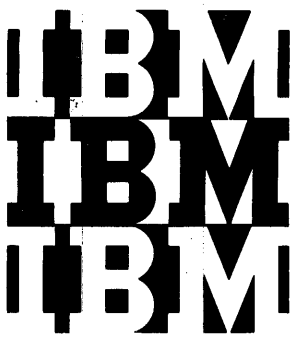
File No. S360/1130-25(OS/DMS)
Form Y27-7161-0



Program Logic

IBM System/360 Operating System and
1130 Disk Monitor System

System/360-1130 Data Transmission for FORTRAN

Program Numbers 360S-LM-542 and 1130-LM-011

This publication describes the internal logic of the processor-to-processor (PTOP) subroutines for FORTRAN IV.

The PTOP subroutines enable a FORTRAN programmer to transmit data between a program being processed in the IBM System/360 Operating System and a program being processed in the IBM 1130 Disk Monitor System Version 2. The telecommunication capabilities provided are similar to those available to an Assembler Language programmer using the binary synchronous communications services of the Basic Telecommunications Access Method in the System/360 and the Synchronous Communications Adapter subroutine for point-to-point operations in the 1130. Although designed specifically for FORTRAN IV, the PTOP routines can also be used in an Assembler Language program.

This program logic manual is directed to the IBM customer engineer who is responsible for program maintenance. It can be used to locate specific areas of the program, and it enables the reader to relate these areas to the corresponding program listings. Because program logic information is not necessary for program operation and use, distribution of this manual is restricted to persons with program-maintenance responsibilities.

First Edition (November 1968)

This publication corresponds to Release 17 of the Operating System. Significant changes or additions to the specifications contained in this publication are continually being made. When using this publication in connection with the operation of IBM equipment, check the latest SRL Newsletter for revisions or contact the local IBM branch office.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print train.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to IBM Corporation, Programming Publications, Department 637, Neighborhood Road, Kingston, New York 12401

This publication discusses the internal logic of the processor-to-processor (PTOP) subroutines for data transmission between a System/360 and an 1130 system.

The publication is organized into five sections as follows:

- Introduction, which describes the purpose, environment, and major divisions of PTOP.
- Method of Operation, which describes the basic operations including control and data flow in PTOP.
- 1130 PTOP Data Transmission Modules, which describes the 1130 data transmission routines.
- System/360 PTOP Data Transmission Modules, which describes the System/360 data transmission routines.
- System/360 Conversion Routines, which describes the data conversion routines available only in the System/360.

Appendixes at the end of this publication include a description of the PTOP control blocks, a module directory, data transmission message formats, data conversion formats, and other reference material related to the program.

USE OF THE PUBLICATION

Persons using this PLM for educational purposes or to recall the structure of the support should read the "Introduction" and "Method of Operation" sections.

Persons using this PLM to perform maintenance or modification will find the detail in the individual module descriptions and the appendixes useful.

For detailed information about programming techniques used in these routines, refer to the applicable program listings.

PREREQUISITE PUBLICATIONS

Familiarity with the following publications and their prerequisites is assumed:

IBM System/360 Operating System and 1130 Disk Monitor System: System/360-1130 Data Transmission for FORTRAN, Form C27-6937

IBM System/360 Operating System: Assembler Language, Form C28-6514

IBM System/360 Operating System: Basic Telecommunications Access Method, Form C30-2004

IBM 1130 Assembler Language, Form C26-5927

RECOMMENDED PUBLICATIONS

In addition, the following publications may be convenient for reference:

IBM System/360 Operating System: Basic Telecommunications Access Method Program Logic Manual, Form Y30-2001

IBM 1130 Subroutine Library, Form C26-5929

IBM 1130 Functional Characteristics, Form A26-5881

IBM 1130 Synchronous Communications Adapter Subroutines, Form C26-3706.

CONTENTS

FIGURES

TABLES

CHARTS

The data transmission and data conversion routines described in this publication constitute a program that provides the capability of transmitting data between an IBM System/360 and one or more remote IBM 1130 Computing Systems. These routines alter the data as necessary to reconcile differences in internal data formats in the two systems.

Within this publication, the data transmission and data conversion routines are referred to as the processor-to-processor program, or PTOP. They are primarily intended for use with the FORTRAN IV language, but may also be used with the Assembler Language. Separate sets of data transmission routines are provided for the System/360 and the 1130 system. Conversion routines are provided only for the System/360.

PURPOSE AND ENVIRONMENT OF THE PROGRAM

PTOP's main purpose is to enable the user to transmit data without detailed knowledge of telecommunications. Its functions include:

- Initializing the communication line.
- Transmitting messages over the communication line.
- Interpreting messages received over the communication line.
- Inserting proper transmission-line control characters.
- Transmitting user data.
- Rearranging data to the format used by the receiving system.
- Receiving user data.
- Checking for input/output errors and making the error condition status information available to the user.

The program is divided into two major portions: one that functions in the 1130 and one that functions in System/360. The portion that functions in the 1130 operates under control of the 1130 Disk Monitor System, Version 2. The portion that functions in System/360 operates under control of the System/360 Operating System.

Execution of the program remains the same whether it is invoked by the user from a FORTRAN program or from an Assembler Language program. In FORTRAN IV, subroutines are activated by the CALL statement. In the Assembler Language, subroutines are activated by the CALL macro instruction (System/360) and the CALL statement (1130).

1130 DISK MONITOR SYSTEM ENVIRONMENT

The synchronous communications adapter (SCA) enables the 1130 system to function as a data transmission terminal which may use either switched or non-switched transmission line facilities. The SCA sends or receives data over communication lines in binary synchronous communications (BSC) mode using the transparency feature.

In the 1130, the Disk Monitor System governs execution of the user's program which directs the use and functioning of the 1130 data transmission routines. In turn, the 1130 data transmission routines use the synchronous communications adapter subroutine (SCAT2) that controls transmission and reception of data over communication lines. SCAT2 is described in the publication IBM 1130 Synchronous Communications Adapter Subroutines, Form C26-3706.

The 1130 data transmission routines must reside in the subroutine library located in the user area or the fixed area of disk storage.

During execution of a program, SCAT2 and the Telecommunications Monitor routine (module GTP01) in the PTOP program must remain resident in storage. All other data transmission routines in the 1130 PTOP program may be designated as LOCAL records to be loaded into an overlay area as they are called.

SYSTEM/360 OPERATING SYSTEM ENVIRONMENT

To send and receive data over communication lines, the System/360 requires an IBM 2701 Data Adapter Unit or an IBM 2703 Transmission Control Unit. These devices must support binary synchronous communications with the transparency feature in a half-duplex point-to-point environment. They also must use the EBCDIC transmission code.

In System/360, the operating system control program governs execution of the

user's program which directs the use and functioning of the data transmission and data conversion routines. In turn, the System/360' data transmission routines use the Basic Telecommunications Access Method (BTAM) that controls transmission and reception of data over communication lines.

The System/360 data transmission routines, their internal support routines, and the data conversion routines are identified in System/360 by 8-character module names. These names are included in the routine descriptions and are listed in Appendix B.

Residence is determined at system generation time via the PTOP macro instruction. If the LIB parameter is specified equal to FORTLIB or if the LIB parameter is not specified, the data conversion routines and the Controller (internal support routine) reside in the FORTRAN library, while the rest of the PTOP routines reside in the link library. If the LIB parameter is specified equal to LINKLIB, all System/360 PTOP routines reside in the link library.

The Controller and the data conversion routines are processed by the linkage editor at the same time as the user's program is processed by the linkage editor. These routines remain in main storage throughout execution of the user's program. The rest of the System/360 PTOP routines are either loaded during the initialization procedure, or linked to whenever they are called.

LINE CONTROL

Standard transmission line control procedures are implemented in the PTOP program. These procedures support binary synchronous communications using the EBCDIC code employed in transparent-text mode, and are described in the IBM publication General Information -- Binary Synchronous Communications, Form A27-3004.

Control characters are required for each data transmission message to define portions of the message and to control its transmission. Appendix C describes and illustrates the formats of the messages supported in this program.

Each message block that is transmitted is checked for transmission accuracy at the receiver before the transmission continues by using CRC-16 (cyclic-redundancy check). CRC-16 is described in the IBM publication General Information -- Binary Synchronous Communications, Form A27-3004.

LINKAGE BETWEEN THE USER'S PROGRAM AND PTOP

Linkage between the user's program and the PTOP program includes transfer of control and passing of information. The following paragraphs describe how this linkage is established in the 1130 and System/360.

Interpretation of the FORTRAN CALL Statement

In a FORTRAN program, the CALL statement is interpreted to produce linkage to the entry point of the specified routine. The arguments specified by the user in the CALL statement are placed in an argument list, referred to as a parameter list. In System/360, when control is passed to the routine, register 1 contains the address of this parameter list. In the 1130 system, the parameter list follows the Branch instruction corresponding to the CALL statement.

In both systems, the parameter list contains an entry for each argument passed in the order designated by the user. Entries consist of the addresses of the actual parameter values. In System/360, each entry occupies a fullword that begins on a fullword boundary. In the 1130 system, each entry occupies one 16-bit word except when the 1130 program calls GTNIT and specifies the name of an asynchronous routine. This exception is shown in the detailed description of the 1130 GTNIT routine.

Figure 1 shows typical parameter lists in both systems after they have been expanded to contain the user-specified arguments. Interpretations of an 1130 call to GTWRT and a System/360 call to GTRED are illustrated. Following the call is the instruction(s) transferring control to the appropriate routine. Next, the parameter list is shown together with the information to which each parameter points. Note that register 1 points to the parameter list in System/360.

Interpretation of the Assembler Language CALL

The CALL statement in the 1130 Assembler Language takes the form of an indirect (two-word) BSI (Branch instruction), after the core image is constructed. Through the transfer vector table, control is passed to the entry point named in the CALL statement and the corresponding ENT (Define Subroutine Entry Point) or ISS (Define Interrupt Service Entry Point) statement. When the BSI is executed, the address of the first word following it is placed in the entry point location, and control is transferred to the first word following the entry

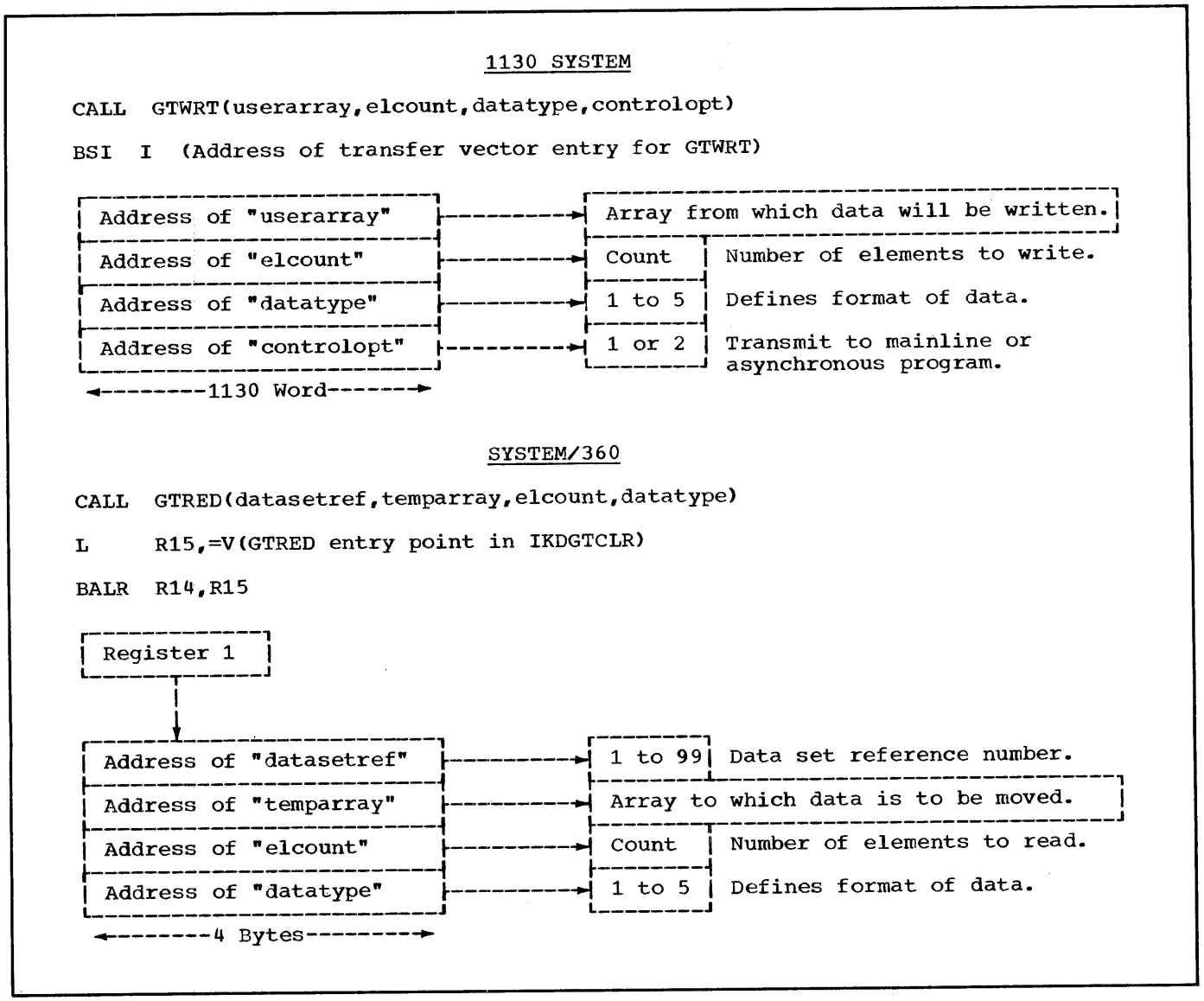


Figure 1. Typical Parameter Lists in the 1130 System and System/360

point. The user's parameter list must have been defined following the CALL statement.

The CALL macro instruction in the System/360 Assembler Language branches and links to the entry point named in the CALL macro instruction and the address parameters are expanded to form the parameter list. When control is passed to the PTOP program, register 1 contains the address of the parameter list. However, the Controller passes the address of the parameter list to the desired subroutine in a different register.

Contents of the parameter lists in both systems are the same as those shown in Figure 1.

MAJOR DIVISIONS OF THE PROGRAM

As mentioned previously, the PTOP program is divided into two portions: one that functions in the 1130 system and one that functions in the System/360. Data transmission facilities are basically the same in both portions. A set of routines is provided in each system to establish communication links between an 1130 problem program and a System/360 problem program, to transmit data between the programs, to receive transmitted data, to end the communication, and to reconcile differences in internal data formats in the two systems. Storage areas (called control blocks) are established in both systems to facilitate this communication.

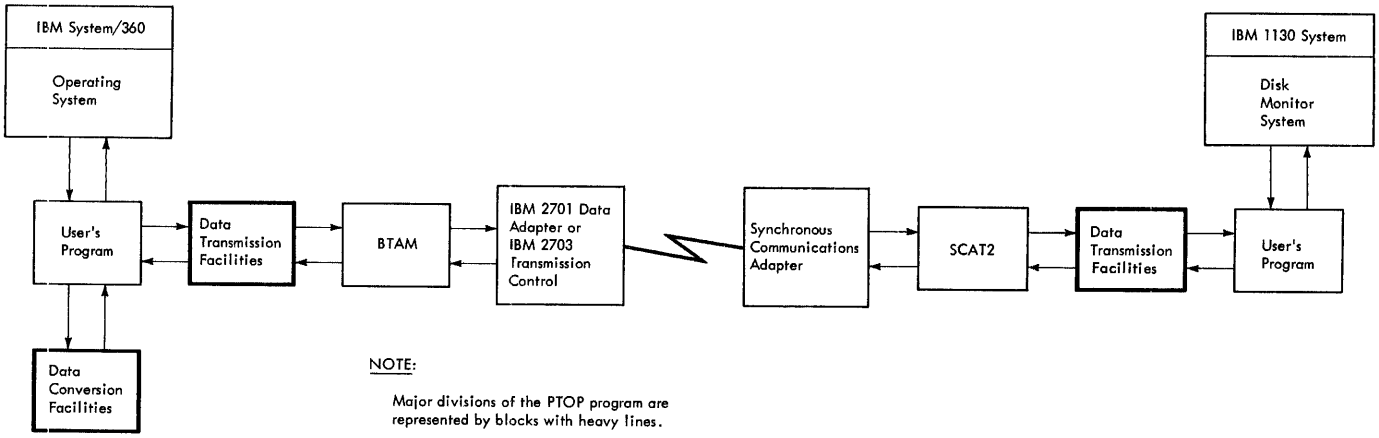


Figure 2. Relationship Among Major Divisions of the Program

Data conversion routines are provided in the System/360 to invert and reformat the FORTRAN arrays. Separate conversion routines are not available in the 1130.

Major divisions of the PTOP program are shown in Figure 2. The figure depicts the relationship among the user programs, data transmission routines, data conversion routines, the System/360 Operating System, and the 1130 Disk Monitor System. Note that the actual transmission of data is done through use of BTAM in System/360 and the synchronous communications adapter subroutine called SCAT2 in the 1130 system.

TERMINOLOGY USED IN THIS PUBLICATION

Terminology used to describe actual data transmission varies with the system (1130 or System/360) being described. Thus, several synonomous terms are used to define similar functions. Terms with related meanings are as follows:

- Send, transmit, or write refers to the sending of data transmission messages (output) to the other system.
- Receive, or read refers to the reception of data transmission messages (input) sent by the other system.

The 1130 PTOP program performs its data transmission by calling SCAT2 and specifying the desired function. Functions used by the 1130 PTOP program are:

- Transmit Initial Transparent Text followed by Transmit End.
- Transmit End (Do Not Close).
- Receive Initial.
- Receive Continue.
- Receive Repeat.
- Close.

The System/360 PTOP program performs its data transmission by passing control to BTAM through the use of macro instructions. Functions used by the System/360 PTOP program are:

- WRITE Initial Transparent with Reset (TIXR).
- WRITE Reset (TR).
- READ Initial (TI).
- READ Continue (TT).
- RESETPL (to reset the communication line).

These functions are mentioned throughout this publication.

FORMAT OF ROUTINE DESCRIPTIONS

Detailed descriptions of the PTOP modules are contained in the following sections:

- "1130 PTOP Data Transmission Modules."

- "System/360 PTOP Data Transmission Modules."
- "System/360 Conversion Routines."

Each description contains the following information:

- Module name (identification) for each routine is included in the heading for that particular routine.
- Chart identification(s), where flowcharts are included for the module. Flowcharts for modules appear at the end of their respective sections.
- Function, which states the purpose of the module.
- Entry, which identifies the point where the module is entered.
- Exit, which states where the module passes or returns control.
- Input, which defines the information passed to the module.
- Output, which defines the information passed from, or made available by, the module.
- Operation, which describes the manner in which the module performs its function(s).

METHOD OF OPERATION

Data transmission between an 1130 system and a System/360 follows a distinct path through the following basic operations:

- Initializing communication.
- Transmitting and receiving data including:
  (1) Processing input/output interruptions;
  (2) Processing input/output errors; and
  (3) Testing input/output status.
- Converting data.
- Monitoring asynchronous routine processing.
- Terminating communication.

This section analyzes each of these operations to describe the control and data flow among the PTOP routines. The section is divided into three major subsections:

- Overall Control and Data Flow, which briefly describes the manner in which the operations are performed in both systems.
- 1130 Control and Data Flow, which builds on the preceding subsection, adding specific details to complete the description of control and data flow among the 1130 PTOP routines.
- System/360 Control and Data Flow, which builds on the first subsection, adding specific details to complete the description of control and data flow among the System/360 PTOP routines.

OVERALL CONTROL AND DATA FLOW

This section briefly describes the way in which the basic PTOP operations are performed in both the 1130 Disk Monitor System and in the System/360 Operating System. Additional details on each of the operations are contained under appropriate headings in the subsections entitled "1130 Control and Data Flow" and "System/360 Control and Data Flow."

Initializing communication is necessary before data transmission can be attempted. The process is invoked when GTNIT is called in the user's program. It involves establishing links through which the PTOP routines can communicate with each other, with the user's program, with the 1130 Disk Monitor System, and with the System/360 Operating System; and involves placing the systems in ready-to-read status.

System ready-to-read status is established by initiating a Receive or Read Initial procedure successfully, thus conditioning the PTOP system to accept data that comes across the transmission lines.

During initialization, the 1130 system always transmits the initialization message to the System/360. Acknowledgment of the message by the System/360 indicates that communication links are properly established in both systems and that the initialization process can be completed.

PTOP communication areas established during initialization are:

- 1130 telecommunications control block (GTCOM),
- System/360 telecommunications control block (GTCB), and
- System/360 unit assignment table (module IKDUATBL).

Through these areas, the PTOP routines obtain information and pass information to other PTOP routines. The format and contents of the GTCOM, the GTCB, and the unit assignment table are illustrated and described in Appendix A.

Transmitting and receiving data is performed according to user-specified parameters in calls to GTWRT and GTRED. GTWRT begins procedures for transmitting data; it starts sending the appropriate data transmission message to the other system. GTRED begins procedures for receiving data; it places the user's program in ready-to-read status thus preparing it to accept data.

Unlike a system ready-to-read, to place the user's program in ready-to-read status, a ready-to-read message is first sent to the other system to indicate that the user is ready to read. Then the Receive or Read Initial procedure is initiated.

Actual transmission of data is controlled by the synchronous communications adapter subroutine (SCAT2) in the 1130 Disk Monitor System and by the Basic Telecom-

munications Access Method (BTAM) in the System/360 Operating System.

Each operation requested by the user goes through various stages during data transmission before the operation is completed. That is, an operation may be: queued but not started; started but not completed; completed satisfactorily; or completed with error condition. During the various stages of data transmission, PTOP routines place an appropriate code in the GTCOM (for the 1130 program) and the GTCB (for the System/360 program) to indicate the status of the last requested operation.

Processing input/output interruptions includes necessary control program processing functions followed by PTOP processing functions for controlling the use of the communication line. The PTOP routines base their actions on the transmission status (data transmission stage) of the interrupted program. This transmission status is reflected by information found in the PTOP control blocks.

An input/output interruption can occur while an input/output operation is in progress as well as when the operation has been completed. Overlapped input/output capability is supported by the PTOP routines.

Processing input/output errors detected as a result of receiving an interruption is performed by SCAT2 and the Telecommunications Monitor routine in the 1130 system and by BTAM and the operating system in System/360.

Input/output errors detected during data transmission are due to transmission inaccuracy (indicated when the CRC-16 character transmitted differs from the CRC-16 character accumulated by the receiver), negative acknowledgment sent by the receiving system, timeout occurrences, contention on the transmission line, and equipment failure. Input/output errors are retried according to standard error recovery procedures in both systems. Their detection and handling are transparent to the user.

PTOP data transmission routines determine the action to be taken after standard error recovery procedures have been applied by the detecting routines. Unrecoverable input/output errors, incorrect length errors, and both systems attempting to read at the same time cause PTOP to post appropriate status codes in the control blocks for the benefit of the user. In the case of contention, the System/360 write operation is delayed so that the 1130 write operation can take place first.

Testing input/output status consists of making information about the user's most recent read or write operation available to his program. This function is performed only when it is invoked by the user via a call to the GTCLT subroutine. When requested, the status information is obtained from the 1130 GTCOM or the System/360 GTCB and placed in the status code variable identified by the user's program.

Converting data involves changing data arrays to resolve the differences in the FORTRAN data structure between the 1130 system and System/360. It is performed only upon requests from the user's System/360 program by calls to the data conversion subroutines. These subroutines are not available in the 1130 program. Therefore, they should be used prior to transmitting data to the 1130 system and after data is received from the 1130 system.

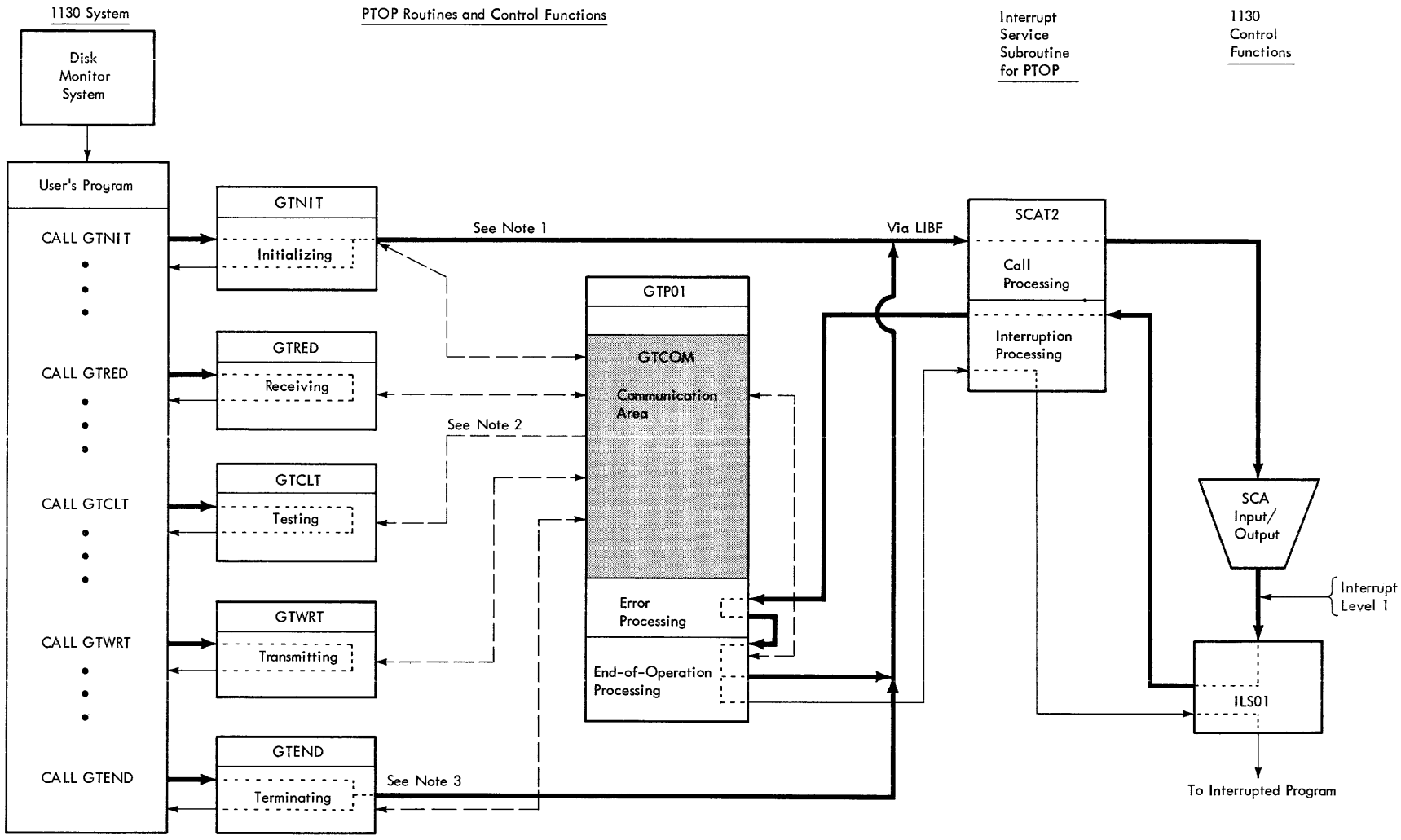
Monitoring asynchronous routine processing involves informing both systems that the user requested an asynchronous routine to be invoked. Normally, the user's mainline programs in the two systems operate synchronously; that is, the operation of one program depends on developments in the other. A request to invoke a user-defined SUBROUTINE subprogram interrupts this synchronization. Hence, this subprogram is called an asynchronous routine.

The system from which the request is made sends a message to the other system. When received, this message triggers the appropriate action to be taken by the PTOP routines so that the asynchronous routine is invoked without disturbing the current transmission status of the mainline program in that system. This involves saving the mainline program information reflecting the transmission status so that asynchronous routine processing does not destroy that information. In 1130, it is saved within the GTCOM. In System/360, it is placed in a special save area created when needed.

Terminating communication involves deactivating the data transmission program within the requesting system. The process may be requested from a user's program in either system via a call to GTEND. To reestablish that communication link, the program that requested the termination must reinitialize the system via a call to GTNIT.

1130 CONTROL AND DATA FLOW

Figure 3 shows generalized control and data flow within the 1130 system. Subroutine (striped) blocks represent the PTOP modules described in this publication. Solid lines and arrows show control flow



NOTES:

1. GTNIT waits for completion of the initialization input/output operation indicated in the GTCOM by GTP01.
2. GTCLT may wait (at the user's request) for completion of an input/output operation indicated in the GTCOM by GTP01.
3. GTEND waits for completion of the input/output operation when the terminate communication link message is sent by GTP01; GTEND then calls SCAT2 to perform the Close operation.

Figure 3. Generalized Control and Data Flow Within 1130 System

from one module to another. Dashed lines and arrows show data flow which consists of information passed between the modules and the GTCOM. For data flow, arrows at both ends of the dashed line indicate the module may derive information from as well as contribute information to the GTCOM. The following paragraphs describe the control and data flow as shown in Figure 3.

Control Flow

Control flow takes place as follows:

- From the user's program, control passes to the appropriate data transmission routine. In 1130, there are five routines corresponding to each of the subroutines that can be called.

- The data transmission routines refer to and place information in the GTCOM (located in module GTP01) as indicated in Figure 3. Except for GTNIT and GTEND, these routines perform their functions completely through the use of the GTCOM, and return control to the user's program.

- GTNIT passes control directly to the call processor in SCAT2 to perform part of the initialization operation, and waits for completion of the operation. (GTCLT may wait for completion of an operation, when requested to do so by the user.)

- GTEND takes action to terminate communication by using the GTCOM. GTEND waits for completion of the operation, and passes control to SCAT2 to close synchronous communications adapter (SCA) operations before returning control to the user's program.

- SCAT2 transmits and receives messages via the SCA. The ensuing interruption causes control to be passed to the Interrupt Level Subroutine ILS01 to start processing the interruption. From there, control passes to SCAT2 which continues processing the interruption.

- At the end of an operation, SCAT2 passes control to GTP01 for further interruption processing which includes input/output error processing, if necessary, and/or end-of-operation processing. GTP01 does not perform error processing and end-of-operation processing at the same time. Entry to the routine is always through the error processing portion, but if no errors are indicated, GTP01 immediately passes control to the end-of-operation processing portion. (For information on error handling at the end of an operation, refer to "Processing Input/Output Errors" subsequently found in this section.)

- GTP01 then starts the next operation through the call processor in SCAT2, and finally returns control to the interruption processing portion of SCAT2, which in turn passes control to ILS01 to complete its level processing function.

- ILS01 returns control to the interrupted program at the next instruction following the point of interruption. This may be GTNIT, GTCLT, or GTEND if one of them is waiting for completion of an operation.

Data Flow

Data flow between the 1130 PTOP routines and the GTCOM as shown in Figure 3 is described in detail in the section "1130 PTOP Data Transmission Modules." In addition, Appendix A contains a summary table showing how the 1130 PTOP routines use the GTCOM fields.

Additional details on each of the operations in the 1130 system are contained under the appropriate headings in the following paragraphs.

INITIALIZING COMMUNICATION (1130)

Linkage between the 1130 system and the System/360 is established whenever the initialization process is completed in both systems.

The initialization process in 1130 is invoked via a call to the GTNIT subroutine in the user's program. This call activates the GTNIT routine which performs the following:

- Calls SCAT2 to transmit the initialization message to the System/360. This process results in the 1130 PTOP system being placed in a ready-to-read status.

- Effects communication and control among the 1130 PTOP routines by initializing addresses and indicators in the GTCOM.

Figure 4 traces the control flow during the 1130 initialization process. Detailed information is contained in the paragraphs that follow.

Using SCAT2

To transmit an initialization message to the System/360, GTNIT calls the SCAT2 subroutine, and waits for SCAT2 to complete

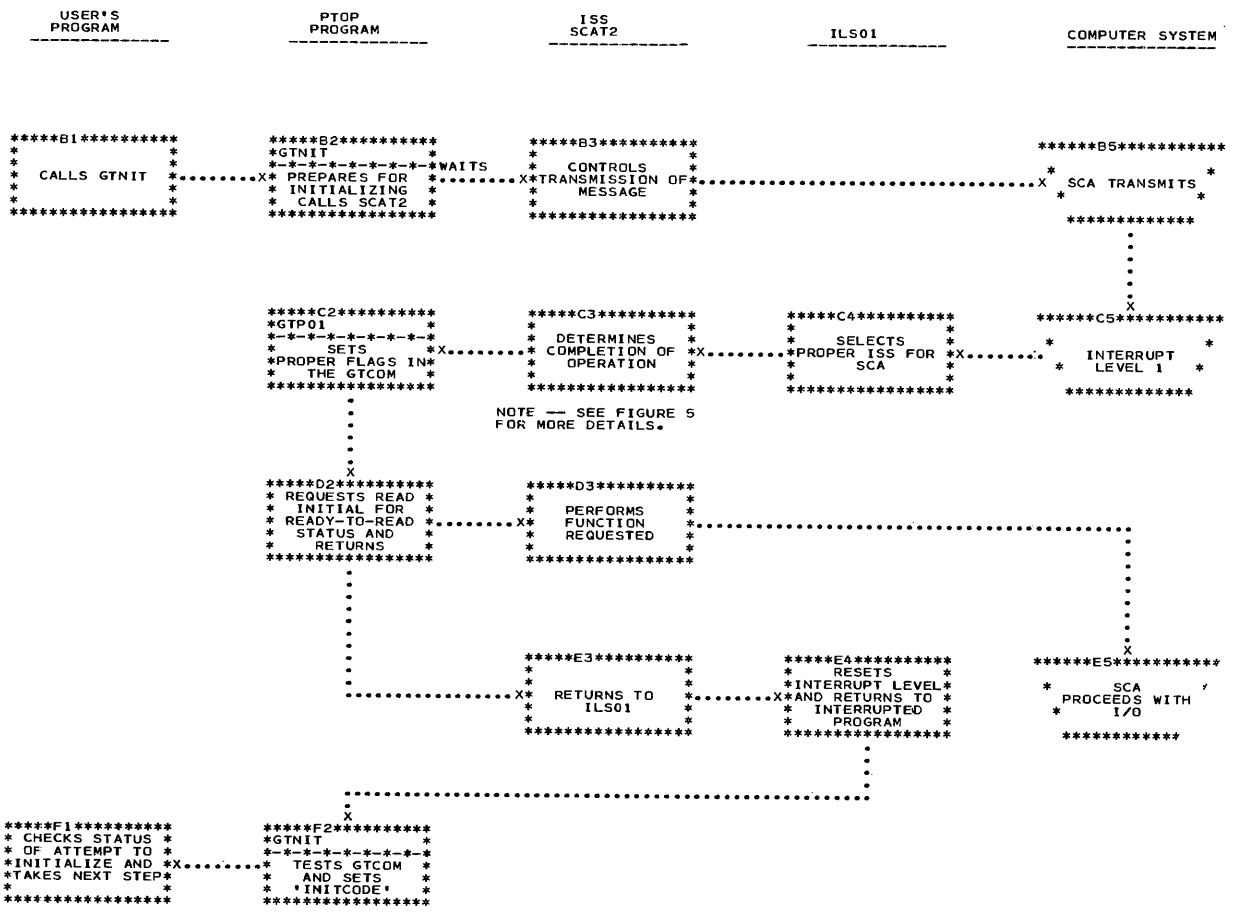


Figure 4. 1130 System Control Flow for PTOP Initialization Process

the transmission. SCAT2 controls the 1130 Synchronous Communications Adapter and checks for errors in the data being transmitted and received. When the transmission operation is completed, GTP01 receives control through the normal interruption processing and determines if the communication line is initialized. Appropriate information is placed in the GTCOM, system ready-to-read status is established, and control is passed back through SCAT2 and ILS01 until it eventually returns to GTNIT.

Effecting Communication and Control

Communication and control among the PTOP routines is accomplished in the 1130 system by use of the GTCOM contained in the GTP01 module. When brought into main storage, the GTCOM contains no information. Its fields are completed and updated during processing by the 1130 PTOP routines.

The GTCOM is described in detail in Appendix A. Included in Appendix A is a summary table showing how each 1130 data transmission routine may affect the information located in the GTCOM fields.

TRANSMITTING AND RECEIVING DATA (1130)

Calls to GTWRT and GTRED start the transmission and reception of user data by invoking the Write Data (GTWRT) and Read Data (GTRED) routines respectively.

Figure 5 traces the control flow for a read/write operation in the 1130 system along with the input/output interruption processing, input/output error processing, and input/output status testing. It depicts the functions involved in transmitting and receiving data which are described in the paragraphs that follow.

Even though the initialization process conditioned the systems to receive data, no actual transfer of data between user programs can occur unless the receiving program is conditioned to accept it (i.e., placed in a ready-to-read status). In the 1130 system, GTRED sets indicators and

completes appropriate fields in the GTCOM to indicate that the user ready-to-read message is to be transmitted and the user's program is to be placed in ready-to-read status.

At interruption processing time, GTP01 first sends the message via SCAT2 to the System/360. Next, it prepares to read the data into the user's read buffer. To do this, GTP01 calls SCAT2 specifying the Receive Initial function. (The Receive Initial completes the process placing the user's program in ready-to-read status.) When data is received, GTP01 moves it to the user-specified working array omitting all control characters for use by the problem program.

The transmission of data is started in the 1130 by GTWRT, which sets indicators and completes appropriate fields in the GTCOM.

Information placed in the GTCOM results in GTP01 preparing either the data message or the asynchronous routine request message for transmission when it receives control at interruption processing time. GTP01 calls SCAT2 to transmit the message.

To prevent transmitting irrelevant information over the communication lines, 1130 integer data is packed and unpacked by GTP01. Before calling SCAT2 to transmit integer data, GTP01 packs the data contained in the user's array by placing only the first word (16 bits) of data from each element into the user's write buffer for transmission; the remainder of the element is irrelevant and, therefore, is ignored. After receiving integer data, GTP01 unpacks the data by placing each word of data appropriately within each element as it moves the data from the read buffer into the user's array.

In the 1130, GTP01 inserts the necessary control characters at the beginning of every message transmitted in transparent-text mode. SCAT2 inserts control characters at the end of every message transmitted.

All data transmission messages including appropriate control characters and header information are illustrated and described in Appendix C.

Processing Input/Output Interruptions

When an interruption occurs in the 1130, control is passed to the Interrupt Level Subroutine ILS01. This subroutine is the 1130 system module designed to service the input/output interruption defined as interrupt level 1 that applies to the SCA. ILS01 performs level processing for the interruption which includes selecting the correct device processing routine, clearing the level when interruption processing is complete, and returning control to the user's program.

The device processing routine for SCA interruptions is the interruption processing portion of SCAT2. An interruption occurs after each character (8 bits of information) is transmitted or received by the SCA.

If the count (number of characters) indicates there is more data to be handled, SCAT2 packs or unpacks a data character, sets or resets storage locations in preparation for further reception or transmission, and returns control to ILS01.

If the interruption indicated the end of an operation, SCAT2 passes control to PTOP's interruption processing routine (GTP01) with the accumulator set to a value that indicates whether or not an input/output error was detected. Zero indicates to GTP01 that error handling functions are not required (completion was satisfactory).

When GTP01 determines it was a satisfactory completion, end-of-operation processing takes place. This processing interprets the information contained in the messages received and in various fields of the GTCOM, and determines which operation had just been performed and which operation should be performed next. GTP01 prepares the buffers as necessary and requests the appropriate function of SCAT2's call processing routine for the next input/output operation.

GTP01 then returns control to SCAT2's interruption processing routine, and SCAT2 returns control to ILS01. ILS01 then clears the level of interruption in preparation for the next operation, and returns control to the first word of the next instruction following the point of interruption in the user's program.

Processing Input/Output Errors

In 1130, input/output errors are detected by SCAT2 upon completion of an input/output operation.

SCAT2 retries certain input/output operations up to seven additional times before placing the appropriate value in the accumulator and passing control to GTP01 for error handling functions. GTP01 then determines from the accumulator setting which input/output error occurred, and processes the error condition by placing the appropriate information in the GTCOM for future reference. GTP01 also determines whether the operation should be restarted.

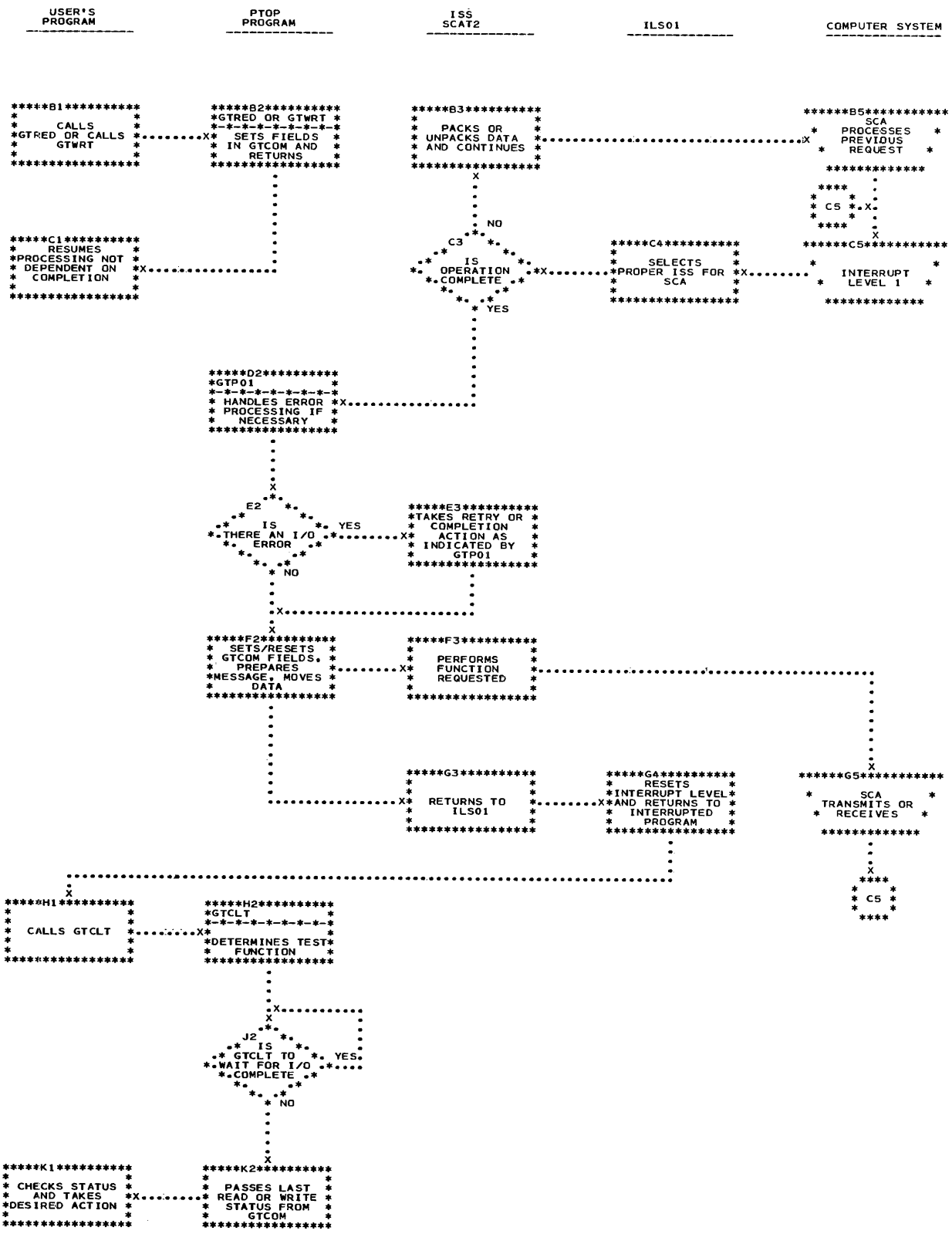


Figure 5. 1130 System Control Flow for PTOP Read/Write Operation with Test Function

If so, it returns to SCAT2 with a non-zero value set in the accumulator that indicates the operation is to be retried. Otherwise, GTP01 returns to SCAT2 with zero in the accumulator that indicates SCAT2 is to act as if a normal completion occurred. SCAT2 performs retry or completion functions and goes back to GTP01 for further error processing or for end-of-operation processing. After end-of-operation processing, GTP01 returns to SCAT2, which then returns to ILS01.

For more information about 1130 error handling procedures, refer to the publication IBM 1130 Subroutine Library, Form C26-5929.

In case of an unrecoverable input/output error that occurred during a user's read or write data operation, GTP01 posts the completion status in the GTCOM indicating that the operation is complete with error.

Testing Input/Output Status

When an input/output operation is requested by the user, GTRED or GTWRT places the appropriate status code in the GTCOM. When the input/output operation is started or completed, GTP01 places the appropriate status code for that operation in the GTCOM overlaying any other status code that was posted previously. The final status code posted by GTP01 for completion of an operation indicates whether the last read or write operation was completed successfully or with errors.

To obtain this status code, the user's program must call the GTCLT subroutine. GTCLT places the status code from the GTCOM into the "statcode" variable specified in the call to the GTCLT subroutine. When requested to do so by the "testcode" argument, GTCLT will wait until the current operation is completed before placing the status code in the variable for the user. However, GTCLT will not wait if a condition exists that warrants immediate return.

CONVERTING DATA (1130)

Data conversion is accomplished in System/360 only. As mentioned previously, it is performed only upon requests from the user's program. These requests invoke the data conversion routines in System/360 so that the data and arrays are transformed either into System/360 format (after being read) or into 1130 system format (before being written).

The data transmission routines do not check any data formats during the processing. All user-defined data is handled as though it were always transmitted in 1130 format over the transmission line. If the user fails to make use of the data conversion routines, unpredictable results may occur.

MONITORING ASYNCHRONOUS ROUTINE PROCESSING (1130)

When a request for the System/360 asynchronous routine is encountered in an 1130 call to GTWRT, the appropriate fields in the GTCOM are completed by GTWRT so that the asynchronous routine request message is sent by GTP01 to notify the System/360. Then, whenever the asynchronous routine is ended, the System/360 transmits the appropriate message to notify the 1130. In both cases, the receiving system communicates this information to its PTOP program through the appropriate fields in the GTCOM and the GTCB.

Asynchronous routine processing in System/360 does not change the processing flow in the 1130 PTOP program. The 1130 program continues performing along the same basic path shown in Figure 3. However, a change does occur in the 1130 PTOP program processing when the 1130 asynchronous routine is requested.

Figure 6 shows the control and data flow for invoking the 1130 asynchronous routine, starting with the request from the System/360. ILS01 and SCAT2 take the same actions they would take for any other data transmission function. PTOP's actions are explained in the following paragraphs.

When an asynchronous routine request message is received by the 1130, GTP01 indicates in the GTCOM that the 1130 asynchronous routine is to be invoked. GTP01 will not honor any ready-to-read message, data message, or user's read operation that may be pending. Thus, the transmission status of the user's mainline program remains unchanged once the asynchronous routine has been requested. GTP01 returns control to SCAT2, and eventually control returns to the user's mainline program. The next data transmission routine (except GTNIT) that obtains control in the 1130 indirectly invokes the asynchronous routine.

When the data transmission routine determines that the asynchronous routine is scheduled, it saves the entry status of the current call so that processing of the requested function can be continued later. It identifies itself as the caller by placing a predetermined value in an index register, and passes control to the secondary entry point named GTP02 in the Telecommunications Monitor routine.

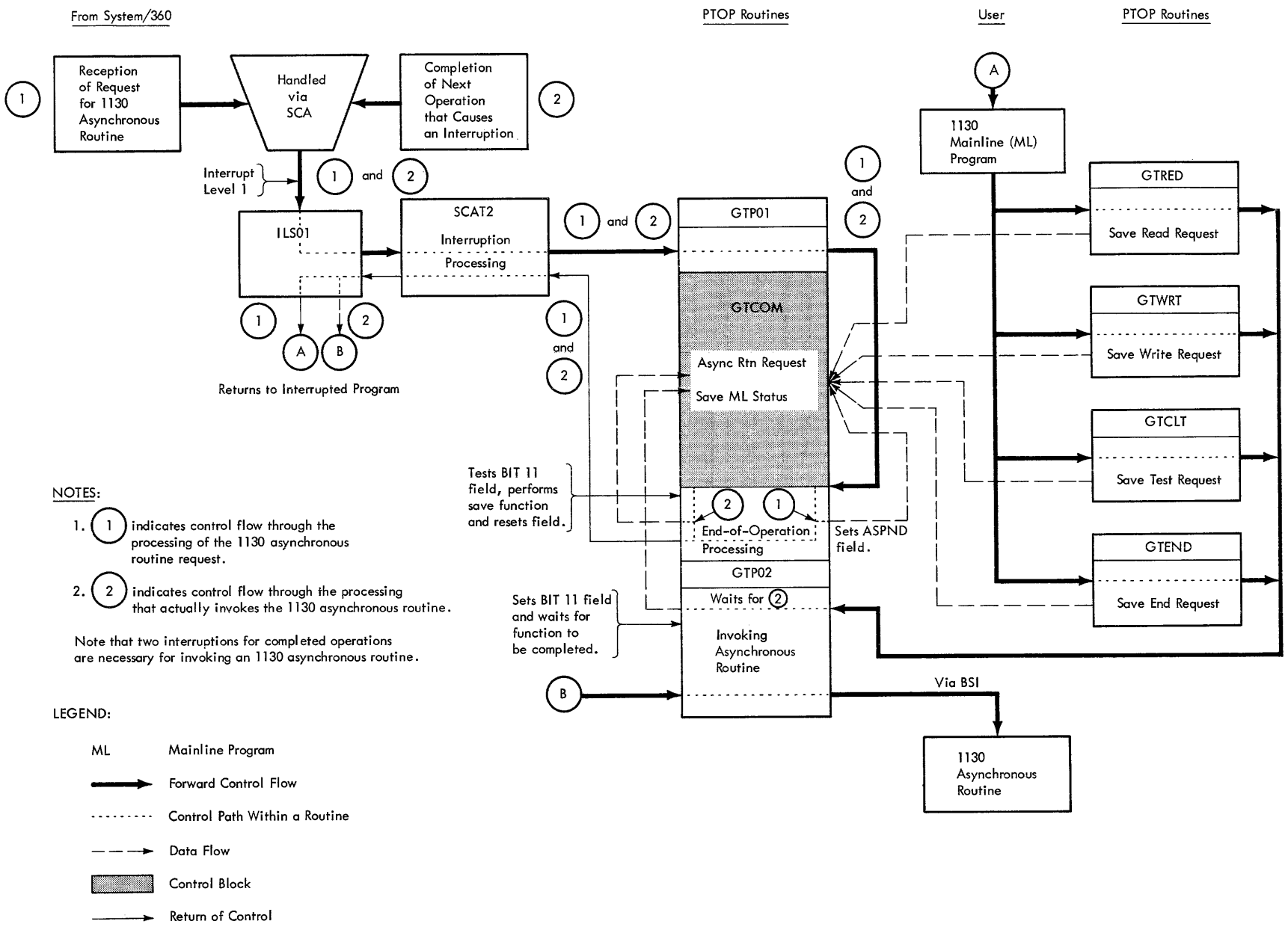


Figure 6. Control and Data Flow for Invoking the 1130 Asynchronous Routine

GTP02 indicates that the mainline program transmission status must be saved by GTP01 at interruption processing time before processing the next operation. GTP02 waits for the transmission status to be saved in the special GTCOM area, and then calls the asynchronous routine.

The asynchronous routine may use the data transmission facilities during its processing. When the asynchronous routine is executing, control flow follows the same path taken during mainline program processing as shown previously in Figure 3. When execution is complete, the asynchronous routine must return control to the first word of the executable instruction following the point at which control was passed to it. This enables PTOP to notify the System/360 and to resume processing the mainline program requests.

At interruption processing time, GTP01 restores the mainline transmission status in the GTCOM. GTP01 then sends the asynchronous routine ended message to the System/360 via SCAT2.

The invoking routine receives control after interruption processing is complete. When it determines that the asynchronous routine ended message has been sent, it returns control to the appropriate data transmission routine. Since another routine may overlay the original caller control is returned by a call to the secondary entry point in the data transmission routine determined by the value passed in the index register. The data transmission routine restores the call status saved for the mainline program, and resumes processing of the call before it returns control to the user's mainline program.

TERMINATING COMMUNICATION (1130)

The End Communication routine (GTEND) becomes active when the GTEND subroutine is called in the user's 1130 program. GTEND begins the transmission of the terminate communication link message to the System/360 by setting appropriate fields in the GTCOM. GTP01 sends the message to the System/360 via SCAT2. GTEND waits for completion of the transmission operation, and then closes the communication line in the 1130.

This communication link can be reestablished by reinitializing the 1130 system while the System/360 is maintaining its communication link. After the communication link is terminated by a call to GTEND in either system the 1130 PTOP program cannot communicate with the System/360. The 1130 PTOP program, however, remains intact (as long as there is no outside intervention). Provided the System/360 remains initialized, another call to GTNIT in the 1130 user's program enables the user's program to resume calling the PTOP data transmission routines just as if the communication link had not been broken.

The communication link between the systems can also be broken by a call to the GTEND subroutine in the user's System/360 mainline program. This call causes the System/360 to transmit the terminate communication link message to the 1130. Upon receiving this message, GTP01 indicates in the GTCOM that the PTOP program will not perform any more input/output operations until the communication link is reestablished.

GTP01 continues to send the initialization message at regular intervals until either (1) the System/360 program calls GTNIT and successfully recognizes that message, or (2) the 1130 program calls GTEND and terminates the 1130 communication link. If a satisfactory response is received from System/360, the communication link is reopened and processing resumes as normal.

Abnormal termination of the 1130 program occurs if a System/360 call to GTWRT specifies that the PTOP program is to transmit a program termination request message to the 1130. Upon receiving this request, the 1130 PTOP program completely relinquishes control to the Disk Monitor System; it cannot be reinitialized to communicate with the program that caused its termination.

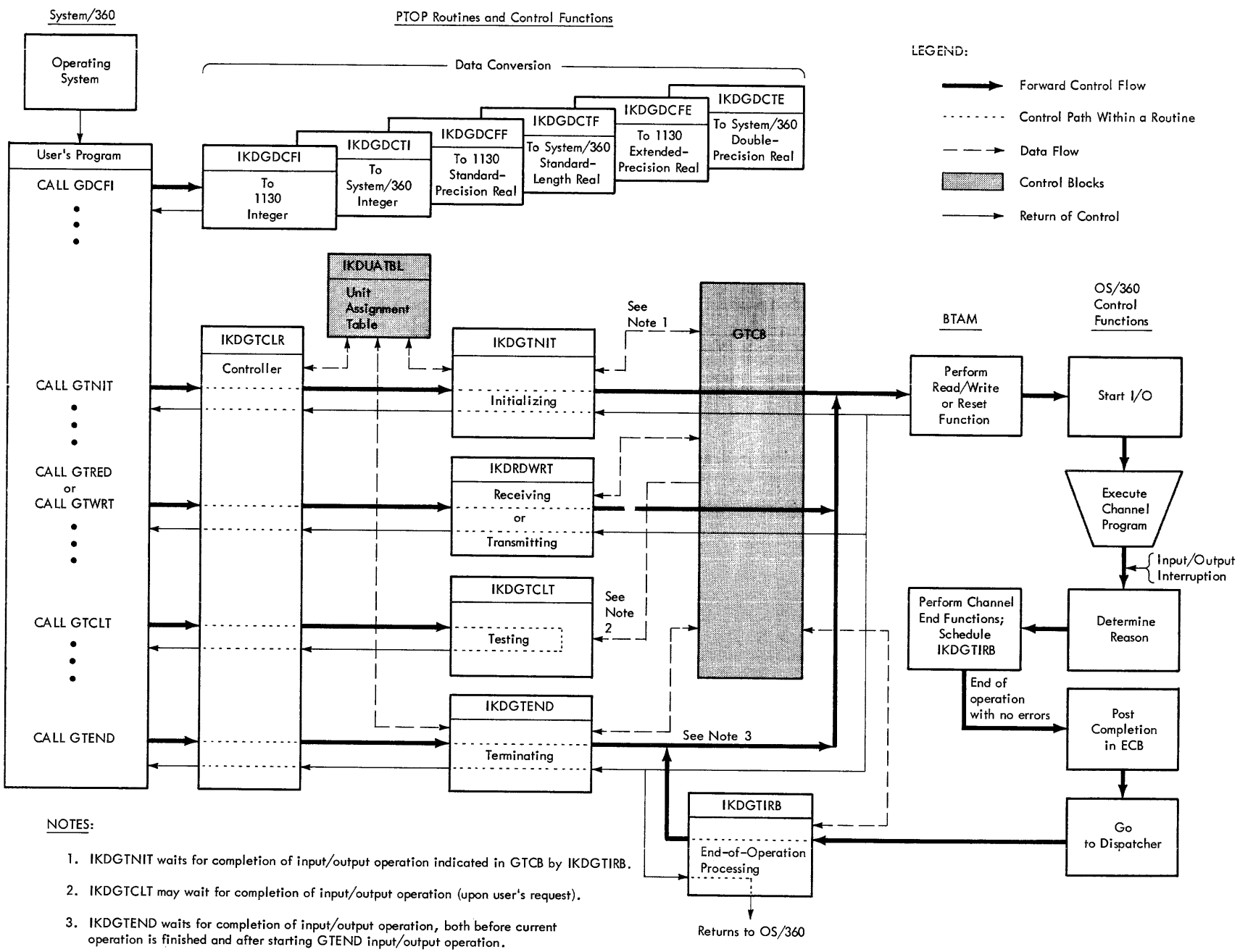
SYSTEM/360 CONTROL AND DATA FLOW

Figure 7 shows generalized control and data flow within the System/360. Subroutine (striped) blocks represent the PTOP modules described in this publication. Solid lines and arrows show control flow from one module to another. Dashed lines and arrows show data flow which consists of information passed between the modules, the GTCB, and the unit assignment table. For data flow, arrows at both ends of the dashed line indicate the module may derive information from as well as contribute information to the control block or table. The following paragraphs describe the control and data flow as shown in Figure 7.

Control Flow

Control flow takes place as follows:

- If a data conversion subroutine is called in the user's program, control passes directly to the appropriate data conversion routine. When its function is complete, the conversion routine returns control to the user's program.



NOTES:

1. IKDGTNIT waits for completion of input/output operation indicated in GTCB by IKDGTIRB.
2. IKDGTCLT may wait for completion of input/output operation (upon user's request).
3. IKDGTEND waits for completion of input/output operation, both before current operation is finished and after starting GTEND input/output operation.

Figure 7. Generalized Control and Data Flow Within System/360

- If a data transmission subroutine is called in the user's program, control first passes to the Controller (IKDGTCLR) which consults the unit assignment table (IKDUATBL) and then routes control to the appropriate data transmission routine.

- The data transmission routines manipulate information in the GTCB. IKDGTNIT and IKDGTEND also manipulate information in the unit assignment table.

- Except for IKDGTCLT, the data transmission routines may pass control to BTAM if an input/output operation is not already in progress.

- BTAM interfaces with the operating system control modules and sends or receives as requested.

- An input/output interruption causes the Input/Output Supervisor to pass control to BTAM's channel end appendage routine which performs channel end functions.

- At the end of an input/output operation, BTAM schedules the PTOP Interface Resolution routine (IKDGTIRB) to receive control.

- IKDGTIRB refers to information in the GTCB, determines the operation that was just completed, places appropriate information in the GTCB, determines the next operation to be performed, and passes control to BTAM to perform the desired operation.

- IKDGTIRB then returns control to the operating system, which returns control to the interrupted program. This may be IKDGTNIT, IKDGTCLT, or IKDGTEND if one of them is waiting for completion of an operation.

Data Flow

Data flow between the System/360 PTOP routines, IKDUATBL, and the GTCB as shown in Figure 7 is described in detail in the section "System/360 PTOP Data Transmission Modules." Appendix A contains a summary table showing how the System/360 routines use the GTCB fields.

Additional details on each of the operations in the System/360 are contained under the appropriate headings in the following paragraphs.

INITIALIZING COMMUNICATION (SYSTEM/360)

Linkage between the System/360 and the 1130 system is established whenever the initialization procedures are completed in both systems. Linkage between the user's program and the System/360 PTOP routines is described in the following paragraphs.

Establishing Linkage

In System/360, linkage is established to the data transmission routines by the Controller which is entered at the appropriate entry point from a call in the user's program. The Controller first verifies the data set reference number. It then determines whether the routine referred to is transient and should be linked to, or whether that routine is resident (loaded), in which case a branch and link instruction is used to pass control to the requested routine. The address for a resident routine is found in the unit assignment table. Linkage between the user's program and the specified routine is completed by passing the address of the user's parameter list in register 11. After execution of the data transmission routine is complete, control is returned to the user's program through the Controller.

Linkage is established to the data conversion routines directly from the user's program. These routines return control to the user's program as soon as they have completed their operation.

The System/360 initialization process is invoked via a call to the GTNIT subroutine in the user's program. This call causes the Controller (IKDGTCLR) to pass control to the Initialization routine (IKDGTNIT). The initialization process performs the following:

- Loads the resident routines, builds operating system control blocks, and establishes a timer mechanism.

- Uses BTAM to start a Read Initial procedure to receive the initialization message containing the password from the 1130 system.

- Effects communication and control among the PTOP routines by establishing the GTCB and placing information in the unit assignment table.

- Acquires storage for the user's read and write buffers. The sizes of these buffers are determined by parameters specified by the user when he calls the GTNIT subroutine.

Figure 8 traces the control flow during the initialization process. Detailed information about the functions is contained in the paragraphs that follow.

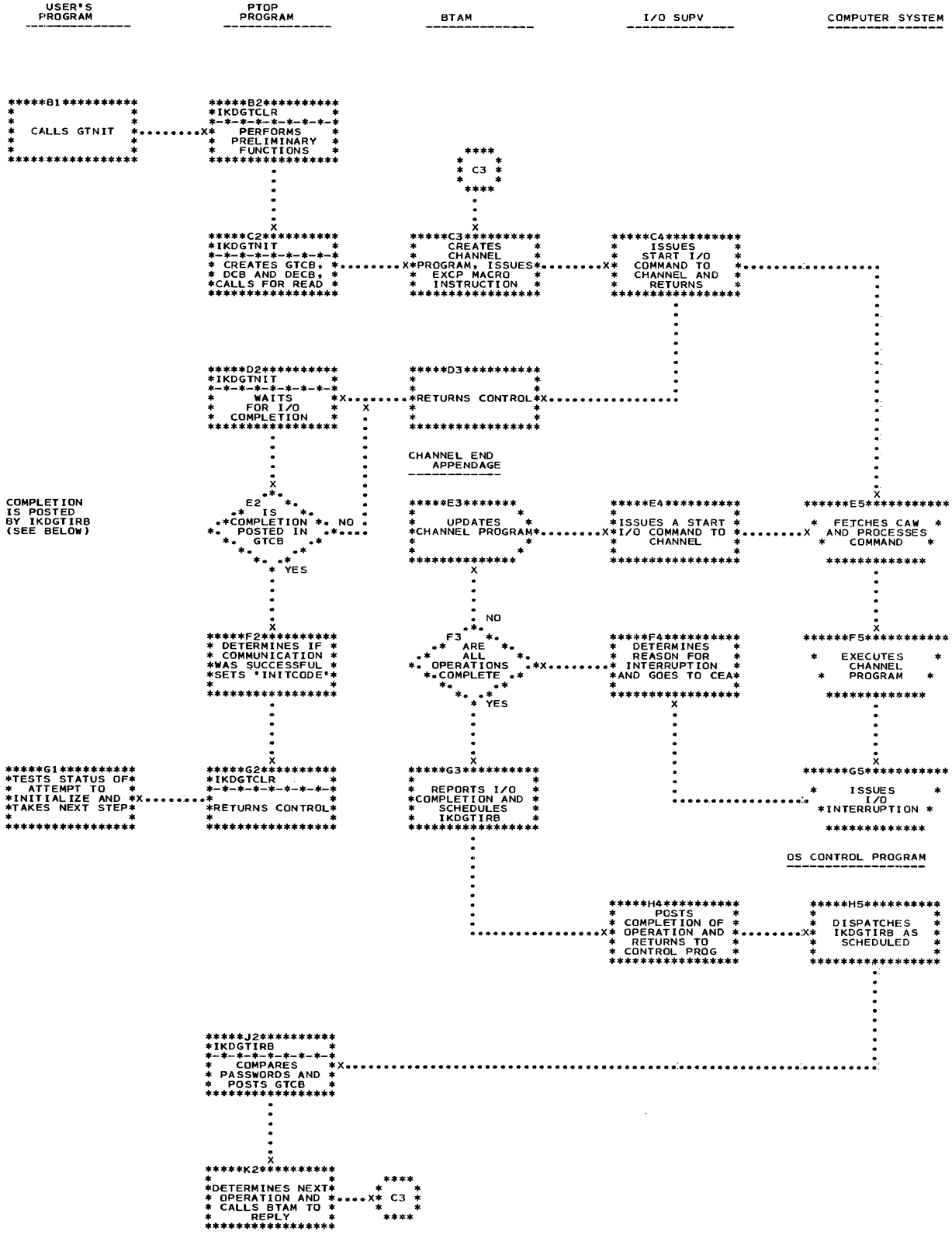


Figure 8. System/360 Control Flow for PTOP Initialization Process

Using BTAM

BTAM controls the data movement over communication lines in System/360. The System/360 PTOP routines use BTAM by creating BTAM control blocks and specifying the proper macro instructions. When these macro instructions are executed, control is passed to BTAM.

BTAM control blocks that are created and their purposes are:

- One data control block (DCB) which is needed to define a data set and thus identify a communication line group. The DCB also furnishes BTAM with the address of PTOP's Interface Resolution routine (IKDGTIRB) so that BTAM's channel end appendage can schedule the routine to be invoked asynchronously after input/output interruption processing is accomplished by the control programs.

- The data event control block (DECB) which provides communication with the BTAM read/write module.

Note: DCB and DECB formats and descriptions are contained in the publications:

IBM System/360 Operating System:

System Control Blocks, Form C28-6628

Basic Telecommunications Access Method, Form C30-2004

Basic Telecommunications Access Method Program Logic Manual, Form Y30-2001

Effecting Communication and Control

Communication and control among the PTOP routines is accomplished in the System/360 by means of the GTCB, a special save area used when the asynchronous routine is in control, and the unit assignment table.

The GTCB is used for storing information needed by the PTOP routines. It is dynamically constructed during processing by the Initialization routine (IKDGTNIT). One GTCB is created and initialized for each data set reference number. The address of each GTCB created is placed in the unit assignment table.

The special save area is used to store the transmission status of the user's mainline program when the PTOP routines are processing for an asynchronous routine. It is dynamically acquired when PTOP determines that the asynchronous routine is to be given control. Its address is placed in the GTCB associated with the mainline program.

The unit assignment table is a PTOP module that provides the Controller an additional means of controlling the execution of the user's program and ensuring that correct linkage is established to the PTOP routines in System/360. It is loaded by the Controller at initialization time and remains in storage until termination time. When it is brought into storage, the unit assignment table contains no information. Its fields are completed during PTOP program processing. It contains the addresses of the GTCBs established during the initialization process.

The unit assignment table furnishes a means of associating a data set reference number passed by the user's program with the appropriate GTCB. The data set reference number is not actually entered in the table, but is implied by the position of the GTCB address within the table. For example, if the GTCB address is located in the first word (four bytes) of the table, data set reference number 1 is implied. If the GTCB address is located in the tenth word (table +36) data set reference number 10 is implied. A maximum of 99 data set reference numbers are allowed.

Figure 9 shows relationships among the unit assignment table, the GTCB, and the asynchronous routine save area. Detailed information about each of these areas is contained in Appendix A. Also included in Appendix A is a summary table showing how each System/360 data transmission routine may affect the information located in the fields of the GTCB.

TRANSMITTING AND RECEIVING DATA (SYSTEM/360)

In System/360, calls to GTWRT and GTRED start the transmission and reception of user's data by invoking the Read and Write Data routine (IKDRDWRT).

Figure 10 traces the control flow for a read/write operation along with the input/output interruption processing and input/output status testing. It depicts the functions involved in transmitting and receiving data which are described in the paragraphs that follow.

Even though the initialization process conditioned the systems to receive data, no actual transfer of user's data occurs unless the receiving program is conditioned to accept it (i.e., placed in a ready-to-read status). A call to GTRED is necessary to condition the System/360 program to accept user's data.

When IKDRDWRT receives this request, it sends a ready-to-read message to the 1130

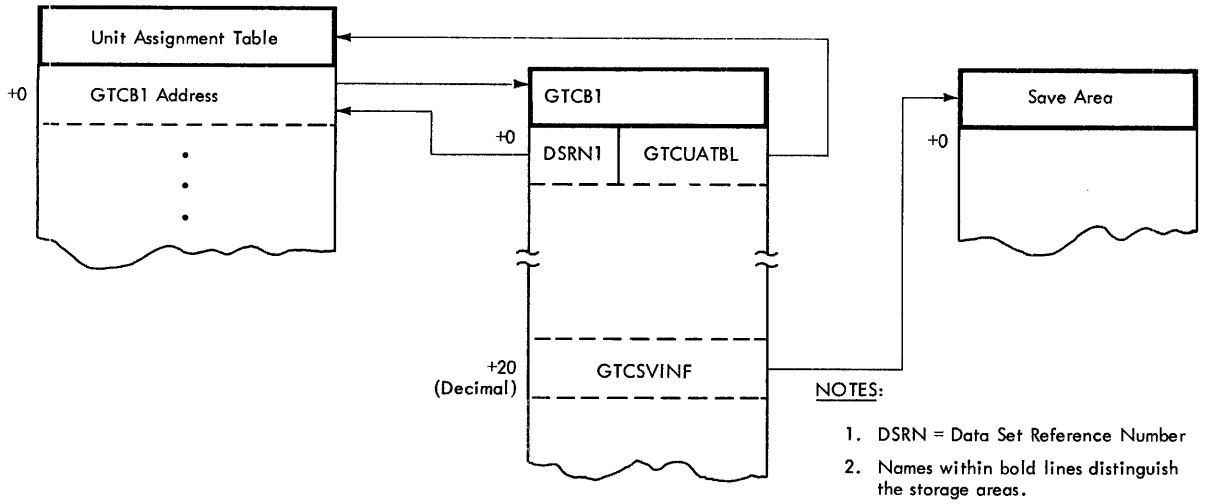


Figure 9. Relationships Among the Unit Assignment Table, the GTCB, and the Asynchronous Routine Save Area

system by using BTAM. Appropriate fields are completed in the GTCB to reflect the status of the input/output operation. Following transmission of the message, a Read Initial procedure is started to read the data into the user's read buffer. When the data is received by the System/360, then IKDGTIRB moves it to the user-specified working array omitting all control characters for the use of the problem program.

A call to GTWRT is necessary to transmit user's data. When IKDRDWRT receives this request, it determines whether the 1130 user is ready to read (has called GTRED). If so, IKDRDWRT sends a data message from the user's write buffer to the 1130 system by using BTAM. Appropriate fields are completed in the GTCB to reflect the status of the input/output operation. If the 1130 user is not ready to read, IKDRDWRT sets appropriate flags in the GTCB so that IKDGTIRB can transmit the data message at a later time (when the 1130 user has called GTRED).

If the call to GTWRT specified that the 1130 asynchronous routine is to be given control, IKDRDWRT sends the asynchronous routine request message to the 1130 system.

System/360 can also terminate the 1130 program by means of GTWRT. When the user specifes a "controlopt" argument equal to 3, IKDRDWRT sends the program termination request message to the 1130 system.

To send any message to the 1130, IKDRDWRT first attempts to halt input/output operations and, if successful, then transmits the appropriate message via BTAM. In all cases, the GTCB is updated and appropriate flags are turned on or off to inform IKDGTIRB of the transmission status of the current operation at interruption processing time. If IKDRDWRT is unable to send the message because an input/output operation is already in progress and cannot be halted, IKDGTIRB sends the message via BTAM when the current input/output operation is completed.

In System/360, the PTOP routine that moves the message to the write buffer inserts the necessary control characters at the beginning of every message transmitted. BTAM inserts control characters at the end of every message.

All data transmission messages including appropriate control characters and header information are illustrated and described in Appendix C.

Processing Input/Output Interruptions

In System/360, a channel end interruption or an abnormal end interruption causes control to be passed to the channel end/abnormal end appendage in BTAM. This appendage handles the input/output interruption, and determines whether the operation is complete.

If the operation is not complete, the appendage routine updates the channel program to continue the operation and returns to the Input/Output Supervisor.

If the operation is complete, the appendage routine schedules the execution of IKDGTIRB via the Asynchronous Exit Effector (AEE) and informs the Input/Output Supervisor that the input/output operation has been completed.

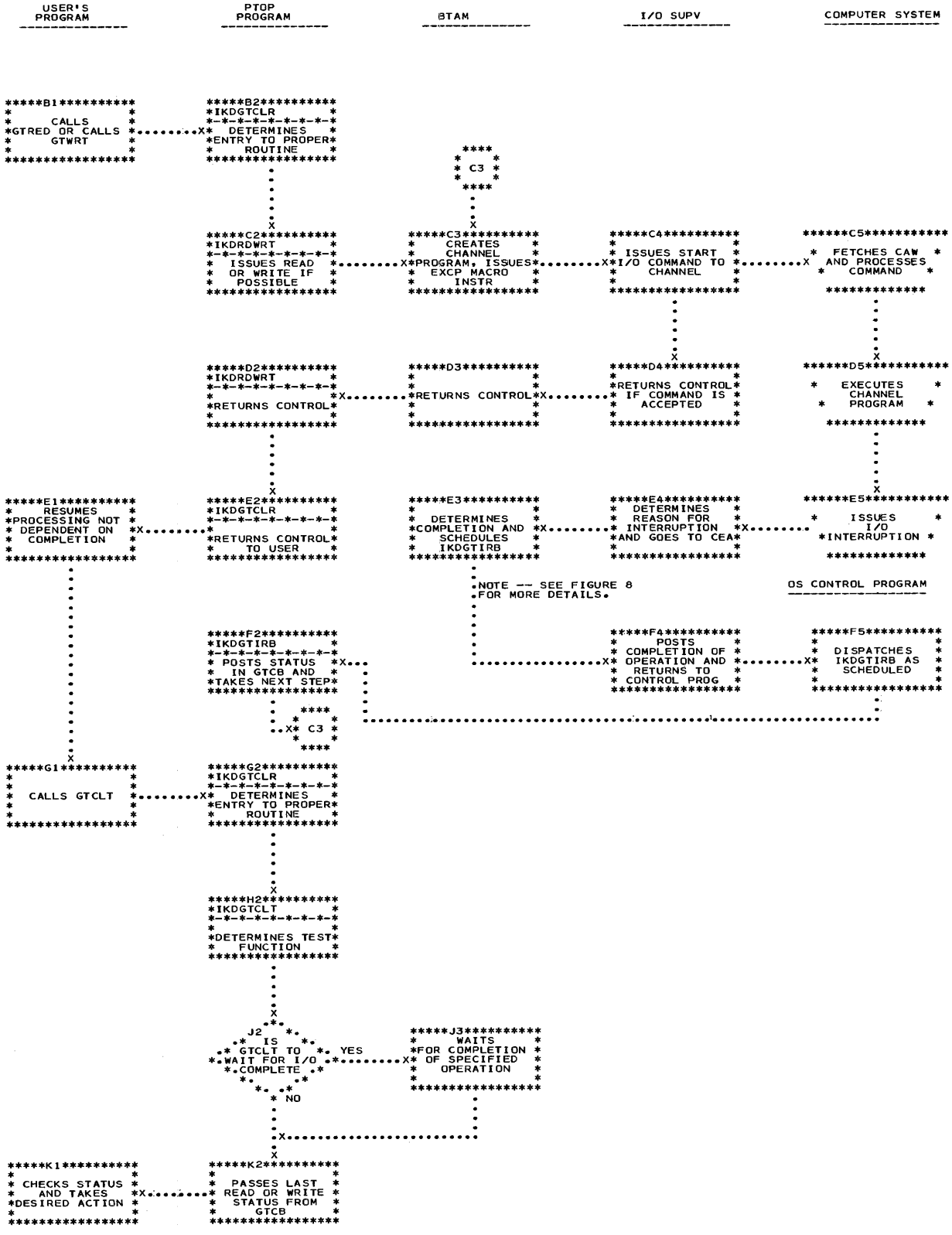


Figure 10. System/360 Control Flow for PTOP Read/Write Operation with Test Function

IKDGTIRB evaluates information in the GTCB to determine the next function it has to perform. Any outstanding input/output operation necessary to complete a user requested read or write operation is sent through BTAM. If no user's request is pending, or no user's request can be performed at this time, IKDGTIRB restores the system ready-to-read status by requesting a read operation specifying the PTOP buffer.

If the operation just completed was user-initiated, IKDGTIRB posts the completion status of the last user's read or write data operation in the GTCB for use by IKDGTCLT.

Processing Input/Output Errors

In System/360, input/output errors are detected by the computing system and indicated in the channel status word (CSW). The channel end appendage in BTAM determines if some of the operations should be retried. By setting appropriate fields in the system control blocks, the channel end appendage informs the Input/Output Supervisor that BTAM Error Recovery Procedures for BSC are to be invoked to diagnose and recover, if possible, from the errors that occurred during the telecommunications processing.

For information on basic error recovery procedures provided in BTAM, refer to the publication IBM System/360 Operating System: Basic Telecommunications Access Method, Form C30-2004. For detailed functions of the error handling routines, refer to the publication IBM System/360 Operating System: Basic Telecommunications Access Method Program Logic Manual, Form Y30-2001.

In case of an unrecoverable input/output error that occurred during a user's read or write data operation, IKDGTIRB posts the completion status in the GTCB indicating that the operation is complete with error.

Testing Input/Output Status

When an input/output operation is requested by the user, IKDRDWRT places the appropriate status code in the GTCB. As soon as the input/output operation is completed, IKDGTIRB posts a status code in the GTCB, overlaying any previous status code, which indicates to the user's program the completion status of the last read or write data operation.

To obtain this status code, the user's program must call the GTCLT subroutine which causes the Controller to pass control to the IKDGTCLT routine. IKDGTCLT places the status code from the GTCB into the "statcode" variable specified in the call to the GTCLT subroutine. When requested to do so by the "testcode" argument, IKDGTCLT waits until the current operation is completed before placing the status code in the variable. However, IKDGTCLT will not wait if a condition exists that warrants immediate return.

CONVERTING DATA (SYSTEM/360)

The data transmission routines do not check any data formats during the processing. All user-defined data is assumed to be transmitted in 1130 format. If conversion is necessary, this means that the System/360 data to be transmitted to the 1130 (by a call to GTWRT) should first be converted to the format acceptable to the 1130 by a call to the proper data conversion subroutine. It also means that the 1130 data that is received and read into the user-specified working array (by a call to GTRED) should then be converted to the format acceptable to the System/360 before further processing takes place.

The data conversion routines in System/360 are independent subroutines. Their functioning depends entirely on the user's program. When called, they perform their functions and return control to the user's program.

Details of the methods used to reorder and reformat data are discussed in the section "System/360 Conversion Routines."

MONITORING ASYNCHRONOUS ROUTINE PROCESSING (SYSTEM/360)

When a request for the 1130 asynchronous routine is encountered in a System/360 call to GTWRT, the asynchronous routine request message is transmitted to the 1130 system. Appropriate fields are set in the GTCB. When the 1130 asynchronous routine returns control to the routine that called it, an appropriate message is transmitted from the 1130 to notify the System/360 that the asynchronous routine is ended. The GTCB fields are reset accordingly.

Asynchronous routine processing in the 1130 does not change the control flow in the System/360 PTOP program. The System/360 program continues performing along the same basic path shown in Figure 7. However, a change does occur in the System/360 PTOP program processing when the System/360 asynchronous routine is requested.

Figure 11 shows the control and data flow for invoking the System/360 asynchronous routine, starting with the request from the 1130. The System/360 control functions remain the same as for any other data transmission function. PTOP's func-

tions are explained in the following paragraphs.

When System/360 receives the asynchronous routine request message from the 1130 program, IKDGTIRB saves the mainline program information, and sets a flag in the GTCB to indicate that the user-specified asynchronous routine is scheduled in the System/360 program. The necessary linkage is provided by an interruption request block (IRB) and an interruption queue element (IQE) constructed by IKDGTIRB, and control is routed to the asynchronous routine through the Asynchronous Exit Effector in the operating system.

The asynchronous routine may use the data transmission facilities provided in PTOP. As soon as the asynchronous routine is requested, IKDGTIRB dynamically acquires a save area, places the address in the GTCB, and stores the GTCB fields containing the transmission status of the mainline program in the save area. This allows the PTOP routines to perform operations for the asynchronous routine in the same manner as they are performed for the mainline program.

The asynchronous routine must end with a call to GTEND; this notifies the PTOP routines that it is finished processing. From this call to GTEND, the End Communication routine (IKDGTEND) is invoked. Then, IKDGTEND sends the asynchronous routine ended message to the 1130 system by using BTAM, and restores the GTCB with control information from the save area so that mainline program processing can be continued. When IKDGTEND returns control, the Controller frees the save area acquired when the asynchronous routine was requested.

TERMINATING COMMUNICATION (SYSTEM/360)

The End Communication routine (IKDGTEND) becomes active when the GTEND subroutine is called in System/360. For the mainline program, it sends the terminate communication link message to the 1130 system via BTAM, closes the specified communication line group, and frees the read buffer, the write buffer, and the GTCB. If no other communication line groups are currently active, IKDGTEND deletes the modules that were loaded by IKDGTNIT during the initialization process. When IKDGTEND deletes these modules, the Controller completes the termination of communication links by deleting the unit assignment table.

The System/360 communication link can be reestablished by reinitializing the System/360 PTOP routines while the 1130 system is maintaining its communication link. As long as the communication links are terminated in one or both of the systems, no PTOP communication can take place between the two systems. A call to GTNIT in the System/360 program will start the initialization process again.

Unlike the 1130 system, System/360 provides for terminating the 1130 communication link. This is accomplished by a call in the user's System/360 program to GTWRT that specifies a "controlopt" argument equal to 3. When IKDRDWRT encounters this request, it halts input/output operations and transmits the program termination request message to the 1130 system through BTAM. This message causes the 1130 PTOP program to relinquish control to the Disk Monitor System. The 1130 program cannot be reinitialized to communicate with the program that caused its termination.

If the 1130 program ends the communication link, the System/360 program and the PTOP routines remain intact although the PTOP program will not perform any more input/output operations until the communication link is reestablished. IKDGTIRB issues a Read Initial procedure placing the System/360 PTOP program in ready-to-read status. Thus, the System/360 can accept the initialization message transmitted from the 1130 system when the 1130 program reinitializes by calling the GTNIT subroutine.

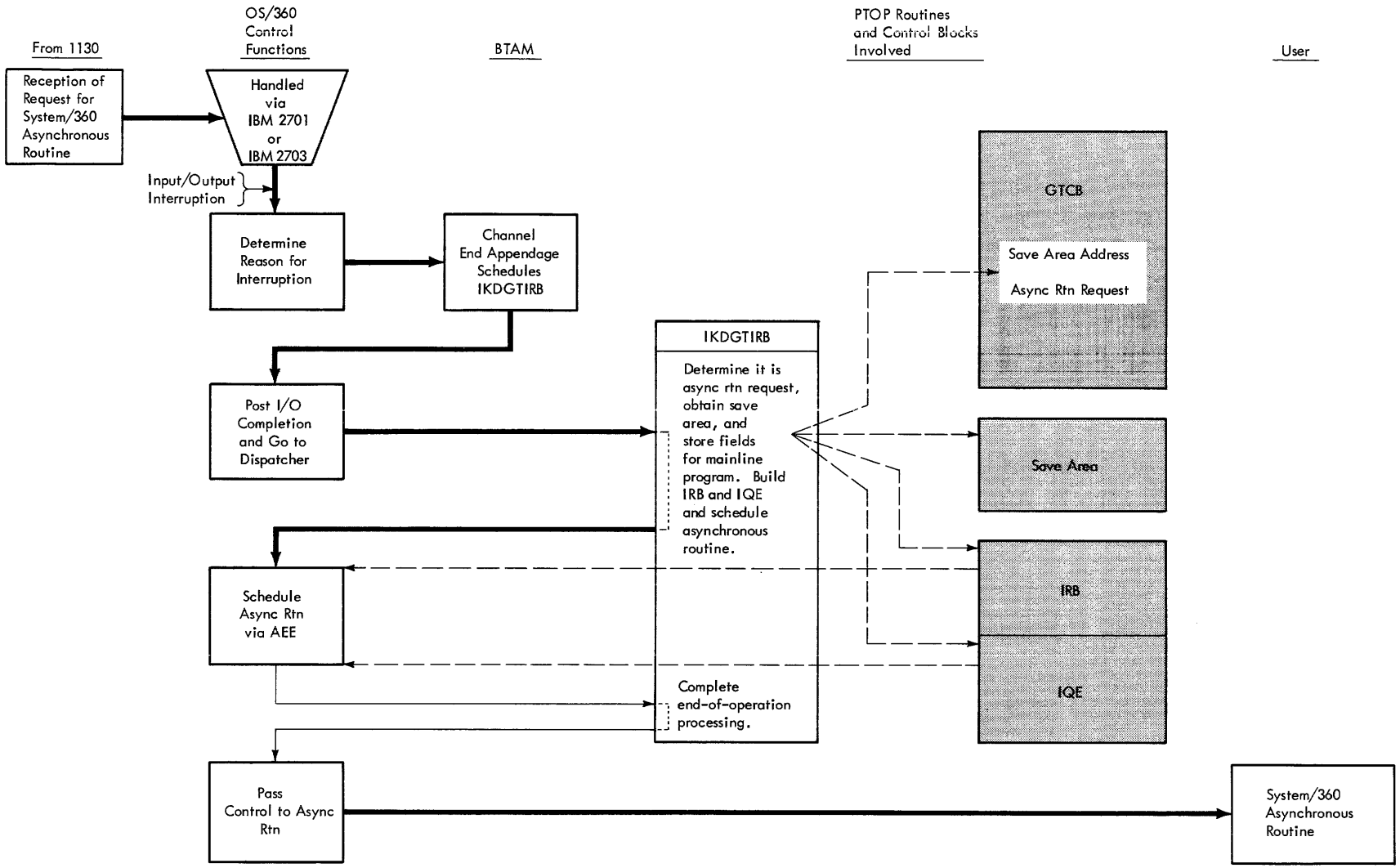


Figure 11. Control and Data Flow for Invoking the System/360 Asynchronous Routine

This section describes the PTOP data transmission modules provided in the 1130 system. The modules are:

- Initialization (GTNIT), which prepares the 1130 system for communication with the System/360 program.
- Read Data (GTRED), which begins the processing of a user's read request.
- Write Data (GTWRT), which begins the processing of a user's write request.
- Control Test (GTCLT), which tests for completion status of a user's data transmission or data reception operation.
- End Communication (GTEND), which breaks communication with the System/360 program.
- Telecommunications Monitor (GTP01), which interprets the information for a completed operation, continues with the data transmission or reception following interruptions, and handles PTOP transmission errors.

Entry Procedure

Upon entry, before performing their specialized functions, each of the transmission routines saves the contents of the index registers, the accumulator, and the status indicators of the calling program. In the detailed descriptions of each routine, this process is referred to as the entry procedure.

Invoking the Asynchronous Routine

As one of their first functions, GTRED, GTWRT, GTEND, and GTCLT use the same procedure to honor any pending request to invoke the 1130 asynchronous routine. To determine if an asynchronous routine request is pending, the processing PTOP routine checks the ASPND flag in the GTCOM. If the asynchronous routine is be invoked, the PTOP routine saves the program call status (that is, the contents of the accumulator and extension, the index registers, and the status indicators) in the GTCOM, and calls GTP02 to invoke the asynchronous routine.

GTP02 is a secondary entry point in the Telecommunications Monitor routine and is used exclusively for invoking the 1130 user's asynchronous routine. Before the asynchronous routine is called, GTP02 indicates that the mainline transmission status is to be saved, and waits for GTP01 to perform this function. After the save function is completed, GTP02 calls the asynchronous routine.

When the asynchronous routine returns control to GTP02, the asynchronous routine ended message is queued. GTP02 waits until GTP01 has sent the message to the System/360, and then returns control to the data transmission routine that started the invoking procedure so that the program call status can be restored and processing can continue as if no asynchronous routine request had been pending.

In the detailed descriptions of GTRED, GTWRT, GTEND and GTCLT, this procedure is referred to as invoking the asynchronous routine.

INITIALIZATION (MODULE NAME GTNIT)

Chart: AA

Function:

- Initializes and updates fields in the 1130 system PTOP communications area (GTCOM) for use by other PTOP routines.
- Transmits an initialization message to the System/360 PTOP program.
- Posts the results of the initialization attempt for the user.

Entry: GTNIT, from the calling program.

Exit: To SCAT2 and to the calling program.

Input: The following information in a parameter list:

+ 0 "writebuf"
address of the array to be used as an output buffer for data transmission.

+ 1 "readbuf"
address of the array to be used as an input buffer for data reception.

+ 2 CALL "asynroutine"
a call to the 1130 user's asynchronous routine.

+ 4 "password"
address of the alphameric constant used to establish correlation between the 1130 and the System/360 programs.

+ 5 "initcode"
address of the variable into which the status of the initialization attempt is to be placed.

Output: The initialization message is transmitted to the System/360 PTOP program, and one of the following values is placed in the "initcode" variable:

- Negative value, if the password transmitted by the 1130 system does not match the System/360 password.
- Zero, if the communication link is established.
- Positive value, if communication linkage cannot be established because of error conditions in the transmission line or if contact is not established with the System/360 within five minutes.

Operation: After the entry procedure, GTNIT sets the RTBSY flag in the GTCOM to indicate that it is busy, and checks the OKNIT flag in the GTCOM to see if the PTOP program had previously been initialized.

If the program is initialized, GTNIT stores the user-specified parameters in the GTCOM, sets the "initcode" variable to zero, and returns to the calling program.

If the program has not yet been initialized, GTNIT ensures that Index Register 3 will contain the proper value for the supervisor to access the transfer vector table by placing the setting from the core image header (symbolic location @XR3X) into the resident supervisor (at symbolic location $XR3X). Then, GTNIT sets up the initialization message in the PTOP system buffer, and calls SCAT2 (using the LIBF statement) to transmit the message to the System/360.

GTNIT then waits for the operation to be completed (indicated when the RTBSY flag in the GTCOM is zero), and determines whether communication linkage has been established with the System/360 by checking the OKNIT flag.

If OKNIT is one, GTNIT stores the addresses of the 1130 password, the read and write buffers, and the asynchronous routine in the GTCOM. It indicates a successful initialization attempt for the user, and returns to the calling program.

If OKNIT is zero (unsuccessful initialization), GTNIT tests for reception of an EOT character (the input/output count area contains zero). If an EOT character was received, the passwords did not agree; GTNIT places a negative value in the "initcode" variable, requests a Close operation of SCAT2, and returns to the calling program. Otherwise, GTNIT assumes a transmission error or no response from the System/360, places a positive value in the "initcode" variable, requests a Close operation of SCAT2, and returns to the calling program.

READ DATA (MODULE NAME GTRED)

Chart: BA

Function: Performs the following scheduling functions for an 1130 user's request to receive data:

- Calls GTP02 to invoke the 1130 asynchronous routine if it is pending.
- Sets pointers and flags in the GTCOM for use by the GTP01 routine.

Entry:

- GTRED, from the calling program.
- GTRD2, from GTP02.

Exit: To GTP02 and to the calling program.

Input: The following information in a parameter list:

+ 0 "userarray"
address of the array into which the transmitted data is to be read.

+ 1 "elcount"
address of the integer that specifies the number of elements to be read.

+ 2 "datatype"
address of the integer that specifies the format of the incoming data.

Output: Appropriate fields are set in the GTCOM to indicate that a user read request is pending.

Operation: After the entry procedure, GTRED checks the OKNIT flag in the GTCOM to see if the PTOP program is initialized. If it has not been initialized, GTRED sets the LSTRD status field in the GTCOM to indicate the operation was not started, and returns to the calling program. Otherwise, GTRED determines if the 1130 asynchronous routine needs to be invoked (see "Invoking the Asynchronous Routine").

If no asynchronous routine need be invoked, GTRED checks the data type code in the LSTRD field of the GTCOM to see if a user read request is active or pending (code is non-zero). If so, GTRED returns immediately to the calling program.

Otherwise, GTRED places information in the GTCOM. It computes the data length and stores it in RDCNT. It then stores the data type code in LSTRD, computes the highest element address of the user array, and stores the address in RDPTR. Next, GTRED sets flags in BITS2 to indicate that an 1130 ready-to-read message and a user read request are pending, turns on the RTBSY flag, sets the LSTRD status to indicate that the operation was not started, and returns to the calling program.

WRITE DATA (MODULE NAME GTWRT)

Chart: CA

Function: Performs the following scheduling functions for an 1130 user's request to transmit data:

- Calls GTP02 to invoke the 1130 asynchronous routine if it is pending.
- Initializes pointers and flags in the GTCOM for use by the GTP01 routine.

Entry:

- GTWRT, from the calling program.
- GTWR2, from GTP02.

Exit: To GTP02 and to the calling program.

Input: The following information in a parameter list:

+ 0 "userarray"
address of the array containing the data to be transmitted.

+ 1 "elcount"
address of the integer that specifies the number of elements to be transmitted.

+ 2 "datatype"
address of the integer that specifies the format of the outgoing data.

+ 3 "controlopt"
address of the integer that specifies the destination of the data to be transmitted.

Output: Appropriate fields are set in the GTCOM to indicate a user write request is pending.

Operation: After the entry procedure, GTWRT checks the OKNIT flag in the GTCOM to see if the PTOP program is initialized. If not, GTWRT sets the LSTWT field in the GTCOM to indicate that the last write request was not started, and returns to the calling program. Otherwise, GTWRT determines if the 1130 asynchronous routine needs to be invoked (see "Invoking the Asynchronous Routine").

If the asynchronous routine does not need to be invoked, GTWRT checks the "controlopt" parameter to see if the write request is a normal data transmission to the System/360 program or a request to schedule the System/360 asynchronous routine.

If the request is for the System/360 asynchronous routine, GTWRT checks the GTCOM to see if the System/360 asynchronous routine is already active. If it is, GTWRT returns immediately to the calling program. If the System/360 asynchronous routine is not active, GTWRT queues the request and moves the data (parameter) to be transmitted into the ASDAT field of the GTCOM. GTWRT sets the RTBSY flag to one and returns to the calling program.

If the request is for normal data transmission to the System/360 program, GTWRT places information in the GTCOM. It computes the highest address of an element in the user array and stores the address in WTPTR. It then computes the data length, and stores it in WTCNT. GTWRT stores the data type code in LSTWT, sets the RTBSY flag to one, and queues the data message. It then sets the status portion of the LSTWT field to indicate that the operation was not started, and returns to the calling program.

CONTROL TEST (MODULE NAME GTCLT)

Chart: DA

Function:

- Calls GTP02 to invoke the 1130 asynchronous routine if it is pending.
- Provides information to the user regarding the status of the most recent read or write request.

Entry:

- GTCLT, from the calling program.
- GTCT2, from GTP02.

Exit: To GTP02 and to the calling program.

Input: A parameter list containing the following:

+ 0 "testcode"
address of the integer that specifies the operation to be tested.

+ 1 "statcode"
address of the variable into which a value indicating the status of the tested read or write operation is to be placed.

Output: The user's "statcode" variable is set to indicate the status of the last read or write operation requested by the user. The various values placed in the "statcode" variable are found in the GTCOM, and the meanings are explained in Appendix A.

Operation: After the entry procedure, GTCLT determines if the 1130 asynchronous routine needs to be invoked (see "Invoking the Asynchronous Routine").

If the asynchronous routine does not need to be invoked, GTCLT checks the value of the "testcode" variable to determine the type of operation to be tested, and if the user specified that GTCLT should wait for the operation to be completed. GTCLT performs the function indicated, and returns to the calling program.

The possible "testcode" values and the functions performed as a result of each value are as follows:

Value of "testcode"	GTCLT Action
1	Places the value of the LSTRD status field into the "statcode" variable.
2	Places the value of the LSTWT status field into the "statcode" variable.
3	Tests the LSTRD status field. If it contains a 1, 4, 5, 6, or 7, GTCLT does not wait, but places that value in the "statcode" variable. For status values of 2 or 3, GTCLT checks LSTRD data type code to see if a valid input/output operation is active or pending. If so, GTCLT waits for the LSTRD status field to equal 1, 4, 5, 6, or 7, and places that value in the "statcode" variable. If no valid input/output operation is indicated, GTCLT does not wait, but places a value of 2 in the "statcode" variable.
4	Tests the LSTWT status field and performs the same functions as described for "testcode" value 3 except that the LSTWT field is used instead of the LSTRD field.

Note: GTCLT honors any request to invoke the 1130 asynchronous routine that occurs while GTCLT is waiting for the status field value to change.

END COMMUNICATION (MODULE NAME GTEND)

Chart: EA

Function:

- Calls GTP02 to invoke the 1130 asynchronous routine if it is pending.
- Halts PTOP communication after active write request is completed.
- Transmits the terminate communication link message to the System/360 PTOP program.
- Clears certain fields in the GTCOM.
- Resets the SCA.

Entry:

- GTEND, from the calling program.
- GTND2, from GTP02.

Exit: To GTP02 and to the calling program.

Input: None.

Output: The System/360 PTOP program is notified that the 1130 program called GTEND, and the GTCOM fields that indicate the status of input/output operations are reset.

Operation: After the entry procedure, GTEND checks the OKNIT flag in the GTCOM to determine if the 1130 PTOP program is still initialized. If OKNIT is zero, GTEND returns immediately to the calling program.

If OKNIT is one, GTEND then checks the ASPND flag in the GTCOM to determine if the 1130 asynchronous routine is to be invoked (see "Invoking the Asynchronous Routine").

If the asynchronous routine does not need to be invoked, GTEND queues the terminate communication link message to be transmitted at the first opportunity. It waits for completion of the operation, indicated when the RTBSY flag in the GTCOM is set to zero by GTP01.

GTEND then resets the SCA by calling SCAT2 to close operations. Upon return from SCAT2, GTEND clears the BITS and BITS2 fields, and places a value of 2 in the status portion and zero in the data type portion of the LSTRD and LSTWT fields in

the GTCOM. GTEND then returns to the calling program.

TELECOMMUNICATIONS MONITOR (MODULE NAME GTP01)

Charts: FA, FB, FC, FD, FE

Function:

- Monitors completion of input/output requests.
- Processes errors encountered during PTOP data transmission.
- Invokes the 1130 user's asynchronous routine.

Entry:

- GTP01, from SCAT2 at interruption time.
- GTP02, from GTRED, GTWRT, GTCLT, or GTEND when the asynchronous routine is to be invoked.

Exit:

- To SCAT2 when entered at GTP01.
- To the secondary entry point of the calling routine when entered at GTP02.

Input:

- When entered at GTP01, the accumulator contains zero or an error code (see "Error Processing Phase").
- When entered at GTP02, Index Register 1 contains a value that identifies the calling routine as follows:

  0 = GTWRT
  2 = GTRED
  4 = GTCLT
  6 = GTEND

Output: The GTCOM is updated to indicate the current transmission status of the PTOP requests.

The accumulator is set with a value to direct SCAT2 operation (see the following description of GTP01 operation for details).

Operation: When the routine is entered at GTP01, the accumulator is checked to determine whether error processing is necessary before end-of-operation processing is performed. If the accumulator is zero, no error processing is necessary and GTP01 enters the end-of-operation processing phase.

If the accumulator is not zero, GTP01 processes the error condition.

Error Processing Phase: After saving contents of the accumulator, Index Register 1, and status indicators, GTP01 checks the code in the accumulator to determine its next action.

The codes, meanings, and related actions taken by GTP01 are as follows. Accumulator settings upon exit from GTP01 are shown in parentheses. Zero in the accumulator directs SCAT2 to end the operation as if a normal completion occurred. A non-zero value in the accumulator directs SCAT2 to retry the current operation up to seven additional times.

Code	Meaning and GTP01 Action
/8000, /0800 /0400, or /0100	Unrecoverable error: Sets the ERRST field in the GTCOM to indicate an unrecoverable error, and returns to SCAT2 (/0000).
/4000	Contention: Sets the timeout indicator in the ERRST field, and returns to SCAT2 (non-zero).
/2000	Receive timeout: If the current operation is Receive Initial, GTP01 inspects the GTCOM for any write operation that may be pending or for an indication that the save mainline status function is to be performed. For either of these cases, GTP01 queues any requested user read operation as pending in the BITS2 field, and returns to SCAT2 (/0000). For any other case, GTP01 returns to SCAT2 (non-zero).
/1000	Overflow: If the current operation is Receive Initial, GTP01 sets the incorrect length indicator in the ERRST field in the GTCOM, and returns to SCAT2 (/0000). Otherwise, it sets the unrecoverable error indicator and returns to SCAT2 (/0000).
/0200	Transmit timeout: When the current operation is Transmit End, GTP01 returns to SCAT2 (/0000). When the current operation is Transmit Initial, GTP01 checks the message being transmitted to determine if it is an initialization message. If it is not, GTP01 indicates a timeout in the ERRST field, and returns to SCAT2 (/0000). For an initialization message, GTP01 checks whether the 1130 PTOP program is initialized, and if it is, GTP01 indicates a timeout and returns to SCAT2 (/0000).

If 1130 PTOP is not initialized, GTP01 decrements the GTCOM retry counter used to control the number of times that the initialization message is to be retried by SCAT2, and returns to SCAT2 for further retries (non-zero). If the retry counter becomes zero, indicating that the time limit of five minutes for an initialization attempt has elapsed, GTP01 indicates a timeout and returns to SCAT2 (/0000).

End-of-Operation Processing Phase: After saving the contents of the accumulator, index registers, and status indicators, GTP01 checks the GTCOM to determine which operation has just been completed. GTP01 takes appropriate action, which may include getting the next operation started (see "Starting the Next Operation").

If the operation is Transmit Initial, with no timeout and no EOT character received in response, GTP01 calls SCAT2 to transmit the EOT character to the System/360, sets the active function flag in the GTCOM, and then returns to SCAT2. If an EOT was received in response to the message, GTP01 sets the error flag in the ERRST field, and checks for the type of message being transmitted to determine the next action. If a Transmit timeout occurred (flagged earlier by GTP01), or if the operation is Transmit End, GTP01 proceeds to check the type of message being transmitted.

Messages transmitted and associated actions taken by GTP01 are as follows:

- For a data message, GTP01 checks the ERRST field and, if any unrecoverable errors occurred, sets the LSTWT status field to indicate a transmission line error. Otherwise, GTP01 checks the incorrect length flag in LSTWT, sets the LSTWT status appropriately, and starts the next operation.

- For a ready-to-read message, GTP01 checks for unrecoverable errors. If any are indicated, GTP01 sets the LSTRD status to indicate a transmission line error and cancels any user read request. GTP01 then starts the next operation.

- For a terminate communication link message, GTP01 sets the RTBSY indicator to zero and returns to SCAT2.

- For an initialization message, GTP01 does one of the following:

  (1) If no EOT was received and there were no transmission errors, GTP01 sets the OKNIT flag to one, places a status code of 2 in the LSTRD and LSTWT fields, and starts the next operation; or,

  (2) If an EOT was received or if there were transmission errors, then GTP01 determines if the 1130 user has called GTEND or if the 1130 PTOP program is not currently initialized. In either case, GTP01 sets the RTBSY indicator to zero and returns to SCAT2. If the 1130 PTOP program is initialized (and is awaiting reinitialization in the System/360), GTP01 queues the initialization message again and starts the next operation.

- For an asynchronous routine request message, GTP01 checks the ERRST field, and if any input/output errors are indicated, it queues the message again and starts the next operation. If no errors are indicated, GTP01 saves the System/360 read status for its mainline program, if necessary, sets the OSRED field to zero, sets the ASACT field to one, and starts the next operation.

- For an asynchronous routine ended message, GTP01 determines if any input/output error is indicated, and if so, queues the message again and starts the next operation. If no errors are indicated, GTP01 dequeues any user input/output operations not completed, dequeues the message just transmitted, and restores the mainline transmission status in the appropriate fields of the GTCOM. If the System/360 user called GTEND from the mainline program, GTP01 sets the status code in the LSTRD and LSTWT fields to 5. GTP01 then starts the next operation.

If the operation is Receive Initial, GTP01 tests for a timeout and various error conditions. If a timeout occurred, GTP01 starts the next operation. If unrecoverable errors occurred (indicated in ERRST), GTP01 calls SCAT2 to send the negative acknowledgment (NAK) via Receive Repeat. If an EOT was received, GTP01 restarts the current input/output operation and calls SCAT2 to perform the Receive Initial again.

If none of the conditions described in the preceding paragraph exists, GTP01 determines if the message received is valid. If not, GTP01 calls SCAT2 and requests Receive Repeat. If the message is valid, GTP01 determines if it is a data message for the user, and if so, sets the LSTRD status code to indicate that the operation is still in process. GTP01 then calls SCAT2 to perform the Receive Continue operation, sets the active function flag in the GTCOM, and returns to SCAT2.

If the operation is Receive Continue/Repeat, GTP01 checks the ERRST field for the existence of an unrecoverable error. If an unrecoverable error occurs while receiving data into the user's read buffer, GTP01 sets the LSTRD status to indicate a transmission line error, and starts the next operation. If an unrecoverable error occurs while using the system buffer, GTP01 starts the next operation. If there were no unrecoverable errors, GTP01 saves the DATA ID from the message received in the DSREF field of the GTCOM, and determines which message was received by checking the control byte (following the starting control characters) of the message header.

Messages received and associated actions are as follows:

- For a data message, GTP01 compares the data count received with the RDCNT, checks the ERRST field for incorrect length, sets the LSTRD status appropriately, transfers the data to the user-specified array, and starts the next operation.

- For an asynchronous routine ended message (from System/360), GTP01 restores the System/360 mainline read status, sets the ASACT field to zero, and starts the next operation.

- For a ready-to-read message, GTP01 determines whether both systems are in ready-to-read status for which it sets the LSTRD field appropriately, stores the System/360 read count in the OSCNT field, sets the OSRED flag to one in the GTCOM, and starts the next operation.

- For a program termination request message, GTP01 modifies the ILS01 return address to return to GTP01 after the interrupt level is reset. When GTP01 is reentered, it calls SCAT2 to reset the SCA, and exits to $EXIT in the resident supervisor.

- For a terminate communication link message, GTP01 sets LSTWT and LSTRD status fields to indicate that the System/360 user has called GTEND. If the 1130 user also called GTEND, GTP01 sets RTBSY to zero and returns to SCAT2. Otherwise, if a user read request is pending, GTP01 queues a ready-to-read message, and then queues an initialization message, which has a higher priority than a user read. GTP01 also determines if the 1130 asynchronous routine has ended (that message is pending) in which case it restores the transmission status of the mainline program. GTP01 then starts the next operation.

- For an asynchronous routine request message, GTP01 turns on the ASPND flag, saves the asynchronous routine parameter in the ASPRM field of the GTCOM, and starts the next operation.

Starting the Next Operation: GTP01 clears the error and operation fields in the GTCOM. If necessary, it saves the 1130 mainline transmission status in the appropriate fields of the GTCOM, and sets the GTCOM fields appropriately. Then, GTP01 checks BITS2 to determine what requests, if any, are pending, and processes them in the order they are found in BITS2. However, if the last completed operation required a transmit function, GTP01 ensures that the next operation is a receive function.

For each message to be transmitted, GTP01 stores the I/O count, header information, necessary text, and line control characters in the system or user write buffer, increments the write sequence number, clears the pending flag, sets the message type flag, calls SCAT2 to transmit the message, and returns to SCAT2.

When the ASPND flag is set to one, indicating a request to invoke the 1130 asynchronous routine, it has priority over a data message, a ready-to-read message, or a user read request which may be pending. In this case, when no other operations are pending, GTP01 sets the RTBSY flag to zero, sets up a system read with the maximum word count, calls SCAT2 to perform a Receive Initial, and returns to SCAT2.

If the next operation is the transmission of the end communication link message, and the 1130 asynchronous routine is active (BIT12 in the GTCOM BITS field is set to one), GTP01 causes the asynchronous routine ended message to be transmitted to the System/360 first. It then transmits the end communication link message.

Before sending the asynchronous routine request message, GTP01 saves the System/360 read status in the SAVST field of the GTCOM. It restores this status when the System/360 sends the asynchronous routine ended message.

When the asynchronous routine ended message is to be sent, GTP01 cancels any pending read or write operation, restores the mainline transmission status in the GTCOM, and sets the BITS12 flag to zero to indicate the 1130 asynchronous routine is no longer active. GTP01 then transmits the message.

GTP01 honors a pending 1130 data message request only after a System/360 ready-to-read message has been received. It sets the LSTWT status to indicate the operation

was started but not complete when it sends the data message.

To honor a user's read request, GTP01 sets up the user's read buffer to receive the data message from System/360, clears the read pending flag, sets the message type flag, calls SCAT2 to perform a Receive Initial, and returns to SCAT2.

Invoking the Asynchronous Routine: When entered at GTP02, the Telecommunications Monitor routine saves the contents of the accumulator, index registers, and status indicators; sets the BIT11 flag in the GTCOM to indicate that the mainline transmission status needs to be saved; and loops until GTP01, through normal interruption processing, saves this status and turns off the flag. Then, GTP02 turns on the BIT12 flag to indicate the 1130 asynchronous routine is active. It branches to the user's asynchronous routine, passing the user-specified parameter in the word following the Branch instruction.

When the asynchronous routine returns control, GTP02 queues the asynchronous routine ended message to be sent. Again, GTP02 loops until the message is transmitted, and returns to the calling routine.

FLOWCHARTS

This section contains autocharts showing the logic flow for the 1130 PTOP data transmission modules. The charts are ordered alphabetically (according to identification) in the sequence in which the routines are described. Refer to Appendix G for an explanation of the symbols used on the autocharts.

Chart AA. Initialization Routine (1130)

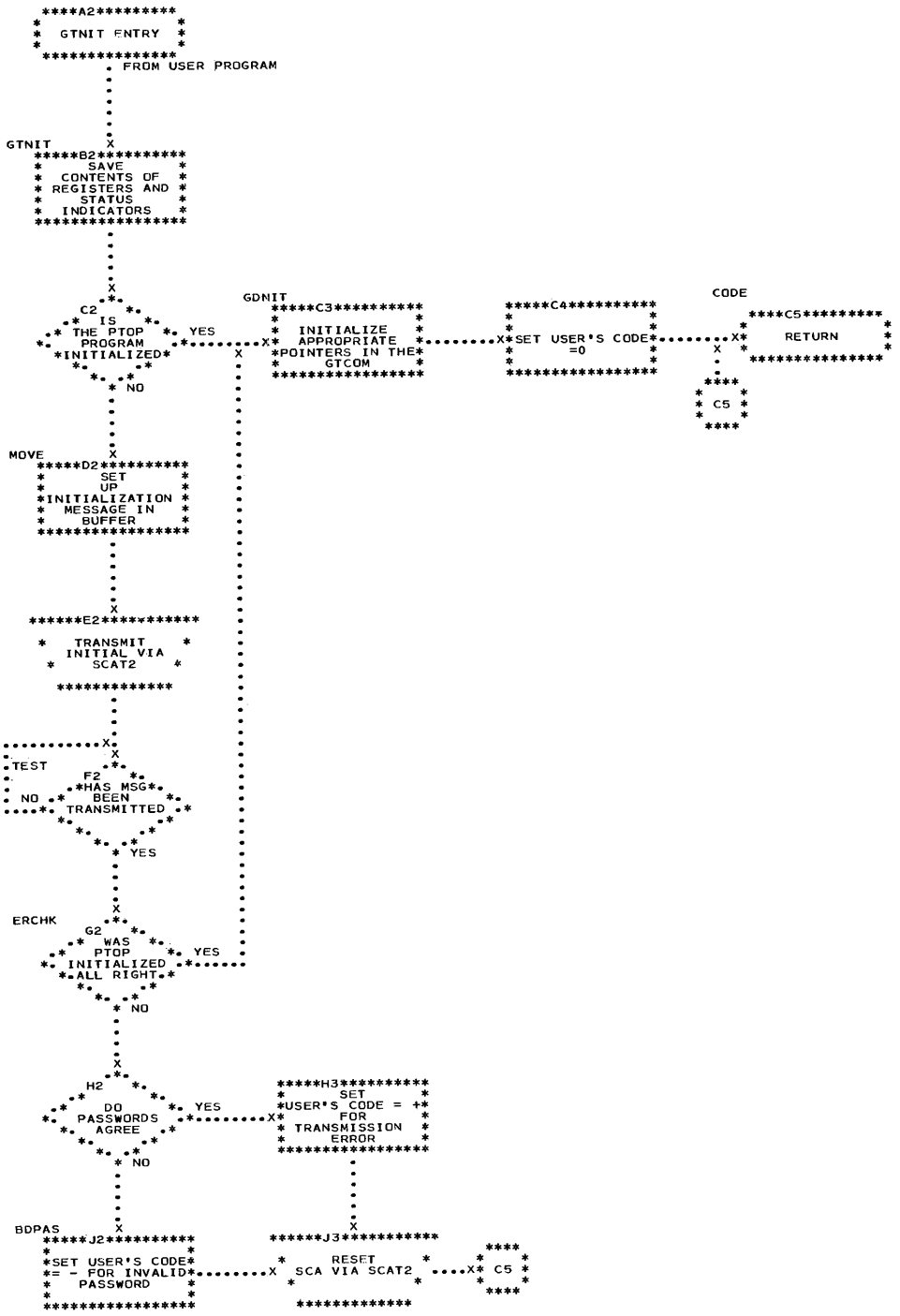


Chart BA. Read Data Routine (1130)

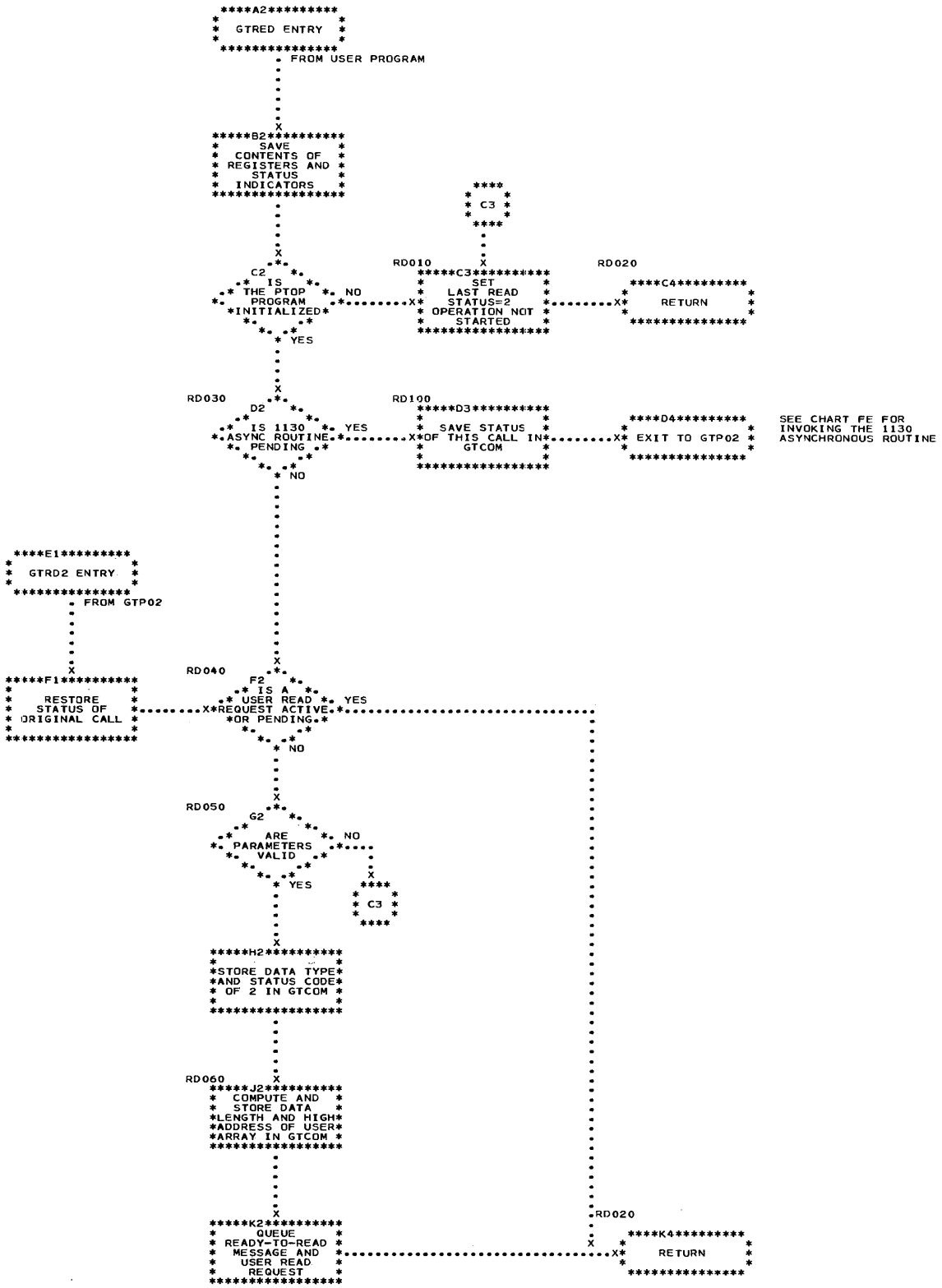


Chart CA. Write Data Routine (1130)

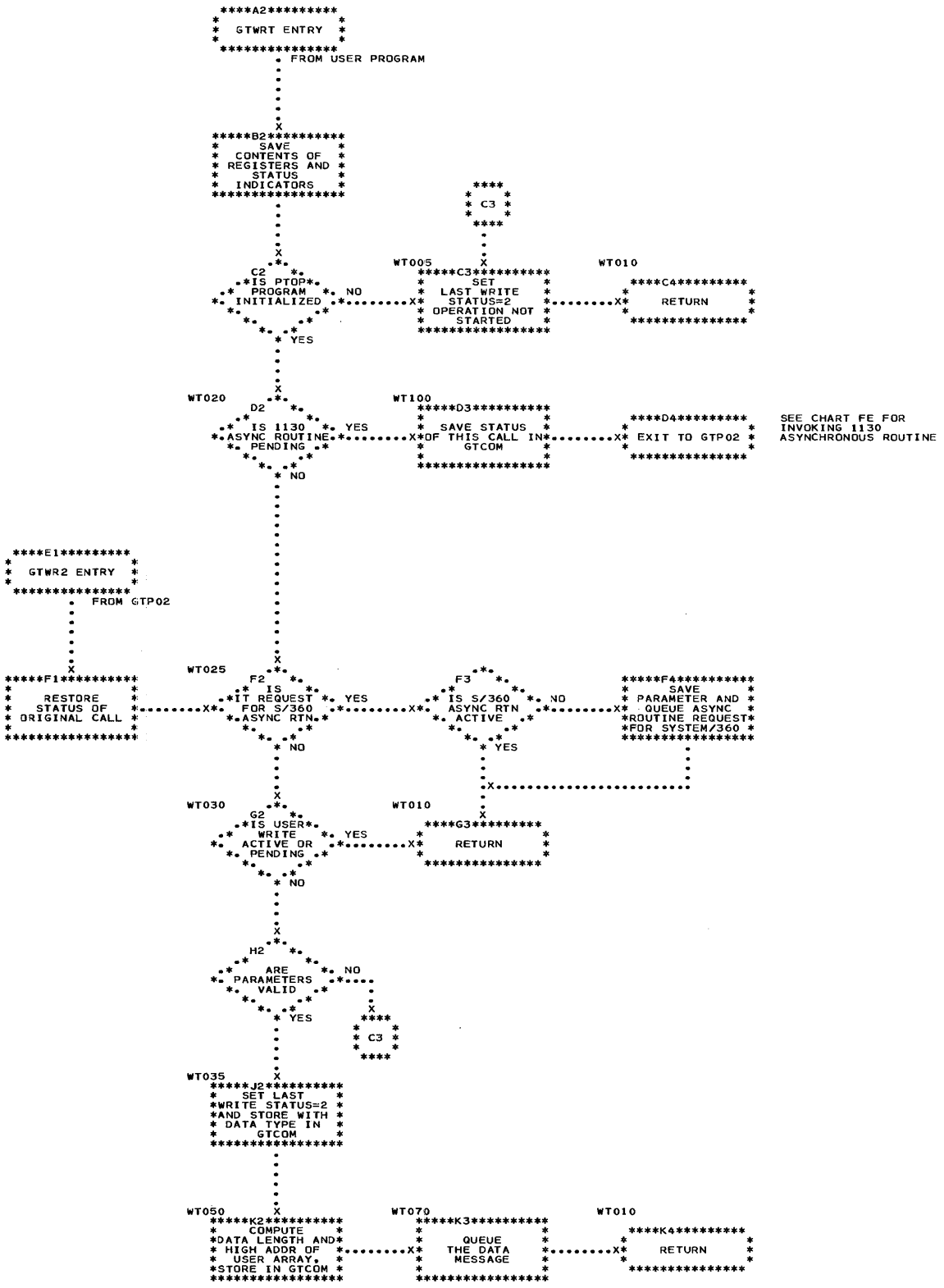


Chart DA. Control Test Routine (1130)

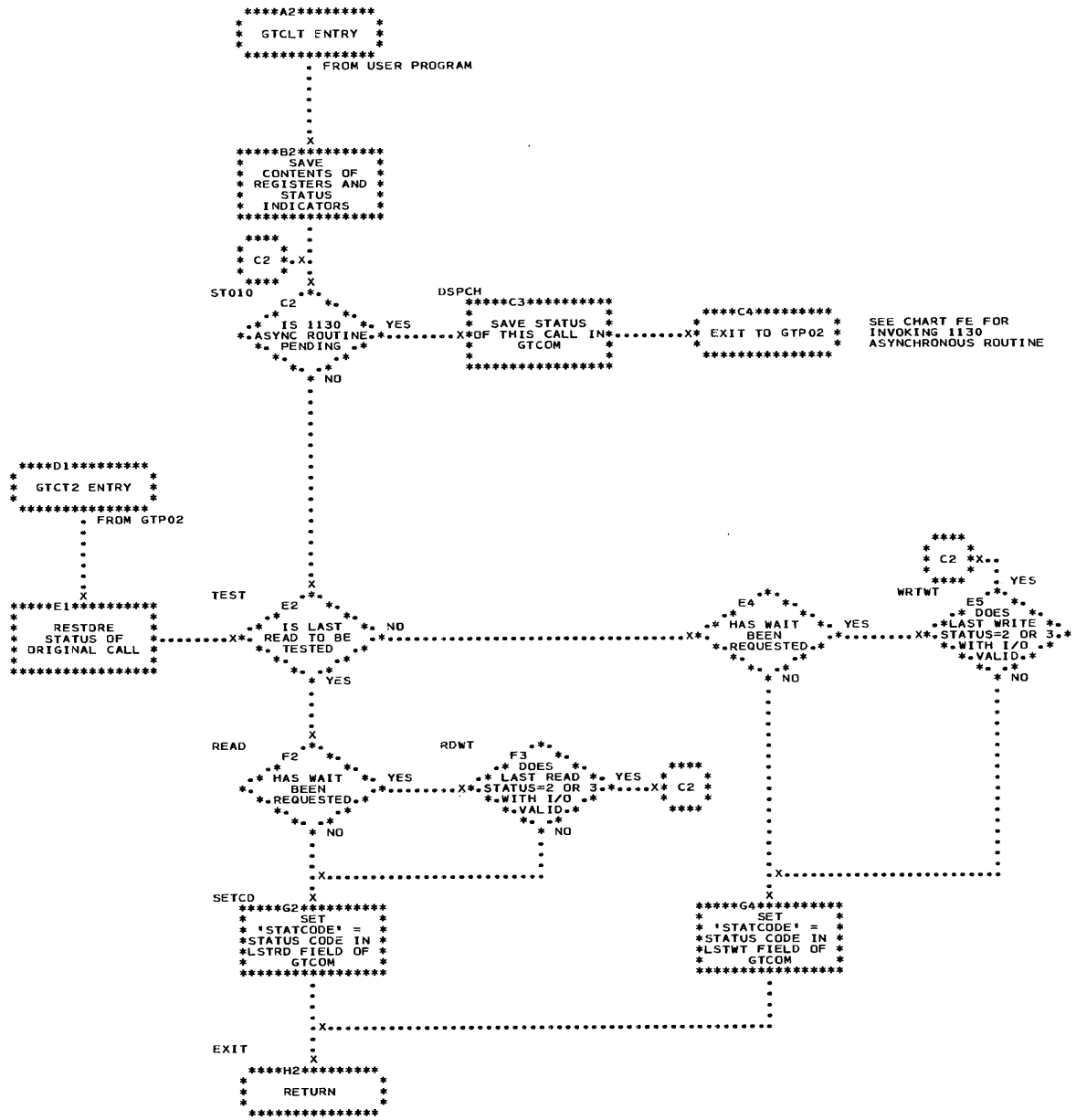


Chart EA. End Communication Routine (1130)

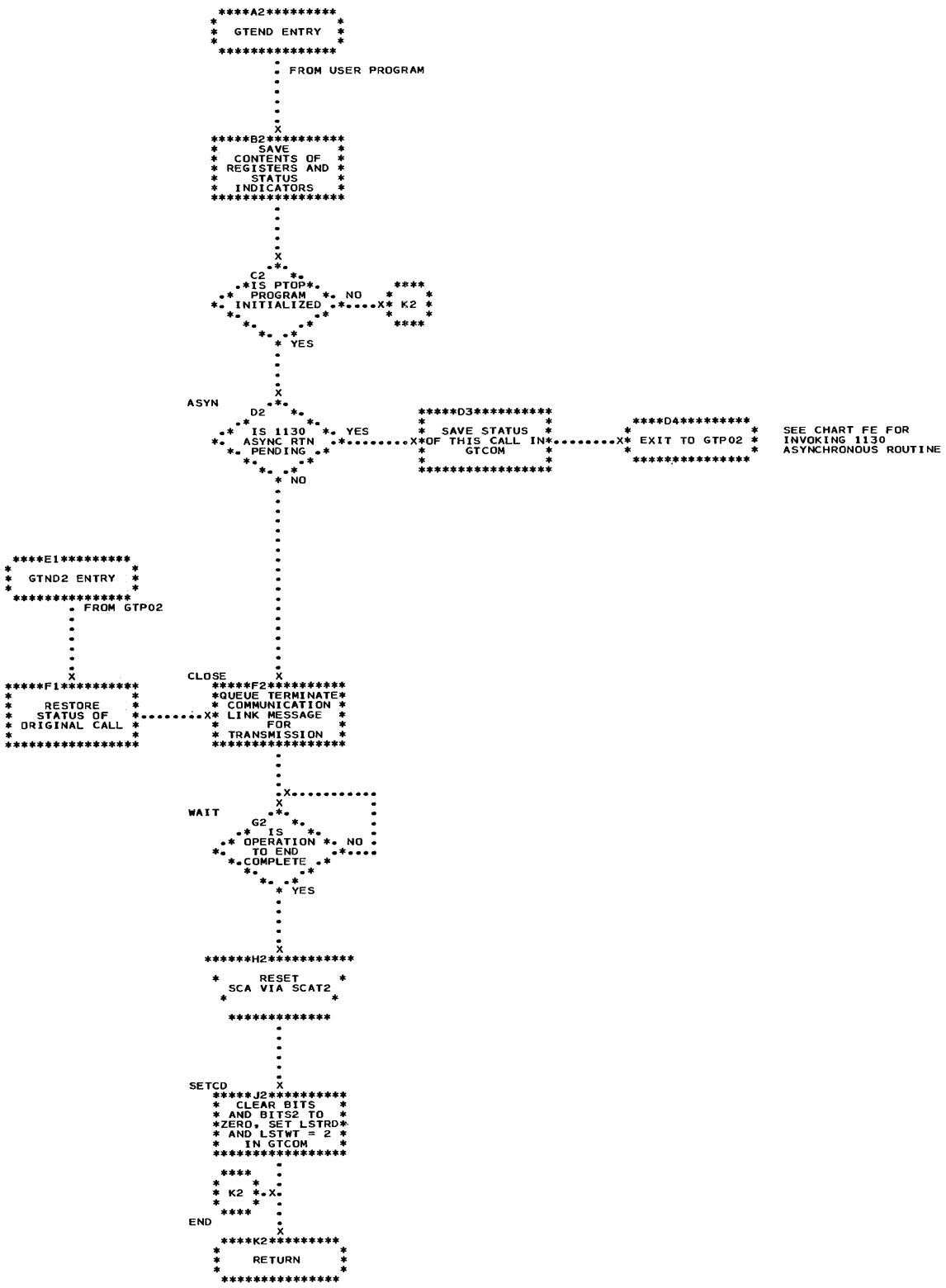


Chart FA. Telecommunications Monitor Routine (1130) (Part 1 of 5)

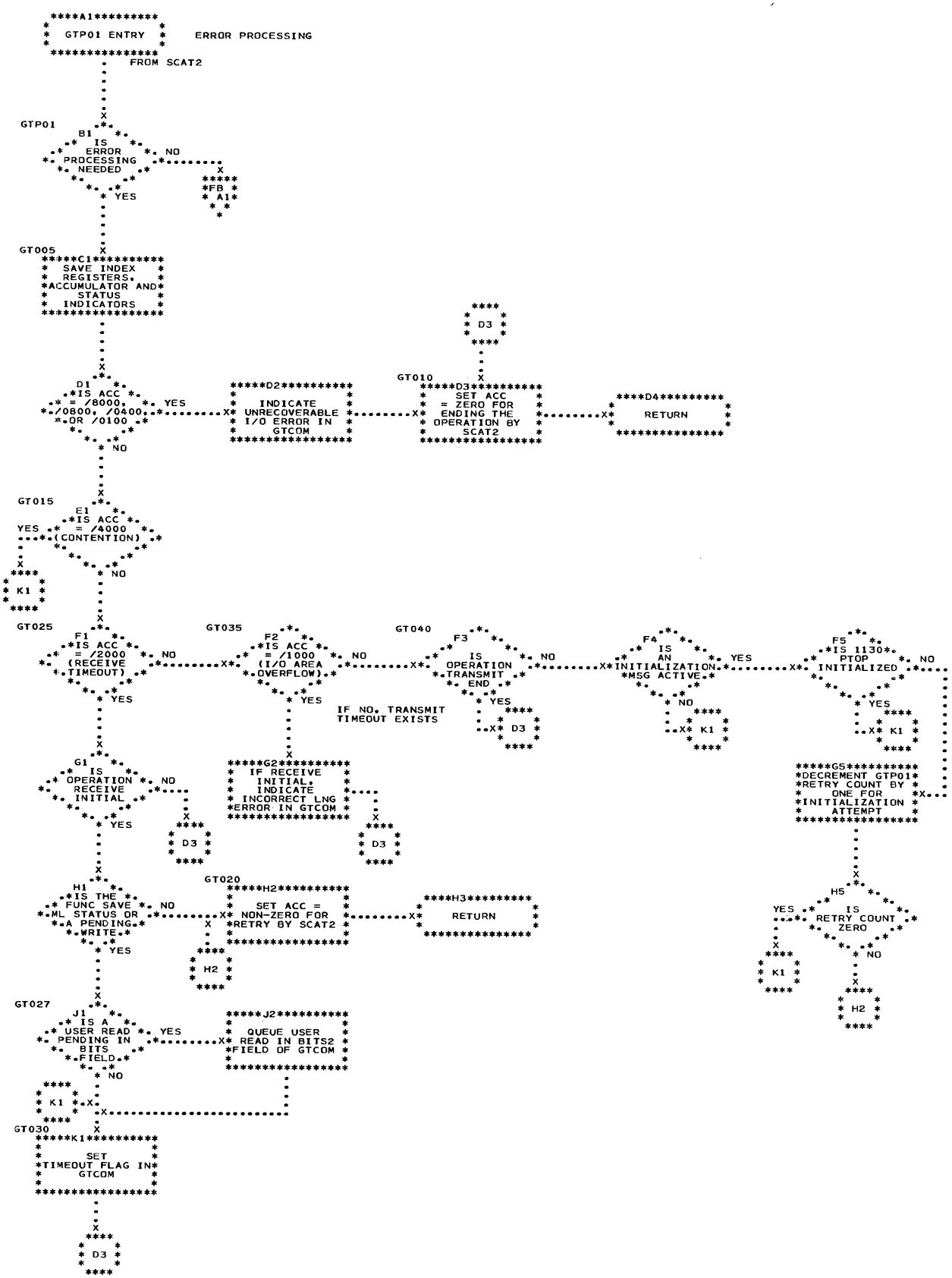


Chart FB. Telecommunications Monitor Routine (Part 2 of 5)

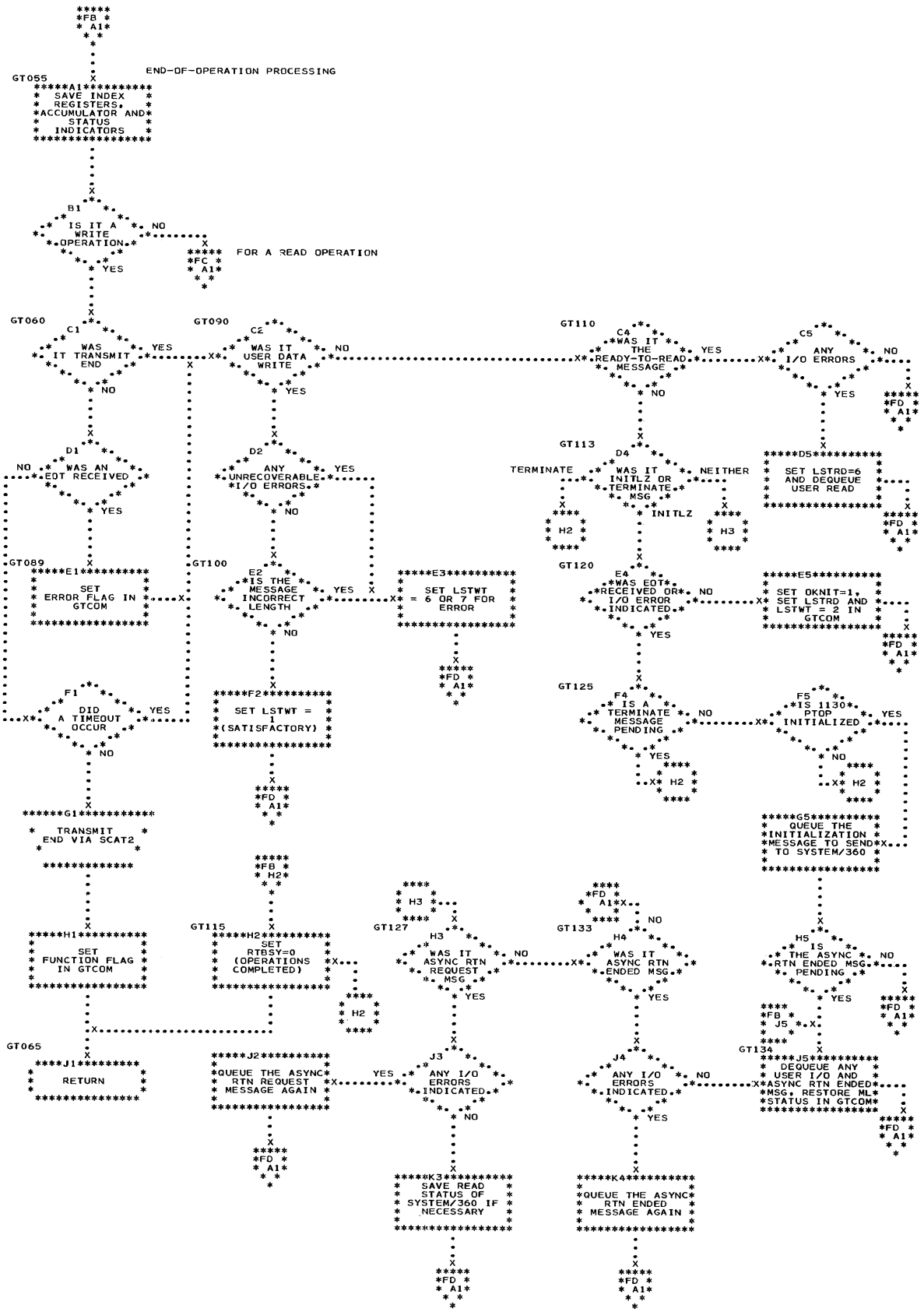


Chart FC. Telecommunications Monitor Routine (Part 3 of 5)

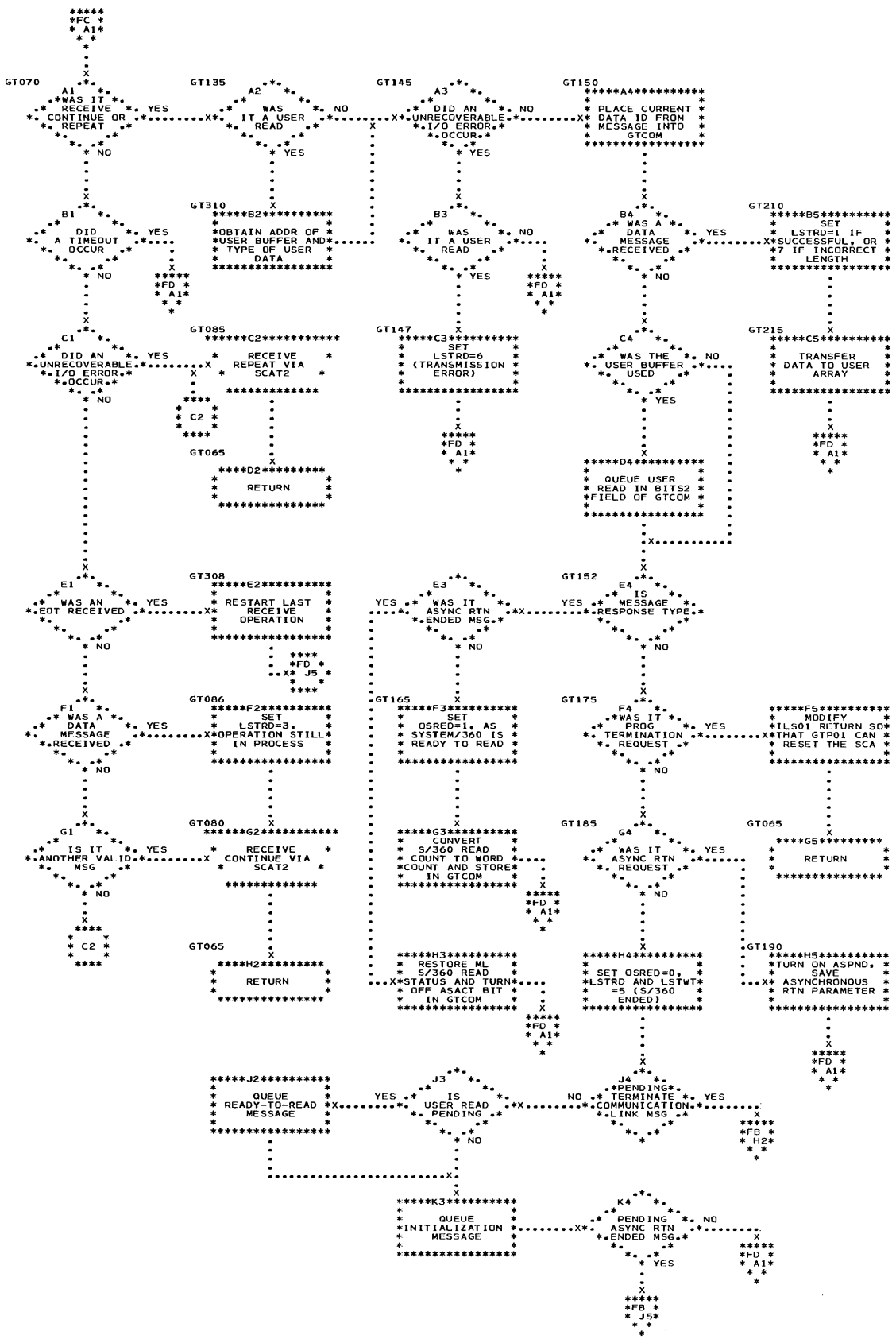


Chart FD. Telecommunications Monitor Routine (Part 4 of 5)

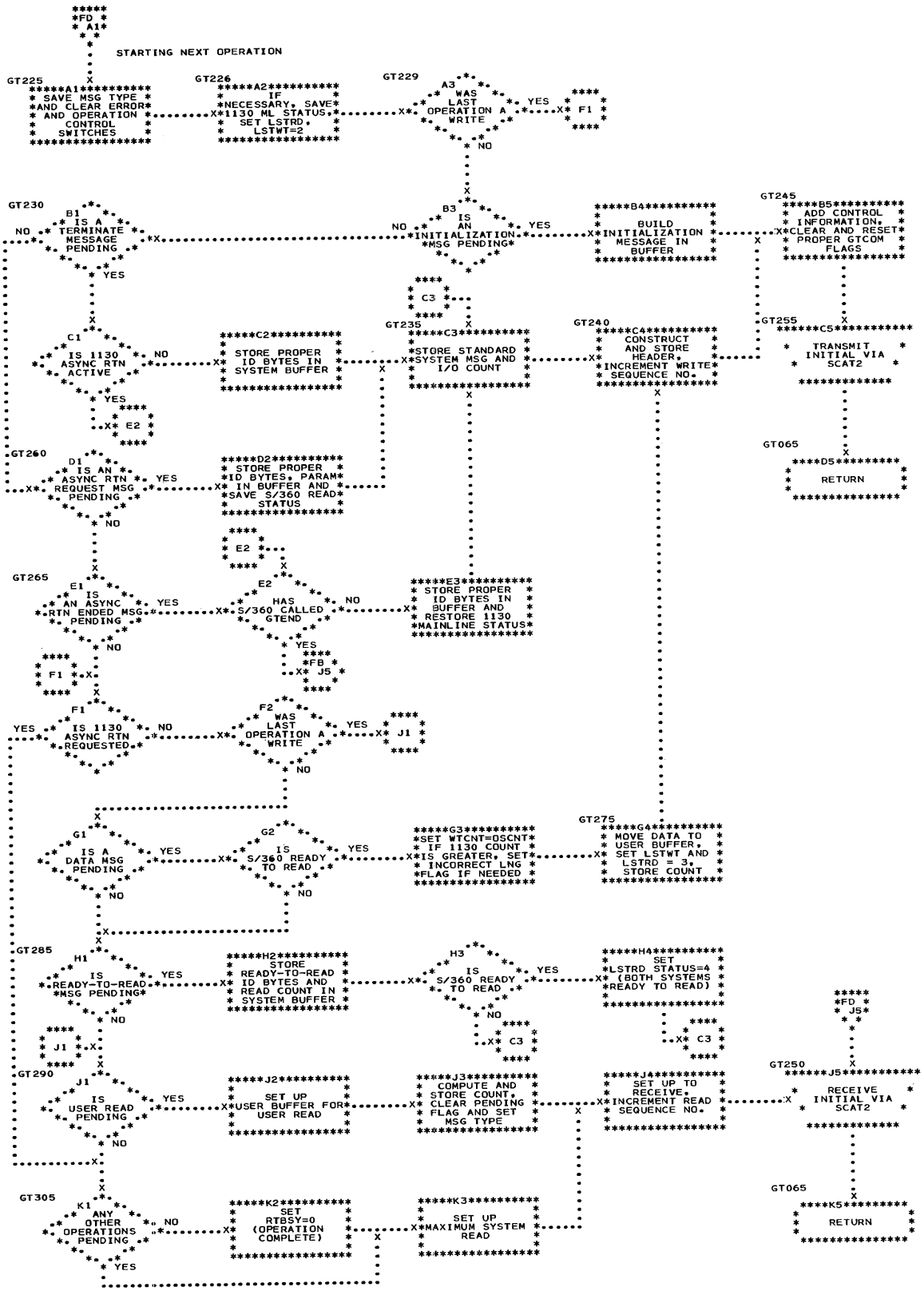
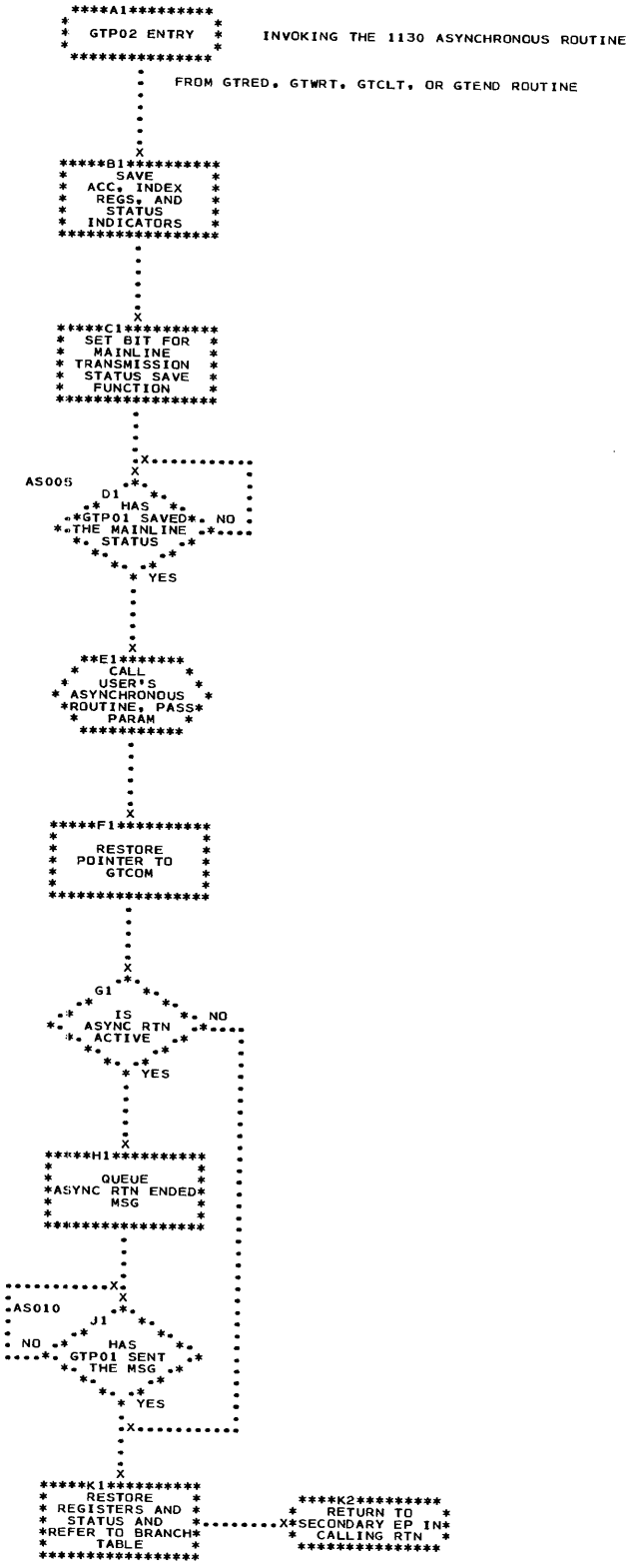


Chart FE. Telecommunications Monitor Routine (Part 5 of 5)



This section describes the PTOP data transmission modules provided in System/360. The modules are:

- Controller (IKDGTCLR), which controls entry into the data transmission routines invoked by calls in the user's program.
- Initialization (IKDGTNIT), which prepares the System/360 for communication with an 1130 program.
- Read and Write Data (IKDRDWRT), which begins data transmission and reception operations requested by the user's program.
- Control Test (IKDGTCLT), which tests for completion status of a user's data transmission or data reception operation.
- End Communication (IKDGTEND), which breaks the communication link with the 1130 program.
- Interface Resolution (IKDGTIRB), which handles the information for a completed operation and determines the next step in data transmission or reception.

CONTROLLER (MODULE NAME IKDGTCLR)

Chart: HA

Function:

- Loads the unit assignment table.
- Intercepts calls to System/360 data transmission subroutines and via the unit assignment table routes control to the modules that perform the functions requested by the calls.
- Deletes the unit assignment table and frees the special save area acquired by IKDGTIRB when necessary.
- Routes control from a System/360 data transmission routine back to the user's program.

Entry: At one of the following points from a corresponding call in the user's program:

- GTNIT for initialization.
- GTEND for termination.
- GTRED for read operation.
- GTWRT for write operation.
- GTCLT for a test function.

Exit: To the calling program.

Input: In register 1, the address of a parameter list. The contents and length of this parameter list are the same as the argument list specified in the call to the data transmission subroutine.

Output: Linkage to the requested data transmission module, passing along the following information:

- In register 11, the address of the input parameter list.
- In register 3, the address of the unit assignment table.
- In register 4, the address of the GTCB (if one exists).

Operation: After saving register contents, if it had not previously done so, the Controller loads the unit assignment table when a valid data set reference number is passed by the user. For an invalid data set reference number, the Controller places a value of 2 in the "statcode" variable if GTCLT was called, or a negative value in the "initcode" variable if GTNIT was called, and returns to the calling program.

Using the value of the "datasetref" parameter as an index, the Controller then refers to the unit assignment table to determine if an appropriate GTCB exists.

If the GTCB does not exist, only a call to GTNIT from the user's mainline program is recognized, in which case the Controller passes control to IKDGTNIT. If a call is made to GTCLT when no GTCB exists, the Controller places a value of 2 in the "statcode" variable and returns to the calling program. In all other cases, the Controller returns immediately to the calling program.

The Controller passes control to the desired module after establishing the appropriate register save area for use by the data transmission routines. This depends on whether the mainline program or the asynchronous routine is in control.

Upon return of control from IKDGTNIT, the Controller tests the return code (in register 15). If the code is 4 (indicating that the initialization attempt was unsuccessful), the Controller deletes the unit assignment table and returns to the calling program.

Upon return of control from IKDRDWRT or IKDGTCLT, the Controller determines whether an asynchronous routine was in control, restores the register contents from the appropriate save area, and returns to the calling program.

Upon return of control from IKDGTEND, the Controller tests the return code (in register 15) to determine the next function it should perform, performs the function, and returns to the calling program.

The possible return codes from IKDGTEND and the functions performed by the Controller in response to them are as follows:

- 0 -- Deletes the unit assignment table because the mainline program is being ended.
- 4 -- Frees the asynchronous routine save area because the asynchronous routine is being ended.
- 8 -- Restores the mainline save area pointer because communication line groups other than the one that has been closed still exist.

INITIALIZATION (MODULE NAME IKDGTNIT)

Chart: JA

Function:

- Creates the GTCB.
- Updates an existing GTCB.
- Acquires storage for the user's read and write buffers.
- Builds and opens the data control block (DCB) that represents the communication line group.
- Builds a data event control block (DECB) for input/output operations.
- Loads other System/360 data transmission modules.
- Sets a timer for the five-minute initialization limit.

Entry: IKDGTNIT, from the Controller.

Exit: To the calling program (the Controller).

Input: The following information made available by the Controller.

- In register 3, the address of the unit assignment table.
- In register 4, the address of the GTCB associated with the data set reference number if the GTCB already exists; zero, if the GTCB does not exist.
- In register 11, the address of a parameter list that contains the following:

+ 0 A("datasetref")
address of data set reference number.

+ 4 A("writebufsize")
address of the buffer size needed for the largest amount of data to be transmitted.

+ 8 A("readbufsize")
address of the buffer size needed for the largest amount of data to be received.

+12 A("asynroutine")
address of user's System/360 asynchronous routine.

+16 A("password")
address of the password defined in System/360.

+20 A("initcode")
address of variable into which the result of the initialization attempt is to be placed.

Output: The GTCB is created and initialized, or updated. Necessary operating system control blocks are established. The user's read and write buffers are established. One of the following values is placed in the user's "initcode" variable:

- Zero, if communication linkage is established.
- Negative, if the password received from the 1130 does not match the password in the System/360, if the data set reference number is invalid, or if both "writebufsize" and "readbufsize" equal zero.
- Positive, if the communication linkage cannot be established because of transmission line error conditions, or if contact is not established with the 1130 within a certain time limit.

Operation: IKDGTNIT determines if the GTCB already exists. For an existing GTCB, IKDGTNIT updates appropriate fields with the new input information, posts a successful initialization attempt value in the "initcode" variable, and returns to the calling program.

If the GTCB does not exist, IKDGTNIT obtains and clears storage for the GTCB, places the GTCB address in the unit assignment table at a location based on the value of the data set reference number, and initializes the fields in the GTCB with the input information.

IKDGTNIT determines the total space needed for the user's read and write buffers, including control characters, and obtains storage for the buffers.

If IKDGTNIT has not already done so, it loads the IKDGTIRB module and places its address in the unit assignment table.

IKDGTNIT then opens and initializes the DCB, builds the DECB, and issues a BTAM Read Initial operation to obtain the password from the 1130 system. IKDGTNIT uses the timer to place a five-minute limit on the initialization attempt.

When the specified time limit elapses, and communication with the 1130 is not complete, IKDGTNIT deletes the IKDGTIRB module, closes the DCB, and frees the GTCB and the space obtained for the user's read and write buffers. If the GTCB count in the unit assignment table is zero, IKDGTNIT returns control to the Controller with a code of 4 in register 15. If the count is not zero, it passes a code of 0 to the Controller in register 15.

If communication with the 1130 is complete within the time limit, IKDGTNIT waits for completion of the read procedure (Read Initial followed by Read Continue or Write Reset) which is posted by IKDGTIRB in the GTCECB field in the GTCB.

Then IKDGTNIT checks the GTCB for the status of the initialization attempt and places the appropriate code in the user's "initcode" variable. If the attempt was unsuccessful, IKDGTNIT performs the same functions as it does when the time limit has elapsed. For a successful initialization, IKDGTNIT loads the IKDRDWRT and IKDGTCLT modules, if necessary, places their addresses in the unit assignment table, updates the GTCB count in the unit assignment table, and returns to the calling program.

READ AND WRITE DATA (MODULE NAME IKDRDWRT)

Chart: KA

Function: Starts a data transmission or data reception operation.

Entry: IKDRDWRT, from the Controller.

Exit: To the calling program (the Controller).

Input: The following information made available by the Controller:

- In register 4, the address of the GTCB.
- In register 6, one of the following codes that determines the operation to be performed:
  Code 0 -- data reception (read).
  Code 4 -- data transmission (write).
- In register 11, the address of the parameter list for this call.

For a read operation, this list contains the following:

+ 0 A("datasetref")
address of data set reference number.

+ 4 A("temparray")
address of the array to which the data is to be moved after it is received in the read buffer.

+ 8 A("elcount")
address of the integer that specifies the number of elements to be moved.

+12 A("datatype")
address of the integer constant that defines the format of the incoming data.

For a write operation, this list contains the following:

+ 0 A("datasetref")
address of data set reference number.

+ 4 A("temparray")
address of the array containing the data to be moved to the write buffer for transmission to the 1130 program.

+ 8 A("elcount")
address of the integer that specifies the number of elements to be transmitted.

+12 A("datatype")
address of the integer that defines the format of the outgoing data.

+16 A("controlopt")
address of the integer that specifies

the destination of the data or control information.

Output: The appropriate GTCB fields are updated to reflect the transmission status of this read or write operation and to include pertinent information about the request.

Operation: IKDRDWRT immediately returns to the calling program if one of the following cases exists:

- A read operation is pending or in progress and another read operation is requested.
- A write data operation is pending or in progress and another write data operation is requested.

IKDRDWRT places the appropriate status code for the operation in either the GTCLSTRD or the GTCLSTWR field of the GTCB and returns control to the calling program if one of the following cases exists:

- A write data operation is requested, but the 1130 user program has not called GTRED.
- A read or write operation is requested, but the 1130 program has called GTEND.

When IKDRDWRT decides that a read operation is requested, it first determines if the user's temporary array for input is in storage assigned to his task. To do this, IKDRDWRT inserts the first character from the array in a register and then stores the character back in the same position in the array. IKDRDWRT determines the end of the user's temporary array from the number of elements and the format of the data; it then inserts the last character from the array in a register and stores it back in the same position. If the user has provided an array location that is not assigned to his task, the control program will handle the protection check in the manner defined by the user's system or application. Otherwise, IKDRDWRT proceeds with its normal read request functions.

For both read and write requests, IKDRDWRT places the user-specified parameters in appropriate fields in the GTCB, sets the status code for the operation, and attempts to halt all input/output operations via BTAM's RESETPL. To ensure that control is not returned to IKDRDWRT before completion of the attempt to halt the input/output operation, IKDRDWRT tests a bit in the UCB which indicates whether the halt instruction has been issued. If an operation is in progress (that is, the System/360 is receiving or transmitting a message), and the input/output operation cannot be halted, BTAM informs IKDRDWRT via a return code. IKDRDWRT then returns to the calling program. (The read or write function will be performed by IKDGTIRB after the current input/output operation has been completed.)

If all input/output operations are stopped, IKDRDWRT performs the requested function as follows:

- For a read request, IKDRDWRT prepares the ready-to-read message for the user and transmits it to the 1130 via BTAM. If the 1130 user's program is also in ready-to-read status, IKDRDWRT sets the proper status code in the GTCLSTRD field of the GTCB.
- For a write data request ("controlopt" argument equals 1), IKDRDWRT moves the user's data to the write buffer specified by the user, adds the appropriate control information, and transmits the data message to the 1130 system via BTAM.
- For a write request that specifies the 1130 asynchronous routine is to be scheduled or that the 1130 program is to be terminated, IKDRDWRT prepares the applicable message (asynchronous routine request or program termination request) and transmits it to the 1130 system via BTAM.

After sending the appropriate message, IKDRDWRT returns to the calling program.

CONTROL TEST (MODULE NAME IKDGTCLT)

Chart: None

Function: Makes the status of the most recently requested read or write operation available to the user's problem program.

Entry: IKDGTCLT, from the Controller.

Exit: To the calling program (the Controller).

Input: In register 11, the address of a parameter list that contains the following:

+ 0 A("datasetref")
address of data set reference number.

+ 4 A("testcode")
address of the integer that specifies the type of test function desired.

+ 8 A("statcode")
address of the variable into which the status code for the tested operation is to be placed.

Output: The status code is placed in the user's "statcode" variable.

Operation: IKDGTCLT determines if the user's program specified that the routine should wait for completion of the operation before passing the status code.

When a wait is requested, IKDGTCLT tests fields in the GTCB to determine if it is feasible to wait for completion of the operation. If a status code of 2 or 3 is found, and a valid input/output operation is pending or active, IKDGTCLT waits for completion of the operation (indicated in the GTCECB field by IKDGTIRB), places the status code in the user's variable, and returns to the calling program.

In certain cases, IKDGTCLT does not wait; instead, it places the appropriate status code in the user's variable and immediately returns to the calling program. It will not wait when the following status codes are to be passed for the stated reasons:

- 1, 6, 7, or 8 -- Operation is completed successfully, completed with error, or cannot be completed.
- 2 -- PTOP is not initialized.
- 4 -- Both programs are in ready-to-read status.
- 5 -- 1130 program has called GTEND.

If the user specifies a test function for an operation not previously issued, IKDGTCLT places a status code of 2 in the user's variable and returns to the calling program.

If the user specifies a test function without waiting, IKDGTCLT merely places the appropriate status code in the user's variable and returns to the calling program.

END COMMUNICATION (MODULE NAME IKDGTEND)

Chart: LA

Function:

- Restores the transmission status of the mainline program (if an asynchronous routine is ending).
- Halts input/output operations after any pending operation is completed (if the mainline program is ending).
- Sends the appropriate message to the 1130 program.
- Deletes resident data transmission modules when necessary.
- Frees the GTCB and the user's buffer(s), and closes the DCB.

Entry: IKDGTEND, from the Controller.

Exit: To the calling program (the Controller).

Input: The following information made available by the Controller:

- In register 3, the address of the unit assignment table.
- In register 4, the address of the GTCB.

Output: In register 15, the appropriate return code to inform the Controller of its next action. Return codes and their meanings are discussed in the detailed description of the Controller.

Operation: IKDGTEND determines if a user input/output operation is started but not complete, and waits for completion if necessary. It then halts any further input/output operations (by using BTAM's RESETPL). To ensure that control is not returned to IKDGTEND before completion of the attempt to halt input/output operations, IKDGTEND tests a bit in the UCB which indicates whether the halt instruction has been issued.

IKDGTEND determines if an asynchronous routine has called it. If IKDGTEND is to end an asynchronous routine, it resets flags in the GTCB, restores the transmission status of the mainline program, transmits an asynchronous routine ended message to the 1130, and returns to the calling program with the appropriate return code in register 15.

If the mainline program called GTEND, IKDGTEND sends a terminate communication link message to the 1130, and decrements the GTCBCNT field by one in the unit assignment table. If this field becomes zero, IKDGTEND deletes the IKDRDWRT, IKDGTCLT, and IKDGTIRB modules, and clears their entry point addresses from the unit assignment table.

IKDGTEND finally closes the DCB, frees storage obtained by IKDGTNIT for the user's read and write buffers and the GTCB, sets the appropriate return code in register 15, and returns to the calling program.

If IKDGTEND determines that the 1130 user's program has already called GTEND (the communication link is broken), no message is sent, but the other ending functions (either for the asynchronous rou-

tine or the mainline program) are performed.

INTERFACE RESOLUTION ROUTINE (MODULE NAME IKDGTIRB)

Charts: MA, MB, MC

Function: After BTAM has handled the input/output interruption, IKDGTIRB performs the following as necessary:

- Maintains system ready-to-read status.
- Monitors completion of the user's input/output requests that are started by IKDRDWRT.
- Schedules the System/360 asynchronous routine for execution upon request from the 1130 program.
- Acquires storage for a save area to be used for storing the mainline program transmission status while the asynchronous routine is in control.

Entry: IKDGTIRB, via the operating system dispatcher. It is scheduled via the Asynchronous Exit Effector, Stage 2, by the channel end appendage or exceptional end appendage in BTAM.

Exit: To the calling program (operating system).

Input:

- In register 0, the address of the interruption queue element (IQE) used to schedule IKDGTIRB.
- In register 1, the address of the input/output block (IOB).

Output: The appropriate GTCB fields are updated to reflect the status of any user's read or write operation that was just completed before IKDGTIRB gained control, and to indicate the operation that IKDGTIRB currently processed.

Operation: IKDGTIRB is scheduled by the System/360 BTAM program at the completion of any input/output operation. Upon entry, IKDGTIRB locates the DECB and GTCB by using the input/output block (IOB) pointer in register 1.

If the completion code in the DECB event control block indicates that a Halt I/O was issued, IKDGTIRB immediately returns to the operating system. Otherwise, it determines the type of input/output operation that was just completed by testing the type field in the DECB. Based on the type of the completed input/output operation, IKDGTIRB performs necessary interface functions to prepare for the next operation, starts the next operation (see "Starting the Next Operation"), and returns to the operating system.

If the completed operation was Read Initial, IKDGTIRB determines if initialization was previously completed. If not, only an initialization message is recognized. IKDGTIRB tests for input/output errors and a valid initialization message. When no errors are detected, IKDGTIRB compares the password transmitted in the initialization message with the password in the System/360 program. If these two passwords agree, IKDGTIRB issues a Read Continue operation to send the proper positive acknowledgment to the 1130. If the passwords are different, or if an input/output error occurred, IKDGTIRB sets the appropriate GTCB fields and issues a Write Reset operation to send an EOT character to the 1130, thus signifying that the initialization was unsuccessful.

If System/360 PTOP has already been initialized, IKDGTIRB checks for input/output errors, for an invalid message format, or, if the 1130 program had previously called GTEND, checks for a reinitialization attempt. If the reinitialization attempt is unsuccessful or if any error condition exists, IKDGTIRB issues a Write Reset operation. Otherwise, depending on the message received, it sets appropriate fields in the GTCB and issues a Read Continue operation.

If input/output errors occurred during a user read operation, IKDGTIRB sets the appropriate status code in the GTCLSTRD field, and resets the appropriate GTCB fields to cancel the operation. If the user has issued a call to GTCLT to wait for the read operation to complete, IKDGTIRB posts the GTCECB field before issuing the Write Reset operation.

If the completed operation was Read Continue, IKDGTIRB's actions depend on the message received from the 1130, as follows:

- For an initialization message, if this is the first initialization attempt, IKDGTIRB posts the GTCECB field for IKDGTNIT, sets the OKNIT flag to one, and establishes the PTOP system ready-to-read status by issuing a Read Initial operation. If this is a reinitialization attempt, IKDGTIRB sets the GTEND flag in the GTCB to zero so that the System/360 user can perform input/output operations again, and then starts the next operation.
- For a data message, IKDGTIRB checks for input/output errors. If there are no errors (except for incorrect length

indication), it moves the data from the user's read buffer to the user's input array, omitting control characters and header information. If there are input/output errors, IKDGTIRB does not move the data from the user's read buffer. Then, IKDGTIRB sets the appropriate status code in the GTCLSTRD field, posts the GTCECB field if the user called GTCLT to wait for completion of the read operation, and starts the next operation.

- For an asynchronous routine request message, IKDGTIRB places the asynchronous routine parameters from the message into the GTCDSRAD and GTCSTART fields of the GTCB. It obtains storage for a save area, places the save area address in the GTCSVINF field of the GTCB, and stores the appropriate GTCB fields in the save area. It then creates an interruption request block (IRB) and an an interruption queue element (IQE) for the System/360 user asynchronous routine, and schedules the asynchronous routine via the Asynchronous Exit Effector, Stage 2.

  IKDGTIRB builds the parameter list (two words) for the user at the end of the IQE (IQE+16) created to schedule the user's asynchronous routine. The parameter list contains the addresses of the asynchronous routine parameters that were placed in the GTCB earlier. IKDGTIRB then inserts the address of the parameter list into the IQE parameter field (IQE+4). The supervisor places the parameter list address in register 1 before invoking the scheduled user asynchronous routine. Thus, when the user asynchronous routine gets control, register 1 points to the parameter list built by IKDGTIRB. IKDG TIRB sets the ASYNC flag to one and issues a Read Initial operation for the PTOP program.

  However, if IKDGTIRB determines that System/360 is waiting to issue a terminate communication link message or a program termination request message to the 1130, it does not honor the asynchronous routine request, but begins the termination process instead.

- For any other message, IKDGTIRB starts the next operation.

If the completed operation was Write Initial Transparent with Reset, IKDGTIRB tests for input/output errors and contention.

For a user's write data operation completed with input/output errors, IKDGTIRB places the proper status code in the GTCB, resets the GTCB fields to cancel the operation, posts the GTCECB field if the user called GTCLT to wait for completion of the write operation, and starts the next operation.

When a write contention for the communication line occurs, IKDGTIRB queues the current write operation and issues a Read Initial operation to allow the 1130 system to proceed to transmit its message.

When there is no error condition detected for this write operation, IKDGTIRB determines which message was sent, and takes action as follows:

- For a program termination request message, IKDGTIRB issues a Read Initial operation to prepare for reintialization from the 1130.

- For a data message, IKDGTIRB determines if the length of data transmitted is the same as that specified in the ready-to-read message received from the 1130, sets the GTCLSTWR field appropriately, posts the GTCECB field if the user called GTCLT to wait for completion of the write operation, and starts the next operation.

- For an asynchronous routine ended message, IKDGTIRB posts the GTCECB field for IKDGTEND, and starts the next operation for the mainline program whose status has been restored.

- For a terminate communication link message, IKDGTIRB posts the GTCECB field for IKDGTEND and returns immediately to the operating system.

- For any other message, IKDGTIRB starts the next operation.

If the completed operation was Write Reset, IKDGTIRB determines if initialization is completed. If not, IKDGTIRB posts the GTCECB field for IKDGTNIT which is waiting for completion of the initialization message, and returns immediately to the operating system. If initialization has completed, IKDGTIRB then determines if IKDGTEND is waiting to end communication. If IKDGTEND is waiting, IKDGTIRB posts the GTCECB field and returns immediately to the operating system. Otherwise, IKDGTIRB starts the next operation.

Starting the Next Operation: After performing appropriate functions for the operation just completed, IKDGTIRB determines from the fields in the GTCB which operation (if any) is pending. It tests for a pending operation and takes action in the following order:

- For a pending terminate communication link message, IKDGTIRB determines if the 1130 has previously called GTEND. If so, IKDGTIRB posts completion in the GTCECB field for IKDGTEND and returns to the operating system. If the 1130 had not previously ended communication, IKDGTIRB issues a Write Initial Transparent with Reset operation to send the message, and sets the GTCB fields appropriately.

- For a pending program termination request message, IKDGTIRB determines if the 1130 has previously called GTEND. If so, IKDGTIRB issues a Read Initial operation to await reinitialization. If the 1130 did not end communication, IKDGTIRB issues a Write Initial Transparent with Reset operation to send the message, and sets the GTCB fields appropriately.

- For a pending write operation queued because of contention, IKDGTIRB starts the operation by setting the proper flags in the GTCB, and issues the Write Initial Transparent with Reset operation for the queued write.

- For a pending asynchronous routine request message, IKDGTIRB prepares the message in the buffer, sets the GTCB fields appropriately, and issues the Write Initial Transparent with Reset operation.

- For a pending ready-to-read message, IKDGTIRB performs the same functions as for the asynchronous routine request message.

- For a pending data message, IKDGTIRB moves the user-specified data to the user's write buffer, inserts the control information, sets the GTCLSTWR status to 3, sets appropriate GTCB fields, and issues the Write Initial Transparent with Reset operation.

If there is no user operation pending, IKDGTIRB issues a Read Initial operation to establish system ready-to-read status.

FLOWCHARTS

This section contains autocharts showing the logic flow for the System/360 PTOP data transmission modules. The charts are ordered alphabetically (according to identification) in the sequence in which the routines are described. Refer to Appendix G for an explanation of the symbols used on the autocharts.

Chart HA. Controller Routine (System/360)

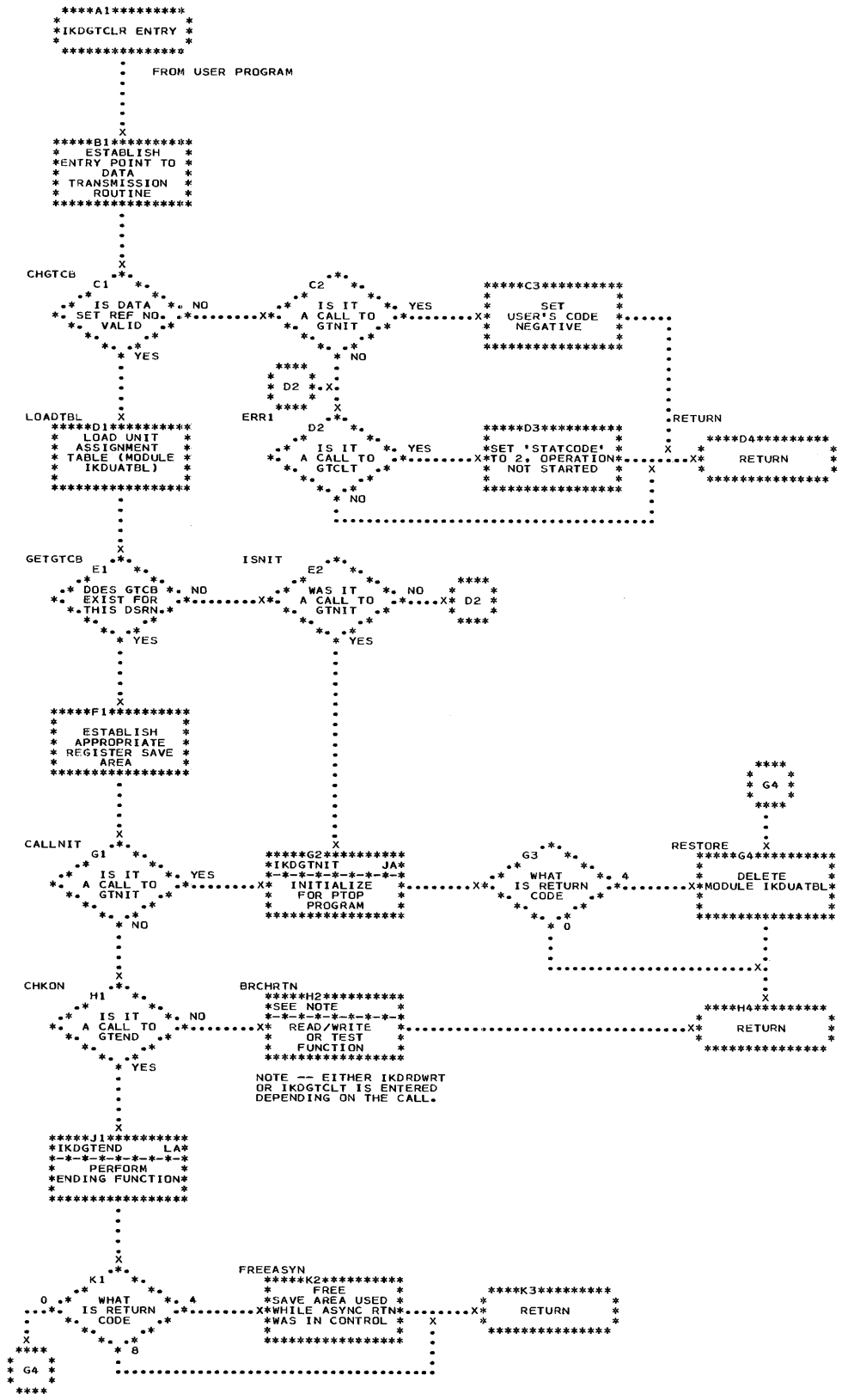


Chart JA. Initialization Routine (System/360)

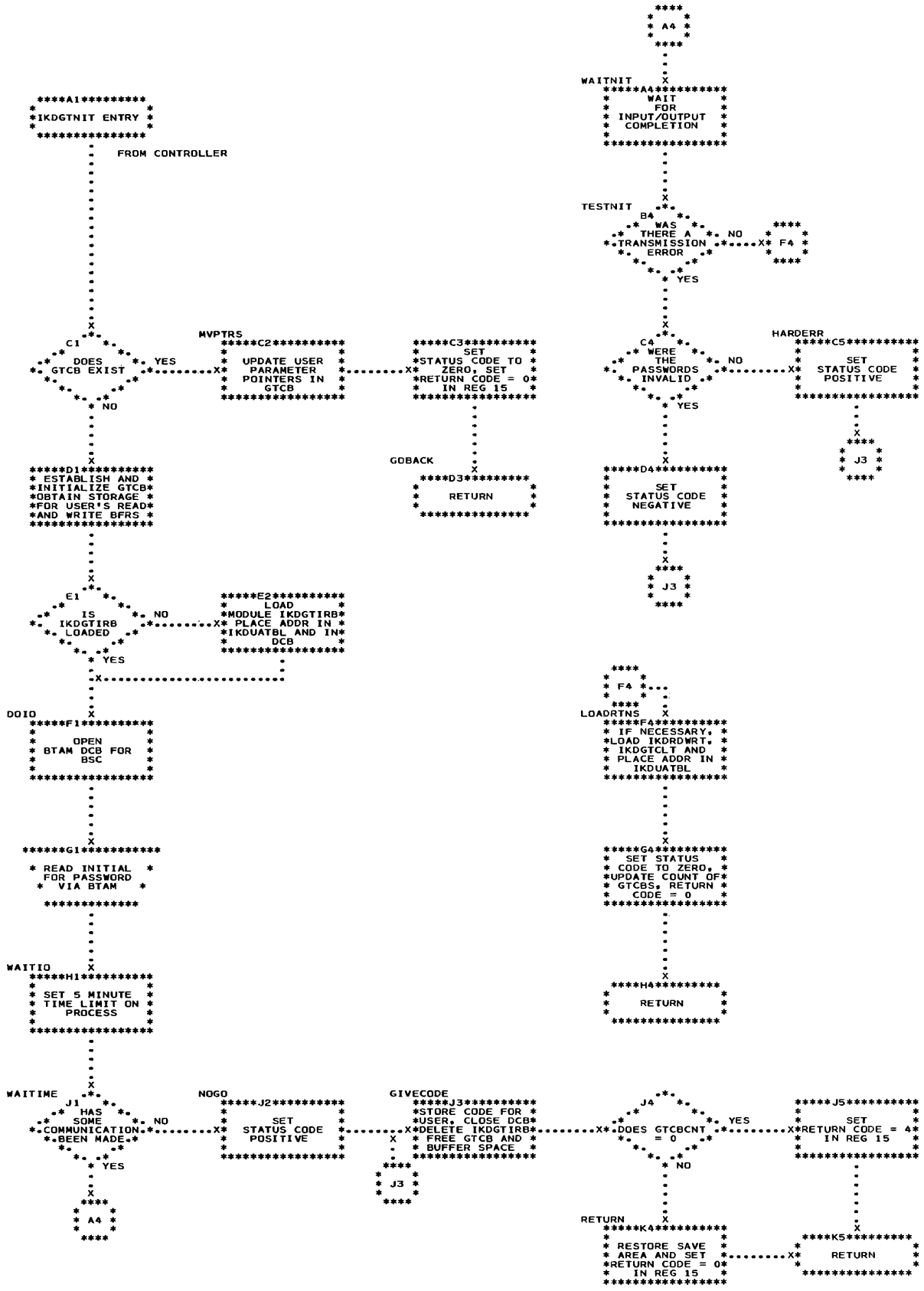


Chart KA. Read and Write Data Routine (System/360)

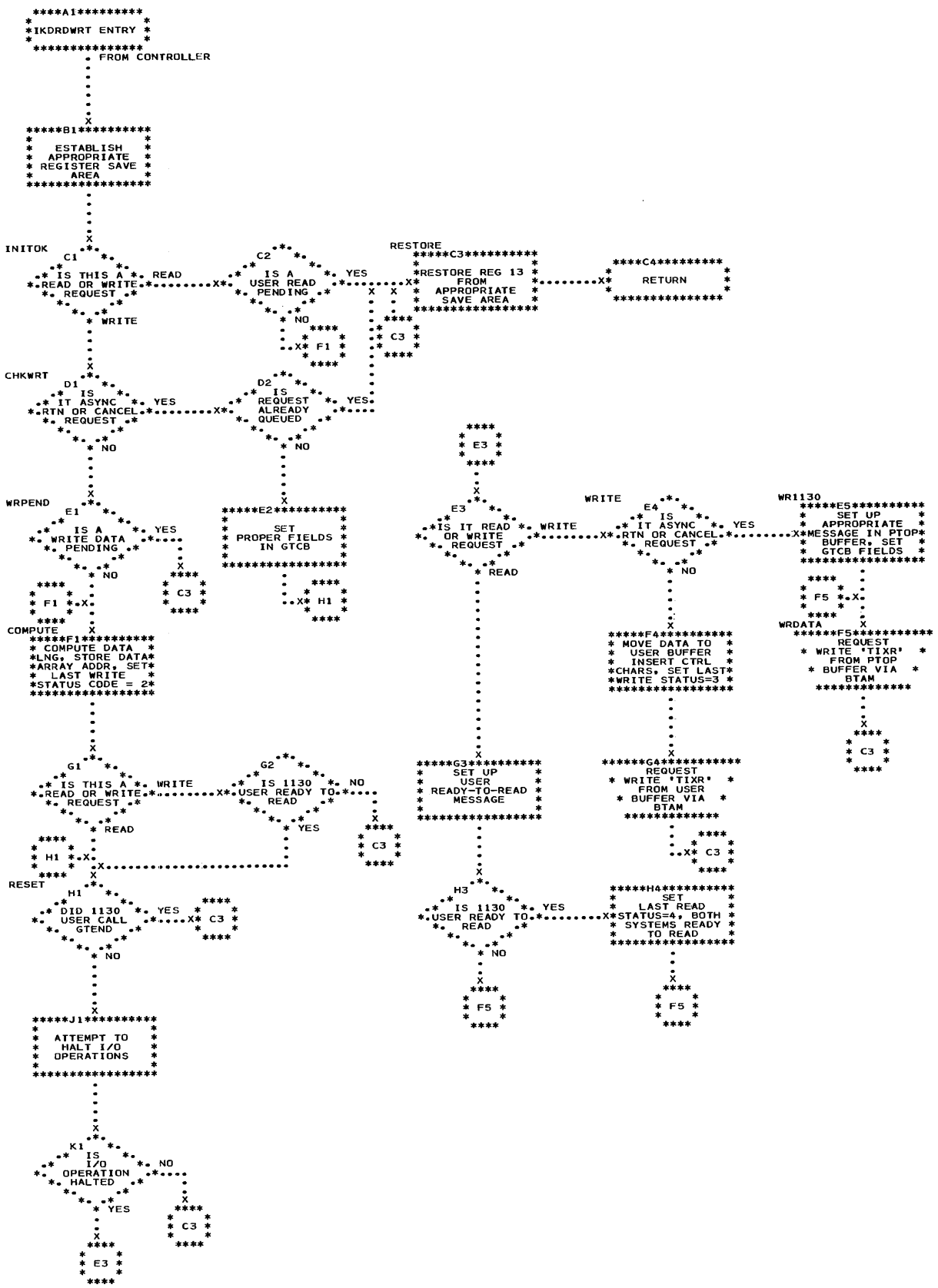


Chart LA. End Communication Routine (System/360)

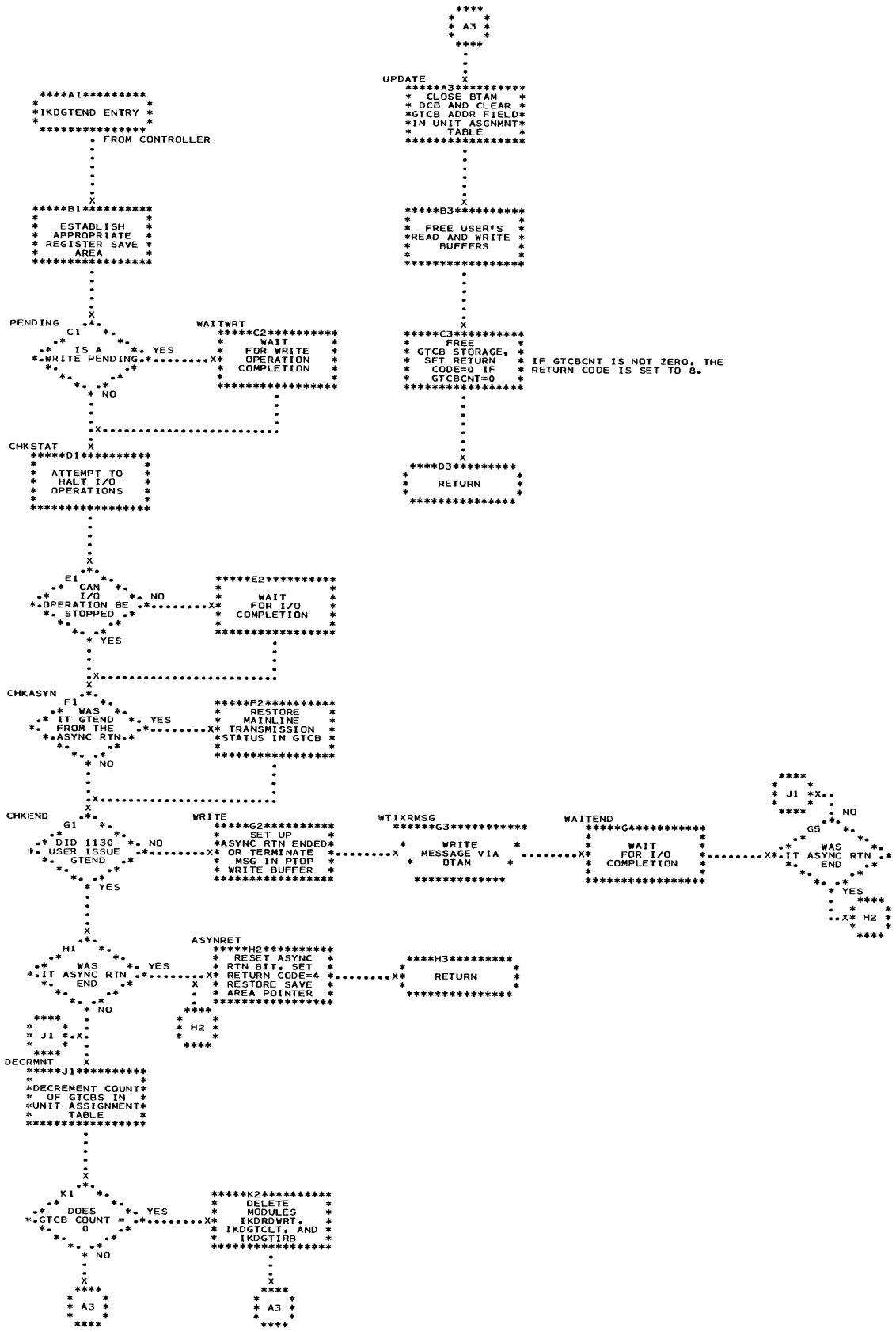


Chart MA. Interface Resolution Routine (System/360) (Part 1 of 3)

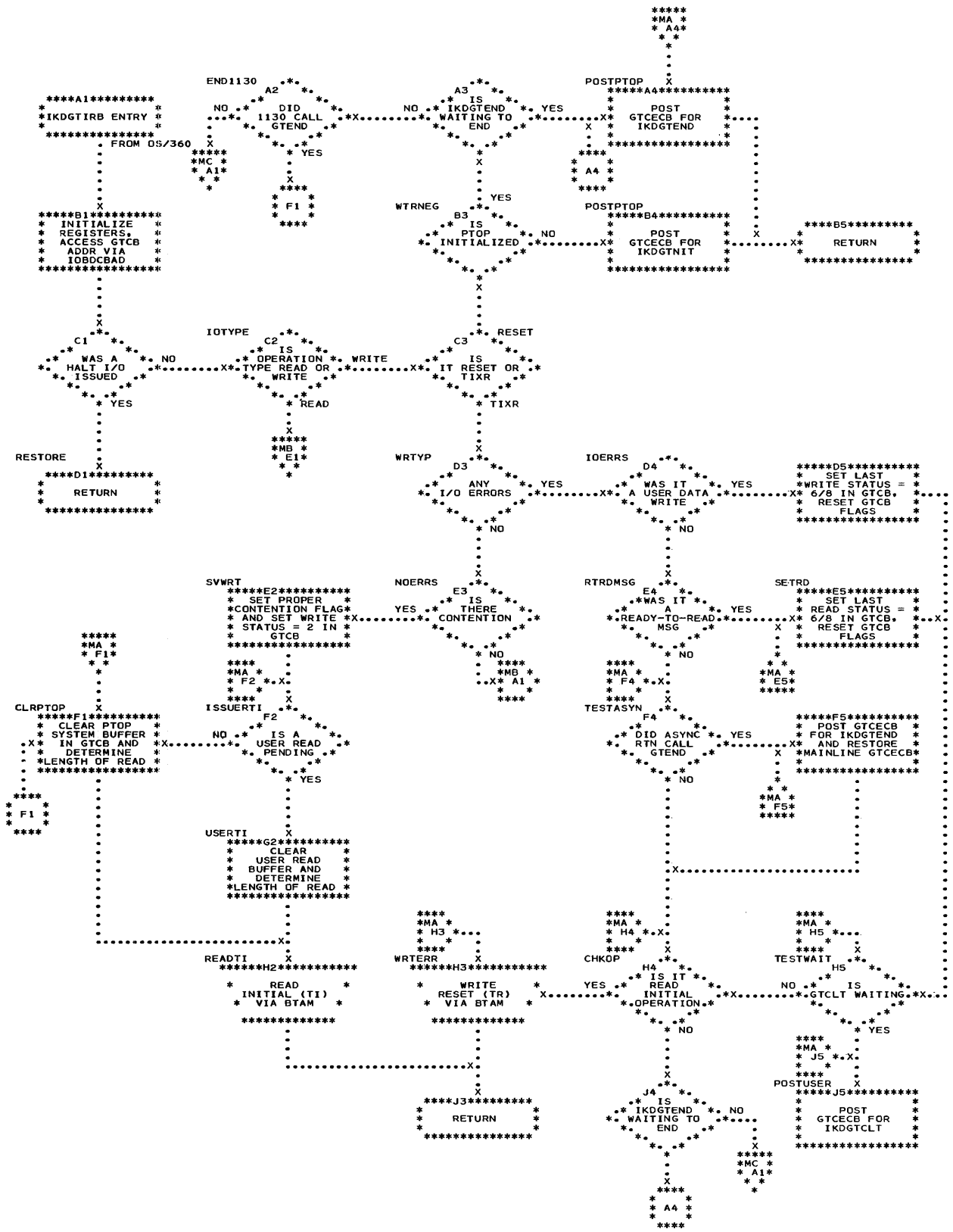


Chart MB. Interface Resolution Routine (System/360) (Part 2 of 3)

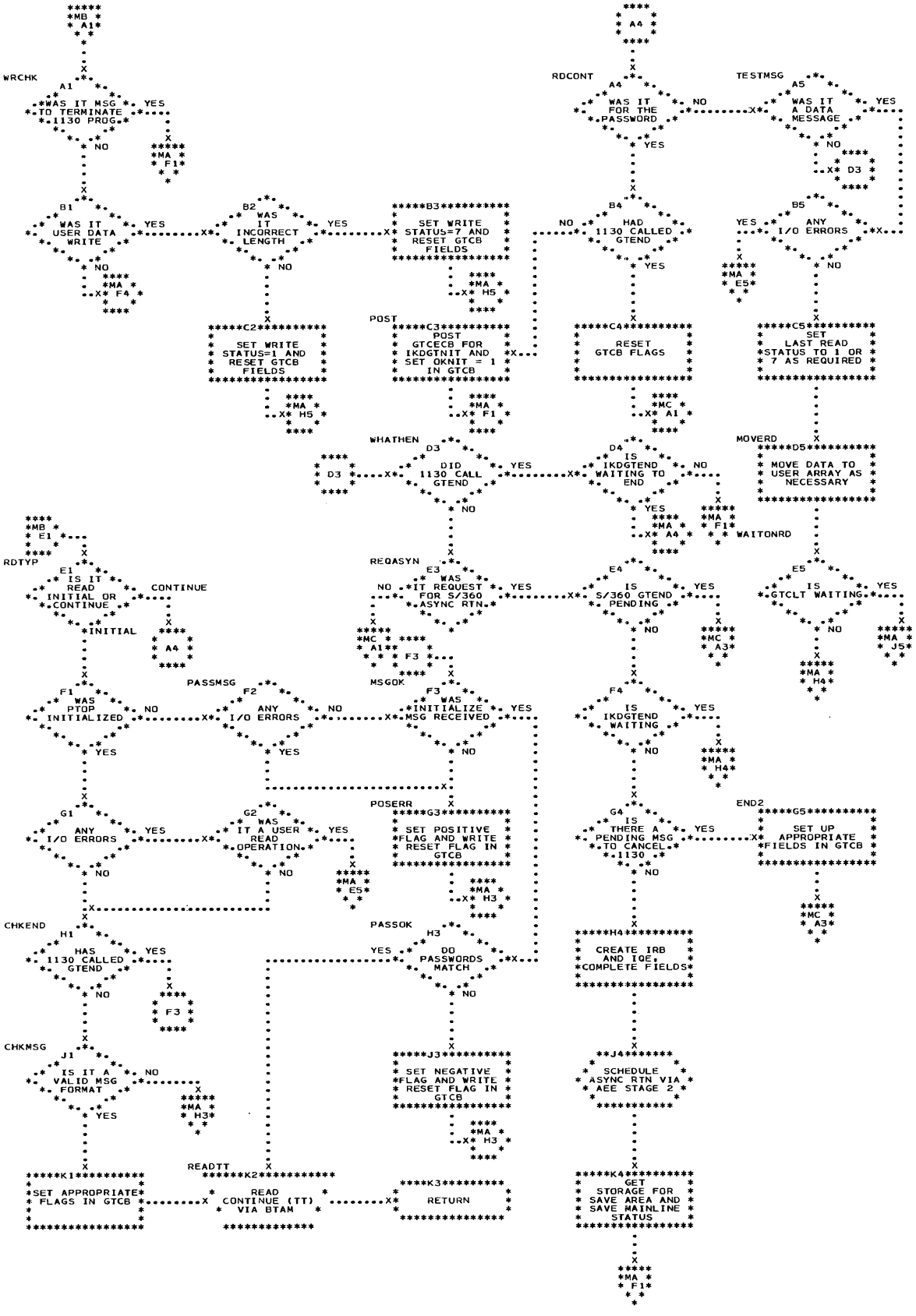
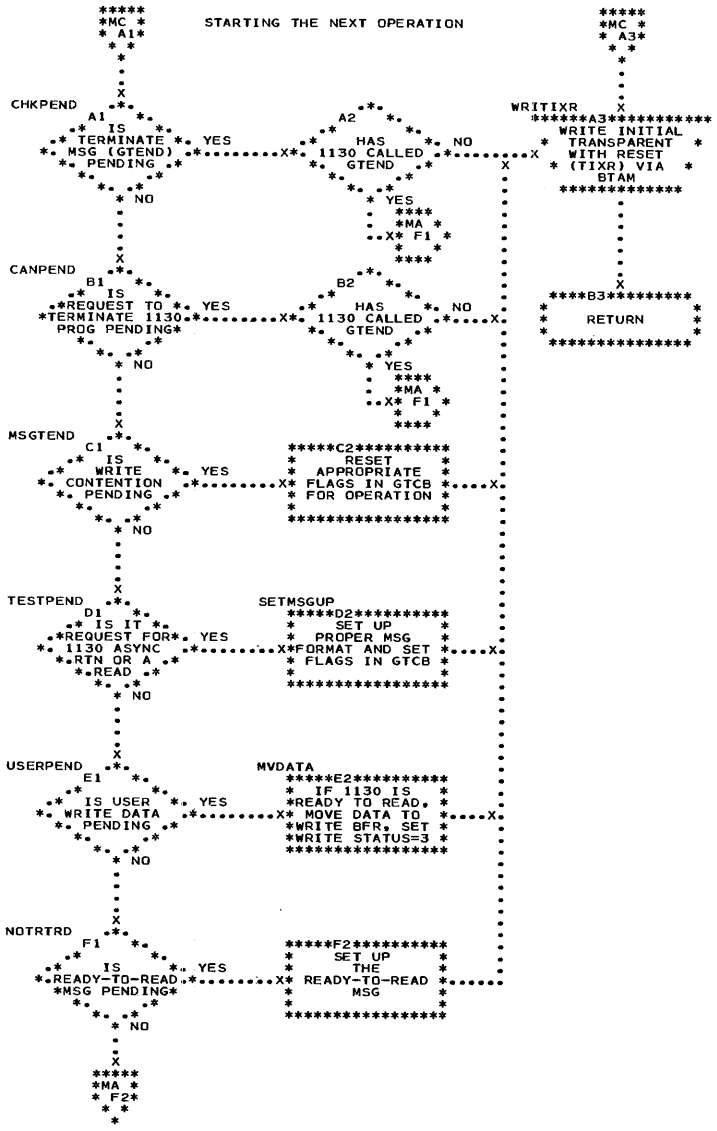


Chart MC. Interface Resolution Routine (System/360) (Part 3 of 3)



SYSTEM/360 CONVERSION ROUTINES

This section describes the PTOP data conversion routines provided in System/360. The routines are:

- Convert from System/360 Integer to 1130 Integer (IKDGDCFI),
- Convert from 1130 Integer to System/360 Integer (IKDGDCTI),
- Convert from System/360 Standard-Length Real to 1130 Standard-Precision Real (IKDGDCFF),
- Convert from 1130 Standard-Precision Real to System/360 Standard-Length Real (IKDGDCTF),
- Convert from System/360 Double-Precision Real to 1130 Extended-Precision Real (IKDGDCFE), and
- Convert from 1130 Extended-Precision Real to System/360 Double-Precision Real (IKDGDCTE).

TERMINOLOGY USED IN THIS SECTION

In this section, the term format refers to the FORTRAN formats in both systems as follows:

- 1130 one-word format = one 1130 word (2 bytes).
- 1130 standard-precision format, integer or real = two 1130 words (4 bytes).
- 1130 extended-precision real format = three 1130 words (6 bytes).
- System/360 halfword format = two bytes.
- System/360 standard-length format, integer or real = one System/360 word (4 bytes).
- System/360 double-precision real format = two System/360 words (8 bytes).

In addition, the term array refers to the format and order of FORTRAN arrays as they appear in 1130 storage or in System/360 storage.

GENERAL INFORMATION

The ordering of elements in an array and the format of the data making up an individual element differ between System/360 and the 1130 system. The conversion routines resolve these differences by changing data of the transmitting system to the order and format required of data used by the receiving system.

Elements in an 1130 array are ordered in descending storage addresses. Elements in the System/360 array are ordered in ascending storage addresses. Consequently, the elements in arrays transmitted from one system must be reordered before they can be used by the other system. Reordering is done for each transmitted array. Reformatting is done only if required.

All arrays are transmitted between the two systems in the order and format used in the 1130 system. Conversion is done only at the System/360 end of the transmission line.

Reordering Arrays

Reordering of arrays is done either between arrays, if they occupy separate storage areas, or in place, if they occupy overlapping storage areas (see "Overlapping Arrays"). Each type of reordering has variations depending upon the type of reformatting required, if any.

When reordering between arrays, the elements of the input array are moved to the output array in reverse order so that the last element of the input array is the first element of the output array (see Figure 12).

When reordering in place, a register is used to temporarily store an element from the input array that is being reordered. The reordering is done by switching the first and last elements, the second and next to last elements, and so on until the entire array has been inverted (see Figure 13).

Reformatting Elements in Arrays

Where the format of the System/360 data is the same as that of the 1130 data, the reformatting step is skipped and conversion merely consists of reordering the input array. If the formats differ, reformatting is necessary for the one system to use the data of the other system. Details of the reformatting procedures are explained in the discussion of each conversion routine. Data conversion formats are illustrated in Appendix D.

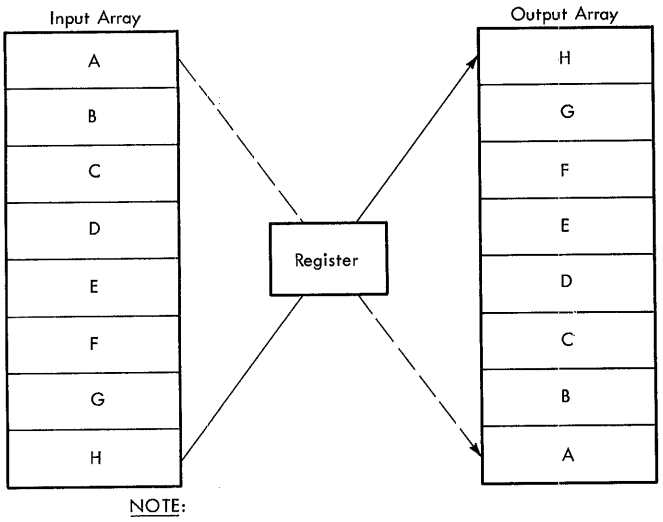


Figure 12. Reordering Between Arrays

Determining Where Conversion Is to Be Done

Before any conversion is done, the input and output arrays are compared to see if they occupy separate or overlapping storage areas. This is done by comparing the starting addresses of the input and output arrays. If these addresses are equal, the arrays coincide and are considered to be overlapped. Otherwise, the greater starting address (of either the input or the output array) is then compared with the address of the last element in the other array calculated by the data conversion routine. If the starting address is less than the last element address, the arrays overlap; if not, the arrays are entirely separate.

If the arrays occupy common storage areas, a switch is set to indicate that conversion is to be done in place. If the arrays overlap, it may be necessary to move data from the input array before any conversion is done.

Note: Conversion done in place overlays the original contents of the input array so that it is no longer valid. Conversion done between arrays does not change the data in the input array.

Overlapping Arrays

Arrays may overlap as illustrated in Figure 14 in any one of the following ways:

1. The input array coincides with the output array. No data is moved; conversion is done in place.

2. The bottom of the output array overlaps the beginning of the input array. To prevent the overlaying of yet-to-be converted data by converted data, the input array data must be moved to the top of the output array before any conversion is done. Conversion is then done in place.

3. The bottom of the input array overlaps the top of the output array. For all routines except IKDGDCFI, to prevent the overlaying of yet-to-be converted data by converted data, the input array data is moved in sections of up

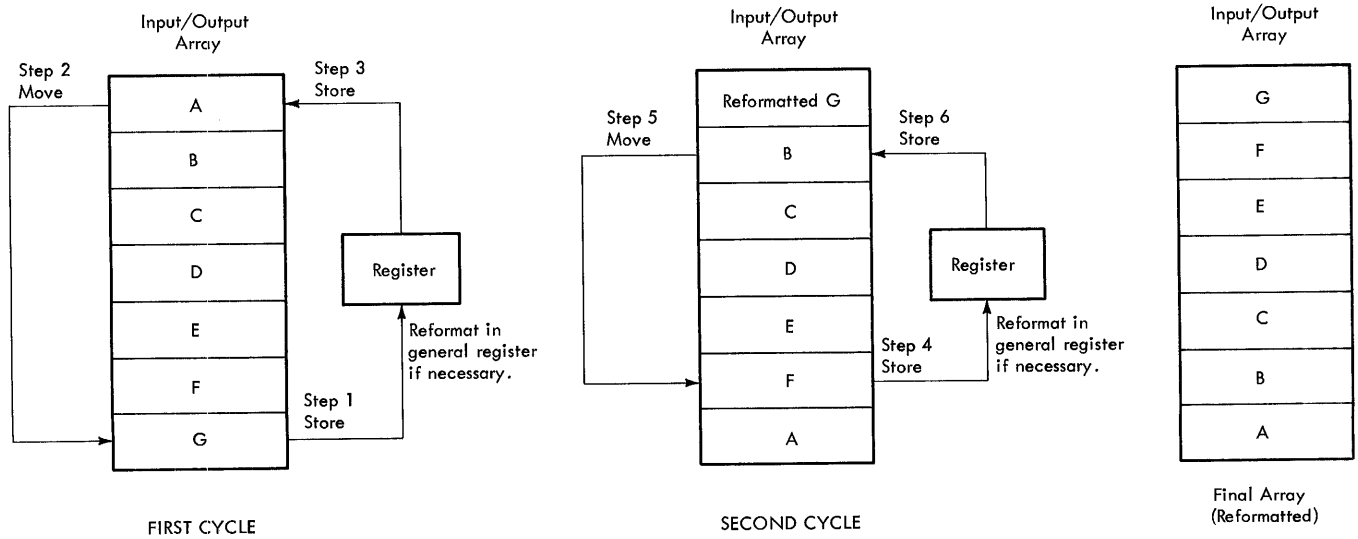


Figure 13. Reordering in Place

to 256 bytes from the bottom of the input array to the bottom of the output array. Conversion is then done in place. When this condition exists, and the IKDGDCFI routine has been invoked, conversion is done in place before the converted data is moved to the output array.

4. The output array is contained within the input array. This can occur when converting 4-byte data to 2-byte data or 8-byte System/360 double-precision real data to 1130 extended-precision real data. Conversion is done in place. The converted data is then moved to the output array.

5. The input array is contained within the output array. This can occur when converting 2-byte data to 4-byte data or 1130 extended-precision real data to System/360 double-precision real data. The input array data is moved to the top of the output array and conversion is done in place.

CONVERT FROM SYSTEM/360 INTEGER TO 1130 INTEGER (MODULE NAME IKDGDCFI)

Charts: NA, NB

Function:

- Converts an array of integers in the System/360 standard-length or halfword format to an array of integers in the 1130 integer format.

- Converts an array of alphameric data in the System/360 halfword format to an array of alphameric data in the 1130 one-word format.

Entry: GDCFI, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing the following:

+ 0 A("userarray")
address of input array containing the data to be converted.

+ 4 A("temparray")
address of output array into which the converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

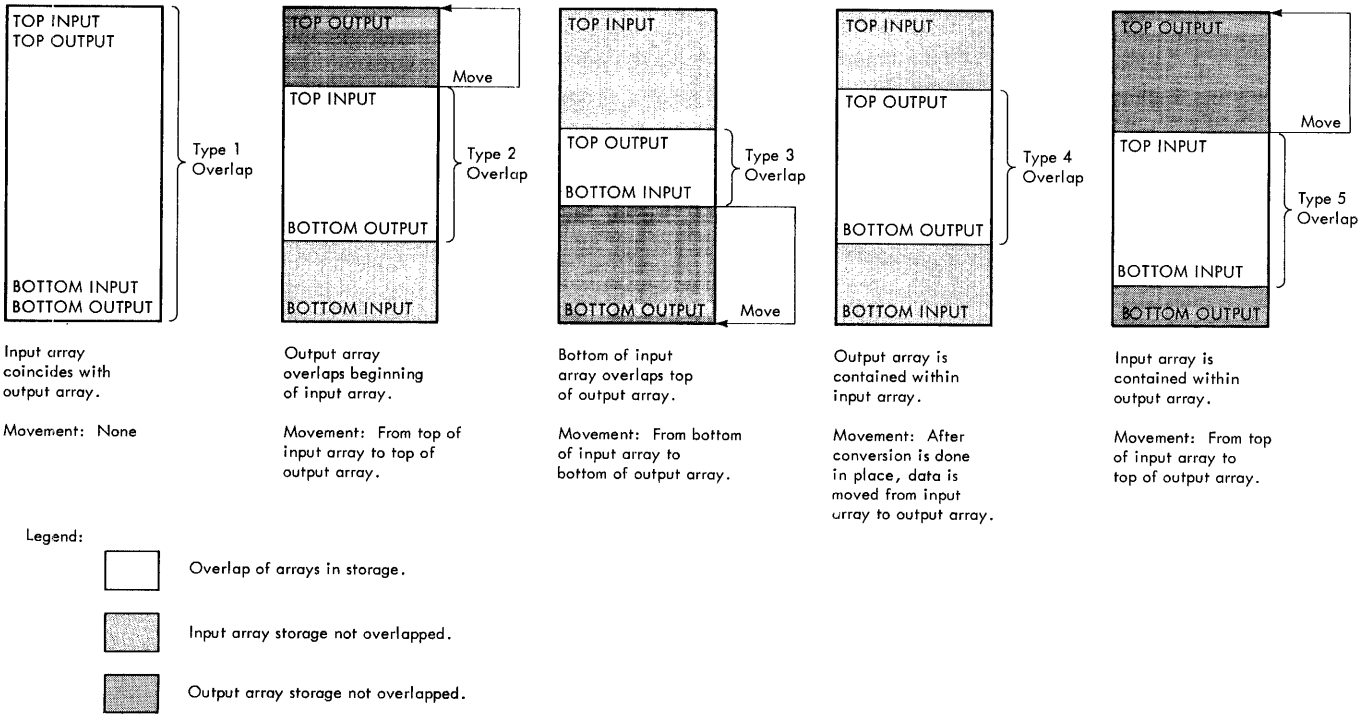


Figure 14. Overlapping Arrays and Related Data Movement

+12 A("elength")
address of the integer that designates the length of each element to be converted.

Output: An array of integers or alphameric data in the 1130 one-word (16-bit) format.

Operation: After saving register contents, IKDGDCFI determines the number of elements to be converted by checking the "elcount" argument. If the number of elements to be converted is zero or less, IKDGDCFI immediately returns to the calling program.

Next, if necessary, IKDGDCFI moves the input array data so that both arrays begin at the same location. It then determines whether conversion is to be done in place or between arrays (see "Determining Where Conversion is to be Done").

At this point, the format of the input data is determined by testing the value of the "elength" argument.

If the input data is in System/360 halfword integer or halfword alphameric format, conversion merely involves reordering the elements either between arrays as shown in Figure 12, or in place as shown in Figure 13.

If the input data is in System/360 standard-length format, conversion involves reformatting each element and then reordering it in the output array. Integers outside the range -32,768 to +32,767 cannot be expressed in the 1130 integer formats. IKDGDCFI assumes that the System/360 input data falls within these limits, and ignores the 16 high-order bits of the input data elements. Thus, the reformatting process reduces each 4-byte element to a 2-byte element by retaining only the sign and the 15 low-order bits.

When reordering reformatted data between arrays, each reformatted element starting with the last is put in sequential halfwords of the output array as shown in Figure 15. Thus, each halfword in the output array will contain the equivalent of an 1130 word of data. This makes it convenient to transmit only significant data to the 1130.

When reordering reformatted data in place, it is accomplished in two phases as shown in Figure 16.

In phase one, the last element of the input array is reformatted in a register (step 1). The first element of the array is then moved into the last location (step

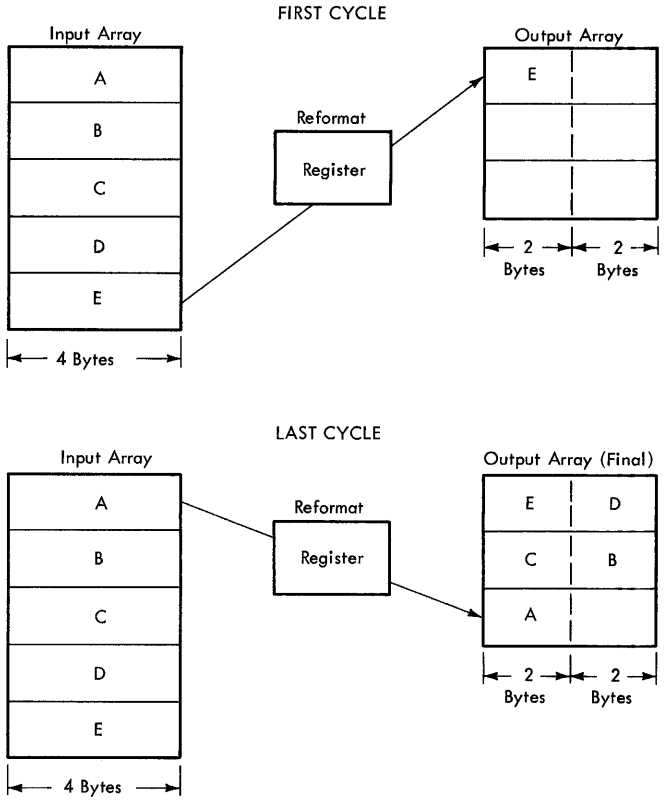


Figure 15. IKDGDCFI Reformatting Between Arrays (From System/360 Integer to 1130 Integer)

2); the reformatted last element is moved into the first halfword of the array (step 3). Then, the next-to-last element is reformatted in the register (step 4), the second and third halfwords of the array (containing meaningless information) are moved into the next-to-last word in the array (step 5); and the newly reformatted element is placed in the second halfword of the array (step 6). This process is continued until the address of the next element to be reformatted is equal to or less than the address where the last reformatted element was stored. At this point, phase two begins.

In phase two, the elements which have not been reformatted have already been reordered by phase one. Between each of these elements is an irrelevant word inserted by phase one. Phase two reformats the elements in the order they are located in the array (step 1) and moves them to the locations in the array following the reformatted elements from phase one (step 2).

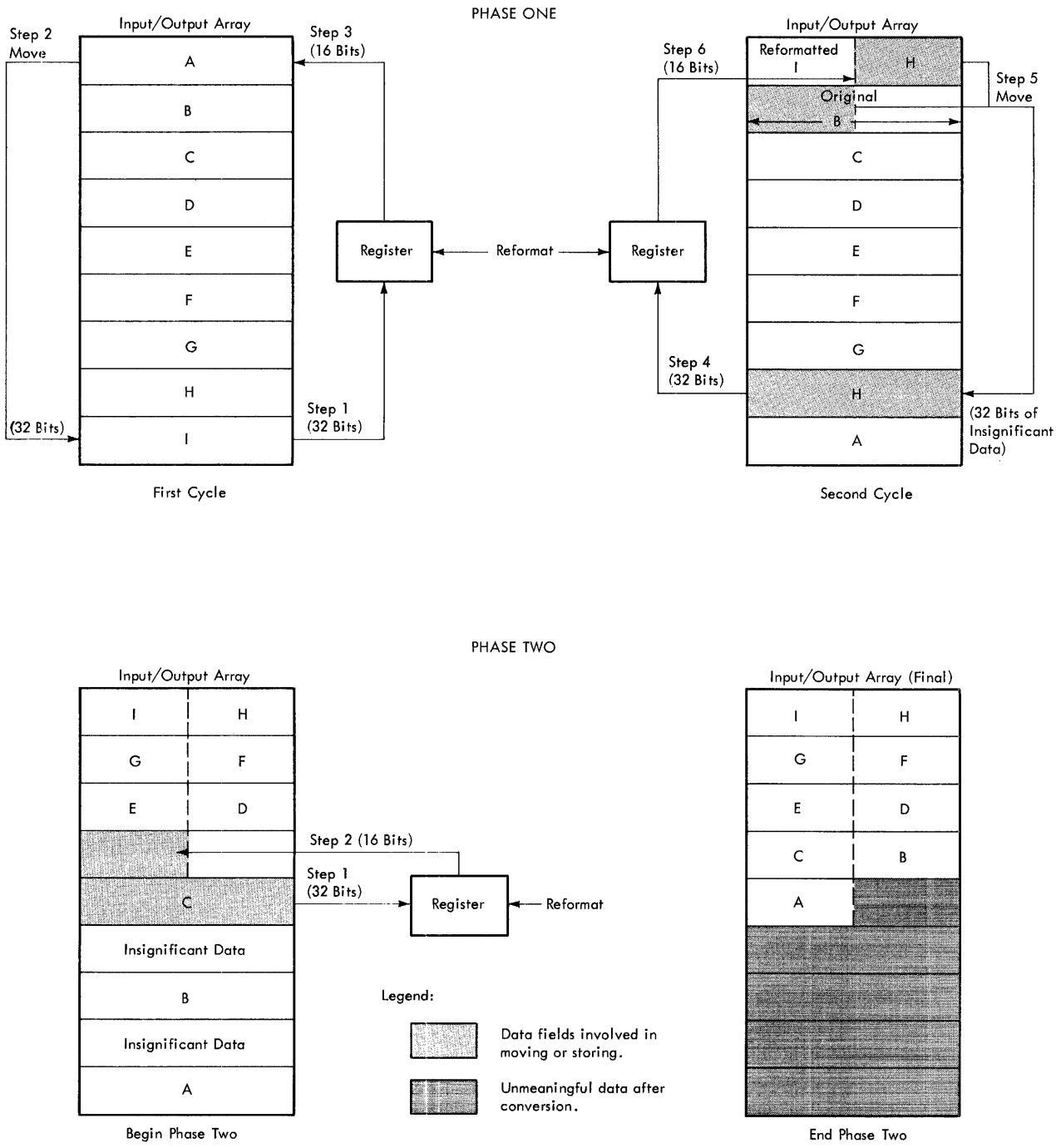


Figure 16. IKDGDCFI Reformatting in Place (From System/360 Integer to 1130 Integer)

CONVERT FROM 1130 INTEGER TO SYSTEM/360 STANDARD-LENGTH OR HALFWORD INTEGER (MODULE NAME IKDGDCTI)

Charts: OA, OB, OC

Function:

- Converts an array of integers in the 1130 format to an array of integers in either the System/360 standard-length of halfword format.
- Converts an array of alphameric data in the 1130 format to an array of alphameric data in the System/360 halfword format.

Entry: GDCTI, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing information as follows:

+ 0 A("temparray")
address of input array containing the data to be converted.

+ 4 A("userarray")
address of output array into which converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

+ 12 A("elength")
address of the integer that designates the length of each converted element.

Output: An array of integers in the System/360 standard-length of halfword format, or an array of 16-bit alphameric data in the System/360 format.

Operation: After saving register contents, IKDGDCTI determines the number of elements to be converted by checking the "elcount" argument. If the number of elements to be converted is zero or less, IKDGDCTI immediately returns to the calling program.

Next, if necessary, IKDGDCTI moves the input array. It then determines whether conversion is to be done in place or between arrays. This is done as described in "Determining Where Conversion is to be Done" except when the bottom of the input array overlaps the top of the output array, and the output data is to be in System/360 standard-length format. IKDGDCTI determines the format of the output array elements by testing the "elength" argument.

If the bottom of the input array overlaps the top of the output array, and the output data is to be in System/360 standard-length format, the input array is inverted in place and then reformatted between arrays as shown in Figure 17. The bottom element of the inverted input array is reformatted in a register and stored in the bottom word of the output array (steps 1 and 2). Then the next-to-last element of the inverted input array is reformatted and stored in the next-to-last word of the output array and so on until all the elements have been converted.

Except for the case described in the preceding paragraph, if the output data is to be in System/360 standard-length format, elements are reordered in the output array one at a time as they are reformatted. The reformatting process expands each two-byte input element by propagating the sign bit.

When conversion is to be done between arrays and reformatting is necessary, each element is reformatted (starting with the first one in the input array) and then put in sequential words of the output array (starting at the bottom of the array). This is similiar to the reformatting and reordering process depicted in Figure 12, except that two or three elements may be reformatted at a time before they are reordered in the output array.

When the elements are to be converted in place, the reordering is done as shown in Figure 18. This procedure converts an array of any length in place without the loss of any data trhough overlapping. The first and second elements of the array are reformatted in registers and stored at the bottom of the output array (steps 1 through 4). The last element of the input array is then reformatted and stored at the top of the input array (steps 5 and 6). This completes the first cycle. In the second cycle, the third and fourth elements are reformatted and stored in the third and fourth words from the bottom of the output array (steps 7 through 10). The next-to-last element of the input array is then reformatted and stored in the second word of the array (steps 11 and 12). This process continues until all the elements have been converted (steps 13 and 14).

If the output data is to be in System/360 halfword integer or alphameric format, conversion merely involves reordering the elements in the input array. This is done as shown in Figure 12. When reordering in place, IKDGDCTI saves elements from the bottom of the input array instead of from the top. This is shown in Figure 13.

CONVERT FROM SYSTEM/360 STANDARD-LENGTH REAL TO 1130 STANDARD-PRECISION REAL (MODULE NAME IKDGDCFF)

Charts: PA, PB

Function:

- Converts an array of real numbers in the System/360 standard-length format to an array of real numbers in the 1130 standard-precision format.
- Converts an array of alphameric data from the System/360 standard-length format to an array of alphameric data in the 1130 standard-precision format.

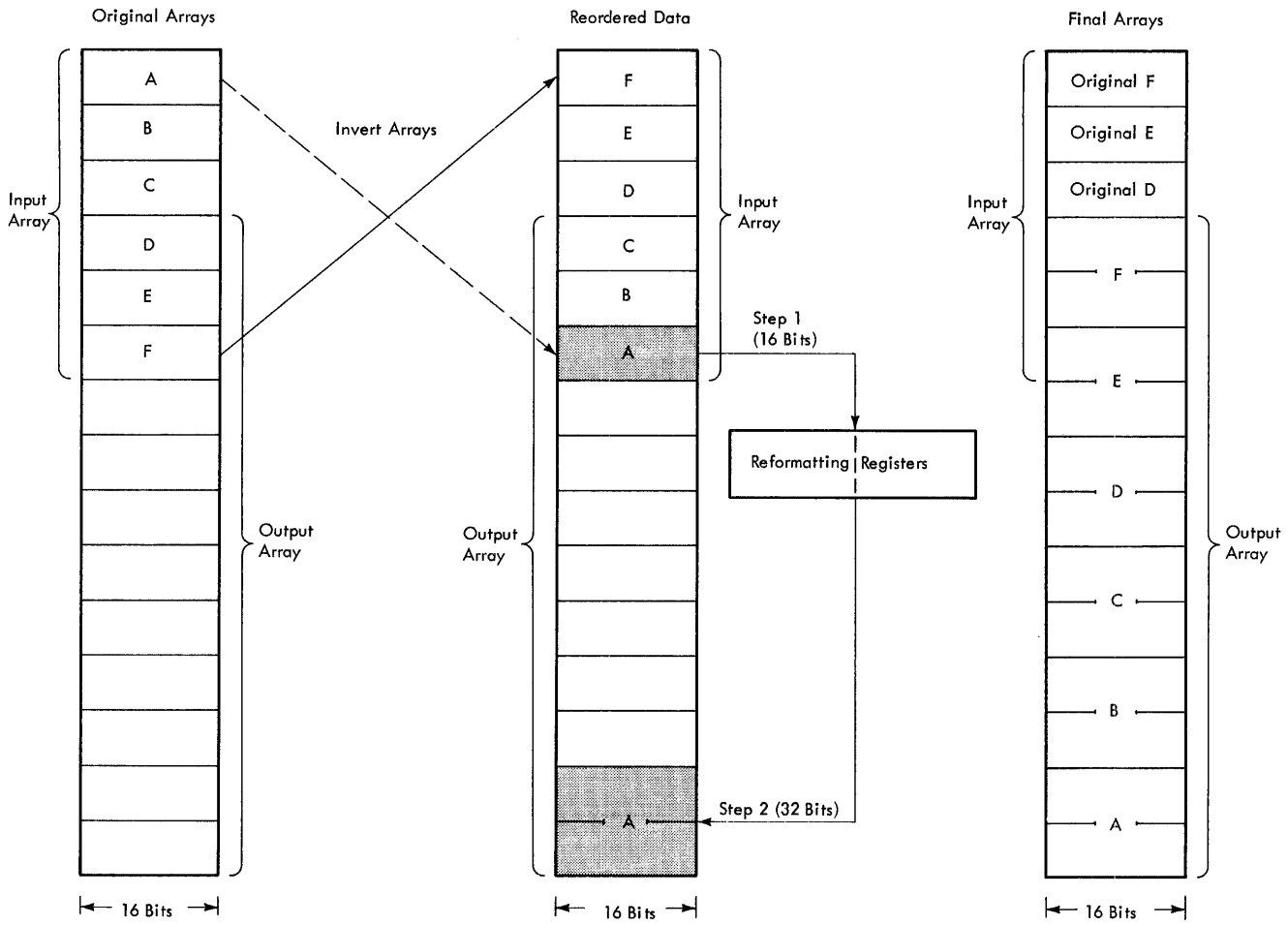


Figure 17. IKDGDCTI Reordering and Reformatting Between Overlapping Arrays (From 1130 Integer to System/360 Standard-Length Integer)

Entry: GDCFF, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing the following:

+ 0 A("userarray")
address of input array containing data to be converted.

+ 4 A("temparray")
address of output array into which the converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

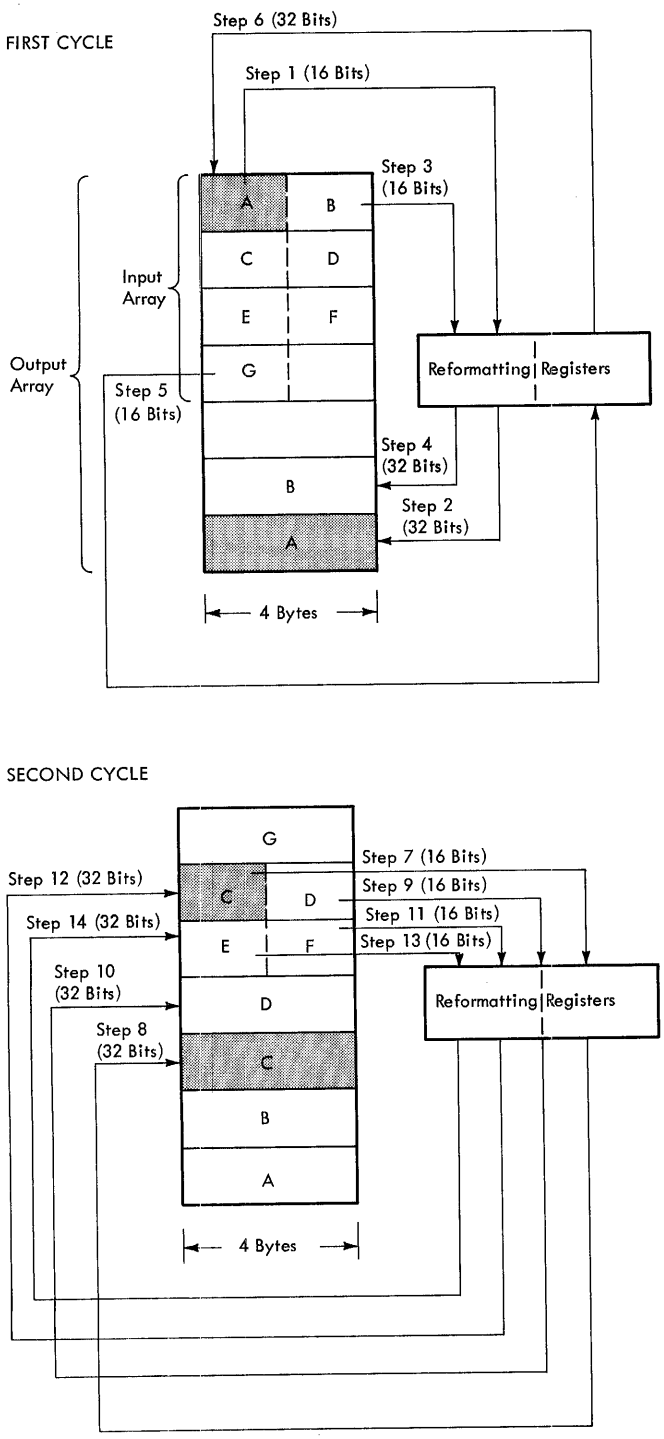
+12 A("typecon")
address of the integer that designates the type of conversion to be done.

Output: An array of real numbers or alphameric data in the 1130 standard-precision format.

Operation: After saving register contents, IKDGDCFF determines the number of elements to be converted by checking the "elcount" argument. If the number of elements to be converted is zero or less, IKDGDCFF immediately returns to the calling program.

Next, if necessary, IKDGDCFF moves the input array. It then determines whether conversion is to be done in place or between arrays (see "Determining Where Conversion is to be Done").

At this point, IKDGDCFF determines the kind of conversion required by testing the value of the "typecon" argument. If the conversion involves alphameric data, conversion merely consists of reordering the elements in the array as shown in Figures 12 and 13.



NOTE:

Shaded areas represent the first input data handled and its associated output data for both cycles.

Figure 18. IKDGDCTI Reordering and Reformatting in Place (From 1130 Integer to System/360 Standard-Length Integer)

If the conversion involves real data, the elements must be reformatted as well as reordered.

To reformat real data, IKDGDCFF isolates the sign bit, the characteristic, and the fraction of an input element in registers. It then normalizes the fraction portion and changes the characteristic to a binary excess 128 characteristic. The resulting fraction, characteristic, and sign bit are then stored in the appropriate portions of the output element. Conversion of the 24-bit System/360 hexadecimal fraction to the 23-bit binary fraction causes the loss of one low-order bit when there is a one in the high-order bit of the System/360 fraction to start with and no normalization is required.

If the reordering of the reformatted data is to be done in place, it is done in two phases as shown in Figure 19.

In phase one, the first element of the input array is reformatted (step 1). Then, to prevent overlay, the last element of the input array is moved into the first location (step 2), and the reformatted first element is stored in the last location (step 3). The same sequence is followed with the second and next-to-last elements and so on until the middle of the array is reached.

At this point phase two begins. The elements which remain to be converted have already been properly ordered by phase one. Phase two simply takes one element at a time starting at the middle of the array, reformats it, and stores it back in the same location from which it was taken (steps 1 and 2).

If reordering of elements is to be done between arrays, data is reordered as shown in Figure 12.

CONVERT FROM 1130 STANDARD-PRECISION REAL TO SYSTEM/360 STANDARD-LENGTH REAL (MODULE NAME IKDGDCTF)

Charts: PA, PB

Function:

- Converts an array of real numbers in the 1130 standard-precision format to an array of real numbers in the System/360 standard-length format.
- Converts an array of 32-bit alphameric data in the 1130 format to an array of alphameric data in the System/360 standard-length format.

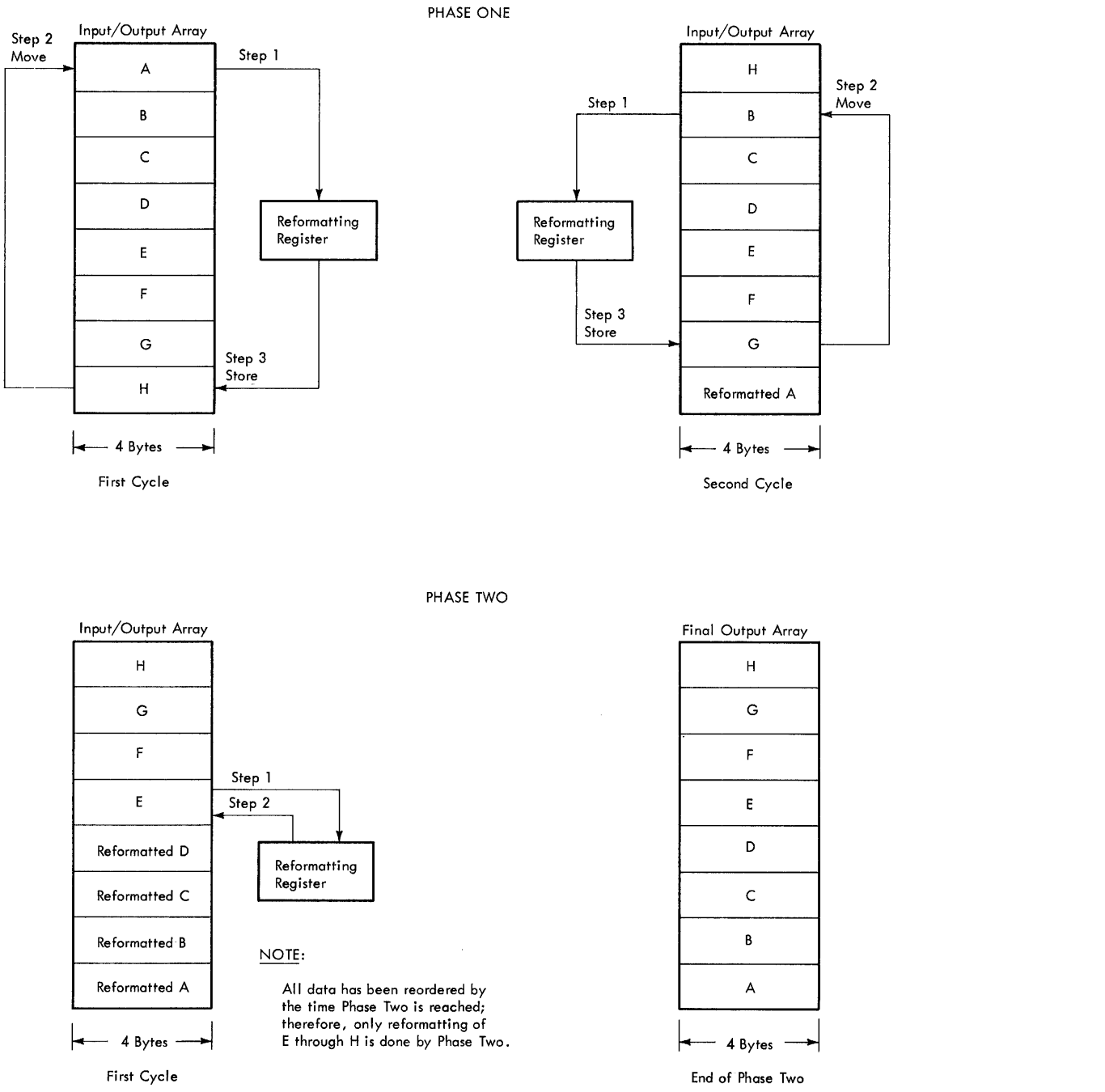


Figure 19. IKDGDCFF and IKDGDCTF Reordering in Place (From System/360 Standard-Length Real to 1130 Standard-Precision Real and Vice Versa)

Entry: GDCTF, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing the following:

+ 0 A("temparray")
address of input array containing the data to be converted.

+ 4 A("userarray")
address of output array into which the converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

+12 A("typecon")
address of the integer that designates the type of conversion to be done.

Output: An array of real numbers or alphameric data in System/360 standard-length format.

Operation: Is the same as described previously for IKDGDCFF, except for the process of reformatting real data.

To reformat real data, IKDGDCTF isolates the sign bit, the characteristic, and the fraction of an input element in registers. It then changes the characteristic from a binary excess 128 characteristic to a hexadecimal excess 64 characteristic and reduces the fraction proportionately. Normalization of the 23-bit 1130 binary fraction to a System/360 24-bit hexadecimal fraction may cause a loss of up to two low-order bits in each converted number.

CONVERT FROM SYSTEM/360 DOUBLE-PRECISION REAL TO 1130 EXTENDED-PRECISION REAL (MODULE NAME IKDGDCFE)

Charts: QA

Function: Converts an array of real numbers in the System/360 double-precision format to an array of real numbers in the 1130 extended-precision format.

Entry: GDCFE, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing the following:

+ 0 A("userarray")
address of input array containing the data to be converted.

+ 4 A("temparray")
address of output array into which converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

Output: An array of real numbers in the 1130 extended-precision format.

Operation: After saving register contents, IKDGDCFE determines the number of elements to be converted by checking the "elcount" argument. If the number of elements to be converted is zero or less, IKDGDCFE immediately returns to the calling program.

IKDGDCFE determines if conversion is to be done in place or between arrays (see "Determining Where Conversion is to be Done"). If it is to be done between arrays, IKDGDCFE reformats the elements in registers beginning with the last element of the input array, and reorders them in the output array.

If reordering and reformatting are to be done in place, it is accomplished in two phases as shown in Figure 20.

In phase one, the last element of the input array is reformatted in registers (step 1). Then, the first element is moved to the last element location (step 2) to prevent overlaying the data in the first element. Next, the reformatted last element is stored in the first six bytes of the input array (step 3). Then, the next-to-last element is reformatted (step 4), and the second element of the input array is moved into the next-to-last location (step 5). The reformatted element is stored in the next six bytes after the last reformatted element (step 6). This process continues until the middle of the array is reached at which point phase two begins.

In phase two, the elements left to be reformatted have already been reordered by phase one. Each unreformatted element is taken in turn (step 1), reformatted, and put into the next 6-byte location after the last converted element (step 2).

When all the elements have been reordered and reformatted in the input array, the converted data is moved to the output array. Since the size of the fraction portions differ between the two formats, as many as 25 low-order bits may be lost in performing the conversion.

Reformatting is done in the same way as described in the discussion of the IKDGDCFF module, except that the size of the input fraction and the positioning of the reformatted characteristic, fraction, and sign bit differ for the two modules as shown in Appendix D.

CONVERT FROM 1130 EXTENDED-PRECISION REAL TO SYSTEM/360 DOUBLE-PRECISION REAL (MODULE NAME IKDGDCTE)

Charts: RA

Function: Converts an array of real numbers in the 1130 extended-precision format to an array of real numbers in the System/360 double-precision format.

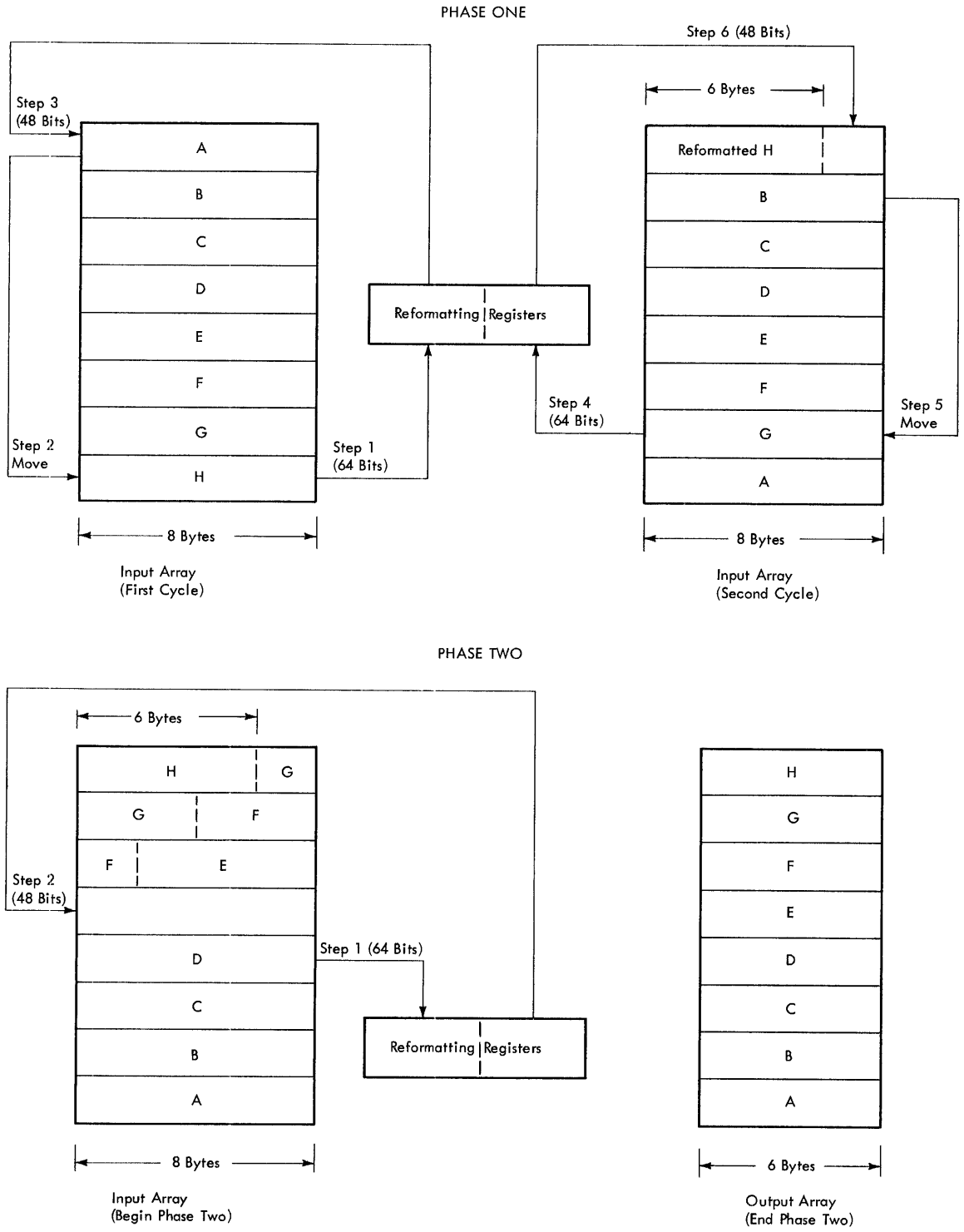


Figure 20. IKDGDCFE Reordering and Reformatting (From System/360 Double-Precision Real to 1130 Extended-Precision Real)

Entry: GDCTE, from a call in the user's program.

Exit: To the calling program.

Input: In register 1, the address of a parameter list containing the following:

+ 0 A("temparray")
address of input array containing the data to be converted.

+ 4 A("userarray")
address of output array into which the converted data is to be placed.

+ 8 A("elcount")
address of the integer that designates the number of elements to be converted.

Output: An array of real numbers in the System/360 double-precision format.

Operation: After saving register contents, IKDGDCTE determines the number of elements to be converted by checking the "elcount" argument. If the number of elements to be converted is zero or less, IKDGDCTE immediately returns to the calling program.

IKDGDCTE determines if conversion is to be done in place or between arrays (see "Determining Where Conversion is to be Done"). If it is to be done between arrays, IKDGDCTE reformats the elements in registers beginning with the last element of the input array, and reorders them in the output array.

When IKDGDCTE determines that the input array and the output array overlap, it may be necessary to move the input data before conversion is done.

If the input array has a greater starting address than the output array, the input data is moved so that the first input element is at the beginning of the output array before conversion takes place. This is shown in Figure 14 (Type 2 and Type 5).

Otherwise, each element of the input array, starting with the last, is reformatted and stored in the output array, starting at the bottom.

Reformatting is done as described in the discussion of the IKDGDCTF module, except that the size of the fraction portion of the output elements and the format of the input elements differ for the two modules. The formats are shown in Appendix D.

FLOWCHARTS

This section contains autocharts showing the logic flow for the System/360 conversion routines. The charts are ordered alphabetically (according to identification) in the sequence in which the routines are described. Refer to Appendix G for an explanation of the symbols used on the autocharts.

Chart NA. GDCFI Routine (Part 1 of 2)

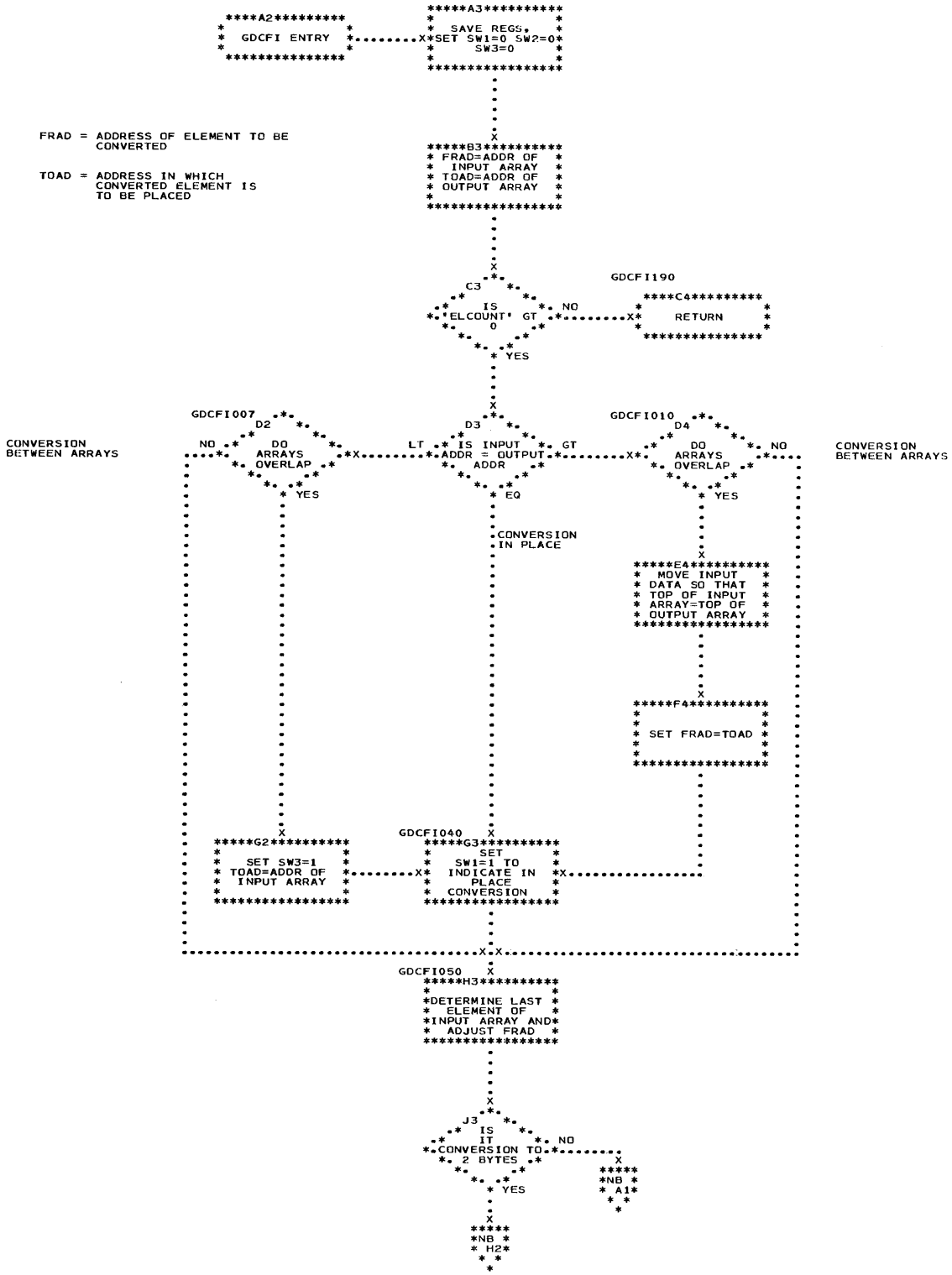


Chart NB. GDCFI Routine (Part 2 of 2)

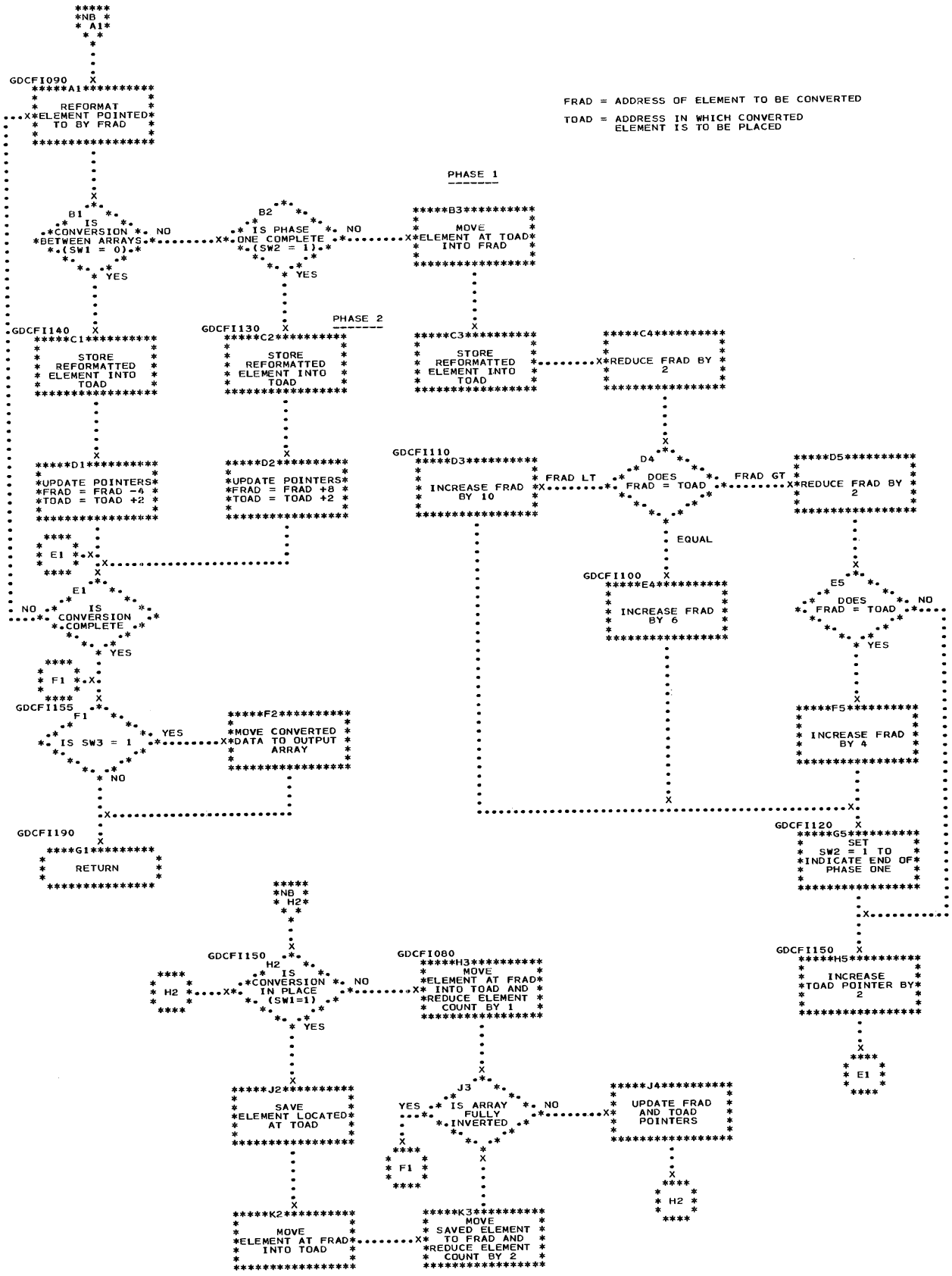


Chart OA. GDCTI Routine (Part 1 of 3)

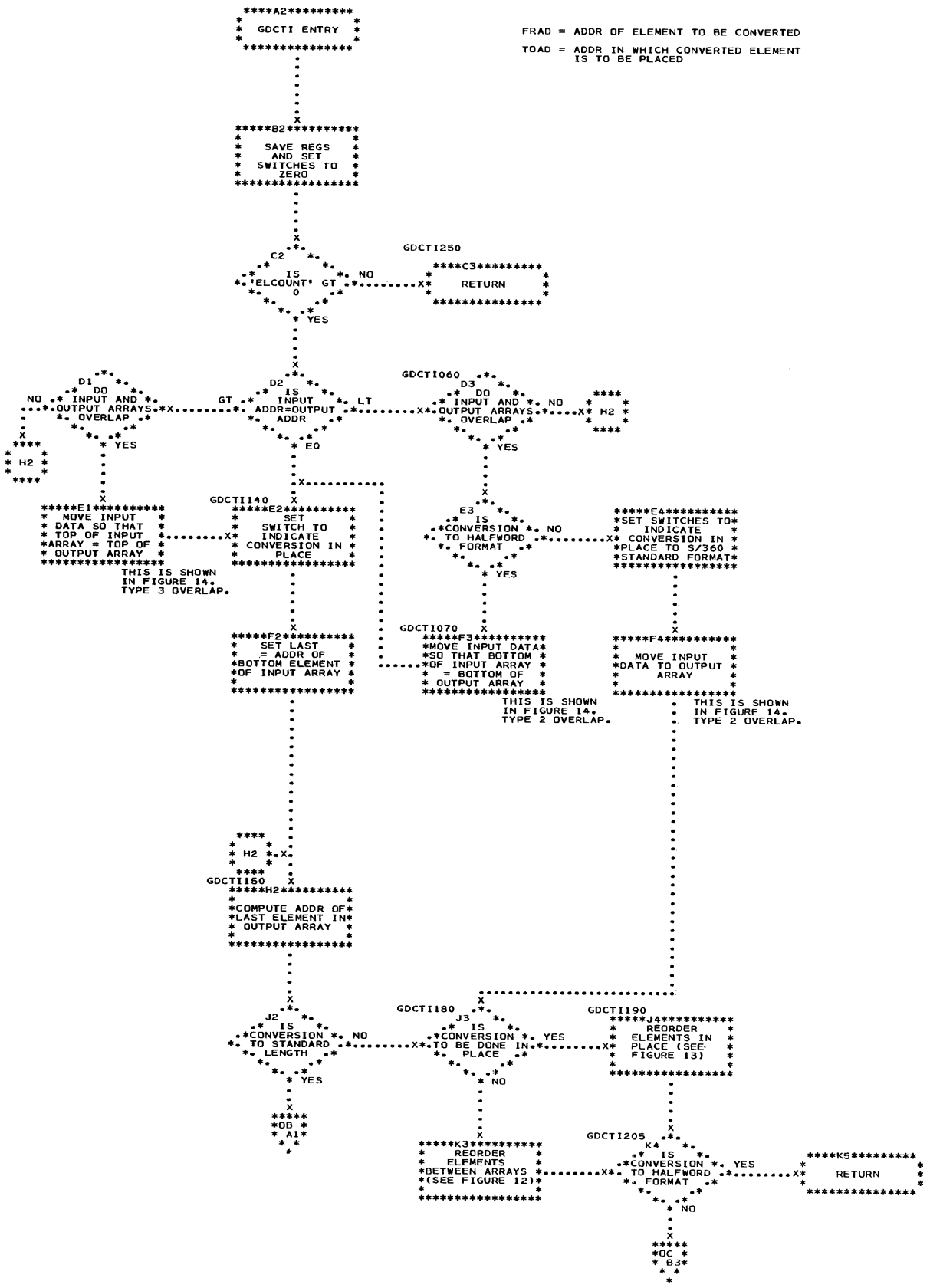


Chart OB. GDCTI Routine (Part 2 of 3)

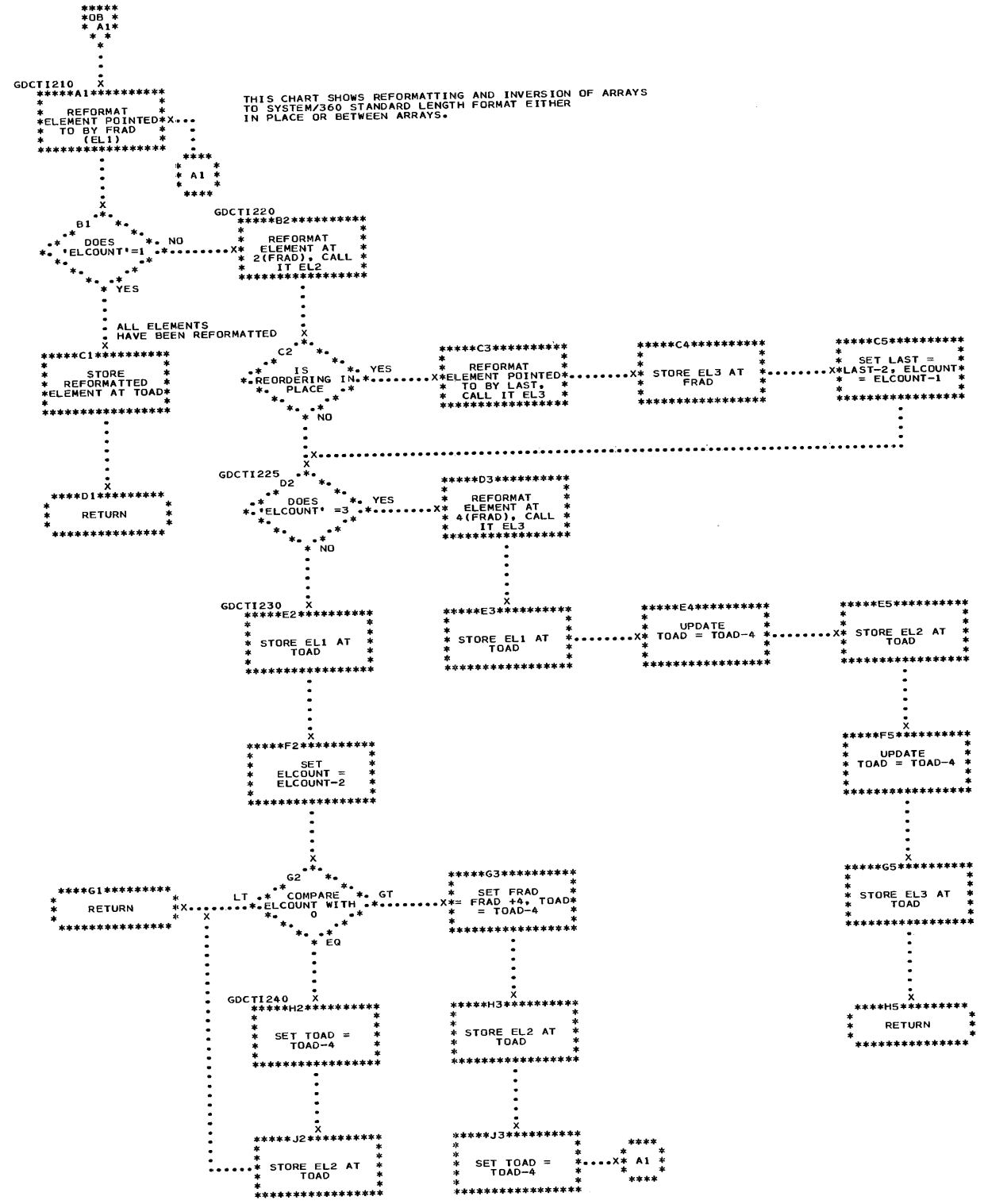


Chart OC. GDCTI Routine (Part 3 of 3)

THIS CHART SHOWS REFORMATTING AND INVERSION OF ARRAYS TO SYSTEM/360 STANDARD LENGTH FORMAT WHEN THE INPUT ARRAY ADDRESS IS LESS THAN THE OUTPUT ARRAY ADDRESS AND THE ARRAYS OVERLAP.

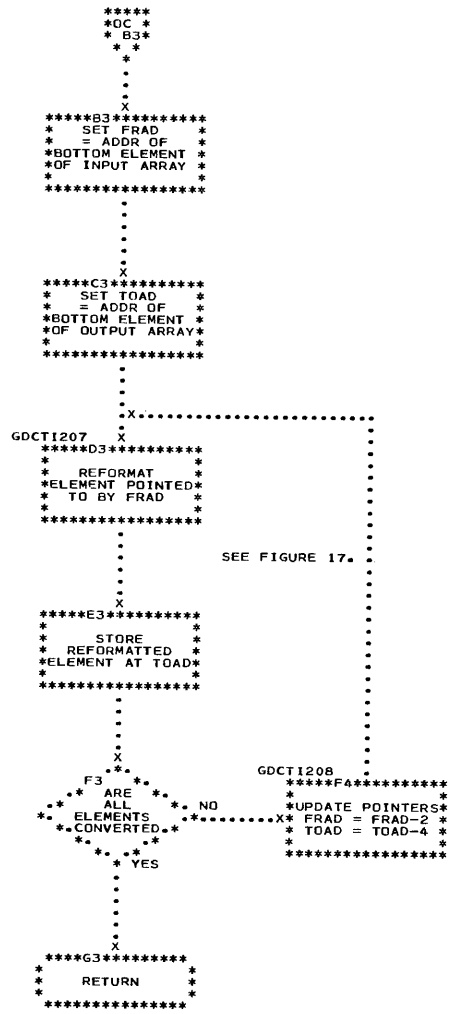


Chart PA. GDCFF and GDCTF Routines (Part 1 of 2)

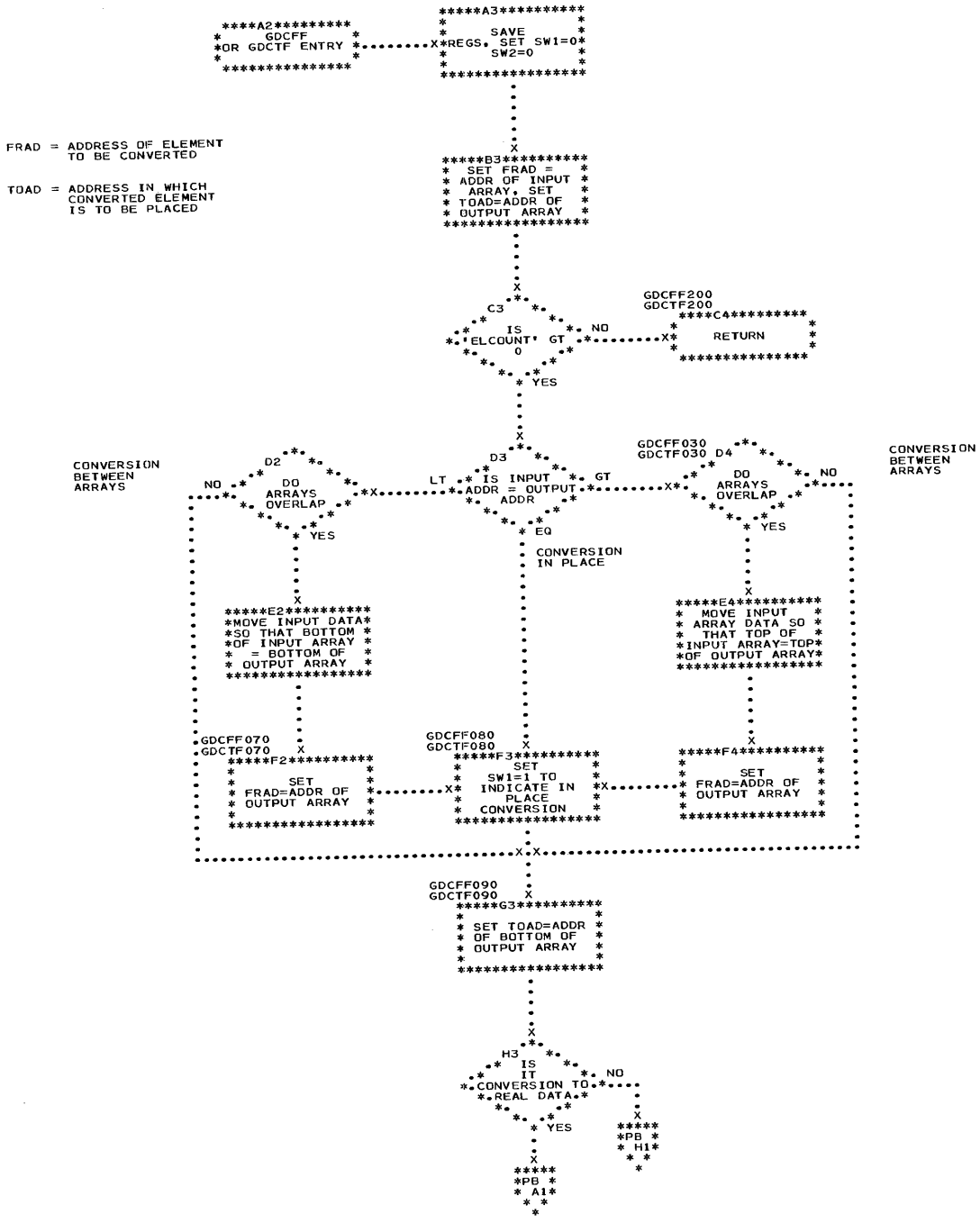


Chart PB. GDCFF and GDCTF Routines (Part 2 of 2)

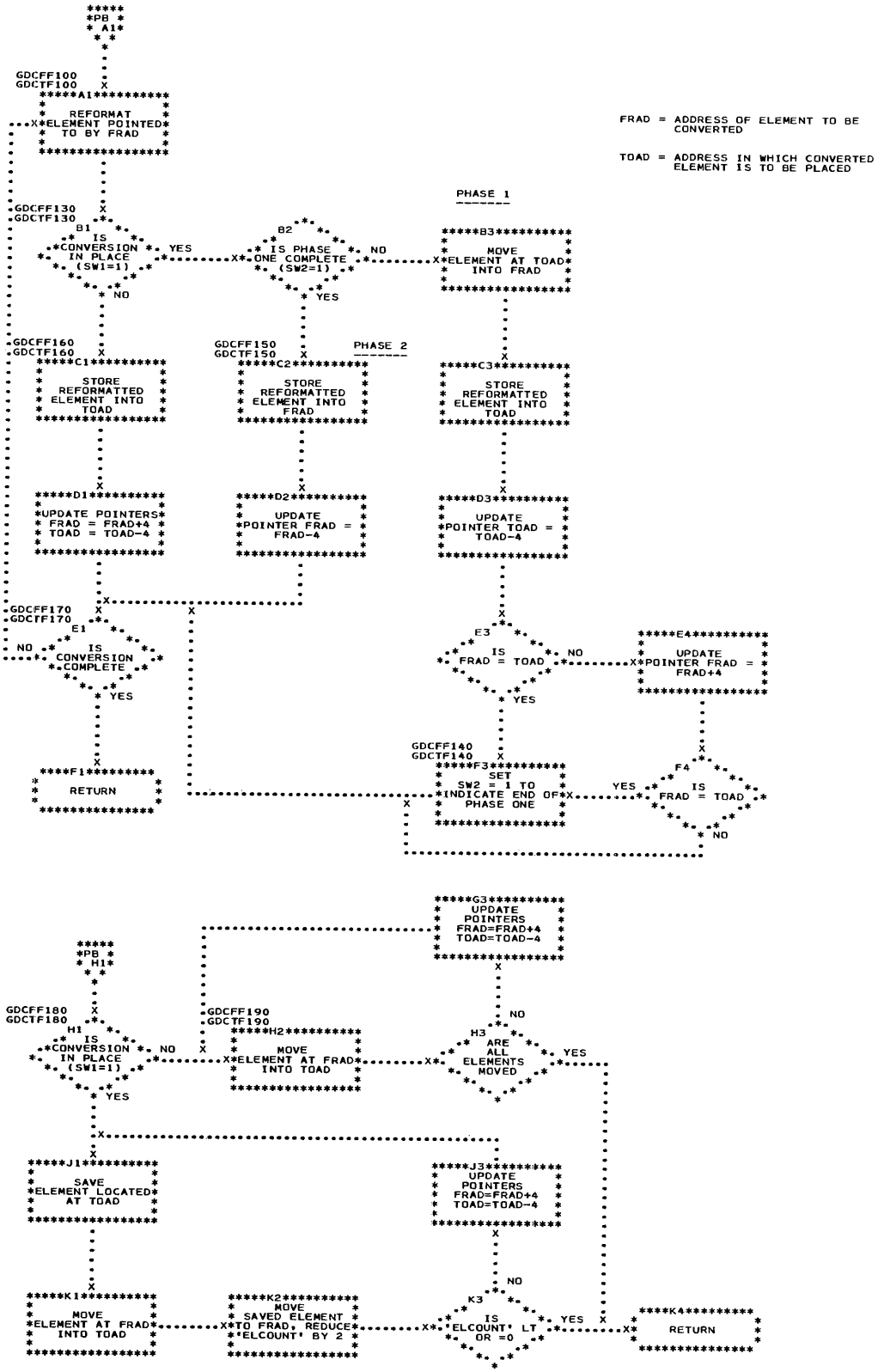


Chart QA. GDCFE Routine

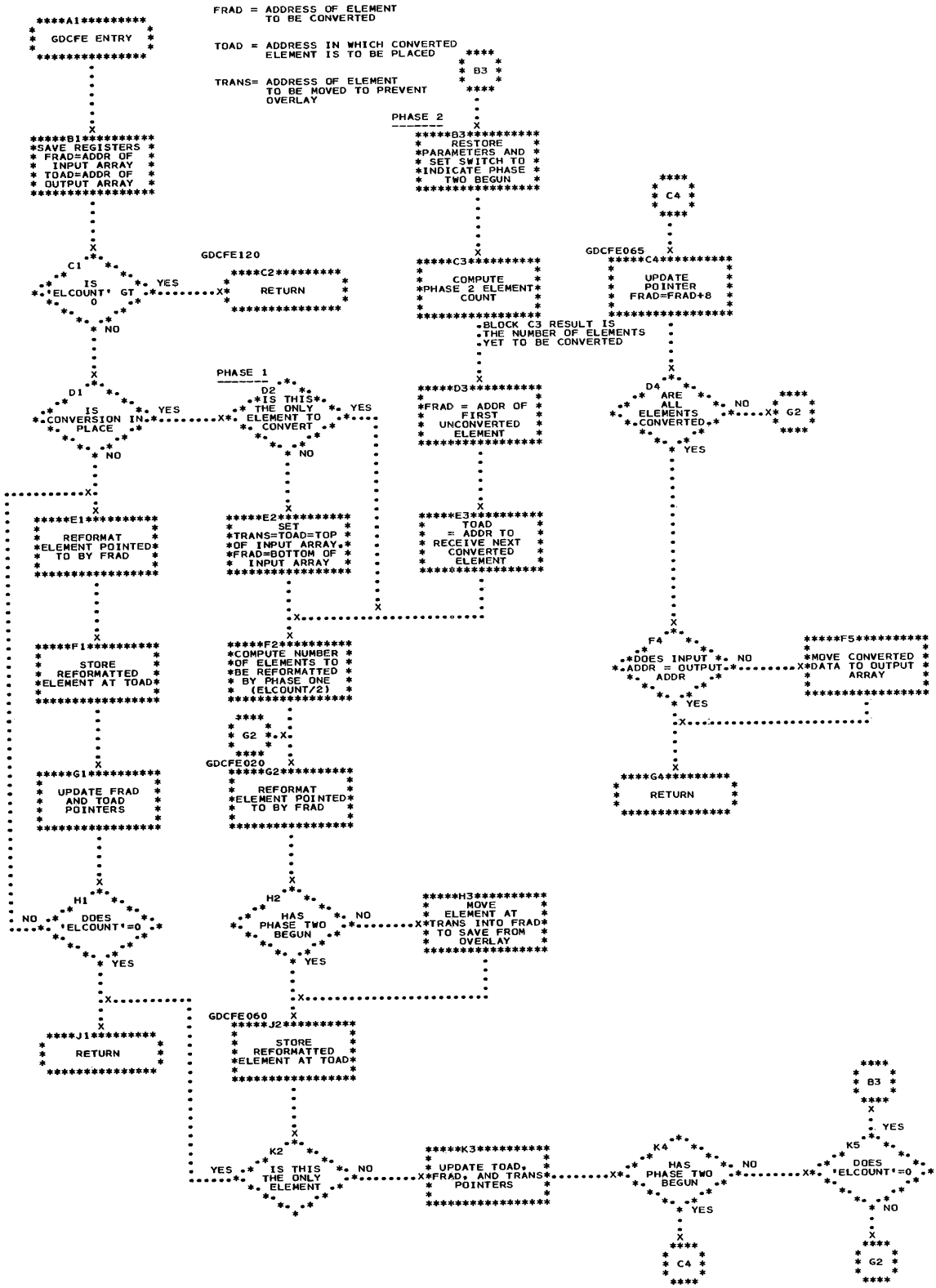
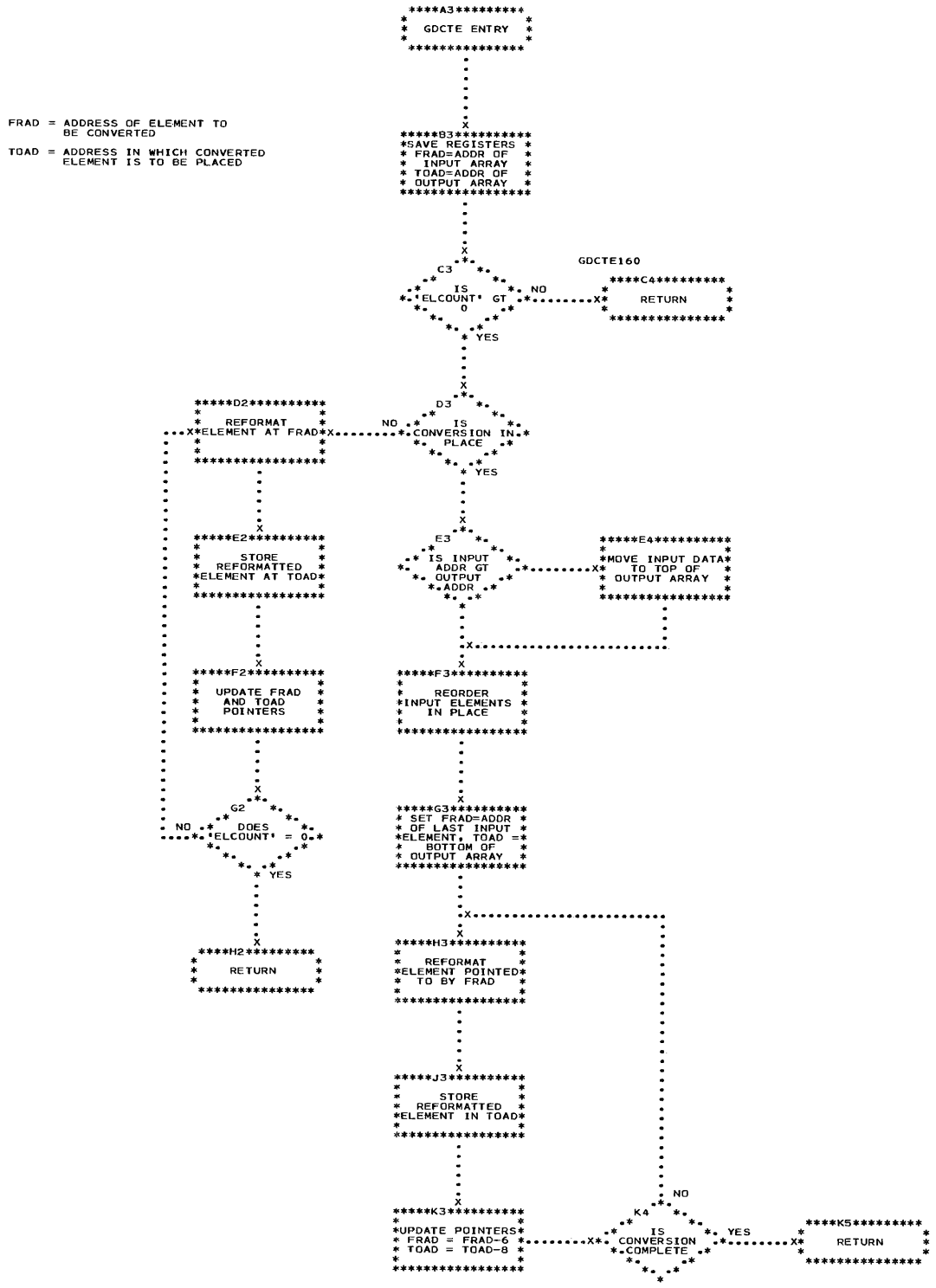


Chart RA. GDCTE Routine



This appendix illustrates and describes the storage areas which provide a means of communication among the PTOP data transmission routines and between the systems. The areas described are:

- 1130 telecommunications control block (GTCOM).
- System/360 telecommunications control block (GTCB).
- System/360 save area used when the asynchronous routine is in control.
- System/360 unit assignment table (module IKDUATBL).

Each description includes information about establishment of the block, general use, and the time at which the block is released.

The format of each block is illustrated, followed by detailed descriptions of the fields within the block. An upright arrow within the field indicates that the contents of the field is an address. Field displacements (in decimal and hexadecimal) are shown at the left side of each illustration. These displacements indicate the location of the named field within the block in terms of the number of words in 1130 or the number of bytes in System/360. The same displacements are shown in parentheses in the field descriptions.

Following the block descriptions in this appendix are tables summarizing the processing of information in the GTCOM and the GTCB (see "Processing Information in Control Blocks").

1130 Telecommunications Control Block (GTCOM)

The GTCOM serves as a communication area for all the 1130 data transmission routines. It resides within the GTP01 routine starting four words beyond the routine entry point.

Pointers, indicators, and flags in the GTCOM are set by all 1130 data transmission routines except GTCLT. Certain fields in the GTCOM are cleared by the GTEND routine.

Figure 21 shows the format of the GTCOM after all fields have been initialized.

Dec.	Hex.	<-------1 Word (16 Bits)------->
+ 0	0	BITS
+ 1	1	BITS2
+ 2	2	LSTRD
+ 3	3	LSTWT
+ 4	4	RDCNT
+ 5	5	WTCNT
+ 6	6	RDPTR ↑
+ 7	7	WTPTR ↑
+ 8 +15	8 F	SAVST (8 Words)
+16	10	RDBUF ↑
+17	11	WTBUF ↑
+18	12	ASYNC ↑
+19	13	PSWRD ↑
+20	14	RDSEQ
+21	15	WTSEQ
+22	16	DSREF
+23	17	ASPRM
+24	18	ASDAT
+25	19	NSI ↑
+26 +27	1A 1B	AQS (2 Words)
+28	1C	XR1S
+29	1D	XR2S
+30	1E	STS
+31	1F	OSCNT
+32 +39	20 27	SYSBF (8 Words)
+40	28	Reserved
+41	29	Reserved
+42	2A	Reserved

Figure 21. GTCOM Format

Table 1. Contents of BITS Field

Bit	Name	Meaning of Contents When Bit is Set to One (On)
0	RTBSY	An input/output request (other than a system read request) is active or pending.
1	OSRED	System/360 user ready-to-read message has been received.
2	ASPND	A request for the 1130 user's asynchronous routine is pending.
3	OKNIT	The 1130 PTOP transmission routines are initialized.
4-5	ERRST	Indicates the error status of the active message, as follows: 0 No errors 1 Unrecoverable error(s) 2 Incorrect length 3 Timeout
6	ASACT	System/360 asynchronous routine is active.
7	FNCTN	Indicates the currently active function, as follows: 0 Transmit Initial, or Receive Initial 1 Transmit End, or Receive Continue/Repeat
8-10	MSGTP	Identifies the message currently being transmitted, as follows: 0 Initialization 1 Terminate communication link 2 Asynchronous routine request 3 Asynchronous routine ended 4 Data message 5 Ready-to-read message 6 User read 7 System read
11	BIT11	Mainline program transmission status should be saved by GTP01 before processing the next operation.
12	BIT12	If one, the 1130 asynchronous routine is active. If zero, the 1130 asynchronous routine is inactive.
13-15		Reserved.

BITS (+0, /00)
contains the flags and switches used by the 1130 data transmitted routines. Table 1 gives the name and the contents of each bit in the BITS field.

BITS2 (+1, /01)
contains bits used to queue PTOP requests. The bits are listed in Table 2 in the order of priority for honoring the requests.

LSTRD (+2, /02)
contains integer codes that indicate (1) the type of data and (2) the status of the last read request specified by the user. The meanings of the codes are defined in Table 3.

LSTWT (+3, /03)
contains integer codes that indicate (1) the data type and (2) the status of the last write request specified by

Table 2. Contents of BITS2 Field

Bit	Meaning of Contents When Bit is On
0-8	Reserved.
9	Initialization message is pending.
10	Terminate communication link message is pending.
11	Asynchronous routine request message is pending.
12	Asynchronous routine ended message is pending.
13	Data message is pending.
14	Ready-to-read message is pending.
15	User read request is pending.

the user. The meanings of the codes are defined in Table 3.

RDCNT (+4, /04)
contains the number of 1130 words to be transmitted (word count) from the last 1130 ready-to-read message.

WTCNT (+5, /05)
contains the word count from the last 1130 data message.

RDPTR (+6, /06)
contains the address of the first element (placed in the highest storage location) in the user's read array.

WTPTR (+7, /07)
contains the address of the first element (placed in the highest storage location) in the user's write array.

SAVST (+8, /08)
is an 8-word save area into which the transmission status of the mainline program is placed before execution of the asynchronous routine. The value of OSCNT is placed in the first word of this save area if OSRED is set to one when the 1130 asynchronous routine is requested. Otherwise, the first word contains zero. The three low-order bits (13, 14, and 15) in BITS2 are saved in the second word, and the six words from locations GTCOM+2 through GTCOM+7 are placed in the remainder of this save area.

RDBUF (+16, /10)
contains the address of the user-specified read buffer.

WTBUF (+17, /11)
contains the address of the user-specified write buffer.

ASYNC (+18, /12)
contains the address of the user's asynchronous routine.

PSWRD (+19, /13)
contains the address of the password.

RDSEQ (+20, /14)
contains the sequence number of the next message to be read.

WTSEQ (+21, /15)
contains the sequence number of the next message to be written.

DSREF (+22, /16)
contains the data set reference number passed by the System/360.

ASPRM (+23, /17)
contains the one-word integer data passed from the System/360 for the 1130 user's asynchronous routine.

ASDAT (+24, /18)
contains the one-word integer data passed by the 1130 user's program for the System/360 asynchronous routine.

NSI (+25, /19)
is a save area used to store the address of the next sequential

Table 3. Contents of LSTRD and LSTWT Fields

Bits	Meaning
0-7	A code that defines the type of data specified by the last transmission request, as follows: 1 One-word integer 2 Standard-precision integer (2 words, only first word used) 3 Extended-precision integer (3 words, only first word used) 4 Standard-precision real (2 words) 5 Extended-precision real (3 words) Note: Alphameric data can be placed in any of the FORTRAN data types; it is treated as data of that format.
8	For LSTWT, if on, this bit indicates that incorrect data length was specified. For LSTRD, this bit should never be on.
9-15	A code that indicates the status of the last operation, as follows: 1 Operation successfully completed 2 Operation not started 3 Operation started, but not complete 4 Both systems are in ready-to-read status 5 System/360 has called GTEND 6 Transmission line error 7 Incorrect length

instruction in the program that called the data transmission routine when the 1130 asynchronous routine is invoked.

AQS (+26, /1A)
is a two-word save area used to store the contents of the accumulator and the accumulator extension when an 1130 data transmission routine invokes the 1130 asynchronous routine.

XR1S (+28, /1C)
is a save area used to store the contents of Index Register 1 when an 1130 data transmission routine invokes the 1130 asynchronous routine.

XR2S (+29, /1D)
is a save area used to store the contents of Index Register 2 when an 1130 data transmission routine invokes the 1130 asynchronous routine.

STS (+30, /1E)
is a save area used to store the status of the carry and overflow indicators when an 1130 data transmission routine invokes the 1130 asynchronous routine.

OSCNT (+31, /1F)
contains the data count obtained from the last System/360 ready-to-read message.

SYSBF (+32, /20)
is an 8-word field which serves as the system message buffer.

System/360 Telecommunications Control Block (GTCB)

The GTCB furnishes the means of communication and control among the System/360 data transmission routines. It is created by the IKDGTNIT routine each time the routine is invoked with a different "datasetref" argument. The address of the GTCB is placed in the unit assignment table at a location correlated with the data set reference number specified in the call that invoked the IKDGTNIT routine. As a cross reference, the address of the unit assignment table and the data set reference number are then placed in the GTCB.

The GTCB is used by all the System/360 data transmission routines, but not by the data conversion routines. Once established, it remains in storage until released by a System/360 call to GTEND in the user's mainline program which specifies the associated data set reference number. This call invokes the IKDGTEND routine, which frees the GTCB.

Figure 22 shows the format of the GTCB after the fields have been initialized.

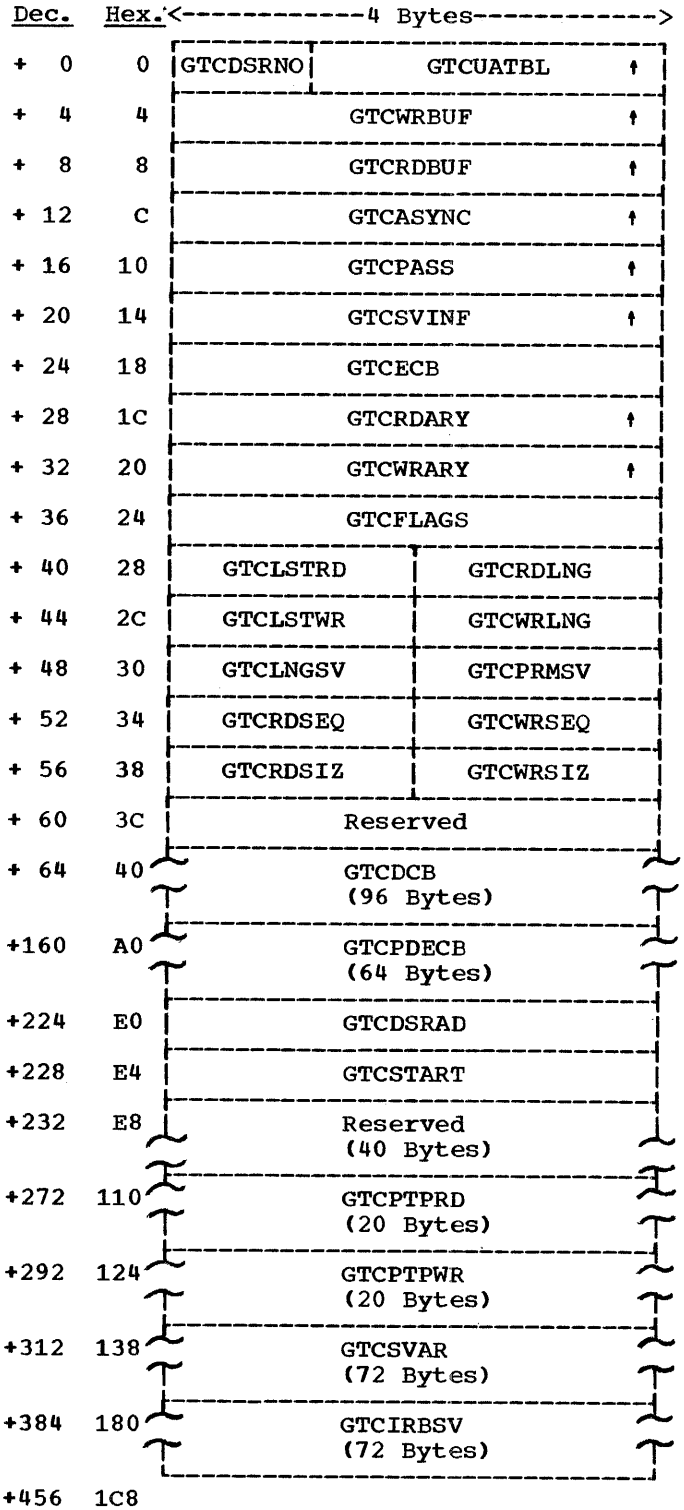


Figure 22. GTCB Format

GTCDSRNO (+0, X'00')
is a one-byte field that contains the data set reference number associated with this GTCB. Its value ranges from 1 through 99.

GTCUATBL (+1, X'01')
is a three-byte field that contains the address of the unit assignment table.

GTCWRBUF (+4, X'04')
is the address of the user's write buffer.

GTCRDBUF (+8, X'08')
is the address of the user's read buffer.

GTCASYNC (+12, X'0C')
is the address of the user-specified asynchronous routine.

GTCPASS (+16, X'10')
is the address of the password.

GTCSVINF (+20, X'14')
is the address of the special save area used for saving the transmission status of the mainline program during asynchronous routine processing.

GTCECB (+24, X'18')
is an event control block for internal use by the PTOP program.

GTCRDARY (+28, X'1C')
is the address of the array to which the user-specified data is to be moved after it is read into the read buffer.

GTCWRARY (+32, X'20')
is the address of the array from which the user-specified data is to be written.

GTCFLAGS (+36, X'24')
contains flags that are used by the System/360 data transmission routines. These one-bit flags are listed and described in Table 4.

GTCLSTRD (+40, X'28')
is the status of the last read operation requested by the user. Its value ranges from 1 through 8, with the following meanings:

1 -- Operation successfully completed.
2 -- Operation not started.
3 -- Operation started but not completed.
4 -- Both systems are in ready-to-read status.
5 -- The 1130 program has called GTEND.
6 -- Operation unsuccessful due to transmission line error.
7 -- Operation completed with incorrect length indicated.
8 -- Operation unsuccessful due to condition that requires operator intervention.

If the high order bit in GTCLSTRD is on, the requested operation is still pending.

GTCRDLNG (+42, X'2A')
is the number of bytes of data the user expects to read from the 1130 as specified in the call to GTRED.

GTCLSTWR (+44, X'2C')
is the status of the last write data operation that was requested by the user. Its value ranges from 1 through 8, with the same meanings as defined for GTCLSTRD. If the high order bit in GTCLSTWR is on, the requested operation is still pending.

GTCWRLNG (+46, X'2E')
is the number of bytes of data the user wishes to write to the 1130 as specified in the call to GTWRT.

GTCLNGSV (+48, X'30')
is the data length specified in the ready-to-read message that was last received from the 1130 program.

GTCPRMSV (+50, X'32')
is the parameter that the System/360 user specified when requesting the 1130 asynchronous routine.

GTCRDSEQ (+52, X'34')
contains the sequence number of the next message to be read.

GTCWRSEQ (+54, X'36')
contains the sequence number of the next message to be transmitted to the 1130 program.

GTCRDSIZ (+56, X'38')
contains the value specified by the user for the "readbufsize" argument in his call to the GTNIT subroutine.

GTCWRSIZ (+58, X'3A')
contains the value specified by the user for the "writebufsize" argument in his call to the GTNIT subroutine.

GTCDCB (+64, X'40')
is the BTAM data control block for the communication line group established for the 1130 and System/360 line connection.

Table 4. Contents of GTCFLAGS Field (Part 1 of 2)

Byte	Bit	Value	Name	Meaning
0 FLGS1	0	X'80'	OKNIT	System/360 is initialized successfully.
	1	X'40'	RTRD	1130 user is in ready-to-read status.
	2	X'20'	ASYNC	System/360 user asynchronous routine is scheduled.
	3	X'10'	ASYNIT	System/360 user asynchronous routine is requested.
	4	X'08'	A1130	System/360 user requested the 1130 asynchronous routine.
	5	X'04'	CANCEL	System/360 user called GTWRT to terminate the 1130 program.
	6	X'02'	SVRTRD	1130 mainline ready-to-read request saved.
	7	X'01'	GTEND	1130 user has called GTEND.
1 FLGS2	0	X'80'	USEROP	The current input/output operation is for the System/360 user.
		X'00'	PTOPOP	The current input/output operation is for the PTOP system.
	1	X'40'	PTPRTI	Indicates current operation is Read Initial.
	2	X'20'	PTPRTT	Indicates current operation is Read Continue.
	3	X'10'	PTPTIXR	Indicates current operation is Write Initial Transparent with Reset.
	4	X'08'	PTPWTR	Indicates current operation is Write Reset.
	5	X'04'	RTPAS	Indicates current operation is Read Continue for the password (compares satisfactorily).
	6	X'02'	ERRNIT1	Error in initialization, negative code (passwords do not match).
	7	X'01'	ERRNIT2	Error in initialization, positive code (unable to communicate with 1130).
2 FLGS3	0	X'80'	WTEND	System/360 mainline program called GTEND, and IKDGTEND is in wait state.
	1	X'40'	WTCLT	System/360 program called GTCLT and IKDGTCLT is in wait state.
	2	X'20'	WTASY	System/360 asynchronous routine called GTEND, and IKDGTEND is in wait state.
	3	X'10'	ENDMSG	Write contention found when terminate communication link message is active.
	4	X'08'	ASYEND	Write contention found when asynchronous routine ended message is active.
	5	X'04'	TENTION	Write contention condition is pending.
	6	X'02'	COMMN	1130 to System/360 communication was made.
	7	X'01'	WTRD	IKDGTCLT is waiting for a read operation if bit 7 is set to one. (IKDGTCLT is waiting for a write operation if bit 7 is zero.)

Table 4. Contents of GTCFLAGS Field (Part 2 of 2)

Byte	Bit	Value	Name	Meaning
3 FLGS4	0	X'80'	CANCTD	Write contention found when program termination request message is active.
	1	X'40'	ASYCTD	Write contention found when asynchronous routine request message to the 1130 is active.
	2	X'20'	RTRCTD	Write contention found when System/360 user ready-to-read message is active.
	3	X'10'	DATATD	Write contention found when System/360 user data message is active.
	4	X'08'	CANPND	The program termination request message is pending.
	5	X'04'	ASYPND	The asynchronous routine request message is pending.
	6	X'02'	RTRPND	The ready-to-read message is pending.
	7	X'01'	SHORT	User data to be read is less than four bytes.

GTCPDECB (+160, X'A0')
is the BTAM data event control block that is built to control input/output operations.

Note: Complete descriptions of the format and contents of the BTAM data control block and the BTAM data event control block are contained in the following publications:

IBM System/360 Operating System:

System Control Blocks, Form C28-6628

Basic Telecommunications Access Method, Form C30-2004

Basic Telecommunications Access Method Program Logic Manual, Form Y30-2001

GTCDSRAD (+224, X'E0')
is the data set reference number for the System/360 user asynchronous routine.

GTCSTART (+228,X'E4')
is the parameter (integer value) passed when the 1130 program calls the System/360 asynchronous routine.

GTCPTPRD (+272, X'110')
is the read buffer for receiving PTOP system messages.

GTCPTPWR (+292, X'124')
is the write buffer for transmitting PTOP system messages.

GTCSVAR (+312, X'138')
is a save area established by the data transmission routines in which BTAM will save register contents during the mainline program processing.

GTCIRBSV (+384, X'180')
is a save area established by IKDGTIRB in which BTAM will save register contents during the mainline program processing.

System/360 Save Area

This save area furnishes the means of preserving the transmission status of the mainline program when the System/360 asynchronous routine is in control.

Since the asynchronous routine may use data transmission facilities, it could cause PTOP to overlay certain fields in the GTCB, thus destroying the interrupted transmission status of the mainline program. To avoid this, IKDGTIRB obtains storage for the save area, places its address in the GTCSVINF field of the GTCB, and stores the specified fields in the save area.

IKDGTEND indicates when the save area is no longer needed, and restores the mainline program transmission status. Then the Controller frees the save area.

Figure 23 shows the fields that are placed in the save area from the GTCB, and their displacements from the beginning of the save area.

GTCECB through GTCPRMSV
correspond to the same fields in the GTCB illustrated in Figure 22. Contents of these fields remain unchanged when they are moved to the save area.

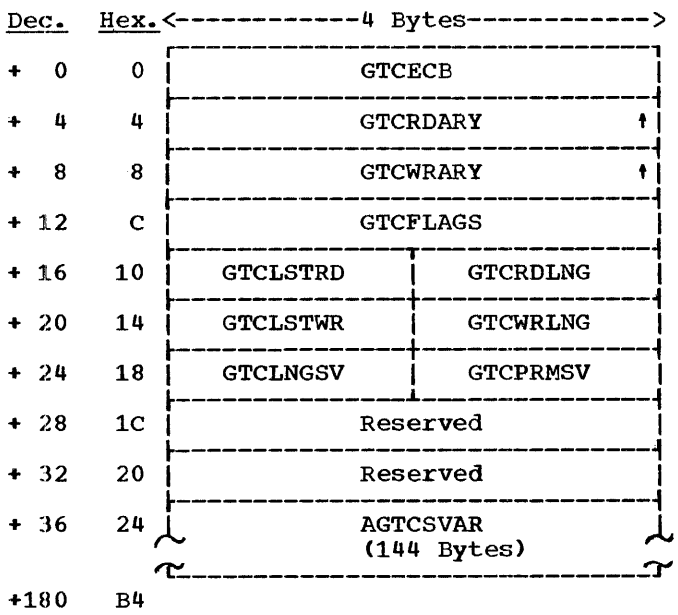


Figure 23. Save Area Format

AGTCSVAR
is a double register save area. The first 72 bytes are established by all the data transmission routines except IKDGTIRB for BATM to save register contents during System/360 asynchronous routine processing. The next 72 bytes are established by IKDGTIRB for BATM to save register contents during System/360 asynchronous routine processing.

Unit Assignment Table (Module Name IKDUATBL)

The unit assignment table is a permanent module named IKDUATBL that resides in the System/360 link library (SYS1.LINKLIB). It is loaded by the Controller upon the first program initialization procedure, is used by some of the data transmission routines during processing of the user's program, is cleared to zeroes by the IKDGTEND routine, and is removed from main storage by the Controller.

In its final state, the unit assignment table contains the addresses of all GTCBs created during initialization processing, the entry points of the System/360 data transmission modules that are loaded by the IKDGTNIT routine, a count of the GTCBs, and a save area. The data transmission routines use the GTCB address fields to determine whether or not a valid GTCB exists for the data set reference number passed by the user.

Initially, the unit assignment table contains all zeroes. The GTCB address fields and the entry point fields are initialized during processing performed by IKDGTNIT.

Figure 24 shows the format of the unit assignment table after all fields have been initialized.

GTCB Addresses (+0, X'00')
are the addresses of the starting locations of the GTCBs that have been created by calls to GTNIT in the System/360 program. Each address is four bytes long. The value of the data set reference number is used as an index factor to locate the appropriate GTCB address.

IKDRDWRT EP Address (+396, X'18C')
is the entry point address of the Read and Write Data routine. This address is filled in when IKDRDWRT is loaded by the IKDGTNIT routine. It is used by the Controller to establish linkage when the user's program calls GTRED.

IKDRDWRT EP Address (+400, X'190')
is the entry point address of the Read and Write Data routine. This address is filled in when IKDRDWRT is loaded by the IKDGTNIT routine. It is used by the Controller to establish linkage when the user's program calls GTWRT.

IKDGTCLT EP Address (+404, X'194')
is the entry point address of the Control Test routine. This address is filled in when IKDGTCLT is loaded by the IKDGTNIT routine.

IKDGTIRB EP Address (+408, X'198')
is the entry point address of the Interface Resolution routine. This

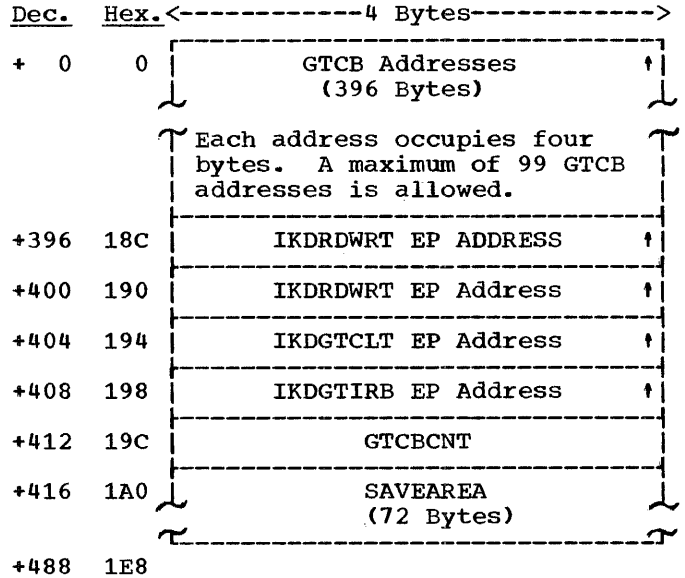


Figure 24. Unit Assignment Table Format

address is filled in when IKDGTIRB is loaded by the IKDGTNIT routine.

GTCBCNT (+412, X'19C')
is a count of the number of GTCBs that have been established. This count is zero initially, and is incremented by one each time a GTCB is created. It is used to determine if the data transmission routines are to be loaded (zero value) or if they are already in main storage (nonzero value).

SAVEAREA (+416, X'1A0')
is a register save area which is used by the Controller whenever the user's mainline program calls one of the data transmission routines. The Controller establishes this save area for the data transmission routine that is invoked to save register contents.

Processing Information in Control Blocks

The processing of information in the PTOP control blocks (GTCOM and GTCB) is summarized in the two tables that follow. Each table lists the individual fields as they appear within the block. Field displacements in decimal and hexadecimal are shown in parentheses. All PTOP routines are listed, and the actions that may be taken in the fields during processing by the routines are indicated in the appropriate columns.

Table 5 summarizes the processing of information in the GTCOM by the 1130 PTOP routines. Table 6 summarizes the processing of information in the GTCB by the System/360 PTOP routines.

Actions are represented by key letters as follows:

- I -- initializes the field with a certain value or address for the first time.
- R -- refers to the field for information (such as flags or parameters) or for use (such as system buffers or save areas).
- M -- modifies the field by replacing its original value to bring the information up to date.
- C -- clears the field to zero because it is no longer needed or does not apply.

Table 5. Processing Information in the GTCOM

Field Name	Module GTNIT	Module GTRED	Module GTWRT	Module GTCLT	Module GTEND	Module GTP01
BITS (+0, /00)						
RTBSY	I,M	M	M		R,C	R,M,C
OSRED			R		C	R,M
ASPND		R,C	R,C	R,C	R,C	R,M
OKNIT	R	R	R		C	R,M
ERRST	R,C				C	R,M
ASACT			I,M		C	R,M
FNCTN					C	R,M
MSGTP		R			C	R,M
BIT11						I-GTP02 C
BIT12						I-GTP02 C
BITS2 (+1, /01)						
9	I	C	C	C	C	R,M
10		C	C	C	C,I	R,M
11		C	I,M,C	C	C	R,M
12		I,M	I,M	I,M	C	R,M
13		C	I,M,C	C	C	R,M
14		I,M,C	C	C	C	R,M
15		I,R,M,C	C	C	C	R,M
LSTRD (+2, /02)						
Data type		I,M			C	R,M
Status		I,M		R	M	R,M
LSTWT (+3, /03)						
Data type			I,M		C	R,M
Data length			I,C		C	R,M
Status			I,M	R	M	R,M
RDCNT (+4, /04)		I,M				R
WTCNT (+5, /05)			I,M			R,M
RDPTR (+6, /06)		I,M				R

Key:
I -- initializes field.
R -- refers to field.
M -- modifies field.
C -- clears field.

(continued)

Table 5. Processing Information in the GTCOM (continued)

Field Name	Module GTNIT	Module GTRED	Module GTWRT	Module GTCLT	Module GTEND	Module GTP01
WTPTR (+7, /07)			I,M			R
SAVST (+8, /08)	I					R,M
RDBUF (+16, /10)	I					R
WTBUF (+17, /11)	I					R
ASYNC (+18, /12)	I					R
PSWRD (+19, /13)	I,R,M					R
RDSEQ (+20, /14)						R,M
WTSEQ (+21, /15)						R,M
DSREF (+22, /16)	C					I,R
ASPRM (+23, /17)						I,R,M
ASDAT (+24, /18)	I		I,M			R,M
NSI (+25, /19)		I,R	I,R	I,R	I,R	
AQS (+26, /1A)		I,R	I,R	I,R	I,R	
XR1S (+28, /1C)		I,R	I,R	I,R	I,R	
XR2S (+29, /1D)		I,R	I,R	I,R	I,R	
STS (+30, /1E)		I,R	I,R	I,R	I,R	
OSCNT (+31, /1F)			R			I,M
SYSBF (+32, /20)	I,R					R,M

Key:
I -- initializes field.
R -- refers to field.
M -- modifies field.
C -- clears field.

Table 6. Processing Information in the GTCB

Field Name	Module IKDGTCLR	Module IKDGTNIT	Module IKDRDWRT	Module IKDGTCLT	Module IKDGTEND	Module IKDGTIRB
GTCDSRNO (+0, X'00')	R	I			C	
GTCUATBL (+1, X'01')	R	I			C	
GTCWRBUF (+4, X'04')		I,M	R		C	R
GTCRDBUF (+8, X'08')		I,M	R		C	R
GTCASYNC (+12, X'0C')		I,M			C	R
GTCPASS (+16, X'10')		I,M			C	R
GTCSVINF (+20, X'14')	R,C				C	I,R,M
GTCECB (+24, X'18')		R		R	R,C	I,M
GTCRDARY (+28, X'1C')			I,R,M		C	R
GTCWRARY (+32, X'20')			I,R,M		C	R
GTCFLAGS (+36, X'24')						
OKNIT		R	R		C	I
RTRD			R		C	I,R,M
ASYNC	R				C	I,R,M
ASYNIT					C	I,M,C
A1130			I		C	R,M,C
CANCEL			I		C	R,M,C
SVRTRD					C	I,R,C
GTEND			R		C	I,R,C
USEROP			I		C	R,M,C
PTOPQP					C	I,R,M,C
PTPRTI			I		C	I,R,M,C
PTPRTT					C	I,R,M,C
PTPTIXR			I		C	I,R,M,C
PTPWTR					C	I,R,M,C
RTPAS		R			C	I,M,C
ERRNIT1		R			C	I,M,C
ERRNIT2		R			C	I,M,C

Key:
I -- initializes field.
R -- refers to field.
M -- modifies field.
C -- clears field.

(continued)

Table 6. Processing Information in the GTCB (continued)

Field Name	Module IKDGTCLR	Module IKDGTNIT	Module IKDRDWRT	Module IKDGTCLT	Module IKDGTEND	Module IKDGTIRB
WTEND					I,C	R
WTCLT				I,C		R
WRASY					I,C	R
ENDMSG			I		C	I,R,M
ASYEND					I,C	I,R,M
TENTION			I		C	I,R,M
COMMN		R			C	I,M
WTRD				I,C	C	R
CANCTD			I		C	I,R,M
ASYCTD			I		C	I,R,M
RTRCTD			I		C	I,R,M
DATATD			I		C	I,R,M
CANPND			I		C	R,M,C
ASYPND			I		C	R,M,C
RTRPND			I		C	R,M,C
SHORT			I		C	R,C
GTCLSTRD (+40, X'28')			I	R	C	M
GTCRDLNG (+42, X'2A')			I,M		C	R
GTCLSTWR (+44, X'2C')			I	R	C	M
GTCWRLNG (+46, X'2E')			I,M		C	R
GTCLNGSV (+48, X'30')					C	I,R,M
GTCPRMSV (+50, X'32')					C	I,R,M
GTCRDSEQ (+52, X'34')			I,R,M		C	I,R,M
GTCWRSEQ (+54, X'36')			I,R,M		C	I,R,M
GTCRDSIZ (+56, X'38')		I		C		
GTCWRSIZ (+58, X'3A')		I		C		
GTCDCB (+64, X'40')		I,M	R		C	R
GTCDECB (+128, X'80')		I	R,M		C	R,M

Key:
I -- initializes field.
R -- refers to field.
M -- modifies field.
C -- clears field.

(continued)

Table 6. Processing Information in the GTCB (continued)

Field Name	Module IKDGTCLR	Module IKDGTNIT	Module IKDRDWRT	Module IKDGTCLT	Module IKDGTEND	Module IKDGTIRB
GTCDSRAD (+176, X'B0')					C	I,R,M
GTCDSRAD (+176, X'B0')					C	I,R,M
GTCSTART (+180, X'B4')					C	I,R,M
GTCPTPRD (+224, X'E0')					C	I,R,M
GTCPTPWR (+244, X'F4')		I			R,C	I,R,M
GTCSVAR (+264, X'108')	R	I	I		I,C	
GTCIRBSV(+336, X'150')	R			C	I	

Key:
I -- initializes field.
R -- refers to field.
M -- modifies field.
C -- clears field.

The microfiche directory shown in Table 7 is a guide to named areas of code in the program listing, which is contained on microfiche cards. The microfiche cards are arranged in alphameric order by object module name. The object module names are listed here in the order they are described in the preceding text. The other columns provide the module's descriptive name, its flowchart identification, its attributes, and a synopsis of its function.

Table 7. Microfiche Directory

Module/ CSECT Name	Routine Name	Chart ID	Attributes	Functions
1130 Transmission				
GTNIT	Initialization	AA	Type 4, Subtype 8	Establishes communication link between 1130 and System/360 CPUs.
GTRED	Read Data	BA	Type 4, Subtype 8	Establishes and maintains ready-to-read environment for 1130 problem program.
GTWRT	Write Data	CA	Type 4, Subtype 8	Processes write requests for 1130 problem program.
GTCLT	Control Test	DA	Type 4, Subtype 8	Provides status information about the most recent read or write request made by the user.
GTEND	End Communication	EA	Type 4, Subtype 8	Terminates communication link between 1130 and System/360 CPUs.
GTP01	Telecommunications Monitor	FA to FE	Type 4, Subtype 0	Processes during interruptions and monitors completion of input/output requests initiated by GTRED or GTWRT.
System/360 Transmission				
IKDGTCLR	Controller	HA	Resident, reenterable, non-privileged	Provides entry points for System/360 data transmission subroutines and establishes linkage to the requested routine.
IKDGTNIT	Initialization	JA	Transient, reenterable, non-privileged	Establishes communication link between System/360 and 1130 CPUs.
IKDRDWRT	Read and Write Data	KA	Resident, reenterable, non-privileged	Establishes and maintains ready-to-read environment and processes write requests for the System/360 problem program.

(continued)

Table 7. Microfiche Directory (continued)

Module/ CSECT Name	Routine Name	Chart ID	Atrributes	Functions
System/360 Transmission (continued)				
IKDGTCLT	Control Test	None	Resident, reenterable, non-privileged	Provides status information about the most recent read or write request made by the user.
IKDGTEND	End Communication	LA	Transient, reenterable, non-privileged	Terminates communication link between System/360 and 1130 CPUs.
IKDGTIRB	Interface Resolution	MA, MB, MC	Resident, reenterable, privileged	Processes after input/output interruptions occur and monitors the completion of input/output operations.
System/360 Conversion				
IKDGDCFI	Convert from System/360 Integer to 1130 Integer	NA, NB	Reenterable, non-privileged	Converts an array of integers in System/360 standard-length or halfword format, or alphameric data in the System/360 halfword format to 1130 format.
IKDGDCTI	Convert from 1130 Integer to System/360 Standard-Length or Halfword Integer	OA, OB, OC	Reenterable, non-privileged	Converts an array of integers in the 1130 format to an array of integers in the System/360 standard-length or halfword format.
IKDGDCFF	Convert from System/360 Standard-Length Real to 1130 Standard-Precision Real	PA, PB	Reenterable, non-privileged	Converts an array of real numbers in the System/360 standard-length format to an array of real numbers in the 1130 standard-precision format.
IKDGDCTF	Convert from 1130 Standard-Precision Real to System/360 Standard-Length Real	PA, PB	Reenterable, non-privileged	Converts an array of real numbers in the 1130 standard-precision format to an array of real numbers in the System/360 standard-length format.
IKDGDCFE	Convert from System/360 Double-Precision Real to 1130 Extended-Precision Real	QA	Reenterable, non-privileged	Converts an array of real numbers in System/360 double-precision format to an array of real numbers in 1130 extended-precision format.
IKDGDCTE	Convert from 1130 Extended-Precision Real to System/360 Double-Precision Real	RA	Reenterable, non-privileged	Converts an array of real numbers in the 1130 extended-precision format to an array of real numbers in System/360 double-precision format.

Seven messages are constructed by the PTOP data transmission routines. They are:

- Initialization message, which is a system command initialization message. The text portion contains the password.
- Ready-to-read message, which is a system response sense/status message. The text portion contains the expected length of data in the corresponding data message.
- Data message, which is a system response message. The text portion contains the actual data transmitted from one system to the other.
- Asynchronous routine request message, which is a system command sense/status message. The text portion contains a parameter for the asynchronous routine.
- Program termination request message, which is a system command sense/status message.
- Asynchronous routine ended message, which is a system response sense/status message.
- Terminate communication link message, which is a system command sense/status message.

Each message begins with the control-character sequence DLE STX indicating the start of text in transparent-text mode. These control characters are supplied by the data transmission routines whenever a message is prepared and placed in the buffer for transmission.

Header information follows the control-character sequence. This information consists of four 1-byte entries. For a description of the entries, see "Field Descriptions."

The text portion of each message follows the header information. It varies in length with the message being transmitted. The number of bytes included in the text portion for the various messages are as follows:

- Initialization message -- six bytes.
- Data message -- variable number of bytes depending on the length of the user-defined data to be transmitted.
- All other messages -- four bytes.

The text portion is followed by the control-character sequence DLE ETX indicating the end of text in the transparent-text mode. These characters are inserted during transmission by SCAT2 in the 1130 system, and by BTAM in the System/360.

Message Formats in the 1130 and the System/360

The 1130 message appears in the array defined as the input/output area in the calling sequence that is input to SCAT2. The first word of this input/output area contains a count of the total number of words in the message. Subsequent words in the input/output area contain the message.

The System/360 message appears in the buffer specified as the input or output area in the BTAM READ or WRITE macro instruction. After receipt of data is complete, the System/360 checks for any transmission errors.

Many of the message fields contain the same kind of information for every message. The text portion of the message varies with each message. Refer to "Field Descriptions" for detailed descriptions of the various fields in each message.

Following the detailed field descriptions, each message is illustrated showing its format. Field displacements (in decimal) for both systems are shown in each illustration from the beginning of the respective input/output area. The numbers at the top of each illustration represent 1130 words of storage. The numbers at the bottom of each illustration represent System/360 bytes of storage. A description of the contents of the text portion of each message follows the illustration of the message.

Field Descriptions

The message fields described in Table 8 appear in the input/output area of storage in the order of description. The first field (I/O Count) appears only in the 1130 system. All other fields appear in both the 1130 and the System/360.

Table 8. Message Fields

Field Name	Size of Field	Description
I/O Count	2 Bytes	The number of words in either (1) the message to be transmitted from the 1130, including the preceding control-character sequence and the header information, or (2) the message that was received from the System/360, including all control characters and header information.
DLE STX	2 Bytes	The control-character sequence preceding a message. This sequence appears in every message format. Control characters are represented by the hexadecimal values 10 and 02 respectively.
TYPE	1 Byte	A hexadecimal value that identifies the type of message being transmitted, as follows: 80 -- system response message with bits 4 through 7 in the field set to zero to indicate a data message. 88 -- system response sense/status message. 94 -- system command initialization message. 98 -- system command sense/status message.
DATA ID	1 Byte	In System/360, the data set reference number specified by the user. Zero in the 1130 initialization message. For all other messages in the 1130, the DATA ID passed in the last message received from the System/360.
SEQ	1 Byte	The read or write sequence number of this message. Sequence numbers start with zero for both read and write, and are incremented by one for each message that is received or transmitted. When a maximum value of 255 is reached, the sequence number is reset to zero. For the initialization message, this field contains a hexadecimal value of 48.
OFFST	1 Byte	The number of bytes between the start of header information and the start of data in the message.
TEXT	Varies	See individual descriptions of TEXT following each format illustration.
DLE ETX	2 Bytes	The control-character sequence following a message. This sequence appears in every message format that is received. Control characters are represented by the hexadecimal values 10 and 03 respectively. Only the ETX character is transferred to storage in System/360.
RSVD	2 Bytes	One reserved word of storage following the message in the 1130.
Note: The numeric values shown with various field names in the illustration of each message are hexadecimal values.		

Initialization Message

Figure 25 depicts the format of the initialization message. In PTOP:

- The 1130 always transmits this message;
- The System/360 always receives it.

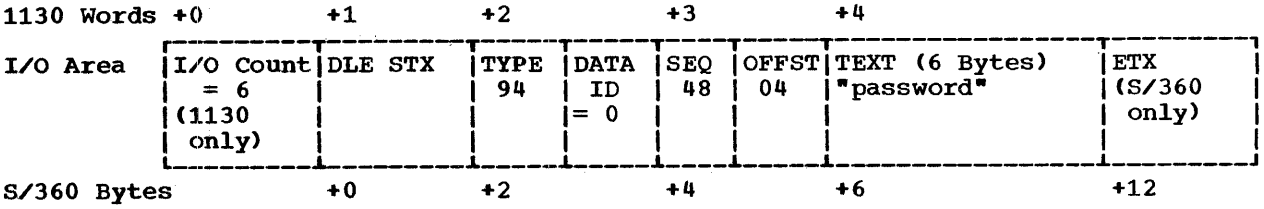
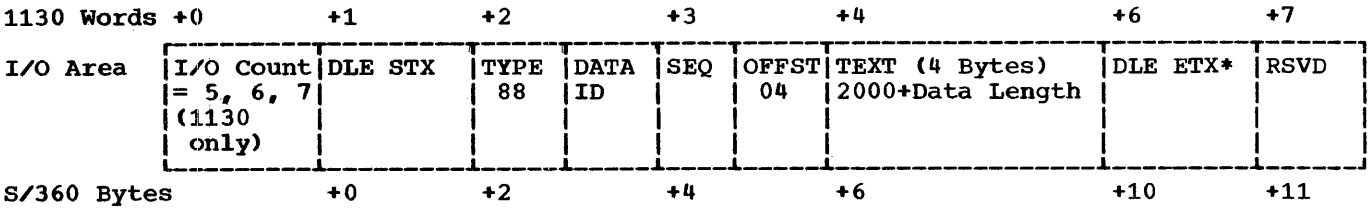


Figure 25. Initialization Message Format

TEXT
contains the password defined in the 1130 user's program.

Ready-To-Read Message

Figure 26 depicts the format of the ready-to-read message. This message is transmitted whenever the user's program calls GTRED.



*ETX appears in the System/360 buffer only when the message is received. DLE ETX appears in the 1130 input/output area only when the message is received.

Figure 26. Ready-To-Read Message Format

TEXT
contains two hexadecimal values, each of which occupies two bytes, as follows:

- 2000, that turns on bit 2 (Read Continue bit) of the field.
- Expected length (in bytes) of the data, excluding control characters and header information, to be received in response to this message.

Note: I/O Count may be 5, 6, or 7 depending on the operation as follows:
Transmit operation -- 5.
Receive operation -- 6 or 7.

Data Message

Figure 27 depicts the format of the data message. This message is transmitted whenever the user's program calls GTWRT and specifies an array of data is to be transmitted.

1130 Words	+0	+1	+2	+3	+4		+(4+n)	+(5+n)
I/O Area	I/O Count = ** (1130 only)	DLE STX	TYPE 80	DATA ID	SEQ	OFFST 04	TEXT Data specified by user (variable number of bytes)	DLE ETX*	RSVD
S/360 Bytes		+0	+2	+4	+6		+(6+x)	+(7+x)

n = number of words of data specified by user.
x = number of bytes of data specified by user.
*ETX appears in the System/360 buffer only when the message is received. DLE ETX appears in the 1130 input/output area only when the message is received.
**I/O Count varies as follows:
For a transmit operation, it equals 3 + n.
For a receive operation, it equals 4 + n or 5 + n.

Figure 27. Data Message Format

TEXT
contains the data from the user's array. The number of words or bytes is determined by the "elcount" argument specified in the call to GTWRT.

Asynchronous Routine Request Message

Figure 28 depicts the format of the asynchronous routine request message. This message is transmitted whenever the user's program calls GTWRT and specifies that an asynchronous routine is to be scheduled in the other system.

1130 Words	+0	+1	+2	+3	+4		+6	+7
I/O Area	I/O Count = 5, 6, 7 (1130 only)	DLE STX	TYPE 98	DATA ID	SEQ	OFFST 04	TEXT (4 Bytes) 0200+Parameter	DLE ETX*	RSVD
S/360 Bytes		+0	+2	+4	+6		+10	+11

*ETX appears in the System/360 buffer only when the message is received. DLE ETX appears in the 1130 input/output area only when the message is received.

Figure 28. Asynchronous Routine Request Message Format

TEXT
contains two hexadecimal values, each of which occupies two bytes, as follows:

- 0200, that turns on bit 6 of the field to indicate the asynchronous routine is to be invoked.
- The parameter (integer value) for the asynchronous routine.

Note: I/O Count may be 5, 6, or 7 depending on the operation as follows:
Transmit operation -- 5.
Receive operation -- 6 or 7.

Program Termination Request Message

Figure 29 depicts the format of the program termination request message. In PTOP, this message is:

- Transmitted by the System/360 whenever the user's program calls GTWRT and specifies that the 1130 program is to be ended.
- Not transmitted by the 1130.

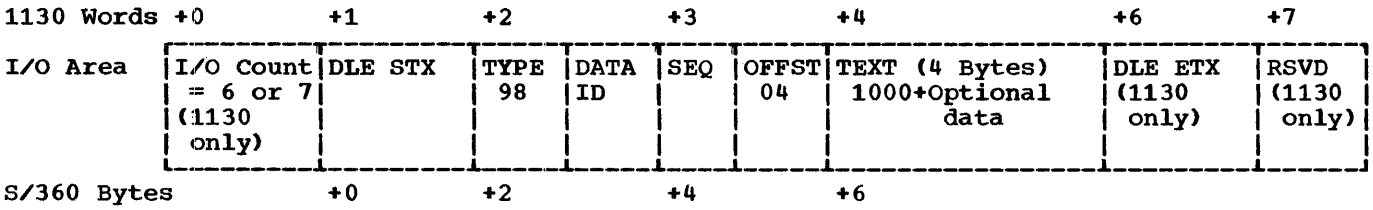


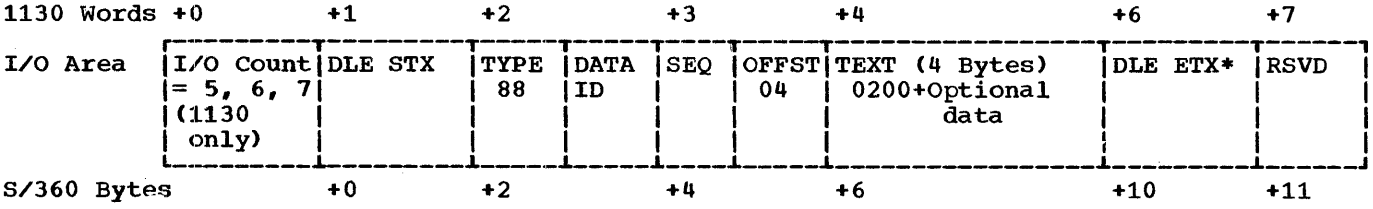
Figure 29. Program Termination Request Message Format

TEXT
contains the hexadecimal value 1000 that turns on bit 3 of the field to indicate the request for abnormal termination of the 1130 program.

Note: I/O Count is 6 or 7 because it is always a receive operation in the 1130.

Asynchronous Routine Ended Message

Figure 30 depicts the format of the asynchronous routine ended message. This message is transmitted whenever the asynchronous routine indicates to the PTOP program that it has completed processing.



*ETX appears in the System/360 buffer only when the message is received. DLE ETX appears in the 1130 input/output area only when the message is received.

Figure 30. Asynchronous Routine Ended Message Format

TEXT
contains the hexadecimal value 0200 that turns on bit 6 of the field to indicate the asynchronous routine is ended.

Note: I/O Count may be 5, 6, or 7 depending on the operation as follows:
Transmit operation -- 5.
Receive operation -- 6 or 7.

Terminate Communication Link Message

Figure 31 depicts the format of the terminate communication link message. This message is transmitted whenever the System/360 user's mainline program calls GTEND, or whenever the 1130 user calls GTEND.

1130 Words	+0	+1	+2	+3	+4	+6	+7
I/O Area	I/O Count = 5, 6, 7 (1130 only)	DLE STX	TYPE 98	DATA ID	SEQ	OFFST 04	TEXT (4 Bytes) 0800+Optional data	DLE ETX*	RSVD
S/360 Bytes		+0	+2	+4	+6	+10	+12

*ETX appears in the System/360 buffer only when the message is received. DLE ETX appears in the 1130 input/output area only when the message is received.

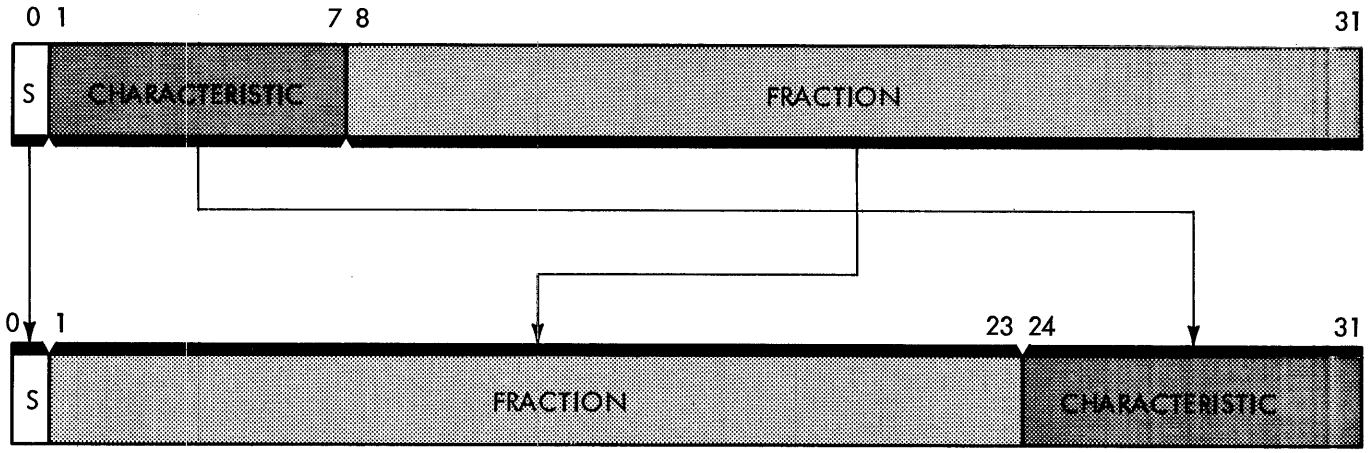
Figure 31. Terminate Communication Link Message Format

TEXT
contains the hexadecimal value 0800 that turns on bit 4 of the field to indicate the closing of a data set.

Note: I/O Count may be 5, 6, or 7 depending on the operation as follows:
Transmit operation -- 5.
Receive operation -- 6 or 7.

This appendix illustrates the manner in which each System/360 data conversion routine converts arithmetic elements (in an array) from the format of one system to the format of the other.

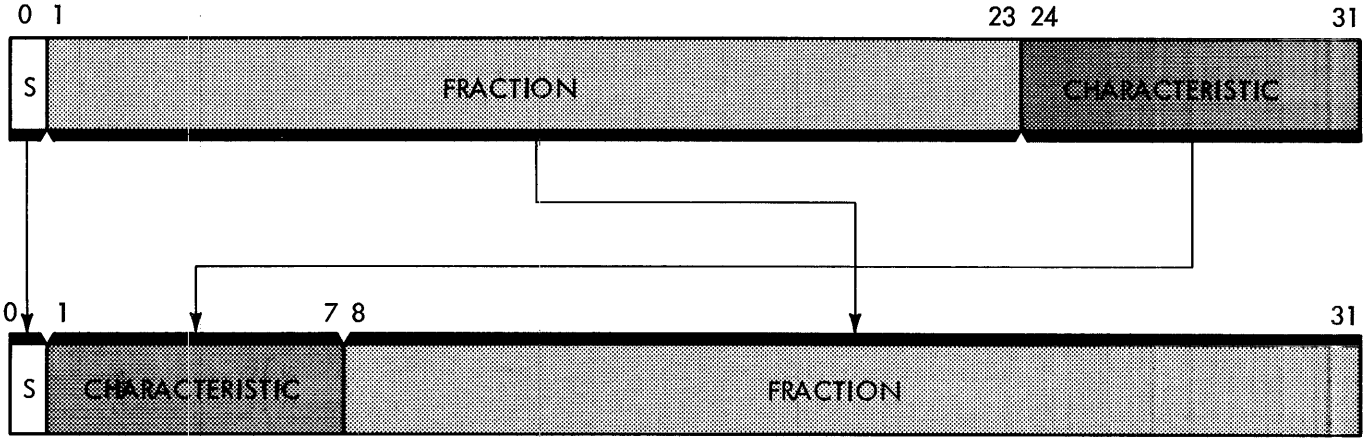
Alphameric conversions are not included in these illustrations since such conversions consist only of inverting the elements in the arrays.



NOTE:

The low-order bit is lost when no normalization is required.

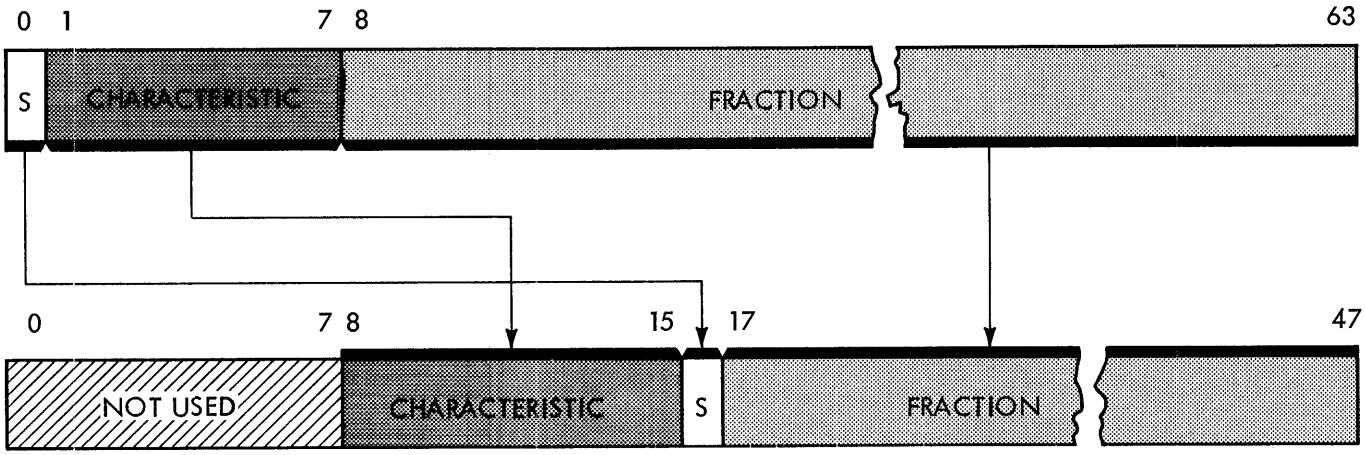
GDCFF--Conversion From System/360 Standard-Length Real to 1130 Standard-Precision Real



NOTE:

The two low-order bits will be lost in the normalization process.

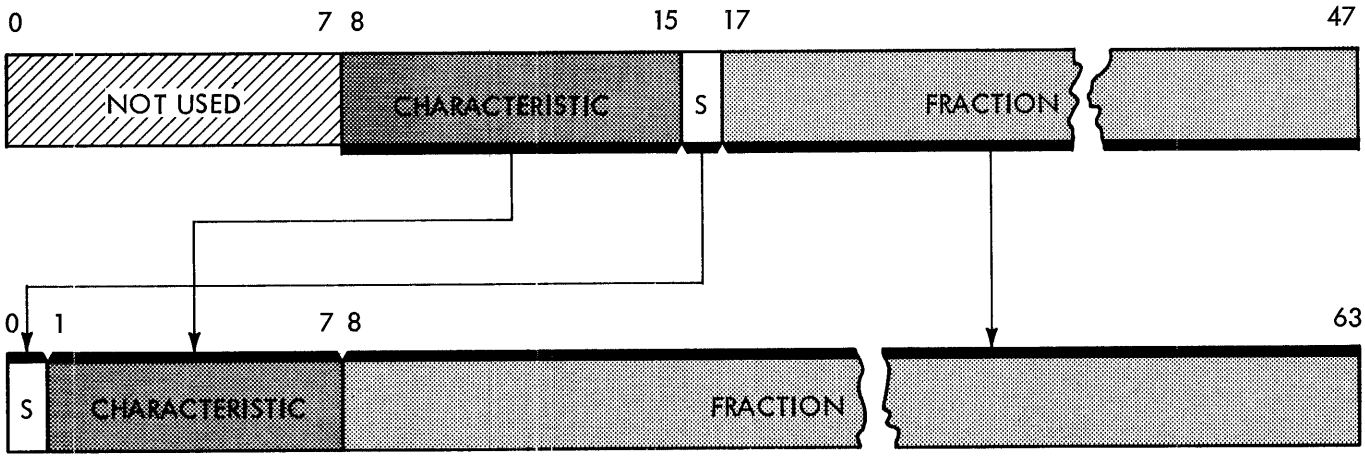
GDCTF--Conversion From 1130 Standard-Precision Real to System/360 Standard-Length Real



NOTE:

25 low-order bits will be lost in the process of contraction and normalization.

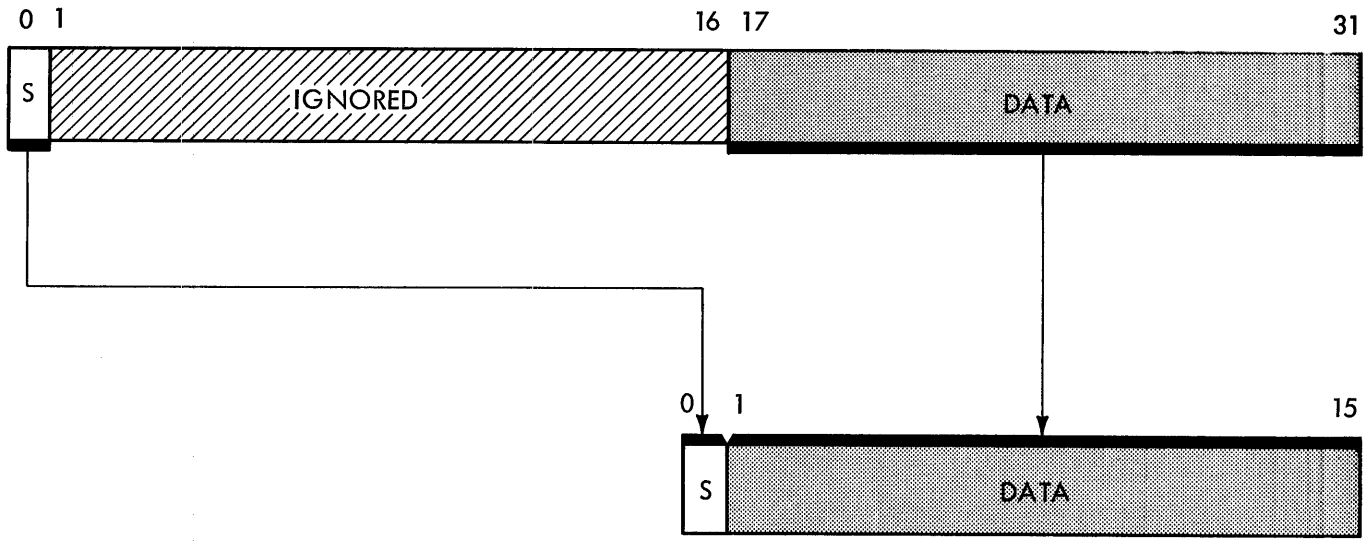
GDCFE--Conversion From System/360 Double-Precision Real to 1130 Extended-Precision Real



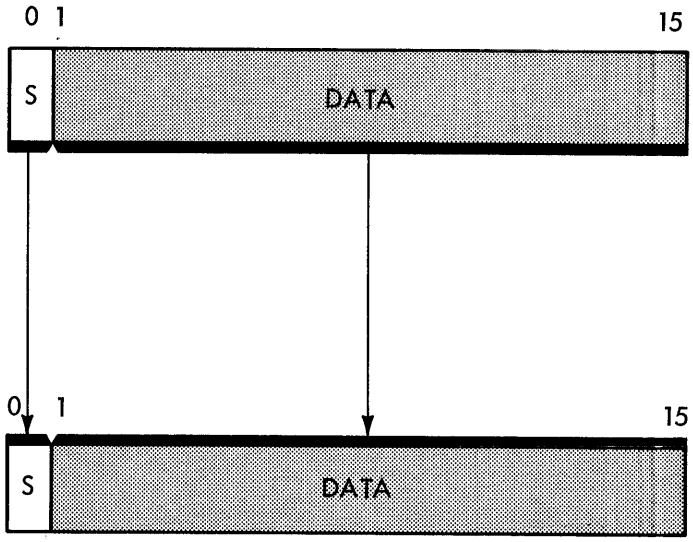
NOTE:

After conversion, the 25 low-order bits of the double-precision data will contain zeros.

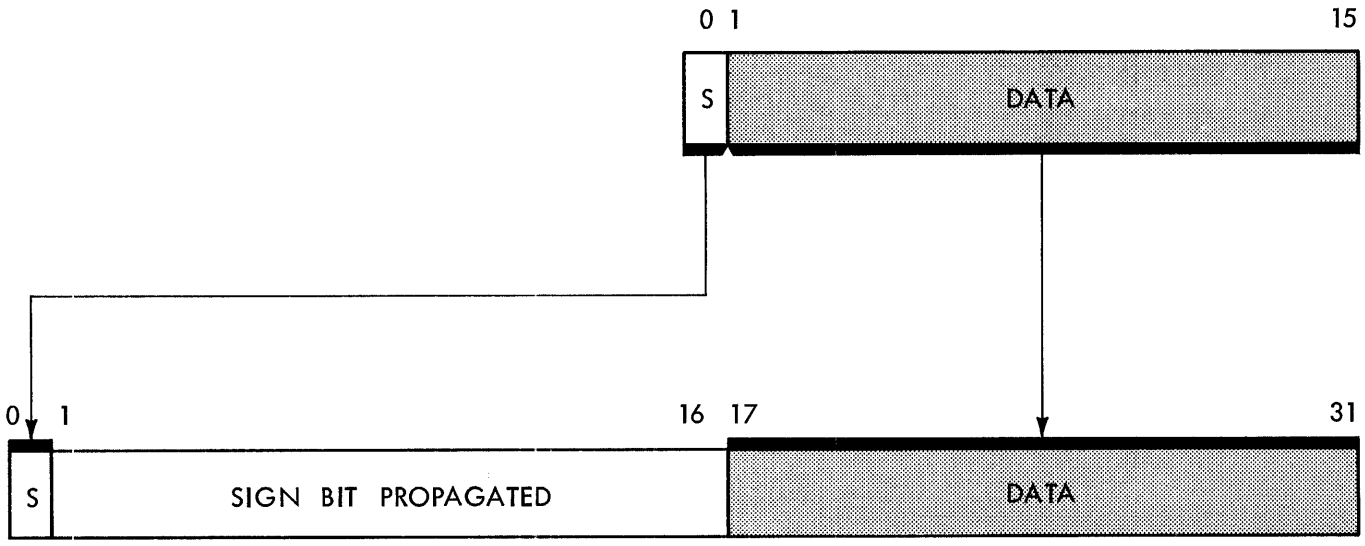
GDCTE--Conversion From 1130 Extended-Precision Real to System/360 Double-Precision Real



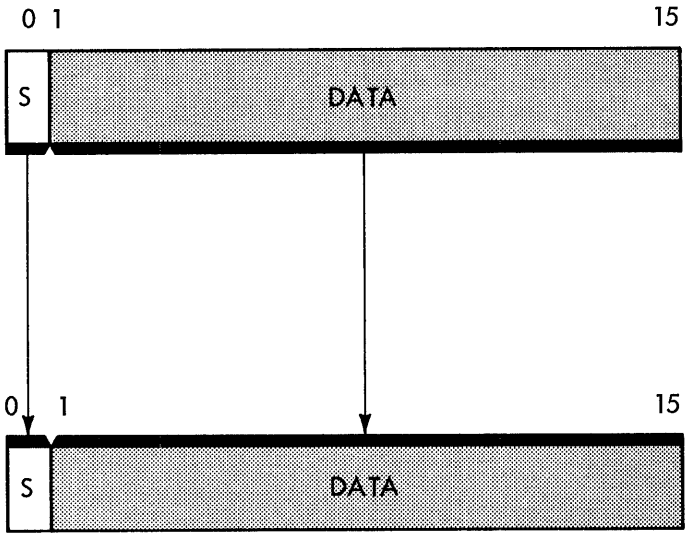
GDCFI--Conversion From System/360 Standard-Length Integer to 1130 Integer



GDCFI--Conversion From System/360 Halfword Integer to 1130 Integer



GDCTI--Conversion From 1130 Integer to System/360 Standard-Length Integer



GDCTI--Conversion From 1130 Integer to System/360 Halfword Integer

This appendix summarizes the contents of registers at various times during processing by the data transmission and data conversion routines. Table 9 lists pertinent register contents upon entry to and throughout processing by each 1130 PTOP routine. Table 10 lists pertinent register contents upon entry to each System/360 PTOP routine.

Table 9. Register Contents for 1130 PTOP Routines

Module Name	Register Contents Upon Entry	Register Contents Throughout Processing
GTCLT	None	Index Register 1 -- Address of parameter list Index Register 2 -- Address of GTCOM
GTEND	None	Index Register 2 -- Address of GTCOM
GTNIT	Index Register 3 -- Address of transfer vector table	Index Register 1 -- Address of parameter list Index Register 2 -- Address of GTCOM
GTP01 Entry Point GTP02	Accumulator -- Zero, or a specific error code Index Register 1 -- Value used to return control to appropriate data transmission routine after 1130 asynchronous routine is invoked (refer to routine description of Telecommunications Monitor routine GTP01)	Index Register 2 -- address used to refer to constants and the GTCOM Index Register 2 -- Address used to refer to constants and the GTCOM
GTRED	None	Index Register 1 -- Address of parameter list Index Register 2 -- Address of GTCOM
GTWRT	None	Index Register 1 -- Address of parameter list Index Register 2 -- Address of GTCOM

Note: GTCOM denotes the 1130 telecommunications control block.

Table 10. Register Contents Upon Entry to System/360 PTOP Routines

Module Name	General Register Contents Upon Entry
IKDGDCFE, IKDGDCFF, IKDGDCFI, IKDGDCTE, IKDGDCTF, IKDGDCTI	1 -- Address of parameter list 13 -- Save area address 14 -- Return address 15 -- Entry point address } System/360 standard conventions
IKDGTCLR	1 -- Address of parameter list 13, 14, 15 -- System/360 standard conventions
IKDGTCLT	4 -- Address of GTCB associated with specified data set reference number 11 -- Address of parameter list 13, 14, 15 -- System/360 standard conventions
IKDGTEND	3 -- Address of unit assignment table (IKDUATBL) 4 -- Address of GTCB associated with specified data set reference number 11 -- Address of parameter list 13, 14, 15 -- System/360 standard conventions
IKDGTIRB	0 -- Address of interruption queue element for this routine 1 -- Address of input/output block associated with input/output operation just completed 13, 14, 15 -- System/360 standard conventions
IKDGTNIT	3 -- Address of unit assignment table (IKDUATBL) 4 -- Address of GTCB associated with specified data set reference number if GTCB already exists; or zero 11 -- Address of parameter list 13, 14, 15 -- System/360 standard conventions
IKDRDWRT	3 -- Address of unit assignment table (IKDUATBL) 4 -- Address of GTCB associated with specified data set reference number 6 -- Code indicating whether user called GTRED (code = 0) or GTWRT (code = 4) 11 -- Address of parameter list 13, 14, 15 -- System/360 standard conventions

Note: GTCB denotes the System/360 telecommunications control block.

This appendix contains additional information that may be useful in diagnosing difficulties with the 1130 or System/360 PTOP programs. Included are: a dummy tracing routine for use with the 1130 PTOP program and pointers to diagnostic aid information that may be useful with the System/360 PTOP program.

1130 Dummy Tracing Routine

Users of the 1130 data transmission routines must ensure that the dummy tracing routine (with entry points IOLOG and CPLOG) is included in the 1130 system. The dummy tracing routine is called by the SCA interrupt service subroutine SCAT2. Entry point IOLOG is entered by a call from SCAT2 during interruption processing. Entry point CPLOG is entered by a call from SCAT2 during call processing.

Upon calling the dummy tracing routine, SCAT2 passes the following information in the accumulator:

- For IOLOG, a value that depends on the operation currently active when the interruption occurs, as follows:

Operation	Value
Write	Character just written, left-justified.
Read	Character just read, right-justified.
Timeout	Hexadecimal 1111.

- For CPLOG, the control parameter passed when SCAT2 was called.

As included in the 1130 system, the dummy tracing routine performs no significant processing; it merely returns control to the calling routine.

The user can incorporate additional coding into this dummy tracing routine to help diagnose problem areas. For example, a tracing routine that stores pertinent information in a user-defined trace table can be incorporated. Standard conventions in regard to saving the contents of the accumulator, index registers, and status indicators should be observed.

Figure 32 depicts the coding for a sample tracing routine that places the information found in the accumulator in a table in storage. Comments included in the figure explain the purpose of the instructions. Instructions enclosed with asterisks are provided in the dummy tracing routine included in the 1130 system. All other instructions represent those added by the installation.

System/360 Diagnostic Information

The contents of general registers upon entry to the System/360 PTOP routines is given in Appendix E. Use of general registers during processing by the PTOP routines is described in the applicable program listings.

In addition to the PTOP control blocks described in Appendix A, the BTAM data control block (DCB) and the BTAM data event control block (DECB) contain information useful as a diagnostic aid. Formats of the DCB and the DECB and their use in BTAM are described in the following publications:

IBM System/360 Operating System:

System Control Blocks, Form C28-6628

Basic Telecommunications Access Method, Form C30-2004

Basic Telecommunications Access Method Program Logic Manual, Form Y30-2001

For additional System/360 diagnostic aids, refer to the publication IBM System/360 Operating System: Programmer's Guide to Debugging, Form C28-6670. It describes the location of control information, the interpretation of dumps, and the tracing of program flow (via the save area chain and the trace table).

```
**********************
*       ENT     IOLOG  *
*       ENT     CPLOG  *
*CPLOG DC      0      *     Entry point for call processing.
**********************
        STO     CPACC       Beginning of user's tracing routine.
        STX   1 CPXR1+1     Save status of calling routine.
        STS     CPST
*
        LDX   L1 *+1        Prepare to log the control parameter in the trace
        MDX     STORE       table and return.
*
        LD      CPACC       Restore status of calling routine.
CPXR1   LDX   L1 *-*
CPST    LDS     0
**********************
*       BSC   I  CPLOG  *   Return to calling routine.
**********************
CPACC   BSS     1           Storage for saving accumulator contents.
POINT   DC      TRABL       Define pointer to next available space in
LIMIT   EQU     400           trace table, and the size of the table.
COUNT   DC      LIMIT       Define counter for number of items that can
TABLG   DC      LIMIT         be stored, and reinitializing constant.
*
STORE   STO   I  POINT      Store accumulator contents in trace table.
        MDX   L  POINT,+1   Update pointer to next space in table.
        MDX   L  COUNT,-1   Update counter for number of items.
        MDX     STEXT       Return if table is not full.
*
        LD      POINT       Reinitialize pointer to beginning of trace table.
        S       TABLG
        STO     POINT
        LD      TABLG       Reinitialize counter for using the trace table again.
        STO   L1 COUNT
*
STEXT   BSC   L1 0          Return to proper instruction in tracing routine.
**********************
*IOLOG DC      0      *     Entry point for interruption processing.
**********************
        STO     IOACC       Save status of calling routine upon entry.
        STX     IOXR1+1
        STS     IOST
*
        LDX   L1 *+1        Prepare to log interruption information in
        MDX     STORE         trace table and return.
*
IOEXT   LD      IOACC       Restore status of calling routine.
IOXR1   LDX     *-*
IOST    LDS     0
**********************
*       BSC   I  IOLOG  *   Return to calling routine.
**********************
IOACC   BSS     1           Storage for saving accumulator contents.
TRABL   BSS     LIMIT       Define storage for trace table.
**********************
*       END    *
**********************
```

Figure 32. Sample Tracing Routine

Figure 33 contains the functional symbols used on the autocharts in the publications. A sample flowchart and detailed explanations of the various symbols are included.

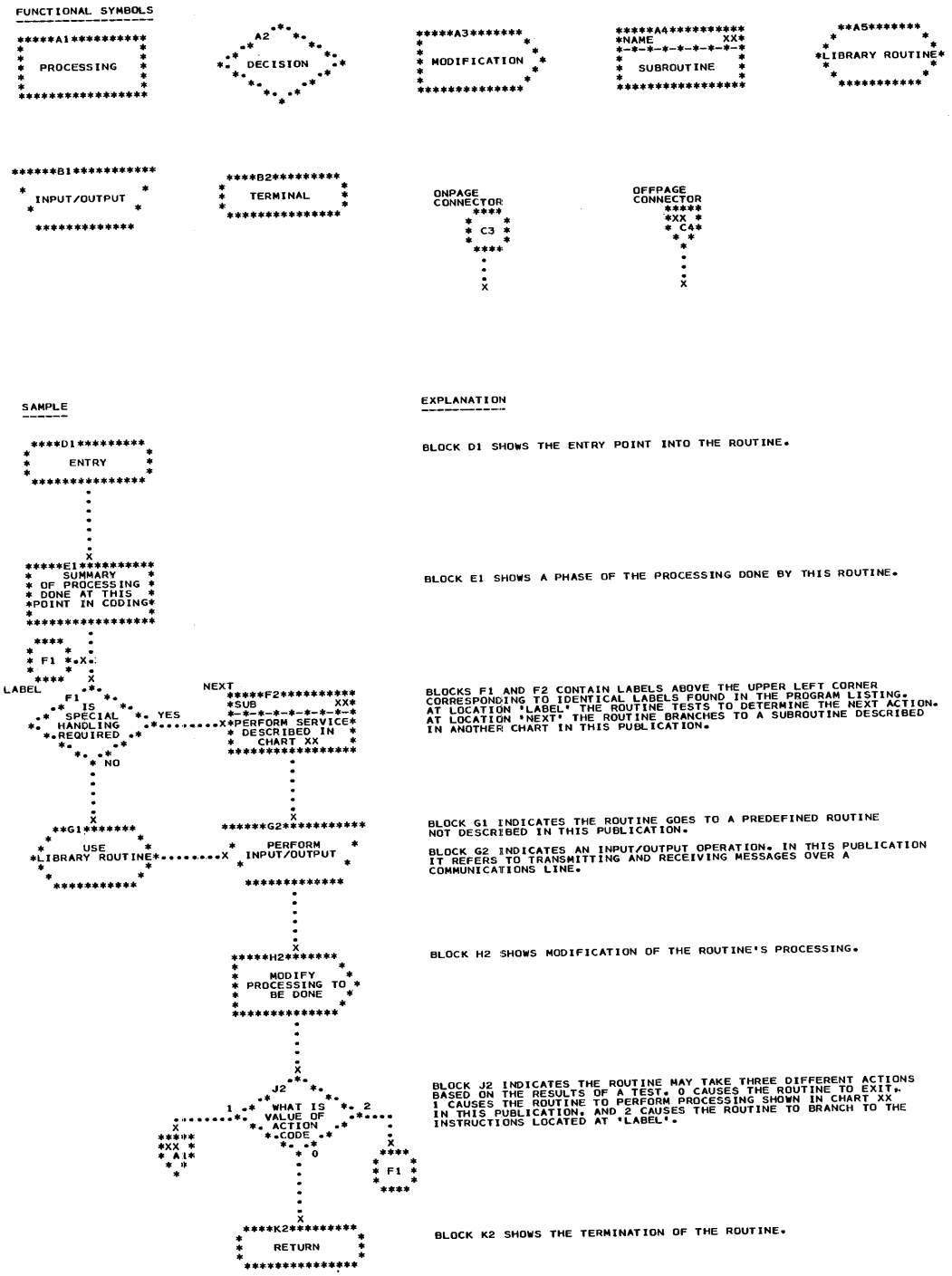


Figure 33. Autochart Symbols and Sample Flowchart

APPENDIX H: ABBREVIATIONS AND ACRONYMS

This appendix lists the abbreviations and acronyms, together with their meanings, that are used in the text, figures, and flowcharts in this publication.

acc	accumulator
ACK	affirmative acknowledgment character
addr	address
AEE	Asynchronous Exit Effector
asgnmnt	assignment
async	asynchronous
bfr	buffer
BSC	binary synchronous communications
BTAM	Basic Telecommunications Access Method
CEA	channel end appendage
chars	characters
ctrl	control
DCB	data control block
dec	decimal
DECB	data event control block
DLE	data-link escape
DSRN	data set reference number
DSW	data status word (1130)
el	element
elcount	element count
ENQ	enquiry character
EOT	end of transmission character
EP	entry point
EQ	equal
ETX	end of text character
GT	greater than
GTCB	System/360 telecommunications control block
GTCBCNT	GTCB count field in unit assignment table
GTCECB	event control block in GTCB
GTCOM	1130 telecommunications control block
hex	hexadecimal
ID	identification
initlz	initialization
intrpt	interruption
I/O	input/output
IOBDCBAD	DCB address in the IOB
IRB	interruption request block
IQE	interruption queue element
ISS	Interrupt Service Subroutine (for 1130)
lng	length
LSTRD	last read status field in GTCOM
LSTWT	last write status field in GTCOM
LT	less than
ML	mainline (program)
msg	message
NAK	negative acknowledgment character
no.	number
op	operation
optn	option
OSCNT	System/360 count field in GTCOM containing expected length of data to be read by System/360 user program
OS/360	operating system for System/360
param	parameter
proc	processing
prog	program
rcv	receive

ref	reference
reg	register
rtn	routine
SCA	synchronous communications adapter
sw	switch
SYN	synchronous idle character
S/360	System/360
TIXR	initial transparent with reset
tmt	transmit
UCB	unit control block
WTCNT	write count field in GTCOM

Where more than one page reference is given, the first page number indicates the major reference.

READER'S COMMENT FORM

IBM System/360 Operating System and 1130 Disk Monitor System:
System/360-1130 Data Transmission for FORTRAN
Program Numbers 360S-LM-542, 1130-LM-011

Form Y27-7161-0

Please check or fill in the items below, adding explanations and other comments in the space provided.

Which of the following terms best describes your job?

¤ Programmer ¤ Systems Analyst ¤ Customer Engineer
¤ Manager ¤ Engineer ¤ Systems Engineer
¤ Operator ¤ Mathematician ¤ Sales Representative
¤ Instructor ¤ Student/Trainee ¤ Other (explain) _____

Does your installation subscribe to the SRL Revision Service? ¤ Yes ¤ No

How did you use this publication?

¤ As an introduction
¤ As a reference manual
¤ As a text (student)
¤ As a text (instructor)
¤ For another purpose (explain) _____

Did you find the material easy to read and understand? ¤ Yes ¤ No (explain below)

Did you find the material organized for convenient use? ¤ Yes ¤ No (explain below)

Specific criticisms (explain below)

Clarifications on pages _____

Additions on pages _____

Deletions on pages _____

Errors on pages _____

Explanations and other comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS PLEASE . . .

This manual is one of a series which serves as reference sources for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

FOLD FOLD

FIRST CLASS
PERMIT NO. 116
KINGSTON, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
NEIGHBORHOOD ROAD
KINGSTON, N. Y. 12401

ATTN: PROGRAMMING PUBLICATIONS
DEPARTMENT 637

FOLD FOLD

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Printed in U.S.A. Y27-7167-0

Printed in U.S.A. Y27-7167-0

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]