Computer Program and User Documentation
Medical Data Input System

TN514

Submitted to
National Aeronautics and Space Administration
Manned Spacecraft Center
Information Systems Branch
Houston, Texas

Philco-Ford Corporation
Houston Operation
COMPUTER PROGRAM AND USER DOCUMENTATION

MEDICAL DATA INPUT SYSTEM

Contract NAS 9-11579

Submitted to the

BIOMEDICAL DATA SYSTEMS OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
INFORMATION SYSTEMS BRANCH
Houston, Texas

Submitted by:

J. Anderson, Cognizant Engineer

Approved by:

M. Brinkman, Supervisor
Data Processing Applications Section

N. Hines, Manager
Support Engineering Department

G. Straty, Manager
System Engineering Activity

PHILCO-FORD CORPORATION
AEROSPACE AND DEFENSE SYSTEMS OPERATIONS
WDL DIVISION
HOUSTON OPERATION
1002 GEMINI AVENUE
HOUSTON, TEXAS

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 SCOPE</td>
<td>1-1</td>
</tr>
<tr>
<td>2.0 BIOMEDICAL INFORMATION SYSTEM OVERVIEW</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1 OVERVIEW SUMMARY</td>
<td>2-6</td>
</tr>
<tr>
<td>3.0 MDIS SYSTEM</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.1 Background</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.2 Functions of the System</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2 TECHNICAL SPECIFICATIONS</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.1 System Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.2 Input</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.3 Processing</td>
<td>3-4</td>
</tr>
<tr>
<td>3.2.4 Output</td>
<td>3-6</td>
</tr>
<tr>
<td>3.2.5 Buffers and Tables</td>
<td>3-7</td>
</tr>
<tr>
<td>3.2.6 System Flow</td>
<td>3-8</td>
</tr>
<tr>
<td>3.2.7 Hardware Configuration</td>
<td>3-12</td>
</tr>
<tr>
<td>3.2.8 System Block Diagram</td>
<td>3-12</td>
</tr>
<tr>
<td>3.3 SUBROUTINES</td>
<td>3-14</td>
</tr>
<tr>
<td>3.3.1 AIOO - Alpha-to-Integer conversion</td>
<td>3-14</td>
</tr>
<tr>
<td>3.3.2 BDOO - Build Display</td>
<td>3-19</td>
</tr>
<tr>
<td>3.3.3 BM00 - Build a MEDATA record</td>
<td>3-40</td>
</tr>
<tr>
<td>3.3.4 XL00 - Extra Line Generation</td>
<td>3-85</td>
</tr>
<tr>
<td>3.3.5 CP00 - Control Program</td>
<td>3-98</td>
</tr>
<tr>
<td>3.3.6 D000 (D00) - Dump to Teletype</td>
<td>3-114</td>
</tr>
<tr>
<td>3.3.7 HC00 - CRT Input/Output Routine</td>
<td>3-127</td>
</tr>
<tr>
<td>3.3.8 HM00 - Magnetic Tape Handler</td>
<td>3-149</td>
</tr>
<tr>
<td>3.3.9 ID00 - Input Display</td>
<td>3-164</td>
</tr>
<tr>
<td>3.3.10 IN00 - Initialization</td>
<td>3-175</td>
</tr>
<tr>
<td>3.3.11 MD00 - Mag Tape Output Routine</td>
<td>3-183</td>
</tr>
<tr>
<td>3.3.12 OD00 - Output Display</td>
<td>3-187</td>
</tr>
<tr>
<td>3.3.13 OZKA - Print Buffer on TTY</td>
<td>3-195</td>
</tr>
<tr>
<td>3.3.14 OZKC - Output Character to TTY</td>
<td>3-200</td>
</tr>
<tr>
<td>3.3.15 OZKR - Output Carriage Return and Line Feed</td>
<td>3-204</td>
</tr>
<tr>
<td>3.3.16 PINT - Power Fail - Restart</td>
<td>3-208</td>
</tr>
<tr>
<td>3.3.17 SC00 - Set Cursor</td>
<td>3-212</td>
</tr>
<tr>
<td>4.0 PROGRAM UTILIZATION</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1 COMPUTER OPERATOR INSTRUCTIONS</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2 OPERATIONAL PROCEDURES</td>
<td>4-2</td>
</tr>
<tr>
<td>4.3 INPUT DESCRIPTION</td>
<td>4-5</td>
</tr>
<tr>
<td>4.4 OUTPUT DESCRIPTION</td>
<td>4-5</td>
</tr>
<tr>
<td>4.4.1 Reports</td>
<td>4-5</td>
</tr>
<tr>
<td>4.4.2 Tapes</td>
<td>4-5</td>
</tr>
<tr>
<td>4.4.3 Messages</td>
<td>4-6</td>
</tr>
<tr>
<td>4.5 RESTRICTIONS</td>
<td>4-8</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX A (Input and Output Tapes Descriptions)</td>
<td>A-1</td>
</tr>
<tr>
<td>APPENDIX B (CRT Data Format)</td>
<td>B-1</td>
</tr>
<tr>
<td>APPENDIX C (LTBS Line Table Format)</td>
<td>C-1</td>
</tr>
<tr>
<td>APPENDIX D (Sample Input)</td>
<td>D-1</td>
</tr>
<tr>
<td>APPENDIX E (Sample Output)</td>
<td>E-1</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>2-1</td>
<td>Varian/MEDATA Storage and Retrieval System</td>
</tr>
<tr>
<td>2-2</td>
<td>Medical Data Input System</td>
</tr>
<tr>
<td>2-3</td>
<td>Medical Data Update System</td>
</tr>
<tr>
<td>2-4</td>
<td>Medical Data Tape Retrieval System (MDTRS)</td>
</tr>
<tr>
<td>3-1</td>
<td>MDIS System Block Diagram</td>
</tr>
<tr>
<td>3-2</td>
<td>Sample Output Buffer</td>
</tr>
<tr>
<td>3-3</td>
<td>Cursor Positioning Criteria</td>
</tr>
<tr>
<td>A-1</td>
<td>Input File Structure</td>
</tr>
<tr>
<td>A-2</td>
<td>Input Record Structure</td>
</tr>
<tr>
<td>A-3</td>
<td>Identification Section Description</td>
</tr>
<tr>
<td>A-4</td>
<td>Body Section Description</td>
</tr>
<tr>
<td>A-5</td>
<td>Output File Structure</td>
</tr>
<tr>
<td>A-6</td>
<td>Output Record Structure</td>
</tr>
<tr>
<td>A-7</td>
<td>Identification Section Description</td>
</tr>
<tr>
<td>A-8</td>
<td>Body Section Description</td>
</tr>
</tbody>
</table>
1.0 SCOPE

The purpose of this volume is to provide the reader with several levels of documentation for this program module of the NASA Medical Directorate Mini-computer Storage and Retrieval System.

The Biomedical Information System Overview (Section 2) describes some of the reasons for the development of the mini-computer storage and retrieval system. It briefly describes all of the program modules which constitute the system.

The General Specifications Section (Paragraph 3.1) describes the purpose and function of the specific program module documented in this volume.

The Technical Specifications (Paragraph 3.2) is oriented to the programmer. It is a technical discussion of the same processing described in general terms in the previous section, but is a high enough level as not to be redundant with the very detailed analysis described in the Subroutine Section 3.3.

The Subroutine Section (Paragraph 3.3) describes each subroutine in enough detail to permit an in-depth understanding of the routines and facilitate program modifications.

The Program Utilization Section (Paragraph 4.0) may be used as a "Users Guide" and is as non-technical as possible.
To eliminate unnecessary reproduction, the program listings are maintained in a separate document which may be obtained from the VARIAN library of computer programs currently maintained in Building 32 at NASA MSC. In addition, another document is to be developed for the Medical Directorate management. The document will describe the mini-computer system on a higher functional level and will illustrate how the mini-computer storage and retrieval system interfaces with the total directorate data management plan.
For several years, the NASA Medical Directorate has been developing a Medical Information Management System (MEDATA). The System, as implemented at NASA, utilized an off-line IBM 1050 for preparation of data input via paper tape and required card deck runs for retrieval of data from the data files. A new phase of development has been implemented utilizing remote terminals and mini-computers. This section describes the development rationale of the remote terminal and mini-computer approach.

Several features of the old MEDATA system presented serious difficulties to the system user and made the system unresponsive. These deficiencies in the system included:

- Input data prepared on punched paper tape
- No online data input capability with the data base (i.e., no remote data entry)
- Data corrections via punched paper tape
- Preparation of retrieval requests via a punched card system
- 48-hour turnaround time required for retrieval outputs.

To improve the overall responsiveness of the system and eliminate these deficiencies, Philco-Ford developed the Varian/MEDATA Storage and Retrieval System (VMSARS) which utilizes the Varian 620I computer systems in Building 32 or 37.
VMSARS consists of the Medical Data Input System (MDIS), the Medical Data Update System (MDUS), and the Medical Data Tape Retrieval System (MDTRS). The Medical Data Input System (MDIS) is used to input data via a CRT. If updates or reviews are desired, the output of MDIS may be updated on the CRT by the Medical Data Update System (MDUS). The data tape (MDIS or MDUS output) is entered into the CAAD MEDATA system where it is sorted and merged into the appropriate Medical Data File. The updated CAAD MEDATA Medical Data Tape File is sent back to the Varian computer system and used as input for the Medical Data Tape Retrieval System (MDTRS) which processes data retrieval requests from remote terminals. The primary advantages of the VMSARS are as follows:

- Use of state-of-the-art input devices such as CRT's and portable acoustic-coupled teleprocessing terminals. (CRT's operate at 2400 bits per second on the telephone lines.)

- Elimination of paper tapes from the system. (Data storage is on magnetic tape.)

- Online data input from remote input station via the telephone lines.

- Error correction capabilities on CRT. (Limited error checking is performed by the computer.)

- Update capabilities on Varian system before data is entered into the CAAD MEDATA system.
Ease of creation and updating of background forms.

- Capability of transmitting MEDATA retrievals over telephone lines to CRT or TTY.

- Capability of performing online retrievals.

VMSARS was used for the Flight Crew Health Stabilization Program for three months preceding the launch of Apollo 14 and Apollo 15. A Computer Communications Inc. (CCI) CRT and keyboard and an Execuport Typewriter were installed in the surveillance command post at KSC, and a surveillance master file was created. The VMSARS was used to collect data, perform updates, and retrieve data from the surveillance master file.

The overall system, including the CAAD MEDATA system interface, is illustrated in Figure 2-1. The individual programs are described as follows:

MEDICAL DATA INPUT SYSTEM

The Medical Data Input System (MDIS) is designed to collect data from a CRT input station and store the data on a magnetic tape. The medical questionnaire forms are maintained on magnetic tape and are read into computer memory at run time. The user selects the appropriate form to be displayed on the CRT. As each form is completed, the data is stored on magnetic tape. This tape may be further updated, transmitted to another terminal, or input to the 1108 MEDATA system. Hard copies of any form are produced at the
user's request. The system requires only one tape unit and operates either via the phone lines or directly online with the computer.
MEDICAL DATA UPDATE SYSTEM

The Medical Data Update System (MDUS) updates any tape created by MDIS. Two tape-drives, a CRT, and printer are required. The old data is read from the MDIS output tape and displayed on the CRT. Changes may be made to the data on the CRT; comments or recommendations may be added to the record, and the new updated record written to the update tape. The updated tape may be either transmitted to another terminal or input to the MEDATA system, or both. Hardcopies of the records are produced if requested by the user. Refer to Figure 2-3 for an illustration of the MDUS components.
MEDICAL DATA TAPE RETRIEVAL SYSTEM (MDTRS)

MDTRS permits the user to make data retrievals from the MEDATA master tapes created by the 1108 MEDATA system. The 1108 system builds the MEDATA master tape from several sources, one of which is the MDIS or MDUS created input tapes. MDTRS outputs preprogrammed retrieval questions from the Varian computer to the requester's CRT or typewriter. The user builds his retrieval request by answering these questions. When the retrieval request is complete, MDTRS searches the MEDATA master tape for the data, formats the selected data for output, and outputs the data to the terminal. The CRT may operate on a private telephone line at 2000 bits per second and the typewriter operates on any commercial telephone line at 300 bits per second. Refer to Figure 2-4 for an illustration of MDTRS components.

2.1 OVERVIEW SUMMARY

In summary, the Varian/MEDATA Storage and Retrieval System provides the user with an online input and retrieval capability previously unavailable. The response time is significantly improved over the old paper tape system. There are still two weak points in the system due to hardware constraints. These are (1) the requirement to update the MEDATA master file on the CAAD 1108 computer system, and (2) the use of tape instead of disk for storage of the data base. A more comprehensive storage and retrieval system is being designed to operate on a Varian 620 mini-computer system utilizing a disk memory storage device and an input/output multiplexing device. The new system will permit immediate update of files, faster response for retrieval requests and multiple terminal users operating simultaneously.
Figure 2-1 Varian/MEDATA Storage and Retrieval System
Figure 2-2 Medical Data Input System
Figure 2-3 Medical Data Update System
Figure 2-4 Medical Data Tape Retrieval System (MDTRS)

2-10
This document describes in detail one of the three components of VMSARS, the Medical Data Update System (MDUS).
3.0 MDIS SYSTEM

3.1 GENERAL SPECIFICATIONS

3.1.1 Background

Beginning in the fall of 1969, under a contract to NASA Manned Spacecraft Center, a team of programmers and analysts designed and implemented a storage and retrieval system at M. D. Anderson Hospital, Houston, Texas. The first demonstration of the system, called MEDATA, was done on an IBM 1401. The data base was created using card inputs, and any requests from the data base were made via the computer console typewriter. Operation of the system was viewed and accepted by MSC, and work began immediately for a similar more complete system on an SDS 930 computer. The new system was to accept input both from card and from paper tape punched on an IBM 1050 terminal. Any requests could again be made from the console, but capability was added to accept punched card requests. Once implemented, MEDATA was transferred to an SDS Sigma 5, and began its operation for NASA.

Before the fall of 1969, work had begun on the implementation of MEDATA on the Univac 1108. This system would have greater processing capabilities, but the principle means of input to the data base was still to be card and paper tape. In May of 1970, Philco-Ford Corporation, under contract to NASA, began the design and implementation of a more sophisticated means of input. This system, called the Medical Data Input System (MDIS), was based on a hardware configuration consisting of a Varian 620i computer, a Cathode Ray Tube (CRT) display, and a magnetic tape drive. The data
was transferred from the CRT display to the computer to the magnetic tape. This tape was then used as input to the MEDATA system on the Univac 1108. The advantages of this system were obvious to the users. They had error correction capabilities that they had never before been allowed. They no longer had to contend with cumbersome paper tapes. From an operational viewpoint, the new system provided a much more efficient means of input.

3.1.2 Functions of the System

3.1.2.1 Input
There are two types of input in MDIS: the forms background input data, and user input medical information. The background data is the same information that a medical examiner sees on a blank collection form. This blank form is completed by the examiner, and then referenced by a MDIS user when he inputs to the CRT display that information collected by the examiner. All of the possible background forms can be grouped onto magnetic tapes, so that at any time a particular set of forms is desired, they may be selected by merely using the correct tape.

3.1.2.2 Processing
Once the user has selected the correct background forms tape to be used, these backgrounds are read into the computer. An index table is created so that any time a user desires a particular form, the table can be referenced to locate the form. Once found, the form is output in segments. These segments, called "pages", are necessary because one form may exceed the size of the CRT screen. As each page is completed the data is input
and saved until the entire form has been accounted for. The data is then recorded onto a magnetic tape and saved.

3.1.2.3 Output

The data that is input from the CRT display is stored and, along with its background data, is output to tape. This tape will eventually be used as an input tape to the Master File maintained by the MEDATA system. Should the user desire a hardcopy of any form, the data will be output to the local teletype.

3.2 TECHNICAL SPECIFICATIONS

3.2.1 System Description

MDIS is an alternative system for input to MEDATA. Prior to its implementation all inputting was done through a paper tape oriented IBM 1050 Data Communications System. The maximum speed of this system was 15 characters per second. With MDIS, transfer rates are greatly increased and typographical errors can easily be corrected.

3.2.2 Input

The input to MDIS is either initialization or user-input data. Initialization data consists of a magnetic tape of the background forms. The format of these forms is strictly governed to such an extent that if a mistake is made in creating a background, that form may be impossible to display. Appendix A contains a detailed description of this format.
User-input data is free form. Transmission of data from the CRT display to the computer is in block mode, which means that an entire screen is transmitted upon command as opposed to one character at a time. An example of CRT input is found in Appendix D.

In Section 3.3.5 the user options that may be input are described.

3.2.3 Processing

Eight functions are performed in the processing of MDIS. They are:

- Initialize background forms
- Select a form
- Build a page of the form
- Output a page to the CRT display
- Input a page from the CRT display
- Build a tape record
- Output a tape record
- Terminate processing

Initialization consists of reading the background forms from tape and storing them in adjacent areas in the tape input buffer, MTNS$. Since each form is one tape record, an index table with the beginning buffer locations of each form is easily created at this time. After this is done, control is transferred to CP00, the control program, which handles all interface between the processing routines and the user at the CRT display.
The user selects a form by keying in a number that is associated with this form. The index table is referenced to determine the location of the requested form, and processing begins. BDOO now formats the first segment of the form for output. The next time BDOO is called it will process the next segment of the form, and so on, until the control program has determined that the last page has been output. After the output to the CRT of each segment of the background, at the user's command, all data on the screen is input. This data is formatted for output to tape, but is not output at this time. It is not until all segments of the form have been viewed by the user and processed by BMOO, that the data is output to tape.

When the user has completed all of his inputting for this session, he can request termination. At this point a special software end-of-file record is written to tape (see Appendix A), and at the local teletype a message is output which specifies the number of records written to the output tape.

The user has five options that he may request from his terminal. They are:

- Position cursor
- Redisplay background for current page
- Redisplay background for the first page of current form
- Display a new form
- List last record at teletype
Each of these options is performed by calling special routines or by using special calling sequences to normal processing routines. Routine SCOO positions the CRT display cursor to the beginning of the next answer area following background data.

To redisplay the background of the current page, the control program calls routine ODOO. Since the CRT display output buffer still has the background stored in it, the routine need only output the same buffer again.

Redisplay of the first page of the current form and display of a new form are both handled similarly. Routine BDOO is called specifying the number of the form to be displayed. The routine references the index table for the location of the requested form and the formats the first segment for output.

The listing of the routine at the teletype is performed by D000. It reads the record last written to tape and formats the data in a manner similar to BDOO.

3.2.4 Output

There are two modes of output in MDIS: tape and teletype. The tape output (the input to MEDATA) is outlined in Appendix A. The teletype output is a user-specified hardcopy of the form.
### 3.2.5 Buffers and Tables

<table>
<thead>
<tr>
<th>Label</th>
<th>Length (Words)</th>
<th>Contents/Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRN$</td>
<td>962</td>
<td>CRT input buffer. Contains the background data and the user data when it is input from the CRT.</td>
</tr>
<tr>
<td>CRO$</td>
<td>974</td>
<td>CRT output buffer. Contains a page of data which is to be output to the CRT.</td>
</tr>
<tr>
<td>DPT$</td>
<td>1</td>
<td>Display output buffer pointer. Used to save the address of the next available location in CRO$ when building a display page.</td>
</tr>
<tr>
<td>FTB$</td>
<td>200</td>
<td>Form table. Contains the beginning location of the form in MTN$</td>
</tr>
<tr>
<td>LTB$</td>
<td>30</td>
<td>Line Table. A table that describes each line of data on the CRT. See Appendix C.</td>
</tr>
<tr>
<td>MPT$</td>
<td>1</td>
<td>Tape output buffer pointer. Used to save the next available location in MTO$ when building a data record.</td>
</tr>
<tr>
<td>MTN$</td>
<td>10,500</td>
<td>Tape input buffer. Storage location for background forms.</td>
</tr>
<tr>
<td>MTO$</td>
<td>2,880</td>
<td>Tape output buffer. Storage location for output data record.</td>
</tr>
</tbody>
</table>
3.2.6 System Flow

MDIS

READ BACKGROUND TAPE

END-OF-FILE

YES

SET INDEX IN FORMS TABLE TO FORM JUST READ

NO

10

10

OZKA

INSTRUCT OPERATOR TO MOUNT SCRATCH TAPE

20
PROCESS PAGE OF INPUT DATA

HAS LAST PAGE BEEN PROCESSED?

YES

NO

BUILD A NEW PAGE FOR OUTPUT

OUTPUT PAGE TO CRT

30

40

45

50

60

3-10
AIDS

60

MDOO
OUTPUT DATA TO TAPE

45

70

STOPZ BOR CHARACTER IN FIRST WORD OF TAPE BUFFER

MDOO
OUTPUT TAPE RECORD

WRITE EOF

REWIND

46000 AIDS
3.2.7 **Hardware Configuration**

Following is a minimum hardware configuration for operation of the MDIS;

1. Varian 620/i computer with 20 K of core memory
2. CCI CC-30 Communication Station with controller (CRT)
3. tape drive
4. teletype
5. 2 Buffer Interlace Controllers (BIC)
6. 1 Priority Interrupt Module with the following interrupts.
   - End of Transmission interrupts on both BIC's.
   - CRT keyboard interrupt

The tape drive and the CRT should be connected to separate BIC's.

3.2.8 **System Block Diagram**

See Figure 3-1.
FIGURE 3.1. MDR SYSTEM BLOCK DIAGRAM

INITIALIZATION

INPUT AND PROCESSING

OUTPUT PROCESSING

INPUT DATA

OUTPUT DATA

BUILD AN OUTPUT PAGE FROM SELECTED BACKGROUND

PROCESS A PAGE AND FOR EVENTUAL OUTPUT

OUTPUT THE COMPLETE FORM TO TELTYPE

OUTPUT THE COMPLETE FORM TO THE CRT

INPUT THE PAGE FROM THE CRT

OUTPUT THE PAGE TO THE CRT

INPUT THE PAGE FROM THE CRT

OUTPUT THE ENTIRE FORM

LISTING OF FORM

COMPLETE FORM

CRT

FORMS BACKGROUND
3.3 SUBROUTINES

3.3.1 A100 - Alpha-to-Integer conversion

3.3.1.1 Purpose

The purpose of this subroutine is to convert a string of ASCII coded digits to their binary value. Specifically, it is used to convert a user-input form number or page number (from the first line of a display page) to a binary value.

3.3.1.2 Technical Description

Three restrictions are placed on the calling routine in order that the converted value be accurate. First, the string of ASCII coded digits must be stored one character per word, right justified. Second, there may not be more than four digits in the string. In the event that there are, the converted value will be -1. The last restriction is that all numbers must be positive, and the string of digits may not be preceded by a sign. There is no validity check of the characters to verify that they are digits. All but the least significant four bits are masked out of the word, and the remaining bits are assumed to be between zero and nine in value. If the characters are digits, this will always be the case, and the conversion will be exact. Otherwise, the results are unpredictable.

The conversion method used is multiplication of each digit by its respective power of ten and sum the products. The resulting sum is stored in the location specified by the calling routine.
3.3.1.2.1 Calling Sequence

CALL AIOO, A, B, C

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>address of the beginning location of the string of ASCII digits</td>
</tr>
<tr>
<td>B</td>
<td>address of a word containing the number of digits in the string</td>
</tr>
<tr>
<td>C</td>
<td>address of the word where the result is to be stored</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>B</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>X</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>May be modified</td>
</tr>
</tbody>
</table>

3.3.1.2.2 General Flow Chart

See Section 3.3.1.5

3.3.1.3 Label Descriptions

3.3.1.3.1 Local

AIFA - a table of the first four powers of ten (1, 10, 100, 1000).
AISA - storage location for beginning address of string of ASCII digits.
AIAN - location for saving the sum of products
3.3.1.3.1 Local (Continued)

AIA - save area for A-Reg
AIB - save area for B-Reg
AIX - save area for X-Reg

3.3.1.3.2 Global

None

3.3.1.3.3 Entry Points

A100 - primary entry point

3.3.1.3.4 External References

None
3.3.1.4 Detailed Flow Chart:

- A100
- Save registers
- Pick up and save starting address from calling sequence
- Pick up and save number of characters of number
- Number of characters ≥5
  - Yes
  - C
  - No
  - Clear sum
- Mask out all but least significant 4 bits
  - A
  - B

3-17
STORE SUM
RESULT IN
INDICATED
CALL

MULTIPLY
DIGIT BY
FACTOR

ADD SUM TO
PRODUCT TO GET
NEW SUM

LAST
DIGIT

A
NO

YES

STORE SUM OR
RESULT IN AREA
INDICATED BY
CALL

RETURN

B

C

SET RESULT
= -1
3.3.2 BDOO - Build Display

3.3.2.1 Purpose
The purpose of this subroutine is to format data input for eventual output to the CRT.

3.3.2.2 Technical Description
The function of BDOO is to format the input data for output from the CRT output buffer (CRO$). Since an entire form will usually fill more than one CRT screen, the data is divided into "pages". To mark this division, EOR (end-of-record) characters have been placed at specific places in the input data so that, when one is encountered during processing, control is returned to the calling routine, which then initiates output to the CRT. The next time BDOO is entered, the processing continues at the point of the last EOR character.

Normally, this procedure is repeated until the entire form has been output. However, since a calling routine may desire a new form to be displayed before the previous one has been completely output, the A-register, upon entry, is used as a flag. If it is zero, processing continues from the last EOR character. Otherwise, the A-register indicates the number of the form requested. The forms table (FTB$) contains the beginning address of each form. Once the beginning address of the requested form is found, BDOO then may begin processing.
With each page of data BDOO builds a line table (LTB$). There is a one word entry in the table associated with each line on the CRT. Appendix C explains the contents of each word of the line table.

The following is a list of special characters and their usage in the input data.

<table>
<thead>
<tr>
<th>Character</th>
<th>Code</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>end-of-block</td>
<td>$243_8$</td>
<td>end of every answer. Precedes every heading. Output as &quot;new line&quot; character.</td>
</tr>
<tr>
<td>carriage return</td>
<td>$336_8$</td>
<td>indicates more data follows. Output as new line.</td>
</tr>
<tr>
<td>ned-of-record</td>
<td>$244_8$</td>
<td>indicates the end of a page of data.</td>
</tr>
<tr>
<td>tab</td>
<td>$334_8$</td>
<td>space over to 21st position on CRT.</td>
</tr>
</tbody>
</table>

The processing is done on a character-by-character basis. First, the level code and sequence number (if any) are picked out of the input buffer and saved in a line table word. Blanks equal in number to the value of the level code, are then stored in the output buffer. The heading and any answers follow immediately. As each end-of-block character is reached, the procedure is repeated. Once an end-of-record character is encountered, control is returned to the calling routine and the page is output. If a page exceeds the output buffer in length, an error message ("PLEASE SELECT ANOTHER FORM") is output.
### 3.3.2.2.1 Calling Sequence

CALL BDOO

<table>
<thead>
<tr>
<th>REGISTERS</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>= 0 indicates continuation of processing from the last EOP character.</td>
<td>&lt; 0 indicates output buffer (CRO$) was exceeded and an error message was output.</td>
</tr>
<tr>
<td></td>
<td>= 1 indicates begin processing at the beginning of the new requested form.</td>
<td>≥ 0 normal return</td>
</tr>
<tr>
<td>B</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>X</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
3.3.2.2.2 General Flow Chart

BUILD DISPLAY

BDOO

PICK UP AND
STORE 1ST LINE
OF CURRENT PAGE
OF FORM

A

GET NEXT
CHARACTER

END OF DISPLAY PAGE?

EOR
CHARACTER

YES

RETURN

NO

SET LINE TABLE
(LTBS) BASED ON
CHARACTER

STORE ALL
CHARACTERS TO
EOB INTO OUTPUT
BUFFER

STORE NEW LINE
CHARACTER

SET LINE TABLE
BASED ON # OF
COLONS ON CURRENT
LINE

3-22
3.3.2.3 Label Description

3.3.2.3.1 Local

BDBL - (240) blank

BDB2 - ("FORM ", 12, 13, 11, 11, "PLEASE SELECT ANOTHER FORM")

Error message output when CROS buffer is exceeded.

BDCC - colon counter; count the number of colons found on one line.

Indicates no answer, one answer, or two answers on a line.

BDCH - save location for second character of word; first character

unpacked, and second character saved.

BDCL - (272) colon

BDCS - switch used to indicate which half of the next available word

is to be unpacked.

BDEB - (243) end-of-block character

BDER - (244) end-of-record

BDE2 - (38) number of words in BDB2 error message

BDFP - pointer into tape input buffer (MTNS) that indicates the next

available word to be unpacked for processing.

BDLP - line table pointer; pointer to next available word in line table.

BDNF - (199) maximum number of forms allowed (used with input system only)

BDNL - (12) new line character

BDSC - sequence counter; counts the number of headings having sequence

numbers (or the number of headings in ID Section of form).

BDSV - three temporary save locations

BDTB - (334) tab

BDTX - (3) end-of-text character (CCI)
3.3.2.3.2 Global

None

3.3.2.3.3 Entry Point

BD00 - primary entry

3.3.2.3.4 External References

CRO$ - display output buffer
DPT$ - display pointer; points to next available location relative to CRO$ for storing output data.
FTB$ - forms table; pointer to beginning of form in memory
LTB$ - line table; see Appendix C
ODOO - output display routine
3.3.2.4 Detailed Flow Chart

BDOO

A-REGISTER

= 0

YES

NO

PICK-UP POINTER

TO 1ST PAGE OF

FORM (FTB$)

SAVE POINTER

(BDFP)

CLEAR

CHARACTER

SWITCH (BDCS)

A

A
CHARACTER IN B-REG

CLEAR LINE TABLE POINTER (BDLP)

CLEAR DISPLAY OUTPUT BUFFER POINTER (DPT$)

BLANK FILL LINE TABLE (LTB$)

CHARACTER IN B-REG

GET A CHARACTER

EOB CHARACTER

STORE CHARACTER

STORE 1ST LINE OF PAGE INTO OUTPUT BUFFER
C

BD85
STORE NEW LINE CHARACTER

CLEAR SEQUENCE COUNTER (BDSC)

D
BD25

CLEAR COLON COUNTER (BDCC)

D1

INCREMENT LINE TABLE POINTER (BDLP)

E
BD30

CHARACTER IN B-REG

BD90
GET A CHARACTER

F
SET MORE LINES
BIT (13) IN LINE
TABLE WORD
(LTB$) FOR
PREVIOUS LINE

SET CONTINUATION
LINE BIT (15) IN
LINE TABLE WORD
(LTB$) FOR
CURRENT LINE

BD8S
STORE NEW
LINE CHARACTER

INCREMENT
LINE TABLE
POINTER
(BDLP)

E
STORE SEQUENCE NUMBER IN LINE TABLE WORD (LTB$)

INCREMENT SEQUENCE COUNTER BDSC

CHARACTER IN B-REG

STORE BLANKS EQUAL IN NUMBER TO LEVEL CODE

STORE BACKGROUND INTO OUTPUT BUFFER

BD90 GET A CHARACTER

BLANK

YES

NO

P

N

3-31
SET NO ANSWER BIT (14) IN LINE TABLE WORD (LTB$) FOR CURRENT LINE

SET 2 QUESTION BIT (12) IN LINE TABLE WORD (LTB$) FOR PREVIOUS LINE
STORE THE CHARACTER, IN B-REGISTER, INTO OUTPUT BUFFER

SAVE X-REGISTER BDSV

INCREMENT OUTPUT BUFFER POINTER (DPT$)

OUTPUT BUFFER EXCEEDED

YES

TRANSFER ERROR MESSAGE TO OUTPUT BUFFER

ØD00

OUTPUT MESSAGE TO CRT

T

U
A-REGISTER
= -1

RETURN THROUGH BDOO

7 LEAST SIGNIFICANT BITS

MASK CHARACTER

STORE CHARACTER IN OUTPUT BUFFER CROS

RESTORE X-REGISTER (BDSV)

RETURN
LOAD BACKGROUND CHARACTER INTO B-REGISTER

SAVE X-REGISTER (BDSV)

CHARACTER SWITCH = 0

YES

CLEAR CHARACTER SWITCH (BDCS)

NO

W

LOAD CHARACTER INTO B-REGISTER

PICK-UP SAVED CHARACTER (BDCH)

INCREMENT FORM POINTER (BDPP)

V
V

RESTORE
X-REGISTER
(BDSV)

RETURN

W

BD95

FETCH WORD
(2 CHARACTERS)

UNPACK WORD

SAVE 2ND
CHARACTER
BDCH

X
3.3.3 BM00 - build a MEDATA record

3.3.3.1 Purpose

The purpose of this subroutine is to restore the saved codes (saved by BD00) and format the user input data for eventual output to a master tape.

3.3.3.2 Technical Description

BM00 processes the user input data which has been stored in an input buffer. This processing performs the following functions:

- restoring the special codes saved by BD00;
- storing user data into the output buffer;
- saving the SS NO and DATE response when present;
- inserting the last SS NO and DATE response when one is not present;
- performing a limited amount of error checking;
- checking for a special character (*) which indicates that the user needs more lines in order to complete his answer;
- inserting an "A" (required by MEDATA) at the end of the ID section of each form.

The first two points make up the majority of the processing. The special codes that are restored are found in the Line Table, LTB$ (see Appendix C). These codes are the level codes and the sequence numbers. In general, once the codes are restored, the remainder of the processing is the fetching of each heading and answer from the display input buffer (CRN$) and the storing of it into the tape output buffer (MTO$).
The last five points are all special cases that should be described individually.

First, if a heading has a sequence number of one or four, then that heading is either SS NO or DATE. The answers to each of these headings are saved; if in either case there is no answer, the one saved previously is inserted. This eliminates the necessity of the user inputting the same social security number or date over and over again.

There are only four headings in a form that have a sequence number. These make up the ID section. The letter "A" must immediately follow this section to indicate the end of identification information.

Each character in every answer must be checked for an asterisk. This character signals that the user needs more blank lines in order to complete the answer. When detected, the portion of the answer already processed is stored in the output buffer, and control is passed to XLO00.

A limited amount of error checking is performed on each heading and answer. The line table (LTB$) tells whether or not a heading allows an answer. If it does, a colon must be present following the heading. If not present the A-register is set to the line number (BMLC) on which the error occurred, and control is returned to the calling routine. If the first character following the colon is a left parenthesis, a right parenthesis must also be present. If there is no right parenthesis, the A-register is set to the current line number (BMLC) and control is returned.
If the tape output buffer overflows when storing data into it, an error message (BUFFER LENGTH ERROR) is output, the A-register is set to -1, and control is returned to the calling routine.

There are three characters placed in the output data that serve special purposes.

<table>
<thead>
<tr>
<th>Character</th>
<th>Code</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end-of-page</td>
<td>(241₈)</td>
<td>indicates the end of a CRT screen of data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When the line table word is equal to 404₀₈, the last data line has been processed, and the character can then be stored.</td>
</tr>
<tr>
<td>end-of-block</td>
<td>(44₈)</td>
<td>indicates the end of the answer. When the end of the line is reached, and the line table indicates no more data follows, this character is inserted.</td>
</tr>
<tr>
<td>carriage return</td>
<td>(136₈)</td>
<td>indicates the end of a CRT line of data, but more data is to follow on the next line.</td>
</tr>
</tbody>
</table>
### 3.3.2.1 Calling Sequence

**CALL BM00**

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N/A</td>
<td>-1 indicates the output buffer has overflowed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;0 indicates the line number of a user-input error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0 normal return</td>
</tr>
<tr>
<td>B</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>X</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
3.3.3.2.2 General Flow Chart

- **A**: Store end of block character into intermediate buffer
- **A1**: Ignore the first line of input data
- **A2**: First line of answer
  - **YES**: Store end of page character into intermediate buffer
  - **NO**: End of page
    - **YES**: Return
    - **NO**: B

Diagram:

```
  BM00
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
B

LST QUESTION OF BODY

YES

NO

STORE LEVEL CODE INTO INTERMEDIATE BUFFER

STORE "A" INTO INTERMEDIATE BUFFER

STORE SEQUENCE NUMBER INTO INTERMEDIATE BUFFER

STORE QUESTION INTO INTERMEDIATE BUFFER

C

D
D

ANSWER FIELD EMPTY

YES

FIRST LINE OF ANSWER

NO

QUESTION SS NO. OR DATE

NO

YES

STORE PREVIOUS SS NO OR DATE INTO INTERMEDIATE BUFFER

H

E

QUESTION SS NO. OR DATE

YES

F

G

NO

3-46
STORE LINE OF ANSWER INTO INTERMEDIATE BUFFER

MORE LINES OF ANSWER TO FOLLOW

WAS ASTERISK IN ANSWER

STORE INTERMEDIATE BUFFER INTO OUTPUT BUFFER

INPUT & STORE EXTRA LINES OF ANSWER

SAVE SS NO. OR DATE ANSWER

STORE INTERMEDIATE BUFFER INTO OUTPUT BUFFER
3.3.3.3 Label Description

3.3.3.3.1 Local

BMAS  \((52_{8})\) asterisk

BMB  ('BUFFER LENGTH ERROR') error message

BMBL  \((40_{8})\) blank

BMCL  \((72_{8})\) colon

BMCP  \((51_{8})\) closed parenthesis

BMCR  \((136_{8})\) carriage return

BMCS  character switch. Indicates into which half of the word the next character is to be packed in the tape output buffer (MTO$).

BMDP  display pointer. Indicates the location of the next character to be processed from the input buffer (CRN$).

BMDT  seven word buffer used for saving the answer to the ID heading DATE.

BME  \((19)\) number of words in the error message BMB.

BMEP  \((241_{8})\) end-of-page

BMIB  intermediate buffer. 46 word buffer where data is stored prior to being placed in tape output buffer (MTO$).

BMIF  ID flag. Flag indicates that the heading with a sequence number of four has just been processed and the ID section is complete.

BMIP  intermediate buffer pointer. Indicates next location in intermediate buffer (BMIB) is for storing data.

BMLC  line counter. Indicates the line number on the CRT that is currently being processed.
BMLP: line table pointer. Indicates the location in the line table associated with the heading and answer currently being processed.

BMNL: (12₈) new line (CCI)

BMOP: (50₈) open parenthesis

BMP: save location for tape output buffer pointer (MPT$)

BMPC: position counter. Counts the number of character positions on the CRT line that have already been processed.

BMPCM: storage location containing the maximum number of CRT positions allowed for the heading and answer of the current line being processed (20 or 40).

BMPS: temporary save location for saving the intermediate buffer pointer (BMDP)

BMSA: temporary save location for saving the display pointer (BMDP).

BMSB: temporary save location for saving the contents of the B-register.

BMSS: eleven word buffer used for saving the answer to the ID heading SS No.

BMSV: two temporary save locations

BMTX: (03) end-of-text

BM1F: sequence number one flag. Indicates that the heading and answer currently being processed have a sequence number of one.

BM4F: sequence number four flag. Indicates that the heading and answer currently being processed have a sequence number of four.
3.3.3.3.2 Global
None

3.3.3.3.3 Entry Point
BMOO

3.3.3.3.4 External References
CRN$ CRT input buffer
CRO$ CRT output buffer
LTB$ line table (see Appendix C).
MPT$ tape output buffer pointer. Points to next available location in buffer.
MTO$ magnetic tape output buffer
ODOO routine to output a buffer (CRO$) to the CRT
XLOO routine to allow user more lines on which to complete his answer
3.3.3.4 Detailed Flow Chart

BM00

0

CLEAR ID FLAG (BMIF), LINE TABLE POINTER (BMLP), INTERMEDIATE BUFFER POINTER (BMIP), DISPLAY INPUT BUFFER POSITION COUNTER (BMPC)

SET LINE COUNTER TO ONE (BMLC)

INITIALIZE DISPLAY INPUT POINTER (BMDP) BEGINNING OF INPUT BUFFER (CRN$+1)

SAVE TAPE OUTPUT BUFFER POINTER (MPT$)

1
ZERO INDICATING NEW RECORD IS BEING BUILT

TAPE OUTPUT BUFFER POINTER = 0

YES

CLEAR CHARACTER SWITCH (BMCS)

END OF BLOCK CHARACTER (044)

BMGO STORE EOB CHARACTER

BMFC < 40

YES

POSITION COUNTER < 40

NO

YES

GET A CHARACTER

CHARACTER = "NEW LINE"

YES

IGNORE FIRST LINE OF DISPLAY

NO
SET RETURN ADDRESS (BM17) AT BMFO

52

5 BM18

TEST CONTINUATION LINE BIT (15) IN LINE TABLE (LTBS) WORD

BIT SET YES

NO

ID FLAG (BMIF) = 0

YES

NO

7

6

9
INDICATES END OF ID SECTION OF RECORD

BIT 0 - 5 OF LTB$

PICK UP LEVEL CODE FROM LINE TABLE (LTB$) WORD

BIT 6-11 OF LTB$

PICK UP SEQUENCE NUMBER FROM LINE TABLE (LTB$) WORD

STORE "A"

CLEAR ID FLAG (BMIF)

STORE LEVEL CODE

BMGO

3-55
SEQ. NO. = 1

SET SEQ. NO. 1 ID FLAG (BM1F)

SEQ. NO. = 4

SET ID FLAG (BM4F)

SET SEQ. NO. 4 ID FLAG (BM4F)

BMGO
STORE SEQ. NUMBER

8

NO

YES

YES

NO

9
2-QUESTION BIT (12) SET

SET POSITION - MAX COUNTER (BMPM) = 20

SET POSITION - MAX COUNTER (BMPM) = 40

NO

ANSWER BIT (14) SET

NO

CHARACTER HAS BEEN SAVED IN B-REGISTER

CHARACTER = "NEW LINE"

NO

11

BM30

12

BM32

13

BM35

14

15

14

47

3-58
END OF TEXT - LAST CHARACTER INPUT FROM DISPLAY

POSITION COUNTER = POSITION - MAX

CHARACTER = ETX

YES

NO

BMGO
STORE CHARACTER

BMPC = BMPM

YES

NO

BMF5
GET A CHARACTER

13

47

3-59
CHARACTER HAS BEEN SAVED IN B-REGISTER

CHARACTER = "NEW LINE"

NO

YES

INPUT BUFFER POINTER (MPT$) TO ITS ORIGINAL VALUE

ERROR - RETURN LINE NUMBER CONTAINING ERROR.

SET A-REGISTER TO VALUE OF LINE COUNTER BMLC

RETURN
HEADING NOT
SS NO.,
BM1F = 0

BM68

CHARACTER = "NEW LINE"

BM70

BM1F = 0

CLEAR SEQ. NO.
1 ID FLAG
(BM1F)

SAVE DISPLAY
POINTER (BMDP)

SAVE CHARACTER
IN B-REGISTER

23

57

26

25

SAVE SS NO.
25) BM76

SAVE SS NO. IN BUFFER (BMSS)

26 BM71

HEADING NOT DATE - NO
BM4F = 0

BM4F = 4

YES

CLEAR SEQ. NO. 4 ID FLAG

SAVE DATE

NO

SAVE DISPLAY POINTER (BMDP)

27

28

29
SAVE CHARACTER IN B-REGISTER

SAVE DATE IN BUFFER (BMDT)

RESTORE DISPLAY POINTER (BMDP)

RESTORE SAVED CHARACTER

CHARACTER = OPEN PARENTHESIS

3-66
30

BMGO
STORE OPEN PARENTHESIS

31
BM75

BMF5
GET A CHARACTER

CHARACTER = "NEW LINE" YES

NO

BMGO
STORE CHARACTER

CHARACTER = CLOSE PARENTHESIS YES

NO

32

33

16

3-67
POSITION COUNTER = POSITION - MAX;
END OF LINE

END OF TEXT
BM83

POSITION COUNTER = POSITION MAX;
END OF LINE

BMPC = BMPM

CHARACTER = "NEW LINE"

CHARACTER = ASTERISK

POSITION BMDP TO BEGINNING OF NEXT DISPLAY LINE

3-69
37 BM84

STORE RETURN ADDRESS (BM86) AT BMFO

BMGO

STORE CARRIAGE RETURN

53

38 BM86

XLOO

INPUT & STORE EXTRA LINES OF ANSWER

0

3-70
POSITION COUNTER = POSITION-MAX;
END OF ANSWER FIELD

POSITION-MAX = 40;
END OF DISPLAY LINE

END OF BLOCK CHARACTER
BM90

TEST MORE - LINES BIT 913) IN LINS TABLE (LTBS) WORD

BIT SET

NO

YES

CLEAR POSITION COUNTER (BMPC)

BM95

BMF5
GET A CHARACTER

CHARACTER = BLANK

YES

NO

CHARACTER = "NEW LINE"

YES

NO

41

42

43

44

46

47

3-72
43

CHARACTER = ETX

YES

NO

LOAD B-REGISTER WITH CARRIAGE RETURN

INCREMENT LINE COUNTER (BMLC)

INCREMENT LINE TABLE POINTER (BMLP)

44

BMAO

DECREMENT INPUT DISPLAY BUFFER POINTER BY POSITION COUNT

45

END OF TEXT
CLEAR POSITION COUNTER (BMPC)

POSITION COUNTER = 40;
END OF DISPLAY LINE

BMCO

SAVE THE B-REGISTER

3-74
48

CHARACTER = CARRIAGE RETURN

YES

50

BMGO
STORE EOB CHARACTER

SET RETURN ADDRESS (BM10) AT BMFO

52

3-75
TEST FOR OVERFLOW OF TAPE OUTPUT BUFFER (MTO$)

"BUFFER LENGTH ERROR"

SET RETURN ADDRESS (BM80) AT BMFO

MPT$ > 2880

NO

TRANSFER ERROR MESSAGE TO DISPLAY OUTPUT BUFFER (CRO$)

OP00 OUTPUT ERROR MESSAGE

SET A-REG. TO -1

RETURN
TEST CHARACTER SWITCH

PICK UP A CHARACTER FROM INTERMEDIATE BUFFER (BMIB)

SAVE X-REGISTER

BMCS = 0

YES

NO

PICK UP LAST WORD STORED IN TAPE OUTPUT BUFFER (MTO$)

CLEAR SECOND HALF OF WORD

53

BMD4

BMD5

54

55
PACK NEW INTERMEDIATE BUFFER CHARACTER INTO SECOND HALF OF TAPE OUTPUT BUFFER WORD

STORE PACKED WORD INTO TAPE OUTPUT BUFFER (MTO$)

CLEAR CHARACTER SWITCH (BMCS)
PICK UP NEXT TAPE OUTPUT BUFFER WORD TO BE PACKED

CLEAR FIRST HALF OF WORD

PACK NEW INTERMEDIATE BUFFER CHARACTER INTO FIRST HALF OF TAPE OUTPUT BUFFER WORD

INCREMENT CHARACTER SWITCH (BMCS)

56
IS EITHER SEQ. NO. 1 OR 4 ID FLAG SET?

BM1F OR BM4F SET

RESTORE INTERMEDIATE BUFFER POINTER (BMIP)

IS SEQ. NO. 1 ID FLAG SET?
(IS CURRENT HEADING SS NO.?)

BM1F SET

CLEAR SEQ. NO. 1 ID FLAG

STORE PREVIOUS SS NO. (BMSS) INTO ANSWER FIELD OF CURRENT SS NO.
CLEAR SEQ.
NO. 4 ID FLAG

STORE PREVIOUS
DATE (BMDT)
INTO ANSWER
FIELD OF
CURRENT DATE

22
I,

INCREMENT POSITION COUNTER (BMPC)

PICK UP A CHARACTER FROM DISPLAY INPUT BUFFER (CRN$) USING POINTER (BMDP)

MASK OUT THE EIGHTH BIT (PARITY BIT)

PUT CHARACTER IN B-REGISTER

RETURN
BMGO

SAVE X-REGISTER

INCREMENT INTERMEDIATE BUFFER POINTER (BMIP)

STORE CHARACTER FOUND IN THE B-REGISTER INTO THE INTERMEDIATE BUFFER (BMIB) AT A LOCATION INDICATED BY POINTER (BMIP)

RESTORE X-REGISTER

RETURN
3.3.4 XLOO - Extra Line Generation

3.3.4.1 Purpose
XLOO is a subroutine whose purpose is to allow more answer space for any question on the MEDATA input forms.

3.3.4.2 Technical Description
When the user hits the interrupt key the Build Medata Record Routine (BM00) begins packing the input data from CRN$ character by character into the CRT Output Buffer (CRO$). On finding the end of text character, the routine clears the CRT screen and resets the cursor. The user then keys in his extra data, comments, etc. XLOO calls ID00 to input the extra data into CRN$ and the data is transferred to the tape output buffer with all necessary control characters inserted. XLOO then calls OD00 to output the unanswered remainder of the CRT page located in CRO$ on which the extra data was inserted. All applicable counters, pointers, etc., and the line table are updated. After all questions on the page have been answered, XLOO calls ID00 to input the data and then returns control to the calling program.

3.3.4.2.1 Calling Sequence
CALL XLOO

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved upon entry</td>
<td>Restored upon exit</td>
</tr>
<tr>
<td>B</td>
<td>Saved upon entry</td>
<td>Restored upon exit</td>
</tr>
<tr>
<td>X</td>
<td>Saved upon entry</td>
<td>Restored upon exit</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3-85
3.3.4.2.2 General Flow Chart

XLOO

MOVE UNANSWERED HEADINGS OF CRN$: TO CRO$

RESET CURSOR, CLEAR SCREEN

SAVE COUNTERS, POINTERS

IDO0 INPUT EXTRA DATA

PACK DATA INTO MTO$

ODOO OUTPUT PREVIOUS UN-ANSWERED DATA

RESET LINE TABLE

IDO0 INPUT ANSWERED CRT DATA

EXIT
3.3.4.3 Label Description

3.3.4.3.1 Local

XLBF  (962) number of words to be input from the CRT during an input call to ID00

XLDEV storage buffer containing the message "CRT DEVICE ERROR"

XLEF counter which contains the number of characters stored in the intermediate buffer BMIB

XLEL (40) CRT line character limit

XLER counter used to halt the program in case of an unrecoverable parity or hardware error.

XLID (1) function used to initiate an output to the CRT

XLPR storage buffer containing the message "CRT PARITY ERROR"

XLSA storage location used for storing the original contents of the A-register

XLSB storage location used for storing the original contents of the B-register

XLSTAT storage location containing the status of an input or output operation

XLSX storage location used for storing the original contents of the X-register

XLWC storage location containing the number of words transferred from the input buffer to the output buffer

XLWD (4) number of control parameters necessary to initiate a CRT clear screen - cursor reset

3.3.4.3.2 Global

N/A

3.3.4.3.3 Entry Points

XLOO - primary entry point
3.3.4.3.4 External References

BMBL (0240) storage location containing the value of the space character

BMCR (0215) storage location containing the value of the Carriage Return character

BMDP storage location containing the input buffer display pointer

BMEP (02vl) storage location containing the value of the End of Page character

BMNL (012) storage location containing the value of the New Line character

BMPC storage location containing the input position counter

BMTX (03) storage location containing the value of the End of Text character

CRN$ CRT input buffer

CRO$ CRT output buffer

LTB$ starting address of the Line Table

XLRC (12,1,8,4) parameters which initiate a CRT Clear Screen - cursor reset
H

INCREMENT LINE TABLE POINTER

LOAD LINE TABLE ENTRY

ALL SPACES?

YES → J

NO

MOVE ENTRY UP IN TABLE

↓

H
3.3.5 CPOO - Control Program

3.3.5.1 Purpose
The purpose of this routine is to control the processing of the user-input display requests. This processing includes:

- acceptance of a request to display a specified form.
- retrieval and display of a different form or a different page of the same form.
- recognition of an error in the input display.
- compliance with a user request to redisplay the page being displayed or the first page of the form being displayed.

3.3.5.2 Technical Description
After control is received from routine INOO, the user is requested to select a form to be displayed. Once that form is selected and input to CPOO, the request is analyzed. Its contents should consist of a form number and a page number to be displayed. If the form number is illegal, an error message is printed on the CRT. A legal form number causes the forms table to be searched for the address of the beginning of the selected form. It is found and its first page is displayed on the CRT to allow the user to fill in the information for the form. Upon completion of the form, the user transmits his next request to the program by affecting the form number or page number. The possible requests that may be received and the subsequent processing
are described below:

a. **Present Page is Complete and Correct**

This request is recognized by the condition that neither the form number nor the page number have changed. Upon receipt of this command, CPOO calls on Build Medata Record (BM00) to build and store the output record in the tape output buffer (MTO$). Should this be the last page of the form, CPOO calls Output Medata Record (MD00) to write the entire tape output buffer. The Medata Output Buffer Pointer (MPT$) is reset and the Medata Output Buffer blank filled. The page number is reset to one (1) and the Build Display routine (BDOO) is called.

b. **Error in Data Recording on Current Page**

This request is recognized by receipt of a page number of X. The user has recognized a mistake in recording the data for this page and wishes to correct it. CPOO simply recalls routine OD00 for a re-display of the present page of the background form.

c. **Error in Current Form**

This request is recognized by the receipt of a form number of X. This requires the display of page one of the current form. CPOO blank fills the Medata Output Buffer (MTO$) and resets the Medata Output Buffer Pointer (MPT$). The page number is reset to one (1). The Build New Display routine is called to build the new CRT Output Buffer. Output Display (OD00) is then called.
d. **New Form Request**

This request is recognized by a change in form number. The Medata Output Buffer is blank filled. The page number is set to one (1) and the Medata Output Buffer Pointer is reset. The Build New Display routine is called with the new form and page number. The Output Display routine is then called.

e. **Dump Record Request**

This request is acknowledged when a "D" is found in the form number. Routine D000 is called to backspace the output tape one record, read the record in, and dump it to the teletype. After the dump is complete, a form number to display is requested via the CRT.

f. **Present Page Complete but Incorrect**

This request is the same as request a. but routine CPO0 detects some error in the input information and will not accept the record for output onto the tape. The line number of the error is determined, converted to ASCII, and displayed, along with an error message, to allow the user the opportunity of correcting the erroneous data.

g. **Terminate Output Request**

This request is recognized by the receipt of an asterisk for a form number. The Medata Output Buffer is blank filled and an End of Record (EOR) is written in the first position of the output buffer. An end of file mark is written on the output tape and it is rewound.
3.3.5.2.1 Calling Sequence

JMP  CP00

PARAMETER

None

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>destroyed</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>destroyed</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>destroyed</td>
<td></td>
</tr>
<tr>
<td>Overflow</td>
<td>destroyed</td>
<td></td>
</tr>
</tbody>
</table>

3-101
3.3.5.3 Label Description

3.3.5.3.1 Local Labels

- **CPB1**: error message buffer "illegal form number"
- **CPEB**: display error message buffer
- **CPEJ**: teletype message buffer
- **CPE1**: error message buffer length
- **CPF1**: input form number
- **CPF0**: output form number
- **CPKB**: ASCII blank
- **CPKE**: end of record character
- **CPLN**: line number, two word ASCII buffer
- **CPNW**: number of words to display
- **CPP1**: page number input
- **CPP0**: page number output
- **CPST**: status indicator

Data constants: CPK1, CPK2, CPK4, CPK7, CPK8, CPKA, CPKS

3.3.5.3.2 Global Labels

None

3.3.5.3.3 Entry Points

- **CP00**: primary entry point
- **CP90**: blank fill of output buffer
- **CP98**: convert line number from binary to ASCII
3.3.5.3.4 External References

AIOO     ASCII to integer conversion entry
BD00     build new display entry
BM00     build Medata Record entry
CRN$     input buffer
DO00     backspace tape and dump record entry
HCO0     CRT handler entry
HM00     mag tape handler
ID00     input display entry point
MD00     output medata record entry
MPT$     output buffer pointer
MTO$     mag tape output buffer
OD00     output display entry
OZKA     TTY message entry
OZKR     carriage return, line feed to TTY entry
046000   entry to AID II
3.3.5.4 Detailed Flow Chart

CP00

INITIALIZE

CP 10

INPUT FORM
NUM & PAGE
NUM FROM CRT

FORM
NO. = "x"

YES

NO

CP70

FORM
NO. = "x"

YES

NO

CP18

FORM
NO. = "p"

NO

YES

A

CP11

3-106
A

BACKSPACE TAPE

D000

OUTPUT "FORM" TO CRT

OD00

CP10

CP11

CONVERT FORM # FROM ASCII TO BINARY

A100

B

3-107
CP 20

CONVERT PAGE NUMBER A100

PAGE NUMBER SAME?

NO

CP40

YES

BUILD METADATA RECORD BMOO

BUFFER EXCEEDED?

YES

CP10

NO

ERROR IN RECORD?

YES

CP60

NO

LAST PAGE?

YES

CP25

NO

STORE CONTROL CHAR AT END OF BUFFER

C

3-109
C

OUTPUT RECORD TO TAPE

RESET PAGE COUNTER

CP50

CP 25

INCREMENT PAGE COUNTER

CP 30

BUILD NEW DISPLAY

ERROR?

YES

NO

CP10

CP40

MDOO

BDOO

3-110
CP 60

SET LINE NUMBER OF ERROR

CONVERT LINE NUMBER TO ASCII

OUTPUT ERROR MSG. TO CRT

HCOO

CP 10

CP 65

DISPLAY ILLEGAL FORM NO. MESSAGE

HCOO

CP 10

3-111
3.3.6 D000 (D00) - Dump to Teletype

3.3.6.1 Purpose

The purpose of this routine is to list on the teletype 1) a form already resident in core (D00) or 2) a form that was just written on tape (D000).

3.3.6.2 Technical Description

This subroutine has two entry points. From one entry (D00), the calling routine specifies in the calling sequence the beginning address of the buffer to be printed. This allows output from any location in memory. From the other entry point (D000), the form to be printed is on tape and must be input to the computer before it can be output to the teletype. In both cases, the format of the buffers is the same as the tape format. (See Appendix A). The processing portion of the routine removes all extraneous codes and formats the data similar to the CRT format (See Appendix B). Appendix E has a sample teletype output. A description of the special codes follows.

<table>
<thead>
<tr>
<th>Character</th>
<th>Code</th>
<th>Purpose/Processing of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>end-of-block</td>
<td>(224_8)</td>
<td>the first one is output as a carriage return and line feed. Any subsequent consecutive such characters are ignored.</td>
</tr>
<tr>
<td>end-of-page</td>
<td>(241_8)</td>
<td>this character is converted to a carriage return and line feed.</td>
</tr>
</tbody>
</table>

3-114
<table>
<thead>
<tr>
<th>Character</th>
<th>Code</th>
<th>Purpose/Processing of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>end-of-record</td>
<td>(236)</td>
<td>this character signals the end of processing (listing).</td>
</tr>
<tr>
<td>carriage return</td>
<td>(215)</td>
<td>this character signals the end of processing (listing).</td>
</tr>
<tr>
<td>level code</td>
<td></td>
<td>This is a digit from one to nine. It is output as consecutive blanks. The number of blanks is equal to the value of the level code.</td>
</tr>
<tr>
<td>sequence number</td>
<td></td>
<td>ignored.</td>
</tr>
</tbody>
</table>

**3.3.6.2.1 Calling Sequence**

CALL D000 or

CALL D00,A,B

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>the address of the beginning location of the buffer containing the form.</td>
</tr>
<tr>
<td>B</td>
<td>the number of words in the buffer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>B</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>X</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3-115
3.3.6.3 Label Description

3.3.6.3.1 Local

DBUF - save location for beginning address of buffer to be output

DCS - a switch used to determine which character is to be unpacked next

DEB - the save location for the ending location +1 of the buffer to be output

DST - the status word used with the tape read

DSV - save location of the second character of a word containing two characters

DX - save location for X-register in DGET.

3.3.6.3.2 Global

None

3.3.6.3.3 Entry Point

D000 - primary entry point

3.3.6.3.4 External References

HMO0 - tape handler used to manipulate magnetic tape

MTO$ - tape output buffer.
3.3.6.4 Detailed Flow Chart

DO00

BACK-SPACE

READ

SAVE BEGINNING LOCATION OF BUFFER IN (DBUF)

SAVE ENDING LOCATION OF BUFFER IN (DEB)

CLEAR CHARACTER SWITCH (DCS)

D20

PROCESS RECORD

RETURN

3-118
DOO

SAVE BEGINNING LOCATION OF BUFFER (DBUF)

COMPUTE AND SAVE ENDING ADDRESS OF BUFFER (DEB)

CLEAR CHARACTER SWITCH (DCS)

D20

PROCESS RECORD

RETURN

3-119
3-120
D45

CHARACTER IN B-REG

E

OZKC
OUTPUT BLANKS EQUAL IN NO. TO LEVEL CODE

F

D45

CHARACTER IN B-REG

END OF BLOCK?

DGET
GET A CHARACTER

EOB CHARACTER

YES

A

CARRIAGE RETURN

YES

G

NO

OZKC
OUTPUT CHARACTER

3-122
DSO

G

OZKR

OUTPUT
CARRIAGE RETURN
- LINE FEED

F
UNPACK A CHARACTER FROM BUFFER (DBUF)

DGET

CHARACTER SWITCH = 0

YES

NO

CLEAR CHARACTER SWITCH (DCS)

PICK UP SAVED CHARACTER (DSV)

RETURN

DG1

H

SAVE X-REGISTER (DX)

I

3-124
I

END OF BUFFER

YES

NO

FETCH CHARACTER

OR-IN EIGHTH BIT IN BOTH CHARACTERS OF WORD

SAVE 2ND CHARACTER (DSV)

INCREMENT BUFFER POINTER (DBUF)

INCREMENT CHARACTER SWITCH (DCS)

J

RETURN THROUGH DIO ENTRY
3.3.7 HCOO - CRT Input/Output Routine

3.3.7.1 Purpose

HCOO is a general purpose CRT input/output routine. The purpose of the routine is to reduce the amount of duplicated effort normally inherent in a programming task when individual users develop their own I/O programming for a commonly used peripheral device. HCOO allows processing in the calling program to take place while the data transfer is occurring. In order to accomplish this, HCOO was designed to return control to the user as soon as the I/O was initiated. The CRT status is made available to the user by HCOO at the completion of the data transfer.

3.3.7.2 Technical Description

HCOO performs three input/output functions: input from the CRT, clear the screen and output to the CRT, and output to the CRT without clearing the screen. In addition to the desired I/O function, the calling program provides HCOO with the address of the data buffer, the number of words to transfer, and a status word for HCOO to update. The calling program must supply in the data buffer the CCI CRT function and control characters necessary to obtain the desired results at the CRT display. Reference the CC-30 Communications Station Reference Manual for a description of these functions and control characters.

All data buffers are transferred by HCOO in a block mode under control of the BIC. The beginning and ending address of the buffer to be transferred are loaded into the BIC initial and final registers by HCOO.
HC00 also sets up the Asynchronous Half Duplex Controller for input or output since this is a one-way device only. The Priority Interrupt Module (PIM) is enabled to honor interrupts caused by the BIC End of Transfer and the INT key on the CCI keyboard. If a clear operation is requested by the function argument, a 'halt local I/O' command and a 'clear' control character are sent to the CCI controller on a programmed I/O basis (BIC not used) before the data buffer is output. If input is requested by the calling program, HC00 outputs a 'transmit' control character to the CRT and then sets the Half Duplex Controller up to receive input.

For both input and output, HC00 returns control to the calling program after the BIC transfer has been initiated. HCIN, the interrupt processing subprogram, regains control when the input/output operation is complete, checks for CRT parity error, number of words transferred, etc., and then after setting the user status word, returns to the calling program. (See Calling Sequence for details on STATUS.) The BIC end of transfer interrupt will cause HCIN to regain control on any fixed length data transfer where the calling program knows the exact length of the record. If the length of the record is not known (such as a variable length input), the user must test for completion of transfer by checking for the termination of the CARRIER ON signal at any time for the on-line CRT, or using a time-out routine for the modem CRT. When the user determines that the variable length input is complete, he must call HCIN to get the status and/or number of words transferred. Again, on fixed length records, HCIN will be called automatically when the BIC EOT interrupt occurs.
User interface notes:

- HCOO assumes that the calling program will determine when HCOO should be called to input from the CRT. HCOO facilitates the use of the INT key for this purpose by setting an indicator HCKI when the INT key is depressed. The calling program may then check HCKI at any time to determine if the CRT user desired to input a screen of data.

- There are two control characters that effect the number of characters actually received from the CRT on an input block transfer. They are the new line character (§) and the end of text character (▲). If the CRT user at the keyboard is free to use these characters at his own discretion, it is very possible to receive variable length records at the computer. This possibility requires the calling program to do two things in relationship to CRT input and the HCOO routine:

  (1) The receiving buffer should be assigned a buffer area large enough to contain the maximum size record that could originate from the CRT. The number of words passed in the calling sequence should reflect the maximum buffer size.

  (2) The calling program must periodically check for termination of input as described above.
3.3.7.2.1 Calling Sequence

CALL HCOO,X,Y,Z,STAT

PARAMETER FUNCTION

X
Address of the I/O function code

X = 0 for input
X = 1 for output preceded by a clear operation
X = 2 for output with no clear operation

Y
Beginning location of data buffer.
If Y is an output buffer, the CRT data must be 7 bit ASCII characters, one character per word right justified.
All other bits are ignored. If Y is an input buffer, at the completion of the input operation the buffer will contain 7 bit ASCII characters with the 8th bit containing the parity bit (even). Each character will be packed right justified, one character per word with the rest of the word zero.

Z
Number of words (characters) to transfer.

STAT
Status of I/O operation

STAT = -1 I/O in progress
STAT = -2 CRT parity error
STAT = ≥0 I/O successfully completed. Number represents the number of words/characters transferred.
Returns

**CALL+6**

HCOO returns here if a previous I/O request has not been completely processed by HCOO. HCIN must have been entered as a result of a BIC EOT interrupt or a call to HCIN for wrap-up of the I/O function.

**CALL+8**

Normal Return. I/O initiated. Calling program must check STAT for completion or determine I/O completion as indicated in the Technical Description.

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>B</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>X</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>Overflow</td>
<td>Not used</td>
<td>Not effected</td>
</tr>
</tbody>
</table>

3-131
3.3.7.2.2 General Flow Chart

- **HC00**
  - **HANDLER BUSY?**
    - **YES**
      - **SET UP BIC AND INTERRUPTS**
    - **NO**
  - **INPUT?**
    - **YES**
      - **SET USER STATUS TO I/O IN PROGRESS**
    - **NO**
  - **PRECEDE OUTPUT WITH CLEAR OPERATION?**
    - **NO**
    - **YES**
      - **OUTPUT CLEAR COMMANDS**
  - **RETURN**

**A**
- **SET HC00 ROUTINE BUSY -1 → HCBU**

3-132
ACTIVATE BIC

SET H/P CONTROLLER FOR OUTPUT

CONNECT WRITE BUFFER TO BIC

RETURN

OUTPUT TRANSMIT CHARACTER

ACTIVATE BIC

SET USER STATUS TO I/O IN PROGRESS

D

3-133
CONNECT READ BUFFER TO BIC

RETURN

ENTERED AS RESULT OF BIC EOT INTERRUPT OR USER PROGRAM CALL

HEMM ENABLE BIC AND KEYBOARD INTERRUPTS

SET UP H/D CONTROLLER FOR INPUT

CRT PARITY ERROR?

YES

STORE PARITY ERROR STATUS (-2) IN USER STATUS

RETURN

NO

COMPUTER NO. OF WORDS TRANSFERRED

E

3-134
3.3.7.3 Label Description

3.3.7.3.1 Local

HCA  Save area for A-Register
HCB  Save area for B-Register
HCBU Handler busy indicator - Set to -1 when HCOO still processing previous I/O request
HCCL Clear control character output to CCI controller
HCHL Halt Local I/O function output to CCI controller
HCII Indirect address for subroutine arguments
HCIO Input/output function being performed
HCSB Beginning buffer address. Employed later to compute the number of words transferred.
HCSH SOH control character output to CCI controller
HCII Transmit control character output to CCI controller
HCX  Save area for X-Register

3.3.7.3.2 Global

HCKI Indicator which is incremented by 1 when an interrupt is caused by the INT key being pressed on the CCI keyboard. Used by calling program to determine if user has requested input of CRT data.

3.3.7.3.3 Entry Points

HCIN Entry when calling program has determined I/O is complete and wishes status of CRT device and/or number of words transferred. (Used only when variable length records are input.)

3.3.7.3.4 External References

None
3.3.7.4 Detailed Flow Chart

HC00

HCSR
SAVE
REGISTERS

HANDLER
BUSY?

YES

A

NO

HCMM
ENABLE CRT
INTERRUPTS

SET CRT
HANDLER BUSY
-1 → HCBU

SAVE I/O
FUNCTION IN
HCIO

NOP BIC
EOT INTERRUPT
LOCATION

B

A

SET HC00
FOR BUSY RETURN
CALL +6

HCRR
SAVE
REGISTERS

RETURN

3-137
B

INITIALIZE CRT AND BIC

SET UP BIC EOT TRAP LOCATION TO JMPM HClN

SET UP KEYBOARD INTERRUPT TRAP TO JMPM HC20

COMPUTE ENDING ADDRESS OF BUFFER

SET BIC INITIAL AND BIC FINAL

HC10=0? INPUT?

YES

G

NO

C
SET USER STATUS TO -1 (I/O IN PROGRESS)

SET HCOO TO NORMAL RETURN CALL+8

HCIO=2?

YES

OUTPUT WITHOUT CLEAR?

NO

SET UP H/D CONTROLLER FOR OUTPUT

OUTPUT START OF HEADER (SOH) TO CRT

HCWW

WAIT FOR WRITE BUFFER TO EMPTY
OUTPUT
HALT LOCAL I/O TO CRT

HCWW

OUTPUT CLEAR
SCREEN CHAR. TO CRT

HCDL

DELAY 50 MILLISECONDS FOR CLEAR
OPERATION TO TAKE PLACE

E

HCSC

HCRR
RESTORE REGISTERS

ACTIVATE BIC

F

3-140
F

SET UP H/D
CONTROLLER FOR
OUTPUT

CONNECT WRITE
BUFFER TO BIC

RETURN

G

HCO1

HCIT
OUTPUT
TRANSMIT

NOP BIC
EOT INTERRUPT
LOCATION

INITIALIZE BIC
ACTIVATE BIC

H
SET USER STATUS TO I/O IN PROGRESS

STORE JMPM IN BIC EOT INT. LOCATION

POSITION HC00 FOR NORMAL RETURN CALL +8

SET USER STATUS TO -1

HCRR RESTORE REGISTERS

CONNECT READ BUFFER TO BIC

RETURN
ENTERED AS RESULT OF A BIC EOT INTERRUPT OR USER PROGRAM CALL.

HCIN

HCSR
SAVE
REGISTERS

NOP BIC
EOT INTERRUPT
LOCATION

HCOMM
ENABLE BIC
INTERRUPTS

SET UP H/D
CONTROLLER FOR
INPUT

HCSS
GET CRT
STATUS

PARITY
ERROR?

STORE PARITY
ERROR STATUS (-2)
IN USER STATUS

SET HANDLER
NOT BUSY
0 → HCBU

HCRR
RESTORE
REGISTERS

RETURN

J

I

NO

YES
BIC INITIAL CONTAINS THE BUFFER ADDRESS WHERE THE LAST CHARACTER WAS INPUT OR OUTPUT.

INPUT BIC INITIALIZE

COMPUTE NUMBER OF WORDS TRANSFERRED

STORE NUMBER OF WORDS TRANSFERRED IN USER STATUS

SET HC00 NOT BUSY

0 → HMBU

HCRR
RESTORE REGISTERS

INITIALIZE BIC

RETURN
OUTPUT TRANSMIT CHARACTER ROUTINE

HCIT

SET UP H/D CONTROLLER FOR OUTPUT

OUTPUT TRANSMIT CONTROL CHARACTER

HCWW WAIT FOR WRITE BUFFER TO EMPTY

SET UP H/D CONTROLLER FOR INPUT

RETURN

SENSE CRT STATUS ROUTINE

HCSS

PARITY ERROR?

YES

LOAD A WITH PARITY ERROR STATUS (-2)

RETURN

NO

CLEAR A-REGISTER
DELAY SUBROUTINE. HCDL DELAYS IN INCREMENTS OF 50 MILLISECONDS. THE NUMBER OF INCREMENTS IS PASSED TO HCDL IN THE X-REGISTER.

DELAY 50 MILLISECONDS

SUBTRACT 1 FROM X-REGISTER

X-REGISTER ZERO?

RETURN

DELAY 50 MILLISECONDS

RETURN
WAIT FOR WRITE BUFFER TO EMPTY ON CRT HALF DUPLEX CONTROLLER

WRITE BUFFER READY?

YES

RETURN

SAVE REGISTERS

HCSR

A REG → HCA
B REG → HCB
X REG → HCX

RETURN

RESTORE REGISTERS

HCRR

HCA → A REG
HCB → B REG
HCX → X REG

RETURN

3-147
ENABLE BIC EOT AND KEYBOARD INTERRUPT ON APPROPRIATE PIM'S

INITIALIZE PIM

CLEAR INTERRUPT REGISTER

OUTPUT MASK

ENABLE INTERRUPTS

RETURN

HC20

SET KEYBOARD INTERRUPT INDICATOR HCKI

RETURN

ENTERED AS RESULT OF A KEYBOARD INTERRUPT CAUSING A TRAP TO THE KEYBOARD INTERRUPT LOCATION WHERE A JMPM TO HC20 IS PERFORMED.
3.3.8 HMOO - Magnetic Tape Handler

3.3.8.1 Purpose

HMOO is a general purpose tape routine capable of handling all available tape functions. The purpose of this routine is to reduce the duplication of effort that is inherent when individual users develop their own coding to handle each specific tape I/O operation. The routine currently does not return control to the calling program until I/O is complete or an error is detected, but as soon as additional interrupts are available on the DOC system, the routine will be modified to allow overlapped processing during tape operations.

3.3.8.2 Technical Description

HMOO allows input or output of variable length tape records. The Model 620i-31 Magnetic Tape System Controller is capable of performing 8 functions (see Calling Sequence 3.3.8.2.1) all of which are handled by HMOO. The BIC is used for input and output operations. HMOO first sets up the BIC instructions by using the MTU number received in the call and saves the MTU number for later use in building the actual tape instruction. The starting buffer address is picked up from the call; the ending address is computed by using the number of words to be transferred in the call, and the BIC initial and BIC final addresses are output to the BIC. The tape function is picked up from the arguments and combined with the MTU number to create the tape instruction. The I/O operation is then executed. After completion of the I/O, HMOO checks the status of the MTU and returns that status to the calling program as the fifth parameter of the call.
3.3.8.2.1 Calling Sequence

CALL HMOO,A,B,C,D,STAT

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Address of location containing tape function to be performed.</td>
</tr>
<tr>
<td>0</td>
<td>input binary</td>
</tr>
<tr>
<td>1</td>
<td>input BCD</td>
</tr>
<tr>
<td>2</td>
<td>output binary</td>
</tr>
<tr>
<td>3</td>
<td>output BCD</td>
</tr>
<tr>
<td>4</td>
<td>Write End of File</td>
</tr>
<tr>
<td>5</td>
<td>Forward one record</td>
</tr>
<tr>
<td>6</td>
<td>Backspace one record</td>
</tr>
<tr>
<td>7</td>
<td>Rewind</td>
</tr>
</tbody>
</table>

| B | Beginning location of buffer (not to exceed 377778) |
| C | Address of location containing number of words to transfer |

| D | Address of location containing unit number (10 or 11) |

| STAT | Address of location containing Status of I/O operation. (Output by HMOO to calling program.) |

STAT = 20 I/O completed successfully. Number indicates number of words transferred.

STAT = -1 User requested I/O still in progress. (to be implemented)

3-150
3.3.8.2.1 Calling Sequence (Continued)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT = -2</td>
<td>End of File</td>
</tr>
<tr>
<td>STAT = -3</td>
<td>Tape Error</td>
</tr>
<tr>
<td>STAT = -4</td>
<td>End of Tape</td>
</tr>
<tr>
<td>STAT = -5</td>
<td>Beginning of Tape</td>
</tr>
</tbody>
</table>

- Busy Return
  - CALL+7
  - Handler busy processing a previous I/O request.

- Normal Return
  - Call +9
  - Requested I/O complete. User must check STAT for errors or number of words transferred.

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved</td>
<td>Same as Entry</td>
</tr>
<tr>
<td>B</td>
<td>Saved</td>
<td>Same as Entry</td>
</tr>
<tr>
<td>X</td>
<td>Saved</td>
<td>Same as Entry</td>
</tr>
<tr>
<td>Overflow</td>
<td>Not effected</td>
<td>Same as Entry</td>
</tr>
</tbody>
</table>
3.3.8.2.2 General Flow Chart

- HMOO
- **HANDLER BUSY?**
  - **YES**
  - <ref>A</ref>
- **NO**
  - **INITIALIZE PROPER BIC**
  - **BUILD UNIT AND FUNCTION IN HMFF**
  - **IS FUNCTION A TAPE POSITION OR REWIND COMMAND**
    - **YES**
    - <ref>B</ref>
    - **NO**
      - **LOAD BIC INITIAL AND FINAL**
      - **ACTIVATE BIC**
  - <ref>A</ref>
- **POSITION RETURN TO CALL+7**
- RETURN
3.3.8.3 Label Description

3.3.8.3.1 Local

HM1 - HM5, HM15  Locations of BIC instructions initialled by HM00

HM01  Location where tape I/O instruction is constructed and executed.

HMA  Save area for A-register.

HMAB  Indirect address for unit number.

HMA3  Initialize BIC 2 instruction. Used for initialization.

HMA4  Initialize BIC 4 instruction.

HMB  Save area for B-register.

HMBA  Beginning address of buffer. Used to compute number of words transferred.

HMBU  Previous I/O in progress (to be implemented).

HMB3  Sense BIC 2 not busy instruction. Used for initialization.

HMB4  Sense BIC 4 not busy instruction. Used for initialization.

HMC3  Set BIC 2 starting address instruction. Used for initialization.

HMC4  Set BIC 4 starting address instruction. Used for initialization.

HMD3  Set BIC 2 final address instruction. Used for initialization.

HMD4  Set BIC 4 starting address instruction. Used for initialization.

HME3  Enable BIC 2 instruction. Used for initialization.

HME4  Enable BIC 4 instruction. Used for initialization.

HMFF  Tape function being performed.

3-153
3.3.8.3.1 Local (Continued)

**HMF3** Location containing a "load BIC 2 initial register" instruction. Used for initialization.

**HMF4** Location containing a "load BIC 4 initial register" instruction. Used for initialization.

**HMUN** Unit number.

**HMX** Save area for X-register.

**HMXX** Indirect argument pointer.

3.3.8.3.2 Global

**HM00** Entry point into subroutine.

3.3.8.3.3 Entry Points

**HM00** primary entry point

3.3.8.3.4 External References

None.
3.3.8.4 Detailed Flow Chart

1. HM00
2. HMR S
   SAVE REGISTERS
3. HANDLER BUSY?
   YES → A
   NO
4. SET HM00 BUSY -1 HMBU
5. SET UP HMAB AS INDIRECT ADDRESS FOR UNIT NUMBER
6. UNIT NUMBER 10?
   NO
   YES → A
7. HMR R
   RESTORE REGISTERS
8. RETURN
9. HM30
   POSITION RETURN TO BUSY RETURN.
   CALL +7

3-155
BUILD EXC INSTRUCTION FROM HMFF

POSITION RETURN TO CALL+9

TAPE COMMAND

HMO3 PROCESS END OF OPERATION

RETURN

HMO3

SENSE STATUS OF APPROPRIATE MTU

SET STATUS FOR CALLING PROGRAM

SET HANDLER NOT BUSY

RETURN

3-156
HM1

INITIALIZE BIC

B IC NOT BUSY?

YES

PICK UP TAPE FUNCTION FROM ARGUMENTS

SHIFT FUNCTION FOR I/O COMMAND FORMAT

STORE FUNCTION IN HMFF

FUNCTION (HMFF) > 3?

YES

NO

C

C1
SET UP A REG WITH ENDING BUFFER ADDRESS

LOAD B WITH BEGINNING BUFFER ADDRESS

HM3
OUTPUT (OBR) BUFFER STARTING LOCATION TO BIC INITIALIZE

HM4
OUTPUT (OAR) BUFFER ENDING LOCATION TO BIC FINAL

HM5
ACTIVATE ENABLE BIC

C1

HMBB
SAVE UNIT NUMBER IN HMUN

D

3-158
SET USER STATUS BUSY (-1)

BUILD TAPE EXC INST FROM HMFF AND HMUN

STORE EXC INSTRUCTION AT HM01

POSITION HM00 FOR NORMAL RETURN

HM20 MTU BUSY

HMRR RESTORE REGISTERS

E
PROCESS END OF I/O OPERATION

OR UNIT NUMBER INTO SENSE INSTRUCTIONS

HM03

HM12 SENSE MTU REWINDING

YES

F

HMRW

SET USER STATUS TO 0

NO

HM04

SET USER STATUS TO -2

HM05

SENSE FILE MARK

YES

NO

G

3-160
I

HM16

SET HANDLER
NOT BUSY
0 → HMBU

HMRR
RESTORE
REGISTERS

RETURN

J

HM15

COMPUTE
NUMBER OF WDS
TRANSFERRED

I

HM20

CHECK MTU READY ROUTINE

OR UNIT NUMBER
INTO SENSE
INSTRUCTION

NO

SENSE
MTU READY

YES

RETURN

3-162
3.3.9 ID00 - Input Display

3.3.9.1 Purpose
This routine inputs a complete display from the CRT upon interrupt from the CRT keyboard.

3.3.9.2 Technical Description
This routine calls the CRT handler (HCOO) to input a display to the input buffer. After the routine is called and the arguments are retrieved, a character is input from the CRT controller buffer. This character must be an interrupt character caused by depressing the interrupt key on the CRT keyboard or a transmit (start of text) character caused by derpressing the transmit key on the CRT keyboard. An interrupt key will cause the cursor to be positioned to the end of the next heading on the displayed form. A start of text character causes an entire display to be input via routine HCOO. An initial character other than interrupt or transmit is ignored. If the transmission of data from the CRT to the input buffer is unsuccessful, the appropriate message stating the nature of the problem (parity error or device not ready) is output to the teletype.
Once the input buffer is being filled, the method of detecting the end of transmission varies. If the modem is employed, a delay of 4.5 seconds must occur to allow ample time for transmission to be completed. Otherwise, completed transmission may be sensed.

Routine HCIN is called to process the full input buffer. Upon return from HCIN, control is returned to the calling routine.
3.3.9.2.1 Calling Sequence

CALL ID00, NUM

PARAMETER

NUM the address of the number of characters to input

FUNCTION

REGISTER CONTENTS UPON ENTRY CONTENTS UPON EXIT

A destroyed

B destroyed

X destroyed

Overflow destroyed
3.3.9.2.2 General Flow Chart

IDOO

INPUT A DISPLAY FROM CRT

HCIN PROCESS INPUT BUFFER

RETURN
3.3.9.3 Label Description

3.3.9.3.1 Local Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNT</td>
<td>number of characters to input</td>
</tr>
<tr>
<td>HCRT</td>
<td>CRT device address</td>
</tr>
<tr>
<td>IDEC</td>
<td>error retry counter</td>
</tr>
<tr>
<td>IDRC</td>
<td>address of buffer to be filled</td>
</tr>
<tr>
<td>IDWD</td>
<td>number of words to send</td>
</tr>
<tr>
<td>ID2</td>
<td>CRT function selector</td>
</tr>
<tr>
<td></td>
<td>0 = input</td>
</tr>
<tr>
<td></td>
<td>1 = output with clear</td>
</tr>
<tr>
<td></td>
<td>2 = output with no clear</td>
</tr>
<tr>
<td>SCXX</td>
<td>cursor position indicator; 0 indicates start of new display</td>
</tr>
<tr>
<td>STAT</td>
<td>status of I/O operation</td>
</tr>
</tbody>
</table>

3.3.9.3.2 Global Labels

WRDS number of words to input from CRT

3.3.9.3.3 Entry Points

primary entry point ID00

3-168
3.3.9.3.4 External References

HCDL    delay routine
HCIN    I/O buffer processing routine
HCMM    enable interrupts
HCOO    input display routine
HCWW    delay routine for CRT write buffer to empty
OZKA    TTY print message routine
OZKR    carriage return, line feed to TTY
SCOO    cursor position routine
3.3.9.4 Detailed Flow Chart

1. **ID00**
   - **Retrieve Argument**
     - **ID 03**
     - **Input Character**
       - **Is Char. Interrupt?**
         - **Yes**
         - **ID04**
         - **No**
           - **Start of Text?**
             - **Yes**
             - **Initialize New Display Indicator**
             - **Output EOT Function**
             - **A**
             - **3-170**
3.3.10 IN00 - Initialization

3.3.10.1 Purpose
This routine reads the forms background tape into core and establishes
a forms table of pointers to the beginning of each form. After all forms
are established, the mounting of an output tape is requested via the
teletype and a message requesting the desired form number from the user
is displayed on the CRT.

3.3.10.2 Technical Description
After the input tape is rewound, the tape input buffer is blanked. One mag
tape record representing one form is then read. The form number is converted
from ASCII to binary. The beginning address of the form number input is then
stored in the forms table using the converted form number as an index.

When a tape read is unsuccessful, the message "TAPE ERROR" is printed on the
TTY. When end of file is encountered, the tape is rewound and the message
"MOUNT FORMS TAPE. REMEMBER WRITE RING. PRESS RUN WHEN TAPE IS READY." is printed and the computer halts. When an illegal form is input, the
message "ILLEGAL FORM NUMBER" is printed.

After the new output tape is mounted and the "run" button is pressed, the
mounted tape is rewound and the request for the desired form number is
displayed on the CRT. Return to Routine CPOO is then made.

3-175
3.3.10.2.1 Calling Sequence

CALL IN00

No parameters are passed in the call; the volatile registers are not saved.
3.3.10.2.2 General Flow Chart

INOO

READ FORMS INTO BUFFER

ESTABLISH FORMS POINTER TABLE

REQUEST OUTPUT TAPE

REQUEST DESIRED FORM NO.

RETURN
3.3.10.3 Label Description

3.3.10.3.1 Local Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>counter for mag tape routine</td>
</tr>
<tr>
<td>NC</td>
<td>number of characters to display</td>
</tr>
<tr>
<td>OP</td>
<td>code for mag tape operation to be performed</td>
</tr>
<tr>
<td>WRDS</td>
<td>maximum number of words to read from one mag tape record</td>
</tr>
<tr>
<td>LU</td>
<td>logical unit to be referenced for mag tape input and output</td>
</tr>
<tr>
<td>TBUF</td>
<td>buffer containing TTY message to mount forms tape</td>
</tr>
<tr>
<td>BLNK</td>
<td>2 blank ASCII character codes</td>
</tr>
<tr>
<td>PLAC</td>
<td>address of digits of form number</td>
</tr>
<tr>
<td>FRMB</td>
<td>&quot;ILLEGAL FORM NUMBER&quot; TTY message buffer</td>
</tr>
<tr>
<td>INTE</td>
<td>&quot;TAPE ERROR&quot; TTY message buffer</td>
</tr>
</tbody>
</table>

3.3.10.3.2 Global Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT</td>
<td>status of tape handler (HM00)</td>
</tr>
<tr>
<td>MTNS</td>
<td>tape input buffer</td>
</tr>
<tr>
<td>SAVE</td>
<td>temporary storage buffer</td>
</tr>
</tbody>
</table>

3.3.10.3.3 Entry Point

IN00 - Primary entry point
3.3.10.3.4 External References

HMOO - tape handler entry point
A100 - ASCII to integer conversion routine
OZKR - carriage return, line feed routine
OZKA - teletype print routine
OD00 - output to display routine
CPO0 - calling routine for IN00; control returns to CPO0 from IN00.
3.3.10.4 Detailed Flow Chart

- IN00
- REWIND FORMS TAPE
- CLEAR TAPE INPUT BUFFER TO BLANKS
- A IN30
- READ TAPE RECORD
- RECORD READ SUCCESSFULLY
- YES
- CONVERT FORM NUMBER TO BINARY
- B
- NO
- D
- HM00
- HM00
- HM00
- A100
B

VALID FORM?

NO

YES

TYPE "ILLEGAL FORM NUMBER"

OZKA

PLACE FORM NUMBER IN FORMS TABLE

COMPUTE BUFFER ADDRESS FOR NEXT READ

A
3.3.11 MDOO - Mag Tape Output Routine

3.3.11.1 Purpose
This routine outputs the contents of the buffer MTO$ to mag tape unit 10.

3.3.11.2 Technical Description
Immediately upon entry to the routine an attempt to write to the mag tape unit is made. When a successful write occurs, the contents of buffer MTO$ have been transferred to the tape. When the unit is not ready to write, an appropriate teletype message is output. When the number of words transferred does not equal the requested number, a teletype message is output and the write is attempted again.

3.3.11.2.1 Calling Sequence
CALL MDOO

No arguments are transferred.

Volatile registers are destroyed.
3.3.11.2.2 General Flow Chart

MDOO

OUTPUT BUFFER

MDOO
3.3.11.3 Label Description

3.3.11.3.1 Local Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDEM</td>
<td>error message buffer for &quot;unit not ready&quot;</td>
</tr>
<tr>
<td>MDEQ</td>
<td>error message buffer for &quot;invalid number of words transferred&quot;</td>
</tr>
<tr>
<td>MDF</td>
<td>tape handler function - output binary record</td>
</tr>
<tr>
<td>MDF1</td>
<td>tape handler function - backspace one record</td>
</tr>
<tr>
<td>MDST</td>
<td>tape handler status indicator</td>
</tr>
<tr>
<td>MDUN</td>
<td>logical unit number</td>
</tr>
<tr>
<td>MDW</td>
<td>number of words to output</td>
</tr>
</tbody>
</table>

3.3.11.3.2 Global Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO$</td>
<td>tape output buffer</td>
</tr>
</tbody>
</table>

3.3.11.3.3 Entry Points

Primary entry point - MDOO

3.3.11.3.4 External References

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>HM00</td>
<td>mag tape handler</td>
</tr>
<tr>
<td>OZKA</td>
<td>TTY output message</td>
</tr>
<tr>
<td>OZKR</td>
<td>TTY carriage return and line feed</td>
</tr>
</tbody>
</table>

3-185
3.3.12 ODOO - Output Display

3.3.12.1 Purpose

ODOO is a subroutine whose purpose is to output a buffer of data (CRO$) to the CRT. This buffer may be any length up to 957 words.

3.3.12.2 Technical Description

Communication with the CRT requires that all message buffers to be transmitted be preceded and followed by special initiation and termination codes respectively. ODOO accepts message buffers from the calling routine, attaches the overhead (initiation and termination codes) information and calls HCOO to clear the screen and output the buffer. (A sample message buffer with the overhead is shown in Figure 3-2.)

The next action is based upon the results of the message transmission.

- If the message transmission is error free, control is returned to the calling routine.

- If the CRT is not ready, the message "CRT NOT READY" is printed at the teletype and the program halts. Depressing the RUN button on the computer console will cause the program to again attempt to output.

- If a parity error occurs, an appropriate error message is typed at the teletype and an automatic retry of the transmission is made. The number of retries is unlimited and an error message is given with each attempt.
3.3.12.2.1 Calling Sequence

CALL ODOO,NO

PARAMETER FUNCTION

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>A memory location whose contents specify the number of words contained in the buffer (CRO$).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>B</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>X</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Overflow</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buffer Character Position</td>
<td>Character</td>
<td>Function</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>SOH</td>
<td>Overhead</td>
</tr>
<tr>
<td>2</td>
<td>Halt Local I/O</td>
<td>Initiation Codes</td>
</tr>
<tr>
<td>3</td>
<td>SOH</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Select Alpha Mode</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SOH</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Enable Refresh</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cursor Reset</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Cursor Reset</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>SOH</td>
<td>Overhead</td>
</tr>
<tr>
<td>29</td>
<td>Enable Transmit</td>
<td>Termination Codes</td>
</tr>
<tr>
<td>30</td>
<td>End of Transmission</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>End of Transmission</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-2 Sample Output Buffer
3.3.12.2.2 General Flow Chart

1. DO
2. PICK UP CALLING SEQUENCE
3. MOVE OUTPUT CODES TO END OF BUFFER
4. HCOO
   OUTPUT BUFFER TO CRT
5. RETURN
3.3.12.3 Label Description

3.3.12.3.1 Local

CONU - (12,1,8,4,4) buffer containing termination codes which must follow each buffer that is output to the CRT.

CONV - (12) total number of control codes preceding and following the user's buffer.

CONW - (4) number of control codes less one which follow the output buffer.

NUM - temporary storage location for the address of the number of words in the output buffer.

NUMT - storage location for the total number of words to be output, including all control codes.

X - (1) function code to be passed to HCOO indicating output after clearing CRT screen.
3.3.12.3.2 Global

CNT - temporary storage location for the address of the number of words to input from the CRT.

DEV - ("CRT NOT READY") error message output to teletype in the event the CRT device is not ready.

PARE - ("PARITY ERROR FROM DISPLAY DEVICE") error message output to teletype in the event of a parity error from the CRT.

3.3.12.3.3 Entry Point

OD00 - primary entry point.

3.3.12.3.4 External References

CONT - Beginning location of initiation codes which must immediately precede output buffer.

CRO$ - CRT output buffer

HC00 - CRT handler for input and output

OZKA - teletype routine that outputs a buffer to the teletype

OZKR - teletype driver to output one carriage return and one line feed

STAT - status word from the CRT handler HC00.

WRDS - storage location for the number of words to input from the CRT.
3.3.12.4 Detailed Flow Chart

ODOO

SAVE NUMBER OF CHARACTERS TO TRANSMIT

PLACE SPECIAL CONTROL CODES AT END OF OUTPUT BUFFER (CRO$)

ADD NUMBER OF SPECIAL CODES TO NUMBER OF CHARACTERS IN OUTPUT BUFFER TO GET COUNT NUM + CONV

A

OD10

HCOO

INITIATE OUTPUT

STATUS (STAT) = -1

YES

NO

B
YES

OZKA
OUTPUT "CRT NOT READY"

A

C

OZKA
OUTPUT "PARITY ERROR DISPLAY DEVICE"

A

3-194
### 3.3.13 OZKA - Print Buffer on TTY

#### 3.3.13.1 Purpose

This routine outputs characters to the ASR-35 teletype.

#### 3.3.13.2 Technical Description

This routine outputs a buffer to the ASR-35 teletype. Each buffer word should contain two eight bit ASCII characters. Both the beginning location to be output and the number of characters to print are specified by the calling routine.

#### 3.3.13.2.1 Calling Sequence

CALL OZKA,START,N

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>beginning address to be printed</td>
</tr>
<tr>
<td>N</td>
<td>number of characters to print</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>B</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>X</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>Overflow</td>
<td>Destroyed</td>
<td></td>
</tr>
</tbody>
</table>

3-195
DETERMINE BEGINNING ADDRESS TO BE PRINTED AND NUMBER OF CHARACTERS TO PRINT

PRINT BUFFER

RETURN
3.3.13.3 Label Description

3.3.13.3.1 Local Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OZ$1</td>
<td>contains address of next word to be output</td>
</tr>
</tbody>
</table>

3.3.13.3.2 Global Labels

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE</td>
<td>buffer used to save volatile registers</td>
</tr>
</tbody>
</table>

3.3.13.3.3 Entry Point

primary entry point - OZKA

3.3.13.3.4 External References

OZKC - routine which outputs one character to the TTY
3.3.13.4 Detailed Flow Chart

OZKA

SAVE REGISTERS

PICKUP ARGUMENTS: BUFFER ADDRESS AND WORD COUNT

A

OUTPUT A CHARACTER

ALL CHARACTERS OUT?

YES

RETURN

NO

OZKC
3.3.13.4 Detailed Flow Chart

OZKA

SAVE REGISTERS

PICKUP ARGUMENTS: BUFFER ADDRESS AND WORD COUNT

A

OUTPUT A CHARACTER

ALL CHARACTERS OUT?

YES

RETURN

NO

OZKC
3.3.14 OZKC - Output Character to TTY

3.3.14.1 Purpose
This routine outputs one character to the ASR-35 teletype.

3.3.14.2 Technical Description
NA

3.3.14.2.1 Calling Sequence
CALL OZKC

The A-register contains the character to be output.
3.3.14.2.2 General Flow Chart

OZKC

OUTPUT ONE CHARACTER

RETURN
3.3.14.3 Label Description

3.3.14.3.1 Local Labels
None

3.3.14.3.2 Global
None

3.3.14.3.3 Entry Point
Primary entry point OZKC

3.3.14.3.4 External References
None
3.3.14.4 Detailed Flow Chart

OZKC

DEVICE READY TO WRITE

NO

YES

OUTPUT A-REGISTER TO ASR-35

RETURN

3-203
3.3.15 OZKR - Output Carriage Return and Line Feed.

3.3.15.1 Purpose
This routine outputs a carriage return and a line feed each time it is called.

3.3.15.2 Technical Description
NA

3.3.15.2.1 Calling Sequence
CALL OZKR

No arguments are passed; no registers are used to transfer information.
3.3.15.2.2 General Flow Chart

OZKR

OUTPUT CARRIAGE RETURN

OUTPUT LINE FEED

RETURN
3.3.15.3 Label Description

3.3.15.3.1 Local
None

3.3.15.3.2 Global
SAVE - used to save the A-register.

3.3.15.3.3 Entry Point
Primary entry point - OZKR

3.3.15.3.4 External References
None
3.3.15.4 Detailed Flow Chart

- OZKR
- Save A-Register
- Ready to Write?
  - No
  - Yes → Output Carriage Return
  - Ready to Write?
    - No
    - Yes → Output Line Feed
- Restore A-Register

RETURN

3-207
3.3.16 PINT - Power Fail - Restart

3.3.16.1 Purpose
This routine provides protection against program failure whenever a physical power failure occurs.

3.3.16.2 Technical Description
When a power failure occurs, the location counter and the volatile registers are saved and a halt occurs. Upon restoration of adequate power, the volatile registers are reinstated and processing resumes at the address specified by the value saved for the location counter.

3.3.16.2.1 Calling Sequence

CALL PINT

PARAMETER FUNCTION
None

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS UPON ENTRY</th>
<th>CONTENTS UPON EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>B</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>X</td>
<td>Saved</td>
<td>Restored</td>
</tr>
<tr>
<td>Overflow</td>
<td>Saved</td>
<td>Restored</td>
</tr>
</tbody>
</table>
3.3.16.2.2 General Flow Chart

POWER FAIL

PINT

SAVE REGISTERS

HALT 777

RESTORE REGISTERS

PINT
3.3.16.3 Label Description

3.3.16.3.1 Local
AR Contents of A-register
BR Contents of B-register
XR Contents of X-register
OF Contents of overflow indicator

3.3.16.3.2 Global
None

3.3.16.3.3 Entry Points
PINT - entry for power failure
PIN2 - entry for power restoration

3.3.16.3.4 External References
None
3.3.16.4 Detailed Flow Chart

- **PINT**
- **SAVE REGISTERS**
- **SET ALL REGISTERS = ALL 1'S**
- **HALT 777**
- **PINT**

- **PIN2**
- **RESTORE REGISTERS**
3.3.17 SCOO - Set Cursor

3.3.17.1 Purpose

SCOO is a subroutine whose purpose is to position the CRT cursor to the next answer line on the MEDATA input forms.

3.3.17.2 Technical Description

Feedback from users of the MEDATA input system showed that manually positioning the CRT cursor from question to question proved to be both cumbersome and time consuming. SCOO automatically positions the cursor to one of several positions for the current CRT data line, depending on the data conditions. Figure 3-3 shows the data conditions which may exist and the corresponding cursor positioning for each condition. The user positions the cursor to the next data line by depressing the interrupt key on the keyboard. If the interrupt key is depressed when the cursor is already positioned to the bottom line of the input form, the cursor is automatically positioned back to the top line.
**DATA CONDITION**

1. New line character detected before the tab position (Column 21) and after a colon on this line.

2. New Line character found before a colon on this line.

3. Colon supercedes tab and first non-space character after colon is a left parenthesis.

4. Colon precedes tab and all spaces between the colon and tab.

5. Colon precedes tab and at least one non-space character between the colon and tab.

6. Colon precedes tab and first non-space character after colon is left parenthesis.

7. Colon supercedes tab and first non-space character after colon unequal left parenthesis.

8. Colon supercedes tab and all spaces following colon.

**CURSOR POSITION**

position to new line character

advance to next line

position to character immediately following the left parenthesis.

position to tab

position to character immediately following the colon.

position to character immediately following left parenthesis.

position to character immediately following the colon.

position to character immediately following the colon.

Figure 3-3 Cursor Positioning Criteria
3.3.17.2.1 Calling Sequence

CALL SCO00, SCXX

PARAMETER

SCXX

FUNCTION

if SCXX is equal to zero, the cursor is to be reset to the first answer area on the page and if SCXX does not equal zero, the cursor is to be set to the answer area on the next data line.

REGISTER

A
B
X

Contents upon entry

saved upon entry
saved upon entry
saved upon entry

Contents upon exit

restored upon exit
restored upon exit
restored upon exit

Overflow

N/A

N/A
3.3.17.2.2 General Flow Chart

- **SC00**
- **NEW PAGE?**
  - **YES**
  - **INITIALIZE COUNTERS, POINTERS**
  - **A**
  - **END OF BUFFER?**
    - **YES**
    - **SET UP CURSOR POSITION PARAMETERS**
    - **B**
    - **NO**
      - **NO**
      - **CONTROL CHARACTER**
      - **YES**
      - **PICK UP DATA CHARACTER**
      - **NO**
      - **COLOM ON THIS LINE?**
    - **NO**
      - **NO**
      - **POSITION TO END OF LINE CHARACTER**
      - **A**
      - **YES**
3.3.17.3 Label Description

3.3.17.3.1 Local

ASTAT  storage location containing the status of an I/O operation

SCA  storage location used for storing the original contents of the
      A-register

SCB  storage location used for storing the original contents of the
      B-register

SCCOL  storage location which contains the number of the current column
        being processed

SCCSAV  storage location used to save the column position whenever a cursor
        reset control character is detected

SCDATA  flag which is set whenever an answer to a particular question is
        detected

SCEM  (1,5,0,0,010,4) CRT parameters necessary to initiate a cursor reset

SCKOL  flag which is set whenever a colon is found on a question line

SCNW  (7) storage locations containing the number of control parameters
       needed to initiate a cursor reset

SCROW  storage location which contains the number of the current row being
        processed

SCRSAV  storage location used to save the row position whenever a cursor
        reset control character is detected

SCTAB  flag which is set whenever the tab column position (column 21) has
        been processed for a data line

SCX  storage location used for storing the original contents of the
     X-register
3.3.17.3.2 Global
N/A

3.3.17.3.3 Entry Points
SCOO - primary entry point

3.3.17.3.4 External References

3.3.17.3.4.1 Labels
CRO$CRT Output Buffer
DPT$display output buffer pointer. This pointer is used in SCOO
to determine when a buffer has been processed.
SCXXcounter which tells the routine if a new CRT page is to be processed

3.3.17.3.4.2 Subroutines
HCDLTime Delay
HCOOCRT Handler
3.3.17.5 Detailed Flow Chart

1. ENTER
2. STORE REGISTERS
3. BEGIN NEW PAGE?
   - NO: A
   - YES: INITIALIZE ROW AND COLUMN COUNTERS
4. CLEAR TOP OF PAGE POINTER
5. B
A

initialize column counter

increment row counter

load data character

increment buffer pointer count

end of line? no

YES

B

3-220
IS CHARACTER A ":"?
YES
SAVE ROW AND COLUMN POSITIONS
SET COLON FOUND IN FLAG
F

E
LOAD DATA CHARACTER
END OF LINE?
YES
INCREMENT ROW COUNT
NO
INCREMENT COLUMN COUNT
INCREMENT BUFFER POINTER COUNT
F
INCREMENT COLUMN COUNT

B
3-222
K

SAVED COLUMN COUNT > TAB? NO L

YES

LOAD DATA CHARACTER

IS CHARACTER A ')'? NO J

YES

POSITION CURSOR TO CURRENT COLUMN

P
4.0 PROGRAM UTILIZATION

The Varian MEDATA input system (MDIS) is a user-oriented medical information storage and retrieval system designed to permit an on-line question and answer environment. The use of Cathode Ray Tube (CRT) displays permits easy viewing, on-line data form preparation, and subsequent corrections or modifications in the preparation of the medical data forms. The medical data forms are maintained on magnetic tape and are read into computer memory at run time. The user selects the required form and the background is displayed on the CRT. As each form is completed, the data is stored on magnetic tape. This tape may be either updated, transmitted to another terminal, or input to the Building 12 UNIVAC 1108 MEDATA system. The user has the capability of producing hardcopies for any form requested. All background forms are maintained on punched cards and any form can be modified or new forms added by changing the card deck and creating a new background forms tape with an "off-line" utility program.

4.1 COMPUTER OPERATOR INSTRUCTIONS

The following procedures should be followed by the computer operator in preparing MDIS for use:

MDIS OPERATIONAL SETUP PROCEDURES

- Load the MDIS program from either paper tape or magnetic tape.
- Turn on the CCI 301 Display Controller by setting the OFF/ON switch to the ON position.
- Turn on the CRT by pulling the OFF/ON switch.

After allowing several seconds for the unit to warm up, press the MASTER CLEAR and the CLEAR keys on the keyboard in that order. The cursor will appear on the screen.
• Turn on the Magnetic Tape Unit by setting the OFF/ON switch to the ON position.
• Mount the user specified master forms tape. Each set of forms (Surveillance, Flight Crew, etc.) is stored on one tape, hence the user must specify which set he wants.
• Begin program execution at location zero (0) by typing in "GO" on the teletype keyboard.
• The program will respond with the message "MOUNT OUTPUT TAPE. REMEMBER WRITE RING. PRESS RUN WHEN TAPE IS READY." Mount a scratch tape with a write ring attached and position the tape to load point.

MDIS OPERATIONAL WRAPUP PROCEDURES
• After the user has completed filling out the forms, the computer will print the message "TERMINAL INPUT COMPLETE, LOG OUTPUT TAPE, CALL PHILCO". Dismount the tape, label the tape, and notify Philco-Ford that the tape is ready for further processing.

4.2 OPERATIONAL PROCEDURES

MAKING CONTACT WITH THE COMPUTER
After the steps in Section 4.1 have been completed by the operator, the user must begin by typing in FORM XXX in the upper left hand corner of the screen where XXX is the number of the form desired. Then press the TRANS key and wait for a response from the computer. The first page of the form called will appear on the CRT screen. Each set of forms (Surveillance, Flight Crew, Laboratory, etc.) is stored on one tape. The computer operator must know which set of forms the user wants during system preparation time.
FORM AND PAGE ADVANCING

Most forms consist of several pages of full CRT screens of data. To advance from one page to the next, simply press TRANS. An XX appears after the page number when the last page for that form appears. After the last page has been transmitted back to the computer by pressing TRANS, the entire background form, including questions and answers, will be output to the magnetic tape. The computer will then cycle back to the first page of the same form. A new form may be requested by changing the form number to the form required and pressing TRANS.

ERROR CORRECTION AND MODIFICATION

Because the entire page is held in the CCI 301 Display Controller and is not transmitted to the computer until the user presses TRANS, it is always possible to correct errors before transmitting the page to the computer. To erase data, it is necessary only to position the cursor to the character to be erased and press the space bar. To replace data, position the cursor and type over the data. To refresh the current background, place an X in the page number and press TRANS. This will delete all answers and refresh all headings and questions. If the X option is used, no data is stored in the computer for the transmitted page. To reinitialize to the first page of the current form, place an X in the first position of the form number and press TRANS. This option works just like the refresh page option except that it blanks out the answers for the entire form and goes back to page one of the form.
CURSOR POSITIONING

Pressing the INT key will automatically position the cursor down to the next answer area on the page. If INT is pressed while the cursor is positioned at the bottom line of the page the cursor will recycle to the first answer area on top of the page.

EXTRA LINE CAPABILITY

In the case where the user desires to enter an answer which is longer than the allocated answer space, pressing the asterisk key followed by the TRANS will cause the CRT screen to be cleared and the cursor positioned to the home position. The user can then type in his extra data, and on completion, press the LINE and TRANS keys in that order. The computer will output the remaining unanswered portion of the current page being processed.

IDENTIFICATION CARRY-OVER

The Social Security Number and Date is carried over from page to page and form to form. This eliminates the requirement for the user to retype repetitious information page after page.

HARDCOPY CAPABILITY

To obtain a hardcopy of the form to be printed out at the computer, the following procedure is followed: after having transmitted the form and cycled to page one of that form, place a "D" in the first position of the form number and press TRANS. The computer will backspace the output tape and read one record to obtain the last form that was just completed and type it out on the teletype. When the form is completely printed, the computer will display FORM on the screen. Type in the desired form number and press TRANS to continue filling out forms.
PROGRAM TERMINATION

To terminate the program, change the form number to "***" and press TRANS. The computer will write an End of File on the output tape, rewind the tape, and type out terminating instructions to the operator.

4.3 INPUT DESCRIPTION

A magnetic tape, containing the background forms, is created off-line from data cards coded by the system user. This method permits easy design of new forms and modifications to old ones. Each form data card contains a sequence number (if required), question heading, answer area, and any control characters required by MEDATA. The magnetic tape containing the background forms is read into the computer at run time. After the required form has been read into the computer and displayed on the CRT, the user inputs his answers into the corresponding answer areas.

4.4 OUTPUT DESCRIPTION

4.4.1 Reports

The system contains the option of obtaining a hard copy of the forms being output to magnetic tape by dumping the data forms out to the on-line Teletype. See Appendix E for a sample form.

4.4.2 Tapes

The system creates, as output, a magnetic tape containing the user input medical data forms in a format acceptable to the Building 12 UNIVAC 1108 MEDATA system. All control characters required by the MEDATA system
have been added to each individual tape record before output. After
generation of the data has been completed, the tape is delivered to the
Data Reduction Computer of the Computation and Analysis Division. Here
the data is merged with the data contained on the MEDATA master file.

4.4.3 Messages

4.4.3.1 Diagnostics

<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>EXPLANATION</th>
<th>CORRECTIVE MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLEGAL FORM NUMBER</td>
<td>User has called for an invalid form</td>
<td>Input another form number and press TRANS</td>
</tr>
<tr>
<td>ERROR LINE</td>
<td>User has accidentally erased or typed over a colon or right parenthesis</td>
<td>Replace the colon or parenthesis</td>
</tr>
<tr>
<td>PLEASE SELECT</td>
<td>An error has been detected in the background form which will not allow the form to be displayed properly</td>
<td>Select another form and press TRANS</td>
</tr>
<tr>
<td>ANOTHER FORM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4-6
### 4.4.3.1 Diagnostics (Continued)

<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>EXPLANATION</th>
<th>CORRECTIVE MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER LENGTH ERROR</td>
<td>Occurs in the case where the length of the Tape Output Buffer (MTO$) has been exceeded</td>
<td>Key in the letters &quot;FORM&quot; in the upper left hand corner of the CRT screen and select a new form</td>
</tr>
</tbody>
</table>

### 4.4.3.2 Advisory

<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>USER ACTION</th>
<th>CORRECTIVE MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM</td>
<td>Input the number of the required form</td>
<td>Dismount master tape and notify Philco that processing has been completed</td>
</tr>
<tr>
<td>TERMINAL INPUT COMPLETE, LOG OUTPUT TAPE, CALL PHILCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECORD COUNT IS --</td>
<td>A count of the number of data records processed</td>
<td></td>
</tr>
<tr>
<td>MOUNT OUTPUT TAPE, REMEMBER WRITE RING. PRESS RUN WHEN TAPE IS READY</td>
<td>Remind the user to mount the output master tape before proceeding</td>
<td></td>
</tr>
<tr>
<td>PARITY ERROR FROM DISPLAY DEVICE</td>
<td>No user action required. Program will retry.</td>
<td></td>
</tr>
</tbody>
</table>
4.4.3.2 Advisory

MESSAGE

USER ACTION

CRT NOT READY

Ready the CRT and then press RUN on the computer console

TAPE UNIT NOT READY

Ready the magnetic tape unit

NUMBER OF WORDS TRANSFERRED IN ERROR

This message will occur only in the case of a hardware error. The program will retry the write.

CRT DEVICE ERROR

No user action required. The program will retry the Input/Output operation

4.5 RESTRICTIONS

Following are a list of commonly made user errors in operating the system.

- **Use of space bar to position cursor.** Do not use the space bar to position the cursor. If the cursor is under a display character and the user presses "space", it will replace the character with a space. Use the keys which are provided for cursor positioning.

- **Placing data after the new line Character "\"."**

A new line character "\"." has been placed at the end of each line as it went out to the CRT from the computer. If the user wishes to extend that line, type over the line character and place a new one at the end of the line by pressing the LINE key. Note that no characters placed after the new line character on any one display line are transmitted to the computer.
4.5 RESTRICTIONS (Continued)

- Changing the form number before transmitting the last page of data on a particular form. In order for the data typed in on a form to be recorded on tape at the computer, the user must press TRANS for the last completed page and let the system cycle back to page one before changing the form number.

- Restricted use of $, L, *, and <. These four special characters are used for control purposes only and may not be used in the body of an answer.
APPENDIX A

INPUT AND OUTPUT TAPES

DESCRIPTIONS

INPUT TAPE

FILE

The input data tape is composed of one multi-record file. As Figure A-1 illustrates, that file is composed of N variable length records.
Each tape record contains the background data for an entire form. Figure A-2 shows the record structure. In general, it is composed of two distinct sections, each of variable length:

(1) Identification Section
(2) Body Section.

Before a thorough understanding of the tape record is possible, it is necessary to study the following definitions.

Item An item is all the data associated with one background question, as well as the descriptive codes related to the background.

A-2
<table>
<thead>
<tr>
<th><strong>Input Tape (Continued)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level code</strong></td>
</tr>
<tr>
<td>This is a one digit code between zero and nine inclusive, found within an Item, and it specifies the position or rank in a hierarchical or outline structure of the Item. (i.e., the number of spaces to be indented).</td>
</tr>
<tr>
<td><strong>Sequence number</strong></td>
</tr>
<tr>
<td>This is a digit from one to four inclusive. It indicates the order in which the ID Items are to be saved on the Master File. The sequence number four must be assigned to Item 4.</td>
</tr>
<tr>
<td><strong>Block character</strong></td>
</tr>
<tr>
<td>(243₈) The purpose of the block or end-of-block character is to separate all Items on a record.</td>
</tr>
<tr>
<td><strong>End-of-page character</strong></td>
</tr>
<tr>
<td>(244₈) This character may follow any Item in the Body Section of the record, and it serves to delimit CRT display data &quot;pages&quot;. These pages are the data necessary to fill a CRT screen.</td>
</tr>
<tr>
<td><strong>Heading-Answer pair</strong></td>
</tr>
<tr>
<td>This is the main component of a data record Item. It will be in one of two forms: Heading and Answer, or Heading alone. If there is an Answer with the Heading, it is preceded by a colon. The number of characters in a heading plus the value of the level code cannot exceed twenty-four. (Thus, if the level code is four, the Heading cannot exceed twenty characters.) The Answer in the Identification Section cannot exceed seventy-two characters; the Answer in the Body Section may be any length.</td>
</tr>
</tbody>
</table>
The Identification Section of a data tape record is made up of four Items preceded by a block character, as shown in Figure A-3. For each Heading-Answer pair in an Item, there is a level code and a sequence number associated with it. The level code is always zero. All possible sequence numbers (1, 2, 3, 4) must occur in ID Items. They may occur in any order with one exception. Sequence number 4 must be associated with Item 4. The only other restriction in the ID Section is that the Heading associated with Item 4 must be DATE.
The Body Section consists of a variable number of Items. A Body Item is like an ID Item except it contains no sequence number, and its Answer may be longer. If the Answer portion is to be longer than one line, consecutive block characters are used, one for each line.
The output data tape is composed of one multi-record file. As Figure A-5 illustrates, that file is composed of N fixed length records, a software end-of-file record, and a hardware end-of-file mark.
DATA RECORD

Each tape record contains the background data and input data for an entire form. While a record is a fixed length of 5760 ASCII characters, the data for the form may be somewhat shorter, in which case, the remainder of the 2880 word record is blank filled. Figure A-6 shows the record structure. In general, it is composed of two distinct sections, each of variable length:

1. Identification Section
2. Body Section.

These two sections are very similar to the input tape record. The definitions for the output record are the same as those of the input record with the following exceptions and additions:
OUTPUT TAPE (Continued)

Block character

End-of-page character

End-of-record (137) This character is the last data character of a form. Since the record is fixed length, all words following the end-of-record character are blank filled.
The Identification Section of a data tape record is made up of four Items preceded by a block character, and followed by the alphabetic character "A", as shown in Figure A-7. For each Heading-Answer pair in an Item, there is a level code and a sequence number associated with it. The level code is always zero. All possible sequence numbers (1,2,3,4) must occur in ID Items. They may occur in any order with one exception. Sequence number 4 must be associated with Item 4. The only other restriction in the ID Section is that the Heading associated with Item 4 must be DATE.
BODY SECTION

The Body Section consists of a variable number of Items, followed by an end-of-record character. A Body Item is like an ID Item except it contains no sequence number, and its Answer may be any length. If the Answer is longer than one line, a carriage return character is stored at the end of the line.
END-OF-FILE RECORD

The last record on the file is the end-of-file record. It is used as a software end-of-file and consists of an end-of-record character in the first word of the 2880 word record. The remainder of the record is ignored.
The data displayed at the CRT is composed of two data types and two special CRT characters. The data types are:

1. Blind headings. A series of 24 or less characters which is used as a descriptor for the heading-answer pairs that follow.

2. Heading-answer pair. This data type is composed of a heading of 24 or less characters and an answer of variable length. The answer field may be filled, or it may be made up of one or more blank lines.

The two special characters are:

1. LINE. (\textbackslash ) \textasciitilde 12 \textsubscript{8} This character is found at the end of most lines. Any data following this character and on the same line will be ignored.

2. ETX. (\textasciitilde ) \textasciitilde 3 \textsubscript{8} This character may be input by the user from the CRT keyboard. Any data following this character, whether it be on the same line or subsequent lines, will not be input to the computer. In the event this character is not displayed by the user, the very last code to be input to the user is this character code. Thus, in all cases, it is the last character to be transmitted.
Appendix C

LTBS Line Table Format

Each display line has a corresponding Line Table word. The format of each word is as follows:

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 5</td>
<td>Level code</td>
</tr>
<tr>
<td>6 - 11</td>
<td>Sequence number</td>
</tr>
<tr>
<td>12</td>
<td>Two question bit. If set, it indicates that the next word contains line table data on a question that is also on the current line.</td>
</tr>
<tr>
<td>13</td>
<td>More lines bit. If set, it indicates that the answer field requires more than one line, and the following line is a continuation of that field.</td>
</tr>
<tr>
<td>14</td>
<td>No answer bit. If set, it indicates that the current line contains a blind heading.</td>
</tr>
<tr>
<td>15</td>
<td>Continuation line bit. If set, it indicates that the current line may contain data that is merely a continuation of the answer field of the previous line.</td>
</tr>
</tbody>
</table>
APPENDIX D
SAMPLE INPUT

SAMPLE TAPE INPUT

FORM 001 PAGE 01 MEDATA#
02RECORD: REGISTRY#
03TYPE: IDENTIFICATION#
01SS NO:
04DATE: #
01NAME: #
01PURPOSE: ( )#
#
01BLOOD TYPE: #
01RACE: #
01SEX: #
01HANDEDNESS: #
01SHOE SIZE: ( )#
01BIRTHDATE: #
01BIRTHPLACE: #
01POSITION: #
01AGENCY: #
01ORGANIZATIONAL UNIT: #
01GOVERNMENT SERVICE#
1MILITARY#
2FROM: ( )#
2TO: #
1CIVILIAN#
2FROM: ( )#
2TO: #
$FORM 001 PAGE 02 XX MEDATA#
0RATING OR SPECIALTY: #
#
1TIME IN THIS CAPACITY#
2TOTAL: ( )HRS#
2LAST 6 MO: ( )HRS#
0PRESENT HOME ADDRESS: #
#
0NEXT OF KIN: #
1RELATIONSHIP: #
1ADDRESS: #
#
0PHYSIOLOGICAL TRAINING: #
0OTHER INFORMATION: #
0COMMENTS: #
#
0TYPES: #
$ +

# - block character
\ - tab character
$ - end-of-page character
< - end-of-form character

This printout shows data by lines. On the tape, all data is packed.
SAMPLE CRT INPUT

FORM 001 PAGE 001
RECORD: REGISTRY
TYPE: IDENTIFICATION
SS NO:
DATE: 01JUN71
NAME:
PURPOSE: (71) DEMONSTRATION

BLOOD TYPE:
RACE: CAUC
SEX: FEM
HANDEDNESS: LEFT
SHOE SIZE: (7C)
BIRTHDATE: 23MAY45
BIRTHPLACE: LOS ANGELES, CALIFORNIA

POSITION:
AGENCY:
ORGANIZATIONAL UNIT:
GOVERNMENT SERVICE
MILITARY
FROM:
TO:
CIVILIAN
FROM: (67)
TO: PRESENT
APPENDIX E

SAMPLE OUTPUT

SAMPLE TAPE OUTPUT

$ 02RECORD: REGISTRY$
03TYPE: IDENTIFICATION$
01SS NO: [Redacted]
04DATE: 01JUN71$
AO NAME: 
0 PURPOSE: (71)DEMONSTRATION$
$ 0 BLOOD TYPE: 
0 RACE: CAUC$
0 SEX: FEM$
0 HANDEDNESS: LEFT$
0 SHOE SIZE: (7C )$
0 BIRTHDATE: 23MAY45 $ 0 BIRTHPLACE: LOS ANGELES, CALIFORNIA $ 0 POSITION: $ 0 AGENCY: 
0 ORGANIZATIONAL UNIT: $ 0 GOVERNMENT SERVICE: 1 MILITARY$
 2 FROM: ( )$ 2 TO: $ 1 CIVILIAN$
 2 FROM: (67)$ 2 TO: PRESENT$ 1$ 0 RATING OR SPECIALTY: DIETICIAN$
$ 1 TIME IN THIS CAPACITY$
 2 TOTAL: ( )HRS$ 2 LAST 6 MO: ( )HRS$ 0 PRESENT HOME ADDRESS: 1429 LAPLACE$
 HOUSTON, TEXAS 77043$ 0 NEXT OF KIN: 
1 RELATIONSHIP: 
1 ADDRESS: 
$ 0 PHYSIOLOGICAL TRAINING: $ 0 OTHER INFORMATION: $ 0 COMMENTS: 
$ 0 TYPED: JB/NRD 1$

$ - block character
| - end-of-page character
← - end-of-record character

This printout shows data by lines.
On the tape, all data is packed.

E-1
SAMPLE CRT OUTPUT

FORM 001 PAGE 01 MEDATA
RECORD: REGISTRY
TYPE: IDENTIFICATION
SS NO:
DATE:
NAME:
PURPOSE: ( )

BLOOD TYPE:
RACE:
SEX:
HANDEDNESS: ( )
SHOE SIZE: ( )
BIRTHDATE:
BIRTHPLACE:
POSITION:
AGENCY:
ORGANIZATIONAL UNIT:
GOVERNMENT SERVICE
MILITARY
FROM: ( )
TO:
CIVILIAN
FROM: ( )
TO:

E-2
SAMPLE TELETYPExE OUTPUT

RECORD: REGISTRY
TYPE: IDENTIFICATION
SS NO: 
DATE: 01JUN71
NAME: 
PURPOSE: (71) DEMONSTRATION
BLOOD TYPE: 
RACE: CAUC
SEX: FEM
HANDEDNESS: LEFT
SHOE SIZE: (7C )
BIRTHDATE: 23MAY45
BIRTHPLACE: LOS ANGELES, CALIFORNIA
POSITION: 
AGENCY: 
ORGANIZATIONAL UNIT: 
GOVERNMENT SERVICE
MILITARY
FROM: 
TO: 
CIVILIAN
FROM: (67)
TO: PRESENT
RATING OR SPECIALTY: DIETICIAN
TIME IN THIS CAPACITY
TOTAL: ( )HRS
LAST 6 MO: ( )HRS
PRESENT HOME ADDRESS: 1429 LAPLACE
HOUSTON, TEXAS 77043
NEXT OF KIN: 
RELATIONSHIP: 
ADDRESS: 
PHYSIOLOGICAL TRAINING: 
OTHER INFORMATION: 
COMMENTS: 
TYPED: JB/NRD