[redacted] [redacted]

copy

Publication Number 420305449-001
April 1977

SC/MP
Simple Cost-effective
MicroProcessor

UTILITY PACKAGE (SUPAK)
USERS MANUAL

Do NOT PUNCH HOLES

© National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051

PREFACE

This publication provides operational information for using the SC/MP Utility Package (SUPAK) on an SC/MP or SC/MP-II Low Cost Development System (LCDS).

The material in this publication is subject to change without notice. Changes will be reported in COMPUTE, the Microprocessors Users Group newsletter.

Copies of this publication and other National Semiconductor publications may be obtained from the National Semiconductor distributor or sales office serving your locality.

Reference Publications:

- SC/MP Low Cost Development System (LCDS) Users Manual, Publication Number 4200105
- SC/MP Assembly Language Programming Manual, Order Number ISP-8S/994Y
- Memory Data Book
- SC/MP PROM Programming Users Manual, Publication Number 4200114

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 GENERAL INFORMATION

1.1 INTRODUCTION

The SC/MP Utility Package, SUPAK, is the program development firmware for the SC/MP Low Cost Development System (LCDS). SUPAK is resident in 4K of Read-Only Memory (ROM) and consists of the following programs:

- SC/MP Editor (SCEDT)
- SC/MP One-Pass Assembly (MINIAS)
- SC/MP PROM Tape Punch (PROM)

The assembler occupies the first 3072 words of ROM and the editor and tape punch programs occupy the remaining 1024 words. Figure 1 illustrates the operational sequence of the SC/MP Utility Package in conjunction with the DEBUG firmware. The DEBUG firmware is used also to transfer control to the programs comprising SUPAK. For details on the DEBUG firmware, see the SC/MP Low Cost Development System (LCDS) Users Manual.

SUPAK may be used to generate and edit paper tape source programs, assemble the source programs directly into memory, and punch ROM/PROM data on paper tape in the following formats:

- PN (Positive/Negative)
- BC (Binary Complement)

The ROM/PROM tapes, in either of the above formats, may be used to program MM5203 or MM5204 PROMs on either an IPC-16P PROM Programmer (IPC-16P/805) using SPPRO, or on an IMP-16P PROM Programmer (IMP-16P/805) using SPRSFT. The tapes also may be used to program n-by-8-bit PROMs on any PROM programmer that accepts paper tapes in the above specified formats.

PN (also called BPNF) formatted paper tapes may be submitted to National Semiconductor for mask programming of ROMs (see Memory Data Book for details).

1.2 SYSTEM CONFIGURATION

The operating environment for SUPAK requires the following hardware/firmware components:

- SC/MP or SC/MP-II Low Cost Development System (LCDS)
- Teletype with paper tape reader, paper tape punch, and reader relay
- SUPAK 4K ROM card
- RAM large enough to contain the user's program

1.3 CONVENTIONS

The following notation is used for examples:

- Nonunderlined upper-case characters indicate computer output. Example:

  EDITOR REV. A
- Underlined upper-case characters indicate user input, and CR indicates a carriage return. Example:

  +4, 6 CR
- Mixer upper/lower case characters indicates notes. Example:

  Turn on punch.

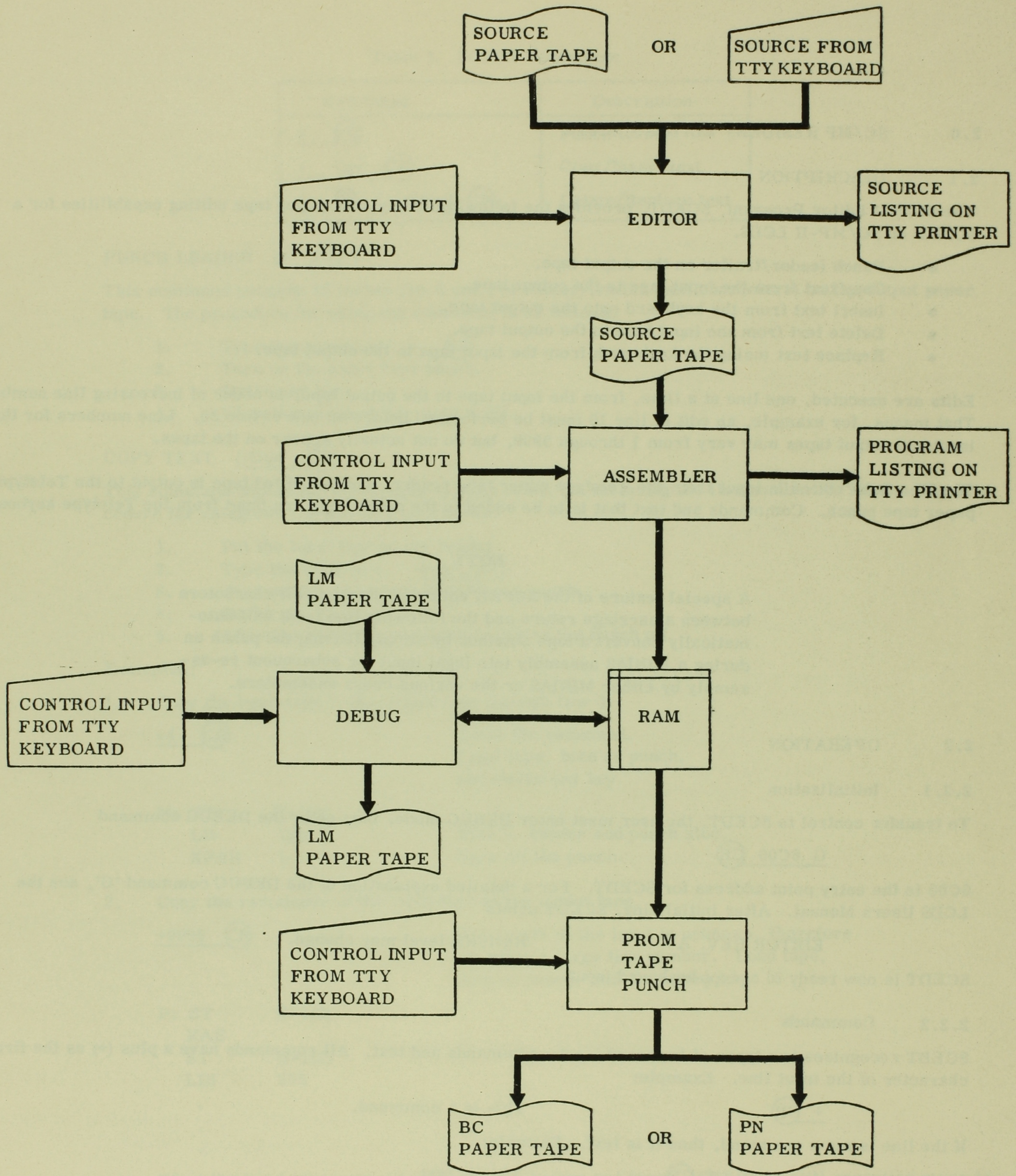

SOURCE PAPER TAPE
OR
SOURCE FROM TTY KEYBOARD
CONTROL INPUT FROM TTY KEYBOARD
EDITOR
SOURCE LISTING ON TTY PRINTER
SOURCE PAPER TAPE
CONTROL INPUT FROM TTY KEYBOARD
ASSEMBLER
PROGRAM LISTING ON TTY PRINTER
LM PAPER TAPE
CONTROL INPUT FROM TTY KEYBOARD
DEBUG
RAM
LM PAPER TAPE
CONTROL INPUT FROM TTY KEYBOARD
PROM TAPE PUNCH
BC PAPER TAPE
OR
PN PAPER TAPE

Figure 1. Operational Sequence of SUPAK

2.0 SC/MP RESIDENT EDITOR - SCEDT

2.1 DESCRIPTION

The SC/MP Editor Program, SCEDT, provides the following Teletype® paper tape editing capabilities for a SC/MP or SC/MP-II LCDS.

- Punch leader/trailer on the output tape.
- Copy text from the input tape to the output tape.
- Insert text from the keyboard onto the output tape.
- Delete text from thc input tape to the output tape.
- Replace text (using the keyboard) from the input tape to the output tape.

Edits are executed, one line at a time, from the input tape to the output tape, in order of increasing line numbers. That means, for example, an edit of line 10 must be performed before an edit of line 20. Line numbers for the input and output tapes may vary from 1 through 9999, but do not actually appear on the tapes.

The tape to be edited is input from the Teletype paper tape reader, and the edited tape is output to the Teletype paper tape punch. Commands and text that is to be added to the output tape are input from the Telctype keyboard.

> NOTE
>
> A special feature of the SUPAK editor (it discards all characters between a carriage return and the following line feed) will automatically convert a tape obtained by simply leaving the punch on during a MINIAS assembly into legal input for subsequent re-assembly by either MINIAS or the various cross assemblers.

2.2 OPERATION

2.2.1 Initialization

To transfer control to SCEDT, the user must enter DEBUG mode, then enter the DEBUG command

```
        G 6C00 CR
```

6C00 is the entry point address for SCEDT. For a detailed explanation of the DEBUG command 'G', see the LCDS Users Manual. After initializing, SCEDT prints

```
        EDITOR REV. A                  Revision level may change.
```

SCEDT is now ready to accept keyboard input.

2.2.2 Commands

SCEDT recognizes two types of keyboard input: commands and text. All commands have a plus (+) as the first character of the input line. Example:

```
        + CR                           This is a command.
```

If the line is not a command, then it is text. Example:

```
        A:    LDI  Ø CR                This is text.
```

The SCEDT commands are summarized in table 1 and described in detail in the following paragraphs.

® Registered trademark of the Teletype Corporation.

Table 1. SCEDT Commands

Command	Description
+ CR	Punch leader
+ line CR	Copy/Insert text
+ line 1, line 2 CR	Delete/Replace text

PUNCH LEADER (+ CR)

This command punches 12 inches (30.5 cm.) of leader/trailer (null characters) on the output paper tape. The procedure for using the command is as follows:

1. Type the command: + CR
2. Turn on the paper tape punch.
3. Strike any key.
4. When the punch stops, turn it off.

COPY TEXT (+line CR)

This command copies the input paper tape up to but not including the line numbered "line". The procedure for using this command is as follows:

1. Put the input tape in the reader.
2. Type the command: +line CR
3. Turn on the paper tape reader and punch.
4. Strike any key.
5. After the reader and punch stop, turn the punch off.

Examples:

1. Copy the input tape to the output tape through line 3.

```
+4 CR                           Enter the command.
                                Load tape, turn on punch,
                                and strike any key.

M:  ST      @-1(2)
    LD      1(3)                Line 3, reader and punch stop.
    XPAH    1                   Turn off the punch.
```

2. Copy the remainder of the input tape to the output tape.

```
+9999 CR                        The length of the input is unknown, therefore
                                we enter a large line number. Load tape,
                                turn on punch, and strike any key.

P:  ST      @-1(2)
    XAE
    ST      @-1(2)
    LDI     255
     .
     .
     .
```

When the input tape runs out, turn off the reader and punch. SCEDT is still expecting more input, therefore, the user must reinitialize the LCDS to continue. If the size of the input tape is known, the user should enter the last line number +1.

INSERT TEXT (+ line CR)

This command copies the input paper tape up to but not including the line numbered "line". The procedure for using this command is as follows:

1. Put the input tape in the reader.
2. Type the command: + line CR
3. Turn on the paper tape reader and punch.
4. Strike any key.
5. After the reader and punch stop, input text from keyboard.
6. After completing text insertion, turn off the punch.

Example:

```
+4 CR                                   Enter command.  Load tape, turn on punch, and
                                        strike any key.

M:  ST      @-1(2)
    LD      1(3)
    XPAH    1                           Line 3, reader and punch stop.
```

Begin inserting text.

```
    ST      @-1(2)  CR
    LD      2 (3)   CR
     .
     .
     .
```

After the end of the text, turn off the punch.

DELETE TEXT (+ line 1, line 2 CR)

This command copies up to but not including the line numbered "line 1" and then deletes the range of lines from "line 1" to "line 2". The procedure for using this command is as follows:

1. Put the input tape in the reader.
2. Type the command: +line 1, line 2 CR
3. Turn on the reader and punch.
4. Strike any key.
5. When the reader and punch stop, turn off the punch.

Example:

```
+10, 12 CR                              Delete lines 10 through 12.  Load tape, turn on
                                        punch, and strike any key.
XPAH    3                               This is line 7.
ST      @-1(2)
XPAL    3                               Line 9, the punch stops.
```

The reader stops after line 12. Turn off the punch.

REPLACE TEXT (+line 1, line 2 CR)

This command copies up to but not including the line numbered "line 1" and then deletes the range of lines from "line 1" to "line 2". The procedure for using this command is as follows:

1. Put the input tape in the reader.
2. Type the command: +line 1, line 2 CR
3. Turn on the reader and punch.
4. Strike any key.
5. When the reader and punch stop, input text from the keyboard.
6. After completing text, turn off the punch.

Example:

```
+10, 12  CR                                 Delete lines 10 through 12.
                                            Load tape, turn on punch, and strike
                                            any key.
XPAH     3                                  This is line 7.
ST       @-1(2)
XPAL     3                                  Line 9, the punch stops.
```

The reader stops after line 12. Input the replacement text:

```
XPAH    2  CR
ST      @-1(3)   CR
XPAL    2  CR
```

At the end of the text, turn off the punch.

2.3 SOURCE INPUT

Input to SCEDT is a paper tape containing variable length ASCII source statements. Each statement is terminated with a carriage return/line feed.

2.4 SOURCE OUTPUT

Output from SCEDT is a corrected paper tape, identical in format to the input tape except for the deletion of any characters that may have occurred between a carriage return and the succeeding line feed. The paper tape is accepted by the SC/MP-resident assembler as source input.

3.0 SC/MP RESIDENT ASSEMBLER — MINIAS

3.1 DESCRIPTION

The SC/MP-resident, one-pass assembler, MINIAS, allows LCDS users to assemble programs in a symboli language, that is a subset of the standard SC/MP assembly language. MINIAS accepts source input, one stat ment at a time, from the Teletype keyboard or paper tape reader and assembles the statement directly into memory.

MINIAS executes on any standard SC/MP LCDS, resides in 3K or ROM beginning at X'6000, and uses the mo board RAM for working storage.

3.2 LIMITATIONS

MINIAS recognizes all SC/MP opcode mnemonics and some of the directives. It allows simple expressions, recognizes the symbol '.' as the current value of the location counter. With one exception, MINIAS is upwar compatible with the other NSC assemblers for SC/MP; thus, in so far as MINIAS checks syntax, nothing is le that would not be legal in the other SC/MP assemblers. (The one exception is the multiple definition of symb see 3.2.3 for details.) The reverse is not true. There are legal constructs for other SC/MP assemblers th are not handled by MINIAS.

3.2.1 Directives

MINIAS recognizes only the following directives:

.BYTE
.DBYTE
.ADDR
.END

MINIAS allows only one operand expression in the .BYTE, .DBYTE, and .ADDR directives. Upon recognitic of a .END directive, MINIAS returns control to the LCDS monitor. MINIAS does not recognize a starting address specified by the .END directive.

All other directives are ignored.

3.2.2 Expressions

MINIAS handles expressions that consist of decimal or hexadecimal constants or symbols combined with the following operators:

Operator	Description
+	Addition
-	Subtraction

Evaluation of expressions is simple left-to right with no precedence implied or parenthetical grouping provide

3.2.3 Symbols

MINIAS handles the 26 single-character identifiers A through Z and the symbol '.', which represents the curr value of the location counter. At the beginning of the assembly, all symbols have the value of zero. Identifie of more than one character may be used, but only their first character will be significant. For example: APPLE = A.

There are two methods of assigning a value to a symbol. The alphabetic identifiers may appear in the progra as labels or any symbol may be assigned a value in an assignment statement (. = expression or identifier = e pression). The value of a symbol may be altered, that is, multiple definitions are allowed; the most recent h ing precedence. This is the only instance in which MINIAS allows an operation that is not allowed in the other SC/MP assemblers.

The statement " . = expression" may be used to specify where in memory any particular section of data or co is to be assembled. There is no protection against overlaying previously assembled code.

3.2.4 Addressing

Jumps or memory references from the end of a 4K memory page to the beginning or vice-versa may not be explicitly specified. However, if the location counter is X'FFD, a jump specified as

```
        JMP     4(0)
```

will cause the next instruction to be selected from location X'0002.

3.3 PROGRAMMING TECHNIQUES

A line-by-line assembler is, by its nature, a one-pass assembler and hence does not handle forward references. More precisely; a symbol must be defined before it is used if the instructions involving it are to be assembled with the correct value.

For this reason, all references that are known in advance (that is constants, RAM storage addresses or offsets) should be defined first, at the beginning of the code in which they are used. This is good programming practice under any circumstances.

Most unavoidable forward references occur in the control portion of the program, e.g., as addresses of jumps which cannot be handled safely with expressions such as ".+2" or "X+1". There is a technique for handling forward references in MINIAS which covers this situation. Suppose a forward jump of unknown length from location F to location Q is desired. The code may be written as follows:

```
F:      JZ      .                       ;Q WILL BE PUT HERE. "." IS ALWAYS LEGAL

        ...

Q:                                      ;BLANK LINE
        .=F                             ;GO BACK TO F
        JZ      Q                       ;FIX UP THE INSTRUCTION
        .=Q                             ;GO BACK TO ASSEMBLING AS Q
```

Note that the symbol F is now available for use again in a similar context; however, this would make the code non-upward compatible. Of course, it is not necessary to use a symbol for this purpose. One can instead use the hexadecimal address assembled next to the forward reference explicitly.

If it is necessary to re-use identifiers in the course of an assembly, this can be done in a way that will not conflict with assembly later on one of the cross assemblers. The method is to use two character identifiers, such as 'A1' the first time 'A' is used and 'A2' the second time 'A' is used. The identifiers 'A1' and 'A2' are the same to MINIAS. It is perhaps best to use regular multiple character identifiers taking care that no two identifiers starting with the same letter are used at the same time.

NOTE

> A special feature of the SUPAK editor (it discards all characters between a carriage return and the following line feed) will automatically convert a tape obtained by simply leaving the punch on during a MINIAS assembly into legal input for subsequent re-assembly by either MINIAS or the various cross assemblers.

3.4 OPERATION

To use MINIAS, the user must have the Teletype connected to the LCDS and ready, and he must have the following cards inserted into the LCDS:

- SUPAK 4K ROM card
- RAM card for the address where the users program will reside

To transfer control to MINIAS the user must enter DEBUG Mode and then type the DEBUG command

```
        G 6000 CR
```

6000 is the entry point address for MINIAS. For a detailed explanation of DEBUG Mode see the SC/MP LCDS Manual. Upon entry, MINIAS resets the value of all symbols to zero, prints

```
        MINIAS
```

and spaces over to column 19 to await input of the first source statement.

If necessary, the user may restart the assembler and perserve the current values of his symbols by entering command

```
          G 6666  CR
```

If a paper tape is to be used for input, it should be placed in the paper tape reader and the reader turned on.

3.5 SOURCE INPUTS

Source statements are input to the assembler one at a time and are assembled immediately. The locations in which they are assembled may be controlled by use of assignment statements (see 3.2). The character set us is ASCII, as described on the back page on the SC/MP instruction guide (4200110), and the statement terminat is a carriage return (X'0D).

Two editing characters are recognized by MINIAS. If a shift-O (X'5F) is input, MINIAS will delete the previou input character (on the current line only) and echo a back arrow (<—). MINIAS also recognizes a CTRL-Q (X as a "restart line" character. The assembler will discard all characters input since the last previous carriag return. It will also echo a carriage return and a line feed (X'0A) then space to column 19 to await the new inp line. The line number is not advanced in this instance.

MINIAS ignores all other ASCII control characters (X'00-X'1F) and the rubout character (X'7F).

3.6 PROGRAM LISTING OUTPUT

Beginning in column 19, MINIAS echos each character of the source statement as it is input. Upon recognizin the carriage return; MINIAS echos a carriage return; spaces over two characters; prints a four character line number, the four-character location counter (hexadecimal), and either one or two bytes of generated code (he decimal). Following this, it outputs a line feed and spaces to column 19 to await input of the next source statement.

If an error is discovered in a statement, the first two columns preceeding the line number will contain a (?) a a back arrow (<—) rather than blanks. The statement number will be updated to provide for possible later pro cessing by the SUPAK editor, but the location counter will not be altered.

3.7 ASSEMBLY EXAMPLE

In the following example, a small program is assembled into memory starting at location X'100.

```
CL 0001                                          Initialize.
-G 6000 CR                                       Jump to MINIAS.
MINIAS
  0001          0100   .=0100 CR
  0002          0000   MS=0 CR
  0003          0001   LS=1 CR
  0004   0100   02     CCL CR
  0005   0101   C400   LDI 0 CR
  0006   0103   FA01   CAD LS(2) CR
  0007   0105   CA01   ST LS(2) CR
  0008   0107   C400   LDI 0 CR
  0009   0109   FA00   CAD MS(2) CR
  0010   010B   CA00   ST MS(2) CR
  0011   010D   00     HALT CR
  0012                 .END              ;TRANSFER CONTROL TO MONITOR CR
CL67F5
```

NOTE

In the above program, the source statements (the portion of the line that is underlined) may be input from the Teletype paper tape reader or from the Teletype keyboard.

4.0 SC/MP RESIDENT PROM TAPE PUNCH — PROM

4.1 DESCRIPTION

The SC/MP-Resident PROM Tape Punch Program, PROM, allows a user to punch PN (Positive/Negative) or BC (Binary Complement) formatted paper tapes for any address range in SC/MP memory. For a description of the paper tape formats, see the SC/MP PROM Programming Users Manual.

A 16-bit checksum is also punched in the trailer section of each tape. The checksum is separated from the data section by 10 null characters following the last rubout character.

4.2 OPERATION

To transfer control to PROM the LCDS must be in DEBUG Mode and the user must enter the DEBUG command

G 6EA0 CR

6EA0 is the entry point address for PROM. Upon entry, PROM prompts with the messages

SC/MP PROM TAPE
PROM SIZE:

The user should enter the PROM size in hexadecimal. Next PROM prints

START ADDR:

The user should respond with the starting address of the memory range to be punched. Addresses are left filled with zeros. If more than four hexadecimal digits are entered, only the last four are saved. Finally PROM prints

FORMAT:

The user should respond with a 'P' or a 'B' for PN- or BC-formatted paper tape respectively. When the response is entered, PROM begins punching the paper tape. The user should immediately turn on the punch as the leader is being punched.

NOTE

PROM uses RAM locations X'7700 to X'771F on the CPU card. The memory range punched should not cross a 4K page boundary.

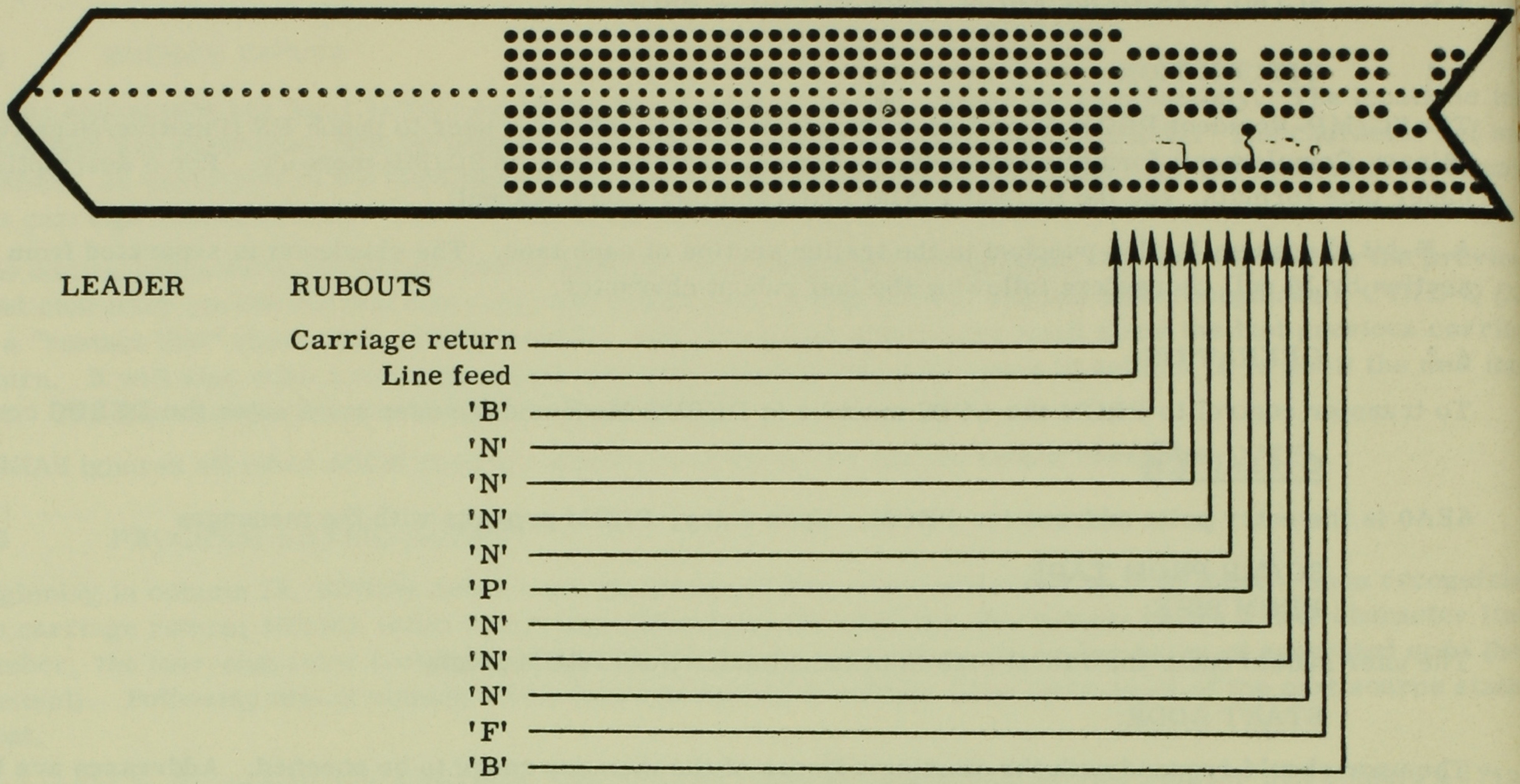
4.3 DATA OUTPUTS

PROM outputs a paper tape in PN (Positive/Negative) or BC (Binary Complement) format.

4.3.1 PN Format

In PN format, there are ten consecutive characters punched on the paper tape for each 8-bit byte output. The first character in the string is a 'B' for begin, the next eight characters are either 'P' for positive (logic '1') or an 'N' for negative (logic '0'), and the last character is an 'F' for finished. The most significant bit of the data byte corresponds to the first 'P' or 'N' character, and the least significant bit of the data byte corresponds to the last 'P' or 'N' character. A carriage return and line feed are punched on the paper tape after every fourth 10-character string so that the tape may be listed conveniently on a Teletype.

Each paper tape contains a leader of 120 null characters followed by 32 rubout characters. The trailer consists of 32 rubout characters followed by 10 null characters and a 16-bit binary checksum and then 120 null characters. The following is an example of the leader section and first data word of a PN formatted paper tape.

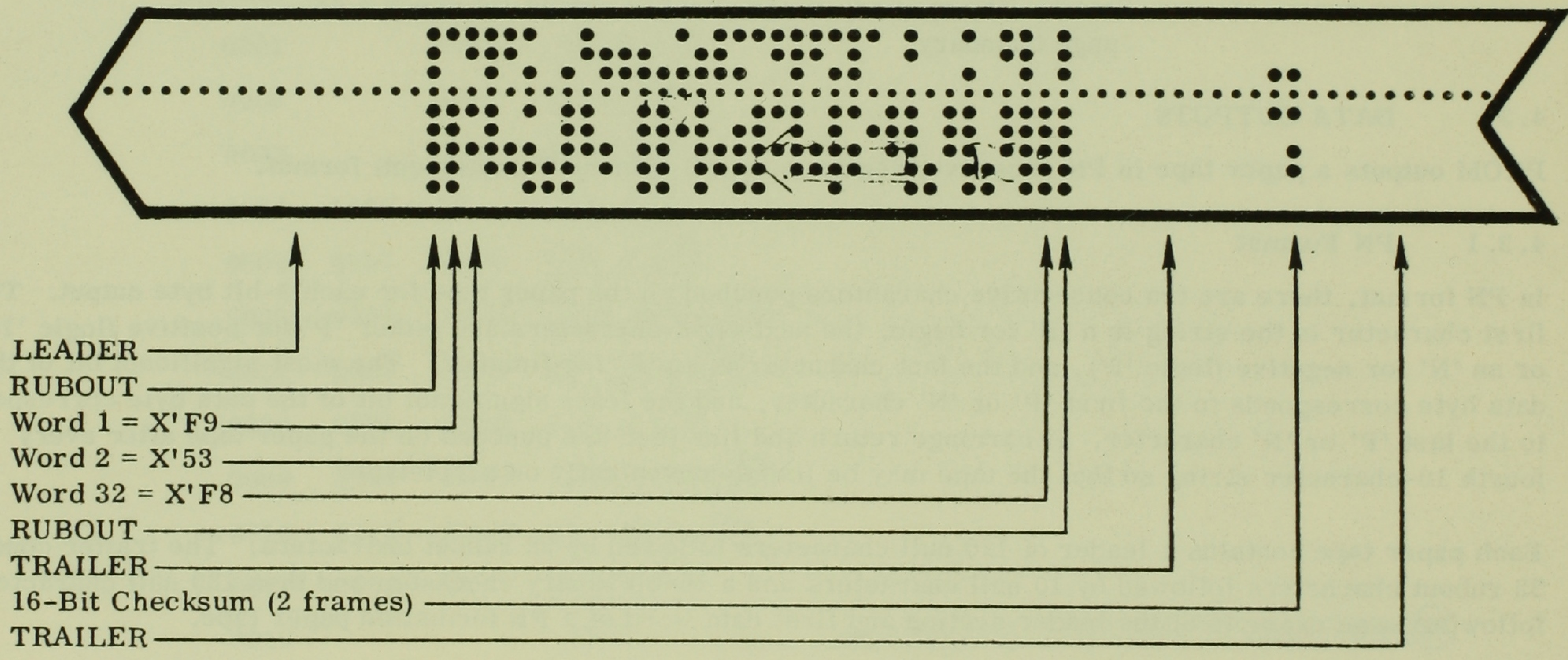


NOTE

There is a carriage return and line feed after every fourth word.

4.3.2 BC Format

In BC format there is one character punched on the paper tape for each 8-bit byte output. The character punche is the ones complement of the actual data in memory. The following is an example of a BC formatted paper tap



4.4 EXAMPLE

The following example shows a 32-by-8-bit PROM tape being punched in PN format. The PROM size is X'20, and the starting address is X'100.

```
                                                  Push INIT switch.
CL  0001
-G  6EA0  CR                                      Jump to PROM.
SC/MP PROM TAPE
PROM SIZE:  020 CR
START ADDR:  0100  CR
FORMAT:  P                                        Turn on punch.
BNNNPNNPNFBNPPPNNNNFBNPNNNNPPFBNNPNNNPNF
BNNNNPNNNFBPPNNPNNNFBNNNPPNNNFBNNNPNNNPF
BNNNNNNNNFBNNNNNNNPFBNNNNNNPNFBPNNNNPNNF
BNNPNNPNNFBPNPPPNNNFBPPNNPNPNFBNNNNNNPNF
BNNNNPNNNFBPPNNPNNNFBNNNPPNNNFBNNNPNNNPF
BNPPPNPNPFBNPNPNPNPFBPPNPNPNNFBNNPNPNPNF
BNNNNNNNNFBNNNNNNNPFBNNNNNNPNFBPNNNNPNNF
BNNNPNNPNFBNPPPNNNNFBNPNNNNPPFBNNPNNNPNF  Turn off punch.
```