NASA CR-
144651

SYSTEMS FUNCTIONAL DESIGN SPECIFICATIONS
FOR THE
SHUTTLE PROGRAM INFORMATION MANAGEMENT SYSTEM
(SPIMS)

Job Order 88-019

Prepared By

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Under Contract NAS 9-12200

For

INSTITUTIONAL DATA SYSTEMS DIVISION

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

December 1975

(NASA-CR-144651) SYSTEMS FUNCTIONAL DESIGN SPECIFICATIONS FOR THE SHUTTLE PROGRAM INFORMATION MANAGEMENT SYSTEM (SPIMS) (Lockheed Electronics Co.) 119 p

N76-71510

Unclas
00/98 09227

JSC-10766

SYSTEMS FUNCTIONAL DESIGN SPECIFICATIONS FOR THE SHUTTLE PROGRAM INFORMATION MANAGEMENT SYSTEM

Job Order 88-019

PREPARED BY

N. C. Wolfe
Shuttle Information Management Systems Department

APPROVED BY

FD6
5531

LEC	NASA
W. B. Hopkins, Supervisor SPIMS, System Design and Integration Section	B. L. Brady, Project Manager SPIMS, Institutional Data Systems Division
R. L. Ready, Manager Shuttle Information Management Systems Department	R. R. Regelbrugge, Chief/FD6 Data Systems Development Branch
H. F. Thompson, Director Software Applications Branch	LaRue W. Burbank, Project Manager FD/SPIMS
	C. R. Huss, Chief Institutional Data Systems Division

Prepared By

Lockheed Electronics Company, Inc.

For

Institutional Data Systems Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

December 1975

LEC-6192

TECHNICAL REPORT INDEX/ABSTRACT
(See instructions on reverse side.)

1. TITLE AND SUBTITLE OF DOCUMENT	Systems Functional Design Specifications for the Shuttle Program Information Management System (SPIMS) Job Order 88-019
2. JSC NO.	JSC-10766
3. CONTRACTOR/ORGANIZATION NAME	LEC/ASD
4. CONTRACT OR GRANT NO.	NAS 9-12200
5. CONTRACTOR/ORIGINATOR DOCUMENT NO.	LEC-6192
6. PUBLICATION DATE (THIS ISSUE)	December 1975
7. SECURITY CLASSIFICATION	Unclassified
8. OPR (OFFICE OF PRIMARY RESPONSIBILITY)	
9. LIMITATIONS — GOVERNMENT HAS UNLIMITED RIGHTS	[X] YES [ ] NO
IF NO, STATE LIMITATIONS AND AUTHORITY	
10. AUTHOR(S)	N. C. Wolfe

11. DOCUMENT CONTRACT REFERENCES		12. HARDWARE CONFIGURATION	
WORK BREAKDOWN STRUCTURE NO.	N/A	SYSTEM	N/A
CONTRACT EXHIBIT NO.	N/A	SUBSYSTEM	N/A
DRL NO. AND REVISION	N/A	MAJOR EQUIPMENT GROUP	N/A
DRL LINE ITEM NO.	N/A		

13. ABSTRACT

The purpose of this document is to provide a functional design specification for each hardware/software element comprising the Shuttle Program Information Management System (SPIMS). These elements are broken down into the following categories: Terminal Communications Network, Terminal Control Subsystems, and the Data Base Management Subsystem.

14. SUBJECT TERMS

SPIMS	KRONOS	CYBER-74
UNIVAC-494	Terminal Control System	Terminal Network
Data Management		

CONTENTS

Section Page

Section Page

FIGURES

Figure		Page

Figure		Page

ACRONYMS

ASCII	American Standard Code for Information Interchange
BCOT	Local batch origin type
CAI	Computer Adapter Interface
CDC	Control Data Corporation
CIO	Common input/output processor
CM	Central Memory
CPA	Control Point Area
CPS	Characters per second
CPU	Central Processor Unit
CPU/CM	Central Processor Unit/Central Memory
CPU/M	Central Processor Unit/Memory
CPUMTR	Central Processor Monitor
CRT	Cathode ray tube
CTM	Communications Terminal Module
DIS	Document Index System
ECS	Extended Core Storage
FL	Field length
FNT	File Name Table
I/O	Input/Output
IQP	Initial queue priority
JOT/QP	Job origin type/Queue priority
JSC	Lyndon B. Johnson Space Center
KRONOS	Control Data Corporations Time-Sharing Operating System
LQP	Lowest queue priority
MMDB	Master Measurements Data Base
MRI	MRI Systems Incorporated
MTOT	Multiterminal origin type
MTR	Periphial processor monitor
PDS	Problem Data Systems

PFM	Permanent file manager
PICRS	Program Information and Coordination Review Service
PLI	Procedural Language Interface
PP	Periphial Processor
RA	Reference address
RSPI	Resources, Scheduling, and Procurement Integration
S2K	System 2000 data base access
S2KMM	System 2000 Multiuser Multithread
SBM	Special binary mode
SIC/RSB	Intercontrol Point communications requests
SIU	Serial Interface Unit
SLAHTS	Stowage List and Hardware Tracking System
SPIMS	Shuttle Program Information Management System
SRT	Shuttle Requirements Traceability
SYOT	System origin type
TCS	Terminal Control System
TTY	Teletype
TXOT	Terminal origin type
UQP	Upper queue priority
VF	Voice frequency
XJP	CPU-to-monitor request

1.0 INTRODUCTION

1.1 BACKGROUND

The Shuttle Program Information Management System (SPIMS) project was established at the Lyndon B. Johnson Space Center (JSC) to automate the storage and retrieval of massive volumes of data relating to several management responsibilities assigned by the Space Shuttle Program Office (SSPO).

Applications developed for SPIMS in support of these management responsibilities were identified and allocated to JSC by SSPO (see the Program Information and Coordination Review Service (PICRS) Management Plan). These applications currently are:

- Document Index System (DIS)
- Shuttle Requirements Traceability (SRT)
- Problem Data Systems (PDS)
- Stowage List and Hardware Tracking System (SLAHTS)
- Master Measurements Data Base (MMDB)
- Shuttle Automated Mass Properties System
- Resources, Scheduling and Procurement Integration (RSPI)

The SPIMS Functional Design is the result of two studies, a feasibility study performed during the second quarter of fiscal year 1974 and a requirements study performed during the third and fourth quarters of 1974. The results of the feasibility study can be found in the Implementation Feasibility Study for Space Shuttle Program

Management System Application Final Report. This study concluded that an automated data management system could be implemented at JSC. The results of the requirements study can be found in the *System Requirements Document for the Shuttle Program Information Management System*.

The feasibility study identified the principle computer for SPIMS applications as a Control Data Corporation (CDC) CYBER 70 series computer. This series of computers provides a distributive processing capability with both upward and downward software compatibility. Additional applications are available using the UNIVAC 1100 computer series. Unless otherwise stated, all interfaces described in this document are relative to the CDC CYBER 74 computer as the applications processor. The KRONOS Time-Sharing Operating System was selected as the terminal interactive control executive for the CDC CYBER 74. MRI Systems Corporation's SYSTEM 2000* was selected to provide data management services to the SPIMS applications. The Terminal Control System (TCS) in the Mission Control Center (MCC) was designated to provide data buffering and terminal network control.

The requirements study provided an introduction to the Shuttle Program Information Management System. An overview of the existing capabilities of the operational systems hardware and software components was presented. A brief description of the applications within SPIMS was presented.

*Trademark of MRI Systems, Corp., Austin, Texas.

Section 3 of the System Requirements Document for the Shuttle Program Information Management System specified the requirements placed upon each functional element of the SPIMS. Figure 1-1 illustrates these functional elements. Certain requirements are currently supported, while others necessitated enhancements to existing capabilities or the development of new capabilities.

1.2 PURPOSE

This Systems Functional Design Document presents the normal actions or activities provided by each element of SPIMS. These generic activities are:

- Support a mixed terminal-type environment
- Support communications control for two basic protocols (synchronous, asynchronous)
- Support an interactive multiprogramming data base/batch applications environment

This document indicates the element-to-element dependencies to support these activities.

1.3 DOCUMENT OVERVIEW

This document provides a description of the functional capabilities and interfaces required for the integrated Shuttle Program Information Management System (SPIMS). First and second level sections and paragraphs provide the reader with a generalized functional design of SPIMS by subsystem. Lower level paragraph numbers within each

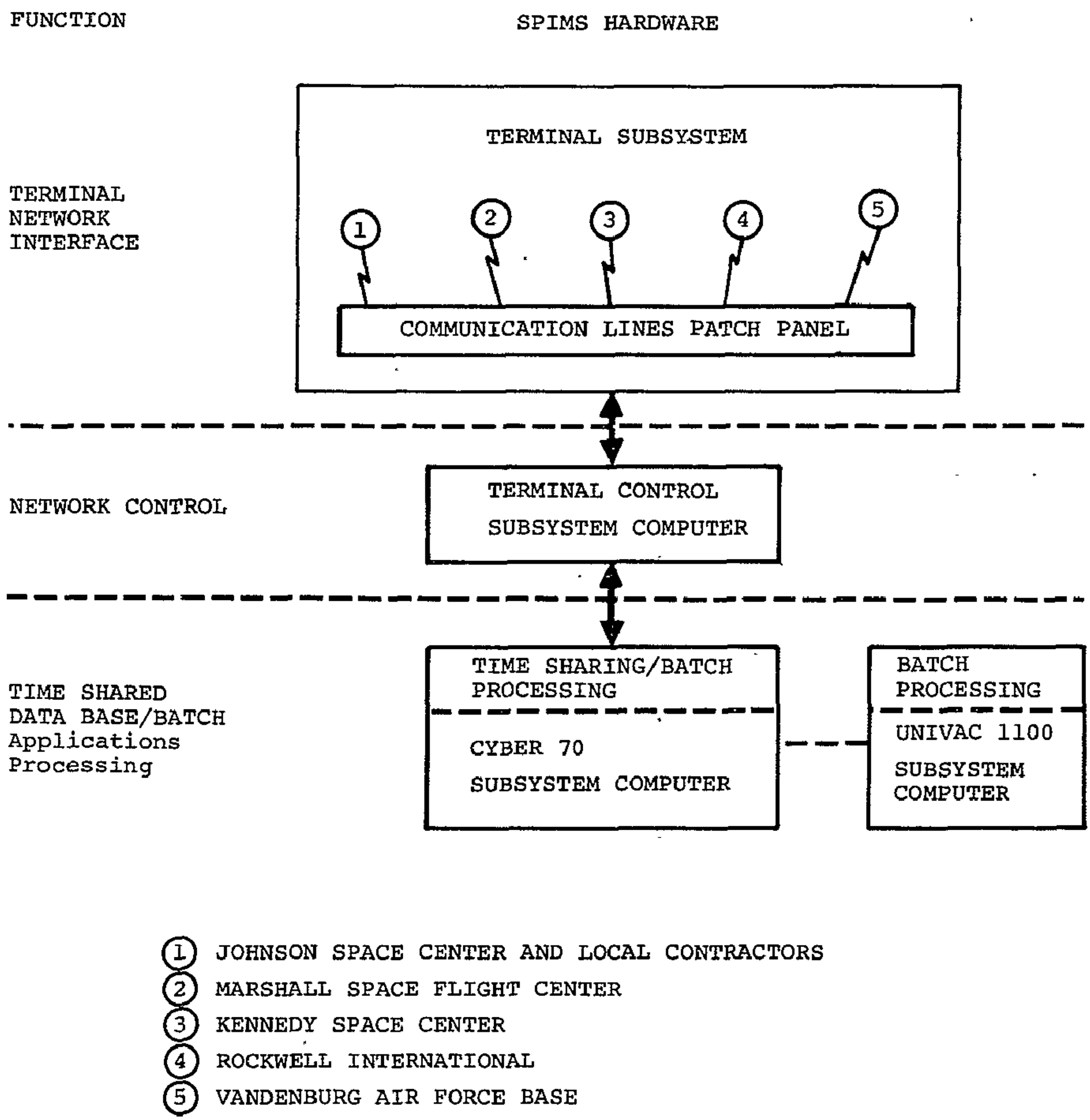


Figure 1-1. – Functional elements of SPIMS.

section provide the reader with a detailed functional design of each element within the subsystem.

Section 2.0 provides a description of the hardware elements of the SPIMS and the functions these elements support. Section 3.0 presents the basic interfaces and responsibilities of each software subsystem within SPIMS. The functions provided by these subsystems are the result of incorporating enhancements with prior capabilities and the addition of required new capabilities. Section 4.0 provides an overview of all terminal data paths provided by the SPIMS.

Section 5.0 presents a generalized SPIMS Applications design utilizing the functional elements of the SPIMS. Several design considerations are presented. Section 6.0 outlines the SPIMS Controllers responsibilities and the functions available within SPIMS to provide the monitoring and control these responsibilities required. Section 7.0 provides a summary of the capabilities provided by this functional design.

2.0 SYSTEM HARDWARE

2.1 SPIMS HARDWARE CONFIGURATION OVERVIEW

SPIMS is an integration of four hardware subsystems - the Terminal Subsystem, Terminal Control Subsystem, CYBER 70 Subsystem, and the UNIVAC 1100 Subsystem. Each subsystem supports a unique function for the total system. These functions are:

- Terminal network
- Network control
- Time shared-data base/Batch applications processing
- Batch applications processing

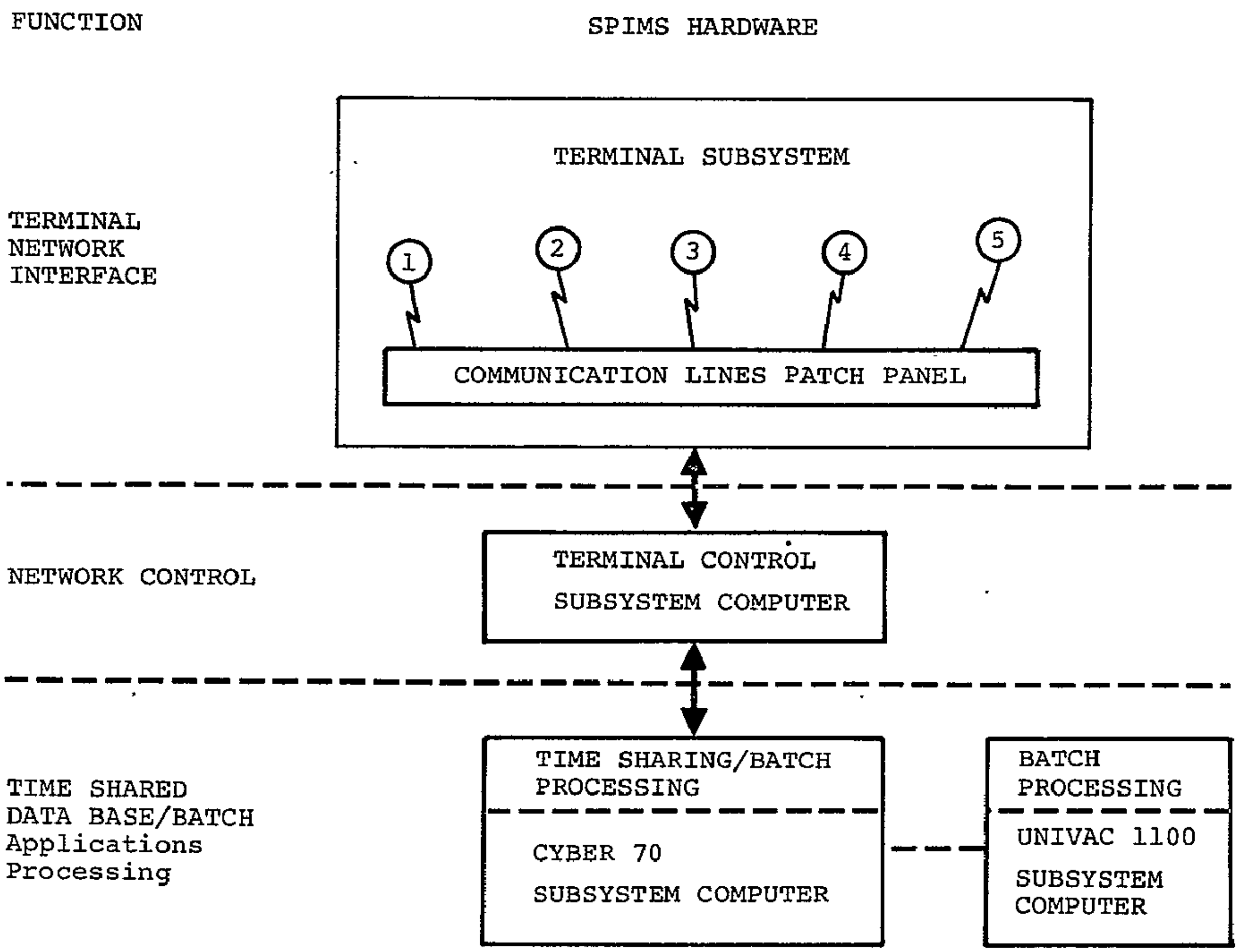
Figure 2-1 illustrates the SPIMS functional hardware configuration.

2.2 TERMINAL SUBSYSTEM CONFIGURATION

The Terminal Subsystem configuration is illustrated in figure 2-2. Each terminal type has different functional capabilities. The Voice Frequency (VF) Patch Bay provides the ability to reconfigure the on site/off site communications lines to their required modern groups.

2.2.1 Teletypewriters

These devices transmit data, one character at a time, using the American Standard Code for Information Interchange (ASCII). Transmission speeds range from 11 to 15 characters per second (CPS). The mode of communication is half duplex.



1 JOHNSON SPACE CENTER AND LOCAL CONTRACTORS
2 MARSHALL SPACE FLIGHT CENTER
3 KENNEDY SPACE CENTER
4 ROCKWELL INTERNATIONAL
5 VANDENBURG AIR FORCE BASE

Figure 2-1. — SPIMS functional hardware configuration.

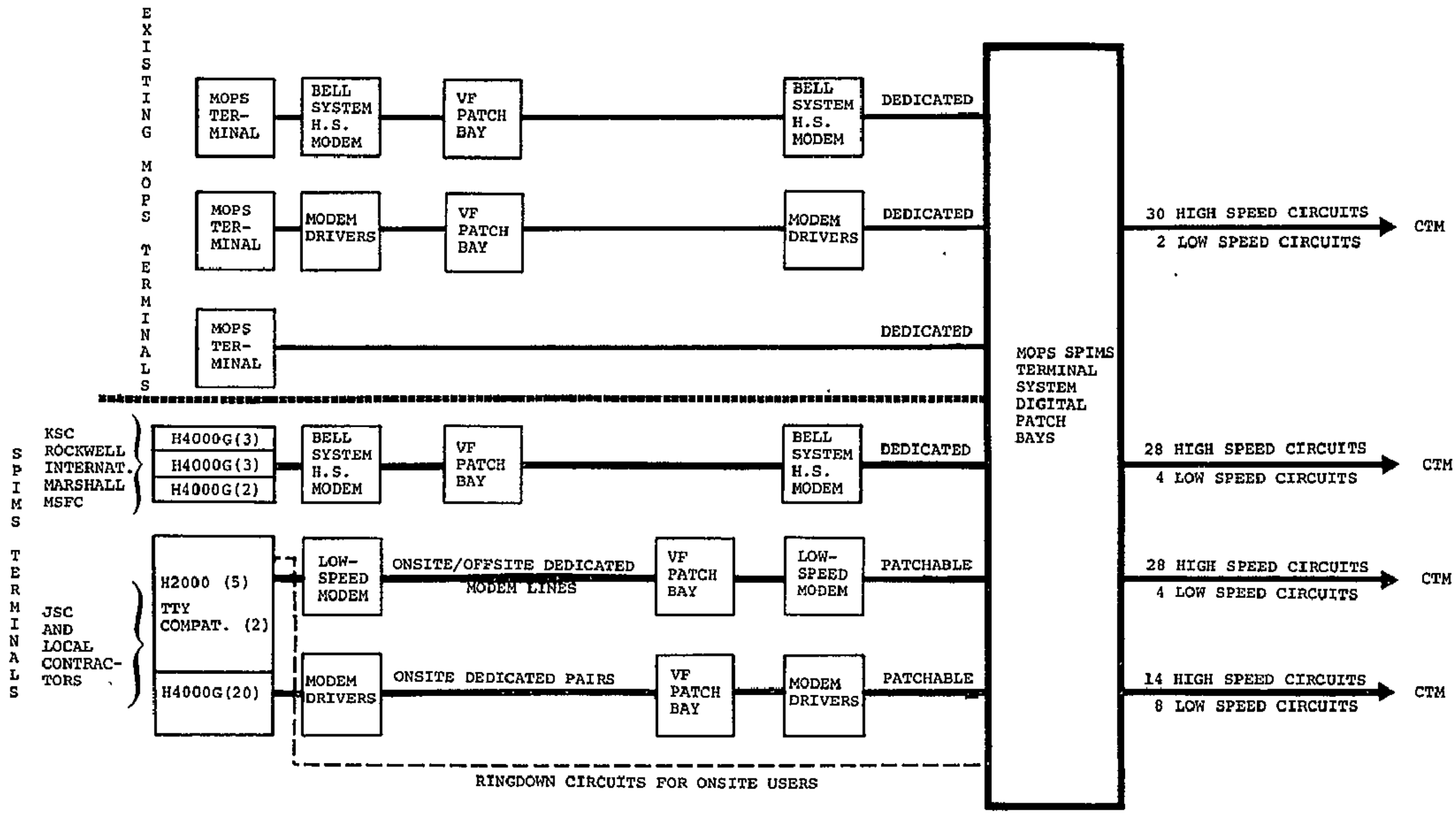


Figure 2-2. — Terminal subsystem configuration.

The protocol is asynchronous. Character parity is odd. In addition to the parity bit each 8 bit character is framed within one start bit and two stop bits for transmission purposes. Thus, each character (including the framing bits) constitutes a "message" on the transmission line.

2.2.2 Teletype (TTY) Compatible Devices

These devices utilize the same communications mode, protocol and parity as a teletypewriter and provide transmission speeds up to 30 CPS. Terminal input/output (I/O) is impact printed and/or displayed on a cathode ray tube (CRT).

2.2.3 Hazeltine 2000 CRT with the Hardcopy Device Option

These devices will operate in Batch mode using the same communications conventions as TTY devices. The Batch mode provides the following significant enhancements to the TTY compatible devices.

- The ability to send or receive asynchronously from 1 to 1,998 ASCII characters plus control bytes in a single transmission
- The ability to address any character position within the 27-lines by 74-character matrix
- The ability to define from the remote computer, data display and data entry only areas
- A local display edit function (e.g., insert line, insert character, delete line, delete character)
- The ability to select a hardcopy of the displayed data independent of other SPIMS subsystems

An indepth description of the Hazeltine 2000 may be found in the Hazeltine 2000 Operating Manual, HI1004, published by the Hazeltine Corporation.

2.2.4 Hazeltine 4000G CRT

Each terminal unit shall be configured with the following:

- A Hazeltine 4000G CRT terminal unit with alphanumeric, limited graphic capability and a detached keyboard
- An optional CRT display copier hardcopy unit
- An optional function key module

These devices shall operate in a mode analogous to the Hazeltine 2000 Batch mode. The communications mode is half duplex. The protocol is synchronous. Character parity is odd. Each character is 8 bits in length. Control characters are added to the beginning and end of each character string for transmission. Thus, a "message" constitutes the two control characters and the data. The Hazeltine 4000G Batch mode in conjunction with the terminal configuration provides the following enhancements:

- The ability to send or receive from 1 to 3,996 alphanumeric characters plus control bytes in a single transmission
- The ability to address any character position within the 53-lines by 74-character matrix
- Synchronous data transmission rates of 600 CPS or 1200 CPS

- User initiatied Selective Transmission of portions of the 53-lines by 74-character matrix. The starting and ending location of the character string on the display are automatically included in the message.
- Limited graphics where horizontal and vertical lines can be displayed
- The ability to transmit function codes via 96 function keys

An indepth description of the Hazeltine 4000G CRT may be found in the CRT Terminal System Software Specifications, PHO SI-09637A.

2.3 TERMINAL CONTROL SUBSYSTEM (TCS) CENTRAL PROCESSOR UNIT (CPU) CONFIGURATION

A TCS single CPU configuration is illustrated in figure 2-3. The normal configuration for TCS provides network control and hardware redundancy for all elements illustrated. This redundancy is accomplished by configuring two CPU's and their critical peripheral equipment to share the control function. The configuration illustrated in figure 2-3 is capable of assuming total terminal network control with a minimum observable degradation. More detailed information about the hardware is available in the Systems Requirements Document for the Shuttle Program Information Management System, JSC-09381.

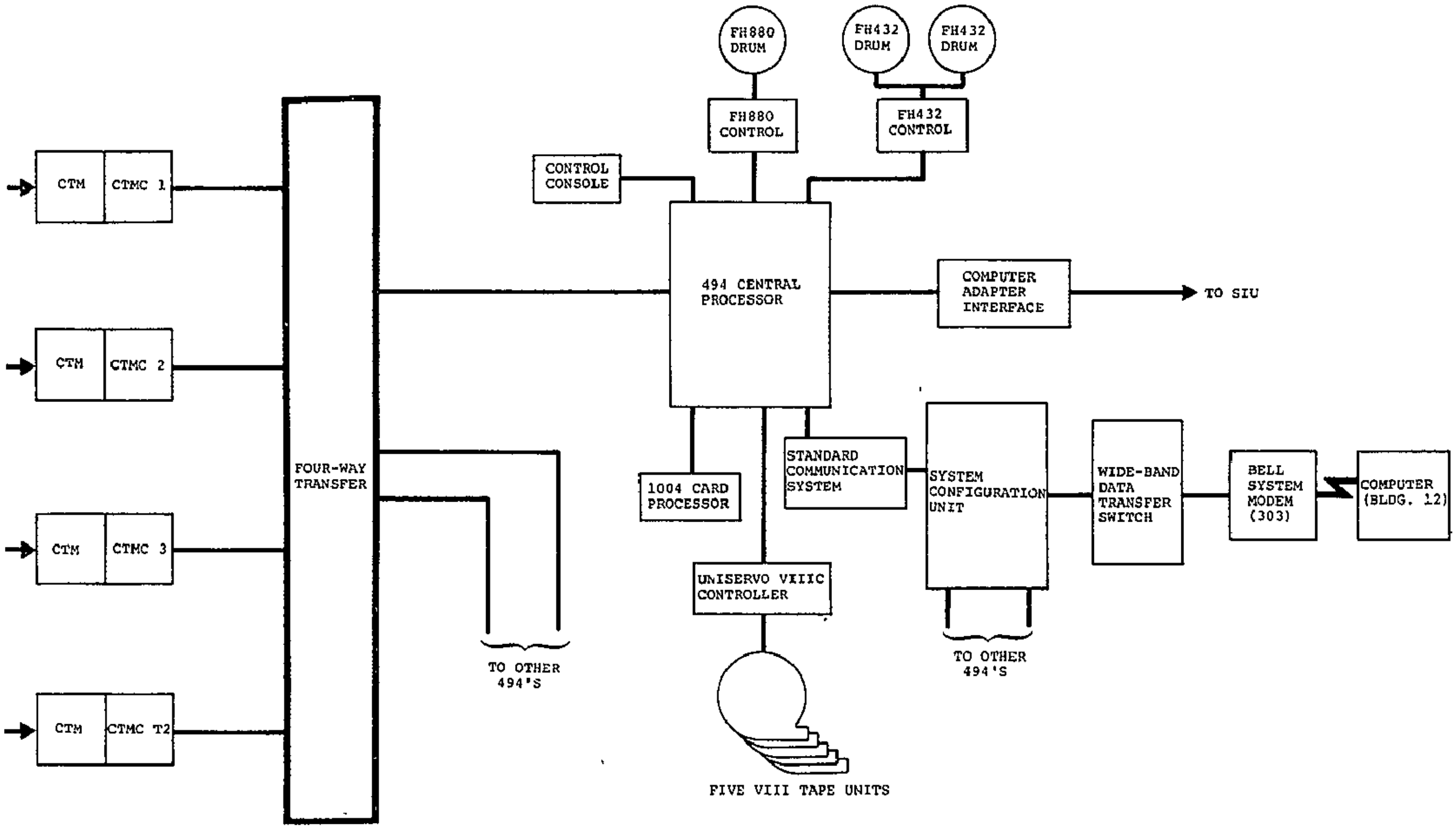


Figure 2-3. — SPIMS single CPU TCS hardware configuration.

2.3.1 U494 Central Processor Unit/Memory (CPU/M)

The CPU/M has an arithmetic, a control, and an I/O section. The memory units total 131,032 30-bit words. A parity bit is provided for each half word (15 bits). A computer control console is attached to provide monitoring and control of the computer operations.

2.3.2 Direct Access Storage Devices and Controllers

Two types of direct access storage are provided by TCS; they are magnetic tape and magnetic drums. The magnetic drums are further classified as high speed or large capacity.

- Magnetic tapes provide for the recording and retrieval of data in a sequential mode. Data is transferred at rates from 24,000 to 96,000 frames per second depending on the recording density.
- Magnetic drums-high speed (FH432) provide for the recording and retrieval in a random or sequential mode for 262,144 30-bit words plus parity. High speed is provided based on an average access time of 4.33 milliseconds and a 240,000 words per second (maximum) transfer rate.
- Magnetic drums-large capacity provide for the recording and retrieval in a direct access mode (word addressable) for 786,432 30-bit words plus parity. This provides the large storage capacity. Average access for any word is 17 milliseconds coupled with a 60,000 words-per-second transfer rate.

2.3.3 Printers/Card Reader and Controllers

Two printer devices are available within the TCS.

- A 600 lines-per-minute, 132 characters-per-line device
- A 1,200 to 1,600 lines-per-minute, 132 characters-per-line device.

One card reader is provided with 80 column Hollerith and column binary modes. The maximum reading speed is 615 cards per minute.

2.3.4 Communications Line Interfaces and Controllers

The Communications Terminal Modules (CTM's) functioning in concert with their controllers govern the interfaces between the CPU and the devices conforming to the accepted standards for serial data transmission in odd parity. Two types of transmission devices are supported:

- Low speed, asynchronous devices with transmission rates up to 30 CPS
- High speed, synchronous devices with transmission rates up to 1200 CPS

2.3.5 Computer Adapter Interface (CAI)

The CAI provides an 81.6 thousand bits per second serial bidirectional channel between the TCS and the CYBER 74 Subsystem.

2.4 CYBER 70 SERIES SUBSYSTEM CONFIGURATION

A representative SPIMS CYBER 74 Subsystem configuration is illustrated in figure 2-4. The CYBER 74 is the primary computer supporting time sharing and Batch processing of the SPIMS data bases.

The CYBER 74 Subsystem provides the resources required to support multidata bases with on-line update and retrieval capabilities and Batch application processing. More detailed information about the hardware is available in the _Systems Requirements Document for the Shuttle Program Information Management System_, JSC-09381.

2.4.1 Central Processor Unit/Central Memory (CPU/CM)

The Central Processor Unit provides 10 functional units. The functional units provide the multiprocessing of arithmetic, manipulative, and logical operations. These instructions are stored and retrieved as needed from Central Memory (CM). Central Memory (CM) consists of 131,072 60-bit words. The CPU to CM interface is one of two interfaces supported by the Central Processor Unit. The second CPU interface is direct transfers to Extended Core Storage (ECS).

2.4.2 Input/Output, Operator Console Interfaces

The CYBER 74 Subsystem provides direct channel to device interfaces. Figure 2-4 illustrates a channel to device configuration. Each channel is accessible by any of

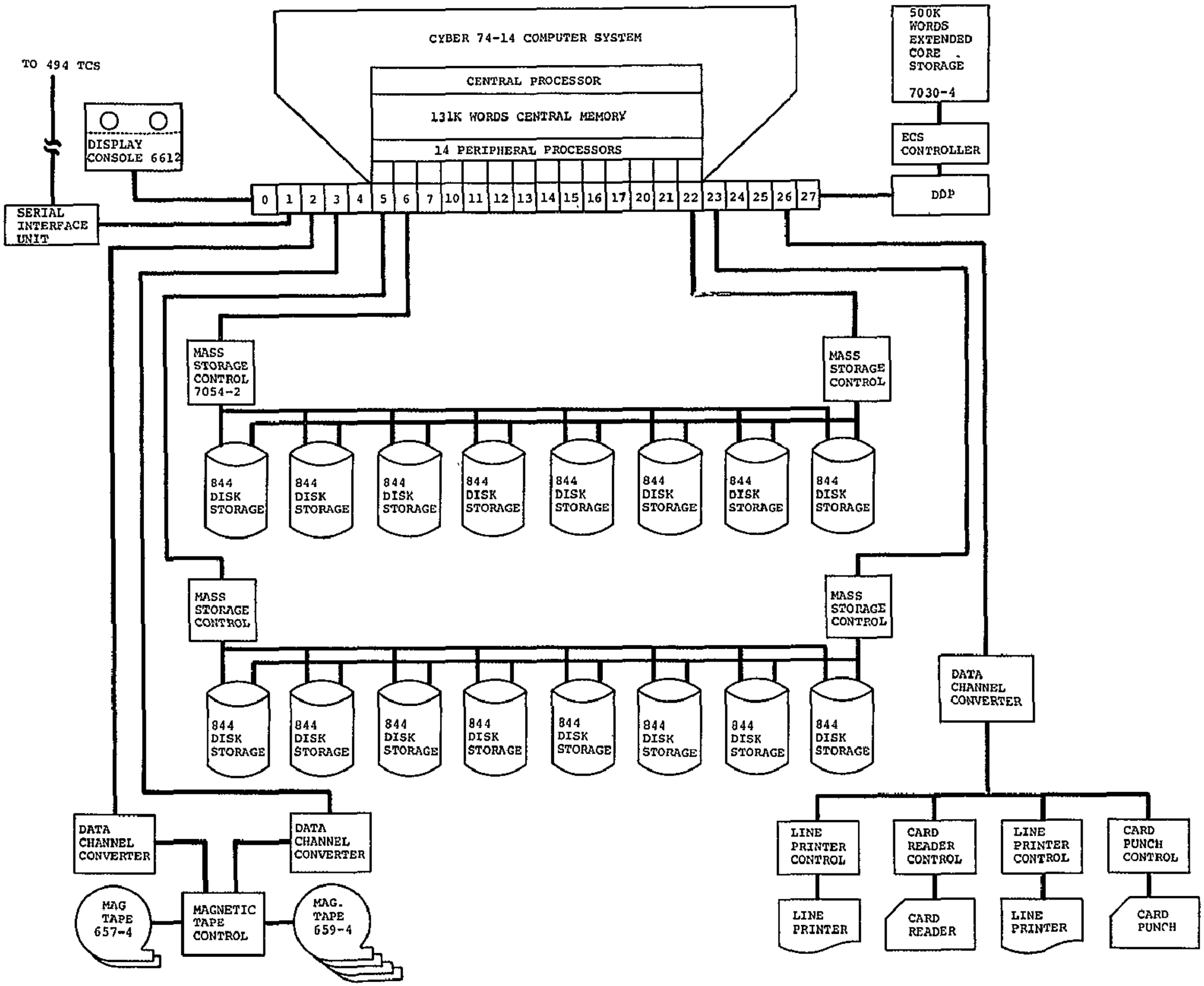


Figure 2-4. — SPIMS CYBER 70 Subsystem configuration.

the 14 Peripheral Processors (PP's). Bidirectional data transfers are supported for each active PP/channel combination concurrently.

All inactive PP's constitute the Peripheral Processor Pool. The PP's operate independent of the CPU and concurrent with each other as programmed interfaces between Central Memory and the peripheral devices. Central Memory program I/O requests are assigned to the first available Pool Peripheral Processor. This PP provides channel selection and data transfer control independent of the CPU and other PP's.

A dedicated PP provides the interface and command interpreter between the Operator Console and the CPU monitor.

2.4.3 Direct Access Storage Devices and Controllers

Three types of direct access storage devices are provided by the CYBER 74 Subsystem - magnetic tapes, disk drives, and extended core storage.

- Magnetic tapes provide the recording and retrieval of data in a sequential mode. Data is transferred at rates from 30 thousand to 240 thousand frames per second.
- Disk drives provide a high speed random access for 110 million characters per removeable pack. Data transfers occur at a maximum rate of 925 thousand CPS with an average access time of 44.3 milliseconds.

- Extended core storage (ECS) provides the storage and retrieval in a direct and indirect access mode of 500 thousand 60-bit words plus parity. Indirect access mode transfers are supported via the Distributive Data Path and any PP. The CPU (being isolated from I/O) continues to process. Direct access mode transfers are supported via a single CPU instruction. The PP's can access CM during an ECS transfer however contention may cause a one microsecond delay for the PP's CM access request

2.4.4 Printer/Card Reader/Card Punch and Controllers

Two printers are available within the CYBER 74 Subsystem. They support a print rate of 1,200 lines per minute, 136 characters per line.

The card reader supports 80 column Hollerith and column binary transfer modes. The maximum reading speed is 600 cards per minute.

The card punch supports 80 column Hollerith and column binary punch modes. The maximum punch speed is 200 cards per minute.

2.4.5 Serial Interface Unit (SIU)

The SIU provides an 81.6 thousand bits-per-second serial, bidirectional interface between the CYBER 74 and the TCS Subsystem.

3.0 SPIMS FUNCTIONAL DESIGN

The _Functional Design Document of the Shuttle Program Information Management System_ delineates the basic interfaces and responsibilities of each subsystem within SPIMS. This design is the result of merging the unique capabilities of already existing subsystems and, where necessary, enhancing these subsystems. The requirements for these enhancements are found in the _Systems Requirements Document for the Shuttle Program Information Management System_. As a result of the enhanced subsystems and the responsibilities assigned to them, a transaction oriented system has been designed.

A SPIMS transaction is defined as the process by which an online system supports a programmed function to a predetermined conclusion. A transaction is initiated by an individual user. A transaction includes the input message, the program controlled computer processing, and the transmission of any output messages associated with the function. Figure 3-1 illustrates a SPIMS transaction and the generic subsystems required to assure completion of the action.

3.1 TERMINAL SUBSYSTEM FUNCTIONAL DESIGN

The Terminal Subsystem is designed to support three basic functions:

- Provide an entry to SPIMS for all terminal users.

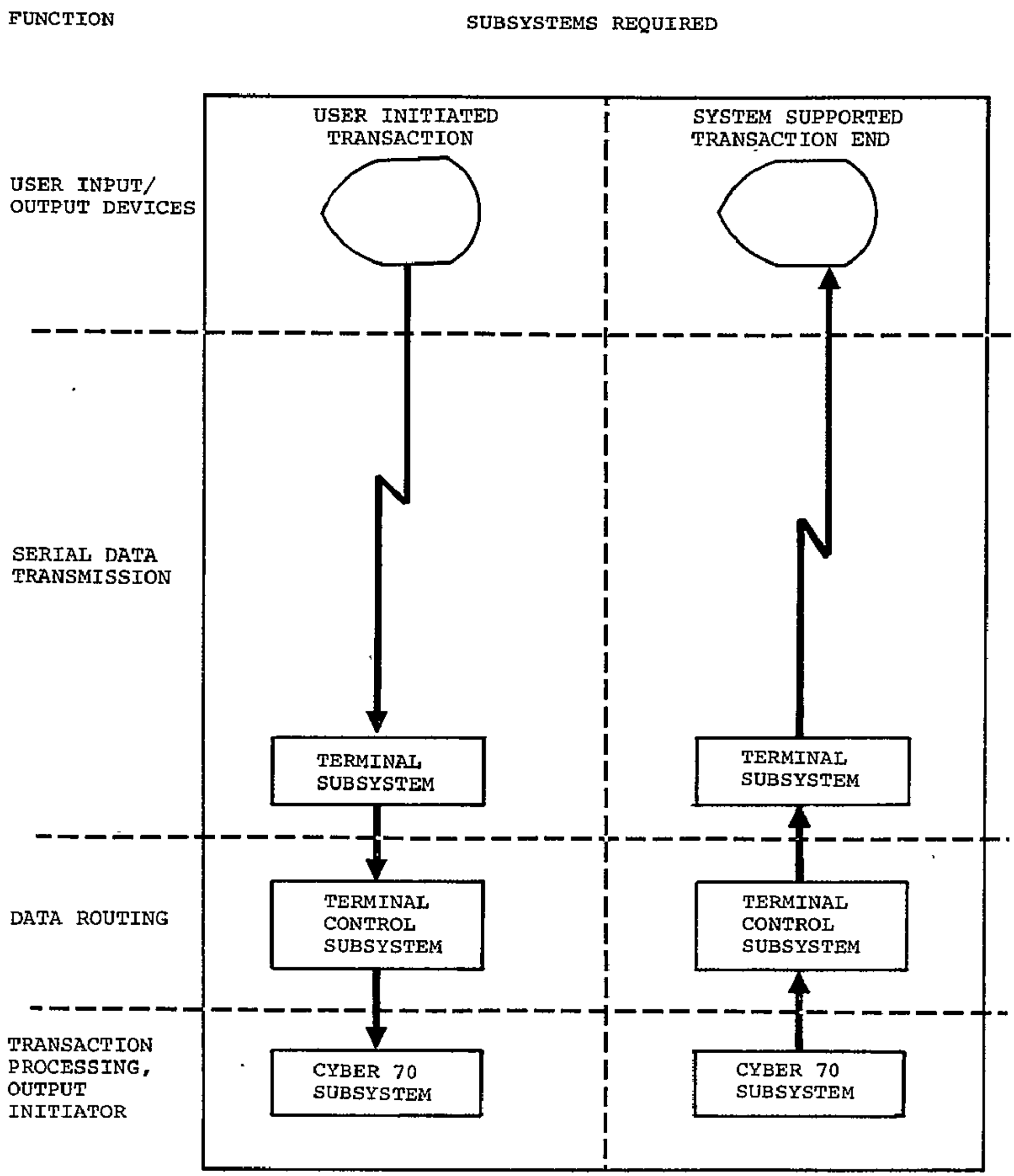


Figure 3-1. – SPIMS transaction support.

- Assure that appropriate telecommunications protocols for the serial data transmissions lines are provided.
- Assure that the terminals provide timely support for the bi-directional transmission of large volumns of data.

These functions have been satisfied (fig. 2-2) by the terminals and the incorporation of high speed modems, low speed modems, modem drivers, and their associated communications lines. This subsystem is a hardwired system with no computer software required.

3.2 TERMINAL CONTROL SUBSYSTEM FUNCTIONAL DESIGN

The Terminal Control Subsystem (TCS) is designed to interface a large number of diverse user-terminal devices with terminal applications residing in a multihost computer environment. In performing this function, the TCS provides a comprehensive set of communications-oriented services to facilitate terminal and applications communications and control. The following subsections describe the functional design of those services provided by TCS and required by SPIMS. More detailed information may be found in the Terminal Control System Requirements and User's Guide, GDSD, SSB-004.

3.2.1 SPIMS Supported TCS Functional Requirements

The Terminal Control Subsystem (TCS) will support the following SPIMS functions:

- Handshaking processing (function codes/commands) to validate all user interfaces
- Recognition and validation of terminal user and terminal type
- Establishment of and continued recognition of terminal routing schemes
- Circuit assurance processing to confirm a working interface between the user terminal and the TCS
- Demand/response processing mode for message flow
- Message format validation to assure compliance to predefined message headers, text, and trailers
- Message traffic accounting
- Maintain a maximum of 42 active terminals

3.2.1.1 <u>Function Code Processing</u>. Fundamental to the design of TCS is a set of predefined function codes which provide the handshaking between elements. These elements may be divided into the following interfaces:

- Hardware to hardware
- Terminal to TCS
- Terminal to application
- Applications to network
- Application to terminal

Hardware to hardware interfaces are provided by the hardware configurations illustrated in figures 2-1, 2-2, 2-3, and 2-4. These hardware connections must exist for SPIMS to operate.

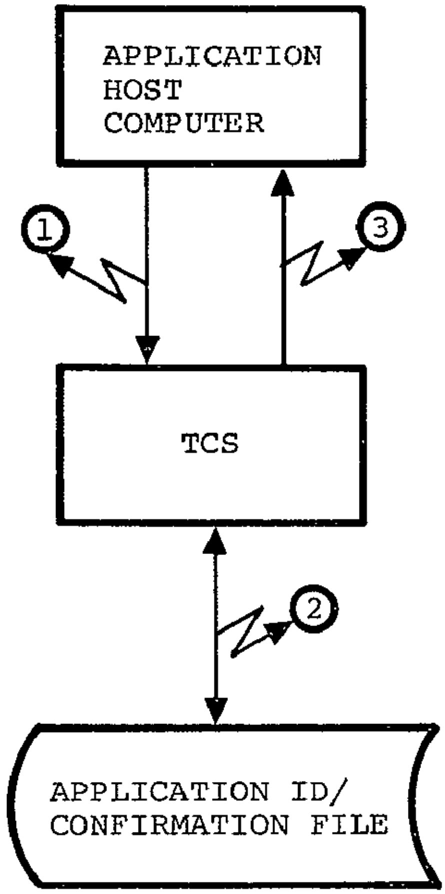
Applications to network accesses are provided by TCS through the Application Initialization/Termination

Functions. Figure 3-2 illustrates these functions and the sequence of events required to indicate an application is online or not currently available to SPIMS users.

Terminal to TCS access is provided by TCS through the processing of the user sent $TON,XXXX,Y command. Figure 3-3 illustrates this sequence of events. The disabling of this access is provided by the command $TOF,XXXX. Figure 3-4 illustrates this sequence.

Terminal to application/application to terminal routing schemes are established by TCS using a logical data path concept. The logical data path (LDP) concept establishes the bidirectional link between the terminal and an application in the host computer. (Section 4.0 is a detailed description of all SPIMS LDP's.)

The message addressing scheme used is: source/destination. The source code indicates the message initiator. The destination code indicates the message recipient. A unique numeric code, Externally Specified Index (ESI), is assigned for each active terminal. A numeric code is assigned for each available SPIMS application. Thus, each terminal user of SPIMS has a unique terminal-application code combination. The position of these codes in the TCS message header establishes the direction TCS is to transmit the data. Figure 3-5 illustrates the functions performed by TCS to establish and maintain the data routing scheme. Figure 3-6 illustrates a TCS data flow utilizing the established LDP's. Figure 3-7 illustrates a TCS data stream overview to SPIMS.



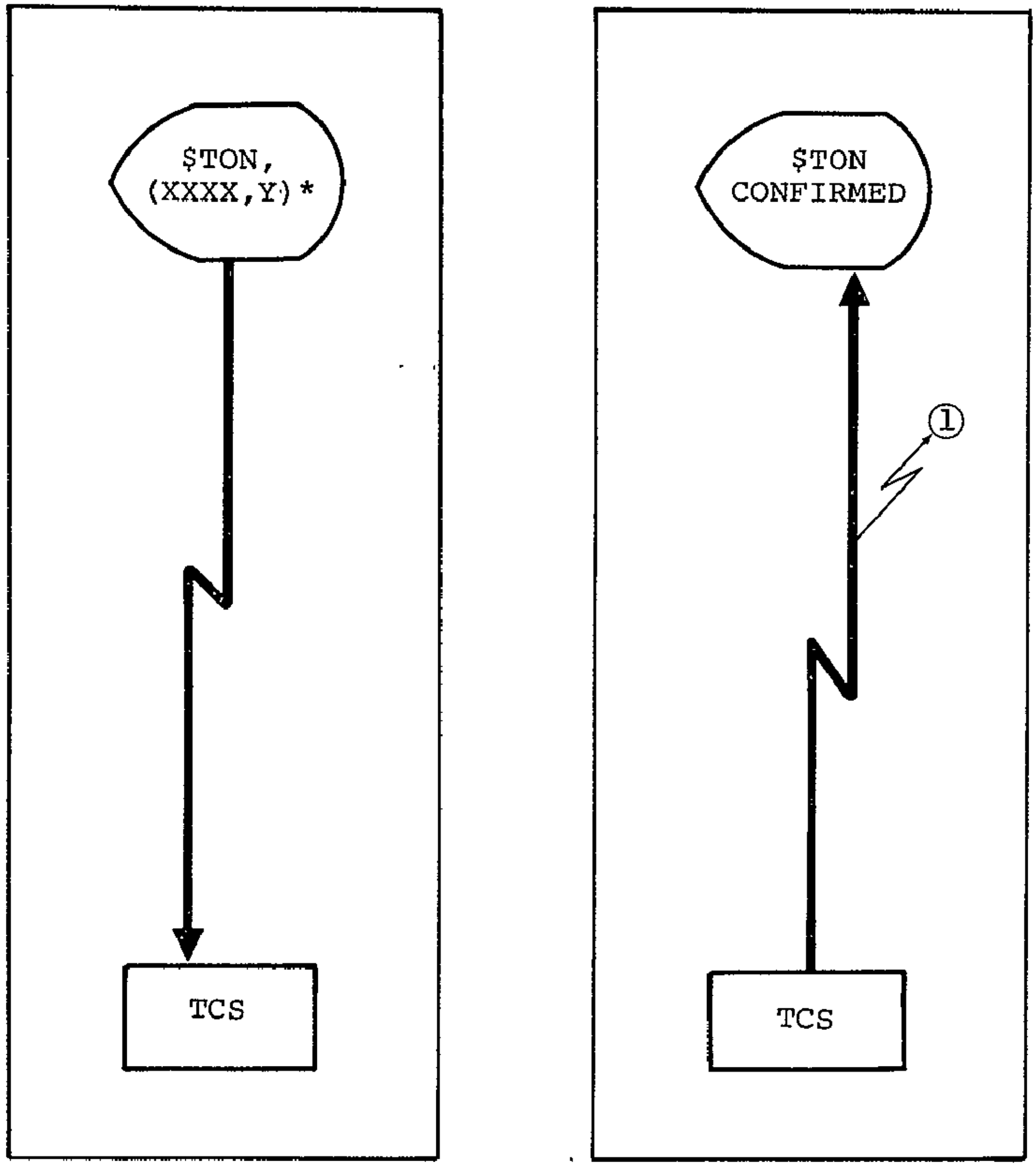
INITIALIZATION

1. APPLICATION INITIALIZATION REQUEST (FC121) BY APPLICATION ID
2. TCS VALIDATION OF APPLICATION ID
3. APPLICATION INITIALIZATION CONFIRMATION (FC122) OR TCS TO HOST ERROR (FC175) MESSAGE: "APPLICATION ID MISSING, APPLICATION ID INHIBITED"

TERMINATION

1. APPLICATION TERMINATION REQUEST (FC124) BY APPLICATION ID
2. TCS VALIDATION OF APPLICATION ID
3. APPLICATION TERMINATION CONFIRMATION (FC125) OR TCS TO HOST ERROR (FC175) MESSAGE: "APPLICATION NOT INITIALIZED"

Figure 3-2. — TCS application initialization/termination function code processing.

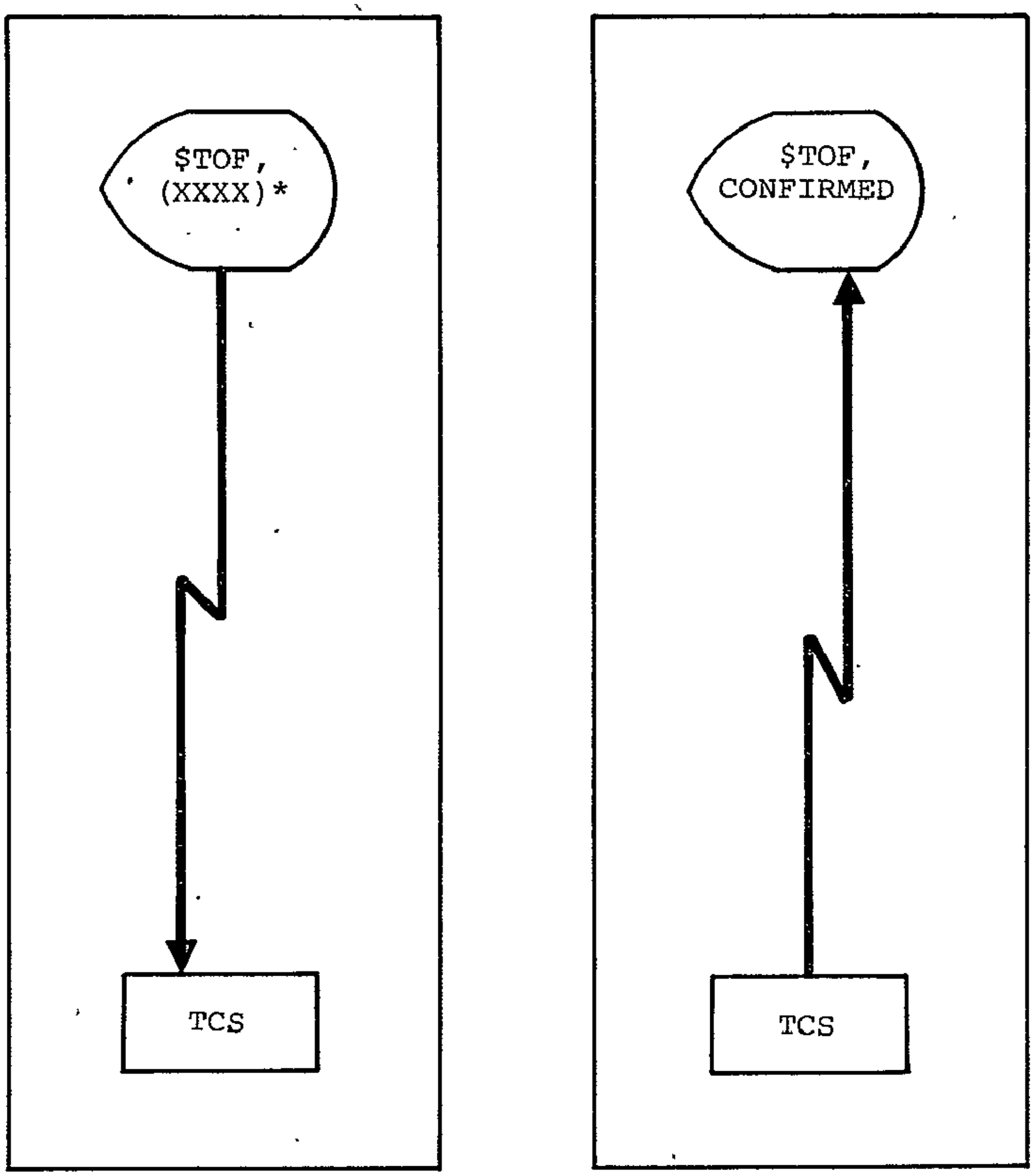


*XXXX = TERMINAL USER ID

Y = DEVICE TYPE 1 – H4000G SYNCHRONOUS
DEVICE TYPE 2 – H4000G ASYCHRONOUS
DEVICE TYPE 3 – H2000G ASYCHRONOUS
DEVICE TYPE 4 – TTY COMPATIBLE ASYCHRONOUS

1 Illegal user ID or terminal type causes nonconfirmation

Figure 3-3. – Terminal-to-TCS access with $TON,XXX,Y command.



*XXXX = TERMINAL USER ID

Figure 3-4. — Terminal-to TCS disconnect with $TOF,XXXX command.

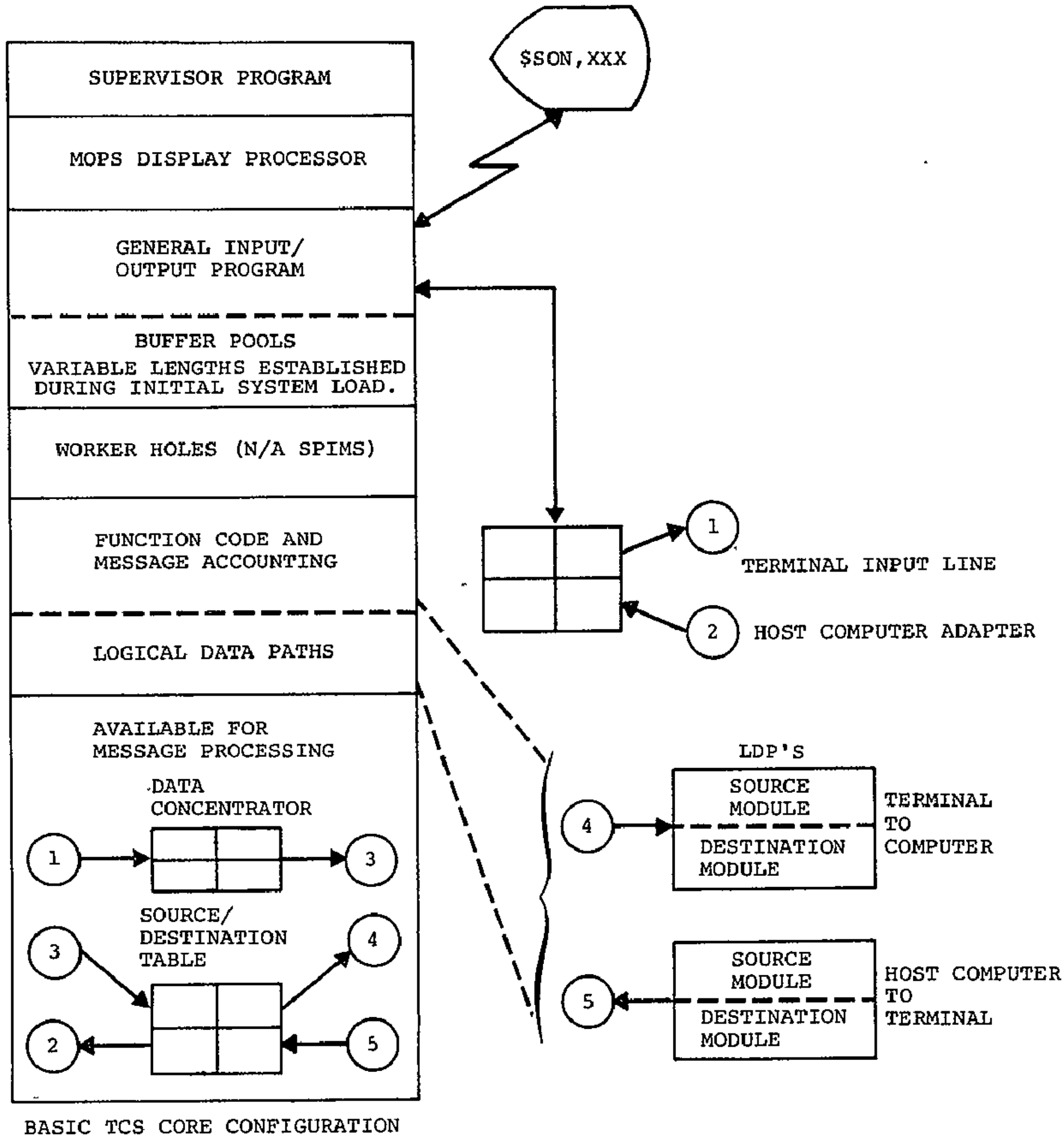


Figure 3-5. — TCS DATA ROUTING SCHEME

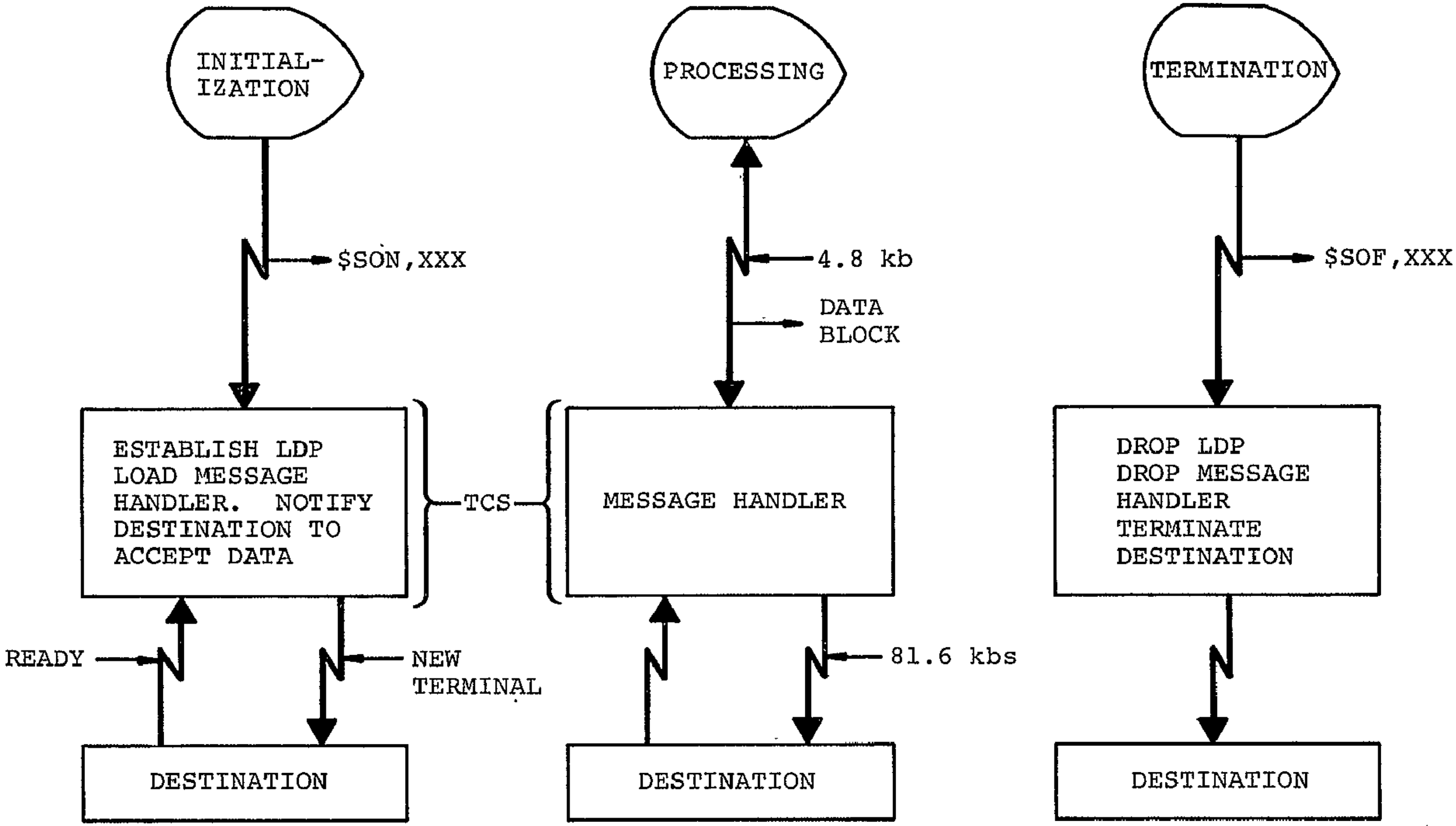


Figure 3-6. — TCS data flow utilizing LDP's.

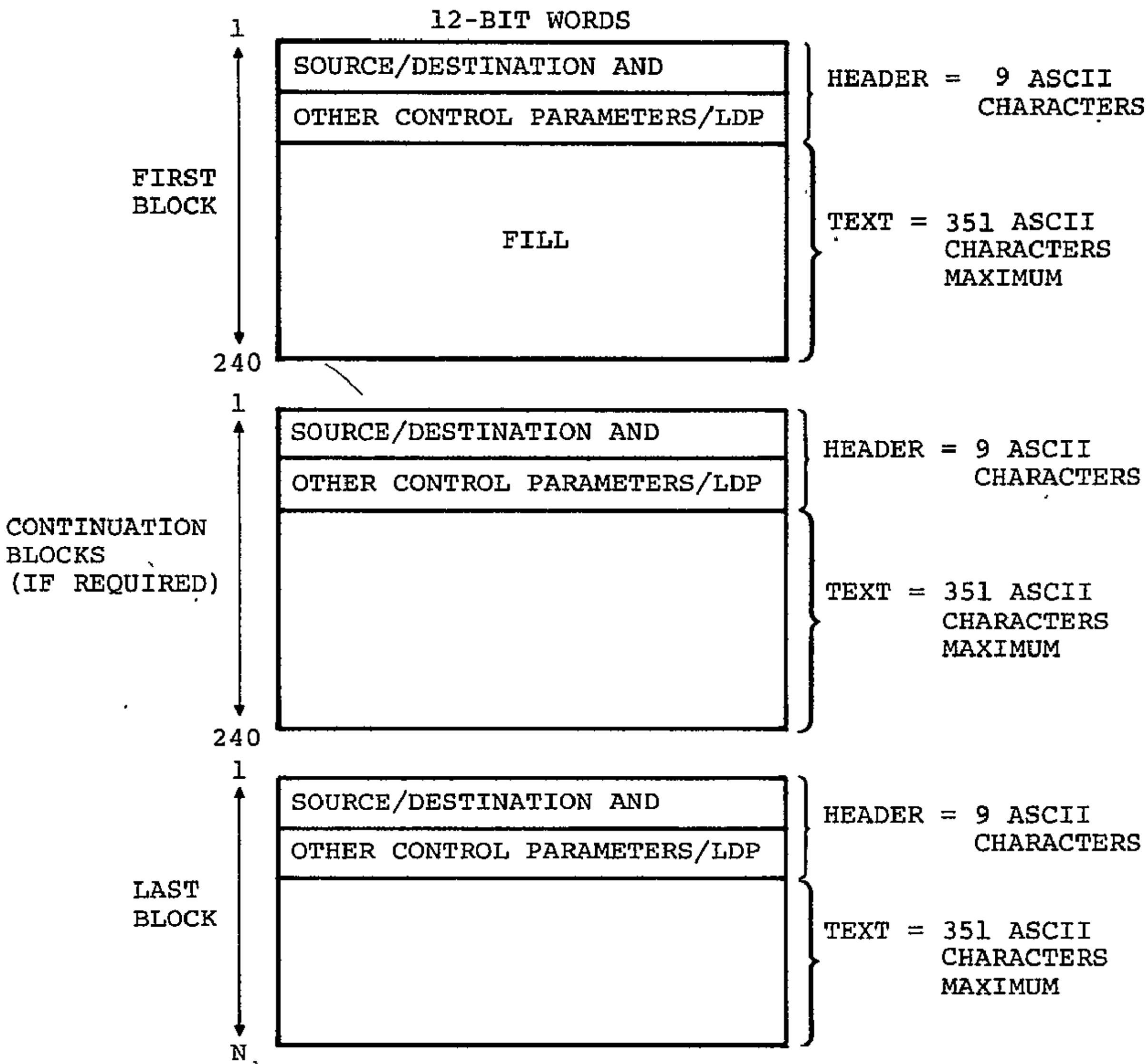


Figure 3-7. — TCS data stream overview.

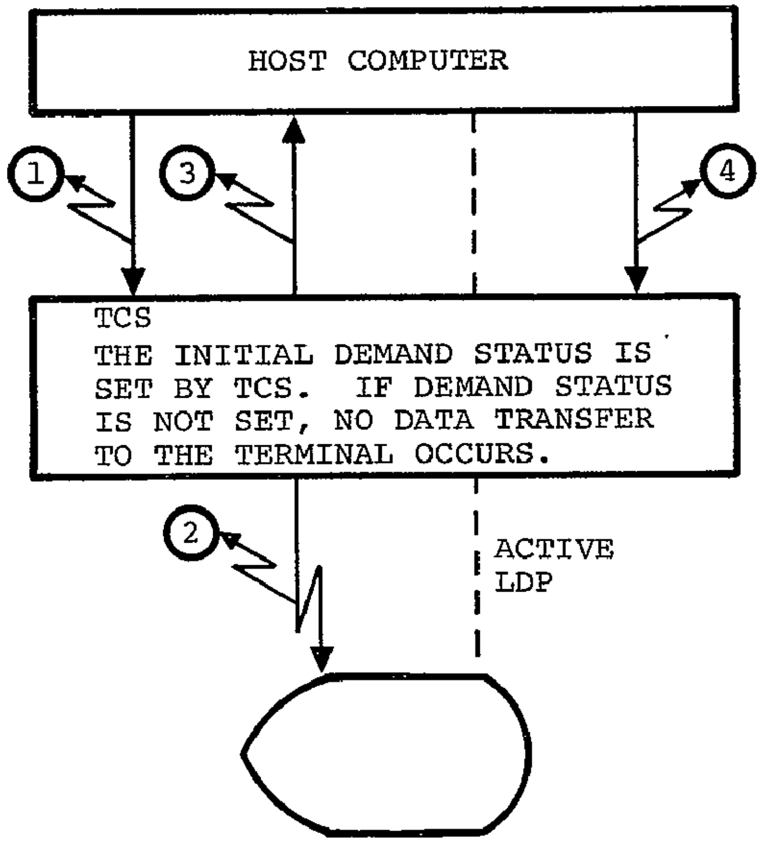
3.2.1.2 Demand/Response Processing Function. This function, in conjunction with the TCS capability to interleave messages, provides control of data transfers among the various elements of SPIMS. Demand/response processing provides the message load leveling required by SPIMS to interface the applications executing in the CYBER 74 Subsystem with terminal users operating at various slower transmission line speeds. See figure 3-8.

Each user terminal within SPIMS is processed by TCS as a "free wheeling" input device. A "free wheeling" terminal's input is not computer controlled. TCS must buffer the input from start of Transmission to end of transmission. Data is collected into fixed message blocks and transmitted to the host computer in an interleaved, segmented fashion. Figures 3-9 and 3-10 illustrate this interleaved, message blocking transmission technique for both directions.

3.2.1.3 Message Format Validation and Accounting. The message format validation function of TCS consists of software checks to assure the following:

- An LDP exists
- Proper header and trailer data envelope the message block

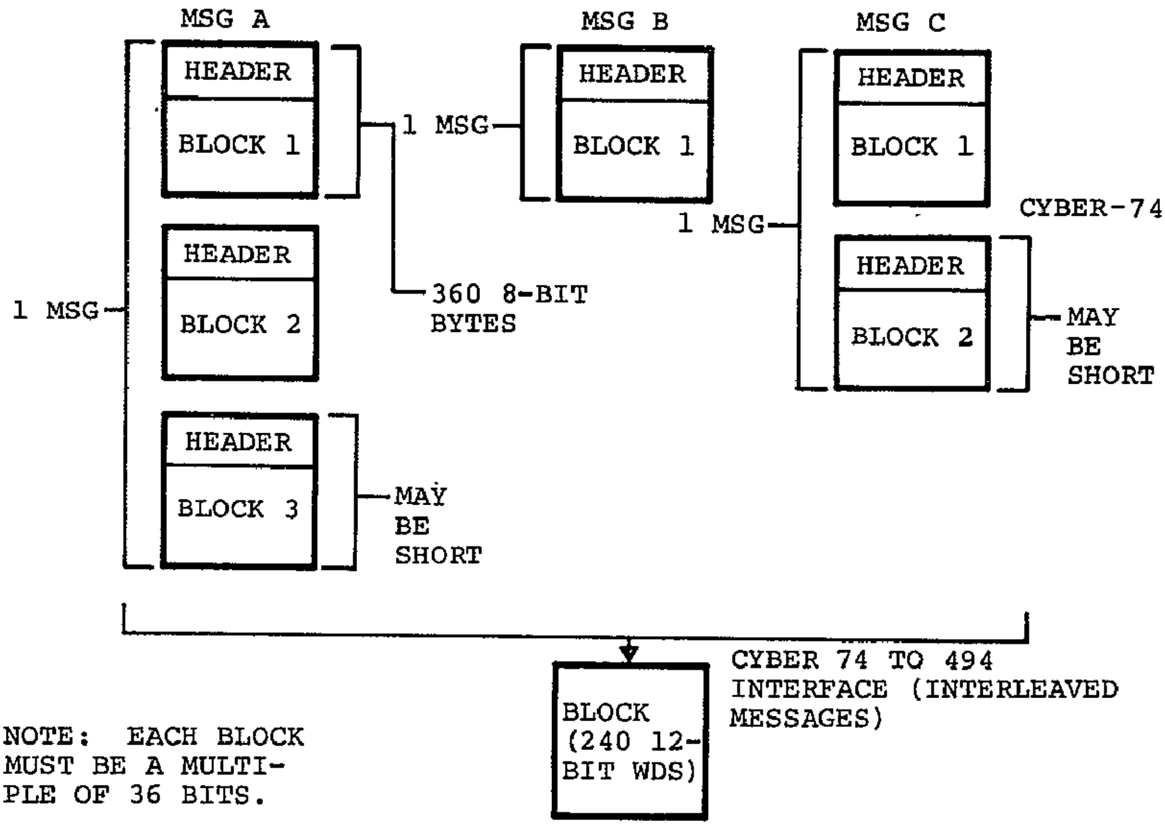
Failure of either of these conditions will cause TCS to issue a diagnostic to the sender.



THIS CONTROL SEQUENCE IS REQUIRED FOR ALL OUTPUT MESSAGE BLOCKS:

1. NORMAL DATA SEND (FIRST MESSAGE BLOCK)
   SOURCE = APPLICATION ID
   DESTINATION = TERMINAL ID
2. DATA MESSAGE BLOCK TO TERMINAL
   (ONE 351-CHARACTER BLOCK)
3. DEMAND REQUEST
   SOURCE = APPLICATION ID
   DESTINATION = TERMINAL ID
4. NORMAL DATA SEND (SECOND MESSAGE BLOCK)
   SOURCE = APPLICATION ID
   DESTINATION = TERMINAL ID

Figure 3-8. — Demand/response data flow host computer to terminal.

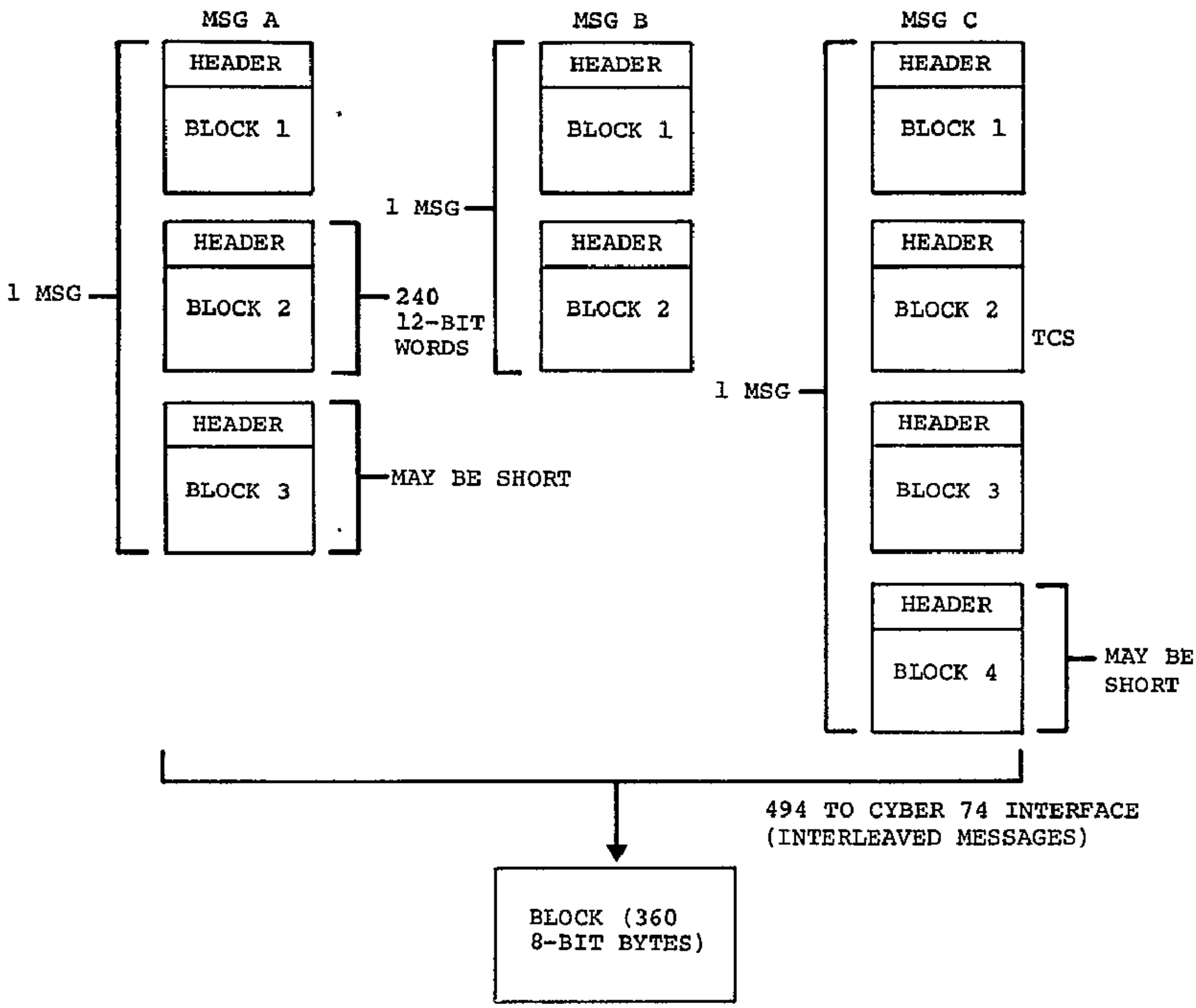


NOTE: EACH BLOCK MUST BE A MULTIPLE OF 36 BITS.

TYPICAL CHRONOLOGICAL SEQUENCE

TCS IN DEMAND MODE FOR ALL 3 LDP'S
CYBER SENDS MESSAGE C BLOCK 1
CYBER SENDS MESSAGE B BLOCK 1
TCS SENDS DEMAND FOR MSG B
CYBER SENDS MESSAGE A BLOCK 1
TCS SENDS DEMAND FOR MSG C
CYBER SENDS MESSAGE C BLOCK 2
TCS SENDS DEMAND FOR MSG C
TCS SENDS DEMAND FOR MSG A
CYBER SENDS MESSAGE A BLOCK 2
TCS SENDS DEMAND FOR MSG A
CYBER SENDS MESSAGE A BLOCK 3
TCS SENDS DEMAND FOR MSG A
TCS IN DEMAND MODE FOR ALL 3 LDP'S

Figure 3-9. – CYBER-to-TCS interleaved interface.



NOTE: EACH BLOCK MUST BE A MULTIPLE OF 36 BITS.

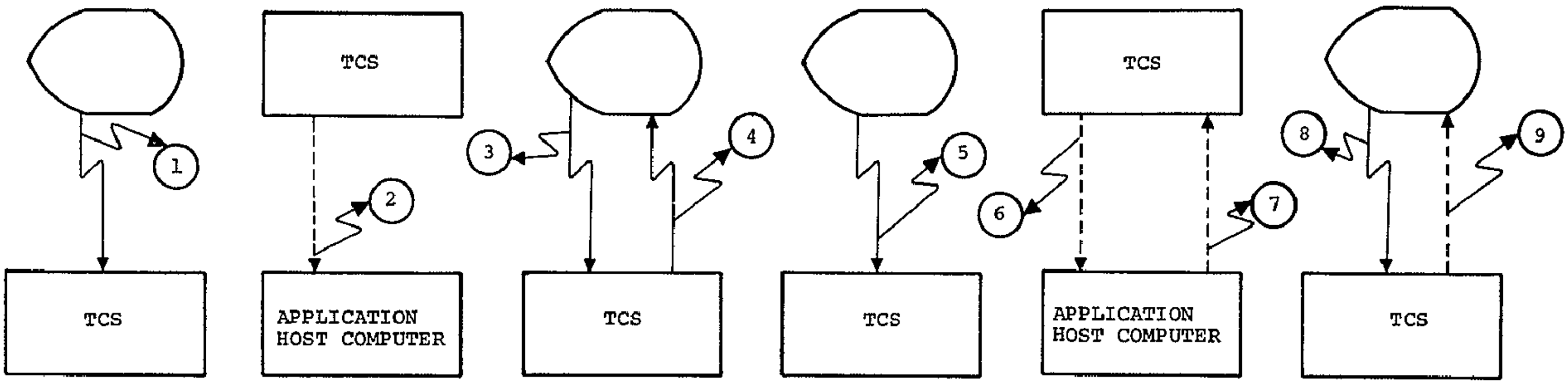
TYPICAL CHRONOLOGICAL SEQUENCE

TCS SENDS MESSAGE C BLOCK 1
TCS SENDS MESSAGE C BLOCK 2
TCS SENDS MESSAGE A BLOCK 1
TCS SENDS MESSAGE B BLOCK 1
TCS SENDS MESSAGE C BLOCK 3
TCS SENDS MESSAGE C BLOCK 4
TCS SENDS MESSAGE B BLOCK 2
TCS SENDS MESSAGE A BLOCK 2
TCS SENDS MESSAGE A BLOCK 3

Figure 3-10. – TCS-to-CYBER interleaved interface.

Message accounting within TCS provides software trapping of messages to a logging file before forwarding this data to its designated destination. The message logging function is option driven.

3.2.1.4 Circuit Assurance. This function permits the terminal user to validate the integrity of his communications line connections to TCS. Figure 3-11 illustrates the TCS circuit assurance data flow.



1 $ START LOOP TEST

2 IF AN LDP EXISTS: LDP BUSY MESSAGE

3 INPUT DATA } MULTIPLE TRANSMISSIONS MAY OCCUR

4 ECHO DATA }

5 $ END LOOP TEST

6 IF AN LDP EXISTS: LDP RESUME

7 IF OUTPUT PENDING, TRANSMIT FIRST BLOCK

8 $ LDP SELECT (APPLICATION)/$SON, APP. ID (SEE SECTION 4.0)

9 IF OUTPUT PENDING, TRANSMIT FIRST BLOCK

Figure 3-11. — TCS circuit assurance sequence of events.

3.3 CYBER 74 SUBSYSTEM FUNCTIONAL DESIGN

The CYBER 74 Subsystem supports the following generic functions:

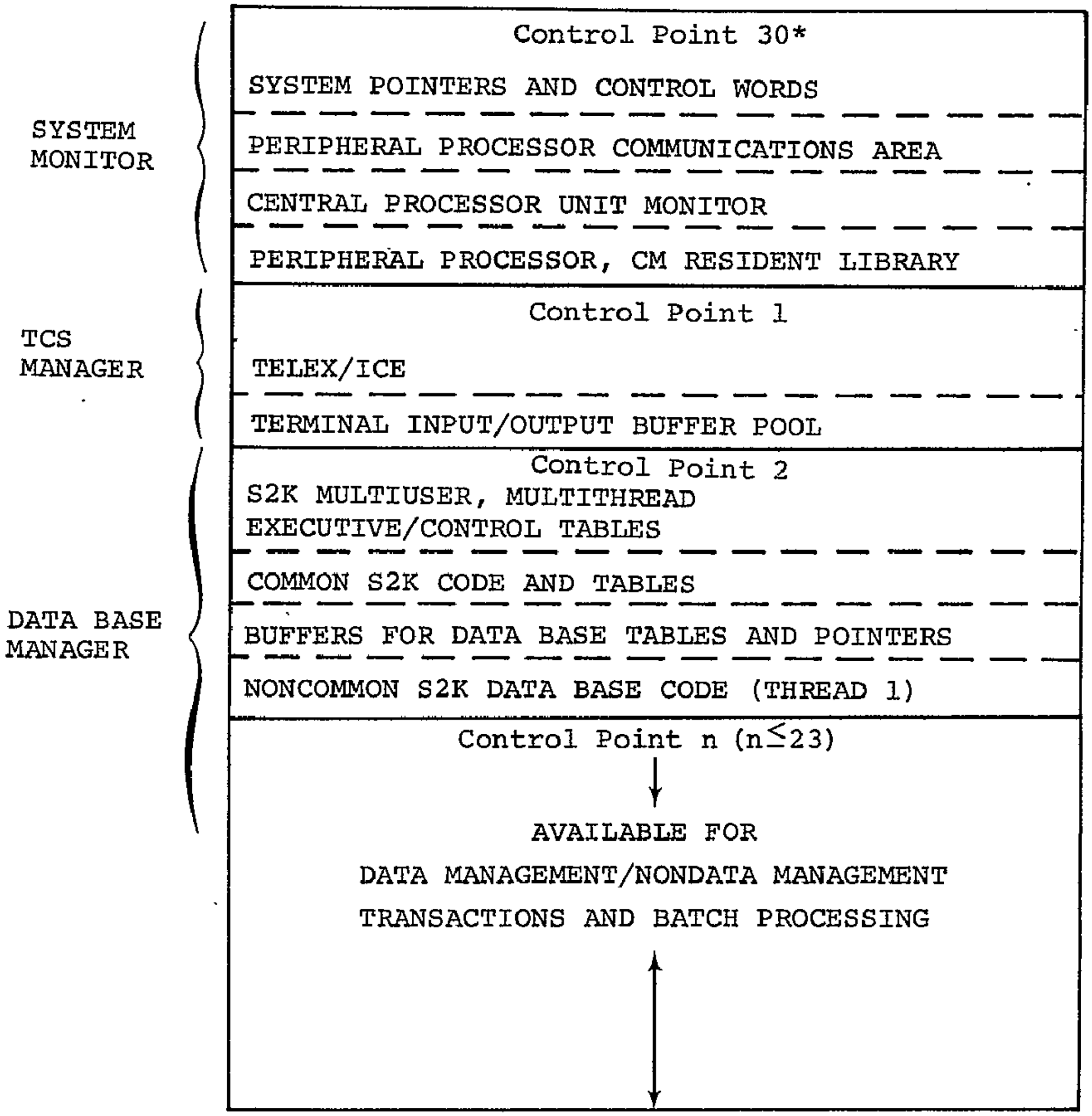
- Low speed/high speed terminals
- Resource shared program processing
- Update/retrieval data base access
- Nonterminal oriented applications processing

3.3.1 CYBER 74 Core Memory

Figure 3-12 illustrates the CYBER 74 central memory (CM) configuration during TCS support. Figure 3-13 illustrates the CYBER 74 CM configuration during non-TCS support. Control point (CP) designates the location in CM for a program at a given point in time. Additional information about the CP concept may be found in section 3.3.2.1 of this document and section 2.3.2.1 of the System Requirements Document for the Shuttle Program Information Management System.

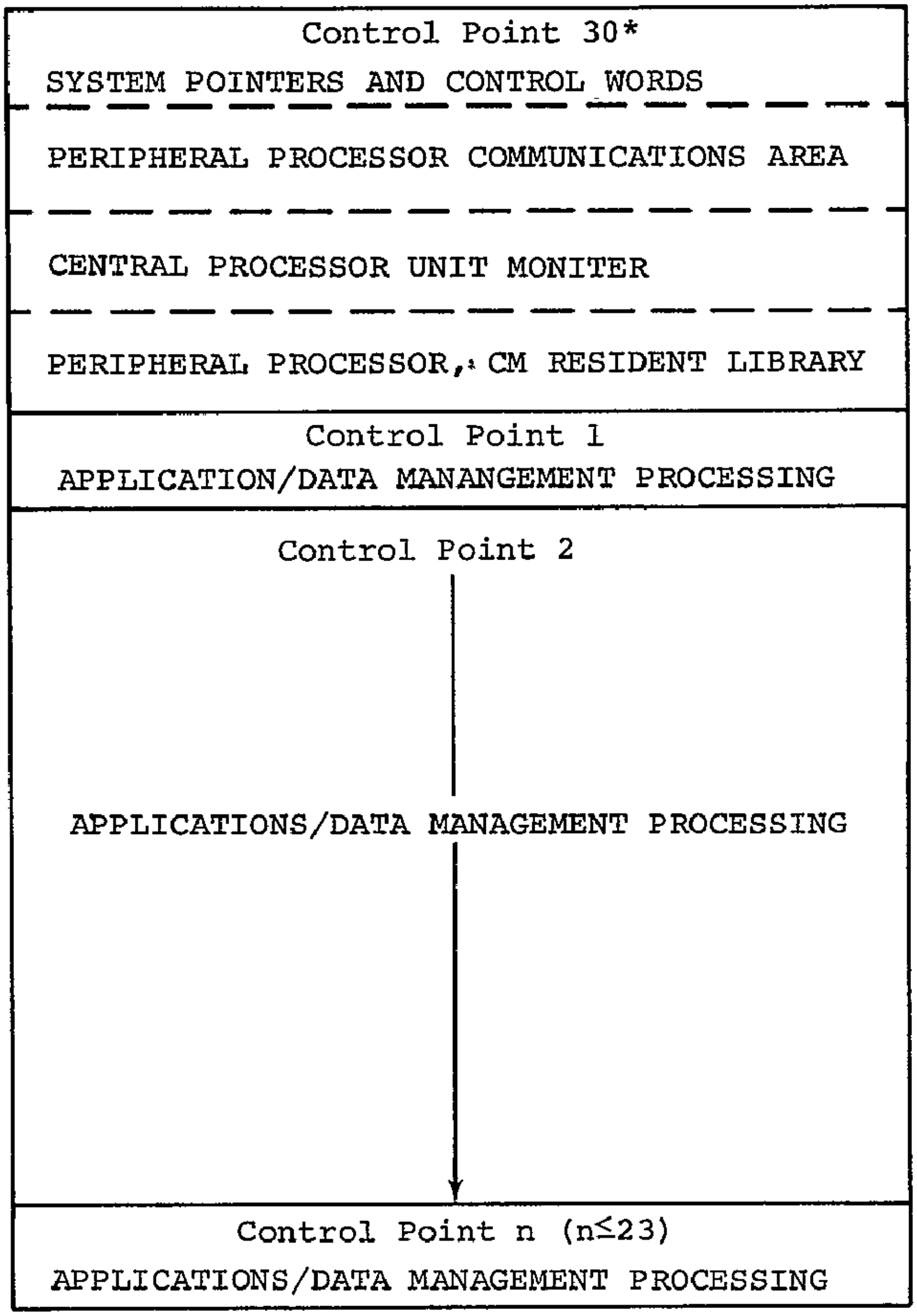
3.3.2 KRONOS Time-Sharing System Functional Design

KRONOS is a complex operating system containing many features. KRONOS is designed to provide support for a mixed processing environment. The primary purpose of this section is to describe the functional design for job processing. A job's processing class in this environment is defined by the origin of the job.



*A PSEUDO CONTROL POINT

Figure 3-12. – CYBER 74 CM configuration (KRONOS-TCS support).



*A PSEUDO CONTROL POINT

Figure 3-13. – CYBER 74 CM configuration (KRONOS-non-terminal support).

SPIMS has the following job origins:

- Time-sharing (interactive)
- Local Batch
- System console

Figure 3-14 illustrates the generic input/output (I/O) paths for the three job origins. Efficient priority processing of job origins is the primary objective of KRONOS. This section describes the basic software elements, their functions, and how these software elements working together accomplish the prime objective. Additional information is available in the Control Data KRONOS 2.1 Workshop Reference Manual, Control Data Corp., 97404700B, May 15, 1974, and the SPIMS/CYBER Interface Control Document, Control Data Corp., 90CLKA00021-1, January 31, 1975.

3.3.2.1 KRONOS definitions. Several hardware/software terms are basic to the functional design of KRONOS. An explanation of these terms and their capabilities is required to clarify the functional design of the KRONOS Time-Sharing System.

3.3.2.1.1 Job origin type/queue priority (JOT/QP): Each job entering the KRONOS controlled environment is assigned a job origin type and a queue priority. The SPIMS job origin types are:

Type	Origin
SYOT	System console - all jobs initiated via the system console.

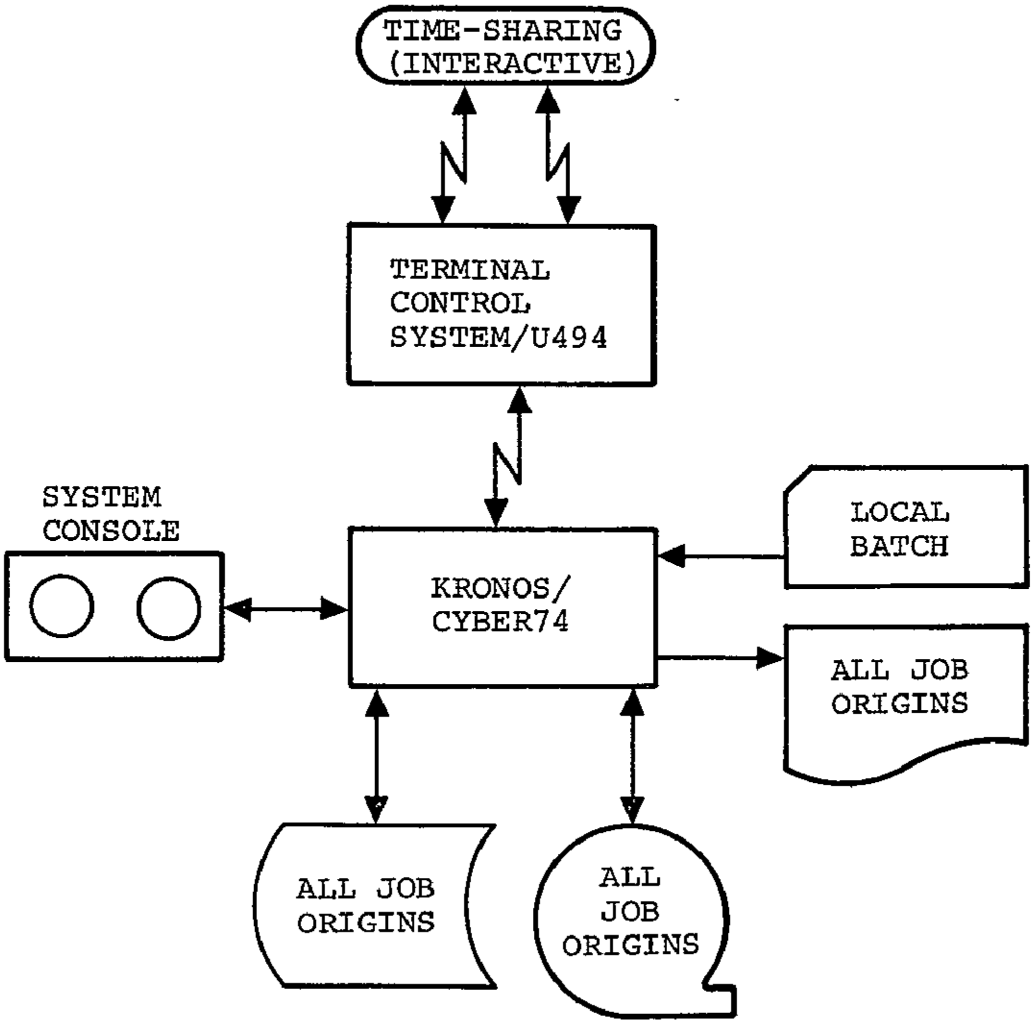


Figure 3-14. – Job origin input/output paths.

BCOT Local batch - jobs entered from all local batch devices, or initiated by the terminal command SUBMIT.

TXOT TELEX/ICE - jobs entered via the time-sharing executive program TELEX/ICE.

MTOT A job doing one specific task for many terminals. These jobs provide global pre-processing of time sharing jobs in a collective manner.

Scheduling parameters for jobs are established and aged based on the job origin type and queue priority.

Basically the job origin type sets the priority for a job request for central memory control points; queue priority sets the priority for servicing a control point program request for the central processor. These parameters are installation options. The standard KRONOS priority order for SPIMS job origin types/queue priorities is:

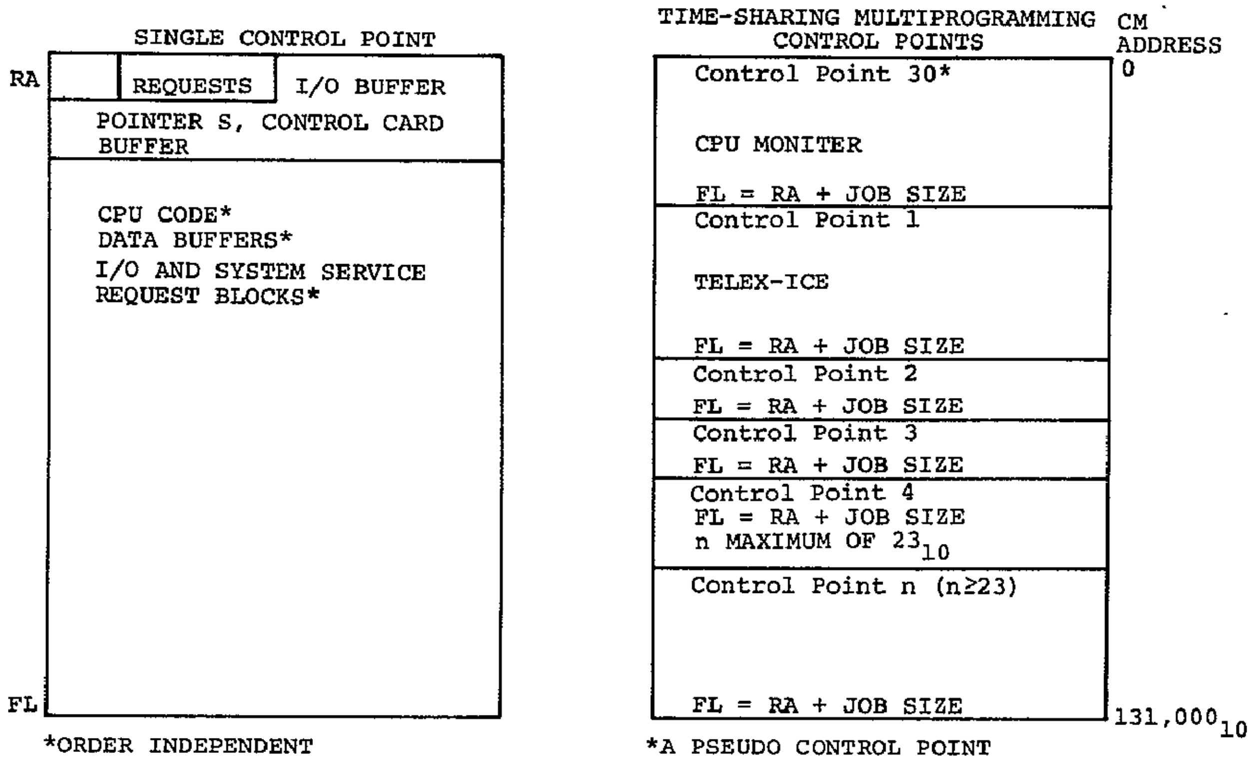
Job Origin Type	Generic (QP)
MTOT	First
SYOT	Second
TXOT	Third
BCOT	Fourth

3.3.2.1.2 Control point: A control point is that area of central memory allocated to a job while the job is executing, waiting I/O completion, or requiring the CPU. This area of central memory is bounded by a reference address (RA) within central memory and a block length or field length (FL). Also associated with each control point is a 200 word area of memory. This block is called the Control Point Area (CPA). The CPA contains accounting data, a buffer for the registers, priority, a control card buffer and user validated capabilities. This area is rolled out with the control point.

The FL for a job is allocated in consecutive 100 octal-word blocks. See Figure 3-15. Control points are assigned independent of an applications previous control point assignment. Subsystems, however, reside at designated control points and control their own subcontrol points.

A control point contains a job's central processing instructions to operate on data and the peripheral device requests required to read and output data. The first 77 octal-words of a control point are used for communications to three ppu programs, I/O buffer pointers, and the current control card to be executed.

Because the CPU cannot read or write to peripheral devices, no central memory is required for perhiperal device drivers.



NOTE: See figure 3-12 for SPIMS core map.

Figure 3-15. – Control points.

Requests for I/O and other system services are placed in the second word of a control point by the application. These requests are called RA+1 requests. These requests are recognized by either of the KRONOS monitors, Central Processor Monitor (CPUMTR) or peripheral processor (PP) Monitor (MTR) and the scheduler, 1SJ.

The allocation, servicing, and management of control points by the KRONOS monitors provide a time-sharing, multiprogramming environment. Section 3.3.2.2 describes the basic functions these monitors perform.

3.3.2.1.3 KRONOS Subsystems: Subsystems are considered special jobs with many privileges not granted to regular jobs within the system. Some of these privileges are:

- Subsystems cannot be rolled out.
- Subsystems can make use of the intercontrol point communications request (SIC/RSB) to receive and send data buffers. The SIC will be used to receive data in the System 2000 Multiuser Multithread Executive.
- Subsystems may divide their control point into subcontrol points.
- Subsystems may be assigned a central processor unit (CPU) request priority above normal control point jobs.
- Subsystems need not be restricted by the standard job control parameters defined to the KRONOS CPU monitor.

- Subsystems execute at a predefined control point. Thus, at load time any job at that control point is moved.

Job advancement, scheduling, and detection of subsystem requests by KRONOS is different than for a normal job (see section 3.3.3). The scheduling routine 1SJ must trap a subsystem in order to ensure its requested control point immediately and assign its unique CPU priority. This priority is normally very high, thus, caution should be used in all subsystems to avoid dominant use of the CPU.

3.3.2.1.4 Subcontrol Points: as the name implies, are divisions of a Central Memory Control Point (see figure 3-12, control point 2): A control point can be established to contain two or more programs; one of these is designated as the "Executive" (subsystem) which monitors the other programs known as subcontrol points.

The Executive controls its subcontrol points in much the same manner that the system monitors manage the control points. When a control point makes a system request, exceeds its time limit, or makes an error, control is given back to the system monitor. Similarly, when a subcontrol point makes a system request, exceeds its time limit, or makes a CPU error, control is given back to the Executive. The Executive sets up each subcontrol point so that, within the field length of the control point, each subcontrol point has its own "RA" and "FL" and cannot go outside its boundaries. The Executive is thus protected from access by the subcontrol points, whereas the Executive's RA and FL define the full control point so the Executive can watch over and control all subcontrol points within the field length.

The subcontrol point concept depends entirely on the Executive program's handling of the subcontrol points. This involves starting, stopping, error processing, and other functions similar to those of the system monitor.

Just as the system monitor keeps track of each control point through the control points exchange package, the Executive can control the subcontrol points through the subcontrol points exchange packages. The Executive is responsible for setting up an exchange package for each subcontrol point; each exchange package must have the appropriate RA, FL, program address, register settings, etc. for the subcontrol point. These exchange packages must be set up somewhere within the Executive's field length but probably not within the field length of the subcontrol point. To start execution of a subcontrol point, the Executive uses a CPU-to-monitor (XJP) request which indicates the address of the exchange package area of the subcontrol point to be activated. When CPUMTR picks up the request, it terminates the Executive and activates the subcontrol point described in the exchange package area indicated in the XJP request. CPUMTR also sets a flag in the control point area showing that at this control point a subcontrol point is now active.

Once activated, a subcontrol point runs until (1) it makes a CPU error, (2) it exceeds its time limit, or (3) it makes an RA+1 request. Under any of these three conditions, control is given back to the Executive.

The Executive can thus monitor processing for the subcontrol points. Errors can be noted and examined without termination of the control point. Upon returning control to the Executive, certain information is set up in the X registers of the CPU:

- (X2) - msec before this subcontrol point began
- (X6) - error flag and RA of this subcontrol point
- (X7) - msec used by this subcontrol point

One of the parameters on the XJP request is the time limit for the subcontrol point. When this time limit is passed, control goes back to the Executive.

When a subcontrol point makes an RA+1 request, control is returned to the Executive; the Executive can then decide whether to (1) ignore the request, (2) handle the request itself, or (3) pass the request on to CPUMTR using RA+1 of the Control Point Executive.

3.3.2.1.5 User Load Leveling (rollin, rollout): A time-sharing, multiprogramming environment assumes the sharing of a set of resources in an equitable manner among the various users. KRONOS provides this capability using a combination of rollout files, queues, priorities, and time slicing.

All information pertaining to a job's accounting, resource requirements, current CPU register contents, and program code is dynamically maintained in a rollout file whenever a job swap occurs. A rollout file is allocated and released on an as-need basis, one for each user of the system. Figure 3-16 illustrates the format of a rollout file.

The queues maintained and checked by the KRONOS monitors are not unique areas of core. The File Name Table (FNT) provides a dual role. When a central memory program first enters the system, the FNT indicates a new job is waiting and its initial priority. When the program is executing, the FNT indicates the files required, their logical names, and their types; i.e., input, output, local.

When a control point program is rolled out, the FNT indicates that a job is in the system and what its priority is. Thus the FNT provides a queue service and a resource allocation service.

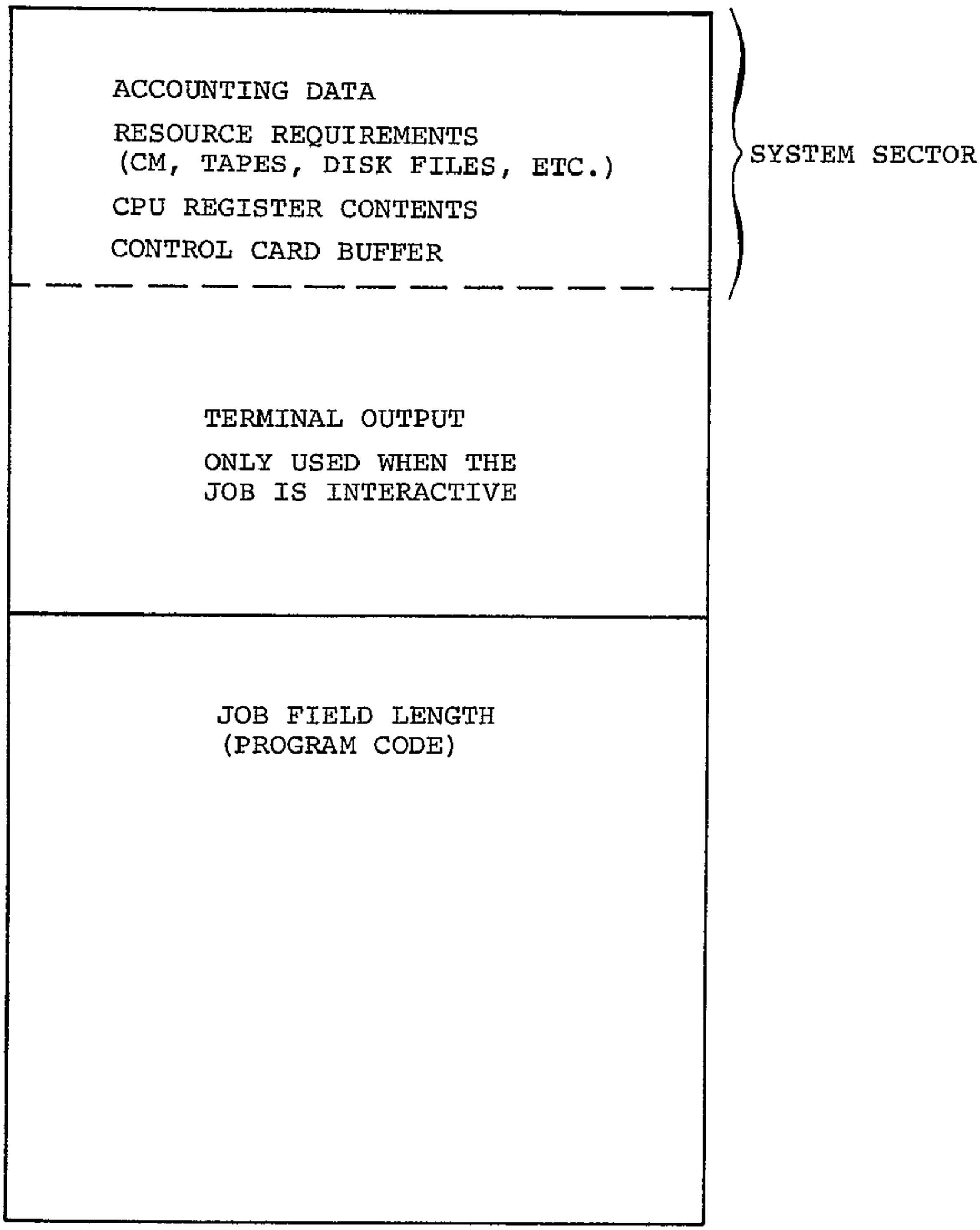


Figure 3-16. — Rollout File Format.

When a queue is being checked, a search of the FNT occurs to find those entries that have the file name and type being sought but not assigned to a control point. Changing the file name or type in the FNT changes the queue. Figure 3-17 illustrates this process.

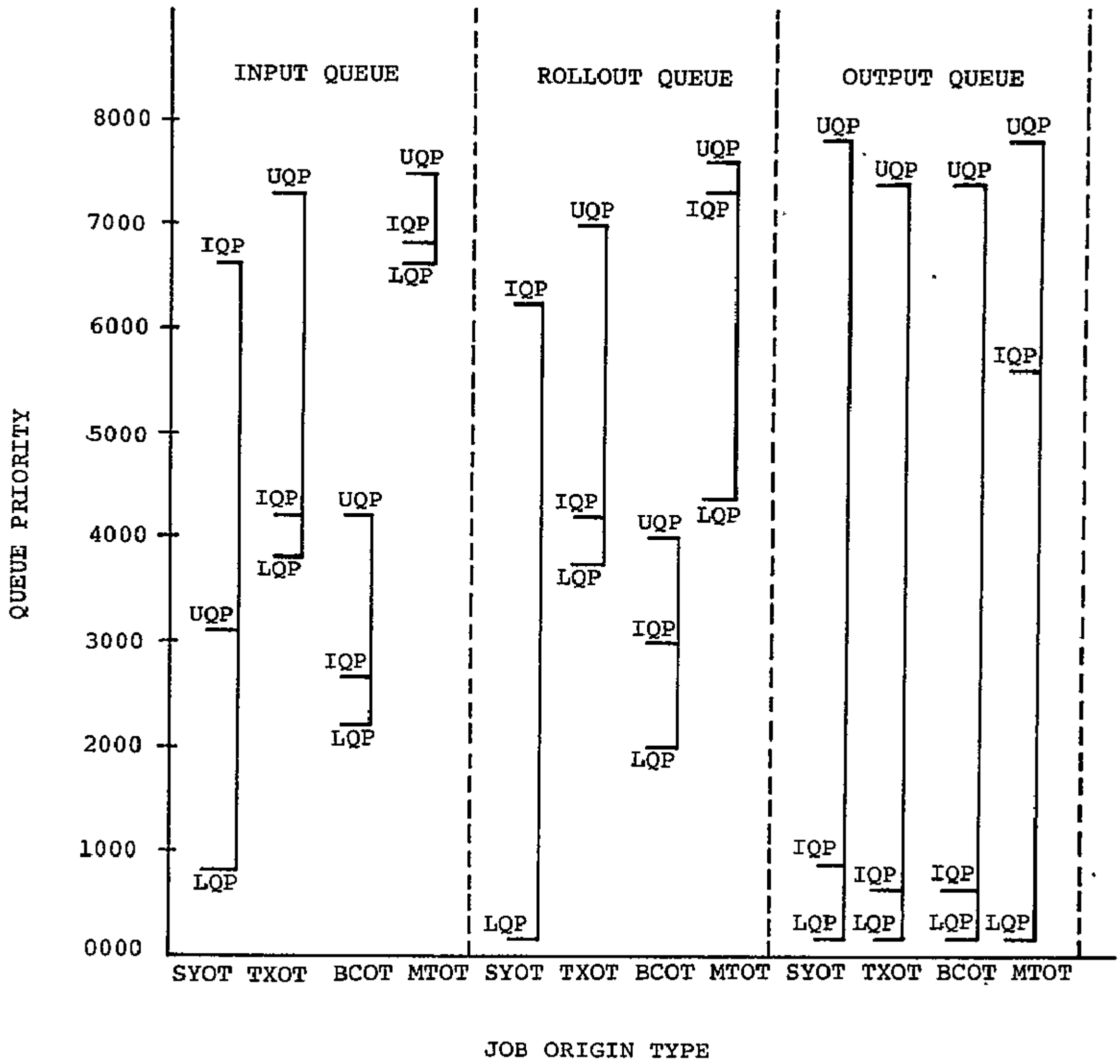
Priorities are assigned based on job origin type. Queues are searched by priority. The priorities assigned to a job origin type are checked by the job scheduler against a predefined set of time slices and resource usage parameters. The aging of one or more of a job's parameters is independent of the job's activity. Age limits will vary according to the job's origin type. Jobs rolled out are aged concurrently with jobs at control points. Figure 3-18 illustrates a typical queue priority scheme.

3.3.2.1.6 Recall: The recall program status is provided by KRONOS to enable efficient use of the central processor and to capitalize on the multiprogramming capability of KRONOS. Whenever a control point program must wait for the completion of an I/O operation before more computations are performed, the control point program asks KRONOS to place the control point into recall. Recall may be automatic or periodic.

Automatic recall can be used whenever a program requests I/O or other system services and cannot proceed until the request is complete. The CPU is assigned to execute a program at some other control

FNT		ENTRY	JOB STATUS
JOBNAME	I	0	JOB IN INPUT QUEUE (FILE TYPE)
INPUT	I	CP NO.	JOB AT CONTROL POINT (FILE INPUT REQUIRED) ADDITIONAL FNT ENTRY MAY BE GENERATED BY THE JOB. THESE ENTRIES ARE WRITTEN IN THE SYSTEM SECTION OF THE ROLLOUT FILE.
JOBNAME	R	0	JOB IN ROLLOUT QUEUE
INPUT	R	CP NO.	JOB IS ROLLED IN (FROM ROLLOUT FILE)
JOBNAME	Ø	0	JOB IS COMPLETE (OUTPUT FILES READY)

Figure 3-17. — File Name Table (FNT) usage.



LEGEND:

SYOT — SYSTEM CONSOLE
TXOT — TIME-SHARING
BCOT — LOCAL BATCH
MTOT — MULTI-TERMINAL
UQP — UPPER QUEUE PRIORITY SETTING
IQP — INITIAL QUEUE PRIORITY SETTING
LQP — LOWEST QUEUE PRIORITY SETTING

Figure 3-18. — Typical queue priority schedule.

point. The KRONOS CPUMTR will not assign the CPU to a control point in automatic recall until the program request is complete.

Periodic recall can be used whenever the control point program is waiting for any one of several I/O requests to be complete. The program will be activated periodically, so it can determine which I/O request has been satisfied and whether or not the control point program can proceed. Concurrent, multiple data transfers from a single control point program are provided by periodic recall.

3.3.2.2 <u>KRONOS monitors</u>. There are two monitors. One monitor is loaded in Peripheral Processor Zero and is called PP-Monitor (MTR). A second monitor is CM resident at pseudo Control Point Thirty and is called CPU-Monitor (CPUMTR).

MTR performs the following basic functions:

- Maintains the Real Time Clock
- Processes nonpriority PP requests
- Allocates central memory
- Checks the CPU for arithmetic errors
- Checks RA+1 of active central point programs for system requests
- Checks the status of active control points and initiates 1AJ (job advancement) if zero status or rollout status occurs

* Checks the output register of each pool PP

CPUMTR performs the following basic functions:

* Processes priority PP requests
* Assigns the CPU to control points
* Initiates 1SJ (job scheduler)
* Checks RA+1 of active control point programs for system requests
* Performs direct Extended Core Storage (ECS) transfers
* Reserves and clears mass storage allocation tables
* Performs central memory storage moves to provide contiguous unused core

These monitors, working in concert, control the scheduling of all PPU programs required for job processing.

3.3.3 KRONOS Job Processing Functional Design

All jobs which flow through the system, regardless of job origin type, will be processed from initiation to completion by each of the following PP programs:

Name	Function
1SJ	Job scheduling
1AJ	Job advancement - control card-to-control card
1SP	Job priority and queue ageing
1RO	Job rollout
1RI	Job rollin

1CJ Job termination (TELEX-ICE provides this function for interactive jobs)

Figure 3-19 illustrates a general system flow for a job entering the system.

Jobs enter the system at the initial queue priority (IQP) for their type (fig. 3-20). They are aged by 1SP as they reside in the input queue; i.e., the queue priority is increased until it reaches the upper queue priority (UQP). Figure 3-18 illustrates a typical queue priority scheme. At any time, the scheduler, 1SJ, may determine that this job is the best candidate (best job) for a control point, and the scheduler attempts to schedule or assign it to a control point (fig. 3-21). This (best job) decision is made by an algorithm that takes into account queue priority and resources desired (FL, etc.).

In order to schedule or assign a control point, 1SJ determines if there is enough unused noncontiguous core available to satisfy the field length requirements for the job. If not, 1SJ will determine if there will be enough unused core after scheduled rollouts of other jobs. If not, 1SJ will attempt to schedule any other jobs with lower priority than the best job.

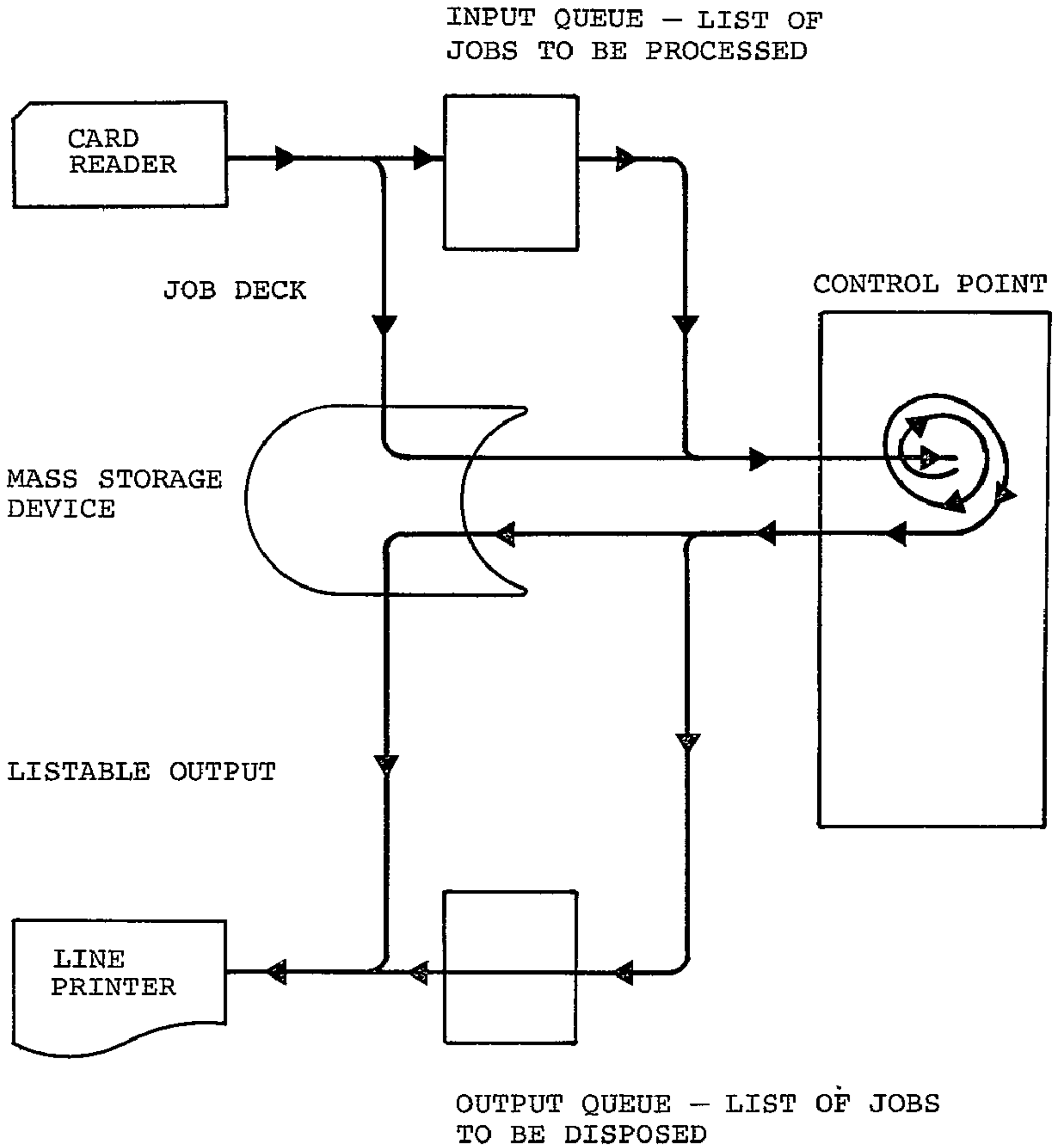
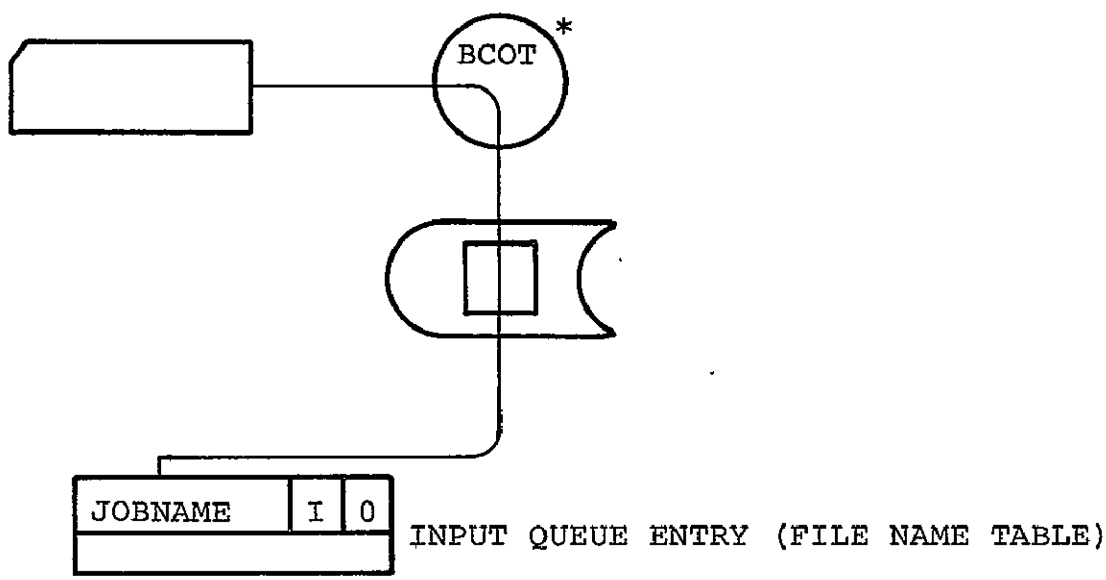


Figure 3-19. — General system flow.



* TXOT/MTOT ARE STARTED
BY TELEX-ICE AND SYOT
IS INITIATED BY THE
SYSTEM CONSOLE

Figure 3-20. — Read card reader.

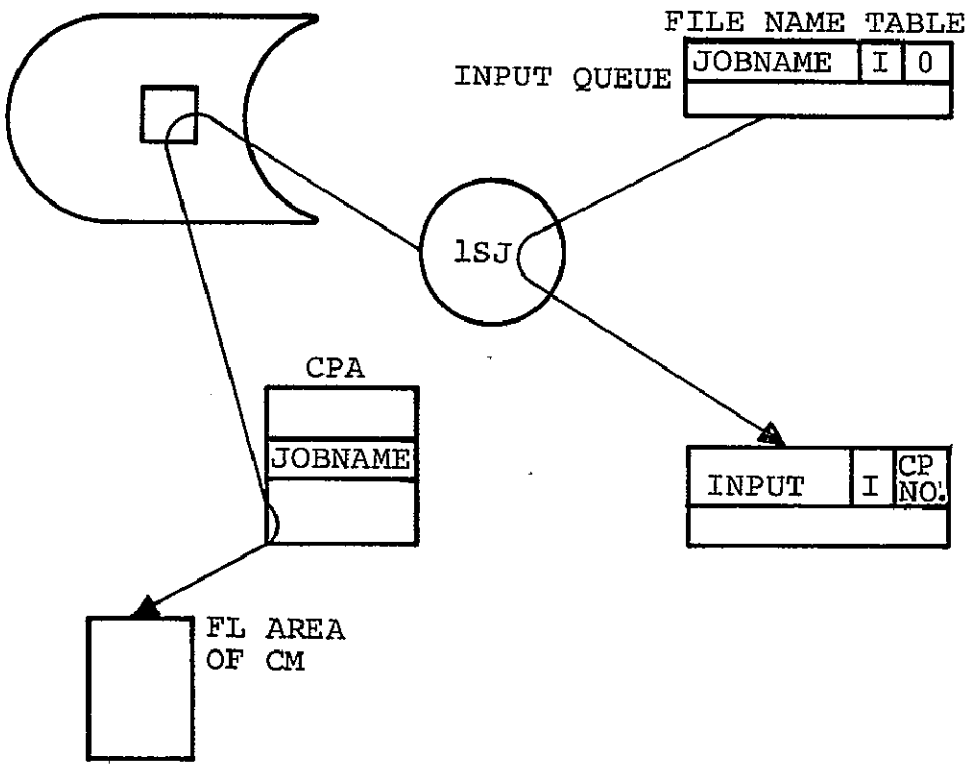


Figure 3-21. – 1SJ prepares a CP for the job.

Assuming enough FL, 1SJ looks for an available control point. If there is no control point, rollout will be scheduled for any job whose priority is lower than the best job. If there are no lower priority jobs, 1SJ drops out until rescheduled by CPUMTR.

When 1SJ assigns the best job to a control point, it will get the FL and set up the control point area (CPA) with information from the predefined user resource validation tables. (These tables contain the resource maximums for a user.) In addition, 1SJ will set the input queue priority to UQP regardless of what its value was when this job was picked. 1SJ will leave the job without an operation status and call 1AJ (fig. 3-22).

The job advancement routine, 1AJ, will note that the job status is empty; i.e., the last operation is completed (in this case, first operation is not started). 1AJ will start this job. The job can at anytime create local files. If the file name is OUTPUT PUNCH PUNCHB or P8, the file will be given special treatment at job completion time (fig. 3-23).

As the job progresses, one of three statuses is defined: wait for CPU "W", recall "X", or auto-recall "R".

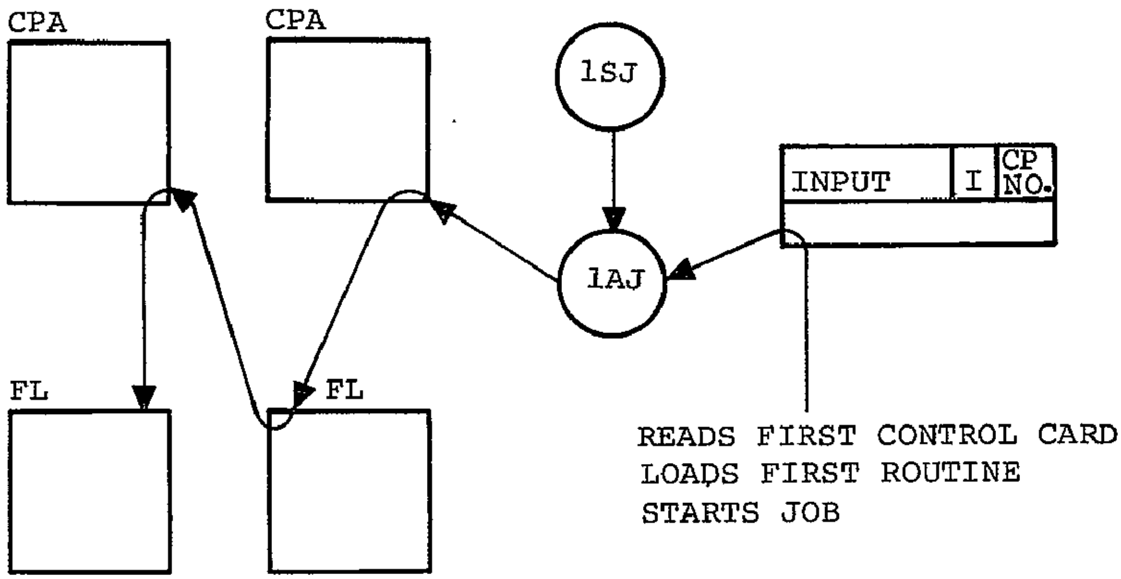


Figure 3-22. - 1AJ starts the job.

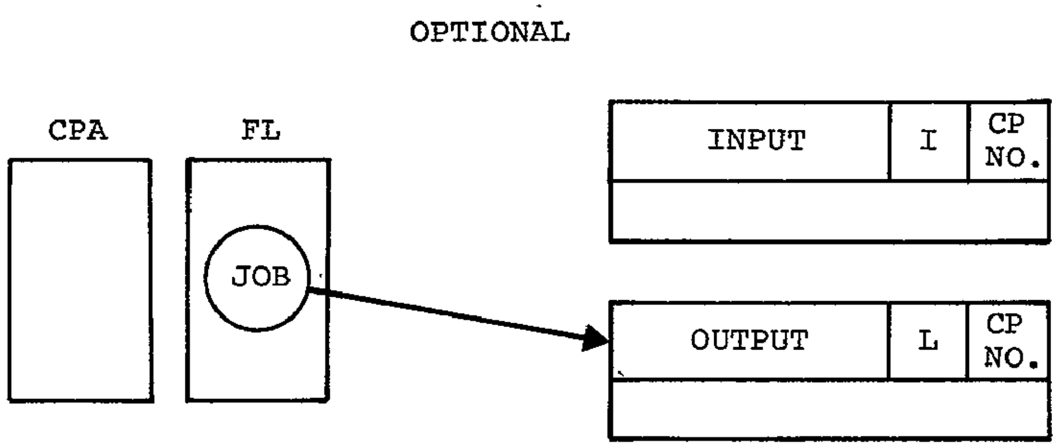


Figure 3-23. — Job creates local file name OUTPUT, PUNCH, PUNCHB, or P8 as denoted by OUTPUT.

CPUMTR and MTR will periodically check all the jobs running at control points and, if either monitor detects "W," "X," and "R" status zero, they will call 1AJ. If the error flag is set, 1AJ will process the error. If the error is not fatal, 1AJ will advance to the next control card. If the error is fatal but an EXIT card exists, then 1AJ will advance to the control card following the EXIT card.

CPUMTR and MTR also monitor the CPU time-slice, and if the job exceeds its time-slice, its queue priority is dropped to the lowest queue priority (LQP) of that type. This does not mean that the job will lose its control point. If 1SJ finds a best job in the input or rollout queues, then low priority jobs are candidates for rollout. Also 1SJ examines all the control points, and if it detects that the CPU time-slice is exceeded before either monitor detects this, 1SJ will lower the queue priority to LQP. An interlock is provided so the queue priority is only dropped once.

1RO may be called by 1AJ, 1SJ, from the system console, or by a subsystem (fig. 3-24). 1RO will dump the job according to the rollout file format (fig. 3-16); will set "W" "X" or "R" status to zero; will request the control point be made available; and will release all FL, and all File Name Table (FNT) entries assigned to this control point. 1RO will release everything else except the input and control card file and will call 1AJ to advance the job. In this way, FNT space is not wasted while a job is rolled out. The job is then placed into the rollout queue with whatever queue priority the job had when it was rolled out, or the rollout queue priority if two or more of the aging parameters have expired. If 1RO is called by a subsystem (e.g., S2KMM Executive), the rollout file will be called DM* and left assigned to this control point.

1RI will read the rollout file and reestablish all the files, equipment, etc. to allow the job to continue (fig. 3-25). It will set "W" "X" and "R" statuses to their former values. The control point will now be a candidate for the CPU.

When 1AJ detects an end-of-job card stream, a fatal error with no recovery, an illegal control card, or some other fatal condition, it calls 1CJ to complete the job. If any of the job flow routines ever detect a type which is not defined (i.e., type not SYOT BCOT TXOT or MTOT), it will call 1CJ immediately to end the job. This is protective coding.

1CJ will locate OUTPUT, the local file assigned to this job, if it exists (fig. 3-26). It will then append the job accounting to the end, write an end-of-information, and move the file to the output queue. This file is renamed with the job name for local printing.

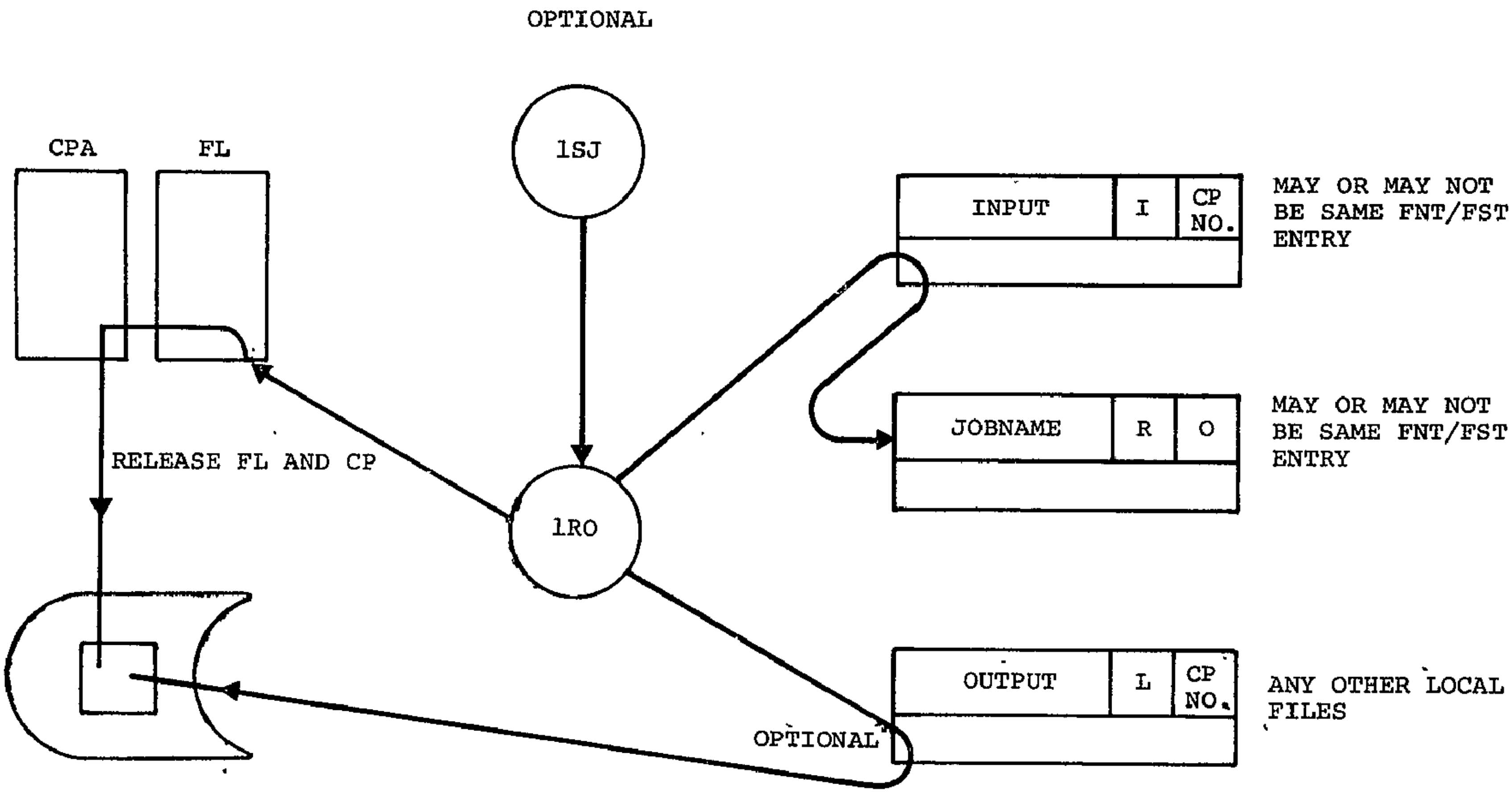


Figure 3-24. — Job is rolled out.

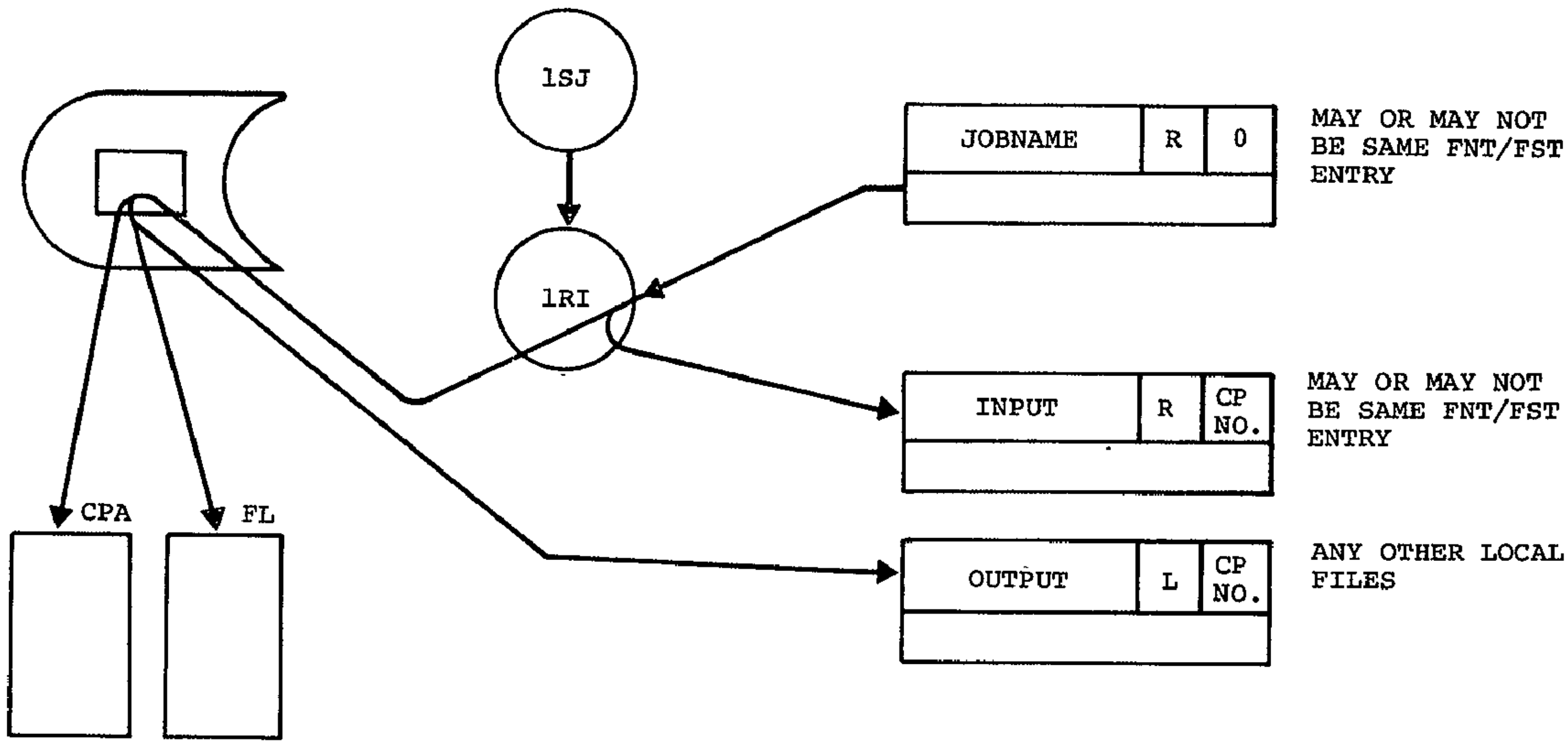


Figure 3-25. — Job is Rolled In (From Rollout).

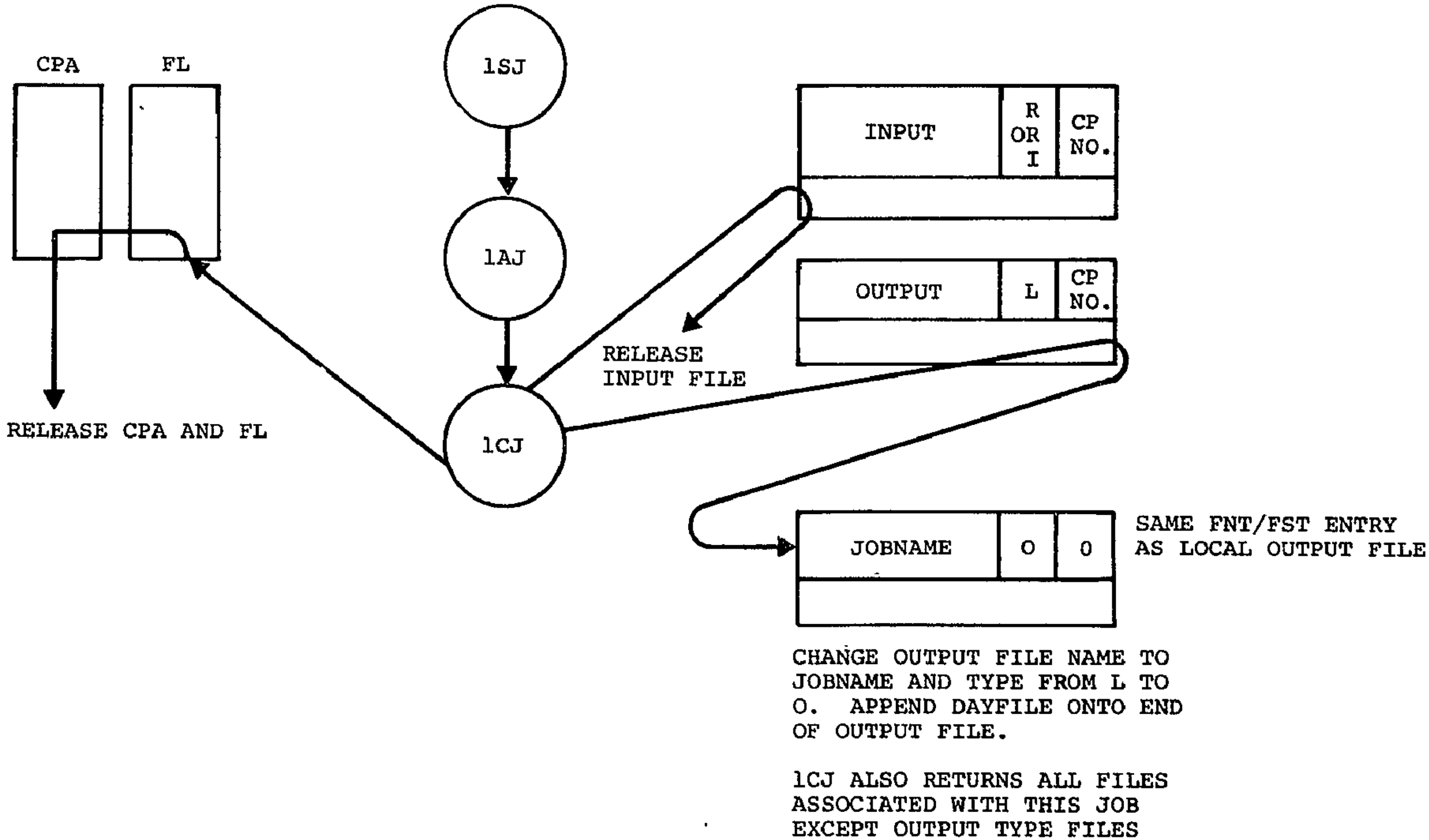


Figure 3-26. — Job completes.

3.3.4 Terminal Support Software (TELEX-ICE) Functional Design

The SPIMS CYBER 74 Terminal Support Software (TELEX-ICE) is a KRONOS subsystem. This subsystem provides the interface and control of interactions between KRONOS and the user at an interactive terminal.

Two communications hardware/software interfaces are supported by TELEX-ICE. Figure 3-27 illustrates these interfaces.

Two data transmission modes are supported by TELEX-ICE:

- Normal Mode
- Special Binary Mode

Normal Mode data transmissions from the terminal are constrained to be no greater than 150 characters per read request. Normal Mode output to the terminal does not have this constraint. TELEX-ICE provides the controls required to assure correct transmission of all data to the terminal.

Special Binary Mode (SBM) data transmissions are constrained to be no greater than 5400 characters per transmission in either direction. Section 4.0 contains a detailed explanation of the initiation and use of these two transmission modes by applications and terminal users.

3.3.4.1 Terminal Support Software Elements (TELEX-ICE). The TELEX-ICE subsystem consists of a central processor (CP) program and several peripheral processor (PP)

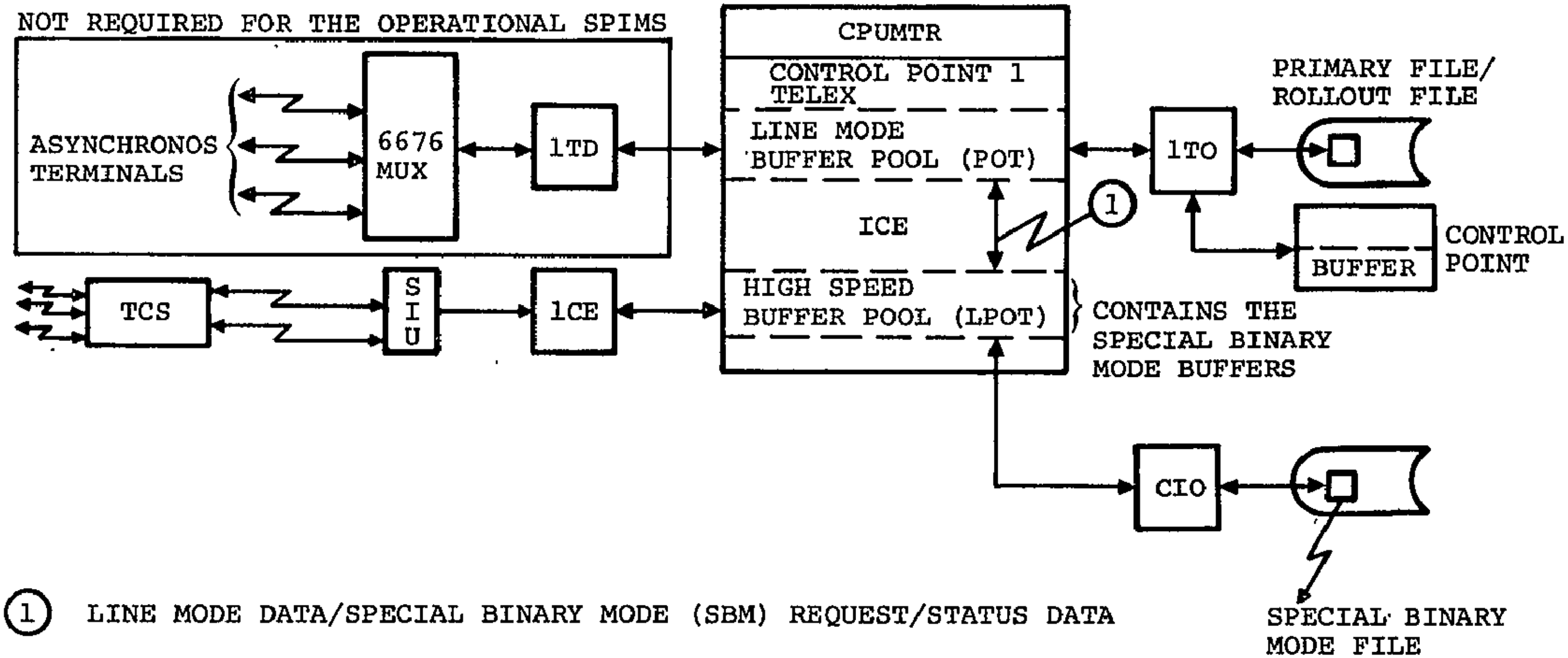


Figure 3-27. — TELEX-ICE communications hardware/software interfaces.

programs. The names and basic functions of these software elements are:

TELEX - Terminal Executive Initialization Routine. This routine is loaded in response to a system console request. TELEX initializes tables and pointers and loads TELEX1.

TELEX1 - Terminal Executive Processor. TELEX1 is the main routine that processes I/O for the terminals using normal mode. TELEX1 cracks and processes commands and makes requests to dump source input to disk and refill output buffers from disk for Normal Mode I/O. TELEX1 in concert with 1TD interfaces directly with the CDC 6676 multiplexer. TELEX1 operates with ICE to support the high-speed interface to TCS.

TELEX2 - Terminal Executive Termination Routine. TELEX2 is executed after detection of an abnormal condition in TELEX or the system console request 1.STOP which terminates the TELEX subsystem.

ICE - ICE is a SPIMS enhancement to TELEX1 and is required to support the high speed communications interface with TCS and Special Binary Mode I/O.

1TA - TELEX Auxiliary Function Processor. This routine processes functions for TELEX which require PP action.

1TD - Terminal Communications Driver. Low speed interactive (600 baud or less). 1TD performs character code conversion and communications between TELEX1 and the CDC 6676 multiplexer.

1CE - Computer-to-computer interface driver. 1CE supports high speed data transfers (81.6 kbps) between the TCS and the CYBER 74.

1TO - Terminal input/output. 1TO is called by TELEX to perform terminal I/O requiring mass storage accesses.

PFM - Permanent File Manager. PFM may be called by TELEX to process Permanent File requests. TELEX processes some PFM calls himself.

CIO - Common input/output. CIO is called by ICE to process Special Binary Mode data.

1UF - ICE calls 1UF to find the File Name Table entries of the Special Binary Mode files for jobs rolled out.

The relationship between the various CP and PP routines is shown in figure 3-28.

3.3.4.2 <u>KRONOS supported terminal input/output (TELEX-ICE)</u>. There are three terminal I/O support functions provided by TELEX-ICE:

- Primary File used to construct source code programs in normal mode.
- Rollout File used to buffer code converted multiline data for an applicatons user.
- Special Binary File used to buffer SBM data for an applications/user.

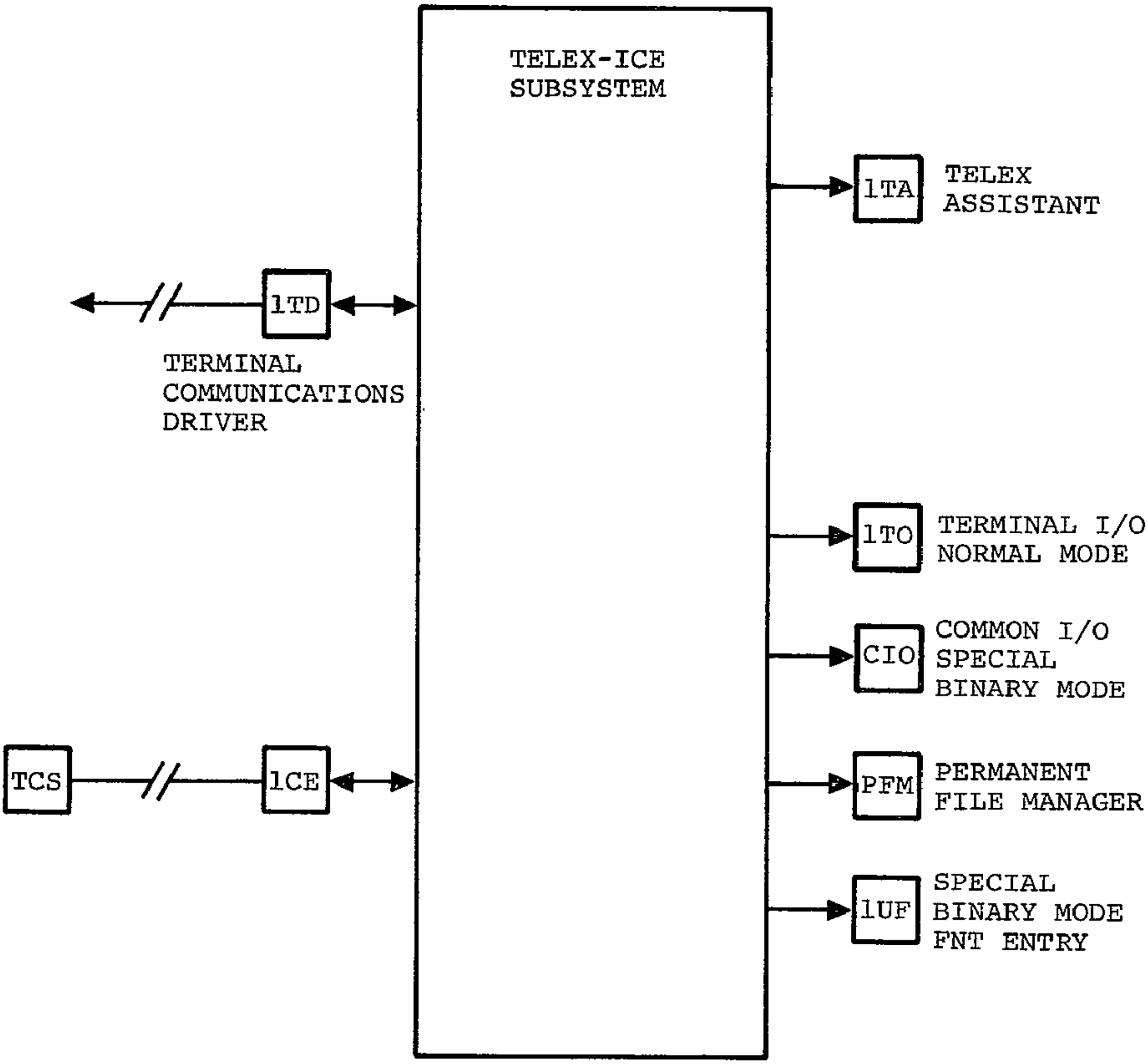


Figure 3-28. — Terminal support software relationships.

Figure 3-27 illustrates Primary File and Rollout File I/O. During interactive normal mode input processing, the terminal data is buffered in POTS by 1TD or LPOTS by ICE. A POT is an eight-word central memory buffer allocated by TELEX. The LPOT is a 48-word central memory buffer allocated by ICE. Each LPOT is edited (deletion of TCS message envelopes) and converted to display code by ICE. A normal mode LPOT contains a maximum of 150 data characters. Characters not in the 63-character display code table are deleted. This data is then established as a POT for TELEX-1TO processing.

During Primary File processing, each POT (terminal line) of data is transferred by 1TO to mass storage, one POT per physical disk address (sector) within the file. During output processing, the Primary File is compressed by packing all statements together. 1TO reads the Primary File from mass storage and stores the data in POTS for 1TD or ICE/1CE output processing.

During Rollout File processing, the input data is transferred directly to a control point program input buffer from the POT by 1TO. During output processing, the first eight words of data are transferred directly from a control point program output buffer to a POT. Additional output is buffered in the system sector of the Rollout File (see fig. 3-16) for subsequent transfer to POTS by 1TO. Special Binary Mode request or status data is transferred directly from a POT to a control point program.

3.3.4.3 Generalized interactive job initiation. Refer to figure 3-29 for this discussion. Assuming that a

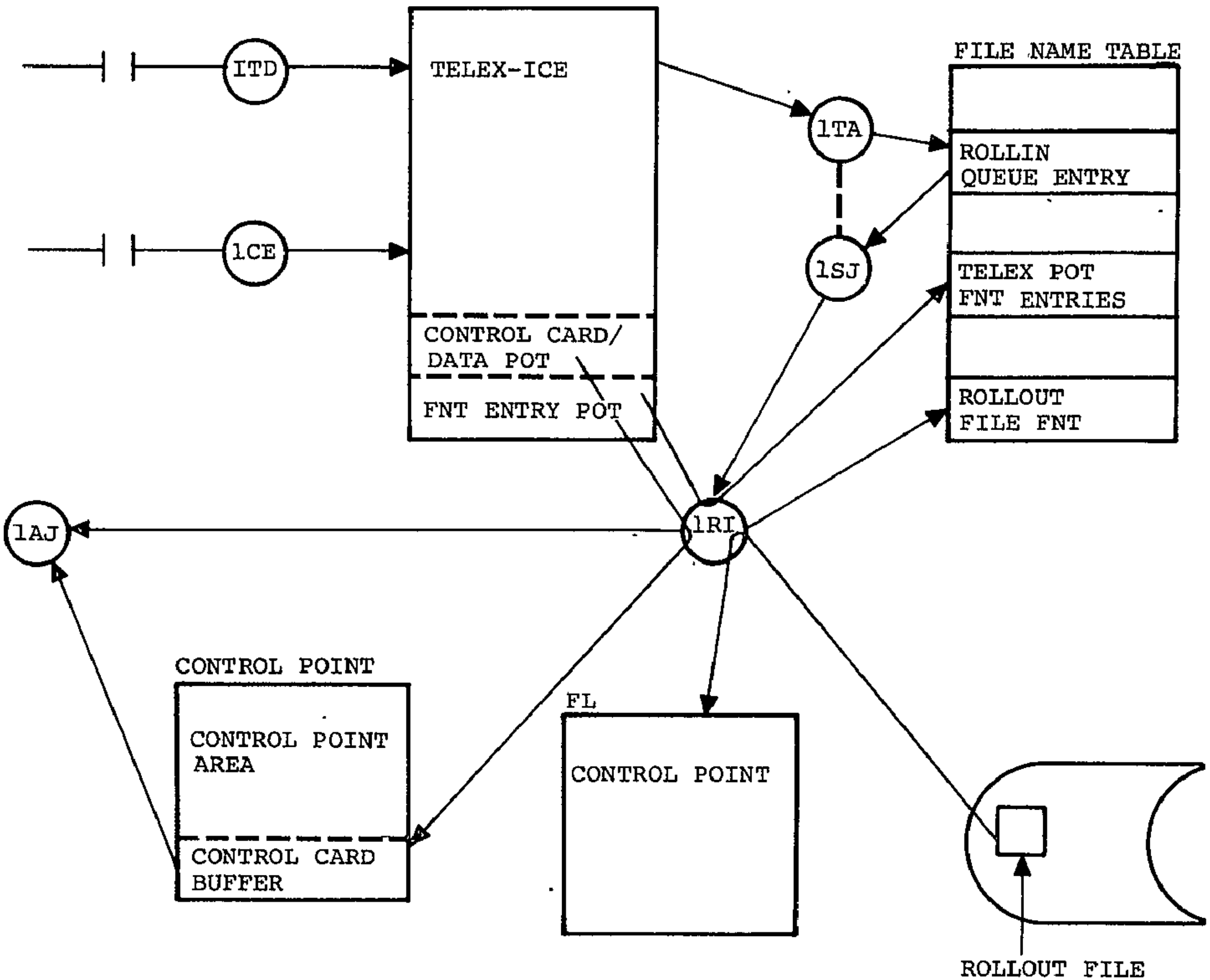
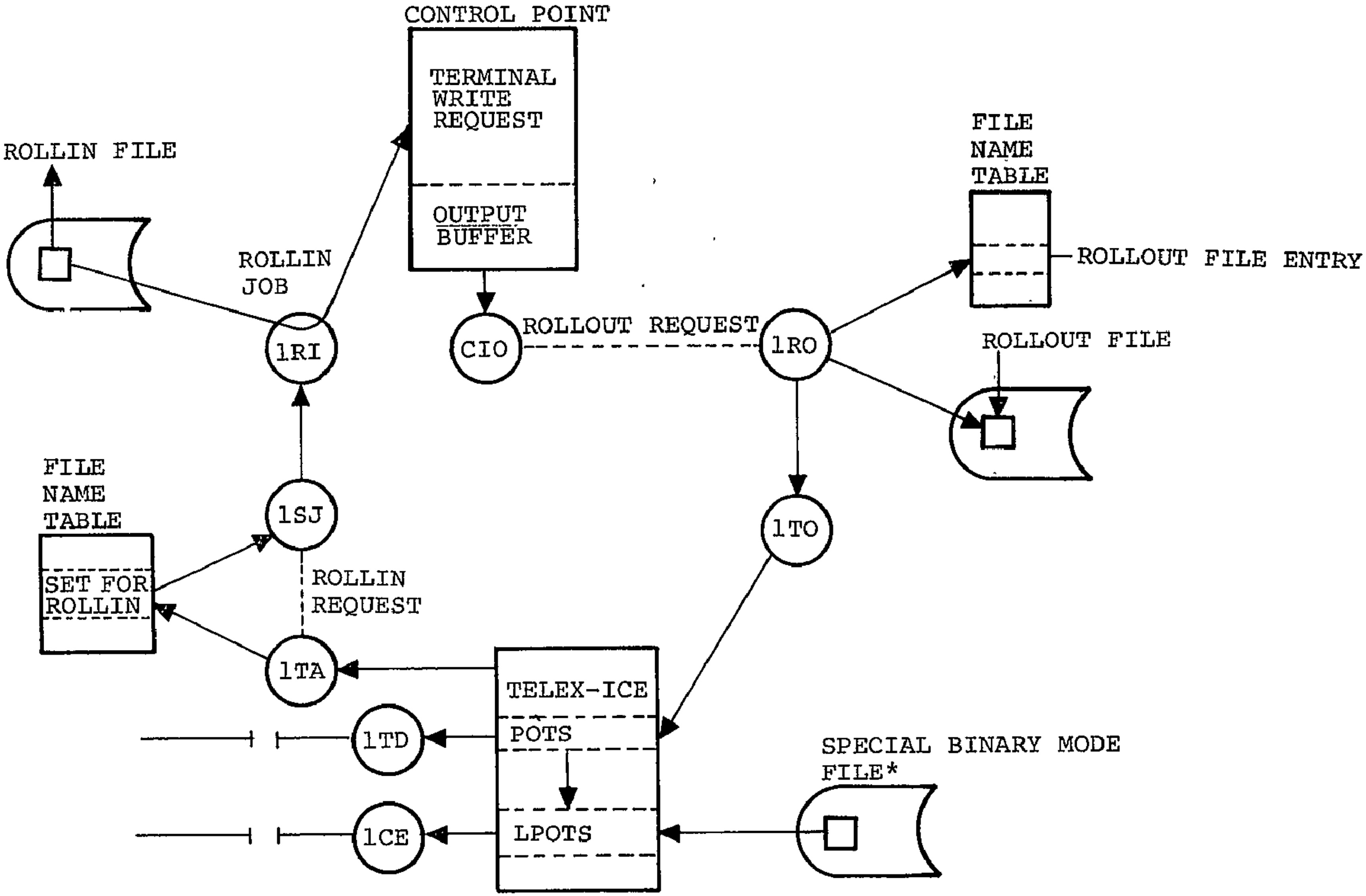


Figure 3-29. — Generalized interactive job initiation.

$SON,___ has been received by TELEX-ICE, the following sequence of events occurs.

- TELEX-ICE builds a Procedure File control card and an FNT entry in a POT and calls 1TA.
- 1TA establishes a rollin queue entry in the system FNT table. This entry points to a Rollout File, the Procedure File control card/data POT and the FNT entry POT.
- 1SJ, the scheduler, determines that this job is the "best job" to initiate. 1SJ assigns a control point and calls 1RI to roll in the job.
- 1RI reads the Rollout File, initializes the control point, sets up the Control Point Area at control point zero, and calls 1AJ. (1AJ is not called to restart an already initiated job.)
- 1AJ advances the job (i.e., reads a control card), detects the procedure file call, and loads the requested application. As the application executes, it interacts with the terminal user.

3.3.4.4 Terminal job interaction - output. Only LDP Type 1, Normal Mode, is discussed here. Applications writing to the terminals via Special Binary Mode are constrained to the following standard system input/output convention. Normal Mode is used to notify TELEX-ICE that a Special Binary Mode File exists and what services (read/write) the application requests. Thus, the following sequence of events occurs regardless of the mode or LDP type. Refer to figure 3-30 for this discussion.



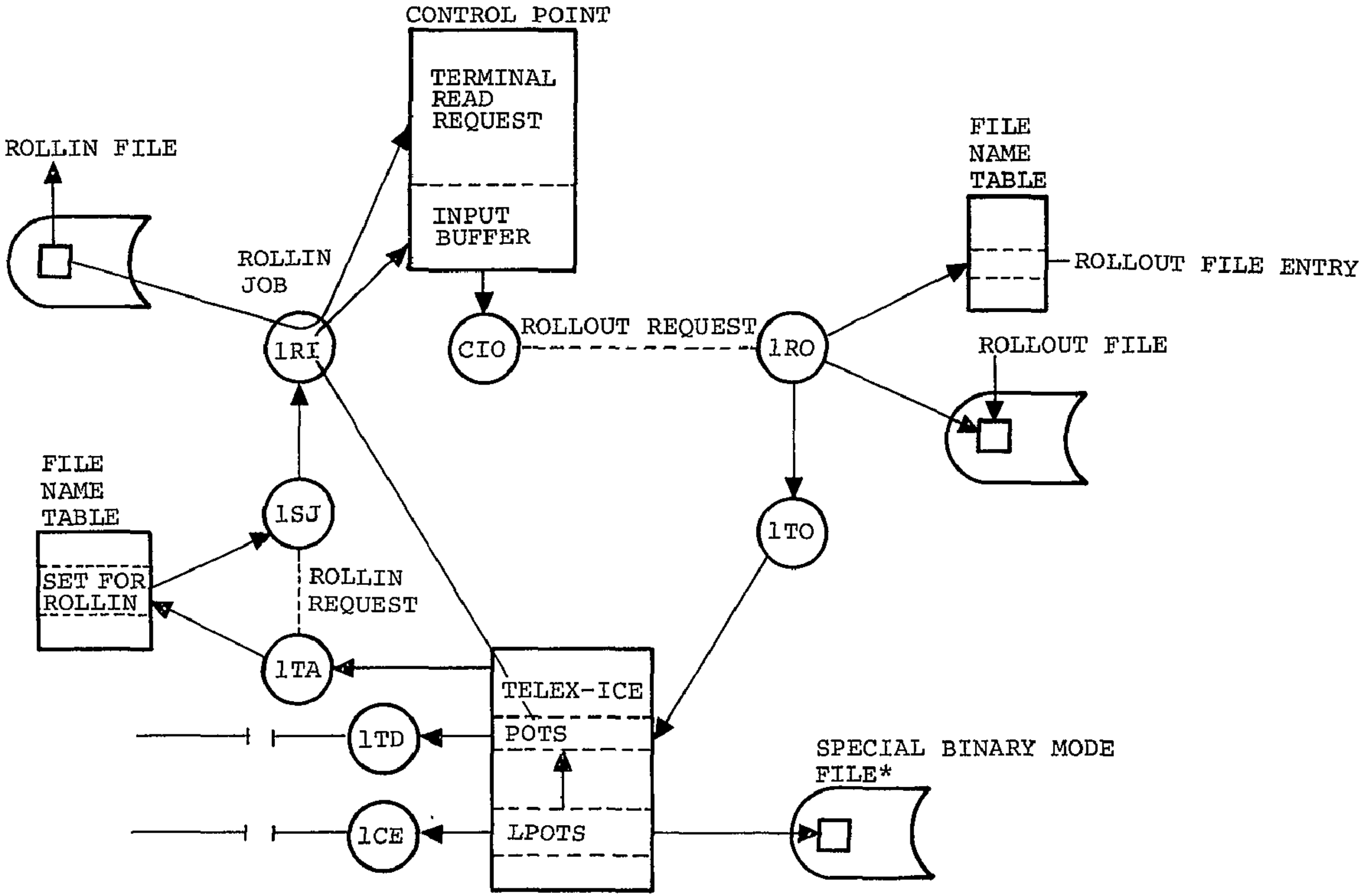
*ICE CALLS 1UF TO LOCATE THE SBM FILE'S FNT IF IT IS ROLLED OUT.

Figure 3-30. – Generalized interactive job – output.

- The job issues a write request to the terminal. The Common Input/Output (CIO) Processor senses that this is an interactive job and issues a request to roll out the control point.

- 1RO initiates the rollout and copies the entire field length (including output data) to the rollout file. 1RO reads the first sector of output data into a special buffer in its PP.
- 1TO is loaded into the same PP as 1RO. The special data buffer is ready to be transferred to TELEX POTS.
- 1TO stores this output data into TELEX POTS and notifies TELEX that terminal output is ready.
- TELEX notifies either 1TD (CDC 6676 multiplexer) or ICE that output data is ready. 1TD transfers the data to the terminal. ICE checks the first two six-bit characters to determine Normal or Special Binary Mode data. Normal Mode data is code converted, given the proper network message envelope, and transmitted to the terminal. Special Binary Mode data is located via 1UF and is read from its file, one message segment at a time, and transmitted to the terminal. No code conversion or message enveloping is generated.
- After all output is transferred, TELEX calls 1TA to reinitiate the job. Section 3.3.4.3 describes this function.

3.3.4.5 Terminal job interaction - input. Assuming that a job is to receive data from a terminal, the following sequence of events occurs (see fig.3-31):



*ICE CALLS 1UF TO LOCATE THE SBM FILE'S FNT IF IT IS ROLLED OUT.

Figure 3-31. — Generalized interactive job — input.

- The job issues a read request on the input file. CIO is called. CIO senses this job is interactive and issues a rollout request.

- 1RO is loaded and executes the rollout operation. 1RO then calls 1TO.
- 1TO checks the Rollout File to assure that all output is transmitted. If there are any Special Binary Mode request codes to be passed, this check forces them to be passed (see section 4.6). 1TO then informs TELEX that the job is ready for input.
- TELEX notifies either 1TD or ICE that an input request has occurred. 1TD issues the prompt character "?". ICE checks the current request code. If Special Binary Mode is requested, no prompt is issued. If Normal Mode is requested, ICE issues a prompt "?".
- 1TD accepts one character at a time from the terminal, code converts it, and stores this data in a TELEX POT. A carriage return code causes 1TD to nofity TELEX that input is complete.
- ICE accepts data, a message segment at a time, and stores this data in an ICE LPOT. An end-of-message indicator in the message head notifies ICE that input is complete.
- ICE checks the input mode for each read request. If Normal Mode is set, ICE strips the network message envelope off and code converts the data. This data is then transferred to a TELEX POT, and TELEX is notified that input is complete. If Special Binary Mode is set, ICE transfers the data to the Special Binary Mode File. When the last message segment has

been transferred to the SBM File, ICE builds a TELEX POT with status information, and TELEX is notified that input is complete.

- TELEX calls 1TA to reinitiate the job. Section 3.4.4.3 describes this function.

3.3.5 S2KMM Executive

This section is to be provided later.

4.0 SPIMS LOGICAL DATA PATHS

A SPIMS Logical Data Path (LDP) is an extension of the defined logical data path (source-host computer) for TCS to illustrate the primary software/hardware interfaces for a SPIMS terminal/application.

4.1 SPIMS LDP TYPES

The LDP's for SPIMS are defined to permit distributed processing control of the functional requirements of an application within the CYBER 74. These functional requirements may be generically defined as follows:

- Terminal interaction (LDP Type 1)
- Data base access, terminal interaction (LDP Type 2)

SPIMS LDP Type 1 is a KRONOS controlled LDP. SPIMS LDP Type 2 is a KRONOS/S2KMM Executive controlled LDP. The illustrations in this section do not specifically show these Executives; this is done for clarity.

4.2 SPIMS LDP MODES

Each LDP type provides two modes for terminal communications:

- Normal Mode - display coded character strings, maximum of 150 characters per read, no limit on writes.

- Special Binary Mode (SBM) - ASCII coded character string, maximum of 5000 characters per read or write.

Normal Mode supports the transfer of display coded character strings, a maximum of 150 characters per terminal read. Terminal writes are buffered (as explained in section 3.3.4.2), thus no output restrictions exist. All terminal control codes and network message enveloping is provided by TELEX-ICE. In addition, Normal Mode is used to communicate SBM request and status data.

SBM supports the transfer of ASCII coded character strings (a maximum of 5000 characters per I/O request).

All code conversion is provided by the SPIMS Common Software package. All terminal control codes and network message enveloping is provided by the SPIMS Common Software. Figure 4-1 illustrates the format of an SBM File on mass storage. Figure 4-2 illustrates a possible format for an applications SBM data buffer.

4.3 USER SIGN-ON

Figure 4-3 illustrates the user/system interaction to establish a connection with TCS. Figure 4-4 illustrates a production application sign-on. Figure 4-5 illustrates a development application sign-on.

All interaction between the user and the SPIMS is initially conducted in Normal Mode via an LDP Type 1.

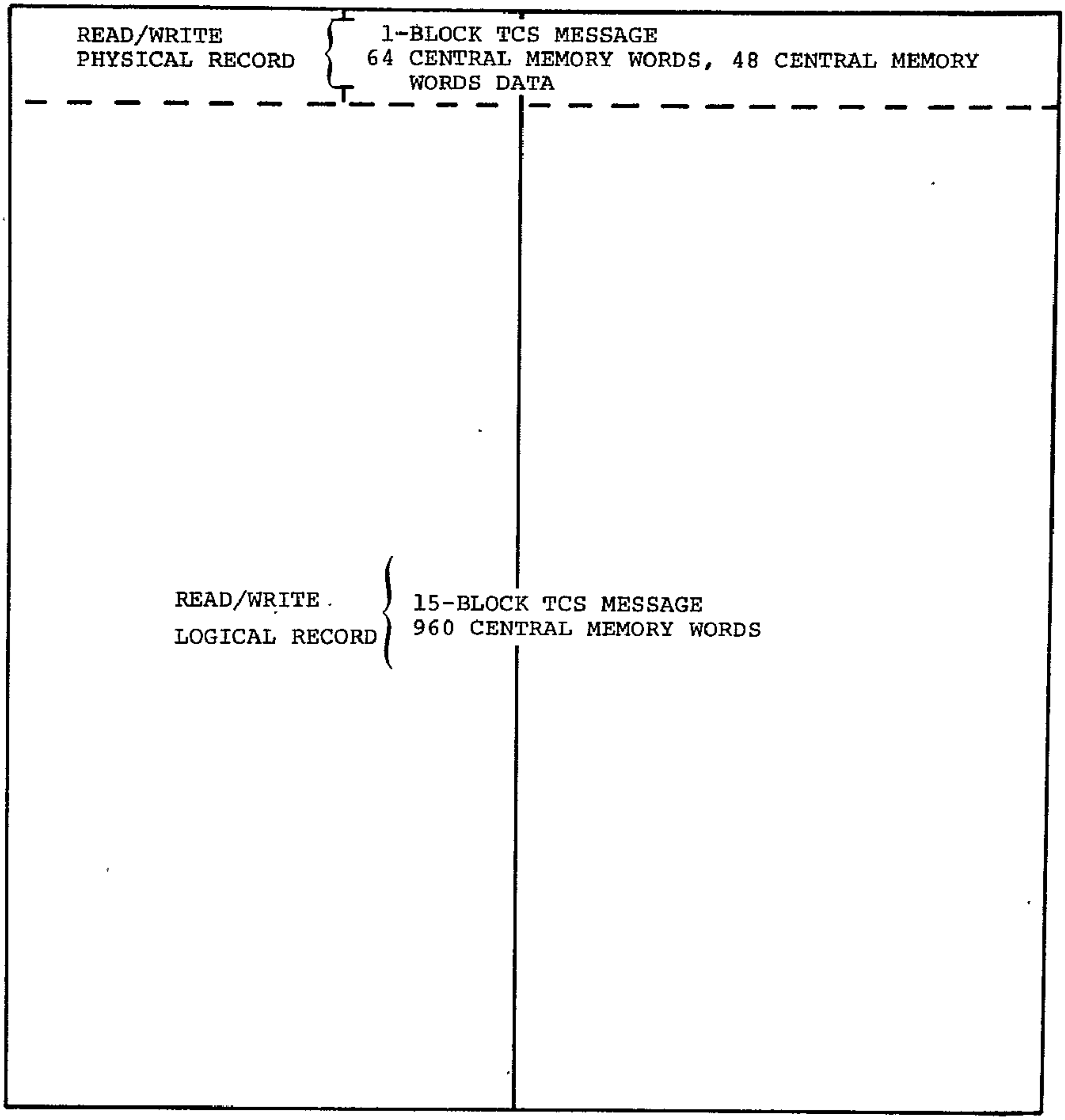
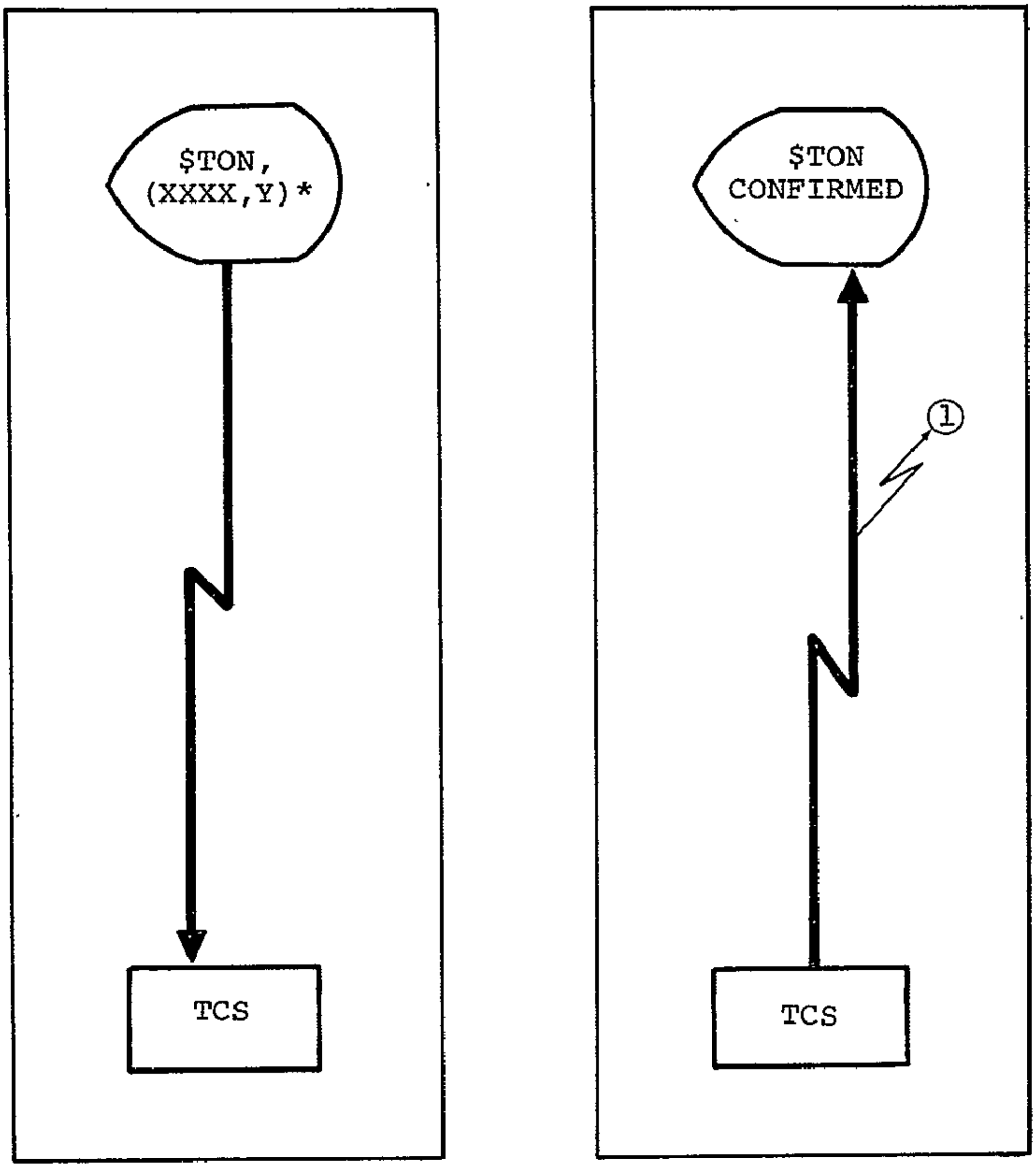


Figure 4-1. - Format of an SBM File.

TCS MESSAGE BLOCK 1	9 ASCII Bytes HEADER 351 ASCII Bytes Data* 351 Characters maximum	64 CENTRAL MEMORY WORDS
TCS MESSAGE BLOCK 2	351 ASCII Bytes Data 702 Characters maximum	
TCS MESSAGE BLOCK 3	351 ASCII Bytes Data 1053 Characters maximum	
TCS MESSAGE BLOCK 4	351 ASCII Bytes Data 1404 Characters maximum	
TCS MESSAGE BLOCK 5	351 ASCII Bytes Data 1755 Characters maximum	
TCS MESSAGE BLOCK 6	351 ASCII Bytes Data 2106 Characters maximum	
TCS MESSAGE BLOCK 7	351 ASCII Bytes Data 2457 Characters maximum	
TCS MESSAGE BLOCK 8	351 ASCII Bytes Data 2808 Characters maximum	
TCS MESSAGE BLOCK 9	351 ASCII Bytes Data 3159 Characters maximum	
TCS MESSAGE BLOCK 10	351 ASCII Bytes Data 3510 Characters maximum	
TCS MESSAGE BLOCK 11	351 ASCII Bytes Data 3861 Characters maximum	
TCS MESSAGE BLOCK 12	351 ASCII Bytes Data 4212 Characters maximum	
TCS MESSAGE BLOCK 13	351 ASCII Bytes Data 4563 Characters maximum	
TCS MESSAGE BLOCK 14	351 ASCII Bytes Data 4914 Characters maximum	
TCS MESSAGE BLOCK 15	351 ASCII Bytes Data 5265 Characters maximum	

*These bytes are inverted

Figure 4-2. — Format of special binary data buffer.



*XXXX = TERMINAL USER ID

Y = DEVICE TYPE 1 — H4000G SYNCHRONOUS
DEVICE TYPE 2 — H4000G ASYCHRONOUS
DEVICE TYPE 3 — H2000G ASYCHRONOUS
DEVICE TYPE 4 — TTY COMPATIBLE ASYCHRONOUS

(1) Illegal user ID or terminal type causes nonconfirmation

Figure 4-3. — Terminal-to-TCS access with $TON,XXX,Y command.

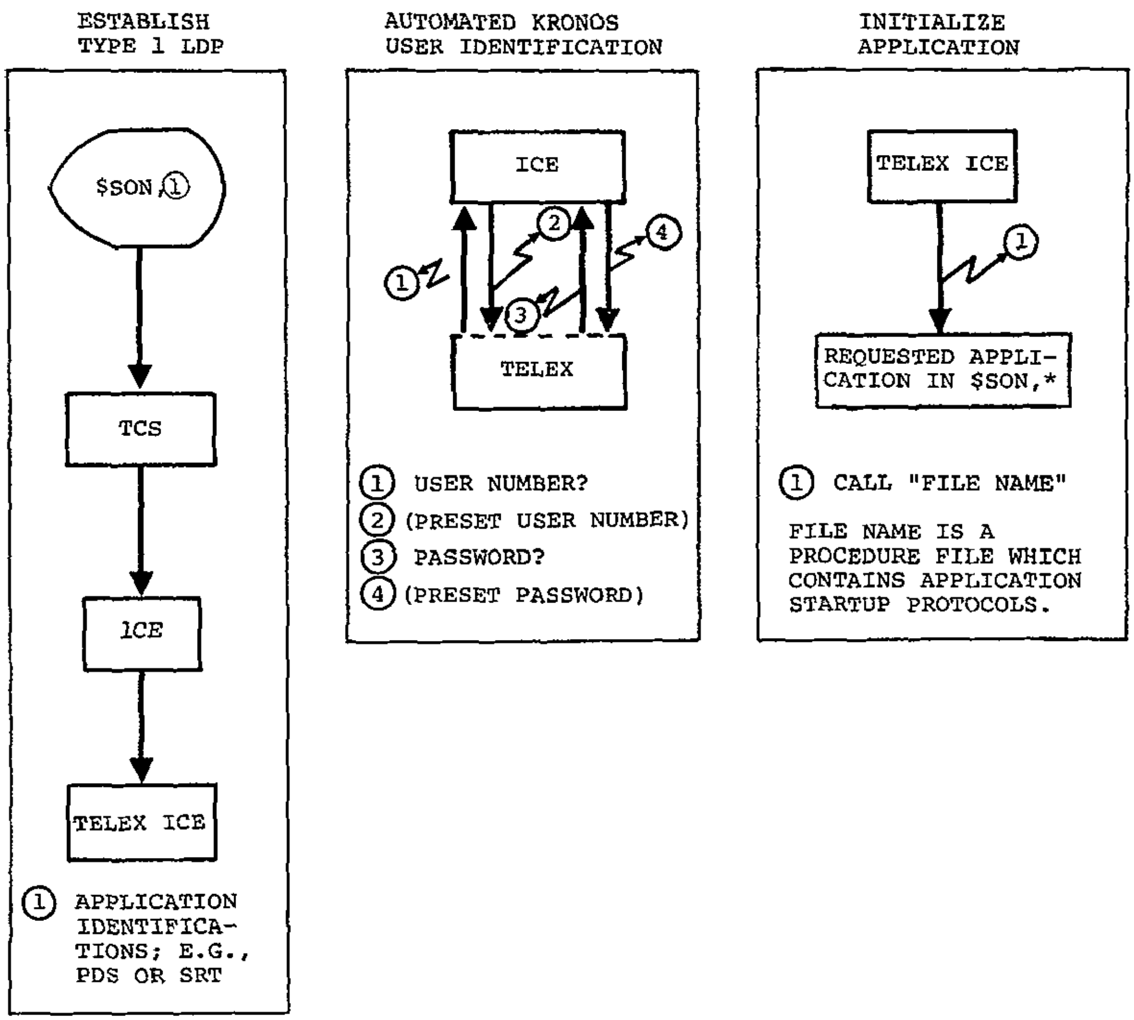
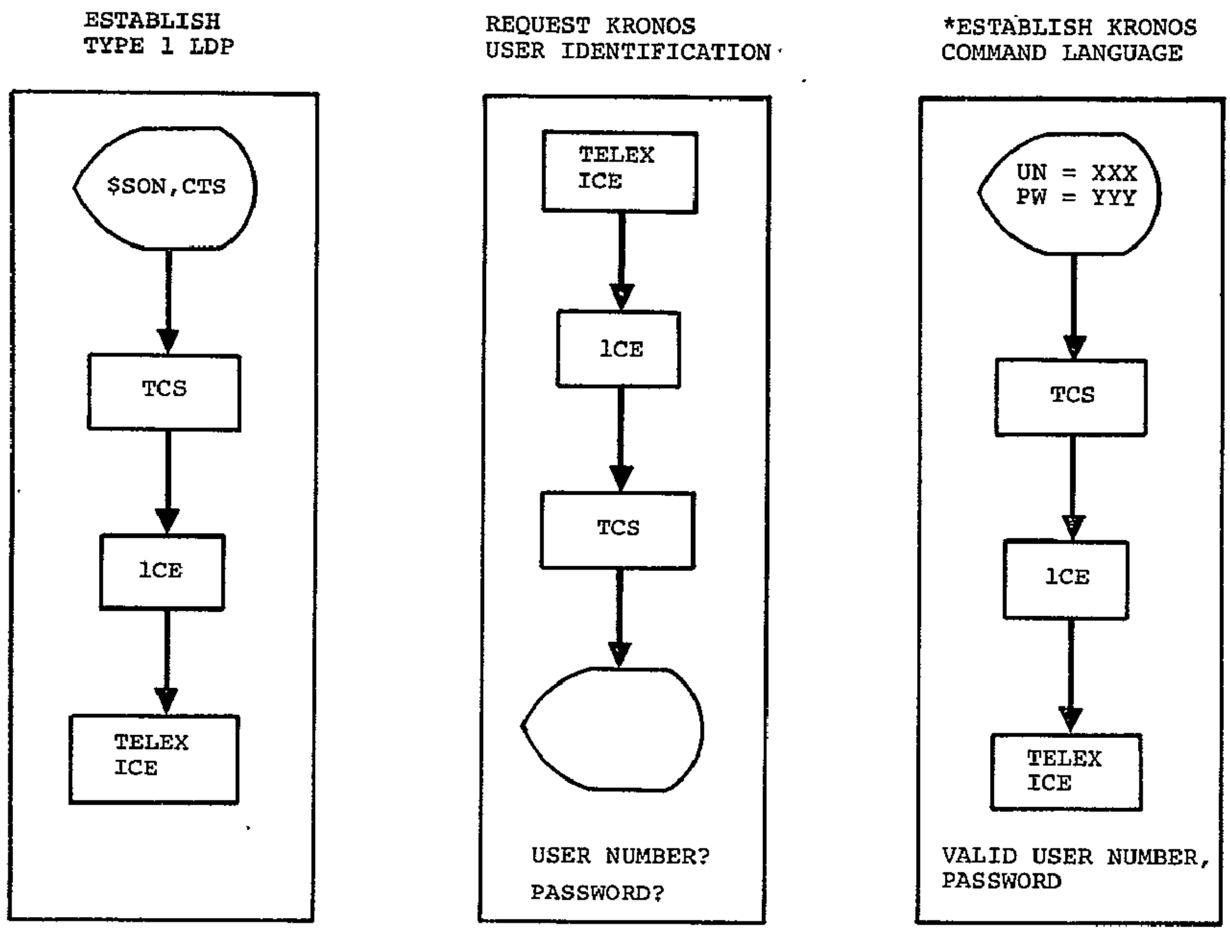


Figure 4-4. — Production application sign-on.



*FIVE RETRIES WITH INVALID USER NUMBER OR PASSWORD WILL TERMINATE THE LDP.

Figure 4-5. — Development application sign-on.

4.4 ESTABLISH AN LDP TYPE 2

A Type 2 LDP is established within the CYBER 74 using any of three methods.

- Control card - all directives are provided by the file input.
- Control card file substitution - all directives are read from a predefined file.

Figure 4-6 illustrates the control card methods.

4.5 TERMINAL INPUT/OUTPUT NORMAL MODE

Figures 4-7 through 4-10 illustrate the interfaces and data flow for terminal reads and writes. This terminal I/O has the following:

- 150 character maximum input
- Display coded data at the Application Level for all I/O.
- Line/Line scrolling at the terminal.

4.6 ESTABLISH TERMINAL I/O SPECIAL BINARY MODE

Special Binary Mode files are created via standard CIO requests. The transfer of these file names to ICE is accomplished using a function code read/write sequence between the application and ICE. Figure 4-11 illustrates this sequence for LDP Type 1. Figure 4-12 illustrates this sequence for a Type 2 LDP.

SPIMS LOGICAL DATA PATH TYPE 2

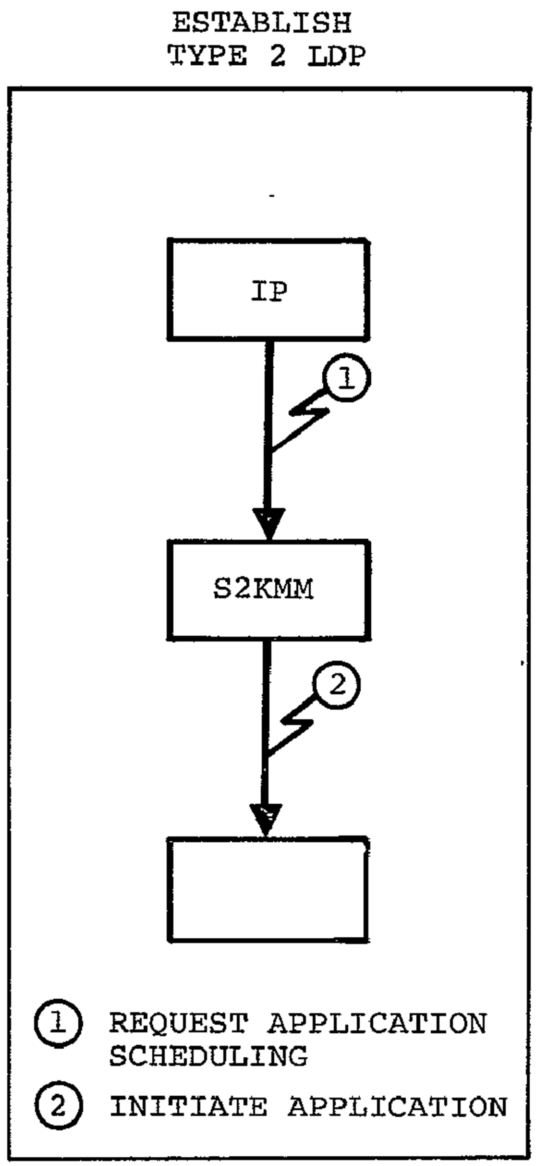


Figure 4-6. – Control card initiated Type 2 LDP.

SPIMS LOGICAL DATA PATH TYPE 1

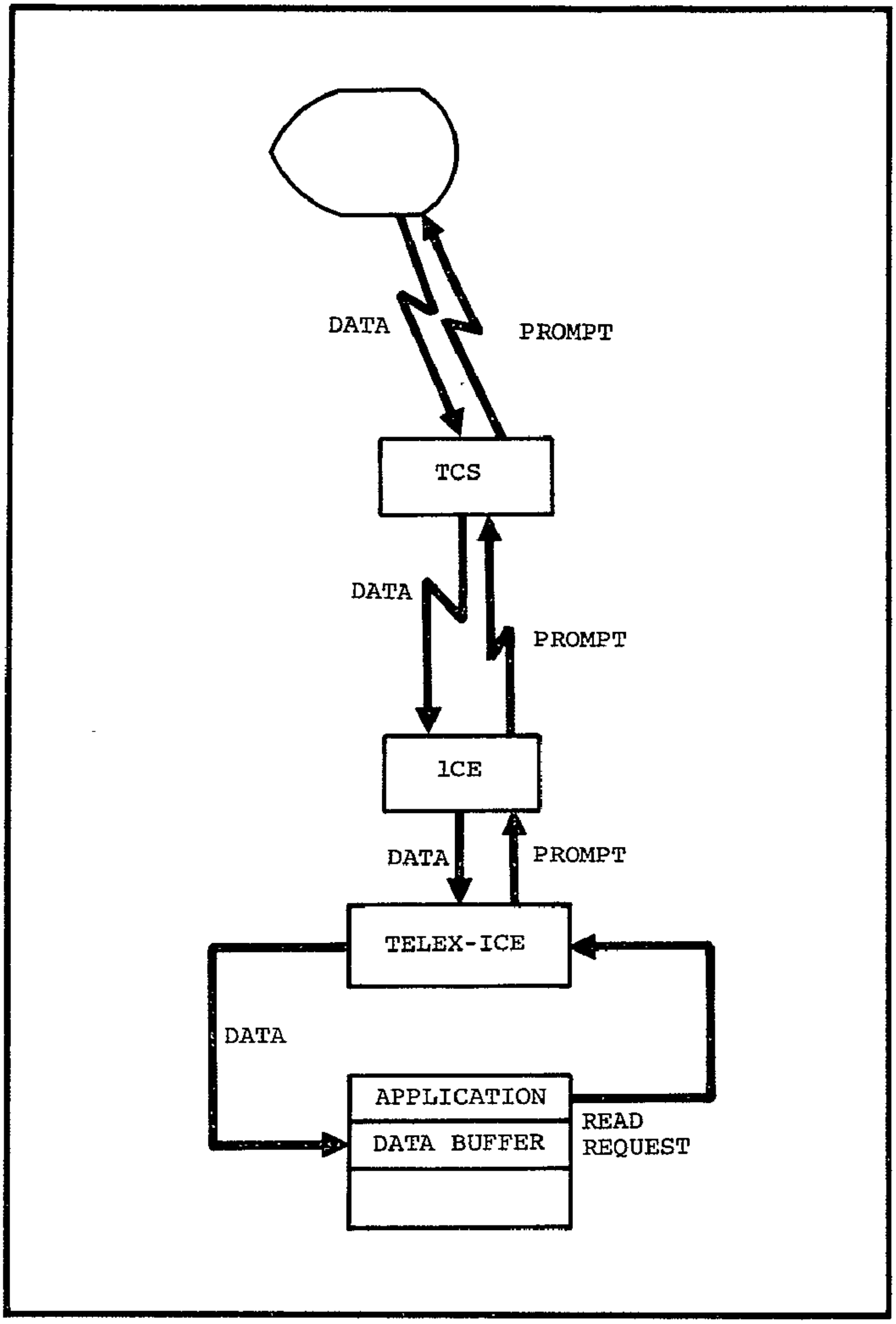


Figure 4-7. — Application-to-terminal read (Normal Mode) for LDP Type 1.

SPIMS LOGICAL DATA PATH TYPE 1

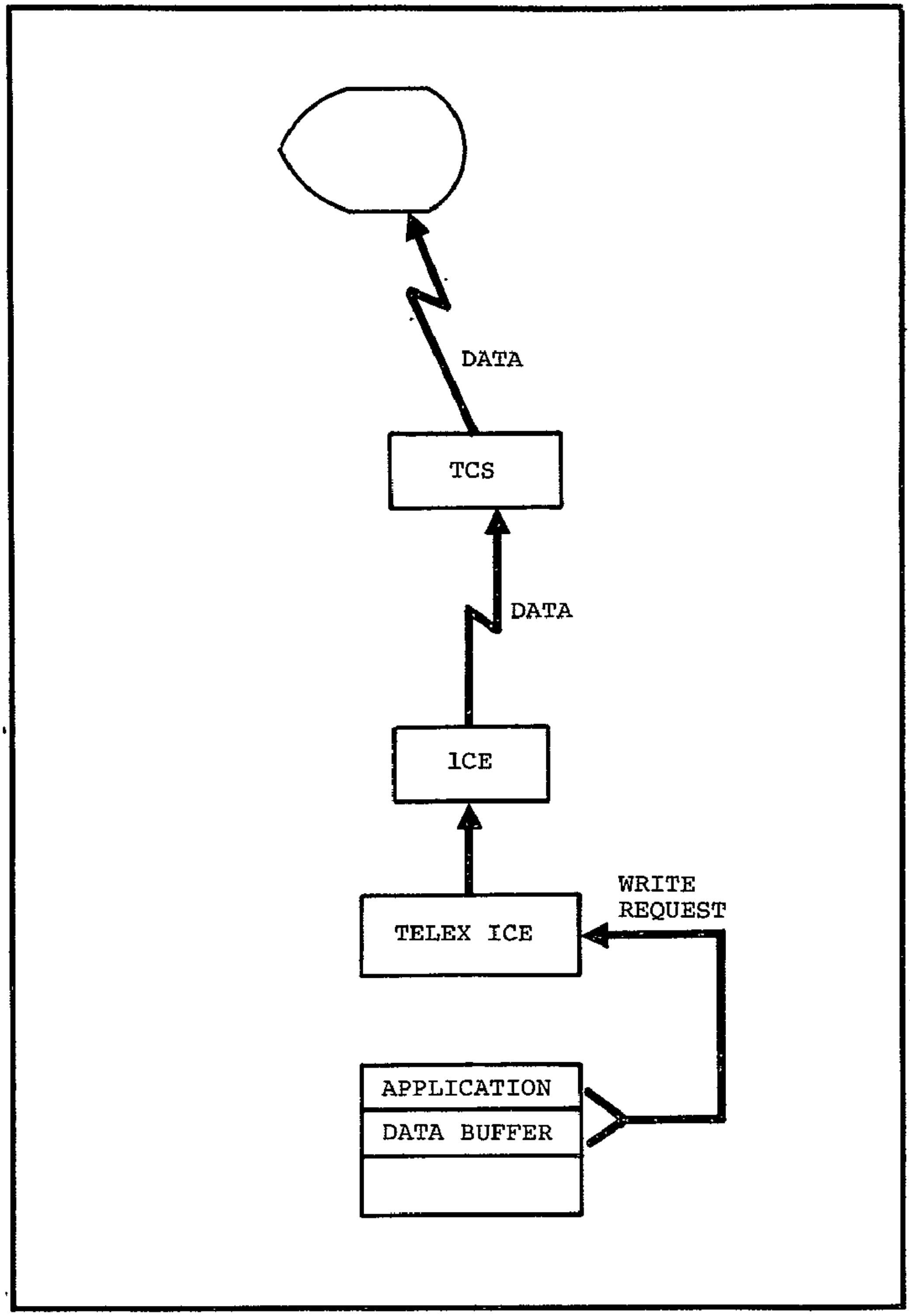


Figure 4-8. - Application-to-terminal write (Normal Mode) for LDP Type 1.

SPIMS LOGICAL DATA PATH TYPE 2

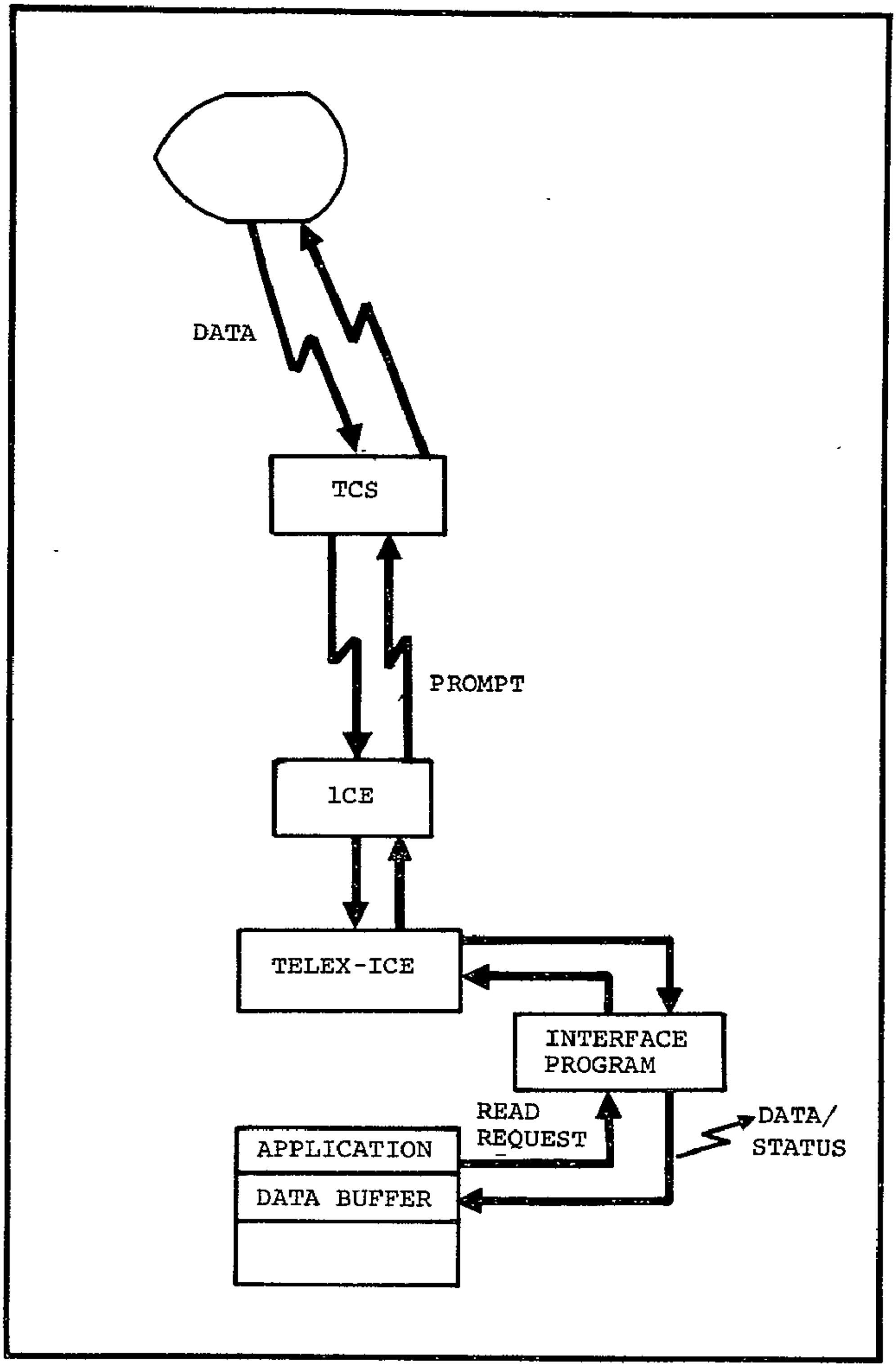


Figure 4-9. — Application-to-terminal read (Normal Mode) for LDP Type 2.

SPIMS LOGICAL DATA PATH TYPE 2

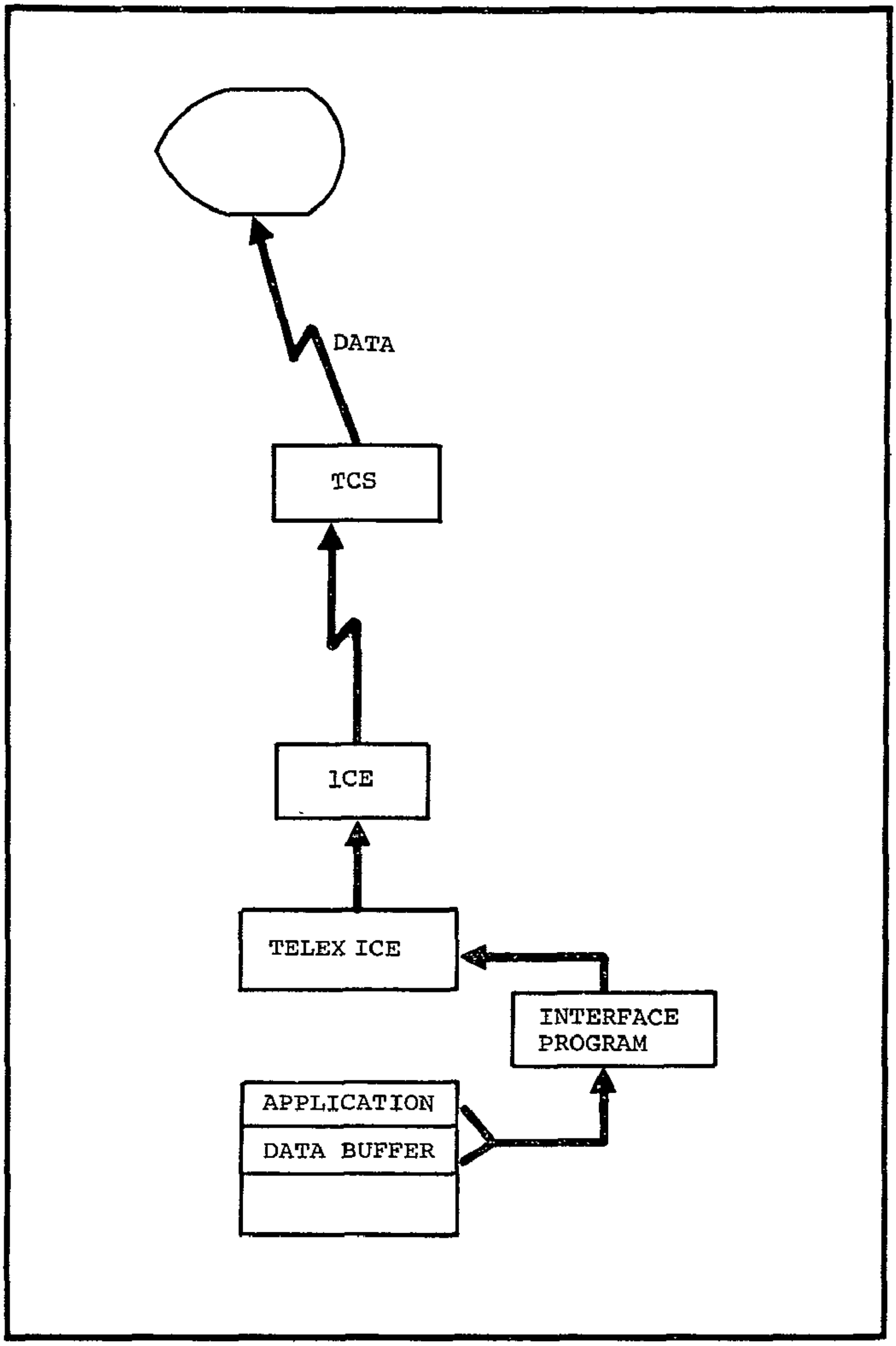
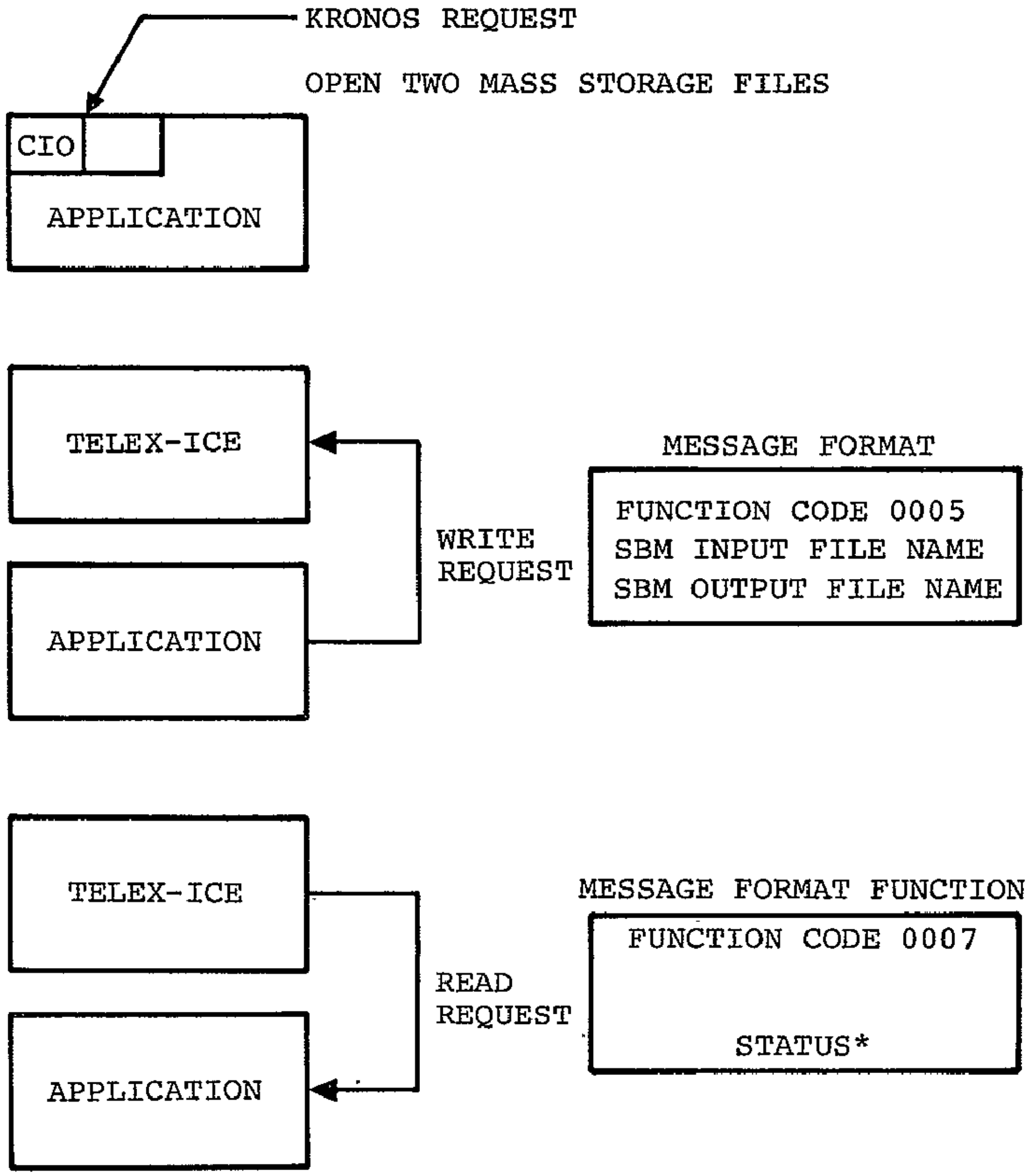


Figure 4-10. — Application-to-terminal write (Normal Mode) for LDP Type 2.



*0000 — SUCCESSFUL
0001 — LOST DATA
0002 — BAD DATA (HEADER)
0003 — FNT ENTRIES NOT FOUND
0004 — ALREADY INITIALIZED
0020 — USER TERMINATED

Figure 4-11. — Establish special binary mode files LDP type 1.

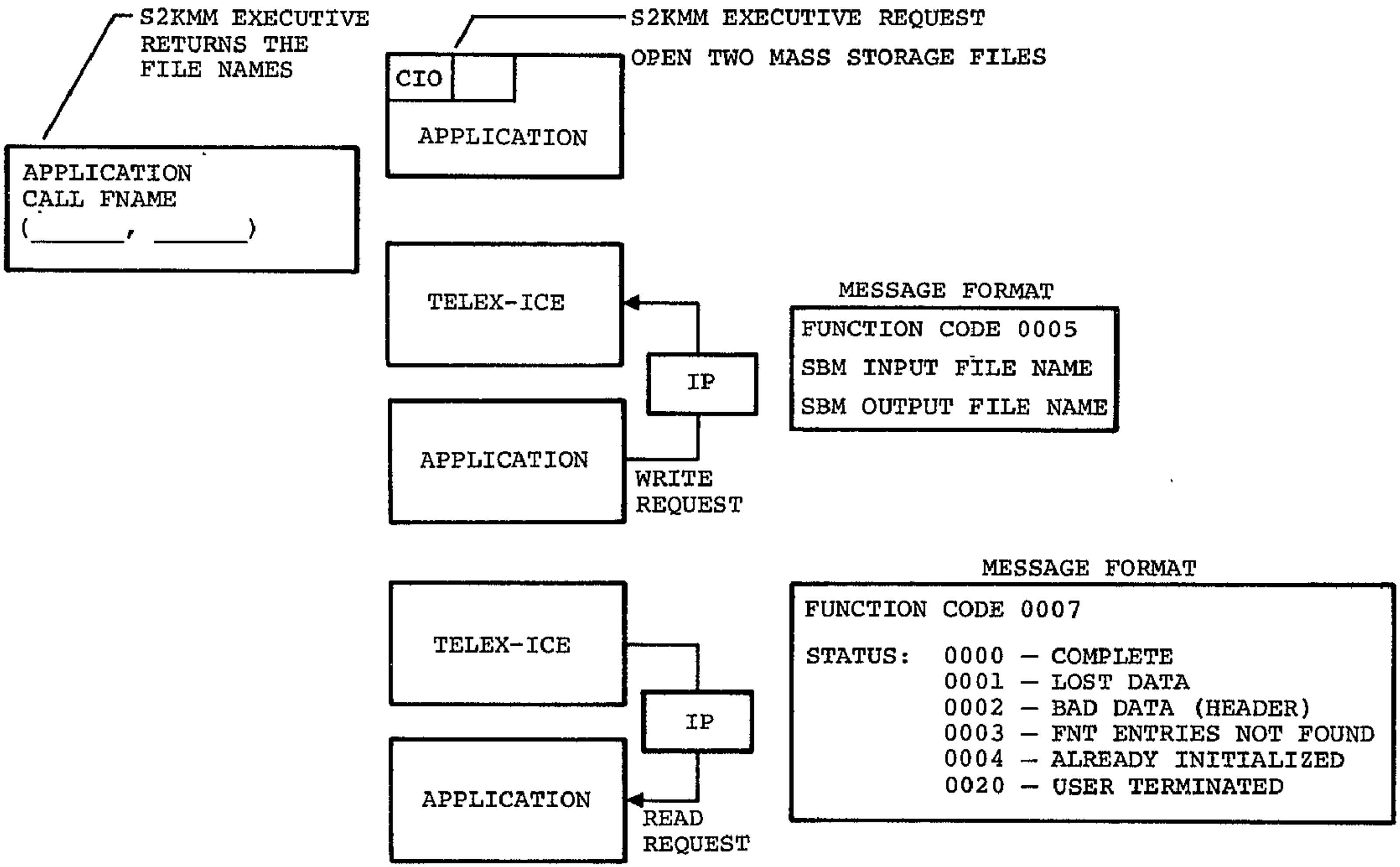


Figure 4-12. — Establish Special Binary Mode Files LDP Type 2.

4.7 TERMINAL I/O SPECIAL BINARY MODE

Special Binary Mode (SBM) files are read or written via the SPIMS Common Software Library and the standard system I/O requests for binary reads and writes. Special function codes have been established to determine the TELEX-ICE I/O processing required. These function codes are the first 12 bits of a Normal Mode Write. The codes and services requested are:

0005 - SBM files Initialization Request
0006 - SBM Input Request
0007 - SBM Output Request
00021 - Application Termination Complete

TELEX-ICE issues a return code and status for each request in the first 12 bits of data. This return code indicates the service requested and in the case of function code 0005 the terminal type interacting with the application. The return codes are:

4040 - Response to SBM Initialization Request (0005)
4047 - Response to SBM Input Request (0006)
4050 - Response to SBM Output Request (0007)

The SBM status codes are:

4040 - Successful Transmission
4041 - Data Lost
4042 - Bad Data (TCS header only)
4043 - FNT Entries Not Found
4044 - SBM Already Initialized
4045 - Applications User Number Not Found
4046 - Message Too Long
4047 - Illegal Function
4051 - Rerequest Previous Function
4061 - User Signed OFF

The Normal Mode path is used for these sequences. Figures 4-13 through 4-16 illustrate the data interfaces and data flows for terminal read/writes.

SPIMS LOGICAL DATA PATH TYPE 2

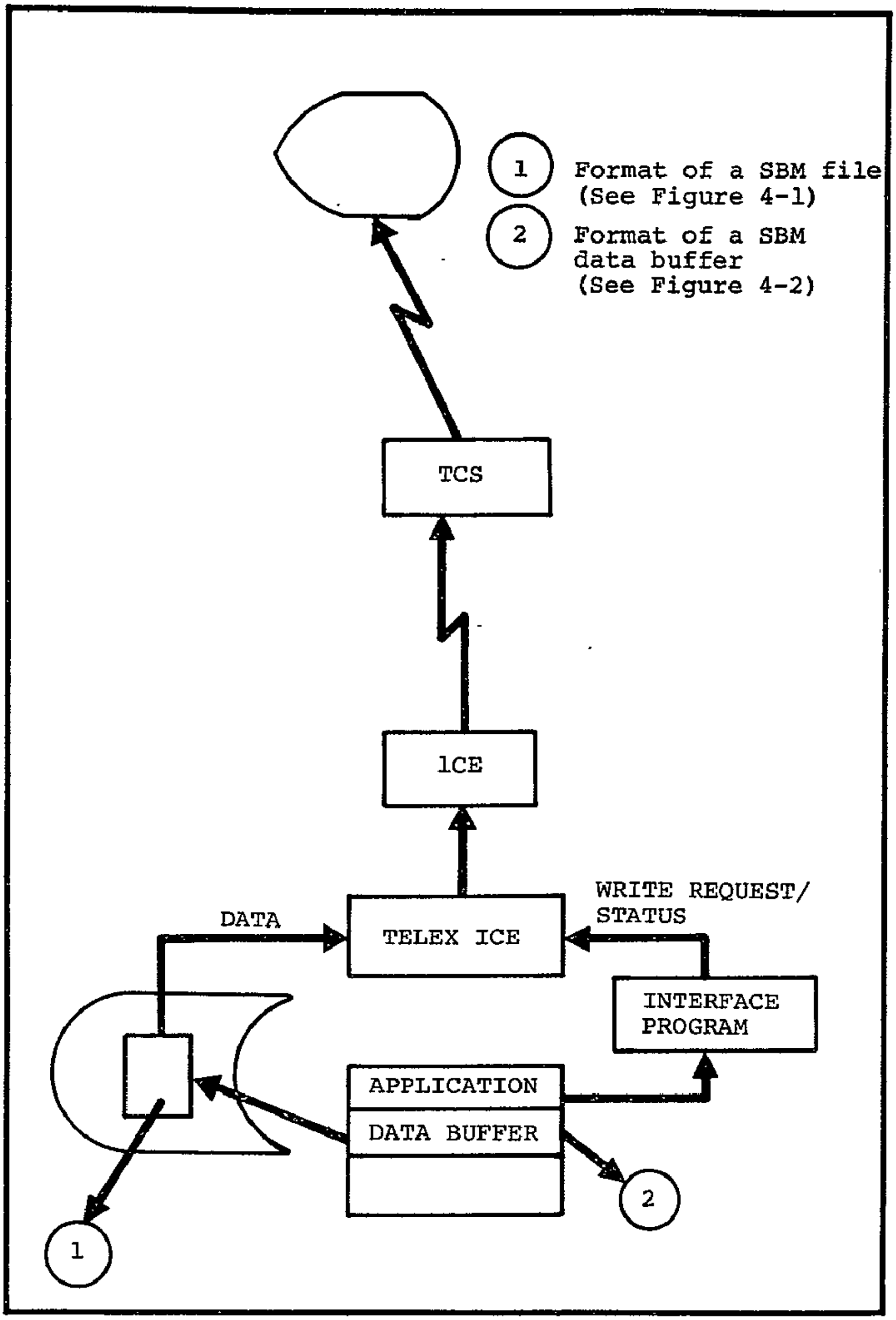


Figure 4-13. – Application-to-terminal write (Special Binary Mode) for LDP Type 2.

SPIMS LOGICAL DATA PATH TYPE 2

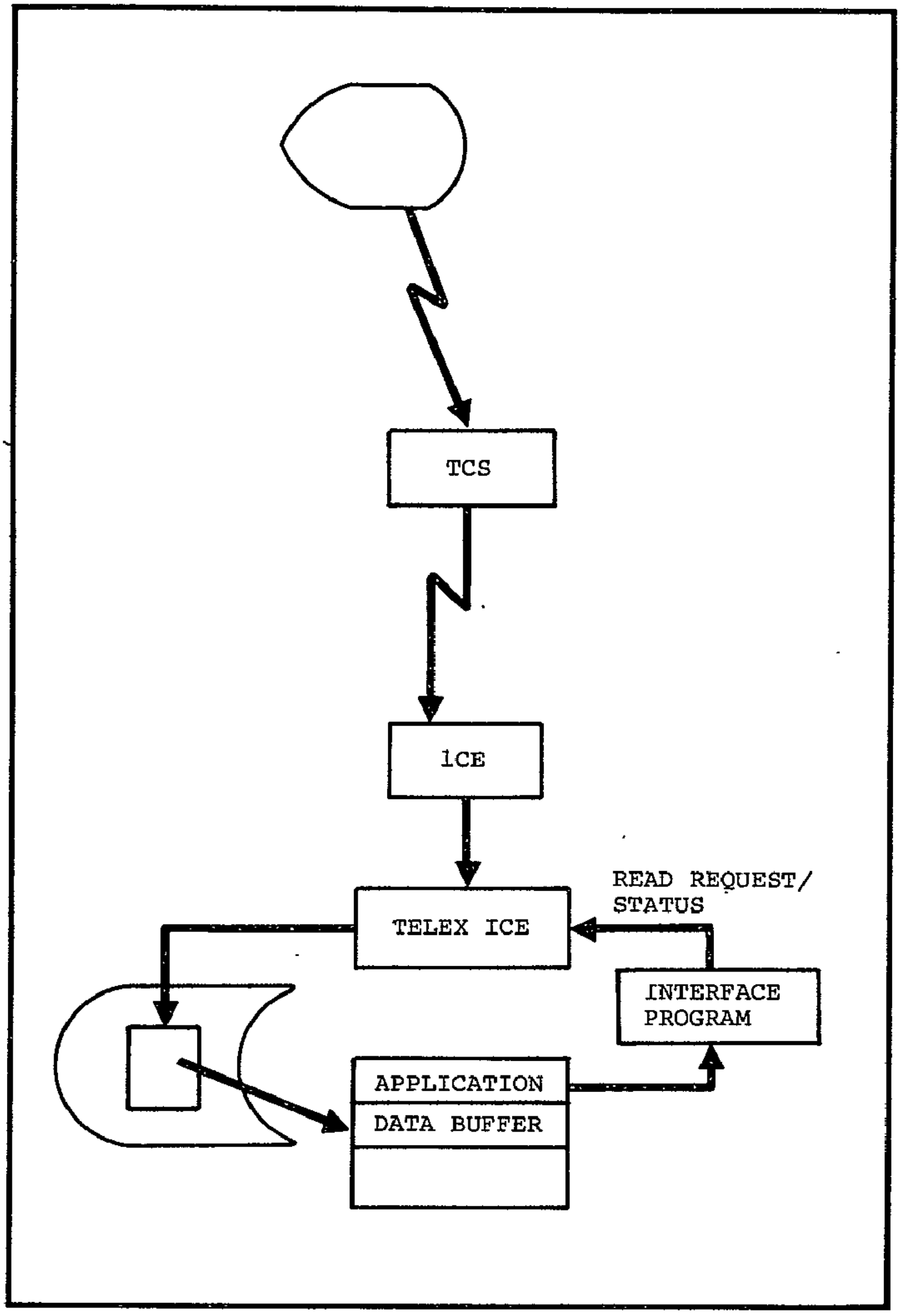


Figure 4-14. – Application-to-terminal read (Special Binary Mode) for LDP Type 2.

SPIMS LOGICAL DATA PATH TYPE 1

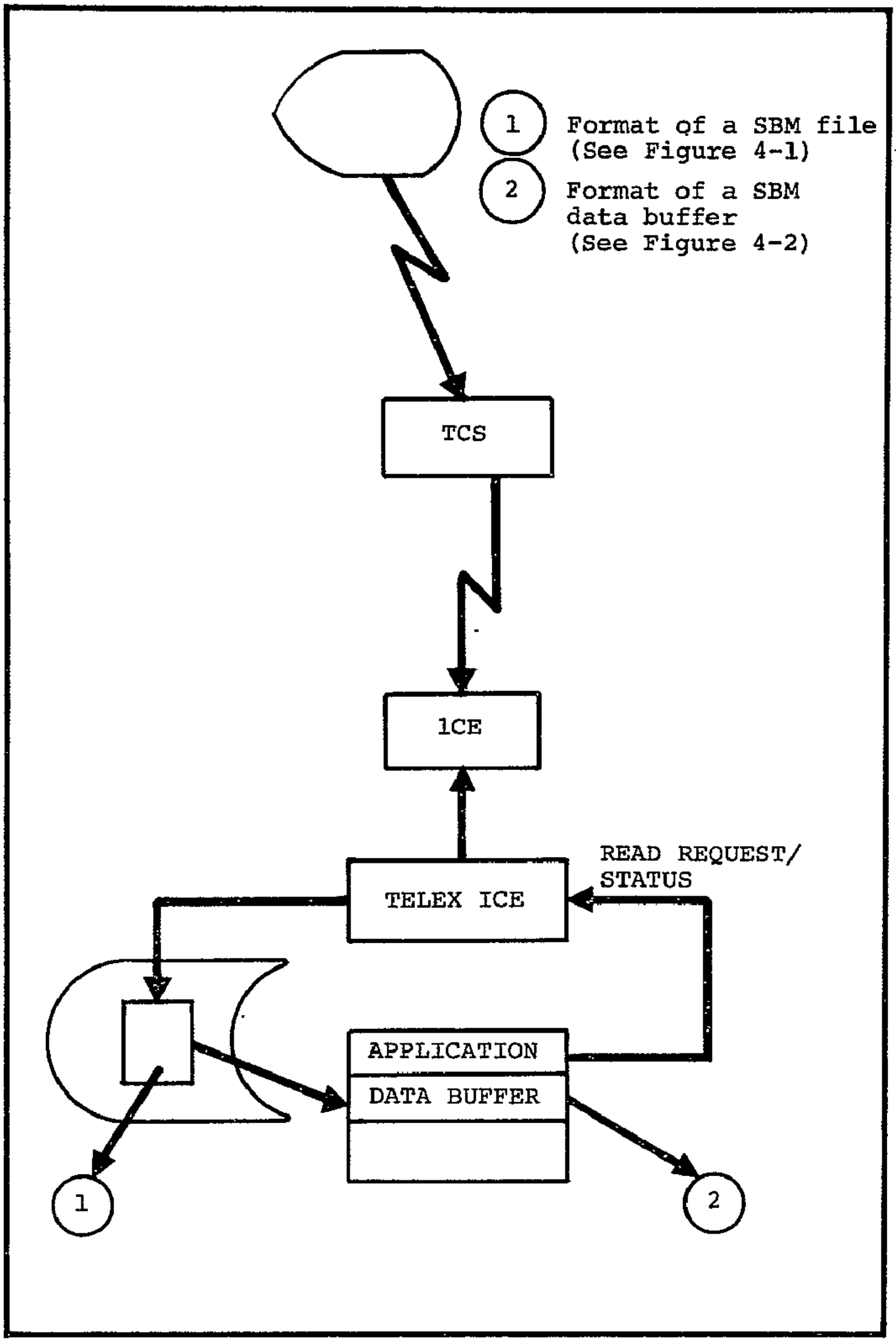


Figure 4-15. — Application-to-terminal read (Special Binary Mode) for LDP Type 1.

SPIMS LOGICAL DATA PATH TYPE 1

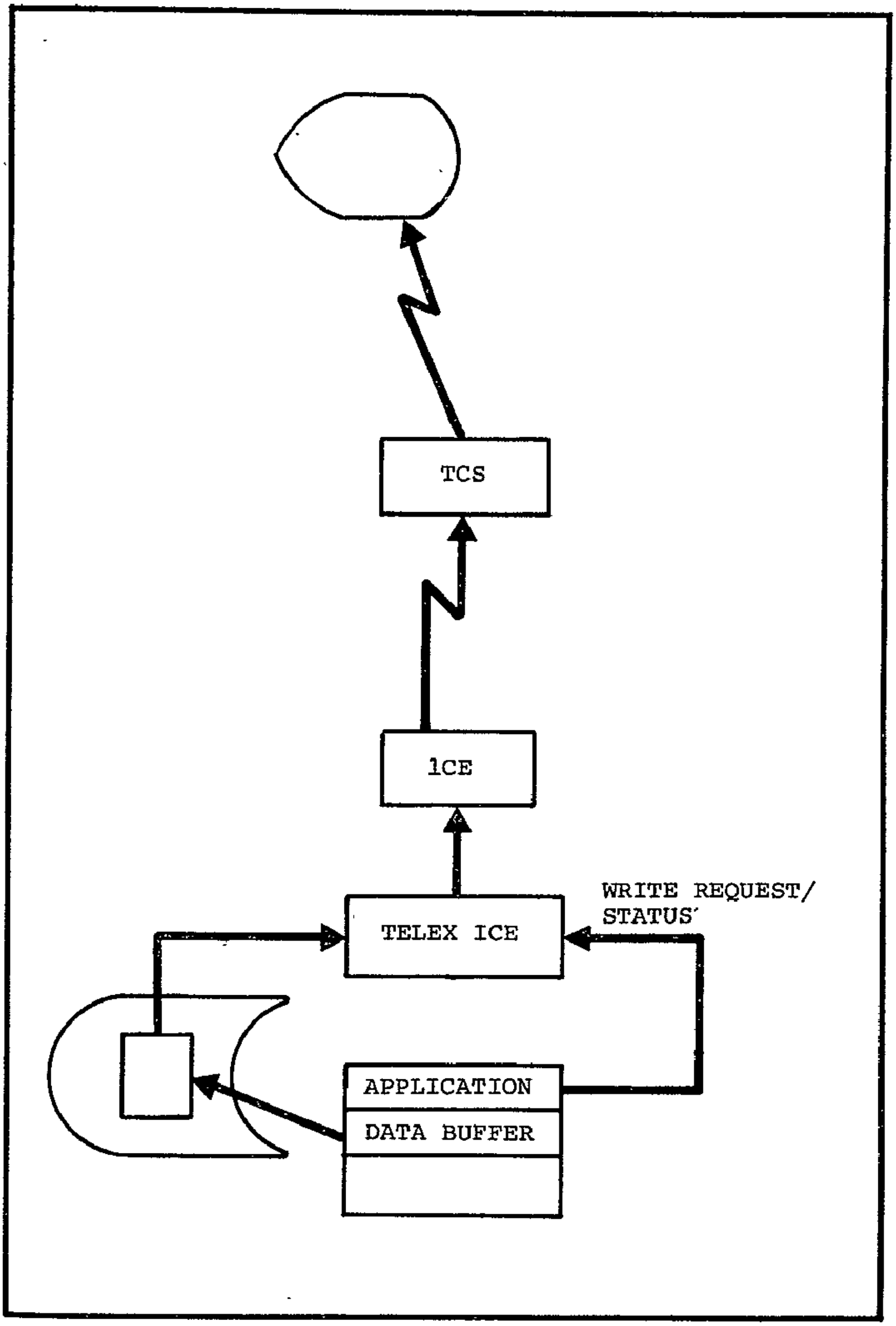


Figure 4-16. — Application-to-terminal write (Special Binary Mode) for LDP Type 1.

4.8 USER SIGN-OFF

Figure 4-17 illustrates a production application sign-off. Figure 4-18 illustrates a development application sign-off. Figure 4-19 illustrates the user commands/system interaction to disconnect from TCS.

SPIMS LOGICAL DATA PATH TYPE 2

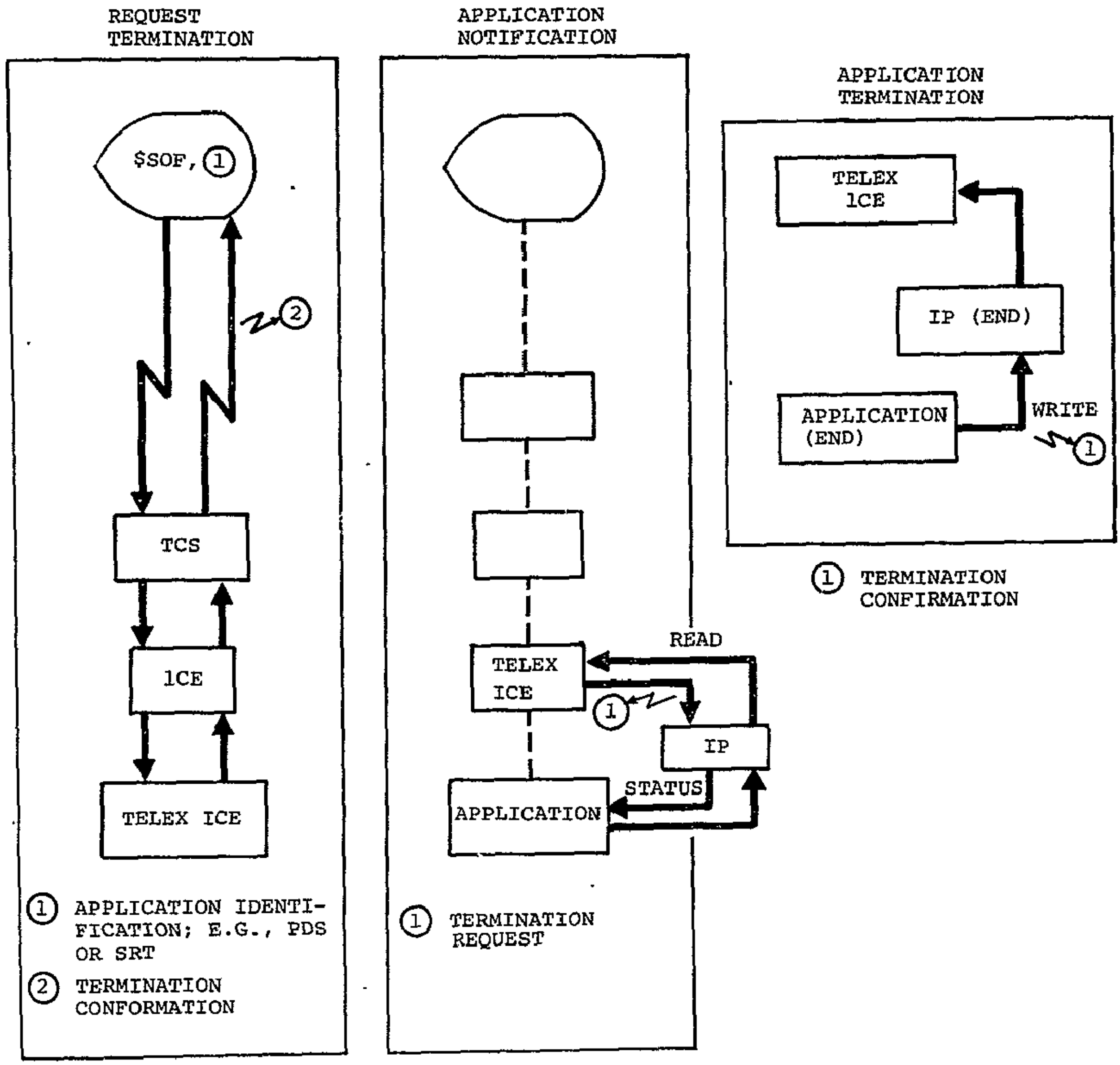


Figure 4-17. – Production application sign-off.

SPIMS LOGICAL DATA PATH TYPE 1

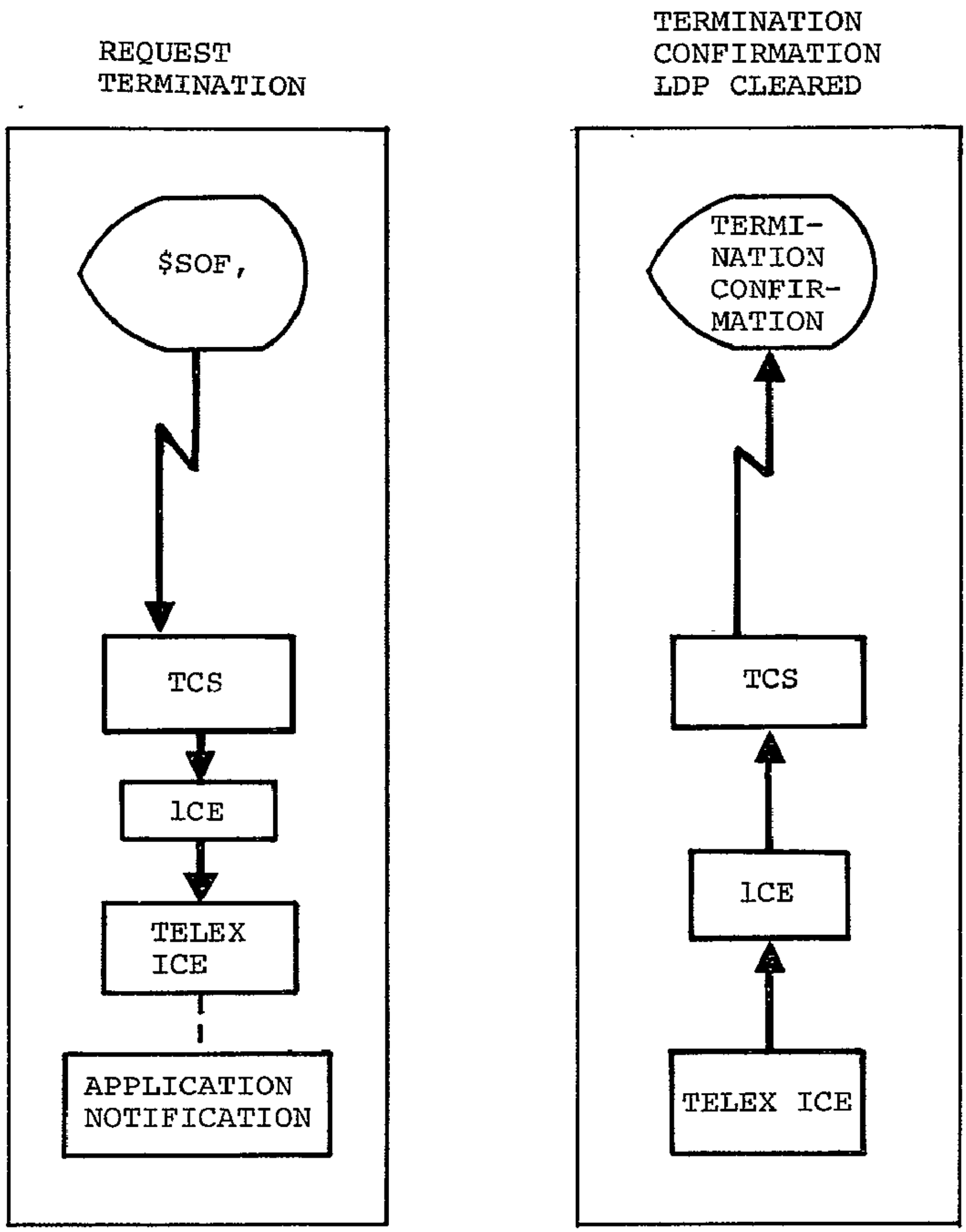
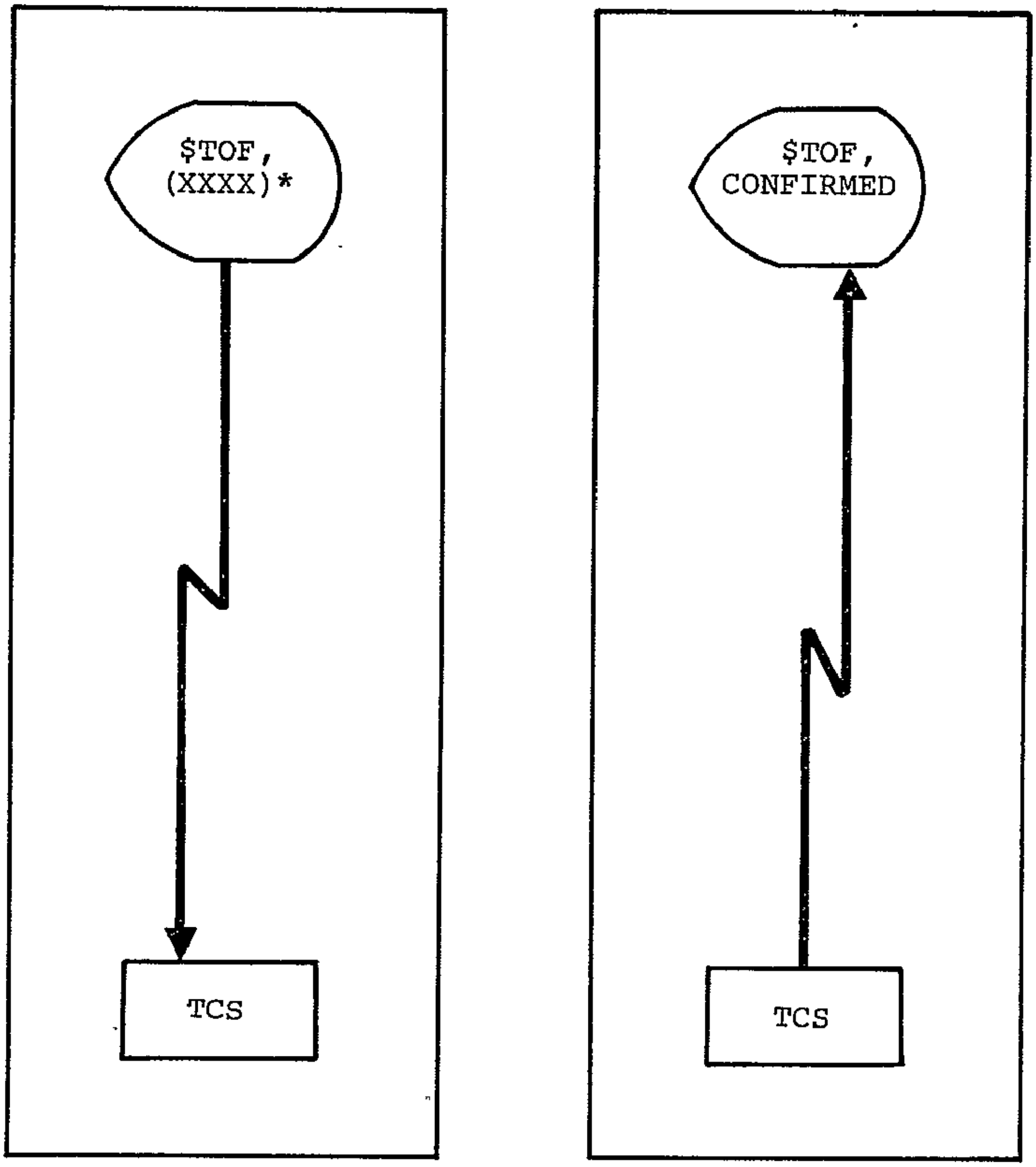


Figure 4-18. — Development application sign-off.



*XXXX = TERMINAL USER ID

Figure 4-19. – Terminal-to-TCS disconnect with $TOF,XXXX command.

5.0 GENERALIZED SPIMS APPLICATIONS DESIGN

TO BE PROVIDED IN RELEASE TWO

6.0 SPIMS CONTROLLER FUNCTIONS

TO BE PROVIDED IN RELEASE TWO

7.0 SUMMARY OF THE SPIMS SUPPORT CAPABILITIES TO BE PROVIDED IN RELEASE TWO

APPENDIX - A

BIBLIOGRAPHY

Program Information and Coordination Review Service (PICRS) Management Plan. JSC-08100, June 28, 1974.

Implementation Feasibility Study for Space Shuttle Program Management Systems Application Final Report. Lockheed Electronics Company, Inc., LEC-1787, March 1974.

System Requirements Document for the Shuttle Program Information Management System, JSC-09381, April 1975.

Hazeltine 2000 Operating Manual. Hazeltine Corp., HI1004, revised August 1973.

CRT Terminal System Software Specifications. Philco Houston Operations, PHO SI-09637A, May 9, 1972.

Institutional Data Systems Procedure Manual, MSCM-2400, Johnson Space Center.

Terminal Control System Requirements and User's Guide, GDSD, SSB0004.

Control Data KRONOS 2.1 Workshop Reference Manual, Control Data Corp., 97404700B.

SPIMS/CYBER Interface Control Document, Control Data Corp., 90CLKA00021-1, January 31, 1975.

Detailed Requirements and Functional Design Document for Shuttle Mass Properties Automated System, Lockheed Electronics Corporation, LEC-5194, December, 1974.