very beginning of the run.

Most installations will also find it necessary to suddenly stop a run because some other program with higher priority must be processed. With periodic storage dump built into the program this Break-Out or Priority Stop is relatively simple to handle. The same thing holds true for Break-In or starting the job over again since in either case the last storage dump can be used as a re-entry point from which to continue processing.

These are some of the problems confronting any installation which is using or contemplates using magnetic tape. **BMTR** is designed to handle these problems.

GLOSSARY

In order to avoid ambiguity, the terms and abbreviations which are most likely to be misinterpreted are defined as they are used in connection with this routine.

Block Count	A tally of the number of blocks read from, or written on, a single lane of tape.
Break-Out	The end-of-day or get-off machine procedure is controlled by one of the console switches; it permits stopping production runs at logical points during normal operation. For instance, Break-Out can be made after an End-of-Lane condition on an output tape, or after a certain number of detail cards have been processed.
Break-In	The opposite of Break-Out. A console switch controls the restoring of the system in order to start processing from the Break-Out point.
Control Data Block	A block of information which: a. Has as its address word, a word of all nines. b. Contains the lane block count for the lane of tape just processed, and the cumulative file count as of the end of that lane.
Cycle	The cycle number distinguishes the output from each run, if a program is run more than once during a working day. a. *Destruction Date* The year and day of the year until which the tape must be retained; after this date it may be over-written or edited. b. *Void Date or Re-use Date* Same as destruction date. c. *Today's Date (Creation Date)* Date which is placed in output tape label. d. *Date Last Processed* Date which input tape was created and appears in input label.
End-of-File	**(EOF)** The condition existing when the End-of-Tape block or control block following the last record of the last lane of tape of a file has been read.
End-of-Lane	**(EOL)** The condition existing when the End-of-Tape block or control block following last record on either lane of an input tape has been read; or, when magnetic End-of-Tape has been reached while writing on either lane of an output tape.
End-of-Job	**(EOJ)** The end of a particular machine run.
EOT-1	The first End-of-Tape block. Control blocks with 7's in the sign digit position of the address word are called "CB's." The CB corresponding to **EOT**-1 is CB-1.
EOT-2	The second End-of-Tape block. The CB corresponding to **EOT**-2 is CB-2.
File Count	A tally of the total number of blocks contained in the reel

	or reels of tape which make up a magnetic-tape file.
K Block	A ten-word block of constants pertaining to each tape file. (See detailed explanation.)
Label	(LBL) A ten-word block of information identifying the type, origin, etc., of the records following it. (See detailed explanation.) The address word of this block is always zeros.
Lane Parallel Operations	A magnetic tape processing technique which uses both lanes of a reel of magnetic tape simultaneously. For instance, the normally active customer data may be placed on one lane of the tape and the secondary or rarely used data, pertaining to the same customer, on the other lane of the tape.
Locator Block	A block which contains a Scan key (identification number). This block precedes information on tape. For example, a locator on a dump tape flags the beginning of a given dump.
Ping-Pong	Programs which process multi-reel files are often designed so that two tape transports are used alternately for each successive reel of a file. This time-saving processing technique is defined as a "ping-pong" operation.
Preblocked Tape	A reel or lane of magnetic tape which has been written on previously and which is going to be "overwritten." Tapes of this type are usually preblocked with fixed-length blocks.
Restart	Machine or operator errors which cause alarm conditions require that a certain portion of a machine run be repeated. The procedure of returning to the previous storage dump and positioning the tapes backward is called Restart.
Save Factor	A numeric quantity representing a number of days. This is used in combination with the creation date of an output tape that produces the date before which the particular tape cannot be destroyed. (Re-Use or Destruction Date)
Sequence Number	Replaces reel number because of the dual-lane feature of BURROUGHS magnetic tape.
Sentinel Block	A block which contains a Scan key (identification number). This block follows information on tape. For example, a sentinel on a data tape containing a dump flags the end of the dump and the beginning of data blocks.
Storage Dump	Entire contents of storage are written out (dumped) following the label on a specified output file or on a special dump tape.

GENERAL DESCRIPTION

BMTR is a general purpose tape-handling routine. It provides an installation with a program which will efficiently handle the majority of beginning and ending tape operations. Because it is general purpose it may not contain certain features or capabilities which a particular installation considers desirable. If such is the case the program can be modified by the installation to suit the situation.

BMTR makes optimum use of the storage cells it occupies. In order to do so many of the routines are "common" in the sense that programing techniques make it possible to use them interchangeably for several purposes. In order to do this, **BMTR** requires that the following be standardized.

1. The format of magnetic-tape labels.
2. The format of magnetic-tape control blocks, the manner in which they must be written on tape, and the locations to which the blocks will transfer control.
3. The format of the Control Data or trailer block located at the end of a lane of tape.
4. The format of the information block which is supplied to **BMTR** by the programer. (This block – called the K Block — specifies file identification, dates, whether the file is an input or output file, and other information vital to the operation of **BMTR**.

BMTR operates on the Constant Block or K Block principle. The programer writes a set of tape-handling constants for each file as an integral part of his program. **BMTR** then uses this K block to handle or control the various tape functions for the following types of files under either single- or multiple-block operations.

1. Lane-serial files using a single lane of tape.
2. Lane-serial files using both lanes of tape sequentially, or alternately.
3. Lane-parallel files.
4. Fixed or variable-length files.
5. Any of the above conditions for initial or overwrite operations.
6. Any of the above conditions for both Tape Storage Units and Datafiles.*

NOTE

Since **BMTR** re-uses areas of its own routine and uses a part of main memory when writing control blocks at end-of-job, each time a program using **BMTR** is run, both the program and **BMTR** must be loaded into memory.

FEATURES OF BMTR

Tape Labels

BMTR provides for checking tape labels both internally and externally. Labels are typed out both as a flag to the Console operator and as a part of the operations log sheet for that day. The program also provides for overriding apparent error conditions in the label by means of Console switches. **BMTR** flags external interventions by means of a Supervisory Printer type-out. Controlling and recording error conditions is an essential part of the **BMTR** label-handling procedure.

Block-Count Verification and Storage Dump

BMTR balances lane block count accumulated by the programer against pre-established lane block count from the Control Data Block (CDB) of the input tape.

Storage dumps can be taken either on a special tape reserved for dumps or on the next tape of the file which reached End-of-Lane. If the "reserved tape" is being used **BMTR** uses a locator block to locate a specific dump in the event of a restart. If the next output tape is the dump tape as well as a data tape, **BMTR** uses a sentinel to position itself past the dump when that tape later becomes an input.

*Trade-mark of Burroughs Corporation

Restart, Break-In and Break-Out

Because of the many ways in which files can be handled, e.g., searched or scanned, it was not possible to design Break-In and Restart routines which would cover every situation.

BMTR uses Console Switch 0 to signal Break-Out. Zero can be set to On at any time during a run and will be queried by **BMTR** after storage is dumped. If Zero is set to On, Break-Out messages will be typed out and tapes rewound, the Break-Out being complete since storage has just been written out. Console Switch 0 is also used for Break-In. When set to On, appropriate messages will be typed out indicating Break-In is taking place.

Restart can be entered after any error condition requiring reprocessing. Briefly, the Restart program reloads memory with the last dump, positions all tapes to the correct processing point, then transfers control to the main program.

Limitations

1. **BMTR** considers any tape file as a continuous one. It makes intermediate checks when End-of-Lane is detected but does not transfer control to the programer's routine until End-of-File is established. At End-of-File it is necessary for the main program to determine if End-of-Job has been reached.
2. The **BMTR** Routine will not handle interspersed control blocks on a lane of tape. If special purpose control blocks are scattered throughout a file the main program must be prepared to handle the resultant transfer of controls.
3. **BMTR** only handles the simplest of Datafile operations, the selective or total updating of the file using externally sequenced activity items.

ABSTRACT AND BMTR CONVENTIONS

The main control features of the BURROUGHS Tape Handling Routine are as follows:

1. Reads input tape labels and verifies file identification and file sequence.
2. Writes output tape labels: As a check prior to writing the label it reads the output label and checks the re-use date and block size, if preblocked tapes are being used; computes new re-use date; writes label; and rereads for verification.
3. Takes a readily identified storage dump when triggered by End-of-Lane condition on selected tapes, and will write the dump on a designated output or special dump tape.
4. Verifies block count when End-of-Lane **(EOL)** is encountered from an input tape if block count is significant. Block count is not significant to this program for files being searched or scanned since the count in the control block will never match the count of records processed.
5. Provides facility for program-controlled Restart, as defined in the Glossary. This is accomplished by calling in a separate routine.
6. Provides manually-controlled Break-In and Break-Out based on the setting of a Program Control switch.
7. Changes unit designations of magnetic-tape instructions if ping-pong (flip-flop) operation is called for. The programer must provide a table of the locations of these instructions.
8. Reads, processes, develops, and writes control blocks (CB's), End-of-Tape **(EOT)** blocks and control information.
9. Provides the installation with the following log of tape operation:
   a. Cycle, sequence number, and date of output files.
   b. Indication of valid or invalid conditions.
   c. Record of corrective actions taken if errors occur.
   d. Messages at End-of-Lane and End-of-Job **(EOJ)**.

Magnetic Tape Format

BMTR assumes that information on magnetic tapes will be arranged in the following format:

1. A ten-word label immediately following Magnetic Beginning of Tape **(MBT)** for all tapes (Figures A, B and C).
2. A Control Data block **(CDB)** sandwiched between two **EOT** blocks at magnetic **EOT** (Figure A). This is a ten-word block.
3. A **CDB** sandwiched between two control blocks (CB) if the end of the file on a tape being *overwritten* occurs at some point prior to magnetic **EOT** (Figure B). This **CDB** is the same size as the other blocks but only the first ten words are significant.
4. A label following Magnetic Beginning of Tape **(MBT)** for edited tapes (Figure C).

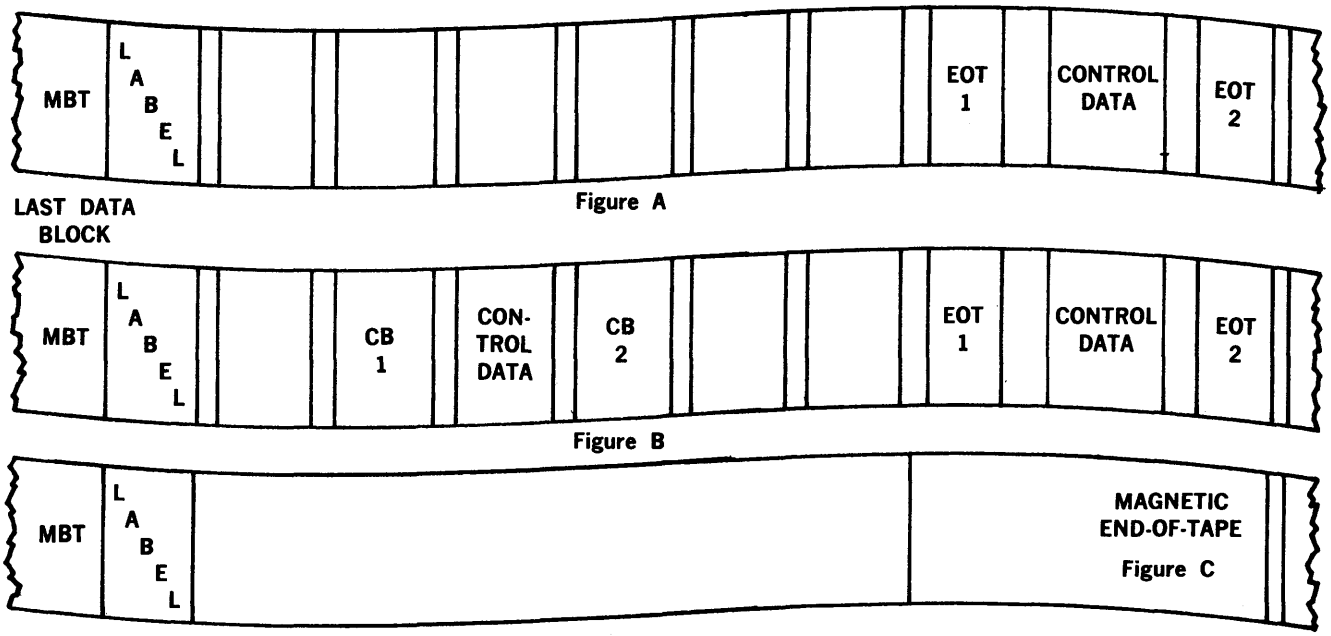


Figure A

Figure B

Figure C

The K Block and Input/Output Tape Labels

THE K BLOCK

The K block is a ten-word, multiple-purpose information block which is written by the programer as an integral part of his program. This information block is referenced by **BMTR** to determine such factors as whether the tape file in question is an input or an output, whether a lane serial or lane parallel operation is called for or whether the tape transport is a **TSU** or a Datafile. **BMTR** also uses the input K block to verify input labels, and the output K blocks as the label for output tapes. In addition, the K block has been designed so that its basic structure will be retained whether it is being used as an input or output K block, as an input or output label, or as the **EOT** or **EOF** Control Data block.

The charts which follow describe the way in which the various elements (fields) of the K block must be set up by the programer depending upon its function as an input or an output K block.

BMTR K BLOCK AND LABEL

	INPUT	OUTPUT
± Word 1 v 0 0 0 0 0 0 0 0 0 0 0	Search Key (All Zeros) V = 0 Normal Rewind V = 1 Rewind off line	Search Key (All Zeros) V = 0 Normal Rewind V = 1 Rewind off line
± Word 2 i	Identification Word (ID) (if security classification is required, include it in the ID).	Identification Word (ID) (if security classification is required, include it in the ID).
± Word 3 A B C D D D E E E E E	A = 1 Input tape B:0 = Lane serial 1 = Lane parallel C:Cycle DDD:Sequence no. (usually 001) EEEEE:Creation date (to be stored by **BMTR** from tape label or supplied by the programer) in the form 2 digit year and 3 digit day.	A = 0 Output tape B:0 = Lane serial 1 = Lane parallel C:Cycle DDD:Sequence no. (usually 001) EEEEE:Creation date (today's date supplied by **BMTR** via **KAD**) in the form 2 digit year and 3 digit day.
± Word 4 F G G H H H I I I I I	F:0 = Read all input 1 = Search or scan GG:09 = Variable preblocked 10-00 = Fixed length preblocked. (Block size.) HHH:Zeros — not significant IIIII:Block count of lane since last storage dump. Programer counts here at all times. **BMTR** will develop total lane count.	F = Zero — not significant GG:09 = Variable-length preblocked tape. 10-00 = Fixed-length preblocked tape. (Block size.) HHH:Save factor (no. of days) IIIII:Block count of lane since last storage dump. Programer counts here at all times. **BMTR** will develop total lane count. NOTE: When writing labels on newly edited tapes, 07 must be entered in sl:22 of label word 4 by the labeling routine.

BMTR K BLOCK AND LABEL (Cont.)

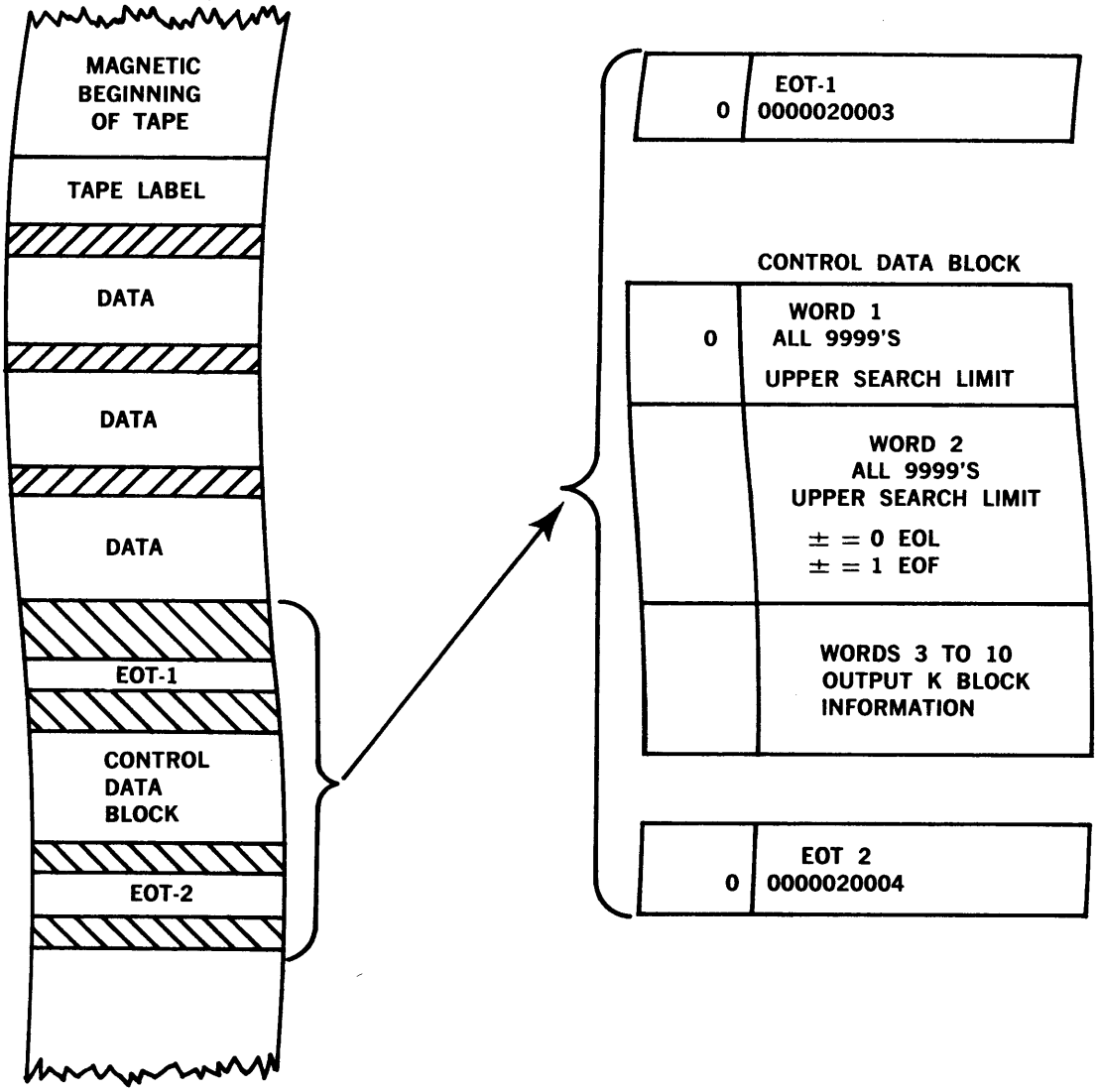
	INPUT	OUTPUT
± Word 5 J K K K L L L M M M M	J = 0 (not significant) KKKLLL: First & second tape unit & lane (ULL) MMMM = Addr. of table of tape instructions (Ping-Pong)	J: 0 = No storage dump at **EOL**. 1 = Storage dump at **EOL** to be written on next output tape. KKKLLL: First & second tape unit & lane (ULL) MMMM: Addr. of table of tape instructions (Ping-Pong)
± Word 6 N Ø Ø Ø P P P Q Q Q Q	N: 0 = TSU 1 = Datafile ØØØPPP: Third & fourth tape unit & lane (ULL) QQQQ: Addr. of programer's EOJ interrogation routine.	N: 0 = TSU 1 = Datafile ØØØPPP: Third & fourth tape unit & lane (ULL) QQQQ: Zeros — not significant.
± Word 7 R S S S T T T T T T T	R = Zero for input SSS: Used by **BMTR** to flag **EOL** conditions TTTTTTT: Block tally of lane (developed by **BMTR** 1). This field must be zeros initially and cannot be disturbed by the programer.	R: 0 = MOW 1 = MIW SSS: Used by **BMTR** to flag **EOL** var. conditions TTTTTTT: Block tally of lane (developed by **BMTR**)
Word 8	File Block Count (developed by **BMTR**)	File Block Count (developed by **BMTR**)
Word 9	Available to the programer	Available to the programer
Word 10	Available to the programer	Available to the programer

NOTES

1. Programer's initial housekeeping must place the address of the first K block in location 0009, SL:04. All K blocks must follow one another in memory.
2. No K block can extend into the last 1000 locations if the programer desires to use **BMTR** Restart or Break-In.
3. The programer must place a word of nines after the *last* K block.
4. Word one of the K block normally contains zeros. If the programer does not want **BMTR** to process a particular K block, his initial housekeeping must replace the zeros with a word of ones.
5. Word five, element MMMM, gives the location of the first entry in a table of addresses of tape instructions required by a file. Each entry contains one address in SL:04. The programer must place a word of zeros at the end of *each* table. If MMMM is set to 0000, it indicates no table for this file.
6. If a storage dump is to be taken on an output master, the K block pertaining to this file should be the last K block.

End-of-Tape or End-of-File Controls

All magnetic tapes except newly edited tapes are bounded by a label and an End-of-Tape block followed by lane controls and a second End-of-Tape block. Below is a diagram of these control blocks.



Control Block Convention

BMTR assumes that all control blocks and **EOT** blocks which affect the routine will always perform the following standard functions:

	STORE C:04 & P IN LOCATION	TRANSFER CONTROL TO LOCATION	FORMAT OF CONTROL WORD
EOT-1 or CB-1	0002	0003	0 00 0002 0003
EOT-2 or CB-2	0002	0004	0 00 0002 0004

When the first **EOT** block is read, the C and P register settings are stored in location 0002 and control is transferred to location 0003 for entry to the End-of-Lane routine, which reads in and processes the Control Data block. Under no circumstances would **EOT**-2 or CB-2 be read by either the main program or by **BMTR**.

If the second **EOT** block is read, either a machine or a programing error has occurred. The C and P register settings are stored in location 0002 and control is transferred to location 0004, an Unconditional Error Halt. Normal processing should never call for a read on the same lane after **EOL** has been reached until the reel has been changed. The block located between the two **EOT** blocks is an updated output label and contains the lane block count and the intermediate file count.

When the last lane of a preblocked output tape has not been completely filled, two control blocks (CB-1 and CB-2) and a block of control data will be written to duplicate the actions of **EOT** blocks as described above.

DETAILED DESCRIPTION OF BMTR

There are three distinct phases to the tape-handling operations which are performed by this program. These phases are the housekeeping operations at the start of the job, the handling of End-of-Lane and End-of-File conditions during the program run and the wrap-up or End-of-Job operations. In addition, a separate program to accomplish Restart or Break-In will be described. Although **BMTR** makes common use of many of these routines in all phases, the description which follows segregates the various operations into distinct groups.

Phase I

After the program has been loaded and the housekeeping functions of the main program have been completed, control is transferred to **BMTR.** This is accomplished by a **STP** 0012 and **BUN** 0006. The latter program immediately calls for keyboard entry of today's date; then proceeds to process tape files based on the K blocks written by the programer.

The ID, cycle, sequence number, and creation date of each lane of each input file is checked against the input K block. If all elements are equal, the program continues. This means the correct creation date has been previously supplied by the programer. If unequal, Program Error Halt 8001 occurs and the operator has the option to accept or reject the reel. If the reel is accepted, ID, cycle, sequence number, and date from the input tape label will be stored in the K block and succeeding reels will be checked against this block (except for sequence number).

Today's date is stored in the output K blocks as the creation date. Output labels are read and the routine verifies that edited tapes and/or preblocked tapes are loaded on the proper units. The creation date and the save factor of the output label are combined to form the re-use or destruction date. This date is compared to today's date to insure that the information written on preblocked tapes is of no further value and may be overwritten. Preblocked tapes are also checked to make sure the block length matches the one specified in the K block. Each label block is reread for verification purposes.

Upon completion of Phase 1, **BMTR** transfers control to location 0012 which in turn returns control to the main program.

Phase II

Phase II is entered automatically when an **EOT** or **EOF** condition is encountered on any input tape or any output tape which is being overwritten. If Initial Write is being used, the programer must determine when magnetic **EOT** has been reached. He then enters **BMTR** by storing in memory cell 0002 (SL:04) the *location* of the instruction following the Initial Write instruction and transfers control to 0003. Phase II performs the following operations:

INPUT TAPE

1. Verifies that the number of blocks read by the main program tallies with the block count in the Control Data block unless the search or scan feature was utilized. See Halt 8008, in the Programed Halt Section which explains the handling of an out-of-balance condition.
2. Checks to see if a storage dump is to be written on a special dump tape.
3. Transfers control to the programer's **EOJ** interrogation routine if **EOF** has been reached.
4. Reads the label of the next lane of tape and verifies the ID, cycle number, and date, and also verifies that the sequence number is one greater than the previous lane. If these conditions are not met, the label elements are typed out for visual verification.
5. Positions past an old storage dump if there is one following the label.

OUTPUT TAPE

1. Writes **EOT** blocks and a Control Data block containing the number of blocks written on this lane. The Control Data block is an updated output K block.
2. Types on the Supervisory Printer the complete identification number, cycle, sequence number, and date for use as an external label for each lane of tape. The block count is typed out; also word nine of the K block which is available to the programer, for storing last item number or similar data to be put on the external label.
3. Makes the same checks as in Phase I and writes a label on the next lane.
4. Writes a storage dump, if specified, following the label of the next lane or on a special dump tape according to element J of the K block.

INPUT AND OUTPUT

1. Rewinds or rewinds and deactivates **EOL** tapes; types a message to the operator to change tape if necessary.
2. Changes unit designation of Read or Write instructions in the main program (according to programer's Table-of-Addresses) if ping-pong operation is indicated.
3. Preserves registers and indicators since normal entrance to Phase II is not under control of the programer.

STORAGE DUMP AND BREAK-OUT

1. As indicated before, the program may designate a storage dump at **EOL** on any output tape. The tape dump will be written following the label on the next output lane. If this option is used, no other storage dumps are available to the programer and Restart, which repositions tapes backward, cannot be used. Break-In, a form of Restart which repositions forward, must be used. (This will be explained in detail in the Restart and Break-In section which follows Phase III.)
2. If *no* output file is designated by the programer as a dump tape, **BMTR** will assume a special dump tape is available. Storage dumps will be taken at **EOL** on *all* input and output files. In addition, the programer can obtain a storage dump on this special tape at any time by executing a **STP** 0012, **BUN** 0008. (Return point to the main program in the event of Restart would be the next instruction.)
3. Since the registers and indicators have been preserved, they will be available in any storage dump and will be restored by the Restart or Break-In program.
4. Since PCS 0 is queried immediately after storage is written, the Break-Out option is available after every storage dump.
5. The special dump tape used by **BMTR** is addressed as Unit 10, even lane. No K block is required for this tape. The label block will be conventional.
6. Since **BMTR** writes all storage dumps in 100-word blocks, output data tapes receiving dumps must, if preblocked, contain 50 blocks of this size followed by a ten-word block for sentinel purposes. Fifty-one such blocks must precede any other size. The label on an output master which contains a dump readily identifies it (element J).

Phase III

Phase III is entered after the main program has reached **EOJ** and after all summaries, statistics, etc., have been completed. Entrance to **BMTR** wrap-up is accomplished by a **BUN** 0007. The functions performed are as follows:

1. Rewinds and deactivates all input units on line.
2. Writes **EOT** blocks on files being initially written or writes control blocks on all output tapes.
3. Types out labels and block counts as outlined in Phase II (Output Tape, Item 2).
4. Rewinds and deactivates all output units on line.

Restart and Break-In

Particular attention must be paid to the distinct differences between the procedures described as Restart and Break-In. The reason for differentiating between the two becomes obvious when the problem of locating the correct reel or reels of the secondary files is considered. For instance, processing one reel of Master File tape may involve posting several reels of activity items. If the output storage dump is being used, and a failure requires return to the last dump (which is at the beginning of the output tape), it is necessary to verify tape labels on the several activity input reels to locate the one which was being processed at the point the dump was taken. On the other hand, if storage dumps are taken on a separate dump tape each time an **EOL** condition is encountered, it is only necessary to go back to the last storage dump to reconstruct memory and reposition tapes. Because of this and related problems, the procedure described below requires that programers carefully differentiate between the two methods of dumping storage. Obviously, the advantages of the one method over the other will be determined by the requirements of each individual application.

Restart and Break-In are accomplished by a separate program which is placed in memory locations 4000 and following, only when Restart or Break-In is called for. If PCS 0 is On, Break-In will take place; if it is Off Restart will be executed. Both procedures use a prior storage dump to reposition tapes and to restore memory, the registers, and the indicators. After completion of Restart or Break-In, control is returned to the Branch Unconditional instruction stored in Location 0010.

RESTART

Restart repositions all input and output tapes in a *backward* direction. In order to employ it, the scratch dump tape method *must be used* since the Restart Routine presumes that no tapes require changing and that storage dumps have been written at **EOL** on all input and output files. Entrance to Restart after an error condition *must* be on the basis that the tapes involved are still mounted *but not manually rewound.* If the tapes have been manually disturbed in any way Break-In must be employed. If Restart has been selected (PCS 0 Off), a Restart message will be printed and a halt will immediately follow.

BREAK-IN

The unit, lane, and number of the *last* dump are supplied to the Break-In Routine via a **KAD** operation. This information was typed out by **BMTR** after each storage dump and must be taken from the Supervisory Printer Log Sheet.

Break-In repositions all tapes in a *forward* direction. It may be employed with *any* storage dump whether on a special tape or on the beginning of an output data tape.

The tapes involved at the time of the selected dump must be remounted as indicated on the log sheet.

Limitations When Using Overwrite and Initial Write

OVERWRITE

When overwriting preblocked tapes, no problems are encountered when breaking out, breaking in or restarting.

INITIAL WRITE

When initial writing on edited tapes no problems are encountered when breaking in at the same dump that was taken at Break-Out. Restart, however, requires that the following be done to any output tape file which has been processed since the last storage dump:

1. The tape should be positioned back to the point at which the last storage dump was taken.
2. That CB-1 be laid down over that block. That the **CDB** be laid down as the next block and that CB-2 be laid down as the third block.

   Obviously there is the possibility that no blocks have been written on this file, or that less than three blocks have been initially written since the last storage dump. **BMTR** checks the block count for each output file. If no blocks have been written, no control blocks will be laid down, no messages will be typed, and the tape need not be disturbed. If less than three blocks have been written **BMTR** will overwrite and initial write the necessary control blocks. If more than three blocks have been written **BMTR** will overwrite the three control blocks. In the last two cases End-of-Lane messages will be typed and the tapes rewound. At this point the console operator must determine from the printout whether the tape is to be immediately dismounted or whether the tape is to be left on so that the alternate lane can be used.

THE "HOW" OF BREAK-IN

When Break-In is called for, the contents of registers and storage are normally insignificant. Based on this assumption the following occurs during Break-In *(on a 5000 word machine)*:

1. Program Control Switch 0 is set to On by the Console operator.
2. The Restart-Break-In Routine is loaded into successively higher locations starting at location 4000 from cards, paper tape, or magnetic tape. The medium on which the program is stored depends on the operating system being used at the installation.
3. After the program is loaded and control transferred to location 4000 the Break-In Routine takes over and does the following:
   a. Locates the dump specified by the **KAD** (keyboard add) instruction.
   b. Reads in the dump label and verifies that the specified dump has been correctly located.
   c. Reloads the first 4000 words of storage.
   d. Repositions the input and output tapes using **BMTR** lane total tally (Element TTTTT), after the tape labels are verified.
   e. Reads in the last 1000 words (overloads the Restart-Break-In Program). Then transfers control to **BMTR** where memory is sum checked and the registers and indicators are restored, prior to entry to the main program or auxiliary housekeeping (Location 0010).

WHAT IS REQUIRED OF THE PROGRAMER USING BMTR

1. Program switches 0 and 9 are reserved for **BMTR** and must *not* be used.
2. The programer must write the K blocks according to the format designated.
3. Block count for any file is kept in word 4 of the K block for the particular file element. (IIIII)
4. When an **EOF** condition is reached, **BMTR** transfers to the End-of-Job Determination Routine. The input area will contain the Control Data block – the first two words being all nines. This is usually helpful to force other input files to be read until they reach **EOF**. When all input areas tested are all nines, a simple program can be written for true **EOJ** determination. At this point the programer finds out whether or not **EOJ** has been reached. When **EOJ** is established, the programer must branch unconditionally to location 0007 which contains the entry to **BMTR** *End-of-Job Routine.*

   If not **EOJ**, the programer's routine must branch to location 0012 which contains a branch back to the main program to continue processing. (When the controlling file reaches **EOF** the programer must provide for completing the processing of pertinent input files.)
5. The programer cannot have any rewind instructions in his program which reference tapes for which K blocks have been provided. All rewind instructions are automatically provided by **BMTR.**
6. The programer must, via his own program, supply to the K block today's cycle, sequence number, and date if these are to be varied from standard.
7. It is important to note that **BMTR** Break-In and Restart procedures can be used only with **BMTR** versions A and B.
8. The main program must never alter any location within **BMTR** except as provided by the standard linkage. A summary of the linkage follows:

   a. To enter Phase I (Initial Housekeeping) insert the following instructions:

   STP 0012

   BUN 0006

   b. To enter Phase II (**EOL** on tape which is being initially written):
   Store in 0002 (SL:04) the address of the instruction following the Initial Write instruction and **BUN** 0003. A routine to detect **EOT** and accomplish the above linkage should appear as follows:

xxxx	MIB	xxxx+2	Test unit for ready status.
xxxx+1	BUN	xxxx	
xxxx+2	MIE	xxxx+6	Test for magnetic **EOT.**
xxxx+3	IFL	aaaa	Add to block count.
xxxx+4	MIW	bbbb	
xxxx+5	BUN	cccc	To continue processing.
xxxx+6	CAD	xxxx+9	
xxxx+7	STA	0002	Partial word, SL:04.
xxxx+8	BUN	0003	Enter **BMTR.**
xxxx+9	HLT	xxxx+5	Constant.

After completing **EOL** operations, **BMTR** will return control to xxxx+3.

c. To enter Phase III (**EOJ**) branch to location 0007 (**BUN** 0007). If **EOF** but not **EOJ** has been reached, execute **BUN** 0012.

d. To obtain storage dump with Break-Out option, insert the following two instructions:

STP 0012

BUN 0008

e. K blocks may be located anywhere in storage except the last 1000 cells. **BMTR** must know where to find the first one. Location 0009 is reserved for a constant which must be inserted. This constant is the address of the first word of the first K block.

f. **BMTR** uses 0012 as its general exit line.

g. Upon completion of **BMTR** Break-In or Restart, a branch will be made to location 0010 which contains a branch to 0011 to return to the main program. If there are special housekeeping routines to be accomplished such as reloading format bands, card files, etc., the address should be changed in location 0010 to cause a branch to the housekeeping routine. The last instruction of this housekeeping routine should be a branch to location 0011.

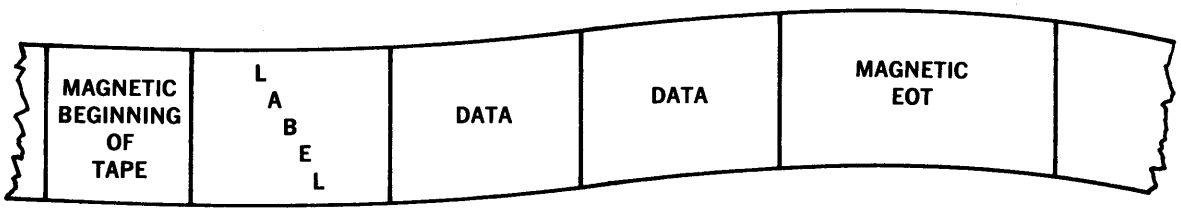
h. The sum check for storage dump is placed in the last location of memory.

TABLE B-1. BMTR LINKAGE LOCATIONS		
LOCATION	FUNCTION	PROGRAMER ACTION
0002	a. rC:04 and rP are stored in this word by both CB's and **EOTB's.** b. During Initial Write, a **STP** is simulated in this location.	Must not contain significant information.
0003	Contains an Unconditional Branch instruction to **BMTR,** Phase II.	Must not be changed. Branch instruction is loaded as part of **BMTR.** If **MIW** is used, **MIE** subroutine branches to 0003.
0004	Contains an Unconditional Halt to which control is transferred if CB-2 or **EOT**-2 is erroneously read.	Must not be changed. Loaded as part of **BMTR.**
0005	Contains an entry from Break-In and Restart programs to the main **BMTR** program.	Must not be changed. Loaded as part of **BMTR.**
0006	Contains the entry to **BMTR,** Phase I.	a. Must not be changed. Loaded as part of **BMTR.** b. Branch to this location after completion of housekeeping. Location 0006 is referenced only once during each run.
0007	Contains the entry to **BMTR** Phase III (End-of-Job)	a. Must not be changed. Branch Instruction is loaded as part of **BMTR.** b. After true **EOJ** has been determined a branch to location 0007 is executed.
0008	Contains the entry to the storage dump routine. **STP** 0012 **BUN** 0008	a. Must not be changed. Branch instruction is loaded as part of **BMTR.** b. Branch to location 0008 at any point in the program to obtain a storage dump.
0009	Contains the actual address of the first word of the K block area.	Programer *must* overload (Patch) this location after assembly. Unless patch is inserted all 9's constant will not be overloaded and alarm stop will occur when attempting to move first K block to the work area.

LOCATION	FUNCTION	PROGRAMER ACTION
0010	Loaded as a Branch Unconditional to 0011. (Restart/Break-In re-entry location)	Must not be changed — unless to execute special subroutine after Break-In or Restart — prior to re-entering main program. For instance, cards may have to be repositioned or format bands reloaded. In the latter situation this location must be patched to branch to the special subroutine. Exit from the subroutine must be a branch to Location 0011.
0011	Contains the entry to continue processing after auxiliary house-keeping, following a Break-In or Restart.	Must not be changed. Branch instruction is loaded as part of **BMTR**.
0012	**BMTR** Exit location	Prior to entering Phase I or the separate storage dump routine, provide for re-entry to the main program by storing the address of the next instruction to be executed in location 0012.

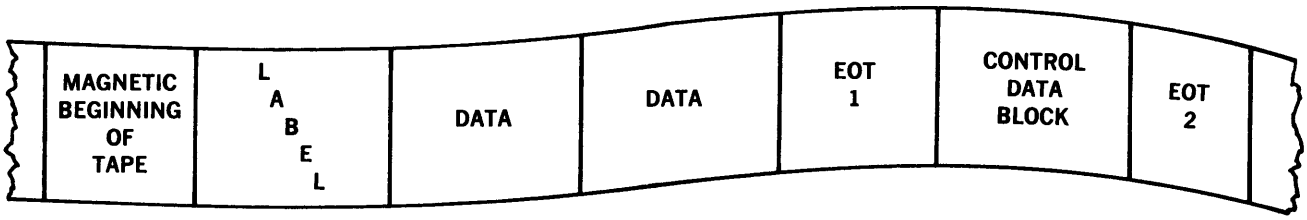
END-OF-LANE AND END-OF-FILE CONDITIONS HANDLED BY BMTR

End-of-Lane on an Edited Tape



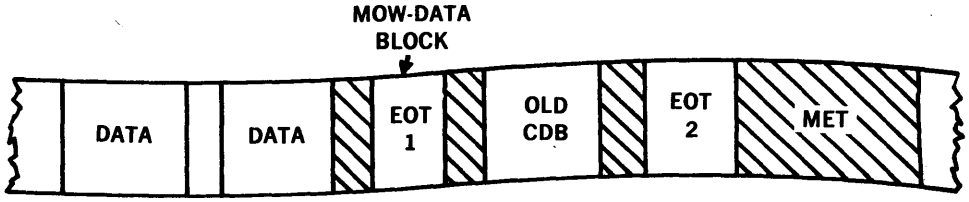
BMTR has been entered as a result of an **MIE** (Magnetic Tape Interrogate End-of-Tape indicator) and functions as follows:

1. Informs Console operator **EOL** occurred. **(SPO)**
2. Writes 10-word Control Data block sandwiched between **EOT**-1 and **EOT**-2 and ends up with a tape having the following format:



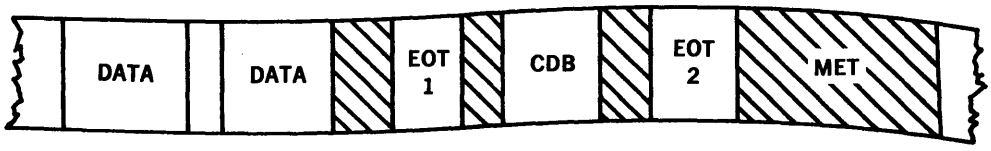
3. **BMTR** then reads the label on next output tape and verifies that the tape can be used. If so, it writes a new label and transfers control to the main program at the instruction after the **MIE**.

End-of-Lane on a Preblocked Tape



The overwrite instructions attempt to lay down a data block over **EOT-1**. **BMTR** handles the resultant transfer of control as follows:

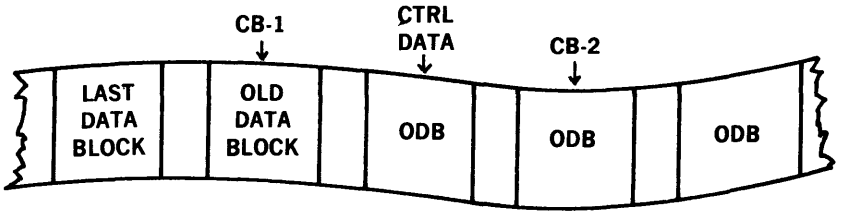
1. Informs the Console operator **EOL** has occurred. **(SPO)**
2. Adjusts the programer's lane block count total.
3. Writes control data in designated area ending up with a tape having the following format:



4. **BMTR** verifies that the next reel of output can be used. It then writes the tape label and the data block or blocks necessary (if multiple block writes are used) to complete the instructions; after this control is transferred to the main program—at the instruction following the **MOW**.

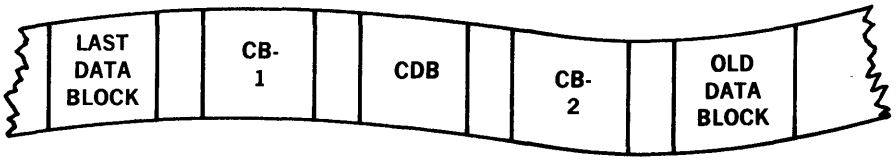
End-of-File on a Preblocked Tape

CASE A. NORMAL END-OF-FILE



Control is transferred to Phase III (End-of-Job) of **BMTR** from the programer's End-of-Job Interrogation Routine after the program has determined that it is at **EOJ. BMTR** then does the following:

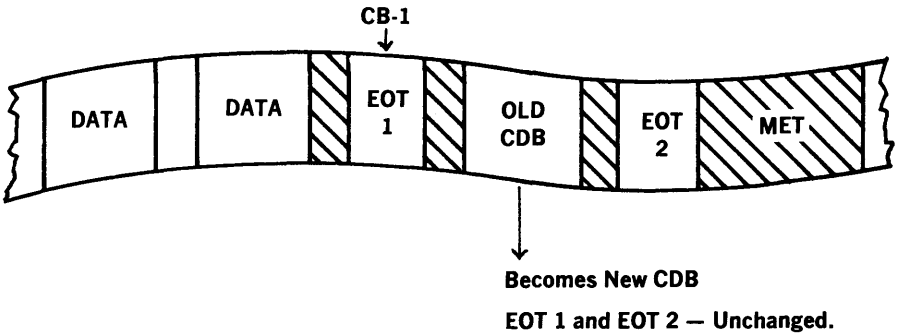
1. Writes CB-1.
2. Writes the Control Data Block. This block will conform in size to the preblocked file size and will have the control data as its first ten words.
3. Writes CB-2 and results with a tape having the following format:



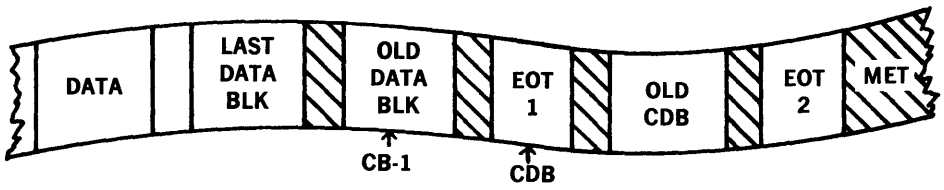
CASE B. END-OF-FILE COINCIDES WITH END-OF-LANE. IN THIS SITUATION AN ATTEMPT WILL BE MADE TO OVERWRITE EOT-1 WITH CB-1.

This situation only arises after it has been determined that **EOJ** has been reached. Control is transferred to Phase III of **BMTR. BMTR** immediately sets Switch 3 in the **EOL** routine which will do the following if **EOT**-1 is encountered:

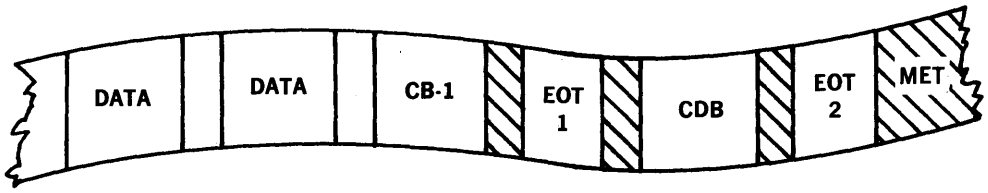
1. Set up to write controls as a ten-word block between **EOT**-1 and **EOT**-2.
2. Set a switch (9) to bypass writing CB-2.
3. Bypass writing CB-1 by transferring directly to the instruction which lays down the Control Data block.



CASE C. END-OF-FILE AND END-OF-LANE SO COINCIDE THAT AN ATTEMPT IS MADE TO OVERWRITE EOT-1 WITH THE CONTROL DATA BLOCK (CDB).

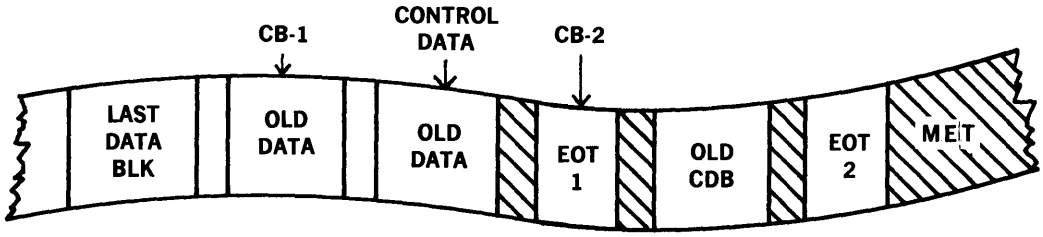


BMTR uses Switch 3 in exactly the same way as it did for Case B, with the resulting tape having the following format:



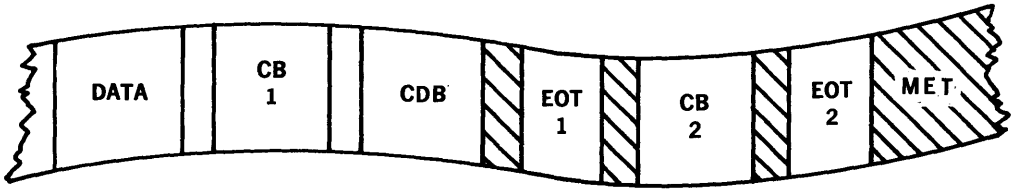
There are two continuous blocks which will cause transfer of control. This is defined as "double End-of-Lane." **BMTR** provides for bypassing the second Block (**EOT-1**) when such a tape is being read. After the first transfer of control is encountered Switch 3 is set to disregard a second such transfer if it should occur.

CASE D. END-OF-FILE AND END-OF-LANE SO COINCIDE THAT AN ATTEMPT IS MADE TO OVERWRITE EOT-1 WITH CB-2.



As in Cases B and C, **BMTR** uses Switch 3 to control this condition if it occurs. After CB-1 and the Control Data block are laid down, the switch is conditioned to exit to a routine which does the following, in the event of a transfer of control when overwriting CB-2:

1. Set up ten-word CB to duplicate the actions of CB-2 (Control word: 0 0000020004).
2. Write this ten-word CB in the Control Data area on tape.
3. Bypass (9) the writing of CB-2 and the result is a tape having the following format:



OPERATING INSTRUCTIONS FOR BMTR

Break-In and Restart

EQUIPMENT REQUIREMENTS

1. One Photoelectric Reader.
2. One Supervisory Printer.
3. The same number of magnetic tape units which were being used when the storage dump was taken.

BILL OF MATERIALS

1. Input forms.
   The **BMTR** Break-In and Restart Routine on Paper Tape. If card input is used for the program being restarted, prepare cards so that the correct card will be read upon Restart or Break-In.
2. Output Forms.
   Paper for the Supervisory Printer.
3. Magnetic Tapes.
   For Restart, the same tapes which are already mounted. These tapes must not be disturbed in any way.

   For Break-In, the tapes which were on line when the storage dump, to which the programer wishes to return, was made.

OPERATOR PROCEDURES

Equipment Setup

1. Punched Paper Tape.
   a. Designate the Photoreader as Unit 1.
   b. Set the reading speed to High.
   c. Mount the **BMTR** Break-In and Restart Routine.
   d. Place the unit into the Remote mode of operation.
2. Supervisory Printer.
   a. Load the Supervisory Printer with the proper output form.
   b. Place it in the Remote mode of operation.
   c. Designate it as the **SPO.**
   d. For further details, refer to the standard installation procedures for setting up the **SPO.**
3. Magnetic Tapes.
   For Restart, the same tapes which are already mounted. These tapes must not be disturbed in any way.

   For Break-In, remount the tapes which were on line when the storage dump, to which the programer wished to return, was made.
4. Prepare the rest of the system in accordance with the Operating Instructions for the program which is being restarted or upon which Break-In is being used.
5. Program Control Switches.

PCS	ON	OFF
#0	Break-In will be executed	Restart will be executed

Running the Program

1. Complete the Equipment Setup.
2. By means of the keyboard, enter the instruction (1000 04 0000) into rD. Transfer rD to rC.
3. Make sure that Program Control Switch 0 is set properly. Depress the Start button. The **BMTR** Break-In and Restart Routine will now run automatically until completion or interruption by a supervisory printout.

The operator will respond to all messages and programed halts, taking the proper action as indicated on the following charts.

SUPERVISORY MESSAGES AND PROGRAMED HALTS

MESSAGE	EXPLANATION
Set PCS 0 on for Break-In, off for Restart (HLT 8998)	Self-explanatory
Break-In, Set PCS 0 off	This is done to insure that an inadvertent Break-Out will not occur when taking the next storage dump.
Dump T LN (KAD 0000 08 0000)	Enter the 3-digit dump number to which the program will return and the ULL of dump tape. U = Unit LL = Lane In SL:06.
Returning to Dump NNN on Tape_____ Lane_____	For verification purposes.
T_____ LN_____	The number of the Tape Storage Unit and its lane, which is being positioned by Break-In.
KBL_____ LBL_____ LBL 8101 (HLT 8101)	If an incorrect reel of tape has been mounted, the K block's ID, cycle number, sequence, and date plus the Tape Label's ID, cycle number, sequence, and date will be printed. Remount correct tape and depress Start.
Scan (HLT 8013) or (HLT 8011)	See Halt Register for explanation.
Hash (HLT 8100)	Memory has been loaded but Hash Total does not agree. Depress Start to recompute Hash Total. If it still fails, try Break-In again.

HALT REGISTER			
HALT NO.	F/C PAGE NO.	CAUSE	CORRECTIVE ACTION
8001	1	Input I.D., cycle, sequence no., or date of the label do not match the corresponding K block elements. Printout displays the I.D., cycle, sequence no., and date read from the label.	1. Console operator must visually verify that the ID, cycle, sequence no., and/or date for the file is the correct one. If it is correct, depress the Start switch to continue processing labels. The ID, cycle, sequence no., and date from the label will be stored in the corresponding elements of the K block. Subsequent labels will be compared with this. 2. If the wrong tape has been loaded, remove reel, load the correct reel, turn PCS 9 on, and depress the Start key.
8002	2	Block size indicated in the preblocked tape label is not the same as specified by the K block.	1. Remove tape. 2. Reload TSU with a reel of preblocked tape whose block length matches that specified by the appropriate K block. 3. Depress the Start switch to transfer to the label routine.
8003	2	Programer has called for an edited tape. Edit/block-length indicator in the label is other than 07 (edited indicator).	1. Remove tape. 2. Reload TSU with an edited reel of tape. 3. Depress the Start switch to verify the label.
8004	2	Re-use or destroy date of preblocked output tape is not equal to or less than the current date.	Console operator has two alternatives. a. Override error condition (not recommended). 1. Turn PCS 9 on. 2. Depress the Start switch. b. Change tape. 1. Turn PCS 9 off. 2. Remove invalid tape from TSU. 3. Reload TSU with a preblocked tape of the same block length and a re-use date less than the current date.

HALT REGISTER			
HALT NO.	F/C PAGE NO.	CAUSE	CORRECTIVE ACTION
8005	8	Dump entered via KAD instruction not on the dump tape.	Verify previous run's **SPO** sheet that there is a dump XXX on that tape and lane.
8007	3	No K block for the tape which has reached End-of-File as a result of one of the following: 1. The K block has been modified incorrectly by the programer. 2. The computer is not functioning correctly. 3. The programer did not provide a complete table of ping-pong addresses. 4. The programer did not provide the address of his table of addresses in word 5 of the K block.	Dump program and desk audit results.
8008	3	Block count read from tape does not equal the count developed by the programer.	Depress the Start key to ignore the mismatching and continue normally.
8009	4	Next lane to be read or written is on the same reel as the lane which has reached an End-of-Lane condition.	1. Wait until tape has rewound. 2. Depress the Start switch.
8011	9	Positioning failure when attempting to bypass a check point on an input tape when the sentinel did not equal that of the Scan key in memory.	Depressing the Start switch will cause a Retry to be attempted.
8012		An attempt to develop an address equal to the C register setting stored in location 0002 (SL:64) has been unsuccessful. The address is developed	

HALT REGISTER			
HALT NO.	F/C PAGE NO.	CAUSE	CORRECTIVE ACTION
		by repeatedly incrementing the original address of the instruction which caused the End-of-Lane condition by the block length—as indicated by the appropriate K block—until an equal comparison is obtained. (The address of the tape command must be register-B modified, if applicable.) There are three principal reasons for this occurrence: 1. A computer malfunction while storing SL:04 positions of register C when an End-of-Lane was encountered. 2. Incorrect block size was specified by the K block. 3. The control word of a control block does not specify location 0002 as the one in which the control information is to be stored.	1. Prove the condition and call the customer engineer. 2. If in error, correct the block size in the K block and depress Start. This will cause a change of control to symbolic location 6.0002.0. For version A this location is 0131; for version B 0110; for version C 0095; for version D 0102. 3. Correct the control block. Place the proper information in location 0002. Change control to symbolic location 6.0002.0 as indicated above.
8013	7	Storage was dumped or reread improperly.	Try Break-In procedure again.
8100	9	Hash total error after storage has been reloaded for a Restart or Break-In.	Depress the Start switch to retry Hash Total.
8101	9	Label mismatch	Reload TSU with a tape containing the correct ID, cycle, or sequence date as printed on the Supervisory Printer. Depress the Start key.
8102	9	All tapes have been repositioned for a Restart or Break-In.	Depress the Start switch to continue processing.
8200	8	PCS 0 on to select Break-In.	1. If Break-In is desired, set PCS 0 Off and Start. 2. If Restart is desired, set PCS to Off and manually branch to location 4000.

HALT NO.	F/C PAGE NO.	CAUSE	CORRECTIVE ACTION
8202		PCS Ø was not set to Off at Halt 8200, when Break-In was selected.	Set PCS Ø to Off and depress Start switch to continue with the Break-In Routine.
8888	5	Break-Out completed.	Dismount tapes and save for Break-In.
8998	7	Ready to Restart or Break-In.	Depress the Start switch to enter the Restart-Break-In Routine.
9999	6	End-of-Job.	Self-explanatory.

SUPERVISORY PRINTER MESSAGES

In order to conserve memory for the main program it was necessary to limit the length of intermittent printouts for **BMTR.** This will not handicap the console operator because the messages are the same for all programs using **BMTR.**

Prior to most messages, the tape and lane designation will be printed.

When an end of lane occurs the message **EOL** will be printed prior to the tape and lane message to signify this condition.

Every program should have a list of each K block and the machine address assigned to each element. This will simplify the verification of error conditions for the console operator and save valuable machine time.

A brief explanation of each message follows:

HALT	PRINTOUT	EXPLANATION
8001	(First two label words)	ID, cycle, sequence no., and date of label do not correspond with corresponding elements in words 2 and 3 of the K block. Set PCS 9 Off and depress the Start switch to store the label elements in the K block or set PCS 9 On, load the correct reel on the TSU and depress the Start switch to verify the new label.
8002	BLK	The block size indicated in word 4 of the preblocked tape label is not the same as specified in word 4 SL:22 of the K block. Load the TSU with a tape of the correct block size and depress the Start switch.
8003	EDIT	An edited tape has not been loaded on the TSU. The edit indicator 07 must appear in word 4 SL:22 of the label. Load the TSU with an edited tape and depress the Start switch.
8004	DATE	An output tape that is to be overwritten does not have a re-use date equal to or less than the current date. If it is desired to override the warning, set PCS 9 On and depress the Start switch. Otherwise load the TSU with another tape that should be old enough, set PCS 9 Off and depress the Start switch to verify the new label.
8005	NO DUMP XXX	Dump XXX has not been located on the separate dump tape. Examine the previous run's SPO sheet to verify that dump XXX was written on the dump tape. Depress the Start switch to re-scan the same tape if valid. Otherwise reset PCS 0 for Break-In, reload the Break-In program and **KAD** the correct dump number when called for.

HALT	PRINTOUT	EXPLANATION
8007	NO KB	**BMTR** has been unable to locate a K block for the tape that reached end of lane. Location 0002 SL:04 contains the address, plus 1, of the Read or Write instruction which caused the EOL condition. Visually examine word 5 SL:11 of each K block to try and determine the cause. Some possibilities are: 1. No K block established by the programer for this tape. 2. The K block has been modified or established incorrectly by the programer. 3. The programer did not provide a table of ping-pong addresses in word 5 SL:04 of the appropriate K block. 4. The table of ping-pong addresses is incomplete. 5. Machine malfunction. Dump the program and desk audit to locate the applicable condition.
8008	CTRL KBL-XXXXXXXXXX TLR-XXXXXXXXXX	The input lane block count developed by **BMTR** in K block word 7 does not match the total in control data block word 8. The two totals are typed out on the Supervisory Printer. Correct the sign position of K block word 4 in the K block work area if a block count was not to be developed. Depress the Start switch to continue processing in either case.
8009	RMV WAIT or RWD ONLY WAIT	The next lane to be read or written is mounted or is to be mounted on the same TSU when the rewind is complete. Depress the Start switch if the message indicates rewind only. Otherwise remove the reel, mount the next reel and depress the Start switch.
8100	HASH	Storage has been reloaded after a Restart or Break-In has been initiated but the Hash Total of memory is not correct. Depress the Start switch to retry the Hash Total Routine. If the Hash error is repeated it is necessary to retry Break-In. Restart cannot be re-initiated.
8101	KBL (ID, SEQ. AND DATE) LBL (ID, SEQ. AND DATE) LBL 8101	An incorrect tape has been loaded for Break-In. Two words of the K block and two words of the label are printed out to show the tape that should be mounted and the tape that was incorrectly mounted. Reload the TSU with the correct reel and depress the Start switch.
8200	BREAK-IN SET PCS 0 OFF	PCS was set On to enter Break-In. It is now necessary to set PCS 0 Off so that it will not be queried at a later time and be set improperly.
8202	SET PCS 0 OFF	PCS 0 was not set to Off at Halt 8200. Set PCS 0 Off at this time and depress the Start switch to enter Break-In.
8888	B/O etc.	This printout, followed by the labels of all tapes that were on-line at Break-Out, indicates which tapes are to be mounted at Break-In.
8998	SET PCS 0 ON FOR BREAK-IN, OFF FOR RESTART	The Break-In/Restart program has been read in successfully. Set PCS as desired and depress the Start switch to enter Break-In or Restart.
9999	EOJ	End-of-Job. Remove tapes, etc.

SWITCH REGISTER			
SWITCH NUMBER	FLOW CHART	FUNCTION	SET OR RESET
Switch 1	Page 2	One-time Switch — Functions during housekeeping as the loop control to continue "first time" or initial processing of tape labels. After the input labels have been verified and output labels written the switch is set to bypass loop return to the label routine. Transfers control to the main program.	*Set to B* — Page 1 — Conn. 2 *Reset* — Not reset "one-time" switch
Switch 2	Page 2	End-of-Lane Switch — Set when reading or overwriting so that Read or Overwrite instructions which encounter control or **EOT** blocks can be completed by the **BMTR** Routine. The Switch 2 subroutine handles single- or multiple-block reads or overwrites.	*Set to B* — Page 2 — Following switch Page 3 — Following **EOF** determination *Set to A* — After determining that the file is to be read or overwritten. Page 3 — Conn. 3E
Switch 3	Page 3	Multiple Exit Switch — Used to control unique End-of-Lane conditions produced by encountering **EOT**-1 when attempting to overwrite CB's or control information. Exit 3A — Normal exit Exit 3B — Set to 3B when overwriting CB's or control information during **EOJ**. If **EOT**-1 is encountered when overwriting CB-1, exit 3B bypasses writing of CB-1 and CB-2 and writes the controls as a ten-word block between the **EOT** blocks. Also functions as above if **EOT**-1 is encountered when trying to overwrite controls. This latter condition causes "double **EOL**" (CB-1 followed by **EOT**-1, followed by controls, followed by **EOT**-2). Exit 3C — Set to 3C when **EOL** encountered during any Read instruction. This exit provides for "double **EOL**" when reading. Bypasses both CB-1 and **EOT**-1 and permits normal read of control totals into the input area.	

SWITCH REGISTER			
SWITCH NUMBER	FLOW CHART	FUNCTION	SET OR RESET
Switch 3	Page 3	Exit 3D — Set at **EOJ** prior to overwriting CB-2. If **EOT**-1 is encountered by this overwrite, exit 3D provides for manufacturing and writing a ten-word CB in the controls information area. This eliminates necessity for overwriting CB-1 with a CB-2 image. Exit 3E — Set to 3E prior to reading the blocks from the next lane of tape to complete the programer's Read instruction. If **EOT**-1 is encountered by this read, Exit 3E prevents **BMTR** from moving a nonupdated K block to the work area which would cause a block count mismatch between the K block and the Control Data block counts. (See Special Conditions Section)	
Switch 4	Page 2	Alternator switch — Functions when writing output labels on lane-parallel tapes. Forces the writing and checking identical tape labels on both lanes of parallel tape files.	*Set to A* — P.2 After Exiting from B side. *Set to B* — P.2 After Exiting from A side.
Switch 5	Page 4	One-time End-of-File Switch — Set to bypass normal **EOL** rewind, ping-pong and so forth. When set to B, transfers control to the **EOJ** wrap-up routine.	*Set to B.* Page 6 — Conn. 18. *Reset* — Not reset — one-time switch.
Switch 7	Page 4	One-time End-of-Job Switch — Set to B at the start of End-of-Job routine. Bypasses normal **EOL** overwrite of control information and transfers to special-purpose routine which provides for abnormal **EOL** conditions.	*Set to B* Page 6 — Conn. 18. *Reset* — Not reset.

SWITCH REGISTER			
SWITCH NUMBER	FLOW CHART	FUNCTION	SET OR RESET
Switch 8	Page 4	Alternator Switch — Functions after **EOL** on lane-parallel tape files forcing the writing of identical output control totals in both control totals blocks of a lane-parallel tape.	*Set to B* Page 4 — Conn. 10. *Reset* Page 4 — Conn. 10.
Switch 9	Page 6	End-of-Job Switch — Set to B when a transfer of control is caused by trying to overwrite **EOT**-1 with CB-1. (See Special Conditions Section)	*Set to B* Page 3 — Conn. 3D *Reset* Page 6 — Conn. 22.
Switch 10	Page 4	One-time Switch — Set at End-of-Phase I if it is determined that storage dumps will be made on output masters rather than on a separate dump tape. Prevents a storage dump to be taken on other than the output master tape.	*Set to A* Page 5 — Conn. 14. *Reset* — Not reset — one-time switch.
Switch 11	Page 5	One-time Switch — Set at End-of-Phase I if it is determined that storage dumps will be made on an output master rather than on a separate dump tape. Allows a sentinel to be written on the output master immediately after the storage dump.	*Set to B* Page 5 — Conn. 14. *Reset* — Not reset — one-time.
Switch 13	Page 5	Set to A after 100 storage dumps have been made to permit the label of the next separate dump tape to be overwritten when required at the next storage dump.	*Set to A* Page 5 — Conn. 15. *Set to B* Page 5 — Conn. 14.

Burroughs Corporation
Detroit 32, Michigan

Offices in Principal Cities
In Canada: Burroughs Business Machines Ltd., Toronto, Ontario

LITHO IN U. S. A. 1-10-6 220-21012-D