

MOBILE

A MOBIDIC COBOL COMPILER

SECOND QUARTERLY PROGRESS REPORT

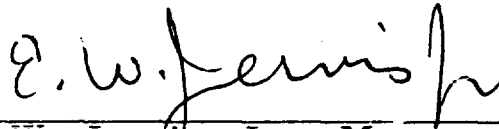
1 May 1962 to 1 October 1962

AD-406 874

Signal Corps  
Technical Requirements  
SCL-2101N

Contract No. DA-36-039-sc-89231

Submitted by:

  
E. W. Jervis, Jr., Manager  
MOBIDIC Projects

**Best Available Copy**

SYLVANIA ELECTRONIC SYSTEMS - EAST

SYLVANIA ELECTRONIC SYSTEMS  
A Division of Sylvania Electric Products Inc.  
189 B Street - Needham Heights 94, Massachusetts

QUALIFIED REQUESTORS MAY OBTAIN COPIES OF THIS REPORT  
FROM ASTIA. ~~ASTIA RELEASE TO OTS NOT AUTHORIZED.~~ *Per ltr 4 Mar 63*

## TABLE OF CONTENTS

| <u>Section</u>                                      | <u>Page</u> |
|-----------------------------------------------------|-------------|
| LIST OF ILLUSTRATIONS                               | vii         |
| LIST OF TABLES                                      | vii         |
| I PURPOSE                                           | 1-1         |
| II ABSTRACT                                         | 2-1         |
| III PUBLICATIONS, LECTURES, REPORTS AND CONFERENCES | 3-1         |
| IV FACTUAL DATA                                     | 4-1         |
| 4.1 General Description                             | 4-1         |
| 4.2 System Flowcharts and Table Description         | 4-1         |
| 4.2.1 Assembly Programs and Languages               | 4-1         |
| 4.2.1.1 DUST                                        | 4-1         |
| 4.2.1.2 94AP                                        | 4-2         |
| 4.2.1.3 MAP                                         | 4-2         |
| 4.2.1.4 MODAL                                       | 4-2         |
| 4.2.1.5 CAP(S)                                      | 4-2         |
| 4.2.1.6 MUST                                        | 4-3         |
| 4.2.2 Relationship of Languages and Assemblers      | 4-3         |
| 4.3 Run Modification Descriptions                   | 4-4         |
| 4.3.1 Run 1.0 Modifications                         | 4-4         |
| 4.3.2 Run 1.1 Modifications                         | 4-5         |
| 4.3.3 Run 1.2 Modifications                         | 4-6         |
| 4.3.4 Run 1.3 Modifications                         | 4-7         |
| 4.3.5 Own Code General Discussion                   | 4-10        |
| 4.3.5.1 ENTER Declarative                           | 4-10        |
| 4.3.5.2 Own Code Formats                            | 4-10        |
| 4.3.5.3 Own Code Implementation                     | 4-11        |
| 4.3.5.3.1 Pass I                                    | 4-11        |
| 4.3.5.3.2 Pass II                                   | 4-12        |
| 4.3.5.3.3 Pass III                                  | 4-12        |

## TABLE OF CONTENTS (Cont.)

| <u>Section</u>                                              | <u>Page</u> |
|-------------------------------------------------------------|-------------|
| 4.3.5.4 Own Code Restrictions                               | 4-13        |
| 4.3.5.5 Own Code Tables                                     | 4-14        |
| 4.3.5.5.1 OW1N                                              | 4-14        |
| 4.3.5.5.2 NONE                                              | 4-15        |
| 4.3.5.5.3 PS1N                                              | 4-15        |
| 4.3.5.5.4 OC1N                                              | 4-16        |
| 4.3.5.6 Instruction Field Interpretation                    | 4-17        |
| 4.3.5.6.1 Group 1—Internal Instructions                     | 4-18        |
| 4.3.5.6.2 Group 2—Pseudo Instructions                       | 4-19        |
| 4.3.6 Run 2 Modifications                                   | 4-25        |
| 4.3.6.1 Former Function                                     | 4-25        |
| 4.3.6.2 Modified Function and Internal Sort Description     | 4-25        |
| 4.3.7 Fun 4 Modifications                                   | 4-32        |
| 4.3.8 Run 3 Modifications                                   | 4-33        |
| 4.3.8.1 Run 3 Service Routines                              | 4-33        |
| 4.3.8.1.1 Purpose                                           | 4-33        |
| 4.3.8.1.2 Input                                             | 4-33        |
| 4.3.8.1.3 Method                                            | 4-33        |
| 4.3.8.1.4 Output                                            | 4-35        |
| 4.3.8.2 Input-Output Generator                              | 4-36        |
| 4.3.8.2.1 Purpose                                           | 4-36        |
| 4.3.8.2.2 Input                                             | 4-37        |
| 4.3.8.2.3 Output                                            | 4-37        |
| 4.3.8.2.4 Input-Output Generator Segments                   | 4-38        |
| 4.3.8.2.5 Description of the Five Segments of I/O Generator | 4-39        |
| 4.3.8.3 Modification of the Non I/O Generators              | 4-41        |
| 4.3.8.3.1 The ACCEPT Generator                              | 4-41        |

## TABLE OF CONTENTS (Cont.)

| <u>Section</u>                                          | <u>Page</u> |
|---------------------------------------------------------|-------------|
| 4.3.8.3.1.1 Purpose                                     | 4-41        |
| 4.3.8.3.1.2 Input                                       | 4-41        |
| 4.3.8.3.1.3 Method                                      | 4-41        |
| 4.3.8.3.1.4 Generator Processing                        | 4-42        |
| 4.3.8.3.2 The DISPLAY Generator                         | 4-43        |
| 4.3.8.3.2.1 Purpose                                     | 4-43        |
| 4.3.8.3.2.2 Method                                      | 4-44        |
| 4.3.9 Run 5 Modifications                               | 4-46        |
| 4.3.9.1 Former Function                                 | 4-46        |
| 4.3.9.2 Modified Function                               | 4-46        |
| 4.4 Tape Allocation and Usage                           | 4-47        |
| 4.4.1 Current Tape Assignments and Usage at Compilation | 4-48        |
| 4.5 Format and Usage of the Compiler Systems Tape       | 4-60        |
| 4.5.1 Contents of the Systems Tape                      | 4-60        |
| 4.5.1.1 Programs on the Systems Tape                    | 4-60        |
| 4.5.1.2 Expansion                                       | 4-61        |
| 4.5.2 Header and Trailer Blocks                         | 4-61        |
| 4.5.2.1 Format of the Header and Trailer Blocks         | 4-61        |
| 4.5.3 Putting Programs On the Systems Tape              | 4-61        |
| 4.5.3.1 Program Starting Location                       | 4-62        |
| V CONCLUSIONS                                           | 5-1         |
| VI PROGRAM FOR THE NEXT PERIOD                          | 6-1         |
| VII IDENTIFICATION OF KEY PERSONNEL                     | 7-1         |
| 7.1 Key Technical Personnel                             | 7-1         |
| 7.2 Approximate Man-Hours Expended                      | 7-1         |
| APPENDIX A - RUN 1.0 PROGRAM LISTING                    |             |
| APPENDIX B - DISTRIBUTION LIST                          |             |

## LIST OF ILLUSTRATIONS

| <u>Figure</u>                                         | <u>Page</u> |
|-------------------------------------------------------|-------------|
| 4-1 Run 1.3 - Procedure Division Analysis             | 4-8         |
| 4-2 Run 1.3 - Segmentation In MOBILE                  | 4-9         |
| 4-3 Own Code (Pass I)                                 | 4-20        |
| 4-4 Own Code (Pass II)                                | 4-21        |
| 4-5 Own Code (Pass III)                               | 4-23        |
| 4-6 Internal Sort Flowchart                           | 4-27        |
| 4-7 Input-Output Tape Assignment At Compilation       | 4-50        |
| 4-8 Usage of Tape 4, 5 and 6 for Each Compilation Run | 4-51        |
| 4-9 Tables on Input-Output Tapes at Compilation Runs  | 4-52        |

## LIST OF TABLES

| <u>Table</u>                                         | <u>Page</u> |
|------------------------------------------------------|-------------|
| 4-1 Information Required in Communication with Run 3 | 4-34        |
| 4-2 Detailed Use of Tape at Compilation              | 4-54        |

## SECTION I

### PURPOSE

#### 1.1 MOBILE, A COBOL COMPILER FOR THE DATA PROCESSING NEEDS OF THE ARMY SIGNAL CORPS

The objective of this procurement is to produce, for MOBIDIC computers (C, D, 7A) a COBOL compiler capable of accepting a Source Program written in COmmon Business Oriented Language, and compiling an Object Program capable of being operated on the above computers.

This procurement will result in delivery to the U. S. Army Signal Corps of a COBOL Compiler (MOBILE I) as specifically defined in Required-COBOL 1961 and with certain features as specified in Elective-COBOL 1961. The MOBILE I Task is being implemented by the Applied Programming Department of the Programming and Analysis Laboratory.

## **SECTION II**

### **ABSTRACT**

The program during this reporting period included unit testing and the beginning of package testing for the MOBIDIC MOBILE Runs. Runs, 1.0, 1.1, 2, 4, 5 and the Non I/O Generators are ready to begin or have begun package testing. The segmentation of Run 1.3 and the I/O Generator is complete. New coding is complete for the Sort Merge Routines. The following Areas are Covered in this report:

- MODIFICATION OF RUNS
- MOBILE I SYSTEMS TAPE
- TAPE ALLOCATION AND USAGE

SECTION III  
PUBLICATIONS, LECTURES, REPORTS AND CONFERENCES

3.1 PUBLICATIONS

None

3.2 LECTURES

1. Title: Introduction to COBOL  
Lectures: Basic COBOL,  
Practice problems and solutions  
Location: Fort Monmouth, New Jersey  
Date: 11 June 1962 - 13 June 1962
2. Title: COBOL Training Course  
Lectures: Detailed discussion of four main divisions  
of COBOL Source Program and statement  
of practice problem.  
Location: Fort Monmouth, New Jersey  
Date: 6 August 1962 - 10 August 1962
3. Title: MOBIDIC MOBILE I, a COBOL compiler  
Lectures: A discussion of the phases and functions of  
the MOBIDIC MOBILE compiler  
Location: Fort Monmouth, New Jersey  
Date: 16 August 1962

3.3 REPORTS

1. Monthly Letter Report 1 June 1962
2. Monthly Letter Report 28 June 1962
3. Monthly Letter Report 30 July 1962
4. Monthly Letter Report 7 September 1962

### 3.4 CONFERENCES

1. Date: 28 May 1962  
Location: Fort Monmouth, New Jersey  
Participants: Signal Corps, Sylvania  
Subject: Overall Project Status
2. Date: 21 June 1962  
Location: Sylvania, Needham  
Participants: Signal Corps, Sylvania  
Subject: Acceptance Testing
3. Date: 13 August 1962 – 15 August 1962  
17 August 1962  
Location: Fort Monmouth, New Jersey  
Participants: Signal Corps, Sylvania  
Subject: Possible Solutions of Acceptance Test Problem
4. Date: 23 August 1962  
Location: Sylvania, Needham  
Participants: Signal Corps, Sylvania  
Subject: Overall Project Status
5. Date: 28 August 1962  
Location: Sylvania, Needham  
Participants: Signal Corps, Sylvania  
Subject: Overall Project Status

## SECTION IV

### FACTUAL DATA

#### 4.1 GENERAL DESCRIPTION

Further development of the MOBIDIC MOBILE I Compiler and its current status are described in detail in this section. The task breakdown is as follows:

- Description of Run Modifications
- Auxiliary program descriptions

#### 4.2 SYSTEM FLOWCHARTS AND TABLE DESCRIPTION

The reader is advised to read the System Flowchart and Table Descriptions of the 1st Quarterly Progress Report in conjunction with the Run Modifications described in Section 4.3.

##### 4.2.1 Assembly Programs and Languages

Since mnemonic names and symbols are inherent in programming, it is appropriate at this time to describe briefly some of the Assembly Program and Languages used during this contract period. The first three Assembly Programs (DUST, 94AP, and MAP) are not contained in the COBOL Compiler whereas the second set of three Assembly Programs and Programming Language (MODAL, CAP, and MUST) is part and parcel of the COBOL Compiler.

##### 4.2.1.1 DUST

DUST is the mnemonic name for the MOBIDIC D Unlimited Symbol Table Assembly Program. This assembly program was adapted from the 9400 Assembly Program for use on the MOBIDIC D computer. It is a more powerful and versatile assembler than MAP. A complete description of the features of DUST can be found in Sylvania Memo MC-TP-233.

#### 4.2.1.2 94AP

94AP is the mnemonic name for the Sylvania 9400 Assembly Program. This assembly program was developed by Sylvania for use on Sylvania's 9400 computer. The basic function of any assembly program is to convert symbolic coding to the machine language and essentially, this is a one-to-one conversion as opposed to a many-to-one conversion inherent in compilers.

#### 4.2.1.3 MAP

MAP is the mnemonic name for the MOBIDIC Assembly Program designed and developed primarily for the MOBIDIC A computer by Sylvania under Signal Corps contract. MAP can be used on the other MOBIDIC Computers, (e.g., C& 7A). This assembly program is not as powerful as 94AP or the DUST Assembler due to the limited hardware configuration of the specified computer, (e.g., 8K core memory and two magnetic tape units).

#### 4.2.1.4 MODAL

MODAL is the mnemonic name for the MOBIDIC Assembly Language. This programming language makes possible the insertion of symbolic coding into COBOL statements in the Procedure Division through the ENTER option. MODAL is a subset of the MOBIDIC instruction set. The difference is that in MODAL no I/O instructions can be used and certain pseudo instructions are not allowed. The final report will contain a complete listing of allowable instructions. It should be noted that the format for ENTER generator is:

ENTER MODAL.

This ties together the language and its use in COBOL.

#### 4.2.1.5 CAP(S)

CAP is the mnemonic name for the Compiler Assembly Program which constitutes runs 7 and 8 of MOBILE. The assembly phase is part of any compilation process which also includes card punching and listings. It can be said that CAP is a much smaller version of DUST and performs the same basic functions.

The user is not to confuse CAP with CAPS, the latter being the definition for the major input in the Assembly Phase (CAP) of the Compiler. Each CAPS consists of two data words which are processed into one word in machine code.

#### 4.2.1.6 MUST

MUST is the mnemonic name for the MOBILE Utility System Translator program. The function of this program is to assemble in a special manner the various subroutines that are used as prepackaged object code for certain COBOL statements. When it was realized that these prepackaged subroutines were all relativized to the first location and that many addresses must be filled in at object time, it was clear that a special purpose assembly program was required, and MUST was developed. The third quarterly will contain a complete section describing MUST and its operation.

#### 4.2.2 Relationship of Languages and Assemblers

| <u>SOURCE LANGUAGE</u>              | <u>ASSEMBLER</u>   | <u>OBJECT LANGUAGE</u>       |
|-------------------------------------|--------------------|------------------------------|
| 9400<br>Symbolic<br>Instructions    | 94AP               | Machine Code                 |
| MOBIDIC<br>Symbolic<br>Instructions | DUST<br>or<br>MAP* | Machine Code<br>Machine Code |
| MOBIDIC<br>Symbolic<br>Instructions | MUST               | CAPS                         |
| MODAL                               | Runs 6, 7, 8       | Machine Code                 |
| CAPS                                | Runs 7, 8          | Machine Code                 |

---

\* Not all symbolic instructions are acceptable

### 4.3 RUN MODIFICATION DESCRIPTIONS

Run 1.0 is complete and its program listing is available in Appendix A. Modification on Runs 1.3, 2.4, 3 and 5 is described in detail in this section. Below is a general outline of the Run modifications:

1. Run 1.3 has been segmented into two sections, 1.3A and 1.3B.
2. Run 2 has been modified to replace the binary sort with an internal sort.
3. Run 4 has been modified to provide compatibility with the core memory capacity for the MOBIDIC Computer.
4. Run 3 modifications involves the Service Routines, segmentation of the Input/Output Generator, and the Non-Input/Output Generators.
5. Run 5 modification involves the use of the internal sort, a reduction in the size of the input buffer, and the addition of a merge phase.

#### 4.3.1 Run 1.0 Modifications

See Run 1.0 Listing in Appendix A.

#### 4.3.2 Run 1.1 Modifications

Modifications have been completed for Run 1.1. Modification was necessary due to the existence of several LXS instructions in the coding. All LXS instructions were changed to two LOD instructions. Any relative addressing using greater than 12 bits were replaced by a CLA and RPA sequence. The main modification was done in changing indexing instructions to make certain that they would execute properly with the MOBIDIC 12-bit index registers.

The Hollerith to FIELDATA Routine (HFC3) was added to Run 1.1. Previously it had been left in core memory by the Control Program. However, by reading in HFC3 as part of the individual runs, it was found that the available core memory space could be optimized.

#### 4.3.3 Run 1.2 Modifications

During this reporting time period, Run 1.2 was completely debugged (with simple addressing information). Following the completion of unit testing, Run 1.2 was completely reassembled. Time then was spent debugging the new assembly and preparing it for entrance into the compiler system for the first time. At this point, there were many errors due to incompatibility between Run 1.2 and the Control Program and other compiler runs. There were certain types of input errors which were detected, but not corrected properly by Run 1.2.

Once the incompatibilities were overcome and the system was running smoothly, the remainder of the reporting period was spent in assembling and debugging complex address functions.

#### 4.3.4 Run 1.3 Modifications

Because core memory is not large enough to contain all the environmental data, Run 1.3 has been segmented into 2 sections, 1.3A and 1.3B. Now, the Data Name List is not in core during the operation of 1.3A, but is available for 1.3B. The size of the running code has caused no trouble to date.

##### 4.3.4.1 Run 1.3A

The main function of 1.3A is to scan the Source Program and produce a new kind of temporary compressed generator call for use as input to 1.3B. This generator call contains the data-name, its character count, and its type, i.e., base name, subscript, or qualifier.

##### 4.3.4.2 Run 1.3B

This segment of Run 1.3 uses the now available Data Name List and the temporary Compressed Generator Calls produced in 1.3A to make up the normal Compressed Generator Calls. Run 1.3B scans the Data Name List and obtains the location word for the data name entry in the temporary Generator Call. The location word is then placed in the final Generator Call.

##### 4.3.4.3 Effects of the Segmentation

Subroutines PT160-168 of Run 1.3A which analyze the input have been changed. Subroutines PT164-PT166 are in Run 1.3B. New coding has been added to both segments to carry out the function of making up the new type generator call as well as to effect a smooth transition from 1.3A to 1.3B. A completely new PT166 has been written for 1.3B that now includes PT167. PT167 as such, no longer exists. A new PT164 has been written for 1.3A and PT164 has been modified for 1.3B.

Additional core space was made available when all non-Procedure Division words were removed from the Reserved Word List. Also, changes in the error routines contributed extra space. Subroutines PT150 and PT154 have been re-named PT180 and PT184, respectively.

Testing has begun on SUBSCRIPTed names and DECLARATIVES and analysis of some of the PERFORM options has started.

•



100

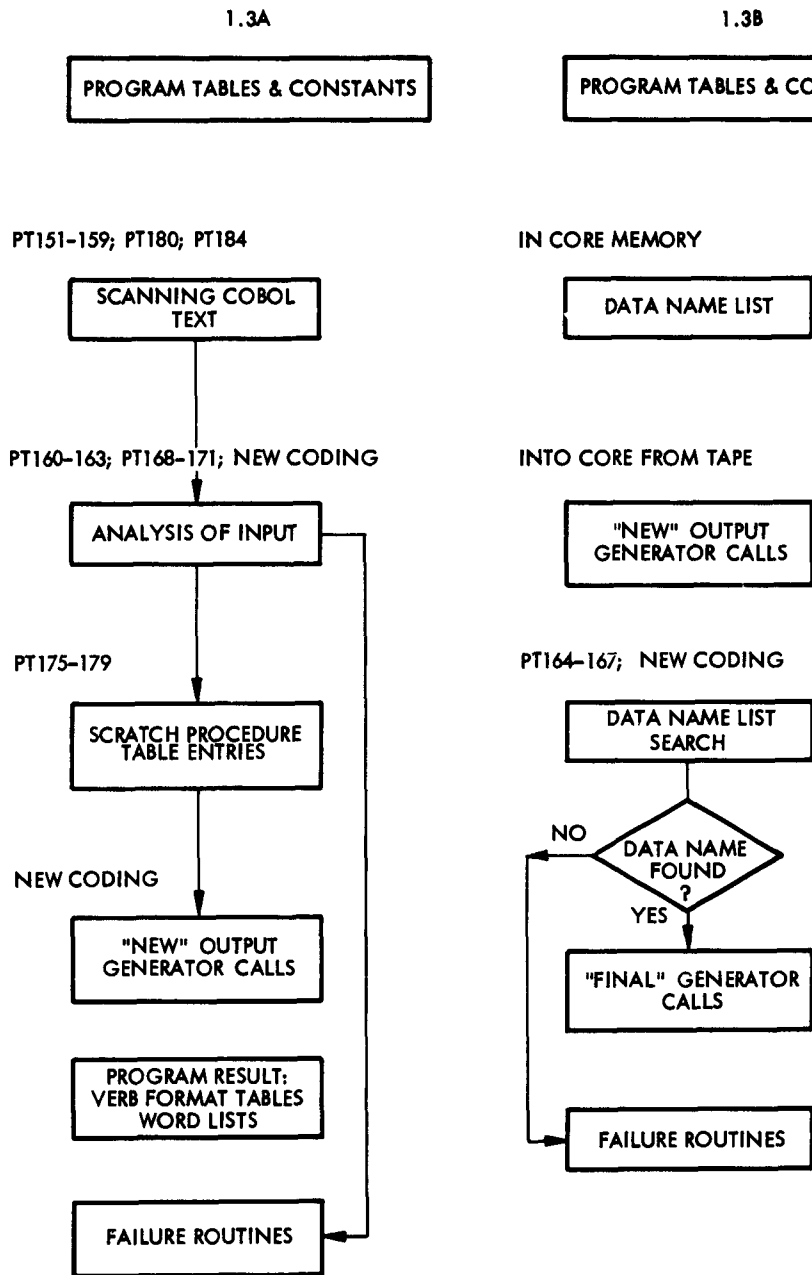


Figure 4-2. Run 1-3—Segmentation In MOBILE

#### 4.3.5 Own Code General Discussion

The COBOL user may at times have difficulty in expressing in COBOL those debugged assembly language subroutines which he desires to use. However, the MOBILE system allows the user to include those assembly instructions called "Own Code" in his COBOL Procedure Division to represent those subroutines. This is done by means of the ENTER verb. The user must adhere to the own code restrictions defined in 4.3.5.4.

##### 4.3.5.1 ENTER Declarative

An ENTER Declarative is used to introduce a block of symbolic coding in "Own Code" language into the COBOL Procedure Division. Each ENTER Declarative occupies an entire Section in the Declarative area. The Section header consists of the Section name, the keyword SECTION, followed by a period and then the ENTER statement. The ENTER statement consists of the keywords ENTER MODAL. The ENTER coding in "Own Code" language is executed only by a PERFORM verb in the Procedure Division which refers to the section name. The "Own Code" instructions must begin on the line after the section header. The keywords ENTER COBOL must follow the final "Own Code" instruction. The next line must be either a new section header or the keywords, END DECLARATIVES.

##### 4.3.5.2 Own Code Formats

"Own Code" may be used any number of times within the main body of a COBOL Procedure Division or in the Declarative Section. Each time it is used it must be preceded by the words ENTER MODAL and ended with the words ENTER COBOL. The programmer must use the following formats.

The format for the ENTER MODAL is free form in that no fixed format is assumed by the translation division of MOBILE.

The format for each "Own Code" instruction in an ENTER statement is fixed, however, as follows:

| <u>Column No.</u> | <u>Item</u>                |
|-------------------|----------------------------|
| 1-6               | COBOL sequence number      |
| 7-12              | Symbolic location          |
| 14-19             | Operation code             |
| 21-76             | Variable field and remarks |
| 77-80             | Program Identification     |

The format for ENTER COBOL is:

| <u>Column No.</u> | <u>Item</u>           |
|-------------------|-----------------------|
| 1-6               | COBOL sequence number |
| 14-24             | ENTER COBOL           |

#### 4.3.5.3 Own Code Implementation

The translation of "Own Code" instructions to CAPS is a three-pass assembly process. The instructions appear to Run 1.4A as 14 word field data input card images on magnetic tape. They are processed one at a time.

An ENTER USAGE will now be defined for future use as those "Own Code" instructions bound in the beginning with the key words ENTER MODAL and in the end with the keywords ENTER COBOL. There may be many ENTER USAGES within a COBOL program.

##### 4.3.5.3.1 Pass I

During the first pass a Symbol table is created, and the mnemonic operational codes within each image are converted to numeric. At the same time a relative line counter is maintained within the Symbol table. The first three words of each image are replaced by a two word CAP containing only CAP type, op code, and part number.

The variable field within each image is scanned for two consecutive blanks which terminate the field. If they are found, the remainder of the card image with the exception of the card count is discarded since it constitutes only remarks which are not processed by the MOBILE system other than appearing in the BSP listing. If they are not found, this means the variable field is continued on the next card image. Similar processing is done until card images for this ENTER USAGE have been exhausted or two blanks have been found.

Each processed card image along with the Symbol table, is now a pseudo-CAP which is the main output of Pass I. Upon completion of Pass I, the Symbol table is sorted on the characters of the name for use in Pass II.

#### 4.3.5.3.2 Pass II

Each pseudo-CAP is now processed by passing its variable fields over the Symbol table until either a match is found or the table has been exhausted. For each match a Mobile Data operand for a self reference plus or minus, (\*±), is inserted into the CAP. If each variable field within a pseudo-CAP leads to a match, a finished or complete CAP can be formed. If any variable field within a pseudo-CAP leads to a no match, that variable field must be carried along with its incomplete CAP to Pass III.

In either case each CAP must be headed by a literal name entry, to allow it to go through the MOBILE system. Each block of CPS in turn must be headed by a Generator Call header. When Pass II over the current ENTER USAGE has been completed, one of three paths of processing must be followed:

1. If more ENTER USAGES exist, then Pass I and Pass II processing over subsequent USAGES must be performed.
2. If no more ENTER USAGES and no references to data-names in Working Storage requiring a third Pass exist, Run 1.4B is entered.
3. If no more ENTER USAGES but references to data-names in Working Storage requiring a third Pass, then Pass III must be entered.

#### 4.3.5.3.3 Pass III

During Pass III all incomplete CAPS, (those with variable fields) are processed by passing them over the Data Name List as many times as there are memory loads of Data Name List. This results in either finding a match or not.

For each match a unique location word for this data name in the Data Design Table is obtained. The variable field is discarded and the location word is appended to its CAP.

For a no-match an entry is inserted into the BSP correction table using the card count mentioned in Pass I. A mobile data operand specifying constant

zero is then inserted into that portion of the CAP to which the variable field belongs. The user can spot check his Assembly listings and binary patch where necessary to avoid a recompilation.

Processing continues until all of the incomplete CAPS are converted into either complete CAPS or CAPS with location words attached which require further information in the form of data replies and hence are not yet complete.

In both cases a literal name header precedes each CAP and each output block of CAPS is headed with an ENTER generator call header. The exit to Run 1.4B is then performed.

#### 4.3.5.4 Own Code Restrictions

- A. An ENTER MODAL may be used in a Declarative Section or in the Main Body of the Procedure Division. The only difference is that in the Declarative Section a Section name followed by the keyword SECTION and a period must precede the key words ENTER MODAL for each ENTER USAGE.
- B. The Own Code instructions must begin on the next line after the ENTER MODAL statement, which can be ended by a comma, a period or a space. After the final symbolic instruction there must be a line which contains a COBOL sequence number, and the keywords ENTER COBOL. The Procedure Division continues on the next line, starting in column 8 or 12 depending on whether a new Procedure name is given or the previous paragraph is continued.
- C. The formats described in 4.3.5.2 must be followed.
- D. Two consecutive spaces terminate the variable field, and anything which follows is considered to be remarks.
- E. The Variable fields in "Own Code" may refer to symbolic locations within its ENTER USAGE or to data names in the Working Storage Section of the Data Division.
- F. The variable fields in "Own Code" may not refer to data names, procedure names, or special names used elsewhere in the COBOL Program other than those data names in Working Storage. References to symbolic locations in other ENTER USAGES are not allowed.
- G. Whenever a qualified data name in Working Storage is referenced, one and only one space must precede and follow the keyword IN or OF.
- H. Qualification of a data name in Working Storage, if necessary, must be used but unnecessary qualification is permitted but not advised.
- I. The pseudo-op ETC allowing continuation of the variable field across multiple cards is allowed. This will be useful in qualification of a Working Storage data name.

- J. References to data names in Working Storage with OCCURS clauses are not permitted.
- K. The "Own Code" user can only use the operational codes listed in Appendix B.
- L. The variable field of an "Own Code" instruction can only contain numeric integers or 30 character data names. Numeric expressions involving addition, subtraction, multiplication, division or exponentiation are not allowed.
- M. In the use of the OCT pseudo-op only numeric integers can be used.
- N. No Input-Output instructions are allowed.

#### 4.3.5.5 Own Code Tables

##### 4.3.5.5.1 OW1N

- A. Table name - "Own Code".
- B. Table symbol - OW1N, set by Run 1.3.
- C. No. of words/entry - 14 words.
- D. No. of entries - one for each "Own Code" instruction used with an ENTER verb.
- E. Table function - result of Hollerith to FIELDATA translation, used by Run 1.4A.
- F. Table format
 

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| word 1:       | COBOL sequence number or blank (05)                                                    |
| word 2:       | Symbolic location or blank (05)                                                        |
| word 3:       |                                                                                        |
| bits 1-6      | Blank (05)                                                                             |
| 7-24          | Operation code                                                                         |
| 25-36         | Blank (05)                                                                             |
| word 4:       |                                                                                        |
| bits 1-12     | Blank (05)                                                                             |
| 13-36         | 1st four characters of variable field                                                  |
| word 5 to 13: | Remaining characters of variable field, followed by remarks if any (6 characters/word) |
| word 14:      |                                                                                        |
| bits 1-12     | Final 2 characters of variable field                                                   |
| 13-24         | Binary card count                                                                      |
| 25-36         | Identification code-fieldata BB                                                        |

Note: First double 05 specifies termination of variable field.

**4.3.5.5.2 None**

- A. Table name - Symbol Table
- B. Table symbol - None assigned, core contained "Own Code".
- C. No. of words/entry - 2
- D. No. of entries - 1800 maximum, dependent upon number of instructions with symbolic locations.
- E. Table function - used to convert "Own Code" instructions to CAP instructions
- F. Table format

word 1:                   Symbolic name in FIELDATA, right justified.  
Master space fill at left end (00).

word 2: Relative line count of symbolic name.

4.3.5.5.3 PS1N

- A. Table name - Pseudo-CAP
- B. Table symbol - PS1N, assigned by Run 1.4A
- C. No. of words/entry - dependent upon size of variable field
- D. No. of entries - one for each "Own Code" instruction
- E. Table function - intermediate output of "Own Code" to CAP translation process
- F. Table format

word 1:

bits 1-3      CAP type  $\begin{cases} = 0, & \text{internal instruction} \\ = 1, & \text{pseudo-op instruction} \end{cases}$

4-9 Machine instruction

|           |             |                                                                                                                              |
|-----------|-------------|------------------------------------------------------------------------------------------------------------------------------|
| 10-11, 13 | Part number | $\left\{ \begin{array}{l} = 001 \text{ A portion} \\ = 010 \text{ M portion} \\ = 100 \text{ I portion} \end{array} \right.$ |
|-----------|-------------|------------------------------------------------------------------------------------------------------------------------------|

12 Statement bit

14-36                      zero

word 2: zero

word 3:

|               |                                                               |
|---------------|---------------------------------------------------------------|
| bits 1-12     | Card count                                                    |
| 13-36         | First four characters of variable field                       |
| words 4 to N: | Remaining characters of variable field<br>(6 characters/word) |

#### 4.3.5.5.4 OC1N

- A. Table Name - Own Code Generator Call
- B. Table Symbol - OC1N set by Run 1.4A
- C. No. of words/entry - 3 or 4
- D. No. of entries - variable
- E. Table function - processed through the MOBILE system,  
resulting in CAP input to Run 8.
- F. Table format

word 1: Generator call header

|          |                                      |
|----------|--------------------------------------|
| bits 1-9 | Generator name                       |
| 10-19    | Block number (for ordering of calls) |
| 20-27    | Number of entries                    |
| 28-36    | Item size                            |

word 2: Generator call header

|           |      |
|-----------|------|
| bits 1-21 | zero |
| 22-36     | OISN |

word 3: Literal Name

|          |                        |
|----------|------------------------|
| bits 1-3 | Literal name key (011) |
| 4-11     | Zero                   |
| 12-13    | Number of words (10)   |
| 14-36    | Zero                   |

word 4: CAP

|           |                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| bits 1-3  | CAP type $\begin{cases} = 0, & \text{internal instruction} \\ = 1, & \text{pseudo-op instruction} \end{cases}$           |
| 4-9       | Machine instruction                                                                                                      |
| 10-11, 13 | Part number $\begin{cases} = 001 & \text{A portion} \\ = 010 & \text{M portion} \\ = 100 & \text{I portion} \end{cases}$ |

|           |                                                                                            |
|-----------|--------------------------------------------------------------------------------------------|
| 12        | Statement bit                                                                              |
| 14        | Zero                                                                                       |
| 15        | Decrement bit $\begin{cases} = 0, \text{ increment} \\ = 1, \text{ decrement} \end{cases}$ |
| 16-36     | Address portion                                                                            |
| word 5:   | <u>CAP</u>                                                                                 |
| bits 1-15 | CAP symbol                                                                                 |
| 16-36     | Index-modifier portion                                                                     |
| word 6:   | <u>Location word</u>                                                                       |
| bits 1- 3 | Location word key (000)                                                                    |
| 4-12      | Jump word                                                                                  |
| 13-19     | zero                                                                                       |
| 20-24     | DDT memory load number                                                                     |
| 25-36     | DDT relative address                                                                       |

Note: The above format of words 3 to 6 constitutes the internal equivalent of a reference in "Own code" to a data name in Working Storage.

Words 3 to 5 are sufficient for a reference to a symbolic location within this ENTER USAGE.

The appropriate format is repeated for each entry within the generator call.

#### 4.3.5.6 Instruction Field Interpretation

The field interpretations are as follows:

A = Address field

I = Index field

M = Modifier field

( ) = May be coded but not required

N = Octal integer

Each "Own code" instruction results in a two word CAP of which only the first word is given in the octal equivalent form. The second word is zero.

#### 4.3.5.6.1 Group 1 – Internal Instructions

| <u>Title</u> | <u>Field Interpretation</u> | <u>CAP Equivalent</u> |
|--------------|-----------------------------|-----------------------|
| ADB          | A, (I), M                   | 02 40000 00000        |
| ADD          | A, (I), M                   | 01 20000 00000        |
| ADM          | A, (I), M                   | 01 30000 00000        |
| CAM          | A, (I)                      | 01 10000 00000        |
| CLA          | A, (I)                      | 01 00000 00000        |
| CLS          | A, (I)                      | 01 40000 00000        |
| CSM          | A, (I)                      | 01 50000 00000        |
| CYL          | A, (I)                      | 03 50000 00000        |
| CYS          | A, (I)                      | 03 40000 00000        |
| DVD          | A, (I), M                   | 02 20000 00000        |
| DVL          | A, (I), M                   | 02 30000 00000        |
| HLT          | (A), (I), (M)               | 00 00000 00000        |
| LGA          | A, (I)                      | 00 30000 00000        |
| LGM          | A, (I)                      | 00 20000 00000        |
| LGN          | A, (I)                      | 00 40000 00000        |
| LOD          | A, (I), M                   | 05 10000 00000        |
| LXS          | A, (I)                      | 05 30000 00000        |
| MLR          | A, (I)                      | 02 10000 00000        |
| MLY          | A, (I)                      | 02 00000 00000        |
| MOV          | A, (I), M                   | 05 20000 00000        |
| MSK          | A, (I)                      | 05 50000 00000        |
| NRM          | A, (I)                      | 03 70000 00000        |
| RPA          | A, (I)                      | 05 40000 00000        |
| RPT          | A, (I), M                   | 00 10000 00000        |
| SBB          | A, (I), M                   | 02 50000 00000        |
| SBM          | A, (I), M                   | 01 70000 00000        |
| SEN          | A, (I), M                   | 00 50000 00000        |
| SHL          | A, (I), M                   | 03 00000 00000        |
| SHR          | A, (I)                      | 03 20000 00000        |
| SLL          | A, (I), M                   | 03 10000 00000        |
| SNR          | A, (I), M                   | 00 70000 00000        |
| SNS          | A, (I), M                   | 00 60000 00000        |

| <u>Title</u> | <u>Field Interpretation</u> | <u>CAP Equivalent</u> |
|--------------|-----------------------------|-----------------------|
| SRL          | A, (I)                      | 03 30000 00000        |
| STR          | A, (I)                      | 05 00000 00000        |
| SUB          | A, (I), M                   | 01 60000 00000        |
| TRC          | A, (I)                      | 04 70000 00000        |
| TRL          | A, (I), (M)                 | 04 10000 00000        |
| TRN          | A, (I)                      | 04 60000 00000        |
| TRP          | A, (I)                      | 04 40000 00000        |
| TRS          |                             | 04 20000 00000        |
| TRU          | A, (I), (M)                 | 04 00000 00000        |
| TRX          | A, I, M                     | 04 30000 00000        |
| TRZ          | A, (I)                      | 04 50000 00000        |

#### 4.3.5.6.2 Group 2 – Instructions

| <u>Title</u> | <u>Field Interpretation</u> | <u>CAP Equivalent</u> |
|--------------|-----------------------------|-----------------------|
| BES          | N                           | 10 10000 00000        |
| BSS          | N                           | 10 20000 00000        |
| OCT          | ±N                          | 10 30000 00000        |
| PZE          | A, I-M                      | 10 50000 00000        |
| MZE          | A, I-M                      | 10 60000 00000        |
| REM          |                             | IGNORED               |
| END          |                             | IGNORED               |

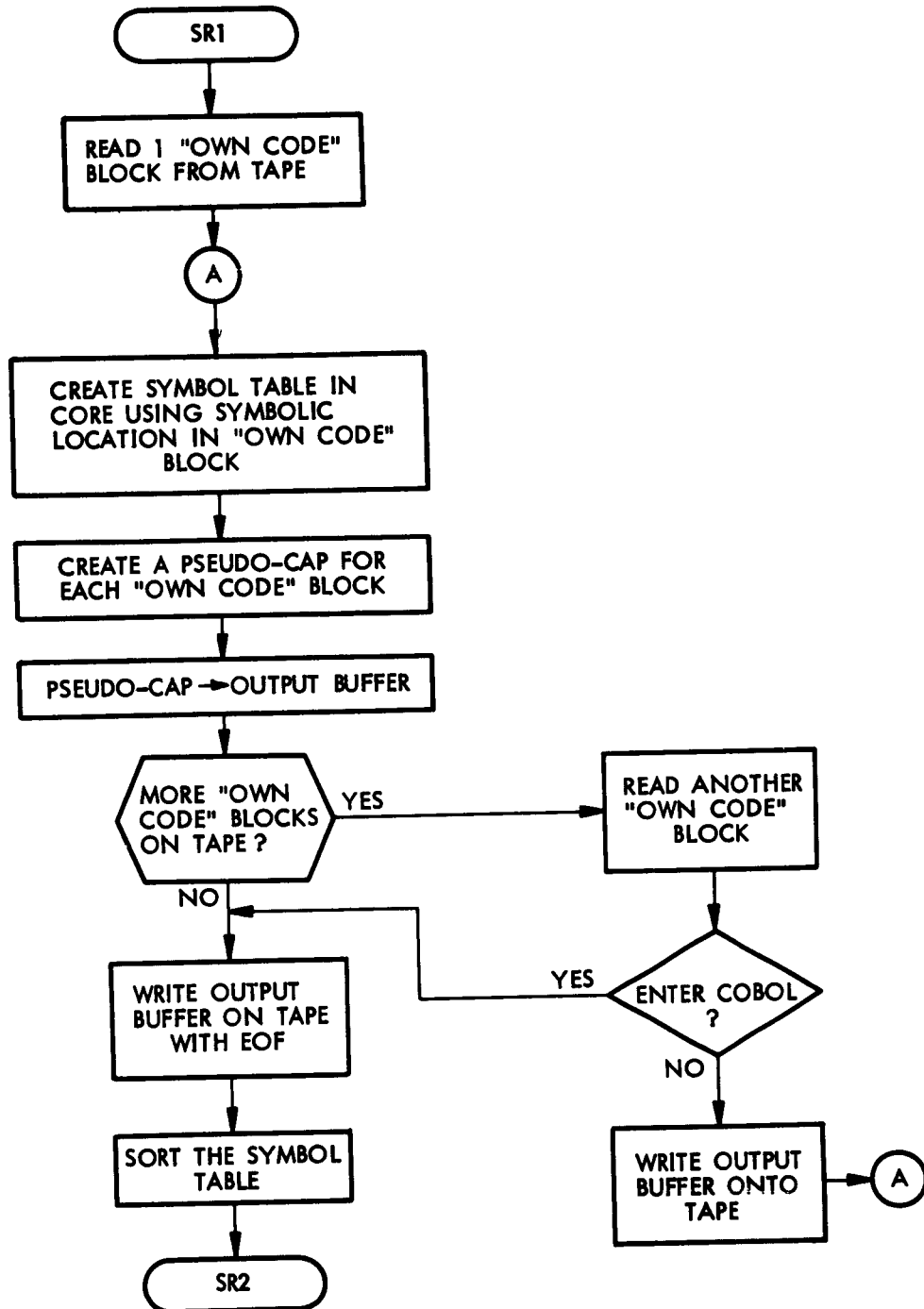


Figure 4-3. Own Code (Pass I)

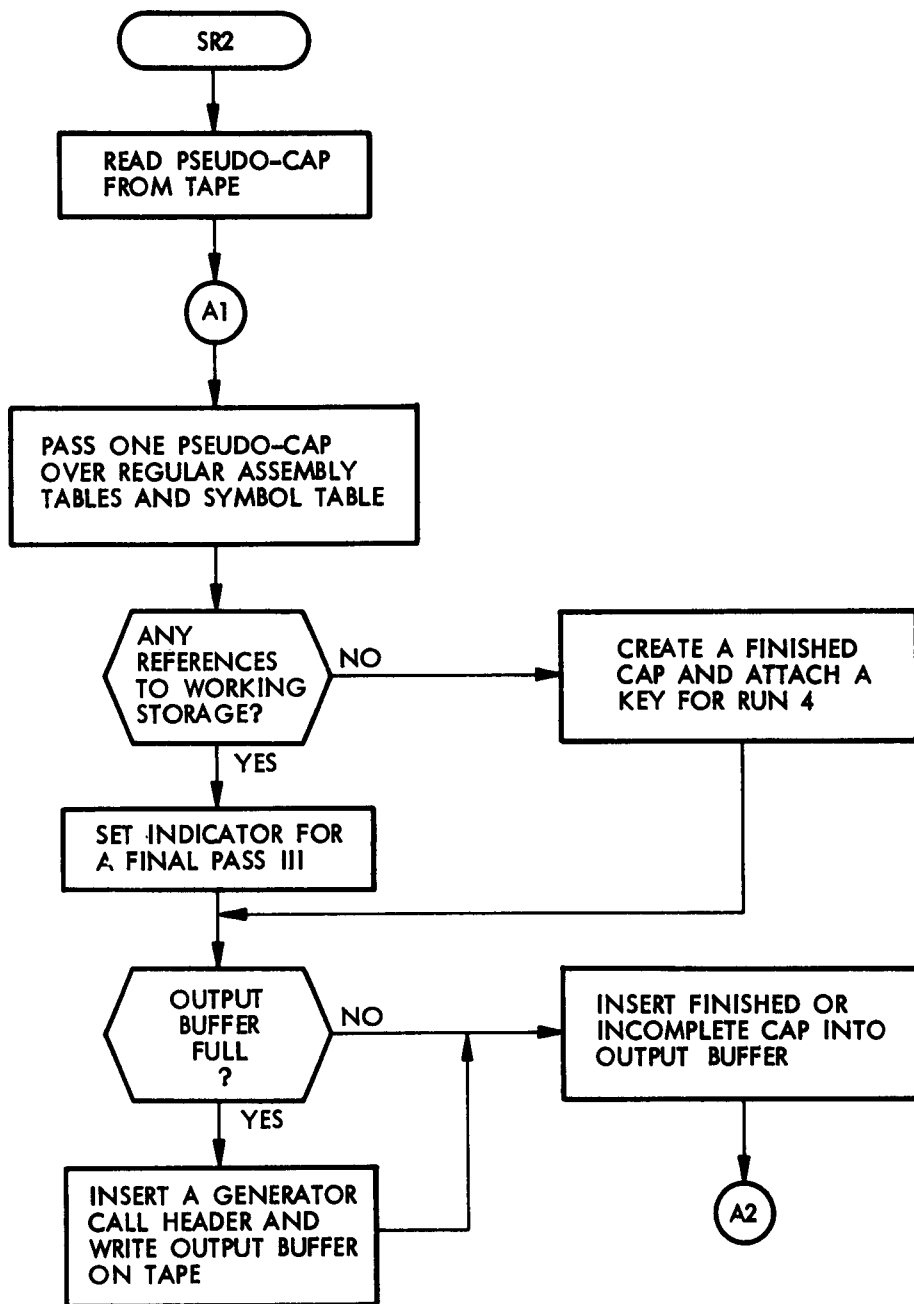


Figure 4-4. Own Code (Pass II)

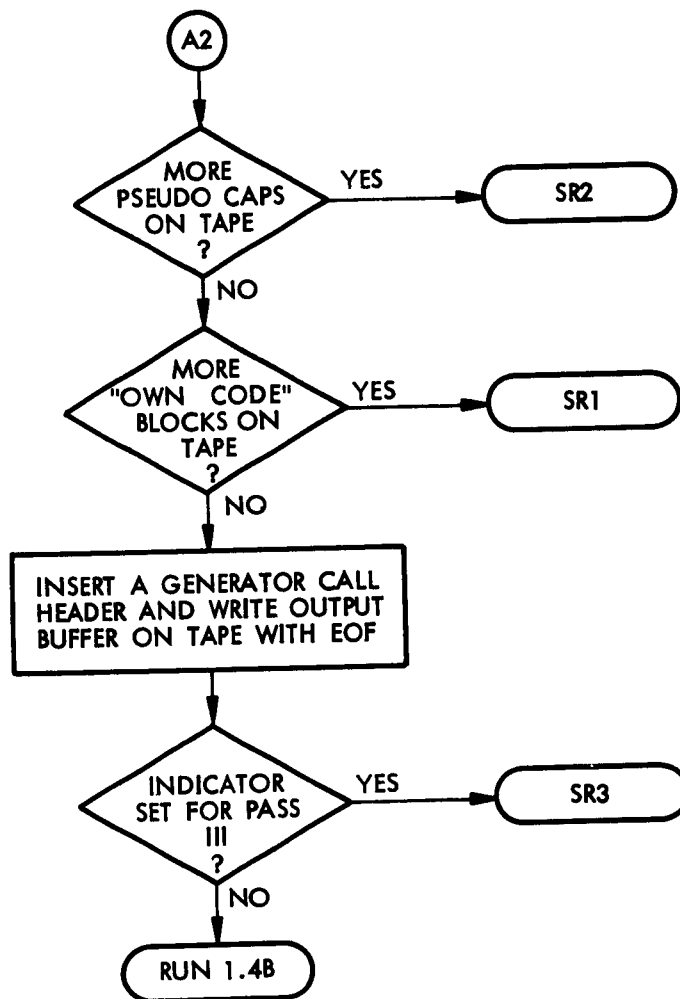


Figure 4-4. Own Code (Pass II) (Cont.)

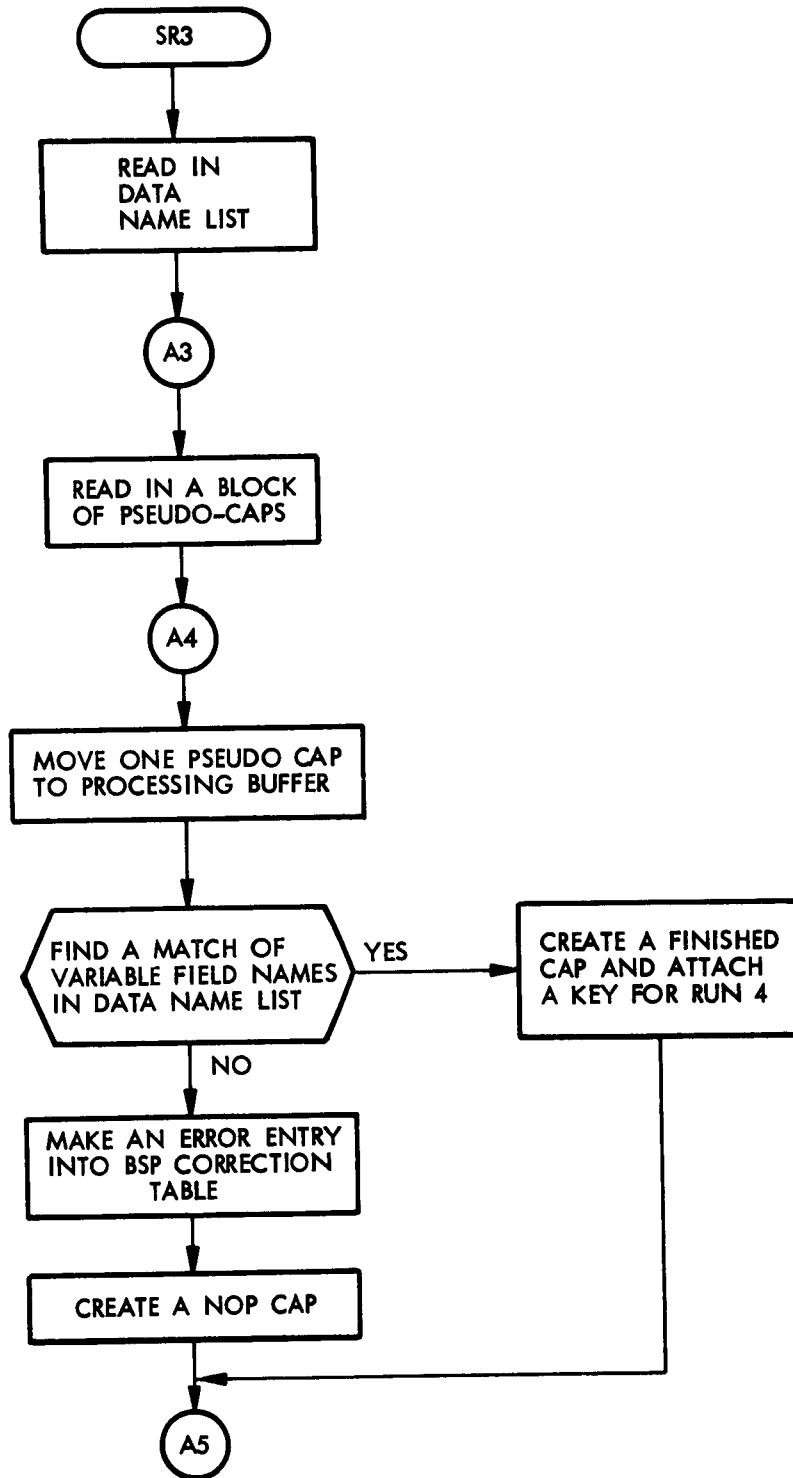


Figure 4-5. Own Code (Pass III)

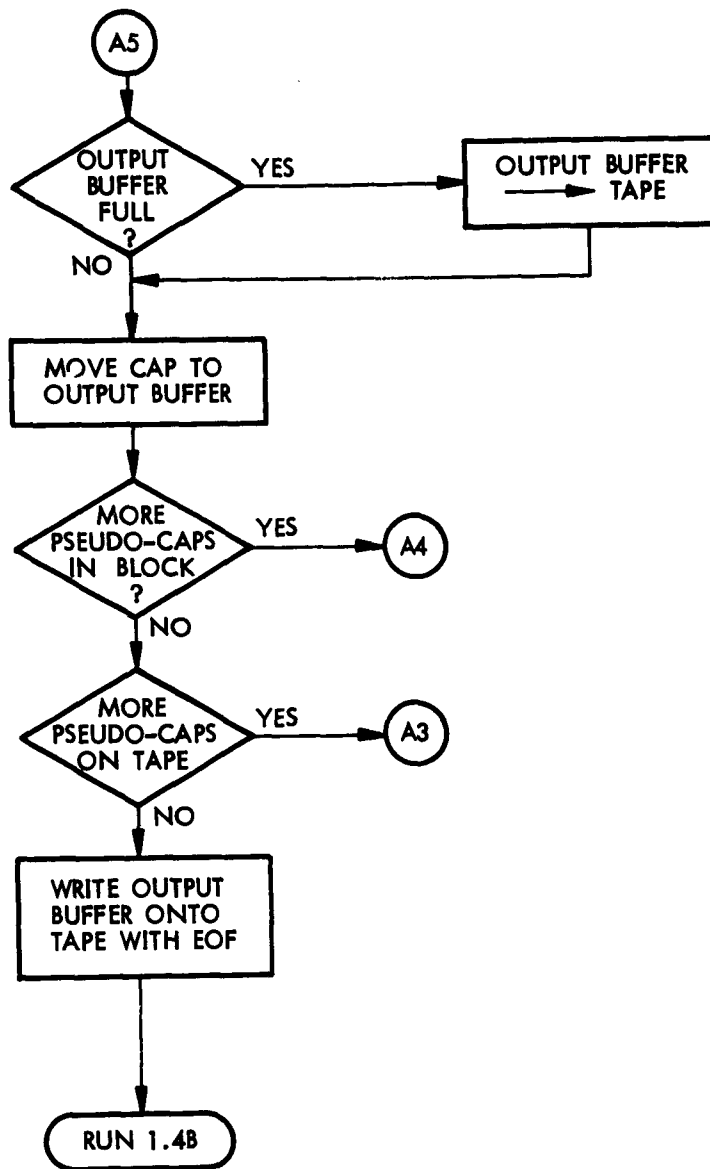


Figure 4-5. Own Code (Pass III) (Cont.)

#### 4.3.6 Run 2 Modifications

##### 4.3.6.1 Former Function

The primary function of Run 2 is twofold, sorting and merging. Minor editing is also done at this time. Associated with each generator call produced by Run 1.3 is a generator number. These generator numbers are serial numbers assigned to each generator. The sort that occurs in Run 2 uses this generator number as a key to arrange the generator calls such that all calls for the same generator occur simultaneously, and prior to the next generator.

##### 4.3.6.2 Modified Function and Internal Sort Description

A major change has taken place in the use of an internal sort to replace the "binary" sort. Each compressed generator call has an entry in a control table. This table gives the size of the compressed generator call and its starting location in core at the time of sorting. By using the "internal" or "in place" sort, the buffer used by this control table during sorting is being halved. After the control table has been formed, any editing to be done is performed at this time. In the internal sort, while the table is being examined and the compressed generator call with the lowest serial number is being found, an exchange is made between the first table entry and the entry for the generator call with the lowest serial number. The generator with the lowest serial number is then moved to the output buffer. Each time this output buffer is filled, (approximately 600 generator calls) it is written on the appropriate tape. Tape 5 is the tape in this case. The size of the table is decreased by one. The next lowest serial number is found, and its entry exchanges places with the second table entry. This process continues in this manner until the control table is completely sorted.

Another major change is in the size of the input buffer which has been reduced considerably because of the decrease in available storage area. It is expected that there will be no more than two buffer loads of the size now available. In this case, it was necessary to add another phase to Run 2, namely, the merge phase. This is not a separate section of Run 2 but an integrated part of Run 2 Coding. If Run 2 control finds that the amount of information on Tape 5 will overflow the input buffer, an indicator is set, and the same internal sort explained

above takes place on one buffer load. This sorted information is now written as a string on Tape 6. Then the remaining generator calls are read into the input buffer and a control table is set up for them.

The size of the table is increased by one. Before sorting the control table, a block of Tape 6 is read into a separate area. (This block contains the first string of sorted information mentioned previously). For the first generator call from that block, a negative table entry is made to the control table. A pass through the sort is made and the lowest entry is found.

If this is a positive entry, an exchange is made, the table size is decreased by one, the compressed generator call is moved to be written on Tape 5, and a return is made for another pass of the sort. If this is a negative entry, an exchange is made; the table size is not decreased by one; that the compressed generator call is moved to be written on Tape 5; a new negative entry is made from the string on Tape 6 and takes the place in the table of the previous negative table entry; and a return is made to the sort phase. When there are no more positive entries or compressed generator calls from Tape 6, the last buffer load is written out with the WRITE END OF FILE (WEF) set.

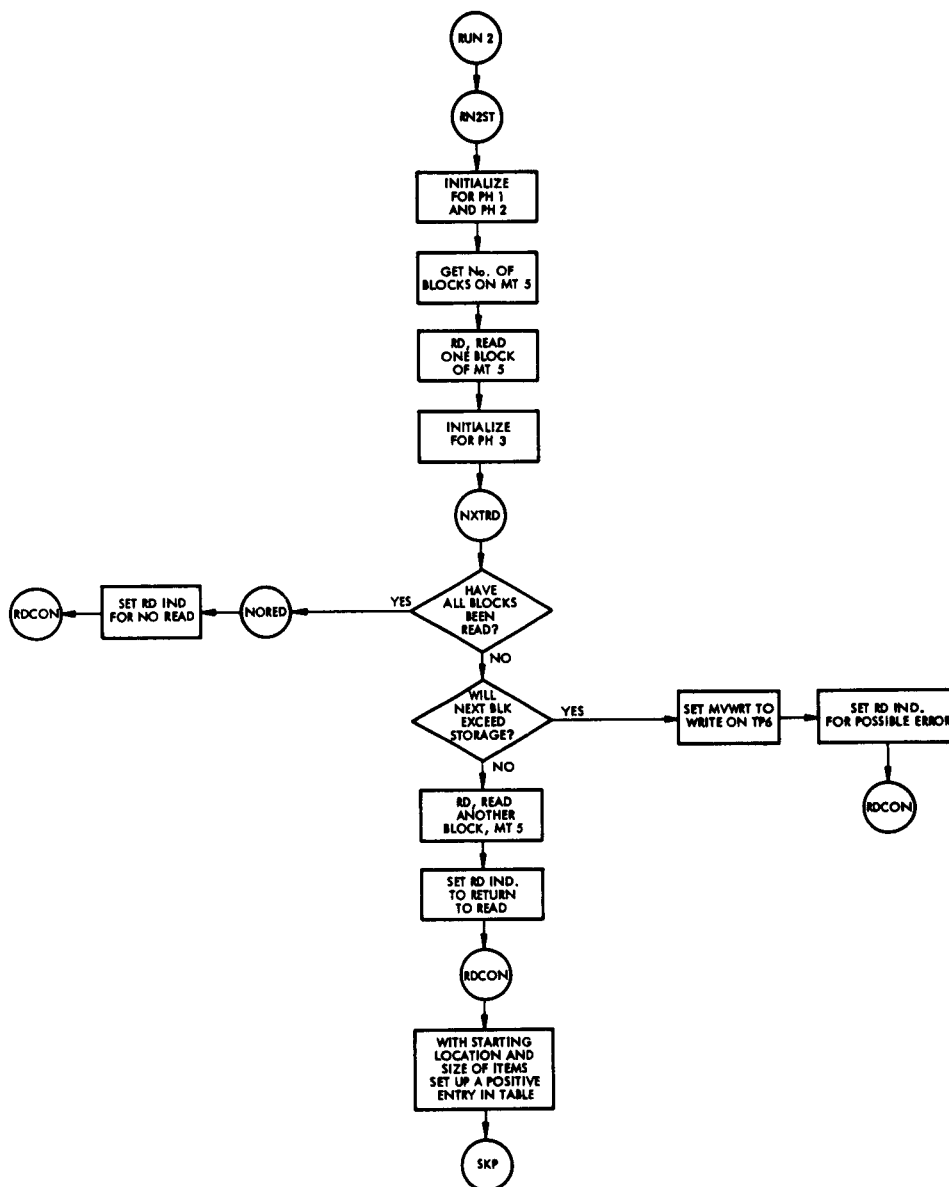


Figure 4-6. Internal Sort Flowchart

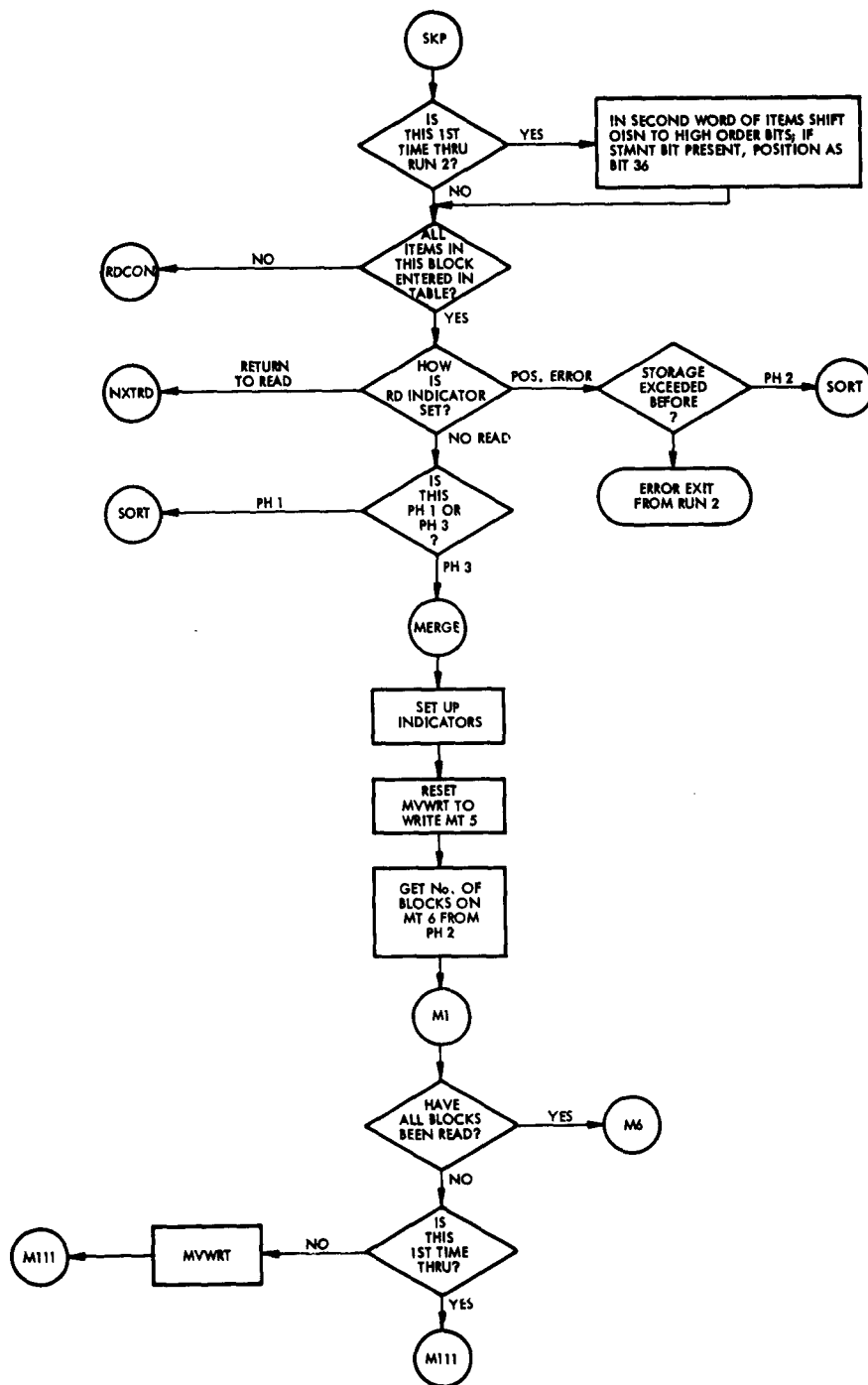


Figure 4-6. Internal Sort Flowchart (Cont.)

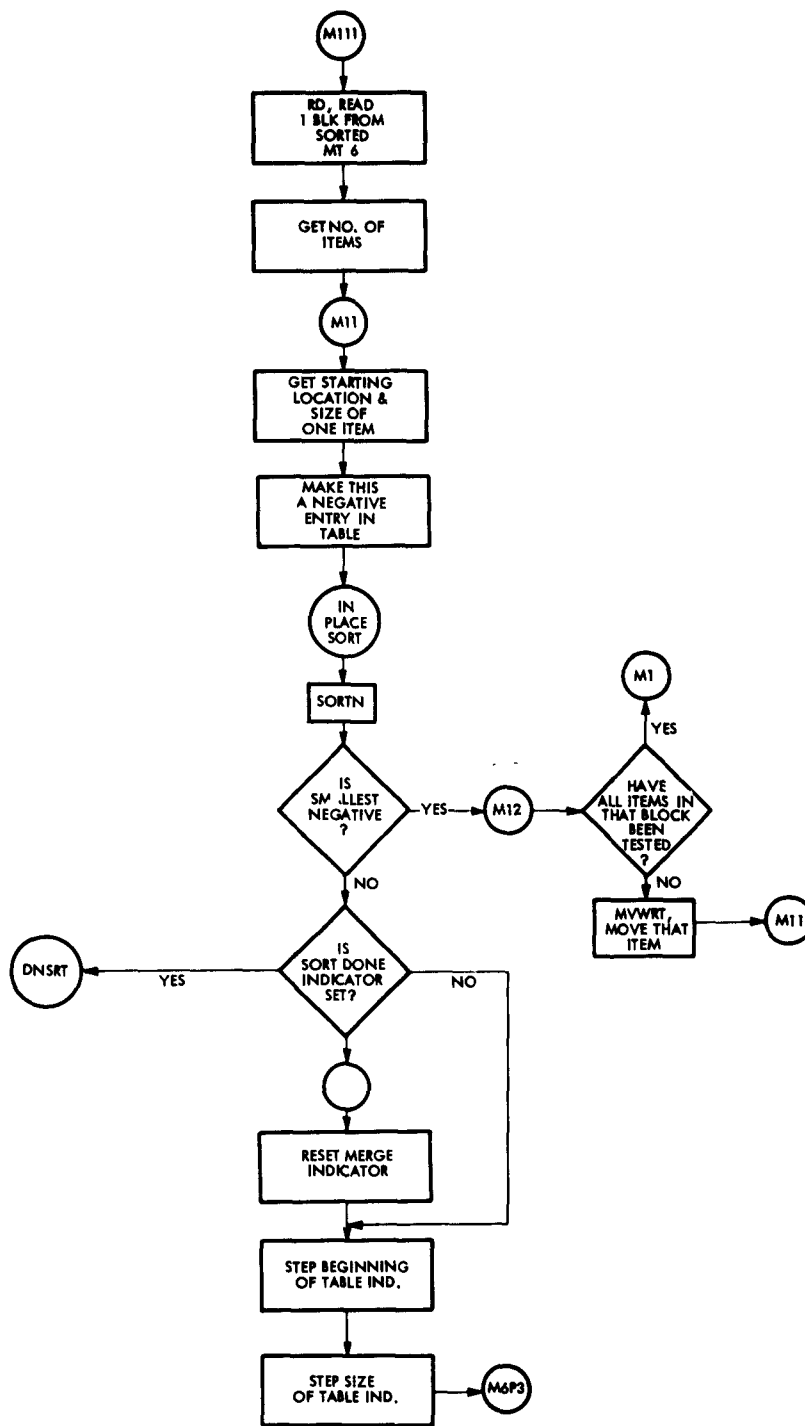


Figure 4-6. Internal Sort Flowchart (Cont.)

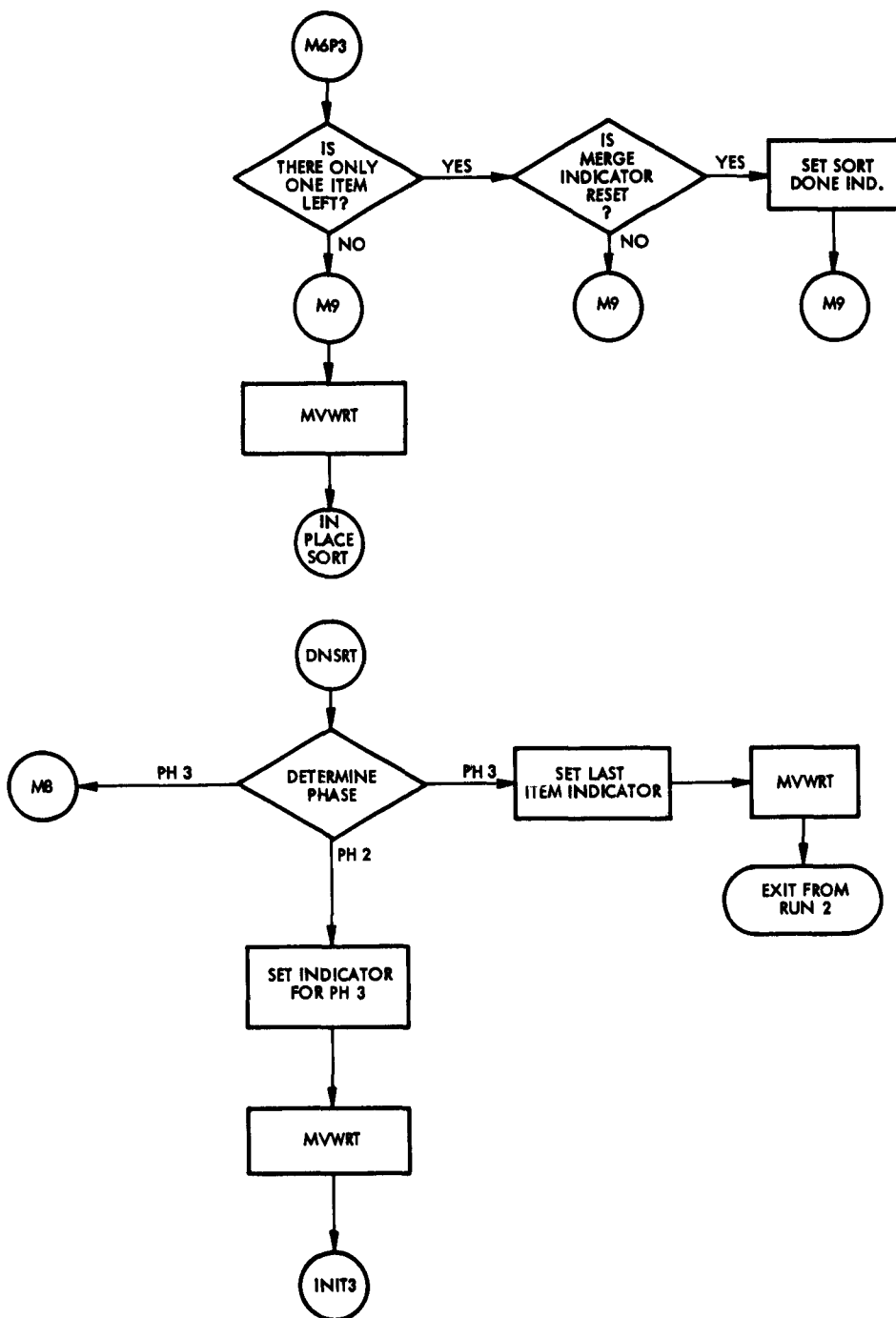


Figure 4-6. Internal Sort Flowchart (Cont.)

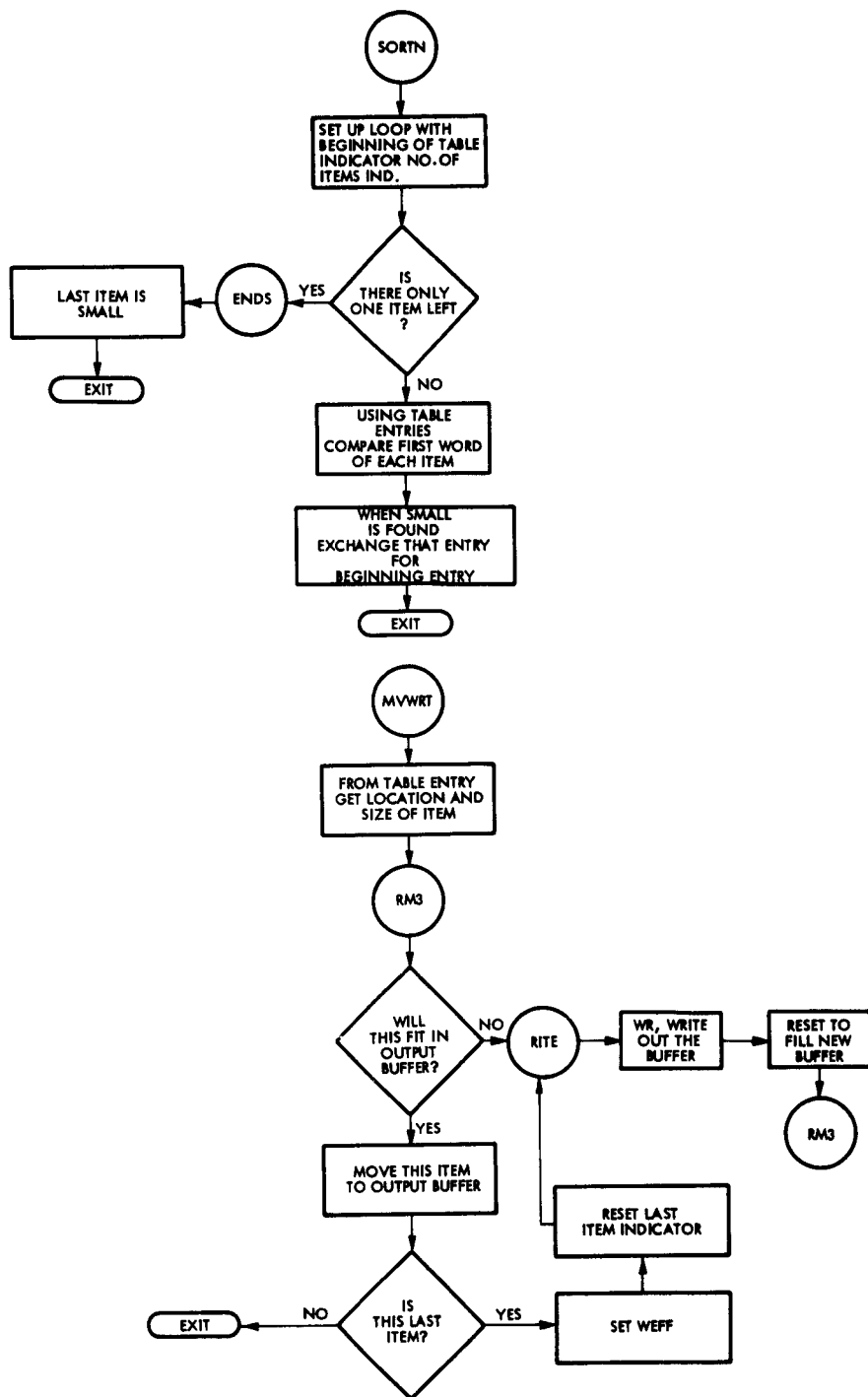


Figure 4-6. Internal Sort Flowchart (Cont.)

#### 4.3.7 Run 4 Modifications

Run 4 appends data replies to compressed generator calls produced by Run 1.3, thus forming complete generator calls. In addition, certain concomitant functions are performed, including making entries to the Data Analyzer Table for each appended data reply.

The 9400 version of Run 4 has been modified to accommodate the core memory capacity of the MOBIDIC Computer. The modification consists of reducing the size of certain buffer areas of core; replacing the "double" buffer technique by a "single" buffer technique for reading in and processing the compressed generator calls.

#### **4.3.8 Run 3 Modifications**

##### **4.3.8.1 Run 3 Service Routines**

###### **4.3.8.1.1 Purpose**

The main function of Run 3 is to read the generators into core, one at a time, feed the calls to the generators, one at a time, and accept output from the generators.

###### **4.3.8.1.2 Input**

1. Generator Calls – Variable length entries produced by Run 1.3 in response to verbs in the Source Program. They are sorted by generator name and have all necessary data replies appended to them.
2. File Description Table – Produced by Run 1.2 for the I/O Generator. It contains information from the Environment and Data Division concerning the files that have been selected by the Source Program.
3. Address Variable Table – Produced by Run 4 and contains a list of data-name subscripts in which changes to subscript values may be noted.
4. File Table – Used by the I/O Generator and contains a one-word entry for each file selected by the source programmer. It also contains information from the Procedure Division regarding the file's use as input and output.
5. Control Block – Contains information from the Environment Division about the source and object computers.

###### **4.3.8.1.3 Method**

Run 3 has a table in core to guide it in the selection of generators to process the generator calls. This table has a one word entry for each existing generator. A marker is inserted in this table to indicate which generators are required to process the Source Program input. Run 3 also adds markers for generators required to process one cycle and two cycle calls.

The communication between Run 3 and the generators is identical for all generators. Run 3 transfers to the generator from a calling sequence which resembles:

|     |     |   |                 |
|-----|-----|---|-----------------|
| A   | TRL | X | GO TO GENERATOR |
| A+1 | HLT |   | INPUT ADDRESS   |
| A+2 | HLT |   | OUTPUT ADDRESS  |
| A+3 |     |   | RETURN LOCATION |

"X" is the first location in each generator. The initial entrance to each generator comes at X.

When a generator wishes to return to Run 3, it transfers (with a TRL) to A+3. If the generator is returning to Run 3 in order to pass on some output, it must first place the starting address of the output in A+2. For each transfer to A+3 the generator must have previously loaded an indicator number into Index Register 4 to tell Run 3 what the output is. If a new generator call or a new generator segment (I/O gen. only) is indicated, Run 3 will return to X. In the case of a new generator call, the absolute address of the call will be in X+1. In all other cases, Run 3 will return by the equivalent of a TRS instruction. In Table 4-1 is a list of the codes, return addresses and other information required in communication with Run 3.

TABLE 4-1. INFORMATION REQUIRED in COMMUNICATION WITH RUN 3

| Octal Code<br>NO. in IR4 | Return<br>Address | Reason for Return<br>Run 3              |
|--------------------------|-------------------|-----------------------------------------|
| 0                        | C(PCS)            | XFISN table entry (7)                   |
| 1                        | C(PCS)            | Constant table entry (1) (2)            |
| 2                        | C(PCS)            | Subroutine call (2)                     |
| 3                        | C(PCS)            | Alter table entry (1) (2) (7)           |
| 4                        | C(PCS)            | Link table entry (2)                    |
| 5                        | C(PCS)            | Tape data table (2) (6)                 |
| 6                        | C(PCS)            | Macro entry (2)                         |
| 7                        | C(PCS)            | File requirement table entry<br>(2) (6) |

**TABLE 4-1. INFORMATION REQUIRED  
IN COMMUNICATION WITH RUN 3 (Cont.)**

| Octal Code<br>No. in IR4 | Return<br>Address | Reason for Return<br>Run 3       |
|--------------------------|-------------------|----------------------------------|
| 10                       | C(PCS)            | File data block (2) (6)          |
| 11                       | C(PCS)            | One cycle generator call (2)     |
| 12                       | C(PCS)            | Complete generator call (2) (7)  |
| 13                       | C(PCS)            | Junk Storage count (3) (7)       |
| 14                       | C(PCS)            | Multiple file data table (2) (6) |
| 15                       | C(PCS)            | Special storage count (3) (7)    |
| 16                       | "X"               | Exit for new generator call (4)  |
| 17                       | "X"               | Exit for new segment (5) (6)     |
| 20                       | C(PCS)            | Two cycle call (2) (7)           |

- (1) Constant or alter serial number which has been assigned by Run 3 will be in the accumulator on return to the generator.
- (2) Starting address of output item must be in A+2 when entering Run 3.
- (3) A+2 contains the absolute number of locations required by the generator, to be reserved in core during object running time for storage purposes.
- (4) When all generator calls have been processed this exit will cause Run 3 to read in the next required generator.
- (5) The desired segment must be indicated by putting a binary key in A+2.
- (6) Used only by I/O Generator.
- (7) Used only by non I/O Generators.

#### 4.3.8.1.4 Output

1. Batch Generator Output—Contains macros and entries to five tables:
  - a. XFISN
  - b. Constant
  - c. Subroutine call
  - d. Alter
  - e. Link

This output is sorted by Run 5.

2. Tape Data Table – Produced by the I/O generator and is used at object running time to keep track of tape usage by all files.
3. File Requirement Table – Produced by the I/O generator and contains information used by Run 7 to set aside proper buffer space in the Object Program for each file selected.
4. File Data Block – Produced by the I/O generator and is used by the object I/O coding in keeping track of the usage of the files selected in the Source Program.
5. One Cycle Calls – Calls on a following generator which are saved by Run 3 until the required generator is in core.
6. Complete Generator Call – Calls which must be recycled to Run 2 for resorting and through Run 4 for data replies.
7. Multiple File Data Table – Produced by the I/O Generator and is used by the object I/O coding in keeping track of the usage of multiple file tape if any were given in the Source Program.
8. Two Cycle Calls – Calls on a following generator which are saved by Run 3 until the required generator is in core. Calls for this group of generators can only be produced by other generators and not by any verb in the Source Program.

This describes the basic design of Run 3 for the MOBIDIC Computer. The required changes of Run 3 of the 9400 MOBILE I have been made.

#### 4.3.8.2 Input-Output Generator

##### 4.3.8.2.1 Purpose

The task of the generator is to inspect those parts of the Source Program relating to input-output operations and to take the action necessary to insure that these operations will be performed in the Object Program. The Source Program is available to the generator in the form of compiler tables and the generator only indirectly produces the actual coding of the Object Program by creating more internal table entries. These table entries cause later runs to generate the proper machine coding.

#### 4.3.8.2.2 Input

All the inputs to the generator are supplied by the Run 3 control program. With the exception of the input-output generator calls, all inputs are already available in core memory when the generator is first entered. Inputs are as follows:

1. File Description Table—This table contains information from the Environment and Data Divisions of the Source Program concerning the files that have been selected by the source programmer.
2. File Table—This table is made up of one word entries for each file "selected" by the source programmer and contains information from the Procedure Division regarding the file's use as input and output.
3. Multiple File Table—This table contains one entry for each "Multiple File Tape Contains..." statement in the Environment Division of the Source Program.
4. Control Block—This table contains information from the Environment Division about the source and object computers.
5. Compressed Generator Calls—These are variable length entries produced by Run 1.3 in response to I/O verbs in the Source Program. They are sorted in a special order and released to the generator, one at a time, by the Run 3 control program.

#### 4.3.8.2.3 Output

Outputs from the generator are all passed on to the Run 3 control program via a calling sequence. They consist of one to three binary tables and various internal table entries. The binary tables are to be placed into the Object Program with no alteration. The various internal table entries will be used by later runs to generate object coding and reserve the proper input-output buffer areas. Outputs are as follows:

1. TAPDAT Table—A table that will be used by the object I/O coding in keeping track of tape usage by all files.

2. **MFRDAT Table**—A table that will be used by the object I/O coding in keeping track of the usage of multiple file tapes if any were given in the Source Program.
3. **FILDAT Table**—A table that will be used by the object I/O coding in keeping track of the usage of the actual files selected in the Source Program.
4. **File Requirement Table**—A table used by Run 7 to set aside the proper buffer space in the Object Program for each file selected.
5. **Subroutine Calls**—Variable length entries made to a table used by Run 6 to initiate the process by which precoded subroutines will appear in the object coding.
6. **Macros**—Variable length entries made to a table used by Run 6 to initiate the process by which precoded instructions will appear in the object coding.
7. **One Cycle Generator Calls**—Variable length entries made to a table used by later (non I/O) generators, to produce coding not directly associated with input-output.
8. **Link Table Entries**—Fixed length entries made to a table used by Run 6 in its optimization of the object coding.
9. **Constant Table Entries**—Fixed length entries made to a table used by Run 6 to generate a constant "pool" in the object coding.

#### 4.3.8.2.4 Input-Output Generator Segments

Due to the size of the tables required by the generator, it has been necessary to divide it into five sections, each of which will overlay the previous one. There is no need for recycling, since each section is used once and only once.

Communication with Run 3 control is the same in all five sections and through use of the prescribed calling sequence all five sections of the generator will be entered at their first location.

All five sections have a short section of initialization coding which prepares the section for communication with Run 3 control, and all five sections

have a short subroutine which does the actual transfer of control to Run 3. This eliminates the problem of initializing many locations and provides a uniform procedure for all transfers. Likewise, all five sections of the generator have a small control section which enters the various tasks of the section in a prescribed order, making each section modular and more easily alterable.

#### 4.3.8.2.5 Description of the Five Segments of I/O Generator

##### Segment 1

Segment 1 does not process any generator calls but merely creates certain tables and does some preliminary work on the File Description Table. Tasks performed are as follows:

1. Assigns tape units to all files, checking for discrepancies.
2. Creates a TAPDAT table for Object Program.
3. Creates a MFRDAT table for Object Program if multiple file reel(s) are present.
4. Reduces all sizes of data in the File Description Table to words and computes sizes of desired buffer areas, current record areas, etc., altering the File Description Table in so doing.
5. Creates a File Requirement Table for Run 7.
6. Further alters the File Description Table by reducing label record information.

##### Segment 2

Segment 2 processes all the I/O generator calls, and produces all the macros for the object coding, but does not produce any subroutine calls. All other output used in later runs is produced in Segment 2. Tasks performed are as follows:

1. Processes USE generator calls and constructs a table of USE FISNS associated with label record USE procedures.
2. Processes USE generator calls and constructs a table of USE FISNS associated with error USE procedures.
3. Processes OPEN INPUT, READ, OPEN OUTPUT, WRITE and CLOSE generator calls, producing a macro for each call and placing the

names of the desired skeleton subroutines in a table to be passed on to Segment 3.1.

4. Produces constant table entries where needed in macros.
5. Produces link table entries for all READ macros that have a conditional fall through and a jump to another FISN.
6. Produces one cycle, special PERFORM generator calls to duplicate AT END procedures where they are not stated.
7. Produces one cycle, special MOVE generator calls to isolate value of data-name in the WRITE record-name AFTER/BEFORE ADVANCING data-name LINES.
8. Sets a special list in that file's File Description Table entry if any file has a CLOSE File-name WITH LOCK statement.

### Segment 3

Segment 3 produces subroutine calls only. Segment 3 uses the table of subroutine names passed on from Segment 2 and produces a subroutine call for each such subroutine name. Segment 3 has been further segmented in the following manner:

Segment 3.1 processes the table of subroutine names passed on from Segment 2 and produces a temporary subroutine call. This temporary subroutine call includes all the necessary deletion words and the fixed header only. Segment 3.1 also produces a temporary subroutine call for all other subroutines required by the subroutine originally requested by Segment 2. It is also the task of Segment 3.1 to insure that only those subroutines designated "duplicatable" will appear more than once in the Object Program. The temporary subroutine calls are passed on to Segment 3.2

Segment 3.2 fills out the temporary subroutine call provided by Segment 3.1. Operands necessary for the subroutine being requested are appended to the temporary subroutine call and the resultant subroutine call is given to Run 3 control to be passed on to Run 6.

### Segment 4

Segment 4 produces the File Data Table only, from information in the File Description Table.

#### 4.3.8.3 Modification of the Non I/O Generators

The generators listed below have been modified for the MOBIDIC Computer: MOVE ALL, SIMPLE PERFORM, ALTER, GO TO, EXAMINE, COMPLEX PERFORM, STOP and EXIT. Included below is a description of the implementation of the ACCEPT and DISPLAY Generators for the MOBIDIC Compiler.

##### 4.3.8.3.1 The ACCEPT Generator

ACCEPT data-name FROM device-name.

###### 4.3.8.3.1.1 Purpose

The ACCEPT generator processes the equivalent of a single ACCEPT statement in the Source Program and determines the type of object code needed to perform the input operations.

###### 4.3.8.3.1.2 Input

The generator processes a single generator call one at a time. The body of the call consists of the location word of the receiving area, an encoded keyword corresponding to the devices (if the "device-name" clause is present), and a complete description of the receiving area.

###### 4.3.8.3.1.3 Method (Object Code Subroutine Used)

All input-output orders will be executed by use of the Input-output subroutines which are inserted into the Object Program by the I/O Generator. The Input-output routines are always entered at subroutine AUXSD1. In addition, the ACCEPT generator may insert one of the following subroutines:

###### ACCEPT 1

The ACCEPT 1 subroutine initializes calling sequence to AUXSD1, and delays return until Input-output indicates the release of the input data.

###### ACCEPT 2

The ACCEPT 2 subroutine double-buffers reads of Hollerith cards, converts the cards into FIELDATA and assembles data in the indicated receiving area.

Both of these subroutines have the same calling sequence. The TRL loads the count for acceptance to IR4. The word following the TRL contains the input instruction with a sign mode indicator (ISN or NISN). In order to produce a MACRO CALL for this calling sequence, the following information has to be extracted from the generator call: Device Address (D), Count (C), Sign Mode (M), Receiving Address (A), Instruction Code (I) and Subroutine (S).

#### 4.3.8.3.1.4 Generator Processing

A. The data description of the receiving area is tested for the following conditions.

- 1) If the data unit is synchronized.
- 2) If the starting bit is equal to zero and the size is a multiple of 36 bits.

If either of the above conditions is true, an indicator is set (IND) indicating that to read directly into the receiving area may be possible.

B. The encoded device from the generator call is compared to a table containing the allowable devices, the actual machine address (D) of each device, and a jump to a routine which will process this device. The Word-Switch-Register (WSR) is not considered as a device.

C. Control is passed to the following process routines:

- 1) Word-Switch Register (WSR). A MACRO CALL is produced to supply a calling sequence, HLT, CLA and WSR. Control is then passed to the PACK subroutine (see page 4-96 of the 1st Quarterly Report), to store the result.
- 2) Paper-Tape-Reader (PTR)
  1. M is NISN
  2. Base is binary, 1 = ROX (Read Octal) C = size in WORDS.  
Base is FIELDATA, 1 = RAN (Read Alphanumeric)  
C = size in characters.
  3. S = ACCOT 1
  4. If IND is not set, a buffer is reserved in the OPEN storage which is equal to the size of the data unit, and A is equal to the address of the buffer. If IND is set, A is equal to the address of the data unit.

### 3) Card-Reader (CDR)-ISN

1. M = ISN
2. I = RAN
3. C = size in cards
4. If IND is set and size is a multiple of card image, A is equal to the address of the card image, otherwise, a buffer is reserved.
5. S = ACCPT 1

### 4) Card-Reader (CDR)

1. If the base of the field is binary, control is passed to CDR-ISN.
2. M = NISN
3. I = RAN
4. S = ACCPT 2
5. A is determined as in subsection 4.3.8.4.1.4 C.2.
6. C = size in cards.

D. After these routines have returned control to the main line generator, the MACRO CALL for the calling sequence is produced (from A, I, S, M, C), and a subroutine call is produced for S.

E. If the data has not been accepted directly into the receiving area, control is passed to the PACK routine (see page 4-96 of the 1st Quarterly Report), to move the data from the buffer area into the correct location.

### 4.3.8.3.2 The DISPLAY Generator

#### 4.3.8.3.2.1 Purpose

The DISPLAY generator resembles the ACCEPT generator in input, general construction, output, and use of the Input-output subroutines. There are two entrance points which correspond to the DISPLAY and STOP statements.

DISPLAY Entrance Point.

### Object Code Subroutines Used

1. ACCPT 1
2. DSPLY 1

The DSPLY routine is a general double-buffered card punch routine which assembles card images in binary in a card buffer, converts them to Hollerith (if needed), and transfers control to I/O for execution of the instruction. The calling sequence for this routine is a TRL, with the number of items to be displayed, loaded in Index Register 4, and followed by one word for each item, containing the address and size of that item. The size is in bits. Two entrance points are associated with this routine, for either binary or Hollerith.

#### 4.3.8.3.2.2 Method

The encoded device from the generator call is compared to a table of allowable devices. Control is transferred to the following devices for special processing.

**Accumulator**—By use of the ISOLATE subroutine, MACRO CALL will be produced to unpack the single item into the accumulator. This will be followed by a HLT MACRO CALL. If the single item is an integer literal, it will be converted to binary and entered into the constant table.

**Line-Printer or Flexowriter**—The object code will make use of the subroutine ACCPT 1. If any item is packed at object running time, in order to be displayed directly on the device, a buffer area is reserved the size of which is equal to the item. MACRO CALLS will be produced by ISOLATE to cause the data to be moved to the buffer, and A is set equal to the buffer address. For every item to be displayed, a calling sequence MACRO is created with the following:

I = WOK (Write Octal) if base is binary.

I = WAN (Write Alphanumeric) if base is FIELDATA.

C = size in words.

M = NISN

A final calling sequence will be requested which will write out a carriage return. Literals are entered into the constant table in FIELDATA.

Paper-Tape-Punch—The processing is the same as the Flexowriter and the Line-Printer, with no final carriage return.

Card-Punch-ISN—The DSPLY subroutine will be used by the object code. This subroutine enters at the binary punch entrance point. As in FLX, a buffer is used as DISPLAY address for an item which cannot be displayed directly. One word is included in the calling sequence for every item containing the address and the size of the item. Literals are converted to binary and are entered into the constant table.

Card-Punch-NISN—The processing is the same as CARD-PUNCH-ISN except that if the base of the field to be displayed is FIELDATA, the object subroutine will be entered at the Hollerith entrance point, and literals will remain in FIELDATA.

STOP: Entrance Point

The Macros needed to display the correct item are supplied by the DISPLAY section. One of the following two MACRO CALLS is then produced:

- 1) If RUN option, HLT TRU \* - 1 sequence requested.
- 2) If not the RUN option a simple HLT Macro call produced.

#### 4 3.9 Run 5 Modifications

##### 4.3.9.1 Former Function

The primary function of Run 5 is twofold, sorting and merging. The purpose of sort-merge is to rearrange the output of the generator calls in an order that will reflect the normal sequence of the running code and the correct tabular form of certain generator output. Being sorted on their sequence numbers, the MACROS, which will be converted to CAPS by Run 6, are put back into the order prescribed in the Procedure Division. The keys for the sort will cause the entries for the various tables to be grouped together properly.

##### 4.3.9.2 Modified Function

The major changes made for Run 5 involve the use of the internal sort, a reduction in the size of the input buffer, and the addition of a merge phase. The internal sort which replaces the "binary" sort is of the same type as that used in Run 2.

The input which is read in from Tape 4 may be an entry for any one of the following tables:

| <u>Table</u>              | <u>Key</u> | <u>Symbol</u> | <u>Tape for<br/>Output</u> | <u>Number of<br/>Output Words</u> |
|---------------------------|------------|---------------|----------------------------|-----------------------------------|
| 1. Data Analysis          | 00         | TK45          | 3                          | Variable                          |
| 2. Address Function Table | 02         | TX45          | 6                          | 2 words*                          |
| 3. Address Function Table | 04         | TY45          | 6                          | 2 words*                          |
| 4. Address Function Table | 05         | TZ45          | 6                          | 2 words*                          |
| 5. Constant Table         | 20         | TC35          | 6                          | Variable                          |
| 6. XFISN Table            | 30         | TD35          | 6                          | 1 word**                          |
| 7. Alter Table            | 40         | TF35          | 6                          | 3 words                           |
| 8. Subroutine Calls***    | 50X        | T635          | 6                          | Variable                          |
| 9. Link Table             | 60         | TH35          | 6                          | 3 words                           |
| 10. MACRO Table           | 70         | MA35          | 5                          | Variable                          |

\*First word removed from input record

\*\*First and third words removed from input record

\*\*\*Entries from this table sorted on 9 bits; note; key written 50X.

When the control table is made up for Run 5, an added indicator will be found. This indicator will appear for each entry of a subroutine call so that the sort may be set up to sort on 9 bits instead of the first six. Each entry gives the starting address of the item to be examined. Each item is at least three words long. This has been done so that all items will be sorted on the first six bits of the first word and also all of the next two words.

Run 5 performs in a similar manner to Run 2 in that the merge phase 1 is used only when more than one input buffer load of information is to be sorted. If no merge is needed a sort is performed to find the items with the lowest serial number. If this item has the same key as the previous lowest item, the new item is moved to the output buffer. When the buffer is full, the information contained within is written out on the appropriate tape. If this item does not have the same key as the previous lowest item, then this new item is "held" until the output buffer is written with the WEF set. Then this new item is tested; its output device is inserted into the calling sequence, and its word suppression indicator is set. The suppression is done while moving the items to the output buffer.

If a merge is necessary, no word suppression indicator for the first batch of input is set, and all items are written in one file on Tape 6. The string on Tape 6 is then merged with the second batch of input, and each item is written out on the appropriate tape.

#### 4.4 TAPE ALLOCATION AND USAGE

There have been two main areas of modification with respect to tapes in the process of producing the MOBIDIC Compiler. First, the segmentation of the runs has produced in some cases entities which must be treated as separate runs. Second, the need to accommodate a smaller memory has increased the use of tapes for storage of both the intermediate and final output. The creation of intermediate output and the consequent modifications to use scratch tapes for this output has been done in such a manner that optimum tape usage is ensured and that the problem of table spills is eliminated.

In all cases, the final output of any MOBILE Run remains as listed below.

#### 4.4.1 Current Tape Assignments and Usage at Compilation

Figure 4-7 indicates the tape assignment for the primary input and output data of the major phases of the program. The Systems Tape, (1), containing the compiler program itself, and the Source Program Tape (2), are used solely as input tapes and therefore can be file protected against accidental writing during a compilation. One tape (3) is reserved as a general input and output tape for all intermediate tables generated by one phase of the compiler and required as input by another phase. Throughout the compilation, the remaining three tapes (4, 5, 6) are required as intermediary tapes for the processing of data. The type data contained is dependent on the particular stage of compilation. At the end of the compilation however, one tape (5) contains the Object Program and another (6) contains the compiler listing information.

Figure 4-8 shows for each run the particular manner in which tapes 4, 5, and 6 are used during a compilation. There are a few specific rules which apply.

1. All tapes are read at all times in a forward direction. Read Reverse is never required.
2. No positioning of tapes by an individual run is at any time required.
3. All listed input tapes for a run can be assumed to be rewound at start of a run.
4. All listed output tapes for a run can be assumed to be positioned correctly at the time a write request is given.
5. A scratch tape will contain only one table at any particular point in time. All scratch tapes are automatically rewound at the completion of a read or write of that table.
6. If a tape is listed as both an input tape and a scratch tape for a run, the first write request determines its use as scratch and the tape will automatically be rewound at that time.

Figure 4-9 shows by run sequence the individual tables which appear on the listed input and output tapes. The data context of Tape 3 is also shown. This is both an input and output tape for all sections of the compiler program as has been mentioned before. All data is assumed to be read from it in a forward direction and it is never necessary for an individual run to position it.

# Symbol Key for Figure 4-9 and Table 4-2

|                       |   |                                                                    |
|-----------------------|---|--------------------------------------------------------------------|
| n                     | = | tape number                                                        |
| $\Delta$              | = | physical beginning of tape                                         |
| SS table name         | = | pertinent information exists prior to particular table referenced  |
| table name SS         | = | pertinent information exists after the particular table referenced |
| table name/table name | = | referenced tables are adjacent on tape                             |
| }                     | = | tape operations could logically occur simultaneously               |
| RD                    | = | Read                                                               |
| WR                    | = | Write                                                              |

Table 4-2 shows by run the detailed use of tapes by sequence of operation. Except where indicated by means of } it is assumed that the previous listed tape operation has been completed prior to the start of the next. The four character table symbol required by the MOBILE I/O routine calling sequence is shown next to the tape operation. A write or read of a table as shown is not meant to indicate that the entire table can either be or not be core contained at one time. It is only intended to show which tables appear on tape, and not whether they were placed there by one or more logical transfers of data.

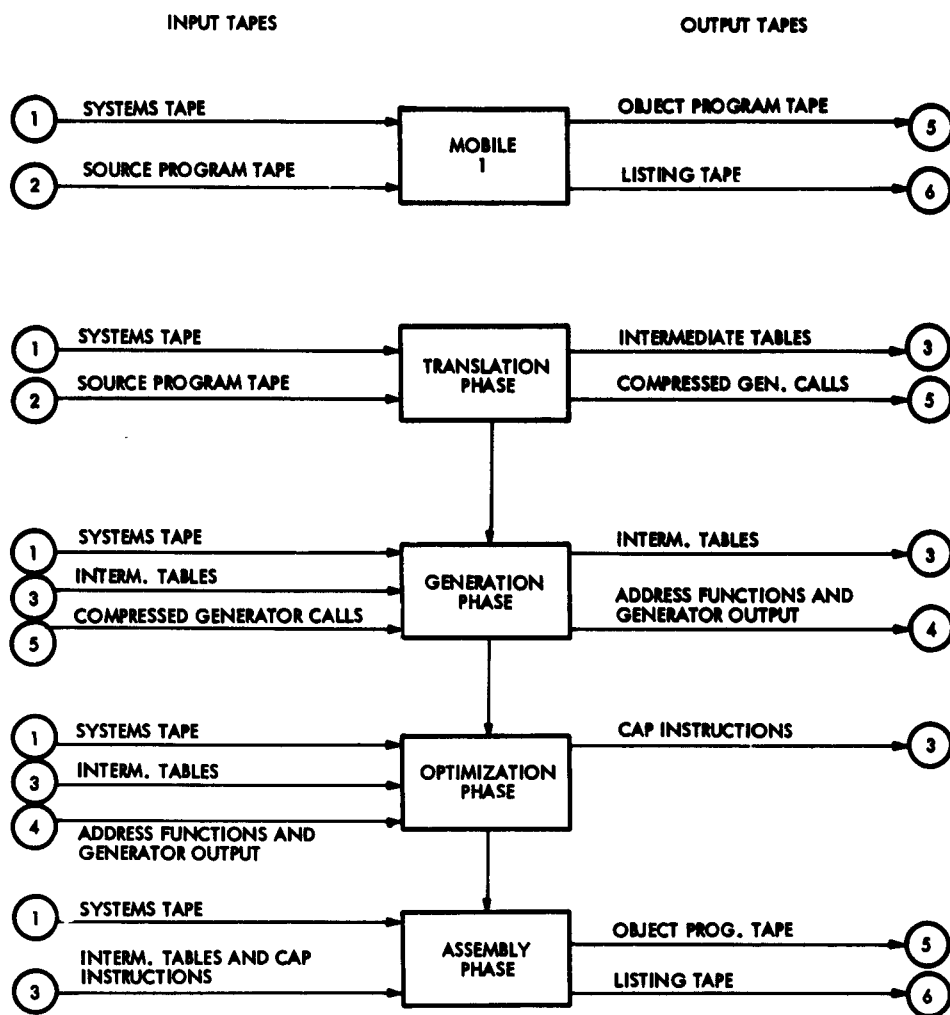


Figure 4-7. Input-Output Tape Assignment At Compilation

| PROGRAM                            | TAPES |        |         |
|------------------------------------|-------|--------|---------|
|                                    | INPUT | OUTPUT | SCRATCH |
| <u>TRANSLATION PHASE</u>           |       |        |         |
| RUN 1.0 INITIALIZATION             |       |        |         |
| RUN 1.1 ENVIRONMENT DIVISION PROC. |       |        |         |
| RUN 1.2 DATA DIVISION PROC.        |       |        | 4,5,6   |
| RUN 1.3 PROCEDURE DIVISION PROC.   |       | 5,6    |         |
| RUN 1.4 END-TRANSLATION            | 6     |        |         |
| <u>GENERATION PHASE</u>            |       |        |         |
| RUN 2 GENERATOR CALL SORT          | 5     | 5      | 6       |
| RUN 4 DATA INFORMATION RUN         | 5     | 4,6    |         |
| RUN 3 GENERATOR RUN                | 6     | 4,5    |         |
| <u>OPTIMIZATION PHASE</u>          |       |        |         |
| RUN 5 MACRO INSTRUCTION SORT       | 4     | 5,6    | 6       |
| RUN 6 OPTIMIZATION                 | 5,6   |        | 4,5,6   |
| <u>ASSEMBLY PHASE</u>              |       |        |         |
| RUN 7 PRE-ASSEMBLY                 |       | 4      |         |
| RUN 8 ASSEMBLY                     |       | 4,5,6  |         |

Figure 4-8. Usage of Tape 4, 5 and 6 for each Compilation Run

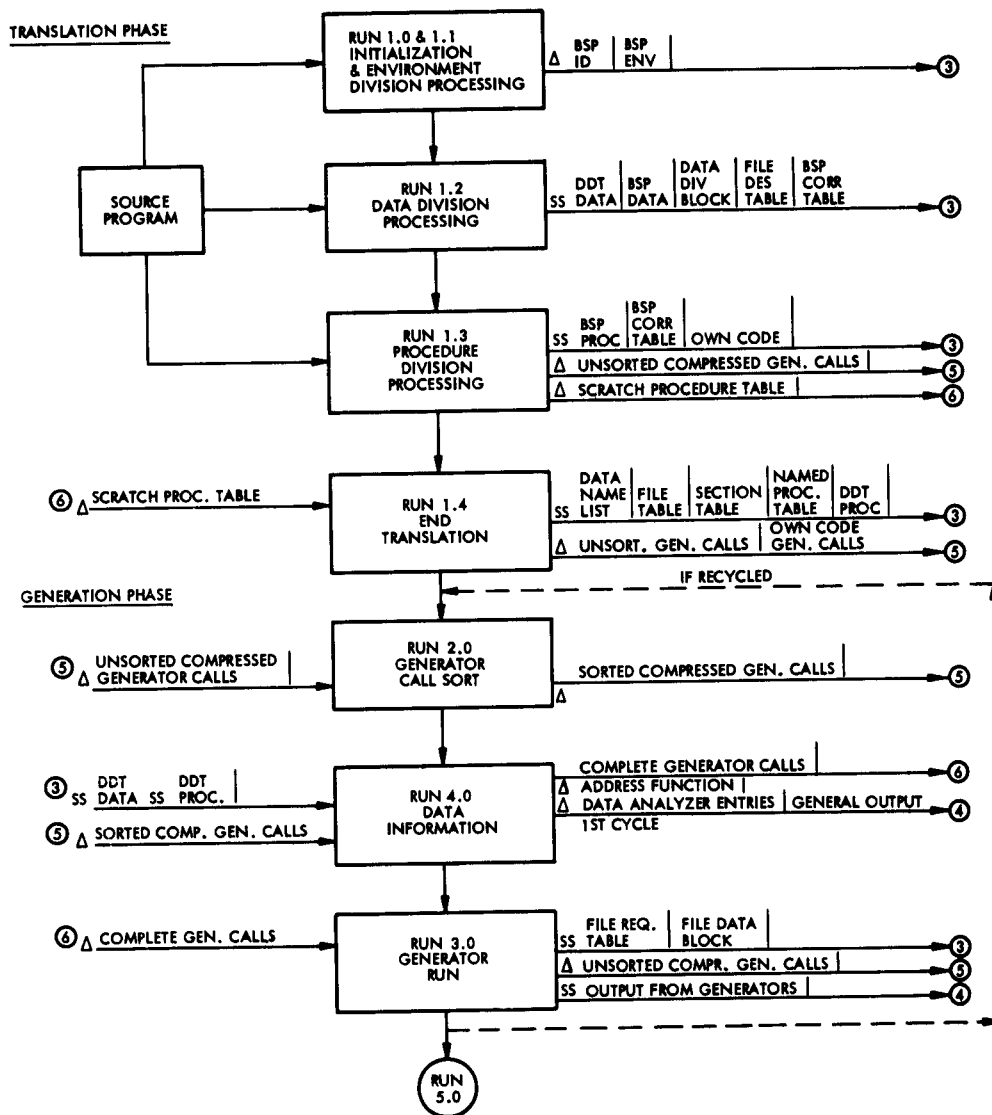


Figure 4-9. Tables On Input-Output Tapes At Compilation Runs

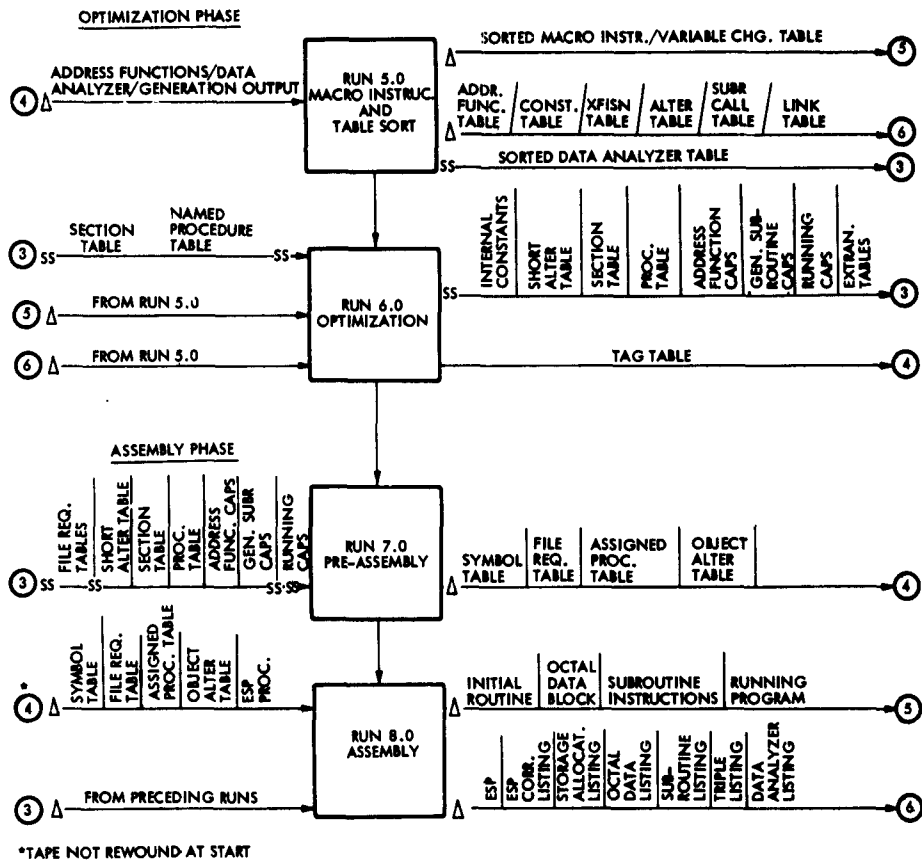


Figure 4-9. Tables on Input-Output Tapes at Compilation Runs (Cont.)

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION

| Run No. | Tape No. | Input                | Tape Operation | Output                    | Tape No. |
|---------|----------|----------------------|----------------|---------------------------|----------|
| 1.0     |          | Source Program       | WR LFON        | Δ Blocked Source Program  | 3        |
| 1.1     |          | Source Program       | WR LFON        | SS Blocked Source Program | 3        |
|         |          |                      | WR T12N        | SS Table Dump             | 3        |
| 1.2     |          | Source Program       | WR } TN00      | Δ Record DDT              | 6        |
|         |          |                      | WR } LM00      | Δ Data List               | 4        |
|         | 4        | Δ Data List          | RD LM00        |                           |          |
|         | 6        | Δ Record DDT         | RD } TN00      |                           |          |
|         |          |                      | WR } TQON      | Δ Unordered Copy Set      | 4        |
|         | 4        | Δ Unordered Copy Set | RD TQON        |                           |          |
|         |          |                      | WR TQ00        | Δ Ordered Copy Set        | 5        |
|         | 6        | Δ Record DDT         | RD } TN00      |                           |          |
|         | 5        | Δ Ordered Copy Set   | RD } TQ00      |                           |          |
|         | 3        |                      | RD T32N        | SS Table Dump             |          |
|         |          |                      | WR } TWON      | Data Design Table         | 3        |
|         |          |                      | WR } TW00      | Data Design Table         | 4        |
|         | 4        | Δ Data Design Table  | RD TW00        |                           |          |
|         |          |                      | WR LPON        | SS Blocked Source Program | 3        |
|         |          |                      | WR BXON        | SS Data Division Block    | 3        |
|         |          |                      | WR TVON        | SS File Description Table | 3        |
|         |          |                      | WR TXON        | SS BSP Correction Table   | 3        |

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION (Cont.)

| Run No. | Tape No. | Input                            | Tape Operation  | Output                                                                       | Tape No. |
|---------|----------|----------------------------------|-----------------|------------------------------------------------------------------------------|----------|
| 1.3     |          | Source Program                   | WR LP1N         | SS BSP Proc. Division                                                        | 3        |
|         |          |                                  | WR OW1N         | SS Own Code                                                                  | 3        |
|         |          |                                  | WR GA1N         | Δ Unsorted Comp. Gen. Calls                                                  | 5        |
|         |          |                                  | WR TH11         | Δ Scratch Proc. Table                                                        | 6        |
|         |          |                                  | WR TX1N         | SS BSP Correction Table                                                      | 3        |
| 1.4     |          |                                  | WR LN01         | SS Data Name List                                                            | 3        |
|         |          |                                  | RD OW1N         | SS Own Code                                                                  | 3        |
|         |          |                                  | WR TU01         | SS File Table                                                                | 3        |
|         |          |                                  | WR TA1N         | SS Section Table                                                             | 3        |
|         |          |                                  | RD TH11         |                                                                              |          |
|         | 6        | Δ Scratch Proc. Table            | WR TB1N         | SS Named Proc. Table                                                         | 3        |
|         |          |                                  | WR TD1N         | SS Data Design Table (Proc.)                                                 | 3        |
|         |          |                                  |                 |                                                                              |          |
|         |          |                                  |                 |                                                                              |          |
|         |          |                                  |                 |                                                                              |          |
| 2       | 5        | Δ Unsorted Compressed Gen. Calls | RD GA1N         |                                                                              |          |
|         |          |                                  | WR GD2N         | Δ Sorted Compr. Gen. Calls                                                   | 5        |
| 4       | 3        | SS DDT Data<br>SS DDT Proc.      | RD TWON<br>TD1N |                                                                              |          |
|         |          |                                  |                 |                                                                              |          |
|         | 5        | Δ Sorted Compressed Gen. Calls   | WR GB4N         | Δ Complete Gen. Calls                                                        | 6        |
|         |          |                                  | WR TA3N         | Address Function &<br>Data Analyzer Entries<br>Δ 1st Cycle<br>SS IF Recycled | 4        |

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION (Cont.)

| Run No. | Tape No. | Input                                                 | Tape Operation | Output                           | Tape No. |
|---------|----------|-------------------------------------------------------|----------------|----------------------------------|----------|
| 3       | 6        | Δ Complete Gen. Calls                                 | RD } GB4N      | SS File Requirement Table        | 3        |
|         |          |                                                       | WR } T19N      |                                  |          |
|         |          |                                                       | WR BY9N        | SS File Data Block               | 3        |
|         |          |                                                       | WR BX9N        | SS TAPDAT Table                  | 5        |
|         | 6        | SS Complete Gen. Calls                                | WR BZ9N        | SS MFPDAT Table                  | 5        |
|         |          |                                                       | RD } GB4N      | Δ Unsorted Compressed Gen. Calls | 5        |
|         |          |                                                       | WR } GA1N      |                                  |          |
|         |          |                                                       | WR TA3N        | SS Output From Generators        | 4        |
| 5       | 4        | Δ Address Function, Data Analyzer & Generation Output | RD TA3N        |                                  |          |
|         |          |                                                       | WR MA35        | Δ Sorted Macro Instruction       | 5        |
|         |          |                                                       | WR TJ45        | Δ Address Function Table         | 6        |
|         |          |                                                       | WR TC35        | SS Constant Table                | 6        |
|         |          |                                                       | WR TD35        | SS XFISN Table                   | 6        |
|         |          |                                                       | WR TF35        | SS Alter Table                   | 6        |
|         |          |                                                       | WR TG35        | SS Subroutine Call Table         | 6        |
|         |          |                                                       | WR TH35        | SS Link Table                    | 6        |
|         |          |                                                       | WR TK45        | SS Data Analyzer Table           | 3        |
| 6       | 6        | Δ Address Function Table SS                           | RD } TJ45      | Δ Compressed Add. Func. Table    | 4        |
|         |          |                                                       | WR } TJ66      |                                  |          |
|         | 4        | Δ Compressed Add. Function Table                      | RD TJ66        |                                  |          |
|         | 6        | SS Constant Table                                     | RD TC35        |                                  |          |
|         |          |                                                       | WR BC6N        | SS Internal Constant Block       | 3        |

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION (Cont.)

| Run No. | Tape No. | Input                           | Tape Operation | Output                   | Tape No. |
|---------|----------|---------------------------------|----------------|--------------------------|----------|
| 7       | 6        | SS XFISN Table                  | RD TD35        | SS Short Alter Table     | 3        |
|         | 6        | SS Alter Table                  | RD TF35        |                          |          |
|         |          |                                 | WR TF6N        |                          |          |
|         | 3        | SS Section Table                | RD TA1N        | SS Section Table         | 3        |
|         | 3        | SS Named Proc. Table            | RD TB1N        |                          |          |
|         |          |                                 | WR TA36        |                          |          |
|         |          |                                 | WR TB36        | SS Procedure Table       | 3        |
|         | 4        | Δ Compr. Address Function Table | RD } TJ66      | SS Address Function Caps | 3        |
|         |          |                                 | WR } MF6N      |                          |          |
|         | 6        | SS Subroutine Call Tables       | RD } TG35      | SS Generator Subr. Caps  | 3        |
|         |          |                                 | WR } MS6N      |                          |          |
|         | 6        | SS Link Table                   | RD TH35        | Δ Modified Macro         | 3        |
|         | 5        | Δ Sorted Macro Inst. SS         | RD } MA35      |                          |          |
|         |          |                                 | WR } TL66      |                          |          |
|         |          |                                 | WR } TK66      |                          |          |
|         |          |                                 | WR } MD66      |                          |          |
|         | 6        | Δ Modified Macro                | RD } MD66      | SS Running Caps          | 3        |
|         |          |                                 | WR } MR6N      |                          |          |
|         |          |                                 | WR             | SS Extran. Tables        | 3        |
|         | 3        | SS File Req. Table              | RD T19N        |                          |          |
|         | 3        | SS Short Alter Table SS         | RD TF6N        |                          |          |
|         | 3        | SS Section Table                | RD TA36        |                          |          |
|         | 3        | SS Proc. Table                  | RD TB36        |                          |          |
|         | 3        | SS Address Func. Caps SS        | RD MF6N        |                          |          |

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION (Cont.)

| Run No. | Tape No. | Input                    | Tape Operation | Output                        | Tape No. |
|---------|----------|--------------------------|----------------|-------------------------------|----------|
| 8       | 3        | SS Gen. Subr. Caps       | RD MS6N        |                               |          |
|         | 3        | SS Running Caps          | RD MR6N        |                               |          |
|         |          |                          | WR TM7N        | Δ Symbol Table                | 4        |
|         |          |                          | WR T197        | SS File Requirement Table     | 4        |
|         |          |                          | WR TB7N        | SS Assigned Procedure Table   | 4        |
|         |          |                          | WR BA7N        | SS Object Alter Table         | 4        |
|         | 3        | SS Data Div. Block       | RD BXON        |                               |          |
|         |          | SS BSP Corr. Table Data  | RD TXON        |                               |          |
|         |          | SS BSP Corr. Table Proc. | RD TX1N        |                               |          |
|         |          | Δ BSP                    | RD LFON        |                               |          |
|         |          | SS BSP                   | RD LPON        |                               |          |
|         |          | SS BSP                   | RD LP1N        |                               |          |
|         |          |                          | WR PA8N        | Δ Expanded Source Program     | 6        |
|         |          |                          | WR LP18        | SS ESP Procedure Division     | 4        |
|         |          |                          | WR PB8N        | SS ESP Correction Listing     | 6        |
|         | 4        | Δ Symbol Table           | RD TM7N        |                               |          |
|         | 4        | SS File Req. Table       | RD T197        |                               |          |
|         | 4        | SS Assigned Proc.        | RD TB 7n       |                               |          |
|         |          |                          | WR BZ8N        | Δ Initial Routine             | 5        |
|         |          |                          | WR PC8N        | SS Storage Allocation Listing | 6        |
|         | 3        | SS File Data BL.         | RD BY9N        |                               |          |
|         | 3        | SS Internal Constant     | RD BC6N        |                               |          |
|         | 4        | SS Object Alter Table    | RD BA7N        |                               |          |

TABLE 4-2. DETAILED USE OF TAPES AT COMPILATION (Cont.)

| Run No. | Tape No. | Input                 | Tape Operation | Output                     | Tape No. |
|---------|----------|-----------------------|----------------|----------------------------|----------|
|         |          |                       | WR } BD8N      | SS Octal Data Block        | 5        |
|         |          |                       | WR } PD8N      | SS Octal Data Listing      | 6        |
|         | 3        | SS Address Func. Caps | RD } MF6N      |                            |          |
|         |          | Gen. Subr. Caps       | RD } MS6N      |                            |          |
|         |          |                       | WR } BS8N      | SS Subroutine Instructions | 5        |
|         |          |                       | WR } PS8N      | SS Subroutine Listing      | 6        |
|         | 3        | SS Running Caps       | RD } MR6N      |                            |          |
|         | 4        | SS ESP Proc.          | RD } LP18      |                            |          |
|         |          |                       | WR } BR8N      | SS Running Program         | 5        |
|         |          |                       | WR } PE8N      | SS Triple Listing          | 6        |
|         | 3        | SS Data Name List     | RD LN01        |                            |          |
|         | 3        | SS Data Analyzer      | RD TK45        |                            |          |
|         |          |                       | WR PG8N        | SS Data Analyzer Listing   | 6        |

## 4.5 FORMAT AND USAGE OF THE COMPILER SYSTEMS TAPE

### 4.5.1 Contents of the Systems Tape

The Systems Tape is made up of labeled files; each file being a particular "Compiler Run" or auxiliary program.

#### 4.5.1.1 Programs on the Systems Tape

The following routines are on the Systems Tape:

| <u>Name</u> | <u>Function</u>                 | <u>File No.</u> |
|-------------|---------------------------------|-----------------|
| LABEL       | Tape Label                      | See Note No. 1  |
| LOADER      | Systems Loader                  | See Note No. 2  |
| SPINA       | Control and Input/Output        | 1               |
| CDIR        | Systems Directory               | 2               |
| SL02        | Systems Tape Generator          | 3               |
| M10N        | Initialization                  | 4               |
| M11N        | Environment Division Processing | 5               |
| M12N        | Data Division Processing        | 6               |
| M13N        | Procedure Division Processing   | 7               |
| M14N        | End Translation                 | 8               |
| M02N        | Generator Call Sort             | 9               |
| M03N        | Data Information Run            | 10              |
| M04N        | Generator Run                   | 11              |
| M05N        | Macro-instruction Sort          | 12              |
| M06N        | Optimization                    | 13              |
| M07N        | Pre-Assembly                    | 14              |
| M08N        | Assembly                        | 15              |

Note No. 1: An 8-word block is the first block on the Systems Tape. This block is the tape label. It contains the MOBIDIC Compiler Systems Tape and the date the tape was generated.

Note No. 2: The Systems Loader is not in a file, but is merely one block that sits as the second block on the tape. By sensing SFF1, the System Loader determines whether to read in the Control Program or the Systems Tape Generator Program.

#### 4.5.1.2 Expansion

The Systems Tape is flexible in that it can be either expanded or contracted; however, it is essentially a complete system.

#### 4.5.2 Header and Trailer Blocks

There is a header block and a trailer block in each file on the Systems Tape. These blocks are created by the Systems Tape Generator Program and are used by the Control Program during compilation time. They serve as the control mechanisms for reading in the various runs.

##### 4.5.2.1 Format of the Header and Trailer Blocks

The header and trailer blocks in any one file are identical to one another. They consist of 10 words each; however, only the first 3 words are used currently. Word 1 contains the program identification, e.g., for Run 1.0, word 1 would contain 226160230505 which is equivalent to M10N, the run identification. Terminating blocks are spaced filled. Word 2 contains the file number and the number of blocks in the file, e.g., if word 2 contained 000001600005<sub>8</sub> it would mean it was file No. 5 and it contained 16 octal blocks. Word 3 contains the starting address of the program in the alpha portion. As was stated previously, words 4-10 are not used currently.

#### 4.5.3 Putting Programs on the Systems Tape

Means by which a program or a new version of a program is put on the Systems Tape:

1. The Control Card must be put in as the first card in the binary deck. The Control Card has to be in the following format: it must be ISN; 12-row left will have a word count of 1 and an address of 37777<sub>8</sub>. (Note: there is no relationship of the 37777<sub>8</sub> on the Control Card and the fact that memory is limited to 17777<sub>8</sub>.); 12-row right may or may not contain a hashsum, but it may not, however, ignore the hashsum punch. 11-row left must have the program identification, e.g., for Run 1.3 11-row left must contain 226163230505 representing M13N, the identification for Run 1.3.

2. The normal transfer card must be replaced with a transfer to 1. However, there is one exception, if the program is the last program in any series of programs being put on the Systems Tape or is the only program being added to the tape, it must have a transfer to 2 instead of 1.

#### 4.5.3.1 Program Starting Locations

The lowest location of any program on the Systems Tape must contain a transfer to the starting location of the program.

## SECTION V

### CONCLUSIONS

#### 5.1 CONCLUSIONS

Most of the coding necessary to modify the runs specified in the First Quarterly Report is complete. The next logical step is to test each run to insure its functioning as a unit and also as an integral part of a system, namely, MOBILE. The unit testing has begun to check out the relation of one run to another in the overall system.

## SECTION VI

### PROGRAM FOR THE NEXT PERIOD

During the next period each run of MOBIDIC MOBILE I will undergo rigorous test procedures until specifications are met. When this occurs, all the runs will be integrated to form the MOBILE system which in turn, will undergo a series of equally rigorous tests. Once MOBILE I functions effectively as a system, it will be subjected to Acceptance Testing to ensure complete satisfaction of the purchaser's contract specifications. These tests will be a series of COBOL source programs from which MOBILE I will produce object code and listings in accordance with the rules of COBOL and MOBILE I. These object programs should then function in the manner prescribed by the user if the COBOL statements used in the Source Program are free from logical and COBOL errors.

**SECTION VII**  
**IDENTIFICATION OF KEY PERSONNEL**

**7.1 KEY TECHNICAL PERSONNEL**

|                   |                                                  |
|-------------------|--------------------------------------------------|
| Alvin H. Hatch    | Manager, Applied Programming Dept.               |
| Arthur S. Morse   | Section Head, Language Implementation<br>Section |
| Herbert S. Hughes | Research Engineer                                |
| Roy Sundgren      | Senior Engineer                                  |
| Richard Mackler   | Senior Engineer                                  |

**7.2 APPROXIMATE MAN-HOURS EXPENDED**

| <u>Name</u>       | <u>Hours</u> |
|-------------------|--------------|
| Alvin H. Hatch    | 132          |
| Arthur S. Morse   | 160          |
| Herbert S. Hughes | 412          |
| Roy Sundgren      | 503          |
| Richard Mackler   | <u>352</u>   |
| Total             | 1559         |

APPENDIX A  
RUN 1.0 PROGRAM LISTING

AUGUST 1962

|       |                         |           |
|-------|-------------------------|-----------|
| REM   | PACKAGE IV              | MOBILE, I |
| LST   | A, I, M                 |           |
| MACRO | 1/0,06/65,3/1,12/M,15/A |           |
| VFD   |                         |           |
| END   |                         |           |
| MACRO | A, I, M                 |           |
| VFD   | 1/0,06/64,3/1,12/M,15/A |           |
| END   |                         |           |
| MACRO | A, I, M                 |           |
| VFD   | 1/0,06/63,3/1,12/M,15/A |           |
| END   |                         |           |
| MACRO | A, I, M                 |           |
| LOD   | =A,0,I                  |           |
| LOD   | =M,0,I+1                |           |
| END   |                         |           |
| WIPE  | 1                       |           |
| RST   |                         |           |
| REM   | /37777                  |           |
| ORG   | 1,MION                  |           |
| BCI   | TEIA-1                  |           |
| ORG   | ZZ                      |           |
| TRU   | 1400110                 |           |
| OCT   | 1/0,9/0,12/11,15/2      |           |
| VFD   | 5,ACCUMULATOR           |           |
| BCZ   | 1/0,9/0,12/10,15/4      |           |
| VFD   | 5,CARD-PUNCH            |           |
| BCZ   | 1/0,9/0,12/14,15/6      |           |
| VFD   | 5,CARD-PUNCH-15N        |           |
| BCZ   | 1/0,9/0,12/11,15/CDR    |           |
| VFD   | 5,CARD-READER           |           |
| BCZ   | 1/0,9/0,12/15,15/5      |           |
| VFD   | 5,CARD-READER-15N       |           |
| BCZ   | 1/0,9/0,12/11,15/FLX    |           |
| VFD   | 5,FLEXOWRITER           |           |
| BCZ   | 1/0,9/0,12/9,15/1       |           |

R110  
R110  
R110  
R100  
R100  
R100  
R100  
R110  
R100  
R100







|           |     |                 |  |                     |      |
|-----------|-----|-----------------|--|---------------------|------|
| 016200    | RPA | PCT             |  |                     | R100 |
| 016300R2  | CLA | M,3             |  |                     | R100 |
| 016400    | TRX | *+1,3,1         |  | ADJUST FOR NO ENTRY | R100 |
| 016500    | RPA | PCT             |  |                     | R100 |
| 016600R3  | RPA | R4              |  | CONTINUE TESTING    | R100 |
| 016700    | ADB | ACC,0,1         |  |                     | R100 |
| 016800    | RPA | R5              |  | FOR NO MATCH        | R100 |
| 016900    | CLA | TB00A+1         |  |                     | R100 |
| 017000R4  | TRC | **              |  |                     | R100 |
| 017100    | TRU | R1              |  | NO MATCH            | R100 |
| 017200    | TRU | R1              |  | NO MATCH            | R100 |
| 017300R5  | CLA | **              |  | MATCH               | R100 |
| 017400    | RPA | PCT             |  |                     | R100 |
| 017500M1  | TRX | R6,3,3          |  |                     | R100 |
| 017600M2  | TRX | R6,3,6          |  |                     | R100 |
| 017700M3  | TRX | R6,3,9          |  |                     | R100 |
| 017800M1A | TRX | R7,3,2          |  |                     | R100 |
| 017900M2A | TRX | R7,3,5          |  |                     | R100 |
| 018000M3A | TRX | R7,3,8          |  |                     | R100 |
| 018100M   | HLT | M2              |  |                     | R100 |
| 018200    | BCZ | 1,CARD          |  |                     | R100 |
| 018300    | HLT | MIC1            |  |                     | R100 |
| 018400    | HLT | M1A             |  |                     | R100 |
| 018500    | BCZ | 1,TAPE          |  |                     | R100 |
| 018600    | HLT | MIE1            |  | ERROR               | R100 |
| 018700    | HLT | MR              |  |                     | R100 |
| 018800    | HLT | M1              |  |                     | R100 |
| 018900    | BCZ | 1,CARD          |  | M1A                 | R100 |
| 019000    | HLT | MCC1            |  |                     | R100 |
| 019100    | HLT | MR              |  | ERROR               | R100 |
| 019200    | HLT | M3              |  |                     | R100 |
| 019300    | BCI | 1,CARD          |  |                     | R100 |
| 019400    | HLT | MOC1            |  |                     | R100 |
| 019500    | HLT | M1A             |  |                     | R100 |
| 019600    | BCZ | 1,MAG-TA        |  |                     | R100 |
| 019700    | VFD | 1/1,21/0,15/MC1 |  |                     | R100 |

|            |      |                          |                          |  |      |
|------------|------|--------------------------|--------------------------|--|------|
| 021800     | HLT  | MOT1                     |                          |  | R100 |
| 021900MC2  | VFD  | 1/0,F6/T,018/050505,12/0 |                          |  | R100 |
| 022100MC3  | HLT  | MR                       |                          |  | R100 |
|            | BCZ  | 1,MBLY                   |                          |  |      |
|            | HLT  | MR                       |                          |  |      |
| 022300MC4  | BCZ  | 1,RT                     |                          |  | R100 |
| 022400     | HLT  | M1                       |                          |  | R100 |
| 022500MIC1 | CLA  | ZERO                     | INPUT ON CARDS           |  | R100 |
| 022600     | LDQ  | =/14                     |                          |  | R100 |
| 022700     | MSK  | TCONA+1                  |                          |  | R100 |
| 022800     | TRU  | M2                       |                          |  | R100 |
| 022900MIE1 | CLA  | =/10                     | GET NEXT WORD            |  | R100 |
| 023000     | LDQ  | =/14                     | INPUT ON TAPE            |  | R100 |
| 023100     | MSK  | TCONA+1                  |                          |  | R100 |
| 023200     | TRU  | M1                       |                          |  | R100 |
| 023300MR   | TRL  | ERR,1,FG1                |                          |  | R100 |
| 023400     | TRU  | M1                       |                          |  | R100 |
|            | SNS  | *+1,0,00FOC              | SET FLIX FOR CORRECTIWS  |  |      |
| 023900     | TRU  | M1                       |                          |  | R100 |
| 024000MOC1 | CLA  | ZERO                     | OUTPUT ON CARDS          |  | R100 |
| 024100     | LDQ  | =/3                      |                          |  | R100 |
| 024200     | MSK  | TCONA+1                  |                          |  | R100 |
| 024300     | TRU  | M3                       |                          |  | R100 |
| 024400MOT1 | CLA  | =/1                      | OUTPUT ON TAPE           |  | R100 |
| 024500     | LDQ  | =/3                      |                          |  | R100 |
| 024600     | MSK  | TCONA+2                  |                          |  | R100 |
| 024700     | TRU  | M2                       |                          |  | R100 |
| 024800MOB1 | CLA  | =/2                      | OUTPUT ON TAPE AND CARDS |  | R100 |
| 024900     | LDQ  | =/3                      |                          |  | R100 |
| 025000     | MSK  | TCONA+2                  |                          |  | R100 |
| 025100     | TRU  | M1                       |                          |  | R100 |
| 025200MCL1 | CLA  | =/2000000000             | COMPATIBLE LISTING       |  | R100 |
| 025300     | LDQ  | =/2000000000             |                          |  | R100 |
| 025400     | MSK  | TCONA+2                  |                          |  | R100 |
|            | TRU  | M1                       |                          |  |      |
|            | CLA  | =/400000                 | NO BSP LISTING           |  |      |
|            | BSP1 |                          |                          |  |      |



|          |     |             |                |      |
|----------|-----|-------------|----------------|------|
| 027700   | LDQ | LA00A+1     | GET THIRD WORD | R100 |
| 027800   | SLL | 24,0,1      |                | R100 |
| 027900   | LDQ | LA00A+2     |                | R100 |
| 028000   | SLL | 12,0,1      |                | R100 |
| 028100   | STR | TB00A       |                | R100 |
| 028200   | CLA | ZERO        |                | R100 |
| 028300   | SLL | 12,0,1      |                | R100 |
| 028400   | SHL | 24,0,1      |                | R100 |
| 028500   | STR | TB00A+1     |                | R100 |
| 028600   | MOV | SA4,S1      |                | R100 |
| 028700   | MOV | ORG,LA00A+2 |                | R100 |
| 028800   | TRS |             | RETURN         | R100 |
| 028900S4 | LDQ | LA00A+2     |                | R100 |
| 029000   | CLA | ZERO        |                | R100 |
| 029100   | SLL | 12,0,1      |                | R100 |
| 029200   | LDQ | LA00A+3     |                | R100 |
| 029300   | SLL | 24,0,1      |                | R100 |
| 029400   | STR | TB00A       |                | R100 |
| 029500   | CLA | ZERO        |                | R100 |
| 029600   | SLL | 12,0,1      |                | R100 |
| 029700   | LDQ | LA00A+4     |                | R100 |
| 029800   | SLL | 12,0,1      |                | R100 |
| 029900   | SHL | 12,0,1      |                | R100 |
| 030000   | STR | TB00A+1     |                | R100 |
| 030100   | MOV | ORG,LA00A+4 |                | R100 |
| 030200   | MOV | SA5,S1      |                | R100 |
| 030300   | TRS |             | RETURN         | R100 |
| 030400S5 | LDQ | LA00A+4     | GET FIFTH WORD | R100 |
| 030500   | CLA | ZERO        |                | R100 |
| 030600   | SLL | 24,0,1      |                | R100 |
| 030700   | LDQ | LA00A+5     |                | R100 |
| 030800   | SLL | 12,0,1      |                | R100 |
| 030900   | STR | TB00A       |                | R100 |
| 031000   | CLA | ZERO        |                | R100 |
| 031100   | SLL | 12,0,1      |                | R100 |
| 031200   | SHL | 24,0,1      |                | R100 |

R100  
R100  
R100  
R100  
R100  
R100  
R100  
R100  
R100

RETURN  
GET SIXTH WORD

R100

STR TB00A+1  
MOV ORG+LA00A+5  
MOV SA6+51  
TRS ZERO  
CLA LA00A+5  
LDO 12+0+1  
SLL LA00A+6  
LDO 6+0+1  
SLL 18+0+1  
SHL TB00A  
STR ORG+LA00A+6  
MOV SA7+51  
TRS ZERO  
CLA LA00A+6  
LDO 12+0+1  
SLL 24+0+1  
SHL TB00A  
STR ORG+LA00A+6  
MOV SA8+51  
TRS ZERO  
CLA LA00A+5  
LDO 12+0+1  
SLL 24+0+1  
SHL TB00A  
STR ORG+LA00A+6  
MOV SA9+51  
TRS ZERO  
CLA LA00A+5  
LDO 6+0+1  
SLL LA00A+7  
LDO 6+0+1  
SLL 24+0+1  
SHL

RETURN

A-10

031300  
031400  
031500  
031600  
031700S6  
031800  
031900  
032000  
032100  
032200  
032300  
S7

S8

S9

|     |             |  |
|-----|-------------|--|
| STR | TB00A       |  |
| MOV | ORG,LA00A+7 |  |
| MOV | SA10,S1     |  |
| TRS |             |  |
| CLA | ZERO        |  |
| LDQ | LA00A+7     |  |
| SLL | 12,0,1      |  |
| SHL | 24,0,1      |  |
| STR | TB00A       |  |
| MOV | ORG,LA00A+7 |  |
| TRS |             |  |
| TRU | S2          |  |
| TRU | S3          |  |
| TRU | S4          |  |
| TRU | S5          |  |
| TRU | S6          |  |
| TRU | S7          |  |
| TRU | S8          |  |
| TRU | S9          |  |
| TRU | S10         |  |
| CLA | TCONA+1     |  |
| LGM | =/14        |  |
| TRZ | *+2         |  |
| TRU | N5          |  |
| SEN | *+2,0,00F0C |  |
| TRU | N5          |  |
| SMS | *+1,0,09F0C |  |
| TRL | A           |  |
| CLA | TB00A       |  |
| TRC | A10E        |  |
| TRU | N7          |  |
| TRU | N7          |  |
| TRL | A           |  |
| CLA | TB00A       |  |
| TRC | N6          |  |
| TRU | N7          |  |

R100  
R100  
R100  
R100  
R100

R100  
R100  
R100  
R100  
R100

R100  
R100  
R100  
R100  
R100  
R100  
R100  
R100  
R100

IS INPUT ON CARDS

INPUT IS NOT ON CARDS  
ARE THERE ANY CORRECTIONS  
NO CORRECTIONS

HAVE WE REACHED  
END CORRECTIONS  
X  
NO  
NO  
TEST NEXT WORD  
X  
X

|          |     |                  |   |                      |  |      |
|----------|-----|------------------|---|----------------------|--|------|
| 035200   | TRU | N7               |   |                      |  | R100 |
| 035300   | CLA | TB00A+1          |   |                      |  | R100 |
| 035400   | TRC | N6+1             |   |                      |  | R100 |
| 035500   | TRU | *+1              |   | ERROR                |  | R100 |
|          | TRL | ERR+1,FGL        |   | ERROR                |  | R100 |
| 035700   | SNS | *+1,0,WEF        |   |                      |  | R100 |
| 035800   | TRL | WR+0,SPO         |   | WRITE LAST BLOCK     |  | R100 |
|          | HLT | LL00A            |   |                      |  | R100 |
| 036000   | HLT |                  | X |                      |  | R100 |
|          | CLA | =/1000000000     |   |                      |  |      |
| 023600   | LDQ | =/1000000000     |   |                      |  | R100 |
| 023700   | MSK | TC0NA+2          |   |                      |  | R100 |
|          | MVZ | LA00C            |   | RESET SEG NUMBERS    |  | R100 |
| 036500N7 | SNR | SCAN+0,09FOC     |   | RESET SPECIAL MODE   |  | R100 |
| 036600   | MVZ | SAV3             |   | RESET FOR NEXT       |  | R100 |
| 036700   | MVZ | SAV4             |   | CARD                 |  | R100 |
| 036800   | MVZ | LA00T            |   |                      |  | R100 |
|          | TRL | WR+0,SPO         |   | WRITE OUT CORRECTION |  | R100 |
|          | HLT | LL00A            |   |                      |  | R100 |
| 037000   | HLT |                  | X |                      |  | R100 |
| 037100   | TRU | N8               |   | RETURN               |  | R100 |
| 037200N6 | BCZ | 2,CORRECTIONS    |   |                      |  | R100 |
| 037400ID | BCZ | 1,IDENTI         |   | IDENTIFICATION       |  | R100 |
| 037500   | VFD | 1/1,21/0,15/IDC8 |   |                      |  | R100 |
| 037600   | HLT | IDA              |   |                      |  | R100 |
| 037700   | BCZ | 1,DIVISI         |   | DIVISION             |  | R100 |
| 037800   | VFD | 1/1,21/0,15/IDC9 |   |                      |  | R100 |
| 037900   | HLT | IDA              |   |                      |  | R100 |
| 038000   | BCZ | 1,PROGRA         |   | PROGRAM-ID           |  | R100 |
| 038100   | VFD | 1/1,21/0,15/IDC1 |   |                      |  | R100 |
| 038200   | HLT | IDA              |   |                      |  | R100 |
| 038300   | BCZ | 1,AUTHOR         |   | AUTHOR               |  | R100 |
| 038400   | HLT | IDD              |   |                      |  | R100 |
| 038500   | HLT | H1A              |   |                      |  | R100 |
| 038600   | BCZ | 1,DATE-W         |   | DATE-WRITTEN         |  | R100 |
| 038700   | VFD | 1/1,21/0,15/IDC2 |   |                      |  | R100 |

|            |     |                  |                   |      |
|------------|-----|------------------|-------------------|------|
| 038800     | HLT | H1A              | DATE-COMPILED     | R100 |
| 038900     | BCZ | 1,DATE-C         |                   | R100 |
| 039000     | VFD | 1/1,21/0,15/IDC3 |                   | R100 |
| 039100     | HLT | H1A              | SECURITY          | R100 |
| 039200     | BCZ | 1,SECURI         |                   | R100 |
| 039300     | VFD | 1/1,21/0,15/IDC4 |                   | R100 |
| 039400     | HLT | H1A              | REMARKS           | R100 |
| 039500     | BCZ | 1,REMARK         |                   | R100 |
| 039600     | VFD | 1/1,21/0,15/IDC5 |                   | R100 |
| 039700     | HLT | H1A              | INSTALLATION      | R100 |
| 039800     | BCZ | 1,INSTAL         |                   | R100 |
| 039900     | VFD | 1/1,21/0,15/IDC6 |                   | R100 |
| 040000     | HLT | H1A              | ENVIRONMENT       | R100 |
| 040100ID1  | BCZ | 1,ENVIRO         |                   | R100 |
| 040200     | VFD | 1/1,21/0,15/IDC7 |                   | R100 |
| 040300     | HLT | IDE              |                   | R100 |
| 040400IDC1 | BCZ | 1,M-ID           |                   | R100 |
| 040500     | HLT | IDP              |                   | R100 |
| 040600IDC2 | BCZ | 1,RITTEN         |                   | R100 |
| 040700     | HLT | IDD              |                   | R100 |
| 040800IDC3 | BCZ | 1,OMPILE         |                   | R100 |
| 040900     | HLT | IDM              |                   | R100 |
| 041000IDC4 | BCZ | 1,TY             |                   | R100 |
| 041100     | HLT | IDD              |                   | R100 |
| 041200IDC5 | BCZ | 1,S              |                   | R100 |
| 041300     | HLT | IDD              |                   | R100 |
| 041400IDC6 | BCZ | 1,LATION         |                   | R100 |
| 041500     | HLT | IDD              |                   | R100 |
| 041600IDC7 | BCZ | 1,NMENT          |                   | R100 |
| 041700     | HLT | IDN              |                   | R100 |
| 041800IDC8 | BCZ | 1,FICATI         |                   | R100 |
| 041900     | HLT | IDA1             |                   | R100 |
| 042000IDC9 | BCZ | 1,ON             |                   | R100 |
| 042100     | HLT | IDA2             |                   | R100 |
| 042200IDA1 | CLA | TB00A+2          | FINISH WORD CHECK | R100 |
| 042300     | TRC | A7D              |                   | R100 |

|            |     |                |      |
|------------|-----|----------------|------|
| 042400     | TRU | ++1            | R100 |
| 042500     | TRL | ERR,1,FG1      | R100 |
| 042600     | TRU | H1             | R100 |
| 042700IDA2 | SEN | H1,0,00F0C     | R100 |
| 042800     | TRL | ERR,1,FG11     | R100 |
| 042820     | TRU | H1             | R100 |
|            | CLA | TB00C          | R100 |
|            | RPA | IDP2           | R100 |
|            | TRU | IDP5           | R100 |
| 043000IDP  | SEN | ++2,0,00F0C    | R100 |
| 043100     | TRL | ERR,1,FG11     | R100 |
| 043200     | TRL | A              | R100 |
| 043300     | TRL | J              | R100 |
| 043400     | TRL | ERR,1,FG1      | R100 |
|            | MVZ | IR1            | R100 |
|            | CLA | TB00C          | R100 |
|            | SUB | =1,0,1         | R100 |
|            | TRZ | IDP3           | R100 |
|            | CLA | ZERO           | R100 |
|            | LDQ | TB00C          | R100 |
|            | DVL | =6,0,1         | R100 |
|            | MOV | TB00W,IR2      | R100 |
|            | RPA | IDP2           | R100 |
|            | TRX | ++1,1,0        | R100 |
| IDP4       | CLA | TB00A,1        | R100 |
|            | STR | T11A+1,1       | R100 |
|            | TRX | IDP4,1,1       | R100 |
| IDP5       | CLA | ZERO           | R100 |
|            | LDQ | =/050505050505 | R100 |
| IDP2       | SLL | ++,0,1         | R100 |
|            | LGA | TB00A+1,1      | R100 |
|            | STR | T11A+2,1       | R100 |
| 043500     | SEN | IDP1,0,00F0C   | R100 |
| 043600     | TRL | ERR,1,FG11     | R100 |
| 043700IDP1 | MOV | =9,IR3         | R100 |
| 043800     | MOV | =/7777,IR4     | R100 |

RESET INDEX REGISTERS

|        |     |                |                      |      |
|--------|-----|----------------|----------------------|------|
| 043900 | MVZ | CNT1           | RETURN TO SCAN       | R100 |
| 044000 | TRU | SCN1           |                      | R100 |
| 044200 | SEN | *+2,0,00F0C    | IS THERE A PERIOD    | R100 |
| 044300 | TRL | ERR,1,FG11     |                      | R100 |
| 044400 | TRL | A              | GET NEXT WORD        | R100 |
| 044500 | MOV | =1,CNT1        |                      | R100 |
| 044600 | TRL | CK             |                      | R100 |
| 044700 | TRL | ERR,1,FG10     |                      | R100 |
| 044800 | SEN | IDP1,0,00F0C   | IS THERE A PERIOD    | R100 |
| 044900 | TRU | IDD1           |                      | R100 |
| 045000 | BCZ | 1,D            |                      | R100 |
| 045100 | CLA | TB00A+2        |                      | R100 |
| 045200 | TRC | IDM2           |                      | R100 |
| 045300 | TRU | *+1            |                      | R100 |
| 045400 | TRL | ERR,1,FG1      |                      | R100 |
|        | TRU | DATE1          | TEST FOR DATE        |      |
| 046100 | MOV | =2,3C          | END OF RUN           | R100 |
| 046600 | MOV | ZYX,PCT        | EXIT FROM RUN        | R100 |
|        | HLT |                | STORAGE FOR EXIT     |      |
|        | SEN | *+2,0,00F0C    | IS THERE A PERIOD    |      |
|        | TRL | ERR,1,FG11     |                      |      |
|        | ADB | ZERO,2,2       |                      |      |
|        | CLA | WSR            | IS DATE GIVEN        |      |
|        | TRZ | IDD            | NO DATE              |      |
|        | SRL | 20             | SET IN PRINTING FORM |      |
|        | LGA | =/050505050560 |                      |      |
|        | SHL | 2,0,7          |                      |      |
|        | SLL | 4,0,7          |                      |      |
|        | LGA | =/60           |                      |      |
|        | STR | BSPA+4         | ALL OF MONTH         |      |
|        | SHL | 8,0,7          |                      |      |
|        | SLL | 4,0,7          |                      |      |
|        | LGA | =/0560         |                      |      |
|        | TRX | DATE2,1,0      |                      |      |
|        | SHL | 2,0,7          |                      |      |
|        | SLL | 4,0,7          |                      |      |



|             |     |                  |   |                      |      |
|-------------|-----|------------------|---|----------------------|------|
| 048100      | MVZ | IR3              | X | PICK UP 6 CHARACTERS | R100 |
| 048200      | CLA | TB00A            |   | SEARCH TABLE         | R100 |
| 048300J3    | TRC | ** , 3           |   |                      | R100 |
| 048400      | TRU | J6               |   |                      | R100 |
| 048500      | TRU | J6               |   |                      | R100 |
| 048600      | TRU | J4               |   | FINISH MATCH         | R100 |
| 048700J6    | TRX | J3 , 3 , 2       |   | HAVE WE FINISHED     | R100 |
| 048800J8    | ADB | PCS , 0 , 1      |   | RETURN NO MATCH      | R100 |
| 048900      | TRU | J7               |   |                      | R100 |
| 049000      | ADB | ACC , 0 , 1      |   |                      | R100 |
| 049100      | RPA | J4               |   |                      | R100 |
| 049200J4    | CLA | ** , 3           |   |                      | R100 |
| 049300      | TRZ | J7               |   |                      | R100 |
| 049400      | RPA | J5               |   |                      | R100 |
| 049500      | SHR | 15               |   |                      | R100 |
| 049600      | STR | IR2              |   |                      | R100 |
| 049700      | MVZ | IR1              |   |                      | R100 |
| 049800J5    | TRC | ** , 1           |   |                      | R100 |
| 049900      | TRU | J6               |   | SEARCH REST OF TABLE | R100 |
| 050000      | TRU | J6               |   | SEARCH REST OF TABLE | R100 |
| 050100      | TRX | J5 , 1 , 1       |   | LOAD IRS             | R100 |
| 050200J7    | MOV | J10 , IR3        | X | RETURN               | R100 |
| 050300      | MOV | J11 , IR4        |   | ILLEGAL CHARACTER    | R100 |
| 050400      | TRS | J8               |   |                      | R100 |
| 050500J9    | TRU |                  |   |                      | R100 |
| 050600J10   | HLT |                  |   |                      | R100 |
| 050700J11   | HLT |                  |   |                      | R100 |
| 050800LG00A | HLT |                  |   |                      | R100 |
| 050900      | HLT |                  |   |                      | R100 |
| 051000      | HLT |                  |   |                      | R100 |
| 051100      | HLT |                  |   |                      | R100 |
| 051200      | HLT |                  |   |                      | R100 |
| 051300      | HLT |                  |   |                      | R100 |
| 051400      | HLT |                  |   |                      | R100 |
| 051500      | HLT |                  |   |                      | R100 |
| 051600      | HLT |                  |   |                      | R100 |
|             |     | A1A , 0 , XA / 2 |   |                      |      |
|             |     | A1B , 0 , XB / 2 |   |                      |      |
|             |     | 1CA , 0 , XC / 2 |   |                      |      |
|             |     | 00               |   |                      |      |
|             |     | 01               |   |                      |      |
|             |     | 02               |   |                      |      |
|             |     | 03               |   |                      |      |
|             |     | 04               |   |                      |      |
|             |     | 05               |   |                      |      |
|             |     | 06               |   |                      |      |
|             |     | 07               |   |                      |      |
|             |     | 10               |   |                      |      |

|        |     |            |    |      |
|--------|-----|------------|----|------|
| 051700 | HLT | A1D,0,XD/2 | 11 | R100 |
| 051800 | HLT | A1E,0,XE/2 | 12 | R100 |
| 051900 | HLT | A1F,0,XF/2 | 13 | R100 |
| 052000 | HLT | A1G,0,XG/2 | 14 | R100 |
| 052100 | HLT | A1H,0,XH/2 | 15 | R100 |
| 052200 | HLT | A1I,0,XI/2 | 16 | R100 |
| 052300 | HLT | A1J,0,XJ/2 | 17 | R100 |
| 052400 | OCT | -0         | 20 | R100 |
| 052500 | HLT | A1L,0,XL/2 | 21 | R100 |
| 052600 | HLT | A1M,0,XM/2 | 22 | R100 |
| 052700 | HLT | A1N,0,XN/2 | 23 | R100 |
| 052800 | HLT | A1O,0,XO/2 | 24 | R100 |
| 052900 | HLT | A1P,0,XP/2 | 25 | R100 |
| 053000 | HLT | A1Q,0,XQ/2 | 26 | R100 |
| 053100 | HLT | A1R,0,XR/2 | 27 | R100 |
| 053200 | HLT | A1S,0,XS/2 | 30 | R100 |
| 053300 | HLT | A1T,0,XT/2 | 31 | R100 |
| 053400 | HLT | A1U,0,XU/2 | 32 | R100 |
| 053500 | HLT | A1V,0,XV/2 | 33 | R100 |
| 053600 | HLT | A1W,0,XW/2 | 34 | R100 |
| 053700 | OCT | -0         | 35 | R100 |
| 053800 | OCT | -0         | 36 | R100 |
| 053900 | HLT | A1Z,0,XZ/2 | 37 | R100 |
| 054000 | OCT | 0          | 40 | R100 |
| 054100 | OCT | -0         | 41 | R100 |
| 054200 | OCT | 0          | 42 | R100 |
| 054300 | OCT | 0          | 43 | R100 |
| 054400 | OCT | 0          | 44 | R100 |
| 054500 | OCT | 0          | 45 | R100 |
| 054600 | OCT | 0          | 46 | R100 |
| 054700 | OCT | 0          | 47 | R100 |
| 054800 | OCT | 0          | 50 | R100 |
| 054900 | OCT | 0          | 51 | R100 |
| 055000 | OCT | 0          | 52 | R100 |
| 055100 | OCT | 0          | 53 | R100 |
| 055200 | OCT | 0          | 54 | R100 |

|           |     |          |    |      |
|-----------|-----|----------|----|------|
| 055300    | OCT | 0        | 55 | R100 |
| 055400    | OCT | 0        | 56 | R100 |
| 055500    | OCT | 0        | 57 | R100 |
| 055600    | OCT | -3       | 60 | R100 |
| 055700    | OCT | -0       | 61 | R100 |
| 055800    | OCT | -0       | 62 | R100 |
| 055900    | OCT | -0       | 63 | R100 |
| 056000    | OCT | -0       | 64 | R100 |
| 056100    | OCT | -0       | 65 | R100 |
| 056200    | OCT | -0       | 66 | R100 |
| 056300    | OCT | -0       | 67 | R100 |
| 056400    | OCT | -0       | 70 | R100 |
| 056500    | OCT | -0       | 71 | R100 |
| 056600    | OCT | -0       | 72 | R100 |
| 056700    | OCT | 0        | 73 | R100 |
| 056800    | OCT | 0        | 74 | R100 |
| 056900    | OCT | 0        | 75 | R100 |
| 057000    | OCT | 0        | 76 | R100 |
| 057100    | OCT | 0        | 77 | R100 |
| 057200A1A | BCZ | 1,ABOUT  |    | R100 |
| 057300    | HLT | 1,ACCEPT |    | R100 |
| 057400    | BCI |          |    | R100 |
| 057500    | HLT |          |    | R100 |
| 057600    | BCZ | 1,ACCUMU |    | R100 |
| 057700    | HLT | A7A,0,1  |    | R100 |
| 057800    | BCZ | 1,ADD    |    | R100 |
| 057900    | HLT |          |    | R100 |
| 058000    | BCZ | 1,ADDRES |    | R100 |
| 058100    | HLT | A2A,0,1  |    | R100 |
| 058200    | BCZ | 1,ADVANC |    | R100 |
| 058300    | HLT | A3A,0,1  |    | R100 |
| 058400    | BCZ | 1,AFTER  |    | R100 |
| 058500    | HLT |          |    | R100 |
| 058600    | BCZ | 1,ALL    |    | R100 |
| 058700    | HLT |          |    | R100 |
| 058800    | BCZ | 1,ALPHAB |    | R100 |

|           |     |          |      |
|-----------|-----|----------|------|
| 058900    | HLT | A4A,0,1  | R100 |
| 059000    | BCZ | 1,ALPHAN | R100 |
| 059100    | HLT | A5A,0,1  | R100 |
| 059200    | BCZ | 1,ALTER  | R100 |
| 059300    | HLT |          | R100 |
| 059400    | BCZ | 1,ALTERN | R100 |
| 059500    | HLT | A6A,0,1  | R100 |
| 059600    | BCZ | 1,AN     | R100 |
| 059700    | HLT |          | R100 |
| 059800    | BCZ | 1,AND    | R100 |
| 059900    | HLT |          | R100 |
| 060000    | BCZ | 1,APPLY  | R100 |
| 060100    | HLT |          | R100 |
| 060200    | BCZ | 1,ARE    | R100 |
| 060300    | HLT |          | R100 |
| 060400    | BCZ | 1,AREA   | R100 |
| 060500    | HLT |          | R100 |
| 060600    | BCZ | 1,AREAS  | R100 |
| 060700    | HLT |          | R100 |
| 060800    | BCZ | 1,AS     | R100 |
| 060900    | HLT |          | R100 |
| 061000    | BCZ | 1,ASSIGN | R100 |
| 061100    | HLT |          | R100 |
| 061200    | BCZ | 1,AT     | R100 |
| 061300    | HLT |          | R100 |
| 061400A1B | BCZ | 1,BEFORE | R100 |
| 061500    | HLT |          | R100 |
| 061600    | BCZ | 1,BEGINN | R100 |
| 061700    | HLT | A2B,0,1  | R100 |
| 061800    | BCZ | 1,BEGINN | R100 |
| 061900    | HLT | A3B,0,3  | R100 |
| 062000    | BCZ | 1,BEGINN | R100 |
| 062100    | HLT | A4B,0,3  | R100 |
| 062200    | BCZ | 1,BIT    | R100 |
| 062300    | HLT |          | R100 |
| 062400    | BCZ | 1,BITS   | R100 |

|        |     |          |      |
|--------|-----|----------|------|
| 062500 | HLT | 1,BLANK  | R100 |
| 062600 | BCZ |          | R100 |
| 062700 | HLT | 1,BLOCK  | R100 |
| 062800 | BCZ |          | R100 |
| 062900 | HLT | 1,BLOCK- | R100 |
| 063000 | BCZ | A5B,0,1  | R100 |
| 063100 | HLT | 1,BY     | R100 |
| 063200 | BCZ |          | R100 |
| 063300 | HLT | 1,CARD-P | R100 |
| 063400 | BCZ | A14C,0,1 | R100 |
| 063500 | HLT | 1,CARD-R | R100 |
| 063600 | BCZ | A13C,0,1 | R100 |
| 063700 | HLT | 1,CHARAC | R100 |
| 063800 | BCZ | 2CA,0,1  | R100 |
| 063900 | HLT | 1,CHARAC | R100 |
| 064000 | BCZ | 3CA,0,1  | R100 |
| 064100 | HLT | 1,CHECK  | R100 |
| 064200 | BCZ |          | R100 |
| 064300 | HLT | 1,CLASS  | R100 |
| 064400 | BCZ |          | R100 |
| 064500 | HLT | 1,CLOCK- | R100 |
| 064600 | BCZ | A4C,0,1  | R100 |
| 064700 | HLT | 1,CLOSE  | R100 |
| 064800 | BCZ |          | R100 |
| 064900 | HLT | 1,COBOL  | R100 |
| 065000 | BCZ |          | R100 |
| 065100 | HLT | 1,COMP-1 | R100 |
| 065200 | BCZ |          | R100 |
| 065300 | HLT | 1,COMP-2 | R100 |
| 065400 | BCZ |          | R100 |
| 065500 | HLT | 1,COMPUT | R100 |
| 065600 | BCZ | A5C,0,2  | R100 |
| 065700 | HLT | 1,COMPUT | R100 |
| 065800 | BCZ | A6C,0,2  | R100 |
| 065900 | HLT | 1,COMPUT | R100 |
| 066000 | BCZ |          | R100 |

|        |     |          |      |
|--------|-----|----------|------|
| 066100 | HLT | A7C,0,1  | R100 |
| 066200 | BCZ | 1,CONSTA | R100 |
| 066300 | HLT | A8C,0,1  | R100 |
| 066400 | BCZ | 1,CONFIG | R100 |
| 066500 | HLT | A9C,0,2  | R100 |
| 066600 | BCZ | 1,CONTAI | R100 |
| 066700 | HLT | A10C,0,1 | R100 |
| 066800 | BCZ | 1,CONTRO | R100 |
| 066900 | HLT | A11C,0,1 | R100 |
| 067000 | BCZ | 1,COPY   | R100 |
| 067100 | HLT |          | R100 |
| 067200 | BCZ | 1,CORRES | R100 |
| 067300 | HLT | A12C,0,2 | R100 |
| 067400 | BCZ | 1,DATA   | R100 |
| 067500 | HLT |          | R100 |
| 067600 | BCZ | 1,DATE-W | R100 |
| 067700 | HLT | A2D,0,1  | R100 |
| 067800 | BCZ | 1,DECLAR | R100 |
| 067900 | HLT | A3D,0,1  | R100 |
| 068000 | BCZ | 1,DEFINE | R100 |
| 068100 | HLT |          | R100 |
| 068200 | BCZ | 1,DEPEND | R100 |
| 068300 | HLT | A4D,0,1  | R100 |
| 068400 | BCZ | 1,DIGIT  | R100 |
| 068500 | HLT |          | R100 |
| 068600 | BCZ | 1,DIGITS | R100 |
| 068700 | HLT |          | R100 |
| 068800 | BCZ | 1,DISPLA | R100 |
| 068900 | HLT | A5D,0,1  | R100 |
| 069000 | BCZ | 1,DIVIDE | R100 |
| 069100 | HLT |          | R100 |
| 069200 | BCZ | 1,DIVIDE | R100 |
| 069300 | HLT | A6D,0,1  | R100 |
| 069400 | BCZ | 1,DIVISI | R100 |
| 069500 | HLT | A7D,0,1  | R100 |
| 069600 | BCZ | 1,DOLLAR | R100 |

|            |     |          |      |
|------------|-----|----------|------|
| 069700     | HLT | 1,ELSE   | R100 |
| 069800A1E  | BCZ |          | R100 |
| 069900     | HLT |          | R100 |
| 070000A10E | BCZ | 1,END    | R100 |
| 070100     | HLT |          | R100 |
| 070200     | BCZ | 1,ENDING | R100 |
| 070300     | HLT |          | R100 |
| 070400     | BCZ | 1,ENDING | R100 |
| 070500     | HLT | A2E,0,2  | R100 |
| 070600     | BCZ | 1,ENDING | R100 |
| 070700     | HLT | A3E,0,2  | R100 |
| 070800     | BCZ | 1,END-OF | R100 |
| 070900     | HLT | A4E,0,1  | R100 |
| 071000     | BCZ | 1,END-OF | R100 |
| 071100     | HLT | A5E,0,1  | R100 |
| 071200     | BCZ | 1,ENTER  | R100 |
| 071300     | HLT |          | R100 |
| 071400     | BCZ | 1,ENVIRO | R100 |
| 071500     | HLT | A6E,0,1  | R100 |
| 071600     | BCZ | 1,EQUAL  | R100 |
| 071700     | HLT |          | R100 |
| 071800     | BCZ | 1,EQUALS | R100 |
| 071900     | HLT |          | R100 |
| 072000     | BCZ | 1,ERROR  | R100 |
| 072100     | HLT |          | R100 |
| 072200     | BCZ | 1,EVERY  | R100 |
| 072300     | HLT |          | R100 |
| 072400     | BCZ | 1,EXAMIN | R100 |
| 072500     | HLT | A7E,0,1  | R100 |
| 072600     | BCZ | 1,EXCEED | R100 |
| 072700     | HLT | A8E,0,1  | R100 |
| 072800     | BCZ | 1,EXIT   | R100 |
| 072900     | HLT |          | R100 |
| 073000     | BCZ | 1,EXPONE | R100 |
| 073100     | HLT | A9E,0,2  | R100 |
| 073200A1F  | BCZ | 1,FD     | R100 |

|           |     |            |      |
|-----------|-----|------------|------|
| 073300    | HLT | 1, FOR     | R100 |
| 073400    | BCZ |            | R100 |
| 073500    | HLT | 1, FILE    | R100 |
| 073600    | BCZ |            | R100 |
| 073700    | HLT |            | R100 |
| 073800    | BCZ | 1, FILE-C  | R100 |
| 073900    | HLT | A2F, 0, 1  | R100 |
| 074000    | BCZ | 1, FILLER  | R100 |
| 074100    | HLT |            | R100 |
| 074200    | BCZ | 1, FILLIN  | R100 |
| 074300    | HLT | A3F, 0, 1  | R100 |
| 074400    | BCZ | 1, FIRST   | R100 |
| 074500    | HLT |            | R100 |
| 074600    | BCZ | 1, FLEXO   | R100 |
| 074700    | HLT | A4F, 0, 1  | R100 |
| 074800    | BCZ | 1, FLOAT   | R100 |
| 074900    | HLT |            | R100 |
| 075000    | BCZ | 1, FORMAT  | R100 |
| 075100    | HLT |            | R100 |
| 075200    | BCZ | 1, FROM    | R100 |
| 075300    | HLT |            | R100 |
| 075400A1G | BCZ | 1, GIVING  | R100 |
| 075500    | HLT |            | R100 |
| 075600    | BCZ | 1, GO      | R100 |
| 075700    | HLT |            | R100 |
| 075800    | BCZ | 1, GREATER | R100 |
| 075900    | HLT | A2G, 0, 1  | R100 |
| 076000A1H | BCZ | 1, HALF-P  | R100 |
| 076100    | HLT | A5H, 0, 1  | R100 |
| 076200    | BCZ | 1, MASH-T  | R100 |
| 076300    | HLT | A2H, 0, 1  | R100 |
| 076400    | BCZ | 1, HIGH-V  | R100 |
| 076500    | HLT | A3H, 0, 1  | R100 |
| 076600    | BCZ | 1, HIGH-V  | R100 |
| 076700    | HLT | A4H, 0, 1  | R100 |
| 076800    | BCZ | 1, HEADER  | R100 |

|           |     |           |      |
|-----------|-----|-----------|------|
| 076900    | HLT | 1,I-O-CO  | R100 |
| 077000A1I | BCZ | A2I,0,1   | R100 |
| 077100    | HLT | 1,IF      | R100 |
| 077200    | BCZ |           | R100 |
| 077300    | HLT |           | R100 |
| 077400    | BCZ | 1,IN      | R100 |
| 077500    | HLT |           | R100 |
| 077600    | BCZ | 1,INCLUDE | R100 |
| 077700    | HLT | A3I,0,1   | R100 |
| 077800    | BCZ | 1,INPUT   | R100 |
| 077900    | HLT |           | R100 |
| 078000    | BCZ | 1,INPUT-  | R100 |
| 078100    | HLT | A4I,0,1   | R100 |
| 078200    | BCZ | 1,INTO    | R100 |
| 078300    | HLT |           | R100 |
| 0784005I  | BCZ | 1,IS      | R100 |
| 078500    | HLT |           | R100 |
| 078600A1J | BCZ | 1,JUSTIF  | R100 |
| 078700    | HLT | A2J,0,1   | R100 |
| 078800A1L | BCZ | 1,LABEL   | R100 |
| 078900    | HLT |           | R100 |
| 079000    | BCZ | 1,LABELL  | R100 |
| 079100    | HLT | A2L,0,2   | R100 |
| 079200    | BCZ | 1,LEADIN  | R100 |
| 079300    | HLT | A3L,0,1   | R100 |
| 079400    | BCZ | 1,LEAVIN  | R100 |
| 079500    | HLT | A4L,0,1   | R100 |
| 079600    | BCZ | 1,LEFT    | R100 |
| 079700    | HLT |           | R100 |
| 079800    | BCZ | 1,LESS    | R100 |
| 079900    | HLT |           | R100 |
| 080000    | BCZ | 1,LIBRAR  | R100 |
| 080100    | HLT | A5L,0,1   | R100 |
| 080200    | BCZ | 1,LINES   | R100 |
| 080300    | HLT |           | R100 |
| 080400    | BCZ | 1,LOCATI  | R100 |

|           |     |          |      |
|-----------|-----|----------|------|
| 080500    | HLT | A6L,0,1  | R100 |
| 080600    | BCZ | 1,LOCK   | R100 |
| 080700    | HLT |          | R100 |
| 080800    | BCZ | 1,LOW-VA | R100 |
| 080900    | HLT | A7L,0,1  | R100 |
| 081000    | BCZ | 1,LOW-VA | R100 |
| 081100    | HLT | A8L,0,1  | R100 |
| 081200    | BCZ | 1,LOWER- | R100 |
| 081300    | HLT | A9L,0,1  | R100 |
| 081400    | BCZ | 1,LOWER- | R100 |
| 081500    | HLT | A10L,0,1 | R100 |
| 081600A1M | BCZ | 1,MEMORY | R100 |
| 081700    | HLT |          | R100 |
| 081800    | BCZ | 1,MEMORY | R100 |
| 081900    | HLT | A2M,0,1  | R100 |
| 082000    | BCZ | 1,MEMORY | R100 |
| 082100    | HLT | A3M,0,2  | R100 |
| 082200    | BCZ | 1,MINUS  | R100 |
| 082300    | HLT |          | R100 |
| 082400    | BCZ | 1,MODE   | R100 |
| 082500    | HLT |          | R100 |
| 082600    | BCZ | 1,MODULE | R100 |
| 082700    | HLT | A4M,0,1  | R100 |
| 082800    | BCZ | 1,MOVE   | R100 |
| 082900    | HLT |          | R100 |
| 083000    | BCZ | 1,MULTIP | R100 |
| 083100    | HLT | A5M,0,1  | R100 |
| 083200    | BCZ | 1,MULTIP | R100 |
| 083300    | HLT | A6M,0,1  | R100 |
| 083400    | BCZ | 1,MULTIP | R100 |
| 083500    | HLT | A7M,0,1  | R100 |
| 083600A1N | BCZ | 1,NEGATI | R100 |
| 083700    | HLT | A2N,0,1  | R100 |
| 083800    | BCZ | 1,NEXT   | R100 |
| 083900    | HLT |          | R100 |
| 084000    | BCZ | 1,NO     | R100 |

|           |     |           |      |
|-----------|-----|-----------|------|
| 084100    | HLT | 1, NO-MEM | R100 |
| 084200    | BCZ | A3N,0,2   | R100 |
| 084300    | HLT | 1, NOT    | R100 |
| 084400    | BCZ |           | R100 |
| 084500    | HLT |           | R100 |
| 084600    | BCZ | 1, NOTE   | R100 |
| 084700    | HLT |           | R100 |
| 084800    | BCZ | 1, NUMERI | R100 |
| 084900    | HLT | A4N,0,1   | R100 |
| 085000A10 | BCZ | 1, OBJECT | R100 |
| 085100    | HLT | A20,0,2   | R100 |
| 085200    | BCZ | 1, OBJECT | R100 |
| 085300    | HLT | A30,0,2   | R100 |
| 085400    | BCZ | 1, OCCURS | R100 |
| 085500    | HLT |           | R100 |
| 085600    | BCZ | 1, OF     | R100 |
| 085700    | HLT |           | R100 |
| 08580080  | BCZ | 1, OFF    | R100 |
| 085900    | HLT |           | R100 |
| 086000    | BCZ | 1, OMITTE | R100 |
| 086100    | HLT | A40,0,1   | R100 |
| 08620070  | BCZ | 1, ON     | R100 |
| 086300    | HLT |           | R100 |
| 086400    | BCZ | 1, ON-LIN | R100 |
| 086500    | HLT | A90,0,2   | R100 |
| 086600    | BCZ | 1, OPEN   | R100 |
| 086700    | HLT |           | R100 |
| 086800    | BCZ | 1, OPTION | R100 |
| 086900    | HLT | A50,0,1   | R100 |
| 087000    | BCZ | 1, OR     | R100 |
| 087100    | HLT |           | R100 |
| 087200    | BCZ | 1, OTHERW | R100 |
| 087300    | HLT | A60,0,1   | R100 |
| 087400    | BCZ | 1, OUTPUT | R100 |
| 087500    | HLT |           | R100 |
| 087600A1P | BCZ | 1, PAGE   | R100 |

|           |     |          |      |
|-----------|-----|----------|------|
| 087700    | HLT | 1,PAPER- | R100 |
| 087800    | BCZ | A13P,0,2 | R100 |
| 087900    | HLT | 1,PAPER- | R100 |
| 088000    | BCZ | A14P,0,2 | R100 |
| 088100    | HLT | 1,PERFOR | R100 |
| 088200    | BCZ | A2P,0,1  | R100 |
| 088300    | HLT | 1,PICTUR | R100 |
| 088400    | BCZ | A3P,0,1  | R100 |
| 088500    | HLT | 1,PLACES | R100 |
| 088600    | BCZ | 1,PLUS   | R100 |
| 088700    | HLT | 1,POINT  | R100 |
| 088800    | BCZ | 1,POSITI | R100 |
| 088900    | HLT | A4P,0,1  | R100 |
| 089000    | BCZ | 1,POSITI | R100 |
| 089100    | HLT | A5P,0,1  | R100 |
| 089200    | BCZ | 1,PREPAR | R100 |
| 089300    | HLT | A6P,0,1  | R100 |
| 089400    | BCZ | 1,PRIORI | R100 |
| 089500    | HLT | A7P,0,1  | R100 |
| 089600    | BCZ | 1,PROCED | R100 |
| 089700    | HLT | A8P,0,1  | R100 |
| 089800    | BCZ | 1,PROCEE | R100 |
| 089900    | HLT | A9P,0,1  | R100 |
| 090000    | BCZ | 1,PROTEC | R100 |
| 090100    | HLT | A10P,0,1 | R100 |
| 090200    | BCZ | 1,PROTEC | R100 |
| 090300    | HLT | A11P,0,1 | R100 |
| 090400    | BCZ | 1,PURGE- | R100 |
| 090500    | HLT | A12P,0,1 | R100 |
| 090600    | BCZ | 1,QUOTE  | R100 |
| 090700    | HLT | 1,RANGE  | R100 |
| 090800    | BCZ |          |      |
| 090900    | HLT |          |      |
| 091000A19 | BCZ |          |      |
| 091100    | HLT |          |      |
| 091200A1R | BCZ |          |      |

|           |     |           |      |
|-----------|-----|-----------|------|
| 091300    | HLT | 1, READ   | R100 |
| 091400    | BCZ |           | R100 |
| 091500    | HLT | 1, RECORD | R100 |
| 091600    | BCZ |           | R100 |
| 091700    | HLT | 1, RECORD | R100 |
| 091800    | BCZ | A2R,0,1   | R100 |
| 091900    | HLT | 1, RECORD | R100 |
| 092000    | BCZ | A3R,0,1   | R100 |
| 092100    | HLT | 1, RECORD | R100 |
| 092200    | BCZ | A4R,0,1   | R100 |
| 092300    | HLT | 1, REDEFI | R100 |
| 092400    | BCZ | A5R,0,1   | R100 |
| 092500    | HLT | 1, REEL   | R100 |
| 092600    | BCZ |           | R100 |
| 092700    | HLT | 1, REEL-N | R100 |
| 092800    | BCZ | A6R,0,1   | R100 |
| 092900    | HLT | 1, RENAMI | R100 |
| 093000    | BCZ | A7R,0,1   | R100 |
| 093100    | HLT | 1, REPLAC | R100 |
| 093200    | BCZ | A8R,0,1   | R100 |
| 093300    | HLT | 1, RERUN  | R100 |
| 093400    | BCZ |           | R100 |
| 093500    | HLT | 1, RESERV | R100 |
| 093600    | BCZ | A9R,0,1   | R100 |
| 093700    | HLT | 1, REVERS | R100 |
| 093800    | BCZ | A10R,0,1  | R100 |
| 093900    | HLT | 1, REWIND | R100 |
| 094000    | BCZ |           | R100 |
| 094100    | HLT | 1, RIGHT  | R100 |
| 094200    | BCZ |           | R100 |
| 094300    | HLT | 1, ROUNDE | R100 |
| 094400    | BCZ | A11R,0,1  | R100 |
| 094500    | HLT | 1, RUN    | R100 |
| 094600    | BCZ |           | R100 |
| 094700    | HLT | 1, SAME   | R100 |
| 094800A1S | BCZ |           |      |

|        |     |            |      |
|--------|-----|------------|------|
| 094900 | HLT | 1, SECTION | R100 |
| 095000 | BCZ | A2S,0,1    | R100 |
| 095100 | HLT | 1, SELECT  | R100 |
| 095200 | BCZ |            | R100 |
| 095300 | HLT | 1, SENTEN  | R100 |
| 095400 | BCZ | A3S,0,1    | R100 |
| 095500 | HLT | 1, SENTIN  | R100 |
| 095600 | BCZ | A4S,0,1    | R100 |
| 095700 | HLT | 1, SEQUEN  | R100 |
| 095800 | BCZ | A5S,0,1    | R100 |
| 095900 | HLT | 1, SIGN    | R100 |
| 096000 | BCZ |            | R100 |
| 096100 | HLT | 1, SIGNED  | R100 |
| 096200 | BCZ |            | R100 |
| 096300 | HLT | 1, SIZE    | R100 |
| 096400 | BCZ |            | R100 |
| 096500 | HLT | 1, SOURCE  | R100 |
| 096600 | BCZ | A6S,0,1    | R100 |
| 096700 | HLT | 1, SPACE   | R100 |
| 096800 | BCZ |            | R100 |
| 096900 | HLT | 1, SPACES  | R100 |
| 097000 | BCZ |            | R100 |
| 097100 | HLT | 1, SPECIA  | R100 |
| 097200 | BCZ | A7S,0,2    | R100 |
| 097300 | HLT | 1, STANDA  | R100 |
| 097400 | BCZ | A8S,0,1    | R100 |
| 097500 | HLT | 1, STATUS  | R100 |
| 097600 | BCZ |            | R100 |
| 097700 | HLT | 1, STOP    | R100 |
| 097800 | BCZ |            | R100 |
| 097900 | HLT | 1, SUBTRA  | R100 |
| 098000 | BCZ | A9S,0,1    | R100 |
| 098100 | HLT | 1, SUPERV  | R100 |
| 098200 | BCZ | A10S,0,1   | R100 |
| 098300 | HLT | 1, SUPPRE  | R100 |
| 098400 | BCZ |            | R100 |

|           |     |           |      |
|-----------|-----|-----------|------|
| 098500    | HLT | A11S,0,1  | R100 |
| 098600    | BCZ | 1,SYNCHR  | R100 |
| 098700    | HLT | A12S,0,1  | R100 |
| 098800A1T | BCZ | 1,TALLY   | R100 |
| 098900    | HLT |           | R100 |
| 099000    | BCZ | 1,TALLYI  | R100 |
| 099100    | HLT | A2T,0,1   | R100 |
| 099200    | BCZ | 1,TAPE    | R100 |
| 099300    | HLT |           | R100 |
| 099400    | BCZ | 1,TEST-P  | R100 |
| 099500    | HLT | A3T,0,1   | R100 |
| 099600    | BCZ | 1,THAN    | R100 |
| 099700    | HLT |           | R100 |
| 099800    | BCZ | 1,THEN    | R100 |
| 099900    | HLT |           | R100 |
| 100000    | BCZ | 1,THROUGH | R100 |
| 100100    | HLT | A4T,0,1   | R100 |
| 100200    | BCZ | 1,THRU    | R100 |
| 100300    | HLT |           | R100 |
| 100400    | BCZ | 1,TIMES   | R100 |
| 100500    | HLT |           | R100 |
| 100600    | BCZ | 1,TO      | R100 |
| 100700    | HLT |           | R100 |
| 100800    | BCZ | 1,TRAIL   | R100 |
| 100900    | HLT | A5T,0,1   | R100 |
| 101000    | BCZ | 1,TYPE    | R100 |
| 101100    | HLT |           | R100 |
| 101200A1U | BCZ | 1,UNEQUA  | R100 |
| 101300    | HLT | A2U,0,1   | R100 |
| 101400    | BCZ | 1,UPPER-  | R100 |
| 101500    | HLT | A3U,0,1   | R100 |
| 101600    | BCZ | 1,UPPER-  | R100 |
| 101700    | HLT | A4U,0,1   | R100 |
| 101800    | BCZ | 1,UNTIL   | R100 |
| 101900    | HLT |           | R100 |
| 102000    | BCZ | 1,UPON    | R100 |

|           |     |                  |      |
|-----------|-----|------------------|------|
| 102100    | HLT | 1,USAGE          | R100 |
| 102200    | BCZ |                  | R100 |
| 102300    | HLT |                  | R100 |
| 102400    | BCZ | 1,USE            | R100 |
| 102500    | HLT |                  | R100 |
| 102600A1V | BCZ | 1,VALUE          | R100 |
| 102700    | HLT |                  | R100 |
| 102800    | BCZ | 1,VARYIN         | R100 |
| 102900    | HLT | A2V,0,1          | R100 |
| 103000A1W | BCZ | 1,WHEN           | R100 |
| 103100    | HLT |                  | R100 |
| 103200    | BCZ | 1,WITH           | R100 |
| 103300    | HLT |                  | R100 |
| 103400    | BCZ | 1,WORDS          | R100 |
| 103500    | HLT |                  | R100 |
| 103600    | BCZ | 1,WORD-S         | R100 |
| 103700    | HLT | A3W,0,3          | R100 |
| 103800    | BCZ | 1,WORKIN         | R100 |
| 103900    | HLT | A2W,0,2          | R100 |
| 104000    | BCZ | 1,WRITE          | R100 |
| 104100    | HLT |                  | R100 |
| 104200A1Z | BCZ | 1,ZERO           | R100 |
| 104300    | HLT |                  | R100 |
| 104400    | BCZ | 1,ZEROES         | R100 |
| 104500    | HLT |                  | R100 |
| 104600    | BCZ | 1,ZEROS          | R100 |
| 104700    | HLT |                  | R100 |
| 104800A2A | BCZ | 1,S              | R100 |
| 104900A3A | BCZ | 1,ING            | R100 |
| 105000A4A | BCZ | 1,ETIC           | R100 |
| 105100A5A | BCZ | 1,UMERIC         | R100 |
| 105200A6A | BCZ | 1,ATE            | R100 |
| 105300A7A | BCZ | 1,LATOR          | R100 |
| 105400A3B | BCZ | 4,ING-FILE-LABEL | R100 |
| 105500A4B | BCZ | 4,ING-TAPE-LABEL | R100 |
| 105600A5B | BCZ | 1,COUNT          | R100 |

|            |     |               |      |
|------------|-----|---------------|------|
| 1057002CA  | BCZ | 1,TER         | R100 |
| 1058003CA  | BCZ | 1,TERS        | R100 |
| 105900A4C  | BCZ | 1,UNITS       | R100 |
| 106000A5C  | BCZ | 2,ATIONAL-1   | R100 |
| 106100A6C  | BCZ | 2,ATIONAL-2   | R100 |
| 106200A7C  | BCZ | 1,E           | R100 |
| 106300A8C  | BCZ | 1,NT          | R100 |
| 106400A9C  | BCZ | 2,URATION     | R100 |
| 106500A10C | BCZ | 1,NS          | R100 |
| 106600A11C | BCZ | 1,L           | R100 |
| 106700A12C | BCZ | 2,PONDING     | R100 |
| 106800A13C | BCZ | 1,EADER       | R100 |
| 106900A14C | BCZ | 1,UNCH        | R100 |
| 107000A2D  | BCZ | 1,ITTEN       | R100 |
| 107100A3D  | BCZ | 1,ATIVES      | R100 |
| 107200A5D  | BCZ | 1,Y           | R100 |
| 107300A6D  | BCZ | 1,D           | R100 |
| 107400A7D  | BCZ | 1,ON          | R100 |
| 107500A2E  | BCZ | 2,-FILE-LABEL | R100 |
| 107600A3E  | BCZ | 2,-TAPE-LABEL | R100 |
| 107700A4E  | BCZ | 1,-FILE       | R100 |
| 107800A5E  | BCZ | 1,-TAPE       | R100 |
| 107900A6E  | BCZ | 1,NMENT       | R100 |
| 108000A9E  | BCZ | 2,NTIATED     | R100 |
| 108100A2F  | BCZ | 1,ONTROL      | R100 |
| 108200A3F  | BCZ | 1,G           | R100 |
| 108300A4F  | BCZ | 1,RTER        | R100 |
| 108400A2G  | BCZ | 1,R           | R100 |
| 108500A2H  | BCZ | 1,OTAL        | R100 |
| 108600A3H  | BCZ | 1,ALUE        | R100 |
| 108700A4H  | BCZ | 1,ALUES       | R100 |
| 108800A5H  | BCZ | 1,AGE         | R100 |
| 108900A2I  | BCZ | 1,NTROL       | R100 |
| 109000A4I  | BCZ | 1,OUTPUT      | R100 |
| 109100A2J  | BCZ | 1,IED         | R100 |
| 109200A2L  | BCZ | 2,ED-ITEM     | R100 |

|            |     |               |      |
|------------|-----|---------------|------|
| 109300A7L  | BCZ | 1,LUE         | R100 |
| 109400A8L  | BCZ | 1,LUES        | R100 |
| 109500A9L  | BCZ | 1,BOUND       | R100 |
| 109600A10L | BCZ | 1,BOUNDS      | R100 |
| 109700A2M  | BCZ | 1,-DUMP       | R100 |
| 109800A3M  | BCZ | 2,-DUMP-KEY   | R100 |
| 109900A5M  | BCZ | 1,LE          | R100 |
| 110000A6M  | BCZ | 1,LIED        | R100 |
| 110100A7M  | BCZ | 1,LY          | R100 |
| 110200A2N  | BCZ | 1,VE          | R100 |
| 110300A3N  | BCZ | 2,ORY-DUMP    | R100 |
| 110400A4N  | BCZ | 1,C           | R100 |
| 110500A2O  | BCZ | 2,-COMPUTER   | R100 |
| 110600A3O  | BCZ | 2,-PROGRAM    | R100 |
| 110700A5O  | BCZ | 1,AL          | R100 |
| 110800A6O  | BCZ | 1,ISE         | R100 |
| 110900A9O  | BCZ | 2,E-PRINTER   | R100 |
| 111000A2P  | BCZ | 1,M           | R100 |
| 111100A6P  | BCZ | 1,ED          | R100 |
| 111200A7P  | BCZ | 1,TY          | R100 |
| 111300A8P  | BCZ | 1,URE         | R100 |
| 111400A10P | BCZ | 1,T           | R100 |
| 111500A11P | BCZ | 1,TION        | R100 |
| 111600A12P | BCZ | 1,DATE        | R100 |
| 111700A13P | BCZ | 2,TAPE-PUNCH  | R100 |
| 111800A14P | BCZ | 2,TAPE-READER | R100 |
| 111900A2R  | BCZ | 1,-COUNT      | R100 |
| 112000A5R  | BCZ | 1,NES         | R100 |
| 112100A6R  | BCZ | 1,UMBER       | R100 |
| 112200A7R  | BCZ | 1,NG          | R100 |
| 112300A2S  | BCZ | 1,N           | R100 |
| 112400A3S  | BCZ | 1,CE          | R100 |
| 112500A4S  | BCZ | 1,EL          | R100 |
| 112600A5S  | BCZ | 1,CED         | R100 |
| 112700A7S  | BCZ | 2,L-NAMES     | R100 |
| 112800A8S  | BCZ | 1,RD          | R100 |

|            |     |                          |      |
|------------|-----|--------------------------|------|
| 112900A9S  | BCZ | 1,CT                     | R100 |
| 113000A10S | BCZ | 1,ISOR                   | R100 |
| 113100A11S | BCZ | 1,SS                     | R100 |
| 113200A12S | BCZ | 1,OMIZED                 | R100 |
| 113300A2T  | BCZ | 1,NG                     | R100 |
| 113400A3T  | BCZ | 1,ATTERN                 | R100 |
| 113500A4T  | BCZ | 1,H                      | R100 |
| 113600A2W  | BCZ | 2,G-STORAGE              | R100 |
| 113700A3W  | BCZ | 3,WITCH-REGISTER         | R100 |
| 113800BSP  | BCZ | 1,BBEE                   | R100 |
| 091000A    | MOV | IR3,J10                  | R110 |
| 091100     | MOV | IR4,J11                  | R110 |
| 091200     | MVZ | IR1                      | R110 |
| 091210     | MOV | =1,IR2                   | R110 |
| 091212     | RPT | 4,0,C                    | R110 |
| 091214     | MVZ | TB00A                    | R110 |
| 091600     | LOD | SAV3,0,IR3               | R110 |
| 091700     | LOD | SAV4,0,IR4               | R110 |
| 091800     | MOV | PCS,SAPCS                | R110 |
| 091900     | MOV | =30,TB00B                | R110 |
| 092000     | MVZ | TB00C                    | R110 |
| 092200     | MVZ | TB00W                    | R110 |
| 092300     | MOV | =1,CNT2                  | R110 |
| 092400     | SNR | *+1,0,00F0C              | R110 |
| 092500     | SNR | *+1,0,01F0C              | R110 |
| 092600     | SNR | *+1,0,04F0C              | R110 |
| 092700     | SNR | *+1,0,07F0C              | R110 |
| 092800A5   | LDQ | LA00A,3                  | R110 |
| 092900A13  | CLA | LA00T                    | R110 |
| 093000     | TRZ | A1                       | R110 |
| 093100     | SUB | =1,0,3                   | R110 |
| 093200     | STR | LA00T                    | R110 |
| 093300     | CLA | ZERO                     | R110 |
| 093400     | SLL | 6,0,3                    | R110 |
| 093500     | STR | IR2                      | R110 |
| 093600     | TRU | TA00A,2                  | R110 |
|            |     | SAVE IR3 AND 4           |      |
|            |     | CLEAR IRI                |      |
|            |     | SAVE PCS                 |      |
|            |     | SET TO ZERO              |      |
|            |     | PERIOD                   |      |
|            |     | COMMA                    |      |
|            |     | CONTINUATION SWITCH      |      |
|            |     | RESET NON NUMERIC SWITCH |      |
|            |     | PICK UP WORD             |      |
|            |     | IS THIS WORD PROCESSED   |      |
|            |     | DECREASE BY ONE          |      |
|            |     | CLEAR ACC                |      |
|            |     | CHARACTER TO INDEX 2     |      |
|            |     | TEST CHARACTER           |      |

|           |     |                |                                  |      |
|-----------|-----|----------------|----------------------------------|------|
| 093700A1  | CLA | =6             | SET CHARACTER COUNT FOR NEW WORD | R110 |
| 093800    | STR | LA00T          | TO SIX                           | R110 |
| 093900    | TRX | A5,3,1         |                                  | R110 |
|           | SEN | A7,0,02F0C     |                                  |      |
| 094900    | ADB | LL00C,0,1      | UPDATE LINE COUNT                | R110 |
| 095000    | LDQ | BSP            |                                  | R110 |
| 095100    | SLL | 12,0,1         |                                  | R110 |
| 095200    | LDQ | =/77777777     |                                  | R110 |
| 095300    | MSK | LL00A+14       |                                  | R110 |
| 095400    | TRL | WR,0,TP3       |                                  | R110 |
|           | PRE | LL00A,0,LF0N   |                                  |      |
| 095600    | PZE | LL00W,LL00C    | LINE ALL PROCESSED               | R110 |
|           | TRL | RD,0,SP0       |                                  |      |
|           | HLT | LL00A          |                                  |      |
| 096100    | HLT |                |                                  | R110 |
| 096200    | MOV | =1,IR2         | SAVE INPUT LINE                  | R110 |
| 096300    | RPT | 13,0,1         | X                                | R110 |
| 096400    | MOV | LL00A+1,LA00A  |                                  | R110 |
| 096500    | LDQ | =/7777         | SPACE FILL LAST                  | R110 |
| 096600    | CLA | =/0505         | TWO CHARACTERS                   | R110 |
| 096700    | MSK | LA00A+12       |                                  | R110 |
| 096800    | MOV | =6,CNT         | COLUMN COUNT                     | R110 |
| 096900    | MVZ | CNT2           |                                  | R110 |
| 097000    | SEN | A24,0,09F0C    | ARE WE IN SPECIAL MODE           |      |
|           | MOV | =-1,LL00B      |                                  |      |
| 097300A23 | CLA | LA00A          | PICK UP SEQUENCE NUMBER          | R110 |
|           | TRC | =/050505050505 |                                  |      |
|           | TRU | *+3            |                                  |      |
|           | TRU | *+2            |                                  |      |
|           | TRU | A8             |                                  |      |
| 097400    | TRC | LA00C          | COMPARE WITH LAST NUMBER         | R110 |
| 097500    | TRL | ERRB,2,FG4     |                                  | R110 |
| 097600    | TRU | A8             | SERIAL NUMBER CHECKS             | R110 |
| 097700    | TRU | *-2            |                                  | R110 |
| 097800A8  | STR | LA00C          | STORE NEW SERIAL NUMBER          | R110 |
| 097900A24 | LSX | 1,3,12         | SET INDEX REGISTERS              | R110 |

|            |     |              |                           |      |
|------------|-----|--------------|---------------------------|------|
| 098000     | LDQ | LA00A+1      | PICK UP SECOND WORD       | R110 |
| 098010     | SMR | *+1,0,04FOC  | RESET CONTINUATION SWITCH | R110 |
| 098100     | CLA | ZERO         | CLEAR ACC                 | R110 |
| 098200     | SLL | 6,0,3        | GET FIRST CHARACTER       | R110 |
| 098300     | SEN | A25,0,09FOC  | ARE WE IN SPECIAL MODE    |      |
|            | SUB | =/05,0,1     | IS THERE A SPACE IN 7     | R110 |
|            | TRZ | A22B         |                           |      |
| 098500     | ADD | =/05,0,1     | NO                        | R110 |
| 098600     | SUB | =/41,0,1     | IS THERE A - IN 7         | R110 |
| 098700     | TRZ | *+2          | YES                       | R110 |
| 098800     | TRU | A22A         | NO                        | R110 |
| 099000     | SMR | *+1,0,03FOC  | RESET SPACE FLIP-FLOP     | R110 |
|            | ADB | LL00B,0,1    |                           |      |
| 099050     | CLA | CNT          | ADJUST COLUMN COUNT       | R110 |
| 099055     | ADD | =1,0,1       |                           | R110 |
| 099060     | STR | CNT          |                           | R110 |
| 099100     | SNS | A25,0,04FOC  | CONTINUATION BIT PRESENT  | R110 |
| 099200     | HLT | H1           |                           | R110 |
| 099220A22A | ADD | =/41,0,1     |                           | R110 |
| 099225     | STR | IR2          |                           | R110 |
| 099230     | MOV | IR1,ERR2     |                           | R110 |
| 099235     | TRL | ERR10,1,FG10 |                           | R110 |
| 099240     | TRU | A14          |                           | R110 |
|            | CLA | TB00C        | TEST CHARACTER COUNT      | R110 |
|            | TRZ | *+2          |                           |      |
|            | TRU | A22          |                           |      |
| 099300A22  | ADB | LL00B,0,1    | CARD WORD COUNT           | R110 |
| 099400     | CLA | TB00C        | TEST CHARACTER COUNT      | R110 |
| 099500     | TRZ | A5           | GET NEXT CHARACTER        | R110 |
| 099600     | MOV | IR3,SAV3     |                           | R110 |
| 099700     | MOV | IR4,SAV4     |                           | R110 |
| 099800     | CLA | TB00B        | CHECK SHIFT COUNT         | R110 |
| 099900     | TRC | =30          |                           | R110 |
| 100000     | TRU | *+3          |                           | R110 |
| 100100     | TRL | ERR,1,FG2    |                           | R110 |
|            | TRU | *+2          |                           |      |

|           |     |              |                                 |      |
|-----------|-----|--------------|---------------------------------|------|
| 100200    | ADB | TB00W,0,1    | UPDATE WORD COUNT               | R110 |
| 100300    | MOV | J10,IR3      | RELOAD IR 3 AND 4               | R110 |
| 100400    | MOV | J11,IR4      | X                               | R110 |
| 100500    | ADB | LL00B,0,1    | CARD WORD COUNT                 |      |
| 100600    | CLA | TB00A        | PICK UP FIRST SIX CHARACTERS    | R110 |
| 100700A25 | LOD | SAPCS,0,PCT  | RETURN                          | R110 |
| 100800    | CLA | LA00T        | DECREASE CHARACTER COUNT BY ONE | R110 |
| 100900    | SUB | =1,0,3       | AND RETURN TO GET               | R110 |
| 101000    | TRU | A13+1        | NEXT CHARACTER                  | R110 |
| 101000A12 | TRL | ERR,1,FG11   |                                 | R110 |
| 101100    | TRU | A14          |                                 | R110 |
| 101190A6  | SNR | A14,0,03F0C  |                                 | R110 |
| 101200    | MOV | IR1,ERR2     |                                 | R110 |
| 101300    | TRL | ERR10,1,FG1  |                                 | R110 |
| 101400A26 | SNR | *+1,0,04F0C  | RESET CONTINUATION SWITCH       | R110 |
| 101500    | SNR | A14,0,03F0C  |                                 | R110 |
| 101600    | SNS | A4,0,07F0C   |                                 | R110 |
| 101700A4  | CLA | CNT          | SET NON-NUMERIC SWITCH          | R110 |
| 101800    | SNR | *+1,0,04F0C  | UP DATE COLUMN COUNT            | R110 |
| 101900    | ADD | =1,0,1       | RESET CONTINUATION SWITCH       | R110 |
| 102000    | STR | CNT          | X                               | R110 |
| 102100    | SEN | A12,0,00F0C  | IS THERE A PERIOD               | R110 |
| 102200    | SEN | A12,0,01F0C  | IS THERE A COMMA                | R110 |
| 102300    | SNR | A14,0,03F0C  | SPACE HAS BEEN FOUND            | R110 |
| 102400    | CLA | TB00C        | UPDATE CHARACTER                | R110 |
| 102500    | ADD | =1,0,3       |                                 | R110 |
| 102600    | STR | TB00C        | MORE THAN 30 CHARACTERS         | R110 |
| 102700    | SUB | =31,0,3      | NO                              | R110 |
| 102800    | TRN | A15          | ERROR FIND END OF               | R110 |
| 102900    | TRL | ERRB,2,FG1   | WORD                            | R110 |
| 102910    | TRU | A13          | ADJUST COLUMN COUNT             | R110 |
| 102990A3  | CLA | CNT          |                                 | R110 |
| 102993    | ADD | =1,0,1       |                                 | R110 |
| 102997    | STR | CNT          | SPECIAL MODE                    | R110 |
| 103000    | SEN | A3AA,0,09F0C | SENSE CONTINUATION SWITCH       | R110 |
|           |     | A25,0,04F0C  |                                 |      |

A-38

|            |     |              |                            |      |
|------------|-----|--------------|----------------------------|------|
| 103100     | CLA | TB00C        | SPACE TEST CHARACTER COUNT | R110 |
| 103200     | TRZ | A13          |                            | R110 |
| 103300     | SEN | A18,0,00F0C  | DOES SPACE FOLLOW PERIOD   | R110 |
| 103400     | SEN | A18,0,01F0C  | OR COMMA                   | R110 |
| 103500     | SNS | A13,0,03F0C  | LOOK FOR NEXT CHARACTER    | R110 |
| 103600     | TRU | A13          |                            | R110 |
|            | CLA | TB00C        | GET CHARACTER COUNT        | R110 |
|            | TRZ | *+2          |                            |      |
|            | TRU | A18          | RETURN NOT BLANK           |      |
|            | CLA | CNT          |                            |      |
|            | TRC | *76          |                            |      |
|            | TRU | A3           |                            |      |
|            | HLT |              |                            |      |
|            | MOV | SAPCS,PCT    |                            |      |
| 103700A14  | CLA | IR2          | REPLACE CHARACTER          | R110 |
| 103800     | SRL | 6            | AND                        | R110 |
| 103900     | CLA | LA00T        | UPDATE                     | R110 |
| 104000     | ADD | *1,0,3       | COUNTER                    | R110 |
| 104100     | STR | LA00T        |                            | R110 |
| 104200A18  | CLA | ORG          | REPLACE WORD IN            | R110 |
| 104300     | STR | LA00A,3      | BUFFER                     | R110 |
| 104400     | TRU | A22          | RETURN                     | R110 |
| 104490A2   | SMR | A14,0,03F0C  |                            | R110 |
| 104500     | MOV | IR1,ERR2     |                            | R110 |
| 104600     | TRL | ERR10,1,FG1  |                            | R110 |
| 104610     | TRU | A26          |                            | R110 |
| 104700A11  | SNS | *+2,0,01F0C  | COMMA FOUND                | R110 |
| 104800     | TRL | ERRB,2,FG11  |                            | R110 |
| 104900     | SMR | A13A,0,04F0C | RESET CONTINUATION SWITCH  | R110 |
| 105000A13A | CLA | CNT          | UPDATE COLUMN COUNT        | R110 |
| 105100     | ADD | *1,0,1       | X                          | R110 |
| 105200     | STR | CNT          | X                          | R110 |
| 105300     | TRU | A13          |                            | R110 |
| 105400A9   | SNS | *+2,0,00F0C  | PERIOD FOUND               | R110 |
| 105500     | TRL | ERRB,2,FG11  |                            | R110 |
| 105600     | SMR | A13A,0,04F0C | RESET CONTINUATION SWITCH  | R110 |

|             |     |           |                       |      |
|-------------|-----|-----------|-----------------------|------|
| 105700      | TRU | A13A      | PACK CHARACTER        | R110 |
| 105800A15   | CLA | TB00B     |                       | R110 |
| 105900      | RPA | A16       |                       | R110 |
| 106000      | CLA | IR2       |                       | R110 |
| 106100A16   | SHL | **0,1     |                       | R110 |
| 106200      | ADD | TB00A,1,1 |                       | R110 |
| 106300      | STR | TB00A,1   |                       | R110 |
| 106400      | CLA | TB00B     |                       | R110 |
| 106500      | SUB | =6,0,3    | REDUCE SHIFT          | R110 |
| 106600      | TRN | A17       | COUNT                 | R110 |
| 106700      | STR | TB00B     | GET NEXT CHARACTER    | R110 |
| 106800      | TRU | A13       |                       | R110 |
| 106900A17   | CLA | IR1       | UPDATE INDEX REGISTER | R110 |
| 107000      | ADD | =1,0,3    | X                     | R110 |
| 107100      | STR | IR1       | X                     | R110 |
| 107200      | MOV | =30,TB00B | RESET SHIFT COUNT     | R110 |
| 107300      | CLA | TB00W     | UPDATE WORD           | R110 |
| 107400      | ADD | =1,0,1    | COUNT                 | R110 |
| 107500      | STR | TB00W     | X                     | R110 |
| 107800      | TRU | A13       |                       | R110 |
| 107900TA00A | TRU | A6        | 00 MS                 | R110 |
| 108000      | TRU | A6        | 01 UC                 | R110 |
| 108100      | TRU | A6        | 02 LC                 | R110 |
| 108200      | TRU | A6        | 03 TB                 | R110 |
| 108300      | TRU | A6        | 04 CR                 | R110 |
| 108400A19   | TRU | A3        | 05 SPACE              | R110 |
| 108500      | TRU | A26       | 06 A                  | R110 |
| 108600      | TRU | A26       | 07 B                  | R110 |
| 108700      | TRU | A26       | 10 C                  | R110 |
| 108800      | TRU | A26       | 11 D                  | R110 |
| 108900      | TRU | A26       | 12 E                  | R110 |
| 109000      | TRU | A26       | 13 F                  | R110 |
| 109100      | TRU | A26       | 14 G                  | R110 |
| 109200      | TRU | A26       | 15 H                  | R110 |
| 109300      | TRU | A26       | 16 I                  | R110 |
| 109400      | TRU | A26       | 17 J                  | R110 |

|           |     |     |    |             |   |      |
|-----------|-----|-----|----|-------------|---|------|
| 109500    | TRU | A26 | 40 | 20          | K | R110 |
| 109600    | TRU | A26 | 41 | 21          | L | R110 |
| 109700    | TRU | A26 | 42 | 22          | M | R110 |
| 109800    | TRU | A26 | 43 | 23          | N | R110 |
| 109900    | TRU | A26 | 44 | 24          | O | R110 |
| 110000    | TRU | A26 | 45 | 25          | P | R110 |
| 110100    | TRU | A26 | 46 | 26          | Q | R110 |
| 110200    | TRU | A26 | 47 | 27          | R | R110 |
| 110300    | TRU | A26 | 48 | 30          | S | R110 |
| 110400    | TRU | A26 | 49 | 31          | T | R110 |
| 110500    | TRU | A26 | 50 | 32          | U | R110 |
| 110600    | TRU | A26 | 51 | 33          | V | R110 |
| 110700    | TRU | A26 | 52 | 34          | W | R110 |
| 110800    | TRU | A26 | 53 | 35          | X | R110 |
| 110900    | TRU | A26 | 54 | 36          | Y | R110 |
| 111000    | TRU | A26 | 55 | 37          | Z | R110 |
| 111100    | TRU | A2  | 56 | RP          |   | R110 |
| 111200A20 | TRU | A4  | 57 | -           |   | R110 |
| 111300    | TRU | A2  | 58 | +           |   | R110 |
| 111400    | TRU | A2  | 59 | LESS        |   | R110 |
| 111500    | TRU | A2  | 60 | =           |   | R110 |
| 111600    | TRU | A2  | 61 | GREATER     |   | R110 |
| 111700    | TRU | A4  | 62 | MINUS       |   | R110 |
| 111800    | TRU | A2  | 63 | DOLLAR SIGN |   | R110 |
| 111900    | TRU | A2  | 64 | *           |   | R110 |
| 112000    | TRU | A2  | 65 | LP          |   | R110 |
| 112100    | TRU | A2  | 66 | QUOTE       |   | R110 |
| 112200    | TRU | A2  | 67 | QUESTION    |   | R110 |
| 112300    | TRU | A2  | 68 |             |   | R110 |
| 112400    | TRU | A2  | 69 |             |   | R110 |
| 112500    | TRU | A11 | 70 |             |   | R110 |
| 112600    | TRU | A6  | 71 | STOP CODE   |   | R110 |
| 112700    | TRU | A4  | 72 | 0           |   | R110 |
| 112800    | TRU | A4  | 73 | 1           |   | R110 |
| 112900    | TRU | A4  | 74 | 2           |   | R110 |
| 113000    | TRU | A4  | 75 | 3           |   | R110 |

R110  
R110  
R110  
R110  
R110  
R110  
R110

64 4  
65 5  
66 6  
67 7  
70 8  
71 9  
72 APOS  
73 .  
74 /  
75 .  
76 SPECIAL  
77 CODE DELETE

R110  
R110  
R110  
R110  
R110  
R100  
R100  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R110  
R100  
R100  
R100  
R100

A-42

PUT NAME IN COLUMN 8

A4  
A4  
A4  
A4  
A4  
A4  
A4  
A2  
A11  
A2  
A9  
A6  
A6  
0  
0

TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
TRU  
OCT  
OCT  
HLT  
TRS  
TRS  
TRS  
HLT  
HLT  
HLT  
CLA  
SUB  
TRC  
TRU  
TRU  
ADB  
TRS  
MOV  
TRL  
MOV  
LSX  
TRL  
TRL  
TRL

113100  
113200  
113300  
113400  
113500  
113600  
113700  
113800  
113900  
114000  
114100  
114200  
114300SAV3  
SAV4  
137000SAPCS  
114500K  
114600K8  
114700K12  
114800CNT  
114900CNT1  
115000CNT2  
115100CK1  
115200CK  
115300  
115400  
115500  
115600  
115700  
115800  
115900CK2  
116000  
116100  
139700SCAN  
139800SCN1  
139900  
140000

CNT  
TB00C,0,1  
=6  
CK2  
\*+1  
PCS,0,1  
PCS,CK1  
ERR,1,FG10  
CK1,PCT  
0,3,7777  
A  
CK  
K12

MINOR ERROR

|             |     |            |                              |      |
|-------------|-----|------------|------------------------------|------|
| 140100SCAN9 | CLA | TB00A      | PICK UP FIRST SIX CHARACTERS | R100 |
| 140400SCAN6 | TRC | ID,3       |                              | R100 |
| 140500SCAN7 | TRU | SCAN1      |                              | R100 |
| 140600      | TRU | SCAN1      |                              | R100 |
| 140700SCAN8 | CLA | ID+1,3     | PICK UP SECOND WORD          | R100 |
| 140800      | TRN | SCAN2      | COMPLETE COMPARISON          | R100 |
| 140900      | RPA | PCT        | TRANSFER TO - MATCH          | R100 |
| 141000SCAN1 | CLA | ID+2,3     | PICK UP THIRD WORD           | R100 |
| 141100      | RPA | PCT        | MATCH                        | R100 |
| 141500H1    | TRX | SCAN1,3,3  |                              | R100 |
| 141600H2    | TRX | SCAN1,3,6  |                              | R100 |
| 141700H3    | TRX | SCAN1,3,9  |                              | R100 |
| 141800H4    | TRX | SCAN1,3,12 |                              | R100 |
| 141900H5    | TRX | SCAN1,3,15 |                              | R100 |
| 142000H6    | TRX | SCAN1,3,18 |                              | R100 |
| 142100H7    | TRX | SCAN1,3,21 |                              | R100 |
| 142200H8    | TRX | SCAN1,3,24 |                              | R100 |
| 142300H9    | TRX | SCAN1,3,27 |                              | R100 |
| 142400H10   | TRX | SCAN1,3,30 |                              | R100 |
| 142500H11   | TRX | SCAN1,3,33 |                              | R100 |
| 142600H12   | TRX | SCAN1,3,36 |                              | R100 |
| 142700H1A   | TRX | SCAN9,3,3  |                              | R100 |
| 142800H2A   | TRX | SCAN9,3,6  |                              | R100 |
| 142900H3A   | TRX | SCAN9,3,9  |                              | R100 |
| 143000H4A   | TRX | SCAN9,3,12 |                              | R100 |
| 143100H5A   | TRX | SCAN9,3,15 |                              | R100 |
| 143200H6A   | TRX | SCAN9,3,18 |                              | R100 |
| 143300H7A   | TRX | SCAN9,3,21 |                              | R100 |
| 143400H8A   | TRX | SCAN9,3,24 |                              | R100 |
| 143500H9A   | TRX | SCAN9,3,27 |                              | R100 |
| 143600H10A  | TRX | SCAN9,3,30 |                              | R100 |
| 143700H11A  | TRX | SCAN9,3,33 |                              | R100 |
| 143800H12A  | TRX | SCAN9,3,36 |                              | R100 |
| 143900SCAN2 | RPA | SCAN3      | SET COMPARE ADDRESS          | R100 |
| 144000      | SBB | ACC,0,1    |                              | R100 |
| 144100      | RPA | SCAN4      |                              | R100 |

|             |     |                 |                     |      |
|-------------|-----|-----------------|---------------------|------|
| 144200SCAN5 | CLA | TB00A+1         | PICK UP SECOND WORD | R100 |
| 144300SCAN3 | TRC | **              | MATCH REST OF WORD  | R100 |
| 144400      | TRU | SCAN1           | NO MATCH            | R100 |
| 144500      | TRU | SCAN1           | NO MATCH            | R100 |
| 144600SCAN4 | CLA | **              | MATCH-TRANSFER      | R100 |
| 144700      | RPA | PC1             | TO ADDRESS          | R100 |
| ERRB        | MOV | IR1,ERR2        |                     | R100 |
|             | MOV | IR2,IR1         |                     | R100 |
|             | TRU | ERR+1           |                     | R100 |
| ERR         | MOV | IR1,ERR2        |                     | R100 |
| ERR10       | MOV | IR2,ERR3        |                     | R100 |
|             | MOV | PCS,ERR1        |                     | R100 |
| 145800      | MOV | ACC,ERR4        |                     | R100 |
| 145900      | MOV | ORG,ERR5        |                     | R100 |
| 146000      | CLA | IR1             |                     | R100 |
|             | SRL | 12              |                     | R100 |
| 146200      | CLA | LL00C           |                     | R100 |
|             | ADD | =1,0,1          |                     | R100 |
| 146300      | SLL | 12,0,1          |                     | R100 |
|             | STR | TD00A+15        |                     | R100 |
| 146500      | TRL | BSPR            |                     | R100 |
| 146600      | CLA | IR1             |                     | R100 |
| 146700      | LDQ | =/7770000000    |                     | R100 |
| 146800      | SLL | 15,0,1          |                     | R100 |
| 146900      | STR | ERR6            |                     | R100 |
| 147000STR   | MOV | IR3,ERR11       |                     | R100 |
| 147100      | MOV | IR4,ERR12       |                     | R100 |
| 147200      | MOV | =1,IR2          |                     | R100 |
| 147300      | RPT | 13,0,1          |                     | R100 |
|             | MOV | LL00A+1,TD00A+1 |                     | R100 |
| 147500      | MOV | TB00W,IR2       |                     | R100 |
| 147600      | MVZ | IR1             |                     | R100 |
| 147700      | MVZ | IR3             |                     | R100 |
| 147800      | MOV | =5,IR4          |                     | R100 |
| 147900STR1  | CLA | TB00A,1         |                     | R100 |
|             | STR | TD00A+16,1      |                     | R100 |
|             |     |                 |                     | A-44 |

|        |     |            |      |
|--------|-----|------------|------|
| 148100 | TRX | *+1,3,1    | R100 |
| 148200 | TRX | STR1,1,1   | R100 |
| 148300 | CLA | ZERO       | R100 |
| 148500 | STR | TD00A+16,3 | R100 |
|        | TRX | *-1,3,1    |      |
|        | MOV | ERR2,IR1   |      |
| 148800 | MOV | ERR11,IR3  | R100 |
| 148900 | MOV | ERR12,IR4  | R100 |
| 149000 | MOV | ERR3,IR2   | R100 |
| 149100 | MOV | ERR4,ACC   | R100 |
| 149200 | MOV | ERR5,GRG   | R100 |
| 149300 | MOV | ERR1,PCT   | R100 |
| 149400 | HLT |            | R100 |
| 149500 | HLT |            | R100 |
| 149600 | HLT |            | R100 |
| 149700 | HLT |            | R100 |
| 149800 | HLT |            | R100 |
| 149900 | HLT |            | R100 |
| 149920 | HLT |            | R100 |
| 149925 | HLT |            | R100 |
|        | MOV | TC00W,IR2  |      |
|        | STR | TC00A,2    |      |
|        | ADB | TC00W,,1   |      |
|        | TRC | =100       |      |
|        | TRS |            |      |
| 150400 | TRU | *+1        | R100 |
| 150500 | MOV | *-2,BSPR   | R100 |
| 150600 | TRS |            | R100 |
| 150700 | TRL | ERR,1,FG1  | R100 |
|        | TRL | ERR,1,FG5  | R100 |
|        | LSX | 0,3,/12    |      |
|        | CLA | TB00A      | R100 |
|        | TRC | ID,3       | R100 |
|        | TRU | W2         | R100 |
|        | TRU | W2         | R100 |
|        | CLA | ID+1,3     | R100 |
|        |     |            | A-45 |

IR3  
IR4

ERROR ROUTINE  
FIND NEXT WORD IN

|     | TRN          | W8 | W6 | W5 | W2 | HFC3 | ALIGN |
|-----|--------------|----|----|----|----|------|-------|
| RPA | PCT          |    |    |    |    |      |       |
| SBB | W6           |    |    |    |    |      |       |
| RPA | ACC,0,1      |    |    |    |    |      |       |
| CLA | W5           |    |    |    |    |      |       |
| TRC | TB00A+1      |    |    |    |    |      |       |
| TRU | **           |    |    |    |    |      |       |
| CLA | W2           |    |    |    |    |      |       |
| RPA | **           |    |    |    |    |      |       |
| TRX | PCT          |    |    |    |    |      |       |
| TRL | W3,3,3       |    |    |    |    |      |       |
| TRU | A            |    |    |    |    |      |       |
| HLT | W4           |    |    |    |    |      |       |
| CLA | HFC3         |    |    |    |    |      |       |
| RPA | IR4          |    |    |    |    |      |       |
| LOD | OUACM        |    |    |    |    |      |       |
| LXS | OUACM+1      |    |    |    |    |      |       |
| MOV | IR3,,IR1     |    |    |    |    |      |       |
| LXS | INTBF+6,2,4  |    |    |    |    |      |       |
| CLA | IR3,IRSAV    |    |    |    |    |      |       |
| LOD | 4,3,3        |    |    |    |    |      |       |
| SLL | 0,1          |    |    |    |    |      |       |
| STR | 1,1,QRG      |    |    |    |    |      |       |
| CLA | 0,3,3        |    |    |    |    |      |       |
| LOD | 0,2          |    |    |    |    |      |       |
| SLL | 1,1          |    |    |    |    |      |       |
| STR | 2,1,QRG      |    |    |    |    |      |       |
| CLA | 0,3,3        |    |    |    |    |      |       |
| LGM | 1,2          |    |    |    |    |      |       |
| STR | QRG          |    |    |    |    |      |       |
| ADB | MASKA        |    |    |    |    |      |       |
| ADB | 2,2          |    |    |    |    |      |       |
| TRX | IR1,,2       |    |    |    |    |      |       |
|     | IR2,,3       |    |    |    |    |      |       |
|     | ALIGN+1,3,12 |    |    |    |    |      |       |

**A-46**

|     |           |                                      |
|-----|-----------|--------------------------------------|
| SBB | IRSAV,0,1 | DECRIMENT ROW GROUP COUNT            |
| TRZ | *+3       | IF COUNT ZERO, PROCEED TO NEXT SECTN |
| ADB | IR1,0,1   | SET INPUT LOCATION FOR NEXT GROUP    |
| TRU | ALIGN     | ALIGN NEXT ROW GROUP                 |
| LOD | OUACM,IR2 | OUTPUT BUFFER LOCATION FOR CLEAR     |
| RPT | 13,0,1    | CLEAR OUTPUT                         |
| STR | 0,2       | BLOCK AREA                           |
| LXS | INTBF,1,3 | 7-9 AND ZONE ZERO INDEX              |
| CLA | 6,1       | FORM LOGICAL SUM                     |
| LGA | 9,1       | OF BITS IN 12 AND 11 ROWS            |
| LGM | 12,1      | MASK BY BITS IN 0 ROW                |
| STR | 0,1       | SET ZONE ZERO BITS IN 4 ROW          |
| TRZ | *+4       | IF ZERO, PROCEED TO 7-9 CHECK        |
| LGM | ACC       | NEXATE WORD FROM 14 ROW              |
| LGM | 12,1      | REMOVE ZERO PUNCH BITS FOR ZONE 0    |
| STR | 12,1      | AND REPLACE IN BUFFER                |
| CLA | 33,1      | GET 7 ROW BITS                       |
| LGM | 39,1      | MASK OFF BITS NOT CONTAINED IN 9 ROW |
| STR | 3,1       | SET LOGICAL PRODUCT IN 13 ROW        |
| TRZ | *+4       | SKIP BIT REMOVAL IF ZERO 13 ROW      |
| LGM | ACC       | COMPLIMENT MASK IN ACC               |
| LGM | 39,1      | REMOVE 13 ROW BITS FROM 9 ROW        |
| STR | 39,1      | RESTORE 9 ROW WORD                   |
| TRX | ROWCM,1,1 | PROCESS NEXT WORDS OF ROWS           |
| LXS | INTBF,1,3 | SET BUFFER LOCATION AND ROW WORD CNT |
| LXS | 0,3,14    | SET ROW COUNT                        |
| CLA | 0,1       | GET WORD OF ROW                      |
| TRZ | *+2       | INDEX FOR NEXT WORD OF ROW IF 0      |
| TRU | *+6       | PROCESS WORD AND REMAINDER OF ROW    |
| TRX | *-3,1,1   | INDEX BUFFER LOCATION AND ROW WORD   |
| ADB | IR1,0,1   | ADJUST BUFFER LOC FOR TRX FALL       |
| LOD | CONS3,IR2 | RESET ROW WORD COUNT                 |
| TRX | RWCHK,3,0 | INDEX COUNT AND PROCESS NEXT ROW     |
| TRU | CDCVT     | TRANSFER TO CODE CONVERSION IF FALL  |
| MOV | IR4,IRSAV | SAVE ROW NUMBER INDICATOR            |
| LOD | CHPRO,IR3 | INDEX FOR ROW VALUE SET-UP           |

|       |               |                                      |
|-------|---------------|--------------------------------------|
| CLA   | VLROW-1,4     | GET EFFECTIVE VALUE FOR ROW          |
| STR   | OPTAB-1,3     | SET CHARACTER VALUE IN TABLE         |
| TRX   | 5             | POSITION FOR NEXT CHARACTER          |
| WDPRO | **2,2,0       | INDEX SET-UP LOOP                    |
| CLA   | 0,1           | GET INPUT WORD FROM BUFFER           |
| TRZ   | WDZRO         | IF ZERO, SKIP WORD PROCESS           |
| ADB   | ZERO,0        | CLEAR ACC AND PUT WORD IN ORG        |
| LOD   | WDPLA-1,2,IR4 | SET OUTPUT WORD INDEX                |
| LOD   | **1,IR3       | WORD CHARACTER INDEX                 |
| SLL   | 6,3           | SELECT CHARACTER                     |
| TRZ   | CHIDX         | SKIP CHARACTER IF ZERO               |
| SHL   | 31,1          | LEFT JUSTIFY AND SHIFT OFF 1 BIT     |
| SNS   | **2,0VA       | SKIP IF ZERO HIGH ORDER BIT          |
| LGA   | OPTAB+5       | INSERT HIGH ORDER CHARACTER          |
| SHL   | 1,1           | SHIFT OFF NEXT BIT                   |
| SNS   | **2,0VA       | SKIP IF 0 SECOND BIT                 |
| LGA   | OPTAB+4       | INSERT SECOND POSITION CHARACTER     |
| SHL   | 1,1           | SHIFT OFF THIRD BIT                  |
| SNS   | **2,0VA       | SKIP IF 0 BIT                        |
| LGA   | OPTAB+3       | INSERT THIRD POSITION BIT            |
| SHL   | 1,1           | SHIFT OFF FOURTH BIT                 |
| SNS   | **2,0VA       | SKIP IF BIT IS ZERO                  |
| LGA   | OPTAB+2       | INSERT FOURTH CHARACTER              |
| SHL   | 1,1           | SHIFT OFF FIFTH BIT                  |
| SNS   | **2,0VA       | SKIP IF BIT IS 0                     |
| LGA   | OPTAB+1       | INSERT FIFTH CHARACTER               |
| SHL   | 1,1           | SHIFT OFF SIXTH BIT                  |
| SNS   | **2,0VA       | SKIP IF ZERO SIXTH BIT               |
| LGA   | OPTAB         | INSERT SIXTH CHARACTER               |
| QUACM | **4,3         | ACCUMULATE INTO OUTPUT AREA          |
| ADD   | **4           | RESTORE OUTPUT WORD                  |
| STR   | ORG           | GET REMAINING WORD FROM ORG          |
| CHIDX | IR4,1         | INCREASE OUTPUT INDEX AND WORD TO Q  |
| ADB   | ZERO          | CLEAR ACCUMULATOR                    |
| CLA   | CHPRO,2,0     | INDEX AND PROCESS REMAINING CHARACTE |
| TRX   | WDPRO,1,1     | INDEX AND PROCESS ROW WORDS          |
| WDZRO |               |                                      |

|              |                                                         |                                      |      |
|--------------|---------------------------------------------------------|--------------------------------------|------|
| ADB          | IR1,,1                                                  | INCREASE INPUT INDEX                 |      |
| LOD          | IRSAV,,IR4                                              | RESTORE ROW INDEX                    |      |
| LXS          | 3,2,0                                                   | RESTORE ROW WORD INDEX               |      |
| TRX          | RWCHK,3,0                                               | INDEX AND PROCESS NEXT ROW           |      |
| CDCVT        | QUACM,,IR1                                              | BUFFER LOCATION FOR CONVERSION       |      |
| LOD          | 14,2,0                                                  | OUTPUT BUFFER WORD INDEX             |      |
| LXS          | MASKB,,QRG                                              | SET MASK FOR CHARACTER LOAD          |      |
| LOD          | 6,3,0                                                   | SET WORD CHARACTER CONVERSION INDEX  |      |
| LXS          | 0,1                                                     | GET WORD FROM OUTPUT BUFFER          |      |
| CLA          | 6                                                       | POSITION CHARACTER IN LOW ORDER ACC  |      |
| CYS          | IR4                                                     | PLACE CHARACTER IN INDEX FOR PICK-UP |      |
| MSK          | MASKC                                                   | CLEAR CHARACTER POSITION IN WORD     |      |
| LGM          | CVTAB,4                                                 | INSERT CONVERSION CHARACTER          |      |
| LGA          | WDCVT+2,2,0                                             | INDEX AND PROCESS NEXT CHARACTER IN  |      |
| TRX          | 0,1                                                     | SET CONVERTED WORD IN OUTPUT BLOCK   |      |
| STR          | WDCVT,1,1                                               | INDEX AND PROCESS NEXT OUTPUT WORD   |      |
| TRX          |                                                         |                                      |      |
| TRS          |                                                         |                                      |      |
| OCT          | 5,61,62,63,64,65,66,67,70,71,73,44,41,0,4,55,42,6,7,10  |                                      |      |
| OCT          | 11,12,13,14,15,16,45,75,40,1,57,52,41,17,20,21,22,23,24 |                                      |      |
| OCT          | 25,26,27,43,47,50,2,77,53,60,74,30,31,32,33,34,35,36,37 |                                      |      |
| OCT          | 72,56,51,3,76,54                                        |                                      |      |
| OCT          | 776000000000                                            |                                      |      |
| OCT          | 77                                                      |                                      |      |
| OCT          | 777777777700                                            |                                      |      |
| OCT          | 6/0,6/9,31/0,6/8,31/0,6/7,31/0,6/6,31/0,6/5,31/0,6/4    |                                      |      |
| VFD          | 6/0,6/3,31/0,6/2,31/0,6/1,31/0,06/60,31/0,06/40         |                                      |      |
| VFD          | 6/0,06/20,31/0,06/3,31/0,06/12                          |                                      |      |
| DEC          | 12,6,0                                                  |                                      |      |
| BSS          | 1                                                       |                                      |      |
| BSS          | 6                                                       |                                      |      |
| BSS          | 60                                                      |                                      |      |
| 150800LCTBLE | A1B-A1A                                                 |                                      | R100 |
| 150900XA     | ICA-A1B                                                 |                                      | R100 |
| 151000XB     | A1D-1CA                                                 |                                      | R100 |
| 151100XC     | A1E-A1D                                                 |                                      | R100 |
| 151200XD     | A1F-A1E                                                 |                                      | R100 |
| 151300XE     |                                                         |                                      | A-49 |
| WDPLA        |                                                         |                                      |      |
| IRSAV        |                                                         |                                      |      |
| OPTAB        |                                                         |                                      |      |



|             |     |      |      |
|-------------|-----|------|------|
| 15500004F0C | DEF | 114  | R100 |
| 15510005F0C | DEF | 115  | R100 |
| 15520006F0C | DEF | 116  | R100 |
| 15530007F0C | DEF | 117  | R100 |
| 15540008F0C | DEF | 120  | R100 |
| 15550009F0C | DEF | 121  | R100 |
| IDA         | SYN | W    | R100 |
| IDE         | SYN | W7   | R100 |
| N5          | SYN | SCAN | R100 |
| 155600A2B   | SYN | A3A  | R100 |
| 155700A4D   | SYN | A3A  | R100 |
| 155800A7E   | SYN | A7C  | R100 |
| 155900A8E   | SYN | A2A  | R100 |
| 156000A3I   | SYN | A7C  | R100 |
| 156100A3L   | SYN | A3F  | R100 |
| 156200A4L   | SYN | A3F  | R100 |
| 156300A5L   | SYN | A5D  | R100 |
| 156400A6L   | SYN | A7D  | R100 |
| 156500A4M   | SYN | A2A  | R100 |
| 156600A4O   | SYN | A6D  | R100 |
| 156700A3P   | SYN | A7C  | R100 |
| 156800A4P   | SYN | A7D  | R100 |
| 156900A5P   | SYN | A2N  | R100 |
| 157000A9P   | SYN | A6D  | R100 |
| 157100A3R   | SYN | A3A  | R100 |
| 157200A4R   | SYN | A2A  | R100 |
| 157300A8R   | SYN | A3A  | R100 |
| 157400A9R   | SYN | A7C  | R100 |
| 157500A10R  | SYN | A6P  | R100 |
| 157600A11R  | SYN | A6D  | R100 |
| 157700A6S   | SYN | A2O  | R100 |
| 157800A5T   | SYN | A2G  | R100 |
| 157900A2U   | SYN | A11C | R100 |
| 158000A3U   | SYN | A9L  | R100 |
| 158100A4U   | SYN | A10L | R100 |
| 158200A2V   | SYN | A3F  | R100 |

A-51

R100

END 1

158800

A-52

## APPENDIX B

### DISTRIBUTION LIST SECOND QUARTERLY REPORT

DA 36-039-sc-89231

|                                                                                                                | <u>No. Copies</u> |
|----------------------------------------------------------------------------------------------------------------|-------------------|
| OASD (R&E) Rm 3E1065<br>ATTN: Technical Library<br>The Pentagon<br>Washington 25, D.C.                         | 1                 |
| Chief of Research and Development<br>OCS, Department of the Army<br>Washington 25, D.C.                        | 1                 |
| Chief Signal Officer<br>ATTN: SIGRD<br>Department of the Army<br>Washington 25, D.C.                           | 1                 |
| Director, U.S. Naval Research Laboratory<br>ATTN: Code 2027<br>Department of the Army<br>Washington 25, D.C.   | 1                 |
| Commanding Officer and Director<br>U.S. Navy Electronics Laboratory<br>San Diego 52, California                | 1                 |
| Commander<br>Aeronautical Systems Division<br>ATTN: ASPRDL<br>Wright-Patterson Air Force Base, Ohio            | 1                 |
| Commander, Rome Air Development Center<br>ATTN: RAALD<br>Griffiss Air Force Base, New York                     | 1                 |
| Commanding General<br>U.S. Army Electronic Proving Ground<br>ATTN: Technical Library<br>Fort Huachuca, Arizona | 1                 |

|                                                                                                                                                                                                             | <u>No. Copies</u> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>Commanding General</b><br><b>U.S. Army Electronic Proving Ground</b><br><b>ATTN: ADP Department</b><br><b>Fort Huachuca, Arizona</b>                                                                     | 1                 |
| <b>Commander, Armed Services Technical Information Agency</b><br><b>ATTN: TIPCR</b><br><b>Arlington Hall Station</b><br><b>Arlington 12, Virginia</b>                                                       | 10                |
| <b>Chief, U.S. Army Security</b><br><b>Arlington Hall Station</b><br><b>Arlington 12, Virginia</b>                                                                                                          | 2                 |
| <b>Deputy President</b><br><b>U.S. Army Security Agency Board</b><br><b>Arlington Hall Station</b><br><b>Arlington 12, Virginia</b>                                                                         | 1                 |
| <b>Commanding Officer</b><br><b>U.S. Army Electronic Material Support Agency</b><br><b>ATTN: SELMS-ADJ</b><br><b>Fort Monmouth, New Jersey</b>                                                              | 1                 |
| <b>Corps of Engineers Liaison Office</b><br><b>U.S. Army Electronic Research &amp; Development Laboratory</b><br><b>Fort Monmouth, New Jersey</b>                                                           | 1                 |
| <b>Commanding Officer</b><br><b>U.S. Army Electronic Research &amp; Development Laboratory</b><br><b>Logistics Division</b><br><b>ATTN: Mr. N.J. Taupeka, SELRA/NPE</b><br><b>Fort Monmouth, New Jersey</b> | 3                 |
| <b>Commanding Officer</b><br><b>U.S. Army Electronic Research &amp; Development Laboratory</b><br><b>Data Equipment Branch, Data Processing Facilities Div.</b><br><b>Fort Monmouth, New Jersey</b>         | 1                 |
| <b>Commanding Officer</b><br><b>U.S. Army Electronic Research &amp; Development Laboratory</b><br><b>ATTN: Director of Engineering</b><br><b>Fort Monmouth, New Jersey</b>                                  | 1                 |
| <b>Commanding Officer</b><br><b>U.S. Army Electronic Research &amp; Development Laboratory</b><br><b>ATTN: Technical Documents Center</b><br><b>Fort Monmouth, New Jersey</b>                               | 1                 |

No. Copies

Commanding Officer  
U.S. Army Electronic Research & Development Laboratory  
ATTN: Technical Information Div.  
Fort Monmouth, New Jersey

3

ADPS Committee  
Officer's Department  
U.S. Army Signal School  
Fort Monmouth, New Jersey

1

Information Processing Branch  
Dept. of Specialist Training  
U.S. Army Signal School  
Fort Monmouth, New Jersey

1

MOBIDIC Project Office  
CCIS-70 PMSO  
U.S.A. Electronics Command  
Fort Monmouth, New Jersey

1

Ordnance Stock Control Agency  
APO 58  
ATTN: Maj. C.S. Moody  
New York, New York

1

7th Army Stock Control Center  
APO 872  
ATTN: Capt. D. Lasher  
New York, New York

1

Philco Corporation  
ATTN: Mr. S. Berkowitz  
3900 Welsh Road  
Willow Grove, Pa.

1

RCA Surface Comm. Division  
ATTN: Mr. A. Coleman  
Camden, New Jersey

1