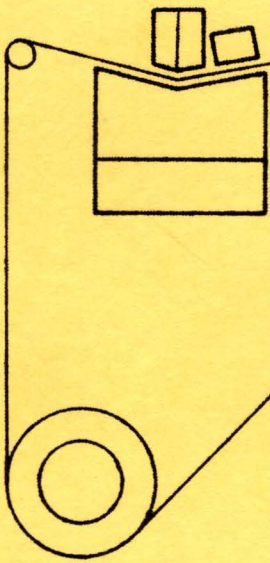
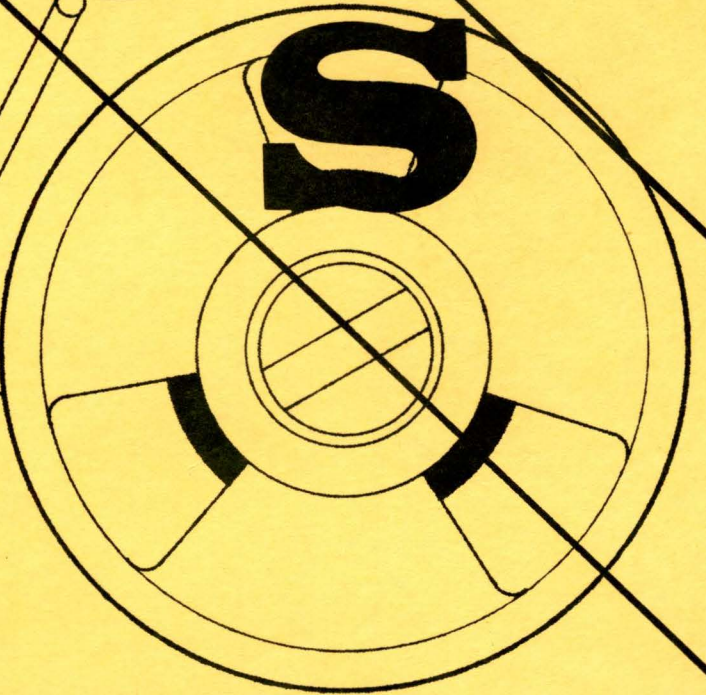
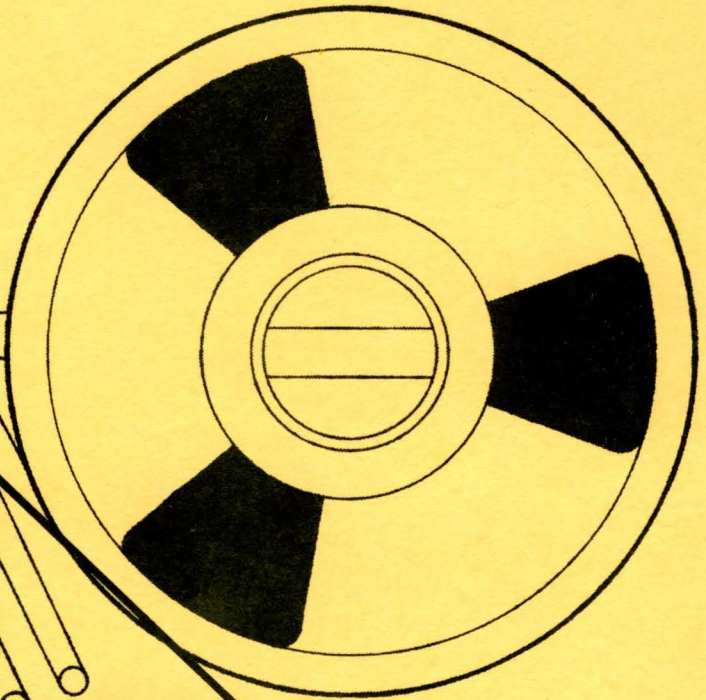


T

I

O

S



SYSTEM TEN BY SINGER

TIOS
(TAPE INPUT/OUTPUT SYSTEM)
REFERENCE MANUAL

TABLE OF CONTENTS

PREFACE

This reference manual explains how to use the TIOS (tape Input/Output System) program to transfer information to and from magnetic tape with great efficiency. The manual is intended for those writing programs in the Assembler I language. Readers should be experienced programmers familiar with the basic System Ten concepts discussed in these manuals:

- * System Ten Programmers Reference Manual
- * System Ten Assembler I Reference Manual.

TABLE OF CONTENTS

0-1	Preface
0-2	Table of Contents
Section 1	INTRODUCTION
1-1	Purpose of System
1-1	Purpose of Manual
1-1	Language and Memory Requirements
1-1	Input/Output Requirements
1-2	Auxiliary Software
1-2	Summary of Sections
Section 2	GENERAL DESCRIPTION
2-1	Features
2-2	Program Operation
Section 3	LABEL AND FILE STRUCTURE
3-1	File Arrangement
3-2	Label Content
3-3	Label Processing
3-3	End-Of-Volume Conventions
Section 4	USING TIOS REQUESTS
4-1	Pre-Processing Procedures
4-4	TIOS Requests and Parameters
4-11	TIOS Request Execution
4-13	Program Termination
4-14	Programming Checklist
4-15	Sample Program
Section 5	CAUSES AND CORRECTIONS OF ERRORS
5-1	OPEN
5-2	GET
5-2	PUT
5-3	CLOSE

TABLE OF CONTENTS

Section 6

| OPERATING INSTRUCTIONS

6-1

Equipment Configuration

6-1

Source Deck Preparation

6-2

Loading and Execution

| GLOSSARY

INTRODUCTION

PURPOSE OF SYSTEM

The TIOS (Tape Input/Output System) program helps the user direct magnetic tape input and output with great programming and storage economy. TIOS transfers data to and from seven or nine-track tape with automatic blocking and deblocking of data, use of up to four tape drives, dynamic assignment of tape I/O buffers, and creation and reporting of tape labels. TIOS reads and records data in USASCII code, and processes files and labels in accordance with ANSI standards for tape file structure. This means that tapes created by TIOS can be processed by other systems that follow these conventions, and vice-versa.

PURPOSE OF MANUAL

This reference manual illustrates how to use the TIOS requests. The manual also discusses label and file structure, special features of the system, and causes and corrections of errors.

LANGUAGE AND MEMORY REQUIREMENTS

TIOS is written in Assembler I language. It is furnished to the customer as a source card package to be incorporated and assembled with his resident source deck. TIOS requires 2418 character locations in one partition for READ/WRITE operations, or 1463 character locations for READ operations only, or 2058 character locations for WRITE operations only. Additional storage is required for buffers and work areas within the user's program. No TIOS operation requires common memory.

INPUT/OUTPUT REQUIREMENTS

Minimum Configuration

- * Singer Model 20 Processor
- * Friden Model 30 Card Reader
- * Friden Model 45 Magnetic Tape Drive
- * Friden Model 50 Line Printer

INTRODUCTION

Optional Equipment

- * Friden Model 60 Paper Tape Reader
- * Friden Model 70 Workstation or
Friden Model 7102 Communication Terminal
- * Additional Friden Model 45 Magnetic Tape Drives

AUXILIARY SOFTWARE

- * System Ten Multipartition Loader or
System Ten Absolute Loader

SUMMARY OF SECTIONS

Section 1 INTRODUCTION

This section describes the purpose of the system, language used in writing the program, core memory requirements, and minimum and optional I/O equipment configurations.

Section 2 GENERAL DESCRIPTION

The four program functions (OPEN, GET, PUT, and CLOSE) and other features of the program are explained in this section. The source deck structure and the TIOS data flow are illustrated. The three distinct TIOS configurations (READ/WRITE, READ ONLY, and WRITE ONLY) are discussed.

Section 3 LABEL AND FILE STRUCTURE

The content and arrangement of the tape labels processed by TIOS are discussed in this section. The structure of files on tape is also described.

Section 4 USING TIOS REQUESTS

The writing and execution of the TIOS requests (OPEN, GET, PUT, and CLOSE) and examples of their use are discussed in section 4. The TIOS buffer parameters, the creation of the file header labels, and error code checking are also explained.

Section 5 CAUSES AND CORRECTIONS OF ERRORS

This section denotes various tape I/O errors, their causes and corrections.

INTRODUCTION

Section 6 OPERATING INSTRUCTIONS

This section describes the System Ten equipment required by TIOS, preparation and assembly of source decks, and loading and execution of the resulting object decks.

FEATURES

TIOS provides a comprehensive, efficient method of recording data with the Friden Model 45 Magnetic Tape Drive. TIOS offers the user:

*Four program functions:

- OPEN, to access a tape drive and open a file.
- GET, to read data from a tape file.
- PUT, to write data onto a tape file.
- CLOSE, to close a tape file and terminate tape input/output processing.

*Use of four tape drives.

*Intercompatibility of all files produced by TIOS among different System Ten applications.

*Creation and reporting of tape file header and trailer labels in accordance with ANSI standards for general labeling compatibility throughout the data processing industry.

*Automatic end-of-volume processing conventions.

*Automatic blocking and deblocking of variable length records.

*Dynamic assignment of tape input/output buffers.

*Choice of on-line or off-line (local) rewinding.

*Error recovery conventions.

- If a parity error is encountered during writing, the write operation is re-attempted three times.
- If a parity error is encountered during reading, re-reading is attempted ten times.
- If a hardware fault error is encountered during processing of a TIOS command, processing may be continued when the error is corrected and the request is resubmitted.
- Normal/Error condition codes are continually available to the user's program.

GENERAL DESCRIPTION

To ensure proper tape processing, the user should carefully follow the guidelines for interfacing with TIOS described in this manual. To ensure file integrity and protection, he must properly schedule and control the File Access Channel (FAC) protect character that allows or prohibits input from or output to magnetic tape.

Use of alternating tape drives during processing of a file is not supported by TIOS. This must be accomplished through user-defined routines.

PROGRAM OPERATION

The TIOS program consists of four modules (the OPEN, CLOSE, GET, and PUT routines) that can be combined and used in three distinct configurations:

*READ/WRITE Configuration requires complete TIOS package (all four modules).

*READ ONLY Configuration requires OPEN, GET, and CLOSE.

*WRITE ONLY Configuration requires OPEN, PUT, and CLOSE.

GENERAL DESCRIPTION

Prior to assembly, the TIOS package is simply inserted within the user's source deck. Generally, TIOS is placed after the last executable statement or constant, but before the END or EXEC card that terminates the complete source deck (figure 2-1). Immediately following each request for a TIOS function, a special branch-and-switch instruction should be included to transfer control to TIOS.

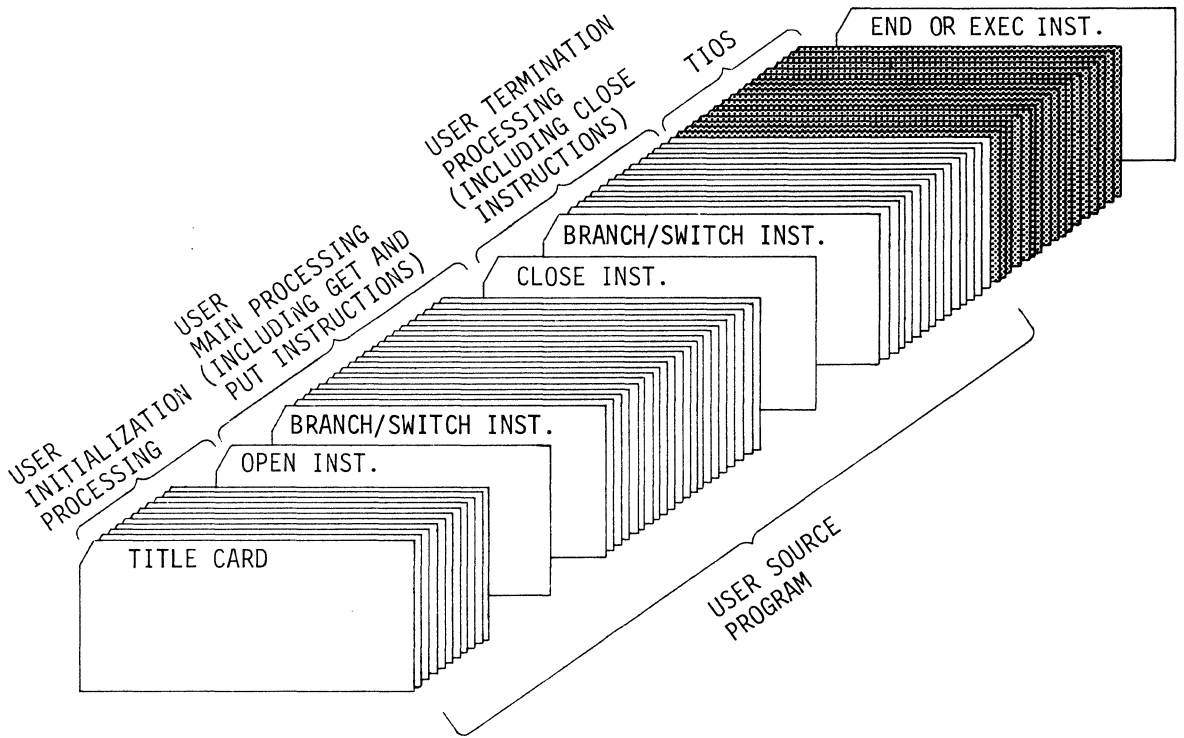


Figure 2-1 Sample Source Deck Structure

GENERAL DESCRIPTION

After the source deck is assembled and loaded into memory, each TIOS request and associated data is passed from a work area in the user's program to an input/output buffer within TIOS. Information is transmitted between the user's program and TIOS in the form of variable-length records. TIOS processes each request received, performing such functions as recording blocked data, retrieving and unblocking data, and communicating device status and error indications to the user's program. Data returned to the user's program is delivered from the TIOS buffer to the user's work area. When a TIOS request has completed processing, execution returns to the user's program. The TIOS data flow is depicted in figure 2-2.

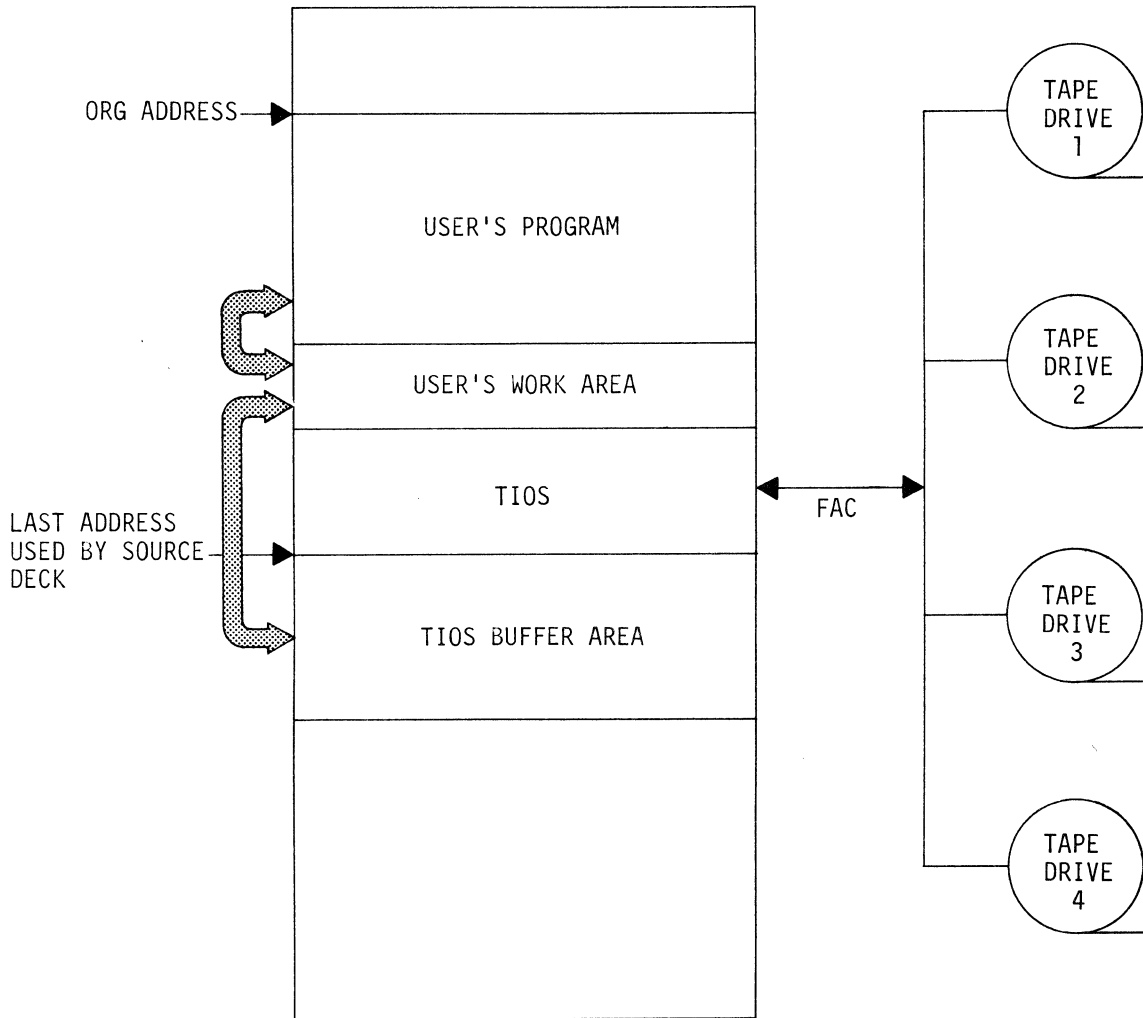


Figure 2-2 TIOS Data Flow

GENERAL DESCRIPTION

Status response codes from TIOS are delivered to the user in a special counter which he monitors through his program. When specific condition codes, such as that for an irrecoverable parity error or a hardware fault are encountered, appropriate user-defined routines can be called.

The TIOS routines provide the following functions:

*OPEN ensures that the tape drive requested in the OPEN request is on-line and is not in use, and that the tape is rewound to its beginning. For input tapes, OPEN reads and reports any file header label. For output tapes, OPEN writes the file header label.

*CLOSE, for an input tape, issues a rewind command to the tape drive referenced and releases the drive from reserved status to permit its use for other purposes. For an output tape, CLOSE writes a tape mark after the last record, then an End-of-File trailer label followed by two other tape marks, and then issues a request to the tape drive to rewind the tape. For both input or output tapes, the user can specify Local or On-Line rewinding.

*GET, when used initially, reads a block of logical records from tape into a buffer specified by the user, then supplies the first record in the buffer to the user's work area. Each subsequent GET request supplies a record to the work area until the buffer is exhausted. Then another block is read from the tape.

*PUT retrieves records from the user work area and places them into a user-specified buffer. When the buffer is full, PUT writes the records from the buffer to tape.

TIOS automatically creates and reports file header and trailer labels in accordance with the recommendations of the American National Standards Institute (ANSI), March 1969. The labeling procedures provided by TIOS are based on these conventions to support general labeling compatibility throughout the data processing industry.

Labeling provides a check against procedural and human errors in handling tape. When a tape is read, its header labels can be checked against label information supplied by the program using the tape. If the label on the tape conflicts with the label information in the program, processing can be halted to avoid use of the wrong tape.

These four types of standard labels are recognized and used throughout the industry:

Volume Header Label	Identified by the characters VOL1, and used to denote the beginning of a reel (volume) of tape. Usually supplied by computer vendor via a tape pre-processing program.
File Header Label	Identified by the characters HDR1, used to denote the beginning of a file on tape.
File Trailer Label	Identified by the characters EOF1, and used to denote the end of a file.
Volume Trailer Label	Identified by the characters EOV1, and used to denote the end of a volume (reel) when a file is continued on another reel.

FILE ARRANGEMENT

TIOS processes single-volume, single-file tapes. Tapes of other file configurations must be processed by user-defined routines. The interrelationship of labels and data on a single-volume, single-file tape is shown in figure 3-1. Header and trailer labels are denoted by their respective identifiers. Tape marks, which are required to separate label information from data, are shown by asterisks. The last label on a tape always terminates with a double tape mark.

LABEL CONTENT

Each label is read or written as an 80-character block. The content of the label is as follows:

Columns	Content
1-4	The identifier denoting the type of label: VOL1 (for volume header), HDR1 (for file header), EOF1 (for file trailer), and EOVL (for volume trailer label).
5-21	File identification code (included in HDR1 and EOF1 labels only). Maximum of 17 characters. Assigns a name to the file.
22-27	File serial number (included on HDR1 or EOF1 labels only). Maximum of six characters. Identifies the set of files to which this file belongs.
28-31	Volume sequence number (included in VOL1 and EOVL labels only).
32-35	File sequence number (included in HDR1 and EOF1 labels only). Identifies the sequence of this file within a volume set.
36-38	Reserved for future Friden use.
39-80	Non-reserved field.

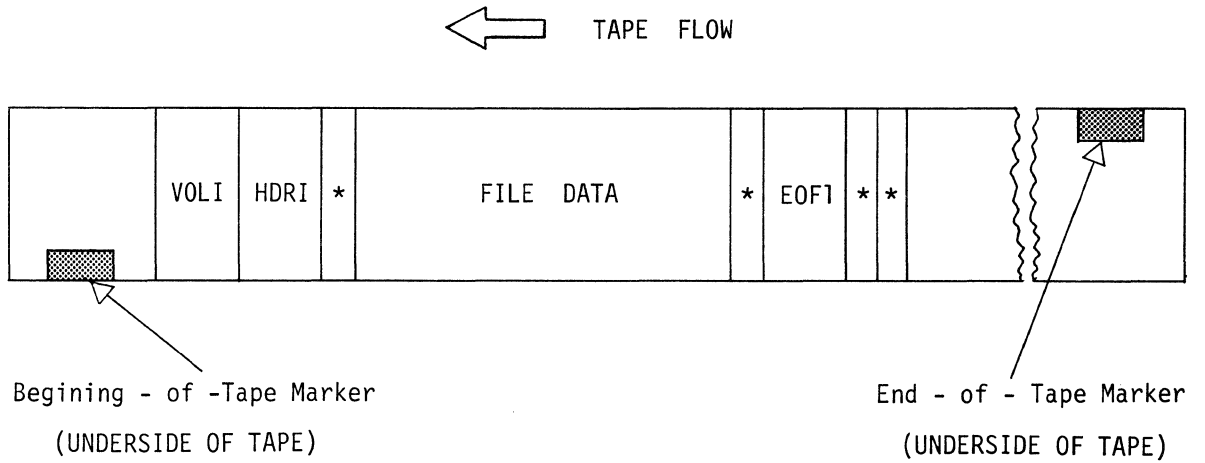


Figure 3-1 Single-Volume, Single-File Tape

LABEL PROCESSING

TIOS automatically handles file label creation and reporting for single-volume, single-file tapes. The labeling information in columns 1 through 4 is provided by TIOS, and the information in columns 5 through 80 is taken from input that the user supplies with the OPEN or CLOSE request (as described in section 4). All tape marks are written automatically by TIOS. When an OPEN request references an output tape, TIOS skips over the volume header label (usually written earlier by a vendor-supplied tape pre-processing program) and writes the file header (HDR1) label on the tape. When an OPEN request references an input tape, TIOS reports the contents of the file header label by moving this information into the user work area. The user can, if he desires, include within his program instructions to compare the tape label with the label information provided by the OPEN request, and terminate processing if they do not match.

When a CLOSE request references an output tape, a file trailer label (EOF1) is written, the tape is rewound, and the drive is released from reservation by the program. When a CLOSE instruction references an input tape, the tape is rewound without label checking.

When an input tape is read by a GET instruction, any EOF1 label is read as if it were user data; the label does not signal the end of information. Therefore, the user's program should contain instructions to continually check the first four characters of the work area for EOF1, and to branch to a CLOSE instruction when such characters are encountered.

END-OF-VOLUME CONVENTIONS

If a reflective end-of-tape marker is encountered during execution of the PUT request, the code 5 is placed in a condition code buffer to be monitored by the user's program. An EOVL label and accompanying tape marks are automatically written on the tape. The tape is then rewound. TIOS does not support the use of multiple (alternating) reels. Thus, if the user wants to continue a file on another reel, he must mount the new reel and supply instructions to space over the volume header label on this tape. He must next write a file header label followed by a tape mark. The user must then issue a special unconditional branch instruction, BC TR23WT(5), to continue processing this file.

The programmer generally employs TIOS in a simple three-phase process. First, he defines within his program certain preprocessing constants such as the limits of the TIOS buffer area, header label information, TIOS requests and parameters, and the size of the user work area. Then, within the body of his program, he includes instructions that move each TIOS request and its parameters into the user work area. Next, he issues a special branch instruction that transmits the request to TIOS for execution. Finally, he closes and rewinds all tape files referenced by TIOS, and terminates the program.

The steps involved in each phase are detailed in this section of the manual.

PRE-PROCESSING PROCEDURES

Setting Buffer Area Limits

Before he can issue TIOS requests within his program, the user must define an area within partition memory that can be used for TIOS buffers. As each tape file used by the program is opened, a buffer will be dynamically created within this area for input/output relating to the file. The size of each buffer is determined by a parameter furnished with the OPEN request. The precise location of the buffer is determined by TIOS. An example of buffer area use is shown in figure 4-1.

The low-address limit of the buffer area is determined by setting a five-digit constant labeled TR80DB. This address is typically the address in memory that follows the last address used by the source program (including TIOS). If this address has not been determined during a prior assembly, the user should offer his best estimate. The constant TR80DB is initially set to 00000, and should be reset by the user to avoid unspecified results. The user defines the value by a DM statement, and then moves this value into TR80DB with an MC statement. The following format can be used. (In this and other explanations of request formats, capital letters denote constants and small letters signify variables.)

```
labdm DM C'nnnnn'
      MC labdm,TR80DB
```

where labdm = label of DM statement

nnnnn = low-address limit of
buffer area

For example, to set a low-address limit of 6000, the user could write:

Label						Operation					Operand																		
1	2	3	4	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
L	A	B	U	F	F	D	M						C	'	0	6	0	0	0	'									
						M	C						L	A	B	U	F	F	,	T	R	8	0	D	B				

The high-address limit of the buffer area is defined by setting another five-digit constant, TR80LM. If this constant is not set, TIOS assumes a value of 10000. If, however, the user wants to enter a lesser value, he can do so with statements of this format:

```
labdm DM C'nnnn'
```

```
MC labdm,TR80LM
```

where labdm = label of DM statement

nnnnn = high-address limit of
buffer area

The value of TR80LM should not exceed the highest partition address available, and TR80DB should not exceed TR80LM. In either case, the error condition code 2 (buffer storage requested is not available) will be returned to the user when the first OPEN request is processed.

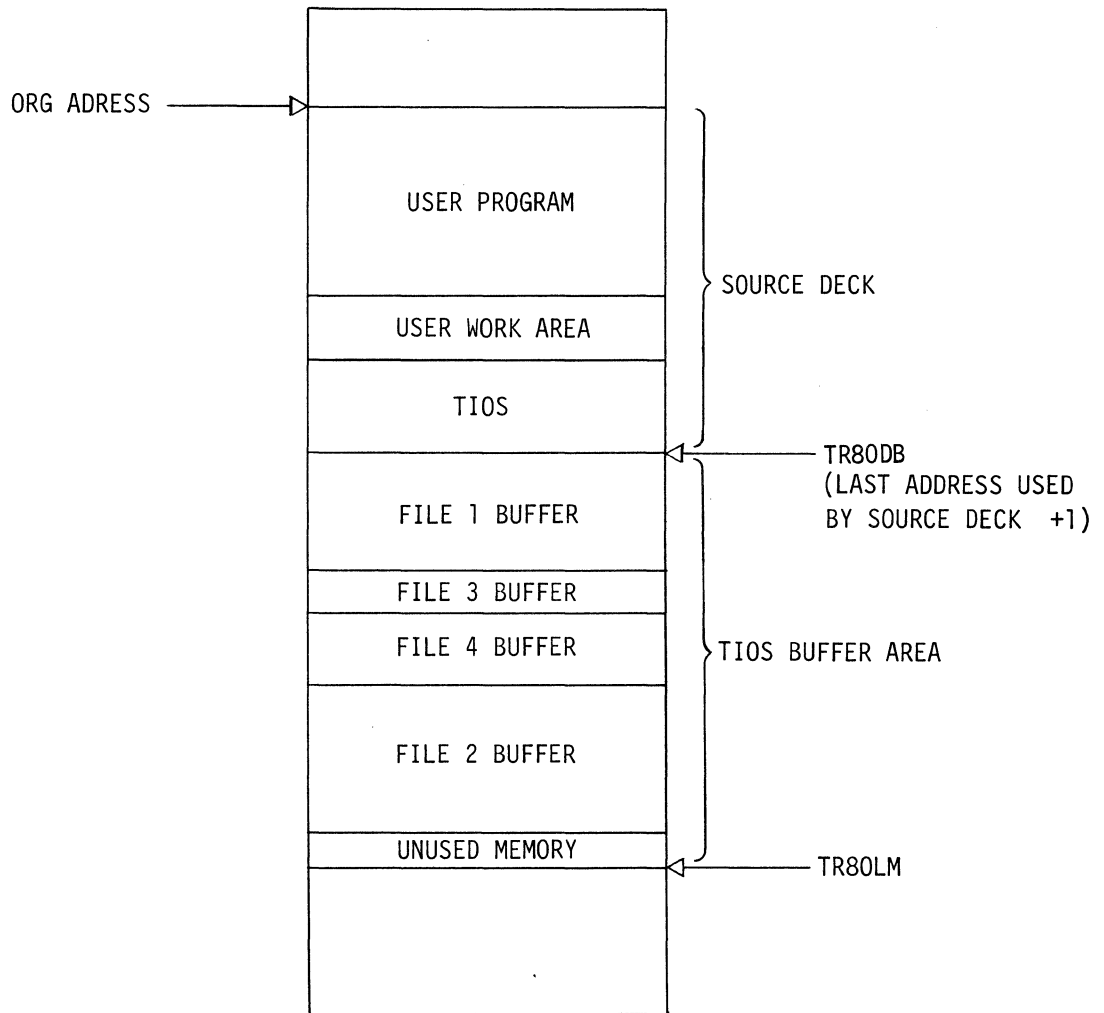


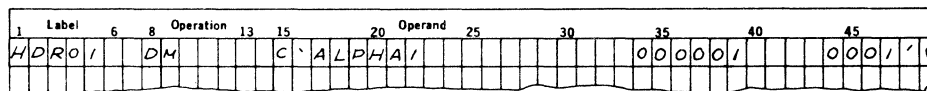
Figure 4-1 TIOS Buffer Area Use

Creating File Labels

Typically, volume header (VOL1) labels are previously written on magnetic tape through a tape pre-processing program supplied by the vendor. However, file header (HDR1) and trailer (EOF1) labels are created for output tapes at the user's option during processing with TIOS. Also, file header label-checking information can be optionally supplied for labeled input tapes. The user supplies file label information in DM statements. The labels can be written in any format, but that indicated in section 3 is recommended for standard labeling conventions. For the file header and trailer labels processed by TIOS, the format is:

<u>Columns</u>	<u>Information</u>
5-21	File identification code (file name)
22-27	File serial number
32-35	File sequence number

The user enters this information with the first character of the file name as the first character of the data field in the DM statement. As an example, to create a header label containing the file name ALPHA, the serial number 000001, and the sequence number 0001, the user could write:



When this information is referenced by a subsequent OPEN request, as shown later in this section, TIOS appends the file header identifier HDR1 into columns 1 through 4 of this field in memory. If the request is for an input tape, the label on the input tape is placed in the users work area for verification by the user. If the request is for an output tape, this information is automatically written as a label on that tape. When the information is referenced by a CLOSE request, TIOS enters the trailer identifier EOF1 in columns 1 through 4 in memory. If the request is for an output tape, the information is automatically written as a file trailer label on that tape. For input tapes, however, the EOF1 label is not transmitted to the user for verification.

TIOS REQUESTS AND PARAMETERS

Before a TIOS request can be executed, the user must specify the particular function (OPEN, GET, PUT, or CLOSE) and associated parameters within a DM statement. He later moves this information into the user work area (labeled TR80CC) and then issues a special branch and link instruction to execute the request.

OPEN Request

Before a tape file is referenced by a GET, PUT, or CLOSE request, it must be made available to the user by an OPEN request. The OPEN request, when executed, ensures that the tape drive specified is on-line but not in use, and that the tape is rewound to its beginning. If an output tape is specified, TIOS uses label information supplied by the user's program to write a file header label. If an input tape is specified, TIOS reads and reports any file header label by placing the label's contents in the user's work area.

Although an OPEN request must be issued for each new file referenced within a program, additional OPEN requests are not required when such a file is continued on another reel of tape.

One significant parameter (buff) in the OPEN request is used to specify the size of the TIOS buffer through which data passes between the user work area and TIOS. The size specified is limited by the amount of memory available to TIOS. For example, if the TIOS buffer area defined by TR80DB and TR80LM contains 2000 characters of memory but two OPEN requests each specify buffers of 1500 characters, the second request is aborted and condition code 2, (insufficient buffer area) is transmitted to the user's program. (Other errors and their condition codes are described in section 5.)

The OPEN request also includes an optional password parameter. When a GET, PUT, or CLOSE request is issued for a particular file, the password included as a parameter with that request is checked against the password supplied with the OPEN request for that file. (When the user omits this parameter, TIOS furnishes four blanks for the password.) If the passwords match, the request is executed. If not, the request is ignored and condition code P is returned to the user in a special location (TR80CS).

If a program abnormally terminates following selection of a tape drive, and the error is caused by the tape, it is recommended that the program be reloaded and re-executed from the beginning. This is desirable because the OPEN pointers required to continue processing may have been destroyed.

The OPEN request is defined by a DM statement of the following format. When an optional parameter is omitted, the user must include blanks in place of the parameter.

```
labelx DM C'@OPTpsbuffpass'
```

where:

labelx = DM statement label. Required parameter.

@ = @ or blank. When user enters parameters from workstation or communication terminal, @ is service request.

t = 0 for output tape; I for input tape. Required parameter.

P = Primary tape drive number (1 through 4). Required parameter.

s = Secondary tape drive number. Optional parameter.

buff = Size of TIOS buffer. Required parameter.

pass = Password. Optional parameter. When omitted, blanks are furnished as the password by default.

As an example of an OPEN request, the following statement would request opening of an input tape file installed on tape drive 3, with information passed through a 400-character TIOS buffer. The password is PASS.

1	Label					6	8	Operation							13	15	Operand															25	30
O	P	E	N	F	L	D	M							C	'	@	0	P	I	3	3	0	4	0	0	P	A	S	S	'			

To execute this request, the user would move it into the user work area and issue a special branch instruction (as noted later).

GET Request

When a GET request is moved into the user's work area and executed, one logical record is read from tape into the data field of the user's work area. Before a GET request for a file can be issued, an OPEN request for that file, specifying an input tape, must be issued. Otherwise, unspecified results will be obtained.

In the blok parameter in the GET request, the user specifies, for each request, the maximum size of the data block that resides on tape. Since blok specifies a maximum length rather than a fixed-length, this permits the user to mount tapes with different sized blocks without altering his program.

The GET request provides the user only one logical record each time it is issued. When the logical record is placed into the user's work area, TIOS enters the size of this record into the blok parameter in memory, overwriting the value entered by the user. Therefore, the user must re-enter the desired block-size value (blok) with each GET request. Because this value is returned with each record, the length of the record received will always be at least four characters less than the block size specified in the blok parameter.

The GET request is defined by a DM statement of the following format. When an optional parameter is omitted, the user must include blanks in place of the parameter.

```
labelx DM C'@GTdpassblok'
```

where:

- labelx = DM statement label.
- @ = @ or blank. When user enters parameters from workstation or communication terminal, @ is a service request.
- d = Tape drive number (1 through 4). Required parameter.
- pass = User-supplied password. Optional parameter.
- blok = Maximum size of data block on tape. (After GET is executed, complete size of logical record will be entered in this position, followed by the data.) Required parameter.

As an example, the following three GET requests (when moved into the user's work area and executed) will each read a record from tape drive 2. Maximum block size is 500 characters, and the password is WXYZ. After each request, the data returned to the user's work area is shown. The first record read is 136 characters long, the second is 100 characters, and the third is 240 characters.

REQUEST

1	Label					6	8	Operation					13	15	Operand					25	30							
G	E	T	L	B			D	M					C	'	@	G	T	2	W	X	Y	Z	0	5	0	0	'	

WORK AREA FOLLOWING

EXECUTION: Information Returned

@GT2WXYZ0136 (136 characters of data)

REQUEST:

1	Label					6	8	Operation					13	15	Operand					25	30							
G	E	T	L	B			D	M					C	'	@	G	T	2	W	X	Y	Z	0	5	0	0	'	

WORK AREA FOLLOWING

EXECUTION: Information Returned

@GT2WXYZ0100 (100 characters of data)

REQUEST:

1	Label					6	8	Operation					13	15	Operand					25	30							
G	E	T	L	B			D	M					C	'	@	G	T	2	W	X	Y	Z	0	5	0	0	'	

WORK AREA FOLLOWING

EXECUTION: Information Returned

@GT2WXYZ0240 (240 characters of data)

If a parity error is encountered during execution of a GET request, reading will be re-attempted ten times. If the attempt still does not succeed, reading will be abandoned and condition code 3 will be returned to the user.

If the password in GET does not match the password in the associated OPEN request, no information is read from the tape and condition code P is made available to the user.

PUT Request

When a PUT request is moved into the user's work area and executed, a logical record of specified length is transmitted to the TIOS buffer. When this buffer is full, its entire contents are automatically written on tape. A PUT request for a file will fail to execute unless an OPEN request for that file, specifying an output tape, was previously issued.

The PUT request is defined by a DM statement of this format. When an optional parameter is omitted, blanks must be included in its place.

```
labelx DM C'@PTdpassdata'
```

where:

- labelx = DM statement label
- @ = @ or blank. When user enters parameters from workstation or communications terminal, @ is a service request.
- d = Tape drive number (1 through 4). Required parameter.
- pass = User-supplied password. Optional parameter.
- data = Length of data to be written. Required parameter.

As an example, the following PUT statement, when executed, transfers a 160-character record to the TIOS buffer. When the buffer is full, its contents are written to tape drive 1. The password is PASS.

Label						Operation						Operand											
1	6	8	13	15	20	25	30	1	6	8	13	15	20	25	30								
L	B	D	A	T	A	D	M	C	'	@	P	T	/	P	A	S	S	O	/	1	6	0	'

If the PUT password does not match the password in the associated OPEN parameter list, no information is written and the condition code P is returned to the user.

If a block of data is being written (a PUT request is being executed) when a fault status error (condition code 4) occurs, the tape may run away (wind forward indefinitely without receiving data). There is no recovery from this condition, and the program must be reloaded. Generally, however, such an error is caused by factors external to the software.

If a normal fault error, such as a tape drive switching to Off-Line status, occurs, the program may be continued after the error is corrected by branching to a special entry point in TIOS (TR22FX). This procedure is described later in this section.

CLOSE Request

When a CLOSE request is moved into the user work area and executed, and the tape specified is an input tape, the tape is rewound and the drive is released from reserved status. When a CLOSE request is executed for an output tape, a tape mark is written after the last record, followed by an end-of-file trailer label and two terminating tape marks. Then the tape is rewound and the drive is released.

The CLOSE request is defined in a DM statement of the following format. When optional parameters are omitted, blanks must be used in their places.

```
labelx DM C'@CLrps pass'
```

where:

labelx = DM statement label.

@ = @ or blank. When user enters parameters from workstation or communications terminal, @ is a service request.

r = N for normal On-Line rewinding. L for transfer to Off-Line and Local rewinding. Required parameter.

p = Primary tape drive number (1 through 4). Required parameter.

s = Secondary tape drive number. Optional parameter.

pass = User-supplied password. Optional parameter. Must be separated from s parameter by four blanks.

In the following example, the tape on tape drive 2 is closed and rewound On-Line. The password MASH is used.

1	Label					6	Operation							13	15	Operand										25	30					
L	A	B	E	L			D	M					C	'	@	C	L	N	2	2								M	A	S	H	'

If the password does not match the password given in the associated OPEN request, the condition code P is returned to the user.

When a CLOSE request is issued, the dynamic buffer assigned to the file referenced is released, but the buffer limit indicator in TIOS is not reset. Therefore, this particular buffer cannot be reassigned through another OPEN request. However, space not previously used within the TIOS buffer area can be requested for use as a buffer. Thus, when the user calculates the total space to be included in the TIOS buffer area, he must consider that each file used will require an individual buffer of a specified size. The TIOS buffer area must be large enough to accommodate all such buffers even when they are no longer in use.

Defining The User Work Area

The user transmits TIOS requests and data to be written through the user's work area; he receives data read through this same area. An essential prerequisite for using TIOS is the definition of this area, which must be labeled TR80CC by a DM statement. The work area must be large enough to include the TIOS requests plus any data input or output by that request. (OPEN and CLOSE requests require 18 characters, and GET and PUT requests require 12 characters exclusive of data). As an example, where card-image records are to be read by a GET request, or written by a PUT request, the following work area could be defined.

1	Label					6	Operation							13	15	Operand										25	30					
T	R	8	0	C	C		D	M					9	2	C	O																
							D	M					1	2	C		'															
							D	M					8	0	C		'															

The TIOS requests would be read into the first twelve-character field in this 92-character work area, and the associated data would be input or output through the last 80-character field, labeled DATA.

TIOS REQUEST EXECUTION

Transmitting Requests and Labels to TIOS

The user must transmit each TIOS request and associated parameters to TIOS with an instruction that moves this data to the field labeled TR80CC. For example, if the OPEN request is defined by a DM instruction labeled OP4, the user could write the following:

1	Label	6	8	Operation	13	15	20	Operand	25	30
OP4		DM					C '@ P I 2 2 0 0 8 0 P A S S '			
		MC					OP4 , TR80CC			

For an OPEN or CLOSE request, the user must also transmit to TIOS any label information required. Typically, this information follows the TIOS request in the work area. Since an OPEN or CLOSE request will occupy 18 character locations in memory, the user could enter this information at TR80CC+18. As an example, if header label information is written in a DM statement labeled HEAD, the user can supply this information to TIOS with the statement:

1	Label	6	8	Operation	13	15	20	Operand	25	30	35	40	45
HEAD		DM					C ' F I L E A				0 0 0 0 0 1		0 0 0 1
		MC					HEAD , TR80CC + 1 8						

In a similar manner, for a PUT request, the user should move the data to be written into the work area following the PUT request. In this case, positions 9 through 12 of the work area must specify the length of the record to be transmitted.

Executing TIOS Requests

After a TIOS request is moved into the user's work area, the request is executed by the following branch-and-switch instruction:

1	Label	6	8	Operation	13	15	20	Operand	25	30	35
				BC				T R O I G 0 + 1 (6) , T R O O A A (5)			

As an example, consider the following statements:

1	Label	6	8	Operation	13	15	20	Operand	25	30	35
			MC				DATA	, TRBOCC	+12		
			MC				XPUT	, TRBOCC			
			BC				TROIGØ+1(6)	, TROOAA(5)			

The first statement passes data to the data field of the user work area. The second statement transmits a PUT request to TIOS from a DM statement labeled XPUT. The third statement executes the PUT request.

Error-Code Checking

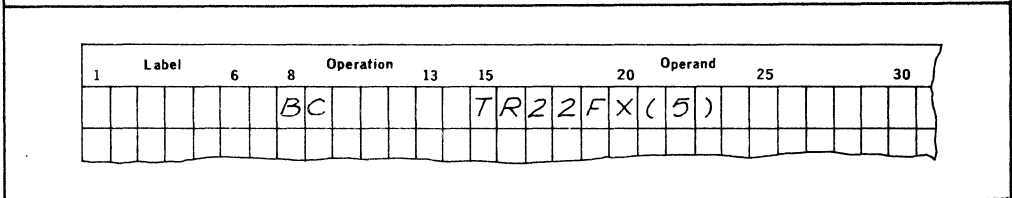
All condition codes established by the execution of TIOS requests are made available to the user through a location within TIOS, labeled TR80CS. The user should monitor this location with a Compare instruction following the Branch instruction that executes the TIOS request. He may then call user-defined transfer or exit routines when certain condition codes, such as a fault or parity-error code, are encountered. The available condition codes vary for each type of TIOS request. A complete list is shown in section 5.

An example of condition-code checking follows:

1	Label	6	8	Operation	13	15	20	Operand	25	30	35
			MC				ØPENI	, TRBOCC			
			BC				TROIGØ+1(6)	, TROOAA(5)			
			C				TRBOCS	, ØKAY			
			BC				*+10(2)	, TILT(5)			

USING TIOS REQUESTS

The first statement passes parameter information for an OPEN request to TIOS, and the second statement executes the OPEN request. The third statement tests to determine that the condition code returned in TR80CS matches the contents of the data area OKAY (defined previously in a DM statement to be zero). (A condition code of zero following an OPEN request signifies that the request was successfully executed. Any other condition code signifies an unusual event.) The fourth statement passes control to the next instruction if the condition code is zero. However, if it is not zero, execution is transferred to the instruction labeled TILT, which may be the entry point of an exit routine or any other appropriate set of instructions. If a hardware fault has occurred following execution of a PUT request, the user should include an option to branch to TR22FX (a special entry point in TIOS), after recovery from the error. This will ensure that, if the fault is corrected, the correct PUT request will be re-entered. If this branch is not taken, the data to be written may be lost. Following correction of the fault, the full instruction would be:



For any other errors, this Branch instruction should not be used. Instead, branching should transfer to normal user-defined routines.

Following each GET request, the user should monitor the first four characters of his work area for the characters EOF1. These characters signify that the end-of-file indicator has been reached and that an appropriate program branch must be taken.

PROGRAM TERMINATION

Prior to the end of a program, each tape file opened should be closed with the CLOSE request. This will issue proper end-of-file information on output tapes. It will also rewind both input and output tapes, and release the tape drives so that they can be used for other purposes. The CLOSE request does not release the buffer used by the referenced file, since the dynamic buffer values are not reset.

PROGRAMMING CHECKLIST

The checklist in figure 4-2 can be used as a guideline for writing TIOS requests within a program. All elements in the checklist should be taken into consideration before submitting the program for execution.

<p><u>FOR OVERALL PROGRAM</u></p> <ul style="list-style-type: none">*Have TIOS buffer area limits (TR80DB and TR80LM) been set?*Has user work area been defined? <p><u>FOR EACH TIOS REQUEST</u></p> <ul style="list-style-type: none">*If this is a GET, PUT, or CLOSE request, has the file referenced been opened previously with an OPEN request?*Has any label to be supplied (with OPEN and CLOSE) been defined in a DM statement?
--

Figure 4-2 TIOS Program Checklist

*Has TIOS request been defined in a DM statement?

*Has TIOS request been moved into user work area?

*Has label or data to be written with TIOS request been placed in the user work area?

*Has TIOS request been followed by this Branch instruction.

BC TR01G0+1(6),TP00AA(5)

*Has this Branch instruction been followed by proper error checking instructions?

FOR PROGRAM TERMINATION

*Has each tape file opened been closed with a CLOSE request prior to the end of the program?

Figure 4-2 TIOS Program Checklist (continued)

SAMPLE PROGRAM

Further examples of TIOS requests are presented in the following sample program. To make this illustration both simple and complete, an elementary program is used. The program's purpose is to read a logical record (an eighty-character card image) from tape, and to copy this card image to a second tape if an X appears in both columns 1 and 2. If a Y appears in column 1, the program is to terminate. The overall program flow is illustrated in figure 4-3. The program itself is presented in figure 4-4. The purposes of the statements are described in the comments field.

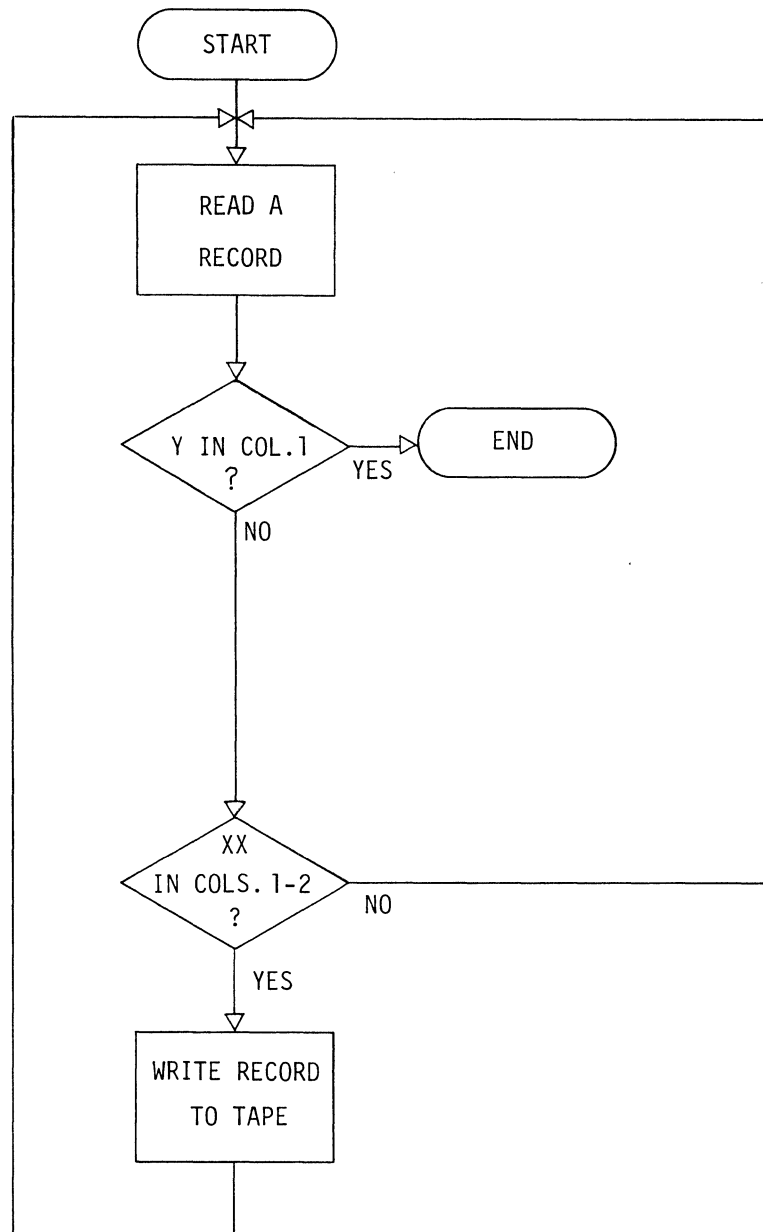


FIGURE 4-3 SAMPLE PROGRAM FLOW CHART

ASSEMBLER CODING FORM

PROGRAM CØPYTAPE		EXTENSION 2475		DATE JAN 20, 1971		PUNCHING INSTRUCTIONS		GRAPHIC		PAGE 1 OF 5	
PROGRAMMER R. A. JONES											
Label	Operation	Operand	Comments	Continuation	Program Identification	Line Number					
	ØRG	0300									
	TITLE	'CØPYTAPE'									
BUFFST	DM	C'06000'			DEF. BUFFER LOW-AD-						
					DRESS LIMIT TO 6K.						
BUFFED	DM	C'09000'			DEF. BUFFER HI-AD-						
					DRESS LIMIT TO 9K.						
HDRLAB	DM	C'CØPYTAPE	000001	0001'	DEFINE HDR LABEL.						
ØPENI	DM	C'@ØPI110400XCAT'			DEFINE ØPEN RE-						
					QUEST FOR INPUT						
					TAPE.						
ØPENØ	DM	C'@ØPØ220400XDØG'			DEFINE ØPEN RE-						
					QUEST FOR ØUTPUT						
					TAPE.						
GETTER	DM	C'@GT1XCAT0080'			DEFINE GET REQUEST.						
PUTTER	DM	C'@PT2XDØG0080'			DEFINE PUT REQUEST.						
CLØSEI	DM	C'@CLN11 XCAT'			DEFINE CLØSE RE-						
					QUEST FOR INPUT						
					TAPE.						

RESTRICTED

524-70124-M48

ASSEMBLER CODING FORM

PROGRAM		COPYTAPE		EXTENSION		2475		DATE		JAN. 20, 1971		PUNCHING INSTRUCTIONS		GRAPHIC		PAGE		2		OF		5					
PROGRAMMER		R. A. JONES		STATEMENT																				Program Identification		Line Number	
1	Label	6	Operation	13	15	20	Operand	25	30	35	40	45	50	55	60	65	69	71	76	77	80						
	CLØSEØ		DM			C '@CLN22			XDØG'						DEFINE CLØSE RE-												
	WHY		DM			C 'y'									QUEST FØR ØUTPUT												
	XX		DM			C 'xx'																					
	FØUR		DM			C '4'																					
	MSG1		DM			C 'TIØS REQUEST NOT COMPLETED'									DEFINE ERRØR MES-												
	MSG2		DM			C 'EØFI ENCØUNTERED'									SAGE.												
	MSG3		DM			C 'WRØNG TAPE'									DEFINE EØF MESSAGE.												
															DEFINE WRØNG TAPE												
															MESSAGE.												

RESTRICTED

524-701124-M48

ASSEMBLER CODING FORM

PROGRAM		EXTENSION		DATE		PUNCHING INSTRUCTIONS		GRAPHIC		PAGE		OF														
CØPYTAPE		2475		JAN 20, 1971						3		5														
PROGRAMMER		STATEMENT												Program Identification		Line Number										
R. A. JØNES																										
1	6	8	Operation	13	15	20	Operand	25	30	35	40	45	Comments	50	55	60	65	69	71	76	77	80				
INTPL		DM					C 'XX TAPE'																			
ZERØ		DM					C 'O'																			
EØF		DM					C 'EØF1'																			
TRBOCC		DM					92CØ																			
		DM					12C' '																			
DATA		DM					80C' '																			
GØ		MC					BUFFST, TR80DB																			
		MC					BUFFED, TR80LM																			
		MC					ØPENI, TR80CC																			
		BC					TROI GØ+1(6), TROOAA(5)																			
		C					TR80CS, ZERØ																			
		BC					*+10(2), ERRØR(5)																			
		C					INTPL(6), TR80CC+18																			
		BC					*+10(2), WRØNGT(5)																			
		MC					ØPENØ, TR80CC																			
		MC					HDRLAB, TR80CC+18																			
		BC					TROI GØ+1(6), TROOAA(5)																			
		C					TR80CS, ZERØ																			
		BC					*+10(2), ERRØR(5)																			

ASSEMBLER CODING FORM

PROGRAM		EXTENSION		DATE		PUNCHING INSTRUCTIONS		GRAPHIC		PAGE		OF							
COPY TAPE		2475		JAN 20, 1971		PUNCH				4		5							
STATEMENT													Continuation	Program Identification		Line Number			
1	6	8	13	15	20	25	30	35	40	45	50	55	60	65	69	71	76	77	80
READ		MC			GETTER, TR80CC								READ A RECORD.						
		BC			TR01GØ+1(6), TR00AA(5)														
		C			TR80CS, ZERØ								BRANCH TØ ERRØR						
		BC			*+10(2), ERRØR(5)								EXIT IF ERRØR						
		C			DATA(4), EØF								BRANCH TØ ENDFL						
		BC			ENDFL(2), *+10(5)								IF EØF! READ.						
		C			DATA(1), WHY								BRANCH TØ TERM IF Y						
		BC			TERM(2), *+10(5)								IS READ IN CØL. 1.						
		C			DATA(2), XX								RETURN TØ READ IF						
		BC			*+10(2), READ(5)								XX NØT IN CØLS. 1-2						
		MC			PUTTER, TR80CC								WRITE RECORD IN						
													WØRK AREA TØ TIØS						
													BUFFER.						
		BC			TR01GØ+1(6), TR00AA(5)														
		C			TR80CS, FØUR								RE-ENTER PUT REQUE-						
		BC			TR22FX(2), *+10(5)								ST IF FAULT ØCCURS						
													AND IS CØRRECTED.						
		C			TR80CS, ZERØ								BRANCH TØ ERRØR-						
		BC			*+10(2), ERRØR(5)								EXIT IF ERRØR.						
		BC			READ(5)								RETURN TØ READ.						

RESTRICTED

524-701124-M48

ASSEMBLER CODING FORM

PROGRAM		EXTENSION		DATE		PUNCHING INSTRUCTIONS		GRAPHIC		PAGE		OF							
COPYTAPE		2475		JAN 20, 1971						5		5							
PROGRAMMER		STATEMENT																	
R.A. JONES																			
1	6	8	13	15	20	25	30	35	40	45	50	55	60	65	69	71	76	77	80
WRONGT	W				MSG3(2),10(1)														
	BC				TERM(5)														
ERROR	W				MSG1(2),26(1)														
	BC				TERM(5)														
ERRORA	W				MSG1(2),26(1)														
					TERMA(5)														
ENDFL	W				MSG2(2),16(1)														
	BC				TERM(5)														
TERM	MC				CLØSEI,TRØCC														
	BC				TROIQØ+1(6),TROØAA(5)														
	C				TRØCCS,ZERØ														
	BC				*+10(2),ERRØR(5)														
TERMA	MC				CLØSEØ,TRØCC														
	BC				TROIQØ+1(6),TROØA(5)														
	C				TRØCCS,ZERØ														
	BC				*+10(2),ERRØRA(5)														
	EXEC				GØ														
	END																		

CAUSES AND CORRECTIONS OF ERRORS

Successful execution of a TIOS request (OPEN, GET, PUT, and CLOSE) is indicated by a zero (0) response code from the magnetic tape I/O controller. Response codes for tape I/O errors, and the causes and corrections of these errors, are shown in figure 5-1. Following each request for TIOS service, the user should include instructions that check the response code returned and execute a branch or other appropriate action if an error is indicated. An example of how this is done is shown in section 4.

OPEN

RESPONSE CODE	CAUSE	CORRECTION
1	Tape drive previously selected.	Change drive number requested and resubmit OPEN request.
2	Memory for buffer size requested is not available.	Reduce size of buffer by significant amount and resubmit OPEN request.
3	Uncorrectable READ error on input tape HDR1 label.	Check parity. Check tape for scratches or wrinkles. Dust and clean tape heads.
4	Hardware fault on tape drive.	Check for power ON. Check for drive ON-LINE. Check that correct number is assigned to drive. Resubmit OPEN request.

Figure 5-1 Errors and Corrections

CAUSES AND CORRECTIONS OF ERRORS

GET

RESPONSE CODE	CAUSE	CORRECTION
3	Uncorrectable READ error on the data block (10 tries failed).	Check parity. Check tape for scratches or wrinkles. Dust and clean tape heads.
4	Hardware fault on tape drive.	Check for power ON. Check for drive ON-LINE. Check that correct number is assigned to drive. Resubmit GET request.
P	Password error. (Unauthorized request for service detected.)	Ensure that an OPEN request preceded this request, and that password in GET parameters matches password in OPEN parameters.

PUT

RESPONSE CODE	CAUSE	CORRECTION
4	Hardware fault on tape drive.	Check for power ON. Check for drive ON-LINE. Check that correct number is assigned to drive. Resubmit PUT request and branch to TR22FX.
5	End Of Volume encountered. Tape mark, EOVL label, and two terminating tape marks are recorded. Tape is rewound and set to Local status.	Mount new reel. Space over volume-label. Write HDR1 label followed by tape mark. Resume processing by branching unconditionally to TR23WT.
P	Password error. (Unauthorized request for service detected.)	Ensure that an OPEN request preceded this request, and that password in PUT parameters matches password in OPEN parameters.

Figure 5-1 Errors and Corrections (continued)

CAUSES AND CORRECTIONS OF ERRORS

CLOSE

RESPONSE CODE	CAUSE	CORRECTION
4	Fault on tape drive.	Check for power ON. Check for drive ON-LINE. Check that correct number is assigned to drive. Resubmit CLOSE request.
P	Password error. (Unauthorized request for service detected.)	Ensure that an OPEN request preceded this request, and that password in CLOSE parameters matches password in OPEN parameters.

Figure 5-1 Errors and Corrections (continued)

OPERATING INSTRUCTIONS

EQUIPMENT CONFIGURATION

Standard Equipment

- *Singer Model 20 Processor
- *Friden Model 30 Card Reader
- *Friden Model 45 Magnetic Tape Drive
- *Friden Model 50 Line Printer

Optional Components

- *Friden Model 60 Paper Tape Reader
- *Friden Model 70 Workstation or
Friden Model 7102 Communications Terminal

SOURCE DECK PREPARATION

The TIOS program is issued to the user as one complete software package with READ/WRITE capability. It is recommended that the package be used in its entirety. However, in certain cases where conservation of memory is extremely important, the GET or PUT modules can be removed, creating packages with WRITE ONLY or READ ONLY capability respectively. This should be done with great caution. Once the user exercises this option, software support for the entire package is invalidated.

READ ONLY Modification

To modify TIOS for READ ONLY operation (no PUT module), remove the following cards:

```
TRXPUT C   TR80CC+1(1),TR90PT   IS IT A PUT REQ
          BC TR20(2),TR01GO(5)   LAST CHECK BEFORE
                                   EXIT
```

Next, remove the group of cards beginning with:

```
TRAPUT ORG *                       BEGIN PUT SUB-MODULE
```

and concluding with:

```
TRZPUT ORG *                       END PUT SUB-MODULE
```

OPERATING INSTRUCTIONS

WRITE ONLY Modification

To modify TIOS for WRITE ONLY capability (no GET module), remove the following cards:

```
TRXGET C TR80CC+1(1),TR90GT IS IT A READ REQUEST
BC TR10(2),*+10(0)
```

Then, remove the group of cards beginning with:

```
TRAGET ORG * BEGIN GET SUB-MODULE
```

and concluding with:

```
TRZGET ORG * END GET SUB-MODULE
TRIOPF BC TROORY+1(6),TROORE(5) USE REREAD COUNT OF
OPEN
BC TR10RR(5) TRY AGAIN
```

Should the user fail to follow these instructions precisely, errors may occur during assembly.

The TIOS package (modified or unmodified) can be inserted anywhere in the source program immediately before any END or EXEC card. It is recommended that this package be placed before the last END or EXEC card, as shown in figure 6-1, to minimize the possibility of programming errors.

LOADING AND EXECUTION

After preparing his source deck (which now includes the selected TIOS package), the user loads and assembles his program using any standard System Ten loader and Assembler I.

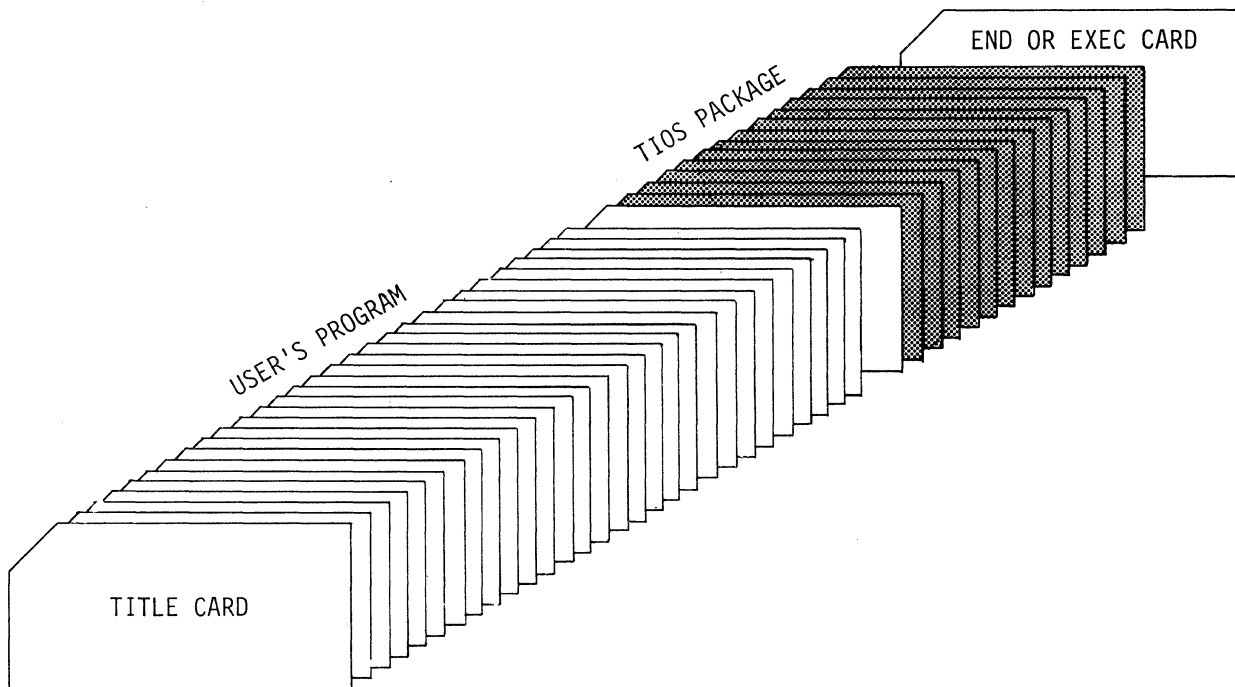


Figure 6-1 Sample Source Deck Structure

INTRODUCTORY NOTE

The purpose of the Glossary is to define all new terms introduced in the text and to define any special use made of standard terms. Standard terms which are used in a standard sense are not included. For elucidation on these, the reader is referred to Computer Dictionary and Handbook by Charles J. Sippl (Howard W. Sams & Co., Inc., Indianapolis, 1966).

Following is a list of the items defined in the Glossary:

Alphabetic Field	Host Partition
Arithmetic and Control Unit (ACU)	Implied Parameter
Auxiliary Storage	Index Register
Block	IOC
Blocking	Label
Bootstrapping	Label Table
Branch	Link
Buffer	Literal
Burst Mode Transmission	Local Mode
Byte	Logical Record
Byte Mode Transmission	Main Memory
Central Processing Unit (CPU)	Memory Module
Channel	Mixed Field
Characters	Multiprogramming
Common Area of Memory	Numeric Field
Control Character	Object Program
Control Field	On-Line Mode
Cycle-Stealing	Operation Code
Deblocking	Overdraft
Disc, Bound	Overflow
Disc, Free	Parity Bit
Double Frame	Partition
Effective Address	Partition Switching
File	Physical Record
Filler Characters	Privileged Area of Memory
Flagged Constant	Protected Area of Memory
Flowcharting Symbols	Return Address
Hexadecimal Number System	Sector
	Source Program
	Tape Mark
	USASCII
	Volume

GLOSSARY

A

Alphabetic Field

A field consisting strictly of the alphabetic characters A thru Z.

Arithmetic and Control Unit (ACU)

In System Ten, that part of the Central Processing Unit (CPU) which controls and performs the execution of machine instructions.

Auxiliary Storage

Storage in addition to the main storage of a computer. Auxiliary storage usually holds much more information than the main storage, and the information is accessible less rapidly. In System Ten, the disc is considered to be auxiliary storage.

B

Block

When used in connection with the Friden Model 45 Magnetic Tape Drive, block means a series of consecutive tape characters. The end of one block and the beginning of the next is signalled by a segment of blank tape called an inter-block gap. By this definition, block is synonymous with the phrase "physical record".

Blocking

The grouping of two or more logical records into one block of data.

Bootstrapping

A technique for loading the first few instructions of a program into storage, then using these instructions to bring the rest of the program into storage. This sometimes involves either the manual entering of a few instructions or the use of a key on a console. In System Ten, the bootstrap sequence is initiated by the depression of a LOAD button on an input device or by the occurrence of certain errors during program execution.

Branch

In System Ten, a departure from the normal sequential processing of instructions as caused by the execution of the Branch instruction. Another type of departure from sequential processing is the switch which passes control to a neighboring partition.

Buffer

Temporary storage used to compensate for the difference in operating speeds of input/output devices and the Central Processing Unit (CPU). In System Ten, the card reader has two card buffers, the card punch has three card buffers, the line printer has two line buffers, etc. Each IOC has one character buffer.

Burst Mode Transmission

A mode of communication between the Central Processing Unit (CPU) and external input/output devices. The information is transmitted without interruption as a solid procession of binary bits. In System Ten, the burst mode is employed in transmitting between Main Memory and the Friden Model 40 Disc Drive.

Byte

In System Ten, a group of 6 adjacent binary bits. The bits are referred to as bit-7, bit-5, bit-4, bit-3, bit-2, and bit-1. Bit-6 of the USASCII Standard Code is not used.

Byte Mode Transmission

A mode of communication between the Central Processing Unit (CPU) and external input/output devices. Transmission proceeds one character at a time on a cycle-stealing basis. In System Ten, all transmission through the Input/Output Channel (IOC) is accomplished in this mode, as is also transmission through the File Access Channel (FAC) when the Friden Model 45 Magnetic Tape Drive is the input/output device.

C

Central Processing Unit (CPU)

In System Ten, the Central Processing Unit (CPU) comprises the Arithmetic and Control Unit (ACU), the File Access Channel (FAC), one to twenty Input/Output Channels (IOC), and Main Memory.

Channel

A path along which information, particularly a series of bits or characters, may flow. In System Ten, each partition has a private Input/Output Channel (IOC) which transmits in the byte mode. Common to all partitions is a single File Access Channel (FAC) which transmits in the burst mode when the disc is used.

Characters

A set of coded symbols that includes the decimal digits 0 thru 9, letters A thru Z, punctuation marks, operation symbols, and other symbols. In System Ten, each character is represented by 6 binary bits.

Common Area of Memory

In System Ten, that portion of memory which is not partitioned. The Common Area comprises Protected Storage (locations 0-299), a Non-Privileged area, and an optional Privileged area accessible only to privileged partitions.

Control Character

A character whose occurrence in a particular context initiates, modifies, or stops a control operation.

Control Field

In System Ten, the second operand used by the Edit instruction. The control field governs the format of the edited result.

Cycle-Stealing

Data channels give the Arithmetic Control Unit (ACU) the ability to delay the execution of a program for communication of an input/output device with memory. If an input unit requires a memory cycle to store data that it has collected, the data channel makes it possible to delay the program prior to the execution of an instruction, and to store the data without changing the logical condition of the ACU. After the data is stored, the program continues as though nothing has occurred. In System Ten, cycle-stealing occurs between instructions, and during the transmission of data between magnetic tape and memory.

GLOSSARY

D

Deblocking

The reduction of one block of data into the individual logical records that comprise it.

Disc, Bound

In System Ten, a disc is bound to a given partition as soon as the partition institutes a seek upon it. The disc remains bound until data transmission is complete. While a disc is bound to a given partition, it cannot be accessed by another partition.

Disc, Free

In System Ten, a disc is free when it is not bound to another partition.

Double Frame

In System Ten, a special method of reading or writing 9-track magnetic tape in which the eight bits of data in each tape row are constructed from (or read into) the numeric portions of two consecutive locations in main memory. The same method can also be used on eight-channel paper tape readers and paper tape punches.

E

Effective Address

The address that is actually used in a particular execution of an instruction. In System Ten the effective address for a given instruction address field is the sum of the address field added to the contents of whatever index register is associated with that field.

F

File

When used in connection with the Friden Model 45 Magnetic Tape Drive, file means a series of consecutive blocks (or "physical records"). The end of one file and the beginning of the next is signalled by a special two character block called a Tape Mark.

GLOSSARY

Filler Characters

All characters used in the edit machine instruction EXCEPT the 'at' sign (@), slash (/), dash (-), comma (,), and decimal point (.).

Flagged Constant

In System Ten Assembler-1 Language, a constant with a C or P appended to indicate a common or partition address. E.G., 495C, 4567P.

Flowcharting Symbols

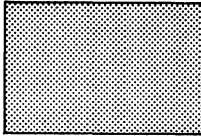
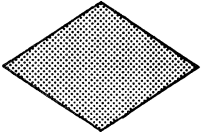
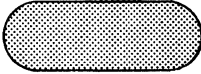

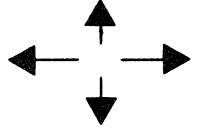
SYMBOL	REPRESENTS
	PROCESSING A GROUP OF PROGRAM INSTRUCTIONS WHICH PERFORM A PROCESSING FUNCTION OF THE PROGRAM.
	DECISION THE DECISION USED TO DOCUMENT POINTS IN THE PROGRAM WHERE A BRANCH TO ALTERNATE PATHS IS POSSIBLE BASED UPON VARIABLE CONDITIONS.
	TERMINAL THE BEGINNING, END, OR POINT OF INTERRUPTION IN A PROGRAM.
	CONNECTOR AN ENTRY FROM, OR AN EXIT TO, ANOTHER PART OF THE PROGRAM FLOWCHART.
	FLOW DIRECTION THE DIRECTION OF PROCESSING OR DATA FLOW.

Figure 6-1

H

Hexadecimal Number System

A number system using the equivalent of the decimal number sixteen as a base. In System Ten, the digits greater than 9 are written as 10, 11, 12, 13, 14, 15.

Host Partition

| The partition in control when the given instruction is executed.

I

Implied Parameters

| The length and index register parameters associated with a label in the label table. Whenever a label appears in the operand field of a statement using length and/or index parameters, the length and index register parameters may be omitted and they will be supplied automatically from the label table. Thus the idea that, by being omitted, the parameters were "implied".

Index Register

| A register whose primary purpose is to modify addresses in computer instructions. In System Ten there are three index registers in each partition. They are referred to by number and occupy partition storage locations 11-14 (index register 1), 21-24 (index register 2), and 31-34 (index register 3).

IOC

| An abbreviation for Input/Output Channel on the System Ten.

L

Label

| When applied to magnetic tape, an 80-character block at the beginning or end of a volume or file which identifies and delimits that volume or file.

Label Table

A table produced by the assembler which contains, for every label in the source program, the address assigned to the label by the assembler, an indication as to whether the address is in common or partition, an implied length associated with the label, and the implied index.

Link

In System Ten, a variant of the Branch instruction ordinarily used in calling subroutines. Link first establishes a return path from a subroutine and then passes control to the beginning of the subroutine.

Literal

An item of data with its value as stated. The assembler permits the use of a literal in the text field of the TITLE assembler command and in the data field of the DM (Define Memory) assembler command. Literals are not used in the machine instruction statements.

Local Mode

The mode in which data may NOT be transmitted between a device and an Input/Output Channel (IOC).

Logical Record

A record whose length is governed by the length of the information it contains. A subset of a physical record, which may or may not be as long as the physical record. A physical record may contain many logical records, each of which represents a logical set of information about a particular subject or item.

M

Main Memory

In System Ten, the entire core storage. This includes the storage occupied by all partitions as well as the entire common region which includes the protected area, the non-privileged area, and the privileged area.

Memory Module

In System Ten, the minimum unit of core storage. A Memory Module contains 10,000 character positions. In System Ten, core storage may contain 1-11 Memory Modules.

Mixed Field

A field which may include any combination of characters including, for example, alphabetic, numeric, and punctuation characters. In System Ten, a field prepared for printing by the Edit instruction is usually a mixed field.

Multiprogramming

A technique for handling numerous routines or programs simultaneously by overlapping or interleaving their execution. In System Ten, the programs being executed simultaneously each reside in a separate memory partition. Multiprogramming is achieved by passing control from one partition to another in round-robin sequence.

N

Numeric Field

A field containing numeric information and sign indication.

O

Object Program

The machine-language representation of a source program. If a source program contains no errors, detectable by the assembler, it is translated to machine-language; the object program is what results from this translation process.

On-Line Mode

The mode which permits the transmission of data between a device and an Input/Output Channel (IOC) or File Access Channel (FAC).

Operation Code

The part of a System Ten machine instruction which specifies the operation to be performed.

Overdraft

In System Ten, an intermediate condition which sometimes occurs in the subtraction process. Not to be confused with Overflow, which yields a wrong answer.

Overflow

In System Ten, an erroneous result caused by an attempt to develop an answer too large for the field assigned to it.

P

Parity Bit

A binary digit (i.e., either 0 or 1) appended to a string of bits to make the sum of all the bits which are ON either always odd or always even.

Partition

In System Ten, a portion of core storage. A system may contain 1-20 partitions. Each partition has 3 index registers and an Input/Output Channel (IOC). Partitions may communicate with each other only through common storage or devices on the File Access Channel (FAC).

Partition Switching

In System Ten, an automatic process by which control passes from one partition to its neighbor. Partition Switching consists essentially of saving status information necessary to resume the program which is relinquishing control, selecting the partition which is to gain control, restoring its Condition Code, and passing control to the appropriate instruction within it.

Physical Record

When used in connection with the Friden Model 45 Magnetic Tape Drive, synonymous with block.

Privileged Area of Memory

In System Ten, an optional hardware setting that reserves an upper portion of common storage for use by privileged partitions which are designated when the option is set.

Protected Area of Memory

In System Ten, locations 0-299 of the common storage area. Programs cannot store information in this area which is used by the ACU to keep information pertinent to partition switching and input/output operations. A program may examine information in the protected area even though it cannot (directly) alter it.

GLOSSARY

R

Return Address

In System Ten, the address of the instruction to which control returns after a particular execution of a subroutine.

S

Sector

A sector is one-fiftieth (1/50th) of a track on the Friden Model 40 Disc Drive. Each sector holds 100 characters (each consisting of 6 bits). Reading and writing on the disc is always done in groups of 100 characters.

Source Program

A program coded in other than machine-language, and which must therefore be translated into machine-language by an assembler or compiler before it can be executed. The assembler processes a source program coded in assembler language and translates it into an object deck.

T

Tape Mark

A special two-character block used as a delimiter on tape, but having no representation in core memory. A single tape mark is used to separate labels and data. A double tape mark signals the end of the information written on tape.

U

USASCII

A contraction for "United States of America Standard Code for Information Interchange". This standard defines the graphics and codes for a 128 character set. Commonly referred to as ASCII.

V

Volume

A set of files.