Application Note

COMMUNICATIONS MULTIPLEXER SOFTWARE PACKAGE

017-000014-00

Ordering No. 017-000014

Printed in the United States of America
Rev. 00, August 1974

INTRODUCTION

This document describes a general purpose software subroutine package which can be used with one of the Data General Real Time Operating Systems (RTOS or RDOS) to control the operation of the Type 4060/4073 Communications Multiplexer System (CMS).

The Data General CMS enables any Data General computer to communicate with and control terminal devices over a variety of communication facilities, including both synchronous and asynchronous lines. A system is configured from a number of essential and optional subsystems which allow the system characteristics to be adjusted to match the anticipated usage. The modularity of the Type 4060/4073 hardware permits simple system expansion or the addition of new features as the need arises.

The basic features of this software package include:

- Reentrant software suitable for a multitasking environment
- Software matches the line characteristics
- Easy implementation of line or terminal procedures
- Complete compatibility with standard DGC Operating Systems
- Optional modem control capabilities

The following DGC publications will assist users of this software package:

014-000004	Type 4060 Asynchronous Multiplexer
014-000015	Programmable Synchronous Line Adapters (Type 4073/4074)
017-000002	User Device Driver Implementation in the Real Time Disk Operating System
017-000006	User Device Driver Implementation in the Real Time Operating System
093-000056	Real Time Operating System Reference Manual
093-000075	Real Time Disk Operating System User's Manual
093-000083	Introduction to the Real Time Disk Operating System
093-000093	Introduction to the Real Time Operating System

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

HARDWARE DESCRIPTION

Introduction

The Communications Multiplexer System (CMS) enables any Data General computer to communicate with and control terminal devices over a variety of communication facilities. The system is composed of one or more Type 4060 and/or Type 4073 multiplexer boards. An appropriate number of Type 4026/4027 modules can optionally be included to provide modem control and auto-answer capabilities.

The 4060/4073 modules can be intermixed to form a combination synchronous/bisynchronous/asynchronous multiplexing system. All circuitry required for the multiplexing function is distributed on interface card subsystems which each handle four lines. There is no common control circuitry; each four-line interface card is self-contained, including all the circuitry necessary to receive, transmit and buffer characters for 4 lines. From 1 to 16 cards serving 4 to 64 lines operate together as a multiplexing system. A number of four-line receiver/transmitter cards appear as if they were a single I/O device connected to the computer under a single device code.

Asynchronous Line Control

The Data General 4060 Asynchronous Multiplexer enables a Data General computer to communicate with and control terminal devices over asynchronous communications facilities.

The modularity of the 4060 hardware provides the following features:

a) Four complete line interfaces on each standard subassembly card.

b) Easy system expansion from a minimum of four lines to a maximum of 64 lines.

c) Direct (current loop) connection, modem (data set) interface, or modem control for automatic answer.

d) Full duplex operation.

e) Hardware character assembly/disassembly with full character buffering.

f) Several distinct communications line speeds in a single system (75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600 baud). Custom baud rates available upon request.

Asynchronous Line Control (Continued)

g) Several transmission codes in a single system (5, 6, 7, and 8-level with 1 or 2 stop bits). One and one half stop bits are available on 5-level code only.

In communicating with the terminal or data set, the multiplexer hardware performs all character assembly and disassembly into the serial bit streams required. Start and stop bits are inserted on transmission and are stripped out on reception. Character buffering is provided on both reception and transmission so that the program has a full character time to respond without losing input data or reducing transmission rate.

The multiplexer is flexible in line capacity, transmission code, and line speed. It can accommodate from four to 64 full duplex asynchronous lines, in multiples of four, at speeds including 9600 baud. The transmission code structure (character size and number of stop bits) and line speeds are selectable by the user so that an installation can be reconfigured with minimal hardware change.

Synchronous Line Control

The 4073 Synchronous/Bisynchronous Line Adapter (SLA) provides an interface between a Data General computer and a Synchronous line. Each 4073 contains 4 lines without modem control circuitry. The 4026/4027 interface board is available for modem control and will interface up to 8 full/half duplex auto-answer data sets.

The basic function of the 4073 line adapter is to provide an interface between a processor and a line, performing character assembly and disassembly into a serial bit stream. Assembled characters are passed to the processor on a program interrupt basis. One complete character of buffering is provided on both reception and transmission so that the program has a full character time to respond without losing input data or reducing transmission rate. SYN characters are detected by the hardware on reception and SYN or DLE SYN sequences are inserted as needed on transmission. On reception, an I/O instruction reads a word containing a line number, a character, and control information. On transmission, an I/O instruction reads a line number, indicating that a character has been transmitted. The program must respond by outputting a word containing the line number and the next character. Multiplexing occurs since the I/O instruction to read line number/character/control words always affects only one line on the several cards. The choice of a specific card to be read is made automatically by the hardware in priority order, lower line numbers having the higher priority. Thus, in a system with mixed data rates, the higher speed lines should be assigned to lower line numbers.

Synchronous Line Control (Continued)

The transmission code structure (character size, SYN, and DLE characters) are selectable by the user under program control so that an installation can be reconfigured with no hardware change. The SLA enforces no control discipline: such discipline is the responsibility of the program.

A clock is provided for systems requiring an external clock for their modems or for systems containing no modem. Square waves are available at 19.2K baud, 9600 baud, 4800 baud, 2400 baud, and 1200 baud. A pair of synchronous line adapters installed in computers up to a few hundred feet apart provide a data path not requiring a modem.

Modem Control

In order to use a four-line receiver/transmitter card with Bell System 103 Type data sets (modems) or equivalent equipped for automatic answer, additional modem control circuitry is required. This circuitry is not required on lines with manual answer or for dedicated (leased) lines. The control must be provided using the type 4026 interface subassembly with two type 4027 interfaces for each group of four lines controlled. The 4027 interface provides control of the Data Terminal Ready (Circuit CD) and permits detection of the Ring Indicator (Circuit CE) and Data Set Ready (Circuit CC). Note that the Clear to Send signal and the Carrier Detector signal (Circuit CF) carry identical signals in normal 103 type modem operation. A maximum of four 4027 interfaces can be supplied with each 4026 interface subassembly, which is sufficient to service 8 lines.

In order to use the four-line receiver/transmitter card with Bell System 202 Type data sets (modems) or equivalent equipped for automatic answer on a two wire (half duplex) line or in a multipoint network, control circuitry is provided for Request-to Send (Circuit CA) and detection of Data Carrier (Circuit CF).

*****

CHAPTER 2

SOFTWARE PACKAGE OVERVIEW

The Communications Multiplexer (CMS) software subroutine package is implemented as a runtime defined device driver for use with one of Data General's Real Time Operating Systems (RTOS or RDOS). RDOS is a real time disk-based operating system that controls real time applications through a powerful multitasking system. RTOS is a core-based, compatible subset of RDOS. The CMS software package makes effective use of these operating systems, needing only the interrupt servicing provided by them and the multitasking capability. These operating systems do provide support for the other non-communications type devices such as card readers, line printers, plotters, disks, magnetic tapes, etc., so that the user will find it most beneficial to use one of these standard, off-the-shelf Data General systems.

The CMS package is procedure-independent and therefore procedure routines must be written to provide the functions of message assembly/disassembly, code conversion and control character detection and generation. A simple interface is provided in the driver for this line procedure. A Teletype * line procedure is provided as part of the standard software package. In addition, an interface exists between a multi-tasking application and the driver/line procedures.

Additional information necessary to modify or to implement other line procedures is provided in Chapter 5 of the document.

To understand fully the operation of the CMS software package, it is helpful to follow data and program control as it passes from the hardware to the user data buffer and from the user data buffer out to one of the communication lines. The software can be thought of as being divided into three functional levels as shown in Figure 2-1.

Not shown in this figure are the various data structures that must be available for these software modules. These data structures include line tables containing the required hardware and handler-related parameters along with information required by each line procedure, control blocks, and buffer areas. An example of some of the line procedure parameters that may be defined are receiver and transmitter flags, timeout and setup information, and code conversion table pointers. The data structures will be elaborated with examples in Chapter 3.

---

*Teletype ® is a registered trademark of Teletype Corporation, Skokie, Illinois. All references to Teletype in this manual shall apply to this mark.

SOFTWARE PACKAGE OVERVIEW (Continued)

The first level provides an interface between one of the Data General Real Time Operating Systems, the CMS hardware, and the CMS interrupt servicing routines. The interrupt servicing routine QISR is entered from RTOS/RDOS for any CMS interrupt (input or output) and control remains there until all CMS interrupting lines have been cleared. Interrupt service is composed of two parts: receive and transmit.

Receive interrupts cause the associated input characters to be passed to the associated input line procedure after it is verified that the line exists and has a line table assigned. Likewise for transmit interrupts, the associated output line procedure is called to get the next character for transmission on the line.

After all receive and transmit interrupts are processed, a return is made to the operating system. It is up to each line procedure to signal the completion of receive or send control blocks to the associated tasks via the enqueuing mechanism of the basic handler.

QFCN contains a number of reentrant subroutine functions which allow the line procedures to issue commands to either the data or modem controllers on a line basis. QENQ/QPENQ are routines in the QLPIF module. These routines enqueue control blocks after blocks of data have been transmitted or received. Additionally they signal user tasks that transmission or reception of data is complete.

The second level provides a set of modules that interface the user tasks with the operating system and the CMS software handler. This level is composed of three primary sections: line/multiplexer control, input/output initiation operations, and the end of input/output operations processing.

Multiplexer control is provided by the IQMS/DQMS module which connects/disconnects the CMS handler to/from the RDOS/RTOS interrupt system using the standard operating system calls. After the handler is initialized, any interrupts generated by the CMS are processed by the interrupt service routine of the handler. The QMOD module retrieves from the hardware (associated 4026/4027) the current status of the modem connected to the indicated line. When called, the QMAA module will scan all modem-controlled lines and perform auto-answer (modem line connect) and auto disconnect functions depending on the modem status of the lines.

SOFTWARE PACKAGE OVERVIEW (Continued)

Input/output operations are initiated by passing control blocks to the handler via the QRECV and QSEND interface modules. After data has been successfully received or sent, user tasks may determine this by making calls to the QDEQ/QNDEQ modules to obtain the address of a processed data control block. A complete description of the functions performed by each of these modules is provided in Chapter 4. Calling sequences are also defined there.

The third level is the user application program. This program will consist of one or more tasks which will initiate data transfers and receptions, monitor line and modem status, and process the data provided by the handler. It will make calls to the CMS handler through macro calls to the user interface routines of level two discussed above. Organization and implementation of the user application program is described in Chapter 6. In addition, a working example of an application program is provided in Appendix C.

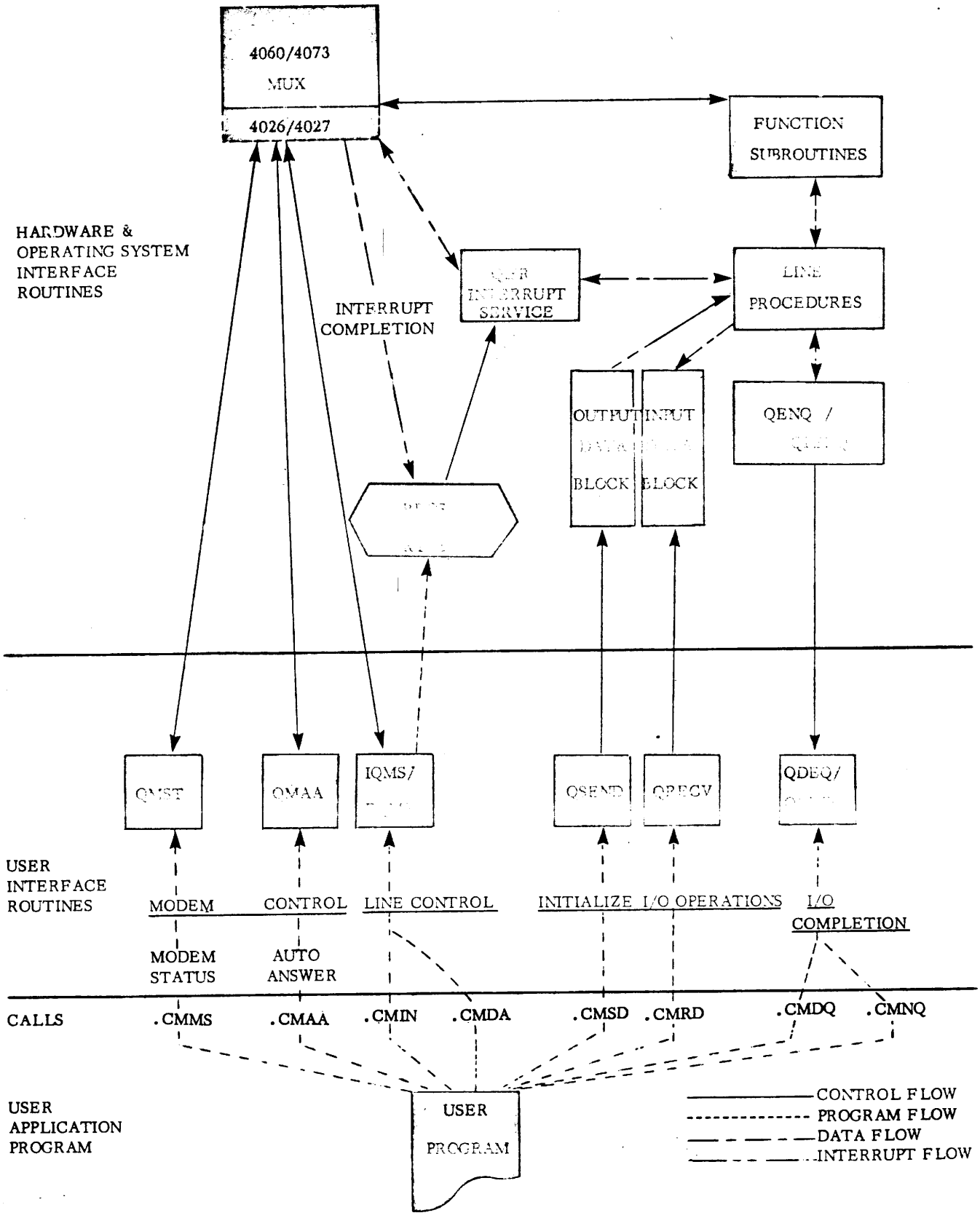


Figure 2-1 Program, Data And Control Flow Diagram

*****

CHAPTER 3

DATA STRUCTURES

In order to support the CMS subroutine package, a number of data structures (tables, parameters and buffers) are necessary. They are the following:

- Device Control Table (DCT) - required for all RTOS/RDOS handlers
- Line Table Directory (LTD) - contains pointers to all line tables
- Line Table (LT) - one table per line containing line-specific information
- Modem Table (MT) - contains I/O instructions and status for all modem-controlled lines
- Line Procedure Table (LPT) - one table per procedure relating procedure to line
- Data Control Block (DCB) - contains information controlling each line reception or transmission
- Configurable Parameters (QMIO) - driver-specific parameters
- Control Block Queue (CBQ) - contains line procedure interface information

All except the device control table, modem table and the configurable parameters must be user-supplied. These tables are all defined in the QMIO module.

Device Control Table (DCT)

There is one DCT for each CMS system, since this is required by any driver routine running under the operating system (RTOS or RDOS). The DCT and its associated storage is thirteen words long; the DCT is constructed as follows:

WORD	MNEMONIC	MEANING
	XISVA	Machine State Save Area (8 words long)
0	DCTSV	Pointer to State Save Area (XISVA)
1	DCTMS	Interrupt Service Mask
2	DCTIS	Address of Interrupt Service Routine
3	DCTHL	Highest Line Number in System (0 - 63_{10})
4	DCTDC	4060/4073 Device Code (30, 70)

The Device Control Table is defined in the QMIO module. A sample DCT and its associated storage appears below:

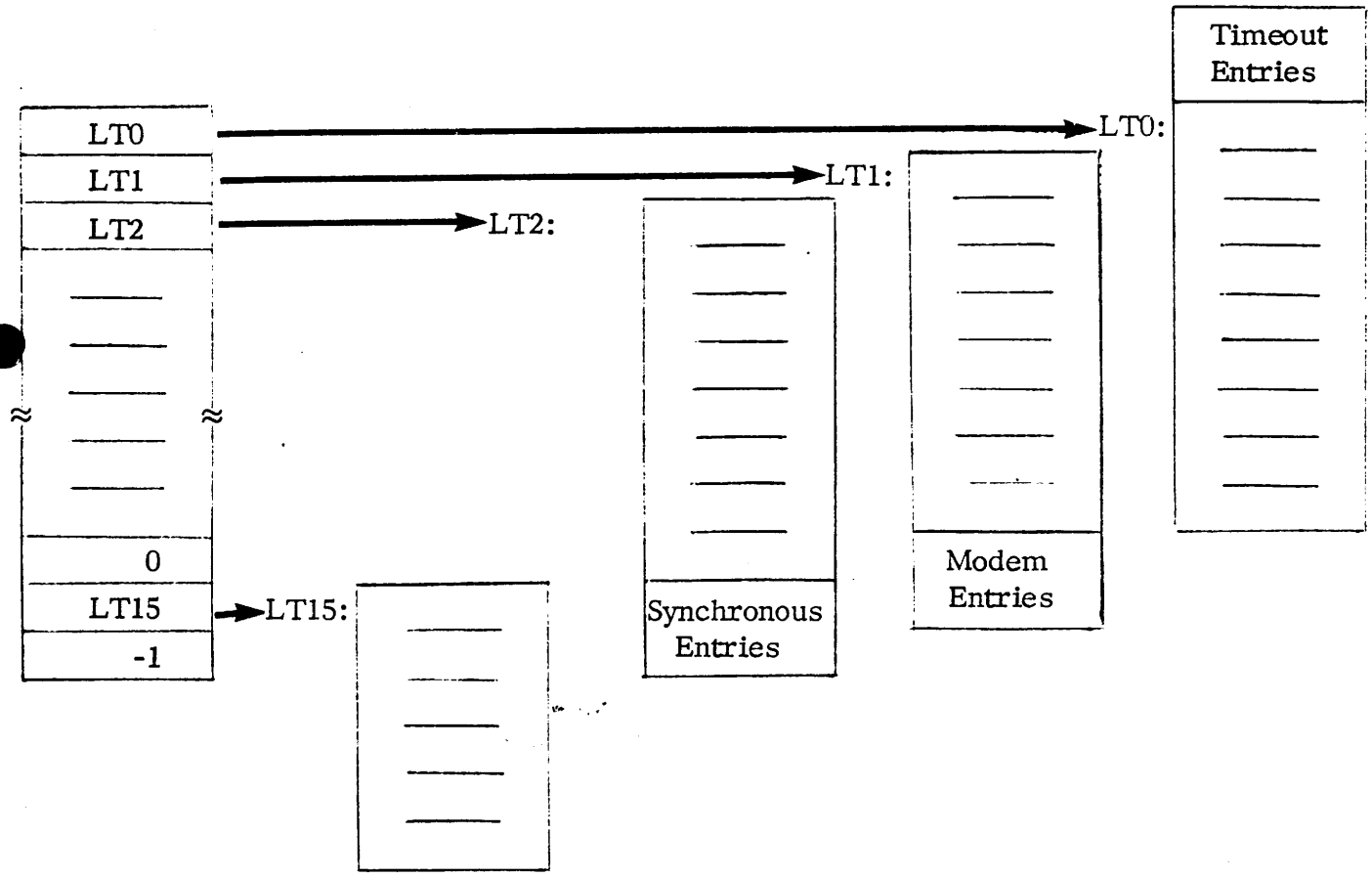
```
DTMSK = 77          ; Interrupt Service Mask
                    ; Specifies the mask priority bits that will be set
                    ; in the hardware while servicing the multiplexor
                    ; interrupt (B14 for the CMS, the others for lower
                    ; priority devices)
QHLIS = 1           ; Highest line number
QM1 = 30            ; Device Code 30
XISVA:  .BLK 10     ; State Save Area

QMDCT: XISVA        ; State Save Area Pointer
       DTMSK        ; Interrupt Service Mask
       QISR         ; Interrupt Service Routine Address
       QHLIS        ; Highest line number in system
       QM1          ; Primary device code
```

Line Table Directory (LTD)

One table is required in the system with one entry for each line table. The number of entries equals QHLIS + 2 where QHLIS = highest active line number in the system (0 - 63). Each entry in the table contains a word address pointer to a line table or is zero indicating no line table exists for this line. The last entry in the table must be equal to all ones (-1). The table is indexed by line number. The address of the table is defined by the parameter QLTDP.

An example of the relationship between the LTD and line tables follow:

Line Table Directory (LTD) (Continued)



Line Table (LT)

There is one table for each active line in the system, and each table can be any length. However there is a minimum number of words required. This number depends on whether the line table defines a procedure which is asynchronous, synchronous, modem controlled and/or requires line timeouts. See figure 3-1 for these requirements. An asterisk beside a field indicates that the value must be present at initialization time. A double asterisk indicates a field set by initialization routine IQMS. The words must be in the following order:

Line Table (LT) (Continued)

WORD	MNEMONIC	BITS	MEANING
-3	LTTTO		TRANSMITTER TIMER
		1-15	Transmitter timeout counter
-2	LTRTO		RECEIVER TIMER
		1-15	Receiver timeout counter
-1	LTTLK		TIMEOUT LINK
		1-15	Pointer to next line in timeout scan chain
0	LTLNC		HARDWARE LINE CHARACTERISTICS
		0	*Line type (Asynch/Synch = 0/1)
		1-5	**00001 (Synch line only)
		6-7	*Character size including parity bit (Synch line only) 00 - 8 bits 01 - 7 bits 10 - 6 bits 11 - illegal
		8-15	*DLE Character right-justified (Synch line only)
1	LTLPT		LINE PROCEDURE TABLE
		0-15	*Line Procedure Table Address
2	LTALA		LINE NUMBER
		0-7	**Line Number (0 - 63_{10})
		8-15	**0 (Zero)
3	LTCHR		TRANSMIT CELL
		0-1	0 Send character only. 1 Send character and leave transparent mode. 2 Send DLE plus the character and enter transparent mode. 3 Send DLE plus the character and leave transparent mode.

Line Table (LT) (Continued)

WORD	MNEMONIC	BITS	MEANING
3	(Continued)		
		2	Generate SYNs if set.
		3	Last character indicator.
		4-7	Not used.
		8-15	Output character (leading zeroes, right-justified).
4	LTSTW		LINE STATUS WORD
		0	*Line active in system
		1	*Line modem controlled
		2	*Line modem controlled with Auto Answer
		3-6	Not used
		7	**Request to Send (RTS) ON/OFF (1/0)
		8	*Receiver ON/OFF (1/0)
		9	**Data Terminal Ready (DTR) ON/OFF (1/0)
		10-13	Not used
		14	*Timeout Enable
		15	*Line full/half duplex (0/1)
5	LTRXT		RECEIVE CONTROL BLOCK
		1-15	Receive Control Block Address
6	LTRXP		RECEIVE BUFFER
		0-15	Receive Buffer Byte Pointer
7	LTTXT		TRANSMIT CONTROL BLOCK
		1-15	Transmit Control Block Address
8	LTTXP		TRANSMIT BUFFER
		0-15	Transmit Buffer Byte Pointer
9	LTSYN		SYN CHARACTER (Synch Line Only)
		0-1	**Both Set (11)
		2-7	*Number of Leading SYNs (1-31)
		8-15	*SYN Character, right-justified
10	LTSCT		SYN COUNT
		1-15	Current number of SYNs to send

Line Table (LT) (Continued)

WORD	MNEMONIC	BITS	MEANING
11	LTMIO		MODEM TABLE (modem-controlled line only)
		1-15	Modem table address
12	LTMSK		MODEM LINE MASK
		0-15	Bit mask for modem-controlled line

The basic structure of each line table is as follows:

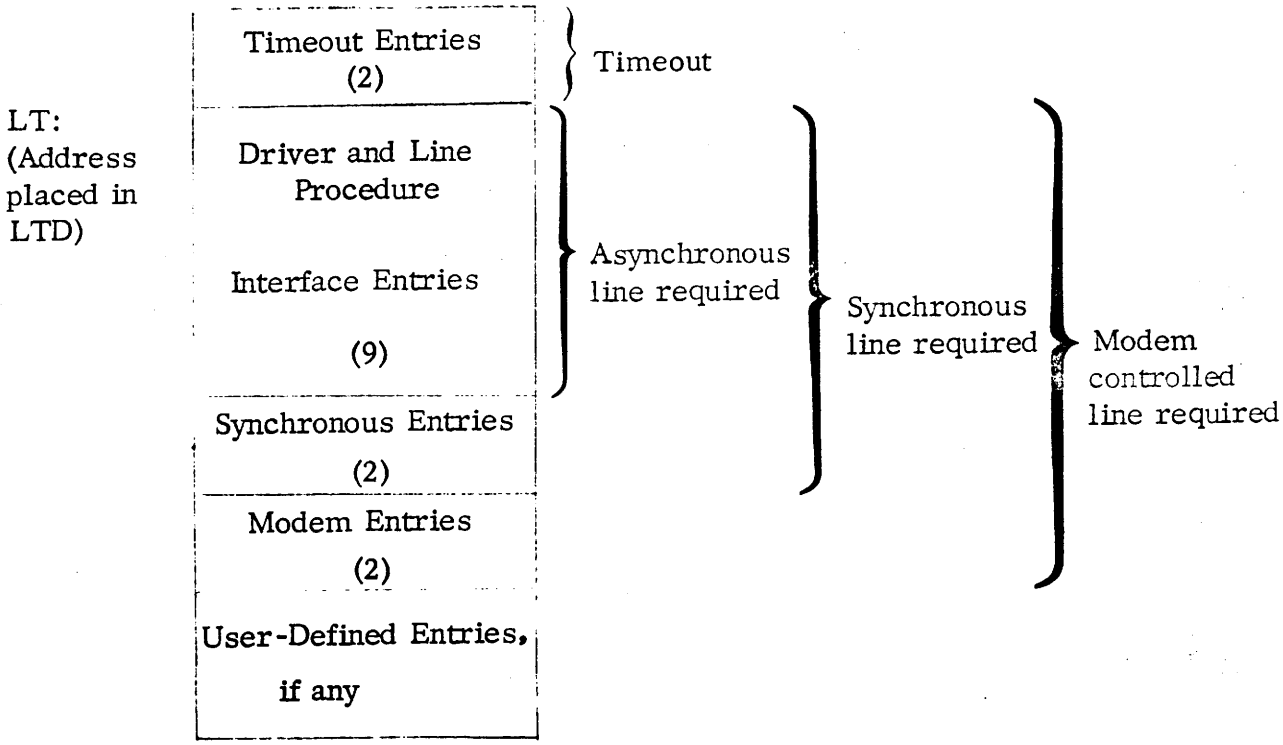


Figure 3-1 Line Table Structure

Modem Table (MT)

One modem table is required if there is at least one modem-controlled line in the system. The table contains I/O instructions (e.g., DIA) for each 4026 device in the system and space for modem status information. The table is generated by the QMIO module.

The table is composed of up to four consecutive 4 word sections - one section for each 4026 in the system. Each section contains information for 8 lines which are numbered 0-7, 8-15, 23-24, 25-31 for sections 1, 2, 3, and 4 respectively. Thus, up to 32 half/full duplex modems can be supported by the system.

The relation between entries in this table and the physical line of the multiplexor is determined by the line's line table entries. The first entry (LTMIO) is the address pointer (QMT1-QMT4) to the appropriate four word section of the modem table. The second (LTMSK) is the bit mask for the line (QML0-QML31). See the QMIO and QMPAR listings in the appendix for descriptions of the parameters.

The structure of each section is as follows (a -1 is required to indicate the end of the table):

WORD	MNEMONIC	MEANING
0	MTRIN	DIA instruction (for device 2x(x=4, 5, 6, 7) to input the Ring Indicator (RI) status (B0-B7).
1	MTDSR MTCDT	DIB instruction to input Data-Set-Ready (DSR) status (B0-B7) and Carrier Detect status (B8-15).
2	MTDTR MTRTS	DOA instruction to output Data-Terminal-Ready (DTR) status (B0-B7) and Request to Send status (B8-B15).
3	MTDVL MTRVL	DTR status (B0-B7) and RTS status (B8-B15)

Modem Table (MT) (Continued)

A sample modem table is as follows:

```
QMT1:   DIA     0,MDC       ;Device 24 - RI
        DIB     0,MDC       ;Device 24 - DSR/CD
        DOA     0,MDC       ;Device 24 - DTR/RTS
        0                   ;DTR/RTS status - lines 0 - 7
QMT2:   DIA     0,MDC+1     ;Device 25 - RI
        DIB     0,MDC+1     ;Device 25 - DSR/CD
        DOA     0,MDC+1     ;Device 25 - DTR/RTS
        0                   ;DTR/RTS status - lines 8 - 15
        -1
```

The line table for a line with a 103 type modem connected to a 4026/4027 combination with device code 25 and connected as line 12 (the fifth line on that device) is as follows:

```
LT:     .
        .
        .
LT+LTMIO        QMT2        ;Points to second section of table.

LT+LTMSK        QML12       ;=4000 (1B4 - fifth bit from left)
```

Line Procedure Table (LPT)

There must be one LPT for every line procedure in the system. A pointer to each LPT is contained in each line table. A minimum of one line table is required for each system. Each LPT must be at least two words long (three if lines are modem-controlled or four if clock functions are used) and contain the following information.

WORD	MNEMONIC	BITS	MEANING
0	LPILP	0-15	Input line procedure address
1	LPOLP	0-15	Output line procedure address
2	LPMLP	0-15	Modem line procedure address (if modem controlled)
3	LPTLP	0-15	Timeout line procedure address (if clock function used)

Line Procedure Table (LPT) (Continued)

An example of a line procedure table for a non-modem-controlled line is shown in the supplied Teletype line procedure listed in Appendix B and is of the following format:

```
QTTLP:      QIP    ;Teletype Input Processor
            QOP    ;Teletype Output Processor
```

Control Block (CB)

The control block is the mechanism required to control line transmission and reception of data via the handler. Each CB contains information for and is maintained by a line procedure. A separate CB is required for each reception or transmission on each line. The following is the required construction of each:

WORD	MNEMONIC	BITS	MEANING
0	CBLK	1-15	Used for linking CBs
1	CBBF	0-15	Start of Buffer Byte Pointer
2	CBCT	0-15	Byte Count
3	CBDL	0	Receive/Transmit flag (1/0)
		1-9	Reserved for handler
		10-15	Line number (0 - 63)
4	CBRT	0-15	Temporary storage (return pointer)

Configurable Parameters (QMIO)

Several parameters must be specified when the driver is included in a system load, and are in addition to the information needed in other data structures (i.e., LTD, LT, LPT). These additional parameters are defined in the QMIO module.

QM1	- 4060/4073 Device Code (30 or 70)
QM3	- Modem Controller (4026) Device Code (24)
QHLIS	- Highest Line Number in system (0 - 63_{10})
QMLNM	- Line number mask (77)
QMDCT	- Device Control Table Address
QMT1-QMT4	- Modem Table Addresses

The following user-supplied parameter is also required:

QLTDP	- Line Table Directory Address

Control Block Queue (CBQ)

The Control Block Queue is maintained by the line procedure interface programs and includes the following information contained in the QLPIF module.

QMCL - The cell used in the .IXMT, .REC calls by the line procedure interface.

QMAD - The control block queue header which describes the queue. It is two words long and is structured as follows:

WORD	MEANING
0	Start of queue (SOQ) pointer (-1 or first control block on queue)
1	End of queue (EOQ) pointer (-1 or last control block on queue)

*****

CHAPTER 4

BASIC HANDLER

The Communications Multiplexer Software Package contains a number of modules making up the basic handler. These permit tasks to communicate on a block or message level basis with any active line in the system. The calls can be made from any task. Unless specified otherwise, the following format is used to describe all CMS calls.

AC(s)	-	input accumulator parameters
		call mnemonic
		error return (if any)
		normal return
AC(s)	-	accumulator outputs

If the error return is reserved and is taken, an appropriate error code will be returned in AC0. AC3 contains the contents of USP (User Stack Pointer) upon any return. Except for AC3, if output accumulator contents are not specified, they are not defined; output accumulator contents are not restored unless it is stated explicitly that they are restored.

The following routines make up the package:

.CMIN	-	Initialize the CMS (IQMS module)
.CMDA	-	Deactivate the CMS (IQMS module)
.CMSD	-	Send Data (QSEND module)
.CMRD	-	Receive Data (QRECV module)
.CMDQ	-	Dequeue Control Block with Pend (QLPIF module)
.CMNQ	-	Dequeue Control Block without Pend (QLPIF module)
.CMMS	-	Get Modem Status (QMOD module)
.CMAA	-	Test Auto Answer/Disconnect (QMAA module)

Any call that is to be issued must be referenced in an .EXTN statement.

Initialize the Communications Multiplexer (.CMIN)

This routine must be called before any attempt is made to use the CMS. It performs the following functions using the initial values in the line tables:

- Connects the CMS to the operating system interrupt structure (including the 4026 modem controller).
- Initializes all synchronous lines with their appropriate line characteristics.
- Turns line receivers on or off on synchronous lines.
- Turns line transmitters off on synchronous lines.
- On modem controlled lines, turns Data Terminal Ready (DTR) and Request to Send off or DTR on depending on whether line is auto-answered or not.

The routine is non-reentrant and all accumulators are destroyed. The format of this call is:

```
.CMIN
error return
normal return
```

On either return, AC1 contains the line table directory address. Possible errors resulting from a .CMIN call are:

AC0	MNEMONIC	MEANING
0	QMSCD	Illegal device code.
1	QMSIU	Device code in use.

Deactivate the Communications Multiplexer (.CMDA)

This routine should be called to remove the CMS from the operating system interrupt structure.

This routine assumes that the CMS has been initialized and all tables are still present. It is non-reentrant.

Deactivate the Communications Multiplexer (.CMDA) (Continued)

The format of this call is:

```
.CMDA
error return
normal return
```

On either return, AC1 contains the line table directory address. A possible error resulting from a .CMDA call is:

AC0	MNEMONIC	MEANING
0	QMSCD	Illegal device code.

Send Data (.CMSD)

This reentrant routine is used to send a block of data to any line on the CMS. The routine determines whether or not the line is ready for a new transmission and then attaches the block to the line and starts transmission. The block byte pointer must be inserted in the control block before a call is made to .CMSD.

.CMSD resets the type field (B0) and stores the line number in the CBDL entry of the control block and uses the CBRT entry to save the routine return address.

Required input to this call is:

```
AC0     -   line number
AC2     -   control block address
```

The format of the call is:

```
.CMSD
error return
normal return
```

On either return, AC1 contains the line table address. Possible error returns from a .CMSD call are:

Send Data (.CMSD) (Continued)

AC0	MNEMONIC	MEANING
3	QMSIA	Illegal line address (> highest line in system).
4	QMSNA	Line not active.
5	QMSLB	Line busy (send in progress, or receive in progress on half-duplex line).
7	QMSLC	Line not configured (no line table entry in the line table directory).

The normal return indicates that the first character (a SYN character if a synchronous line) has been transmitted and, if the line is modem-controlled, request-to-send is turned on. For synchronous lines, a number of SYN characters equal to the count in LTSYN will be transmitted before the line's output procedure is called.

Receive Data (.CMRD)

This reentrant routine is used to set up a line for reception of data. If the line is ready to receive a new block of data, the block is attached to the line. The block byte pointer must be inserted into the control block before a call is made to .CMRD. This routine sets the type field (B0) and stores the line number of the CBDL entry in the control block and uses the CBRT entry to save the routine return address.

Required input to this call is:

AC0 - line number
AC2 - control block address

The format of the call is:

```
.CMRD
error return
normal return
```

On either return, AC1 contains the line table address. Possible error returns from a .CMRD call are:

Receive Data (.CMRD) (Continued)

AC0	MNEMONIC	MEANING
3	QMSIA	Illegal line address (> highest line number in system).
4	QMSNA	Line not active.
5	QMSLB	Line busy (receive in progress, or send in progress on a half-duplex line).
7	QMSLC	Line not configured (no line table entry in the line table directory).

Dequeue Control Block with Pend (.CMDQ)

This reentrant routine is called to extract an entry from the top of the control block queue after being processed by the appropriate line procedure. If the queue is empty, this routine suspends the calling task; otherwise control returns with the top entry of the queue.

The format of the call is:

```
.CMDQ
empty return
normal return
```

The empty return is never taken by this routine. On the normal return, the registers contain the following:

AC0	-	the contents of the CBDL entry in the control block
AC1	-	control block address
AC2	-	return address
AC3	-	User Stack Pointer

The task is suspended until a control block becomes available. The normal return is taken when the queue becomes non-empty. The calling task is rescheduled.

Dequeue Control Block Without Pend (.CMNQ)

This reentrant routine is called to extract an entry from the top of the control block queue after being processed by the appropriate line procedure. This routine takes the empty return if the queue is empty; otherwise, control returns with the top entry of the queue.

Dequeue Control Block Without Pend (.CMNQ) (Continued)

The format of the call is:

```
.CMNQ
empty return
normal return
```

On the empty return the registers contain the following:

AC1	-	0
AC2	-	return address
AC3	-	User Stack Pointer

On the normal return the registers contain the following:

AC0	-	the contents of CBDL word in the control block
AC1	-	control block address
AC2	-	return address
AC3	-	User Stack Pointer

Get Modem Status (.CMMS)

This reentrant routine gets the current status of the requested modem signal on the indicated line.

Required input to this call is:

AC0	-	line number
AC1	-	0/1/2 for RI/DSR/CD
AC2	-	line table pointer

The format of this call is:

```
.CMMS
error return
normal return
```

AC1 contains the line table pointer on either return. On the normal return, the requested modem signal status will be in the appropriate bit in AC0 as follows:

Get Modem Status (.CMMS) (Continued)

B0	-	Ring Indicator (RI)
B1	-	Data Set Ready (DSR)
B3	-	Carrier Detect (CD)
B4-B15	-	0

Possible error returns from a .CMMS call are:

AC0	MNEMONIC	MEANING
4	QMSNA	Line not active.
10	QMSNM	Line not modem controlled.
11	QMSBA	Modem signal not present.

Auto Answer/Disconnect (.CMAA)

This non-reentrant routine examines each active modem-controlled line in the system for either an auto-answer or auto-disconnect condition. This routine should be called periodically by the user to monitor the condition of the switched lines in the system.

.CMAA requires no input and has the following format:

```
.CMAA
normal return
```

When either the auto-answer (RI set) or auto-disconnect (DSR reset while DTR is set) condition is sensed, the line procedure table (see Chapter 3) for the associated line is examined to determine whether or not a modem line procedure exists. If the user has defined this procedure, control is transferred to the procedure with AC0 containing a condition code. Bit 0 of the condition code is set if RI (ring indicator) is present or is reset for a disconnect condition. Bits 1 - 15 are not defined. See Chapter 5 for the modem line procedure interface.

If no modem line procedure has been specified in the line procedure table, .CMAA responds to line-connect/disconnect conditions in the following manner. For line-connect, the data-terminal-ready (DTR) is raised (set) to complete the connection. For line-disconnect, DTR is lowered (reset) to make the disconnection.

******

CHAPTER 5

LINE PROCEDURE INTERFACE

A line procedure routine must be associated with every active line in the system. These are the routines which perform the functions of buffer character packing and unpacking, control character checking and generating, code conversion and, in general, all terminal or procedure-related functions. There must exist at least one line procedure per system, and there may be as many as the number of lines in the system. One procedure may service any number of lines, and it need not be reentrant since it operates at interrupt level, not task level.

Each line procedure must be user-supplied and must operate with the CMS handler. The line tables, line procedure tables, control blocks and data blocks are also user-supplied.

The interface between the handler and the line procedures is composed of five parts:

1. character processing
2. control processing
3. block processing
4. modem/auto-answer and auto-disconnect processing (optional)
5. timeout procedures (optional).

Character Processing

Two types of calls are made by the handler to each line procedure, input or output. Each call is to a different entry point in the procedure determined from the line procedure table. The procedures must process these calls and must always return to the driver at the indicated points with AC2 containing the line table address.

The input procedure is called whenever a character is received on the associated line. The format of the call is:

```
AC1     -   Character (B8 - B15), B0 - B7 = 0
AC2     -   Line table address
AC3     -   Line procedure table address

JSR @ LPILP,3   ;AC3 contains the line procedure table address
normal return

AC2     -   Line table address
```

The output procedure is called after every character starts to be transmitted on the line or, when indicated by the procedure, after the character has been transmitted. With synchronous lines, the call is not made until all leading SYN characters have been transmitted.

The format of the call is:

```
AC2     -   Line table address
AC3     -   Line procedure table address

JSR @ LPOLP,3   ;AC3 contains the line procedure table address
null return
normal return

AC2     -   Line Table address
AC1     -   See the following page
```

Character Processing (Continued)

The normal return is taken to output the next character, and AC1 is set as follows:

B0-B1	Mode indication (see below).
B2	Zero.
B3	If set, call output procedure after character in B8-B15 has been transmitted, otherwise at the start of transmission.
B4-B7	Zero.
B8-B15	Character, right justified.

B0-B1 apply only to synchronous lines and indicate the following:

Value	Function
0	Send character only.
1	Send character and leave transparent mode.
2	Send DLE plus the character and enter transparent mode.
3	Send DLE plus the character and leave transparent mode.

The procedure must take the null return if it has no more characters to output. This will allow the transmitter to be turned off, and no further output interrupts will be processed for the line until the next .CMSD call is made and the transmitter is turned on again.

Control Processing (QFCN)

A number of subroutine functions are provided which allow the line procedures to issue commands to either the data or modem controllers on a line basis. They are all reentrant and maintain the line status in the LTSTW entry of the line tables.

This includes the following functions:

QF0	-	Turn receiver on. (Synch line only)
QF1	-	Turn Data Terminal Ready (DTR) on.
QF2	-	Turn receiver off. (Synch line only).
QF3	-	Turn DTR off.
QF4	-	Send character.
QF5	-	Turn Request to Send (RTS) on.
QF7	-	Turn RTS off.
QF9	-	Change line characteristics (Synch line only).

Control Processing (QFCN) (Continued)

The format of each call is:

```
AC2        -    Line table pointer
AC3        -    Return address

.EXTN  QFx      ;External Entry Point
JSR @  .QFx     ;x = (0, 1, ..., 9)
normal return

.QFx:      QFx  ;address of function
```

On return, AC2 is preserved and interrupts are enabled.

Functions QF4 and QF9 have additional requirements as follows:

QF4 assumes that the character to be sent has already been stored in the LTCHR entry for the line with B0 - B1 set appropriately (see normal return from output line procedure call for meaning).

QF9 is used for synchronous lines only and will reissue line configuration information to the line when called. This includes the character size and DLE characters contained in the LTLNC word and the SYN character in the LTSYN word of the line table.

Block Processing

When control blocks are terminated, it is necessary to start a user task to process them. The control blocks must be linked to the control block queue (CBQ) and the processing task must be signalled by issuing an .IXMT or .XMT call through the QMCL entry (see Chapter 3) in the control block queue. As a result, a task that is suspended by a previous .CMDQ call (see Chapter 4) is readied by the task scheduler.

Control block termination may result from normal completion of a reception or transmission, a timeout condition during reception or transmission, or for some other reason determined by a user task (e.g. a line connect or disconnect condition). Routines that allow control block enqueuing and task signalling for the three types of termination are described on the following page.

Block Processing (Continued)

In the case where termination is the result of a normal completion of a reception or transmission, the enqueuing process is invoked from the input or output line procedures. Two routines, CPENQ/CENQ (both part of the QLPIF module), are available to be called at interrupt level by the line procedures. CPENQ will add a control block to the top of the CBQ in LIFO (last in-first out) fashion while CENQ will add the control block to the end of the CBQ in FIFO (first in-first out) fashion.

Required input to these calls is:

AC0 - Control block address.

The format of the calls is:

```
.EXTN           CENQ, CPENQ
  .
  .
  .
JSR @           PENQ (.ENQ)
normal return
  .
  .
  .
.PENQ: CPENQ    ;Priority enqueue routine (LIFO)
.ENQ:  CENQ     ;Non-priority enqueue routine (FIFO)
```

On return, the control block will have been enqueued and the task signalled. AC2 is preserved by these routines.

If the Communications Timer Package (see Appendix D) is included in the system to perform timeout functions, the user must also supply a timeout line procedure (see Chapter 3) which is called by the user clock routine TIMER, described in Appendix D. CLENQ, part of the TKCLQ module, is provided to allow the enqueuing process to be invoked from this procedure. This would be necessary if a line timeout and the control block attached to it require processing by a task awaiting completion of a reception or transmission.

Required input to this call is:

AC0 - Control Block Address
AC1 - Pointer to the first entry in the Control Block Queue (QMCL)

Block Processing (Continued)

The format of the call is:

```
        .EXTN           CLENQ
            .
            .
            .
        JSR@            .CLNQ
        normal return
            .
            .
            .
.CLNQ:      CLENQ   ;User clock enqueuing routine
```

On return, the control block will have been put on the queue specified by the AC1 input parameter in FIFO fashion and a .IXMT will have been issued to the message address indicated by the AC1 input parameter. AC2 is saved and restored by this routine.

It is also possible for a block to require enqueuing as a result of some operation that has been deferred to a user task, for example a secondary timer function (see Appendix D).

A third type of queuing routine, TENQ (also part of the TKCLQ module), is available to be called by user tasks. TENQ allows the enqueuing process to be evoked from outside the interrupt structure.

Required input to this call is:

AC0	-	Control Block Address
AC1	-	Pointer to the first entry in the Control Block Queue (QMCL)

The format of the call is:

```
        .EXTN           TENQ
            .
            .
            .
        JSR@    .TENQ
        normal return
            .
            .
            .
.TENQ:    TENQ      ;User task enqueuing routine
```

Block Processing (Continued)

On return, the control block will have been put on the queue specified by the AC1 input parameter in FIFO fashion and a .XMT will have been issued to the message address indicated by the AC1 input parameter. AC2 is saved and restored by this routine.

Modem/Auto-Answer and Auto Disconnect Processing

Optional routines may also be provided to perform auto-answer/auto-disconnect processing for modem-controlled lines in the system. The address of each modem line procedure routine (MLP) should be stored in the third word (LPMLP entry) of the line procedure table for the line.

The auto-answer/auto-disconnect function is performed by the .CMAA routine (see Chapter 4). This routine should be called periodically, at least once a second. When either of the line connect/disconnect conditions are found to be present on a line, the .CMAA routine will call the MLP for the line if the address is found in the line procedure table.

The format of each call is:

AC0	-	Condition code (B0 = 0/1 - line connect/disconnect)
AC2	-	Line Table address
AC3	-	Line Procedure Table address

JSR@ LPMLP, 3
normal return
action return

AC2	-	Line Table Address

The normal return indicates no action and that .CMAA must continue scanning the other lines in the system.

The action return is taken by the MLP if the DTR status is to be changed. DTR will be set/reset depending upon whether it was a line-connect or disconnect respectively.

The communications timer package provides a mechanism via the secondary timer function for periodically calling the .CMAA routine. Refer to Appendix D for a more complete description.

Timeout Processing (Optional)

An optional timeout line procedure routine may also be included to provide a line timeout capability. The function of this routine and the implementation of a timer routine as a user-defined clock are described in detail in Appendix D.

*****

CHAPTER 6

APPLICATION PROGRAMMING

The following information provides some insight into the use of the CMS handler in a real time application. The example in Appendix C illustrates most of these points.

Configuration

The user must accurately translate his particular hardware and application information into CMS configuration parameters. This includes information in the QMIO module and user-supplied line table directory, line and line procedure tables.

The user must assure that modem table information in the line tables applies to the right entries in the modem table (see Chapter 3).

The line table directory is indexed by line number; thus its length must agree with the QHLIS parameter. A zero entry in the table indicates to the handler that a line does not exist.

Start/Restart

In order to initialize and start up the CMS handler, it is necessary to make the .CMIN call. The IQMS module may optionally be an overlay.

To restart the system, a call must be made first to .CMDA and then to .CMIN. .CMDA will not deactivate the lines on the device in an orderly fashion but will place the lines in an inactive state regardless of any line activity. Thus all activity should be stopped, and queues must be processed and control blocks must be removed from the lines before the CMS is disconnected from the system and communication lines.

The line tables must be loaded with the minimum required information. In particular, the line characteristics word (LTLNC) must be exactly what is desired for each line if the line is synchronous. The line tables can be extended for application or line procedure purposes.

Data Processing

After the system is initialized, it is then necessary to make a call to the line procedure interface (.CMDQ/.CMNQ) in order to process completed reception blocks, transmission blocks or to wait for the result of previous .CMSD and .CMRD calls. Generally, one task should be responsible for making the call and then routing the control blocks to other tasks for detailed processing. The use of a routing task, of course, depends on the application.

Any other tasks in the system then make calls to .CMSD, .CMRD, or .CMMS in order to pass data back and forth from the CPU to terminals. To avoid conflict on the lines, either the calling tasks assure that multiple requests are not made on the same line and allow for busy returns, or the coordination of data processing is handled by one task, the routing task mentioned above.

*****

APPENDIX A

PACKAGE PREPARATION

In addition to the CMS software package, there are three types of modules which must be supplied by the user whenever he generates an RDOS or RTOS application. These module types are:

- configuration code
- line procedure(s)
- application program

The configuration module must be created, assembled and loaded with the system. In addition to an edited version of the QMIO module, the configuration module must include the line table directory, line tables, and line procedure tables. Make certain that all tables adhere to the order and minimum requirements detailed in Chapter 3. Appendix C contains a sample configuration module for a system with two lines and one controller.

The line procedures must be coded and assembled using the QMPAR module (see the following listings) to operate according to the interface detailed in Chapter 5. Line procedure tables can be included with the line procedures if desired as long as their addresses appear in LTLPT of each line table.

The application programs too must be coded and assembled using the QMPAR module. The line procedures and application programs must include .EXTN statements for any handler subroutine called.

Handler subroutines include the following routines and names:

.CMAA	-	Auto-Answer/Disconnect
.CMIN	-	Initialize (activate) CMS
.CMDA	-	Deactivate CMS
.CMMS	-	Get modem status
.CMSD	-	Send data
.CMRD	-	Receive data
.CMDQ	-	Remove control block from top of queue (with suspend)
.CMNQ	-	Remove control block from top of queue (without suspend)
CPENQ	-	Add control block to top of input queue
CENQ	-	Add control block to end of input queue
QF0	-	Turn Receiver on
QF1	-	Turn Data Terminal ready on
QF2	-	Turn Receiver off
QF3	-	Turn Data Terminal ready off
QF4	-	Send character

QF5	-	Turn Request-to-Send on
QF7	-	Turn Request-to-Send off
QF9	-	Change the line characteristics

System Generation

In order to change any of the standard options of the CMS package or to create a save file for running under RTOS/RDOS, the user must assemble and reload modules in the package. These modules are:

QMPAR.SR	-	This source file contains handler parameter information defining device table and line table displacements, and error codes. QMPAR.SR, listed at the end of this section, must be used when assembling line procedures or applications programs.
QMIO.SR	-	This source file contains configuration information, including default parameters, the device control table, and input/output instructions. It must be edited if necessary and assembled before loading.
CM.LB	-	This is a library file which contains the modules for the CMS handler, line procedure interface and the teletype line procedure.

There are several steps to be taken to create a save file using the CMS package with RTOS or RDOS.

1) The configuration module (QMIO.SR) must be edited and assembled to create a relocatable binary. This module is listed at the end of this appendix.

2) The user's configuration module, UCM, (see CONFIG listing in Appendix C) which includes line table directory, line tables, and line procedure tables must be written and assembled to create a relocatable binary. UCM may be assembled separately or together with the QMIO module.

3) The user must then write any additional line procedures, ULP, (see QTTYLP listing in Appendix B) and application programs, UAP, (see QMTST listing in Appendix C). They then must be assembled using the QMPAR.SR module which defines CMS table symbols.

4) The save file must then be created by loading the assembled modules with the relocatable loader.

For an RTOS system, the following modules must be loaded:

RTOSGEN module, UAP, ULP, UCM, QMIO, CM.LB, RTOS1.LB, RTOS2.LB

For an RDOS system, the modules to be loaded are:

UAP, ULP, UCM, QMIO, CM.LB

In either RTOS or RDOS loads, the order in which the modules are listed above is the order in which they must be loaded.

Note 1: An external reference to .CMMS (Get Modem Status) in a user-supplied module will force the loading of the QMOD module; otherwise no modem processing will be supported.

Note 2: It must be realized that to use the CMS system, all asynchronous and synchronous lines must be configured for the same device code.

Note 3: If the CMS system is to use device code 30, the user must not have generated either the RTOS or the RDOS system to include the system QTY driver which also utilizes device code 30.

Note 4: If the CMS system is to be used with the synchronous communication software package as described in application note 017-000001, the device code of one of the systems must be changed from 30.

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
0001 QMPAR
                        .TITL   QMPAR   ; TYPE 4060/4073 CMS PARAMETER TAPE
02
03              ;   EQUATE MODULE FOR RTOS/ROOS CMS SOFTWARE PACKAGE
04
05              ;   DEFINE THE DEVICE TABLE SYMBOLS AND RELATIVE DISPLACEMENTS
06
07      000000          DCTSV = 0               ;PTR TO MACHINE STATE SAVE AREA
08      000001          DCTMS = DCTSV+1         ;INTERRUPT MASK
09      000002          DCTIS = DCTMS+1         ;INTERRUPT SERVICE ROUTINE
10      000003          DCTHL = DCTIS+1         ;HIGHEST LINE #
11      000004          DCTDC = DCTHL+1         ;PRIMARY DEVICE CODE
12
13              ;   DEFINE THE LINE TABLE SYMBOLS AND RELATIVE DISPLACEMENT
14              ;   FOR THE DRIVER INTERFACE:
15
16      000000          LTLNC = 0               ;LINE CHARACTERISTICS
17      000001          LTLPT = LTLNC+1         ;LINE PROCEDURES TABLE PTR
18      000002          LTALA = LTLPT+1         ; LINE  NUMBER
19      000003          LTCHR = LTALA+1         ;TRANSMIT CELL
20      000004          LTSTW = LTCHR+1         ;STATUS WORD
21
22              ;   FOR PROCEDURES INTERFACE:
23
24      000005          LTRXT = LTSTW+1         ;RECEIVE CONTROL BLOCK PTR
25      000006          LTRXP = LTRXT+1         ;RECEIVE BUFFER BYTE PTR
26      000007          LTTXT = LTRXP+1         ;TRANSMIT CONTROL BLOCK  PTR
27      000010          LTTXP = LTTXT+1         ;TRANSMIT BUFFER BYTE PTR
28      000011          LTSYN = LTTXP+1         ;SYN CHARACTER +SYN COUNT
29      000012          LTSCT = LTSYN+1         ;SYN COUNT
30      000013          LTMIO = LTSCT+1         ;MODEM I/O INS
31      000014          LTMSK = LTMIO+1         ;MODEM LINE MASK
32
33              ;       FOR CLOCK(TIMEOUT) INTERFACE
34
35      177777          LTTLK = LTLNC-1         ;TIMEOUT LINK PTR
36      177776          LTRTO = LTLNC-2         ;RECEIVE TIMEOUT CELL
37      177775          LTTTO = LTLNC-3         ;TRANSMIT TIMEOUT CELL
38
39              ;   DEFINE THE LINE PROCEDURE TABLE SYMBOLS AND
40              ;   RELATIVE DISPLACEMENT:
41
42      000000          LPILP = 0               ;INPUT LINE PROCEDURE ADDR
43      000001          LPOLP = LPILP+1         ;OUTPUT LINE PROCEDURE ADDR
44      000002          LPMLP = LPOLP+1         ;MODEM LINE PROCEDURE ADDR
45      000003          LPTLP = LPMLP+1         ;TIMEOUT LINE PROCEDURE ADDR
46
47              ;   DEFINE THE CONTROL BLOCK SYMBOLS AND RELATIVE DISPLACEMENT:
48
49      000000          CBLK = 0                ;LINK
50      000001          CBBF = CBLK+1           ;BUFFER BYTE PTR
51      000002          CBCT = CBBF+1           ;BUFFER BYTE COUNT
52      000003          CBUL = CBCT+1           ;RX/TX FLAG + LINE NUMBER
53      000004          CBRT = CBUL+1           ;WORK CELL
54
```

Figure A-1 Type 4060/4073 Multiplexer Parameter Listing (QMPAR)

```
                LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10002 QMPAR
01                      ;       DEFINE THE ERROR CODES
02
03          000000              QMSCD = 0           ;ILLEGAL DEVICE CODE
04          000001              QMSIU = 1           ;DEVICE CODE IN USE
05          000003              QMSIA = 3           ;ILLEGAL LINE NO.
06          000004              QMSNA = 4           ;LINE NOT ACTIVE
07          000005              QMSLB = 5           ;LINE  BUSY
08          000006              QMSMR = 6           ;MODEM NOT READY
09          000007              QMSLC = 7           ;LINE NOT CONFIGURED
10          000010              QMSNM = 10          ;NO MODEM ON LINE
11          000011              QMSBA = 11          ;MODEM SIGNAL NOT PRESENT
12          000030              QMUNM = 30          ;RDOS ERROR CODE(ERUNM)=ILL.D.C.
13
14                      ;       DEFINE THE MODEM TABLE SYMBOLS AND REL. DISPLACEMENTS
15
16          000000              MTRIN = 0           ;RI INSTRUCTION
17          000001              MTDSR = MTRIN+1     ;DSR INSTRUCTION
18          000002              MTDTR = MTDSR+1     ;DTR INSTRUCTION
19          000003              MTDVL = MTDTR+1     ;DTR VALUE
20          000000              MTCTS = MTRIN       ;CTS  INSTRUCTION
21          000001              MTCDT = MTCTS+1     ;CDT INSTRUCTION
22          000002              MTRTS = MTCDT+1     ;RTS INSTRUCTION
23          000003              MTRVL = MTRTS+1     ;RTS VALUE
24
25                      ;       DEFINE THE MODEM LINE TABLE MASKS
26          100000              QML0 = 100          ;LINE 0
27          040000              QML1 = 101          ;LINE 1
28          020000              QML2 = 102          ;LINE 2
29          010000              QML3 = 103          ;LINE 3
30          004000              QML4 = 104          ;LINE 4
31          002000              QML5 = 105          ;LINE 5
32          001000              QML6 = 106          ;LINE 6
33          000400              QML7 = 107          ;LINE 7
34          100000              QML8 = QML0         ;LINE 8
35          040000              QML9 = QML1         ;LINE 9
36          020000              QML10 = QML2        ;LINE 10
37          010000              QML11 = QML3        ;LINE 11
38          004000              QML12 = QML4        ;LINE 12
39          002000              QML13 = QML5        ;LINE 13
40          001000              QML14 = QML6        ;LINE 14
41          000400              QML15 = QML7        ;LINE 15
42          100000              QML16 = QML0        ;LINE 16
43          040000              QML17 = QML1        ;LINE 17
44          020000              QML18 = QML2        ;LINE 18
45          010000              QML19 = QML3        ;LINE 19
46          004000              QML20 = QML4        ;LINE 20
47          002000              QML21 = QML5        ;LINE 21
48          001000              QML22 = QML6        ;LINE 22
49          000400              QML23 = QML7        ;LINE 23
50          100000              QML24 = QML0        ;LINE 24
51          040000              QML25 = QML1        ;LINE 25
52          020000              QML26 = QML2        ;LINE 26
53          010000              QML27 = QML3        ;LINE 27
54          004000              QML28 = QML4        ;LINE 28
55          002000              QML29 = QML5        ;LINE 29
56          001000              QML30 = QML0        ;LINE 30
57          000400              QML31 = QML7        ;LINE 31
58
59                              .EOT
```

Figure A-1 QMPAR.SR Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0001 QMIO
01
02
03
04                ;       THIS MODULE IS USED TO INITIALIZE CONFIGURATION
05                ;       PARAMETERS WITHIN THE CMS (4060/75) SOFTWARE PACKAGE.
06                ;       ANY OF THE FOLLOWING PARAMETERS CAN BE CHANGED
07                ;       FOR THE CONFIGURATION DESIRED:
08
09
10                ;       SELECT PRIMARY DEVICE CODE (3# OR 7#)
11
12        0000030               PRIDC = 30
13
14                ;       SELECT NUMBER OF MODEM CONTROLLERS(0=NO,1-4=YES)
15
16        0000000               MODEM = 0
17
18                ;       SELECT NUMBER OF LINES (1-64)
19
20        0000002               NUML = 2
21
22
23
24                ;       SELECT OCT INTERRUPT SERVICE MASK (E14 FOR MUX =
25                ;        + PLUS ANY LOWER PRIORITY DEVICES)
26
27        0000077               OTMSK = 1014+75
28
29                ;       ***** END OF CONFIGURABLE PARAMETERS *****
```

Figure A-2 CMS Software Configuration Parameters (QMIO)

*****

APPENDIX B

LINE PROCEDURE EXAMPLE

QTTYLP is an example line procedure servicing a teletype-like terminal. The routine is composed of two parts, input (QIP) and output (QOP), with separate entry points. A description of the routines and supporting data structures follows.

QTTYLP makes use of an extended line table to hold procedural-dependent information such as cumulative block status (LTRXA, LTTXA), block byte count (LTRCT, LTTCT), message format (LTDCH) and a control flag (LTTLF). There are also four configurable parameters: YLL, to control maximum line length; YCM, character size mask to strip received or transmitted characters; YDL, the character-delete character; and YSL, a line-delete character.

The line procedure will operate in one of two modes, line or sequential. In line mode (B0 of LTDCH is set), reception or transmission is terminated and the control block is placed on the queue whenever any of the following events occur: a control character is encountered (carriage return, form feed, or null detected on transmission), the data received exceeds the maximum line length parameter, or the data received or transmitted exceeds the size of the control block. In addition, if the echo bit (B10 of LTDCH) is set, every character received is echoed (transmitted) back to the terminal. This works in a receive-only operation. The input message can also be edited, deleting a single character (rubout) or deleting an entire line (backslash). A line feed will be provided after each carriage return if indicated by B6 of LTDCH.

In sequential mode (B0 of LTDCH reset), the reception or transmission will be terminated when the byte count matches that specified by the CBCT word of the control block. The only other checking done is for line errors; no echo or line feed functions are performed.

At block termination, the line tables are reinitialized and the block status is stored in word CBRT of the control block. This word is not available to pass information to the line procedures but can be used to pass information from the line procedures to the task processing the control block queue. This word will contain a zero if no errors occurred during reception. Also, bits 9-15 will contain 6 or 22 respectively to indicate that termination was due to either an end of file (ETX) detection or line limit being exceeded. Optionally, by setting B7 of LTDCH, reception can be terminated whenever a line error is detected.

Teletype Line Procedure Line Table

Word	Mnemonic	Bits	Meaning
0	LTLNC		HARDWARE LINE CHARACTERISTICS
		0	0 (Asynchronous Line type)
		1-15	Not Used
1	LTLPT		LINE PROCEDURE TABLE
		0-15	Line Procedure Table Address (QTTLP)
2	LTALA		LINE NUMBER
		0-9	Not used
		10-15	Line Number ($0 - 63_{10}$)
3	LTCHR		TRANSMIT CELL
		0-2	Not used
		3	Last Character Indicator
		4-7	Not used
		8-15	Output Character
4	LTSTW		LINE STATUS WORD
		0-2	Line Status
		3-6	Not used
		7-9	Receiver/Transmitter Status
		10-14	Not used
		15	Line Full/Half Duplex (0/1)
5	LTRXT	1-15	RECEIVE CONTROL BLOCK ADDRESS
6	LTRXP	0-15	RECEIVE BUFFER BYTE POINTER
7	LTTXT	1-15	TRANSMIT CONTROL BLOCK ADDRESS
8	LTTXP	0-15	TRANSMIT BUFFER BYTE POINTER
9	LTRXA	0-15	RECEIVE PARAMETER (STATUS) (Note: user parameters may start here for non-sync, non-modem lines.)
10	LTRCT	1-15	RECEIVE BYTE COUNT
11	LTTXA	0-15	TRANSMIT PARAMETER (STATUS)

Teletype Line Procedure Line Table (Continued)

Word	Mnemonic	Bits	Meaning
12	LTTCT	1-15	TRANSMIT BYTE COUNT
13	LTDCH		CHARACTERISTICS WORD
		0	Line/Sequential Mode (0/1)
		1-5	Not used
		6	Line Feed Flag
		7	Block Error Terminate (Yes/No, 1/0)
		8-9	Not used
		10	Echo Mode Flag (Yes/No, 1/0)
		11-15	Not used
14	LTTLF		LINE FEED FLAG
		0	Echo Line Feed (Yes/No, 1/0)
		1-15	Not used

Teletype Line Procedure Entry Points

QTTLP	-	Line Procedure Table Address
QIP	-	Teletype Input Processor
QOP	-	Teletype Output Processor

Routine Used by Procedure Processors

CENQ	-	Add Control Block to End of Input Queue

Parameters Defined External to Procedure

YLL	-	Maximum Line Length
YCM	-	Character Size Mask
YSB	-	Error Substitute Character
YDL	-	Delimiter (Rubout)
YSL	-	Backslash

Functions Used by Procedure Processors

QF4	-	Send Character

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0001 QTTYL
                        .TITLE  QTTYLP  ; CMS TELETYPE LINE PROCEDURE
 02
 03                     .ENT    QTTLP,QOP,QIP
 04                     .ENT    TTEN,REND
 05                     .EXTN   CENQ
 06                     .EXTN   YDL,YSL,YLL,YCM,YSUB,QF4
 07
 08                     .NREL
 09
 10                     ;LINE TABLE ENTRIES
 11
 12       000011 LTRXA=LTTXP+1            ;RECEIVE PARAMETER (STATUS)
 13       000012 LTRCT=LTRXA+1            ;RECEIVE BYTE COUNT
 14       000013 LTTXA=LTRCT+1            ;TMIT PARAMETER(STATUS)
 15       000014 LTTCT=LTTXA+1            ;TMIT BYTE COUNT
 16       000015 LTDCH=LTTCT+1            ;CHARACTERISTICS WORD
 17                                       ;B0 -0/1 = LINE/SEQUENTIAL MODE
 18                                       ;B6 - LF FLAG
 19                                       ;B7 - BLOCK ERROR TERMINATE
 20                                       ;B10- ECHO MODE FLAG
 21       000016 LTTLF=LTDCH+1            ;LINE FEED FLAG
 22                                       ;B0-SET BY TIP TO ECHO LF
 23                                       ;OTHERWISE SET BY TOP
 24
 25                     ;CONTROL CHARACTERS
 26
 27       000006 YEF= 6                   ;END OF FILE(ETX)
 28       000015 YCR=15                   ;CARRIAGE RETURN
 29       000012 YLF=12                   ;LINE FEED
 30       000014 YFF=14                   ;FORM FEED
 31       000022 YLE=22                   ;LINE LIMIT ERROR
 32
 33               ;       CONFIGURABLE CONTROL CHARACTERS
 34               ;       (THESE PARAMETERS MUST BE DEFINED IN THE
 35               ;        USER APPLICATION PACKAGE AS ENTRY POINTS)
 36
 37               ;YLL            MAXIMUM LINE LENGTH
 38               ;YCM            CHARACTER SIZE MASK
 39               ;YSUB           ERROR SUBSTITUTE CHARACTER
 40               ;YDL            CHARACTER DELETE CHARACTER
 41               ;YSL            LINE DELETE CHARACTER
 42
 43
 44
 45               ; TYPE 4060 MULTIPLEXOR TELETYPE LINE PROCEDURE TABLE
 46
 47 000000'000143'QTTLP:  QIP             ; TELETYPE INPUT PROCEDURE
 48 000001'000002'        QOP             ; TELETYPE OUTPUT PROCEDURE
```

Figure B-1 Teletype Procedure (QTTYLP) Listing

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10002 QTTYL
01
02
03
04                ;               TELETYPE OUTPUT PROCESSOR
05                ;               AC2=LIT ADDR
06                ;               JSR TOP
07                ;               NULL EXIT
08                ;               EXIT-CHAR ONLY(AC1)
09
10                QOP:
11 00002'054413 TOP:    STA     3,RLOC          ;SAVE RETURN
12 00003'035010         LDA     3,LTTLF,2       ;CHECK LINE FEED FLAG
13 00004'175014         MOV#    3,3,SZR
14 00005'000411         JMP     TLAC            ;YES INSERT
15 00006'025016         LDA     1,LTTXP,2       ;GET BYTE PTR
16 00007'125015         MOV#    1,1,SNR         ;TEST IF OUTPUTTING
17 00010'000415         JMP     TYRX            ;NO TURN OFF TMITTER
18 00011'124014         COM#    1,1,SZR         ;TEST FOR END OF TXT
19 00012'000421         JMP     TOUT            ;OUTPUT NEXT CHARACTER
20 00013'004465 TOPE:   JSR     TTEN            ;TERMINATE CB
21
22                TYRN:                         ;SET UP FOR NULL FCN EXIT
23 00014'002401         JMP@    RLOC            ;RETURN
24 00015'000000 RLOC:   0                       ;SAVE RETURN
25
26                ;       INSERT LINE FEED
27 00016'024406 TLAC:   LDA     1,LF            ;OUTPUT LINE FEED CHAR
28 00017'102400         SUB     0,0             ;CLEAR LINE FEED FLAG
29 00020'041016         STA     0,LTTLF,2
30 00021'175122         MOVZL   3,3,SZC         ;TEST IF ECHOING A LINE FEED
31 00022'000004         JMP     TWE+2           ;YES-JUST SEND CHAR
32 00023'000401         JMP     TWE             ;CHAR OUTPUT ONLY EXIT
33 00024'000012 LF:     YLF                     ;LINE FEED CHAR
34
35 00025'025016 TYRX:   LDA     1,LTRXP,2       ;CHECK IF TERMINATING RECEIVE
36 00026'124014         COM#    1,1,SZR
37 00027'000405         JMP     TYRN            ;NO-NULL EXIT
38 00030'004002         JSR@    ENDR            ;YES-TERMINATE
39 00031'000765         JMP     TYRN            ;NULL EXIT
40 00032'000312'ENDR:   REND                    ;TERMINATE RECEIVE SUBR
41
```

Figure B-1 QTTYLP Listing (Continued)

```
                LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
00003 QTTYL
01
02
03                              ;
04 000033'035007 TOUT:  LDA     3,LTTXT,2       ;CB PTR TO AC3
05 000034'015402        DSZ     CBCT,3          ;TEST IF END OF TMIT
06 000035'000402        JMP     .+2             ;NO -CONTINUE
07 000036'000735        JMP     TOPE            ;YES-TERMINATE BLOCK
08 000037'133220        MOVZR   1,3             ;GET NEXT OUTPUT CHAR.
09 000040'021400        LDA     0,0,3
10 000041'101063        MOV     0,0,SNC
11 000042'101300        MOVS    0,0
12 000043'024434        LDA     1,CM
13 000044'107400        AND     0,1             ;MASK CHAR
14 000045'035007        LDA     3,LTTXT,2       ;UPDATE BYTE PTR
15 000046'011414        ISZ     LTTCT,2         ;INCREMENT BYTE COUNT
16 000047'011010        ISZ     LTTXP,2         ;INCREMENT BYTE PTR
17 000050'021015        LDA     0,LTDCH,2       ;TEST IF WRITE SEQ.
18 000051'101122        MOVZL   0,0,SZC
19 000052'000446        JMP     TWS             ;YES
20
21                      ; OUTPUT OPERATING IN WRITE LINE MODE
22
23 000053'125415                MOV#    1,1,SNR         ;TEST IF NULL CHAR
24 000054'000450                JMP     TWE             ;YES
25 000055'020417                LDA     0,CRTN          ;NO CHECK FOR CR
26 000056'100415                SUB#    0,1,SNR
27 000057'000453                JMP     TRC             ;YES
28 000060'020415                LDA     0,FFEED         ;CHECK FOR FORM FEED
29 000061'100415                SUB#    0,1,SNR
30 000062'000442                JMP     TWE             ;YES
31 000063'020002                LDA     0,LGTH          ;NO CHECK MAX LINE
32 000064'035414                LDA     3,LTTCT,2       ;LENGTH
33 000065'110414                SUB#    0,3,SZR
34 000066'000442                JMP     RTX             ;NO
35 000067'020407                LDA     0,NLL1          ;YES SET ERROR CODE
36 000070'035013                LDA     3,LTTXA,2       ;UPDATE TMIT STATUS
37 000071'117000                ADD     0,3
38 000072'035013                STA     3,LTTXA,2
39 000073'000431                JMP     TWE
40 000074'000015 CRTN:  YCR                             ;CARRIAGE RETURN
41 000075'000014 FFEED: YFF                             ;FORM FEED
42 000076'000022 NLL1:  YLE                             ;LINE LIMIT ERROR
43 000077'177777 CM:    YCM                             ;CHAR. MASK
```

Figure B-1 QTTYLP Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
16004 QTTYL
01
02
03                ;       END OF TRANSMISSION- ENQUE CONTROL BLOCK
04
05 00100'054417 TTEN:    STA     3,TLOC          ;SAVE RETURN
06 00101'102400          SUB     0,0             ;ZERO
07 00102'041010          STA     0,LTTXP,2       ;BYTE PTR
08 00103'035007          LDA     3,LTTXT,2       ;CB ADDR TO INDEX
09 00104'025014          LDA     1,LTTCT,2       ;SAVE CT IN CB
10 00105'125400          INC     1,1             ;ADD 1 TO ADJUST FOR FIRST CHAR
11 00106'045402          STA     1,CBCT,3
12 00107'041014          STA     0,LTTCT,2       ;INIT. CT.
13 00110'041007          STA     0,LTTXT,2       ;CB ADDR = 0
14 00111'025010          LDA     1,LTTXA,2
15 00112'045404          STA     1,CBRT,3
16 00113'041010          STA     0,LTTXA,2       ;ZERO STATUS
17 00114'101000          MOV     3,0             ;CB ADDR TO AC0
18 00115'000003          JSR@    .ENQ            ;ENQUE CB
19 00116'002401          JMP@    TLOC            ;RETURN
20 00117'000001 TLOC:    .BLK    1               ;SAVE RETURN
21
22                ;   OUTPUT OPERATING IN WRITE SEQUENTIAL MODE
23
24 00120'000402 TWS:     LDA     3,CBCT,3        ;VERIFY COUNT
25 00121'102000          ADC     0,0             ;MINUS ONE
26 00122'100004          ADD     3,0,SZR         ;SKIP IF COUNTS EQUAL
27 00123'000405          JMP     RTX             ;EXIT
28
29 00124'102000 TWE:     ADC     0,0             ;SET TO -1 FOR LAST CALL
30 00125'041010          STA     0,LTTXP,2
31 00126'025413          LDA     0,LCBT          ;GET LAST CHAR BIT
32 00127'101000          ADD     0,1             ;SET IN CHARACTER
33 00130'034005 RTX:     LDA     3,RLOC          ;RESTORE RETURN
34 00131'000401          JMP     1,3             ;EXIT TO OUTPUT CHAR
35
36 00132'025410 TWC:     LDA     0,LAC           ;CHECK IF LF
37 00133'035015          LDA     3,LTOCH,2       ;NEEDED
38 00134'100400          AND     3,0,SNR
39 00135'000407          JMP     TWE             ;NO SET LAST CHAR FLAG
40 00136'041016          STA     0,LTTLF,2       ;YES SET FLAG
41 00137'000771          JMP     RTX             ;GO TO OUTPUT
42 00140'000100 LAC:     100                     ;LF MASK
43 00141'000001 LCBT:    103                     ;LAST CHARACTER FLAG
44 00142'000377 C377:    377                     ;BYTE MASK
```

Figure B-1 QTTYLP Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10065 QTTYL
01
02                      ;               TELETYPE INPUT PROCEDURES
03                      ;                       AC1=CHAR
04                      ;                       AC2=LINE TABLE ADDR
05                      ;                       JSR TIP
06                      ;                       NORMAL EXIT
07
08
09                      QIP:
10 00143'054515 TIP:    STA     3,ILOC  ;SAVE RETURN
11 00144'035005         LDA     3,LTRXT,2       ;TEST IF RECEIVER ACTIVE
12 00145'175000         MOV     3,3,SNR         ;CB PTR EQ 0
13 00146'002514         JMP@    ILOC            ;YES-RETURN
14 00147'035000         LDA     3,LTRXP,2       ;TEST IF RECEIVER TERMINATE
15 00150'174015         COM#    3,3,SNR         ;IN PROGRESS
16 00151'002505         JMP@    ILOC            ;YES-IGNORE CHARACTER
17 00152'000426         JMP     T10             ;NO OK
18
19                      ;               STORE CHARACTER SUBROUTINE
20                      ;                       AC1 = CHARACTER
21                      ;                       AC2 = LINE TABLE POINTER
22
23 00153'054423 STCR:   STA     3,CLOC          ;SAVE RETURN POINTER
24 00154'050023         STA     2,LTP           ;SAVE LT POINTER
25 00155'024722         LDA     0,CM            ;GET CHARACTER MASK
26 00156'107400         AND     0,1             ;MASK CHARACTER
27 00157'044470         STA     1,CHR           ;SAVE CHARACTER
28 00160'035000         LDA     3,LTRXP,2       ;GET BUFFER POINTER
29 00161'030761         LDA     2,C377          ;LOAD BYTE MASK
30 00162'175222         MOVZR   3,3,SZC         ;WORD ADDR TO AC3
31 00163'151300         MOVS    2,2             ;SWAP MASK TO RIGHT BYTE
32 00164'021400         LDA     0,0,3           ;LOAD BUFFER WORD
33 00165'140403         AND     2,0,SNC         ;MASK OFF BYTE
34 00166'125300         MOVS    1,1             ;POSITION CHARACTER
35 00167'125000         ADD     1,0             ;MERGE CHARACTER AND BUFFER
36 00170'041400         STA     0,0,3           ;STORE BACK
37 00171'030406         LDA     2,LTP           ;RESTORE LT POINTER
38 00172'035005         LDA     3,LTRXT,2       ;LOAD CB PTR IN AC3
39 00173'011040         ISZ     LTRXP,2         ;UPDATE BYTE POINTER
40 00174'011012         ISZ     LTRCT,2         ;UPDATE BYTE COUNT
41 00175'024401         JMP@    .CLOC           ;RETURN
42 00176'000000 CLOC:   0                       ;SAVE RETURN ADDR
43 00177'000000 LTP:    0                       ;SAVE LT POINTER
44
45
46 00200'004753 T10:    JSR     STCR            ;STORE CHARACTER INTO BLOCK
47 00201'021015         LDA     0,LTDCH,2               ;GET SEQ INDIC.
48 00202'101112         MOVL#   0,0,SZC         ;TEST FOR READ SEQ
49 00203'000526         JMP     TXS             ;YES
```

Figure B-1 QTTYLP Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10000 QTTYL
01
02
03
04                ;    INPUT OPERATING IN READ LINE MODE
05
06                ;                    TEST IF ECHO REQUIRED
07 00204'024442           LDA     1,KEY           ;GET MASK
08 00205'123400           AND     1,0
09 00206'040452           STA     0,ECHO          ;SET ECHO MODE
10 00207'101005           MOV     0,0,SNR         ;TEST IF ECHO
11 00210'000413           JMP     TNE             ;NO
12 00211'024443           LDA     0,ROUT          ;GET RUBOUT CHAR
13 00212'024435           LDA     1,CHR
14 00213'100415           SUB#    0,1,SNR         ;TEST FOR RUBOUT
15 00214'000534           JMP     TRBT            ;YES PROCESS
16 00215'024440           LDA     0,BSL           ;TEST FOR BACKSLASH
17 00216'100415           SUB#    0,1,SNR
18 00217'000521           JMP     TSHL            ;YES
19 00220'024437           LDA     0,EOF           ;CHECK FOR END OF FILE
20 00221'100415           SUB#    0,1,SNR
21 00222'000524           JMP     TEOF            ;YES
22 00223'024424 TNE:      LDA     1,CHR           ;RELOAD CHAR
23 00224'024426           LDA     0,CR            ;CHECK FOR CARRIAGE RETURN
24 00225'100415           SUB#    0,1,SNR
25 00226'000433           JMP     TCR             ;YES
26 00227'024424           LDA     0,FF            ;CHECK FOR FORM FEED
27 00230'100415           SUB#    0,1,SNR         ;
28 00231'000502           JMP     TEOB            ;YES TERMINATE BLOCK
29 00232'024402           LDA     0,CBCT,3        ;BYTE COUNT TO AC0
30 00233'035412           LDA     3,LTRCT,2       ;
31 00234'110415           SUB#    0,3,SNR         ;TST FOR MAX INPUT BYTE COUNT
32 00235'000525           JMP     TLL1            ;YES
33 00236'024407           LDA     0,LGTH
34 00237'110415           SUB#    0,3,SNR         ;TST IF LINE LENGTH EXCEEDED
35 00240'000522           JMP     TLL1            ;YES
36 00241'024417           LDA     0,ECHO          ;TEST FOR ECHO MODE
37 00242'101005           MOV     0,0,SNR
38 00243'000445           JMP     TTRN            ;NO=RETURN
39 00244'000442           JMP     TBT1            ;YES- GO OUTPUT CHAR
40
41 00245'177777 LGTH:    YLL                     ;LENGTH OF LINE
42 00246'000040 KEY:     1B10                    ;ECHO CONTROL BIT
43 00247'000000 CHR:     0                       ;SAVE CHAR
44 00250'177777 .ENQ:    CENQ                    ;ENQUE ROUTINE
45 00251'177777 .F4:     GF4                     ;SEND CHARACTER FUNCTION
46 00252'000015 CR:      YLR                     ;CARRIAGE RETURN
47 00253'000014 FF:      YFF                     ;FORM FEED
48 00254'177777 ROUT:    YUL                     ;RUBOUT
49 00255'177777 BSL:     YSL                     ;BACKSPACE
50 00256'000000 ILOC:    0                       ;SAVE RETURN
51 00257'000000 EOF:     YEF                     ;END OF FILE CHAR
52 00260'000000 ECHO:    0                       ;ECHO MODE INDICATOR
```

Figure B-1 QTTYLP Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
16000 QTTYL
01
02
03
04
05                      ;               INSERT LINE FEED IF NEEDED
06 00201'024057 TCR:    LDA     0,LAC           ;GET LF
07 00202'125015         LDA     1,LTDCH,2       ;GET CHARACTERISTICS
08 00203'123403         AND     1,0,SNR         ;TEST IF NEEDED
09 00204'000403         JMP     TEND            ;NO
10 00205'122520         SUBZR   0,0             ;SET B0=1
11 00206'041015         STA     0,LTTLF,2       ;YES SET LF FLAG
12 00207'024771 TEND:   LDA     0,ECHO          ;TEST IF ECHO MODE
13 00270'024757         LDA     1,CHR           ;RESTORE CHAR.
14 00271'101004         MOV     0,0,SZR
15 00272'000410         JMP     TEND1           ;YES
16 00273'025015         LDA     1,LTTLF,2       ;NO-TEST IF OUTPUT LF
17 00274'103122         MOVZL   1,1,SZC
18 00275'000403         JMP     .+3             ;YES
19 00276'000414         JSR     REND            ;NO-TERMINATE RECEIVE BLOCK
20 00277'000411         JMP     TTRN            ;EXIT
21 00300'045015         STA     1,LTTLF,2       ;CLEAR LF FLAG
22 00301'024410         LDA     1,ZLF           ;LOAD LF AND
23                      TEND1:
24 00302'122400         SUB     0,0             ;CLEAR ECHO MODE
25 00303'040755         STA     0,ECHO
26 00304'122400         ADC     0,0             ;SET TERMINATE RECEIVE FLAG
27 00305'041000         STA     0,LTRXP,2
28 00306'045003 TBT1:   STA     1,LTCHR,2       ;STORE FOR OUTPUT
29 00307'000742         JSR@    .+4             ;SEND CHARACTER
30 00310'002746 TTRN:   JMP@    ILOC            ;RETURN TO DRIVER
31 00311'000012 ZLF:    YLF                     ;LF CHARACTER
32
33 00312'054410 REND:   STA     3,SLOC          ;SAVE RETURN
34 00313'025011         LDA     1,LTRXA,2       ;INSERT INTO STATUS WORD
35 00314'035005         LDA     3,LTRXT,2       ;CB PTR TO AC3
36 00315'045404         STA     1,CBRT,3        ;SAVE RXT STATUS IN CB
37 00316'025012         LDA     1,LTRCT,2       ;SAVE COUNT IN CB
38 00317'045402         STA     1,CBCT,3
39 00320'122400         SUB     0,0
40 00321'041012         STA     0,LTRCT,2       ;CLEAR BYTE COUNT
41 00322'041000         STA     0,LTRXP,2       ;CLEAR BYTE PTR
42 00323'041005         STA     0,LTRXT,2       ;AND CB ADDR
43 00324'041011         STA     0,LTRXA,2       ;CLEAR STATUS
44 00325'101000         MOV     3,0             ;CB ADDR TO ACM
45 00326'000722         JSR@    .ENQ            ;ENQUE BLOCK
46 00327'002401         JMP@    SLOC            ;RETURN
47 00330'000000 SLOC:   0                       ;SAVE RETURN
```

Figure B-1 QTTYLP Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10000 QTTYL
01
02
03
04                ;    INPUT OPERATING IN READ SEQUENTIAL MODE (IF ANY)
05
06 000001'0134U2 TRS:     DSZ     CBCT,3          ;TEST FOR END OF RECV
07 000002'000735          JMP     TTRN            ;NO
08 000003'102400 TEOB:    SUB     0,0             ;EOB IO
09 000004'102011 BEND:    LDA     1,LTRXA,2       ;UPDATE
10 000005'107000          ADD     0,1             ;RECEIVE STATUS
11 000006'040011          STA     1,LTRXA,2
12 000007'000736          JMP     TEND
13
14                ;                 HANDLE BACKSLASH
15 000040'120401 TSHL:    LDA     1,CBBF,3        ;GET INIT BYTE PTR
16 000041'040000          STA     1,LTRXP,2       ;RESET CURRENT
17 000042'120400          SUB     1,1
18 000043'040012          STA     1,LTRCT,2       ;BYTE COUNT
19 000044'024700          LDA     1,CHR           ;ECHO BACKSLASH
20 000045'000741          JMP     TOT1
21
22                ;                 END OF FILE
23 000046'020711 TEOF:    LDA     0,EOF           ;GET ERROR PARAMETER
24 000047'000735          JMP     BEND
25
26                ;                 HANDLE RUBOUT
27 000050'010012 TRBT:    DSZ     LTRCT,2         ;DECREMENT BYTE COUNT
28 000051'010012          DSZ     LTRCT,2         ;ONLY ONCE IF ZERO
29 000052'000401          JMP     .+1
30 000053'021012          LDA     0,LTRCT,2       ;GET BYTE COUNT
31 000054'120401          LDA     1,CBBF,3        ;GET ORIGINAL BYTE PTR
32 000055'107000          ADD     0,1             ;CORRECT BYTE PTR
33 000056'040000          STA     1,LTRXP,2
34 000057'024402          LDA     1,BARR          ;GET BACK ARROW
35 000060'000726          JMP     TOT1            ;OUTPUT CHAR
36 000061'000137 BARR:    137                     ;BACK ARROW
37
38                ;                 LINE LIMIT EXCEEDED
39 000062'020402 TLL1:    LDA     0,LL1           ;GET ERROR PARAMETER
40 000063'000731          JMP     BEND
41
42 000064'000022 LL1:     YLE                     ;LINE LIMIT EXCEEDED
43                        .END
```

Figure B-1 QTTYLP Listing (Continued)

```
BARK    000301'          8/34    8/36
BEND    000334'          8/09    8/24    8/40
BSL     000255'          6/16    6/49
C377    000142'          4/44    5/29
CBBF    000001           8/15    8/31
CBCT    000002           3/05    4/1?    4/24    6/29    7/38    8/06
CBRT    000004           4/15    7/36
CENG    000250' XN       1/05    6/44
CHR     000247'          5/27    6/13    6/22    6/43    7/13    8/19
CLOC    000170'          5/23    5/41    5/42
CM      000077'          3/12    3/43    5/25
CR      000252'          6/23    6/46
CRTN    000074'          3/25    3/40
ECHO    000260'          6/09    6/36    6/52    7/12    7/25
ENDR    000062'          2/38    2/40
EOF     000257'          6/19    6/51    8/23
FF      000253'          6/26    6/47
FFEED   000075'          3/28    3/41
ILOC    000255'          5/10    5/13    5/16    6/50    7/30
KEY     000240'          6/07    6/42
LAC     000140'          4/36    4/42    7/06
LCBT    000141'          4/31    4/43
LF      000024'          2/27    2/33
LGTH    000245'          3/31    6/33    6/41
LL1     000304'          8/39    8/42
LTCHR   000003           7/28
LTDCH   000015           1/16    1/21    3/17    4/37    5/47    7/07
LTP     000177'          5/24    5/37    5/43
LTRCT   000012           1/13    1/14    5/40    6/30    7/37    7/40    8/18
                         8/27    8/28    8/30
LTRXA   000011           1/12    1/13    7/34    7/43    8/09    8/11
LTRXP   000000           2/35    5/14    5/28    5/39    7/27    7/41    8/16
                         8/33
LTRXT   000003           5/11    5/38    7/35    7/42
LTTCT   000014           1/15    1/16    3/15    3/32    4/09    4/12
LTTLP   000010           1/21    2/12    2/29    4/40    7/11    7/16    7/21
LTTXA   000013           1/14    1/15    3/36    3/38    4/14    4/16
LTTXP   000012           1/12    2/15    3/16    4/07    4/30
LTTXT   000007           3/04    3/14    4/36    4/13
QF4     000251' XN       1/00    6/45
QIP     000143' EN       1/03    1/47    5/09
QLL1    000070'          3/35    3/42
QOP     000002' EN       1/03    1/46    2/16
QTTLP   000002' EN       1/03    1/47
REND    000312' EN       1/04    2/40    7/19    7/33
RLOC    000015'          2/11    2/23    2/24    4/33
ROUT    000254'          6/12    6/48
RTX     000130'          3/34    4/27    4/33    4/41
SLOC    000333'          7/33    7/46    7/47
STCR    000155'          5/23    5/46
T10     000200'          5/17    5/46
TBT1    000300'          6/39    7/26    8/20    8/35
TCR     000251'          6/25    7/06
TEND    000007'          7/09    7/12    8/12
TEND1   000002'          7/15    7/23
TEOB    000333'          6/26    8/08
TEOF    000346'          6/21    8/23
TIP     000143'          5/10
TLL1    000352'          6/32    6/35    8/39
```

Figure B-1 QTTYLP Listing (Continued)

TLOC	000117'		4/05	4/19	4/20				
TNE	000223'		6/11	6/22					
TOP	000062'		2/11						
TOPE	000013'		2/20	3/07					
TOUT	000033'		2/19	3/04					
TRBT	000353'		6/15	8/27					
TRS	000331'		5/49	8/06					
TSHL	000346'		6/18	8/15					
TTEN	000100'	EN	1/04	2/20	4/05				
TTRN	000310'		6/38	7/20	7/30	8/07			
TWC	000132'		3/27	4/30					
TWE	000124'		2/31	2/32	3/24	3/30	3/39	4/29	4/39
TWS	000120'		3/19	4/24					
TYRN	000014'		2/22	2/37	2/39				
TYRX	000025'		2/17	2/35					
YCM	000077'	XN	1/00	3/43					
YCR	000015		1/28	3/40	6/46				
YDL	000254'	XN	1/00	6/48					
YEF	000003		1/27	6/51					
YFF	000014		1/30	3/41	6/47				
YLE	000022		1/31	3/42	8/42				
YLF	000012		1/29	2/33	7/31				
YLL	000245'	XN	1/00	6/41					
YSL	000255'	XN	1/00	6/49					
YSUB	177777	XN	1/00						
ZLF	000011'		7/22	7/31					
.BNQ	000250'		4/18	6/44	7/45				
.P4	000251'		6/45	7/29					

Figure B-1 QTTYLP Listing (Continued)

*****

APPENDIX C

APPLICATION PROGRAM EXAMPLE

The application program (QMTST) is an example of a user task which uses the CMS driver. QMTST is designed to send and receive to any line configured in the system. By means of a configurable option table (TAB), the send or receive control blocks enqueued for each line may have the control block and/or buffer printed on the teletype, and/or the message sent to the same or to another line in the system.

The program is composed of two tasks: one to start or restart the CMS, and the other to perform on-line processing of received and transmitted messages. Use is made of the .CMIN, .CMDA, .CMSD, .CMRD, .CMDQ, and .CMNQ routines to illustrate the handler's operation. Note how both the calls .CMDQ and .CMNQ are used (see label RBP) to access the control block queue. Even though the one call .CMDQ would suffice, both are included to illustrate their functions.

Two other modules are utilized for this application, BLOK and QCONFIG. BLOK maintains a queue of available control blocks and buffers for use by the QMTST program and QCONFIG is the user-supplied configuration module. It includes a description of a single date controller, two-line system using the QTTYLP line procedure. Also the TAB table and control block queue reflect the application.

This application is set up to receive on line zero (in line mode with echo), and to send to line one (in sequential mode), and then to send from line one back to line zero.

The RDOS load command necessary to create this save file is:

1) to run under RDOS:

   RLDR 1/C 2/K QMTST BLOK QMIO QCONFIG CM.LB)

2) to run under RTOS:

   RLDR/C RTOS QMTST BLOK QMIO QCONFIG CM.LB RTOS1.LB RTOS2.LB)

A listing and flowchart of the application program follow at the end of this appendix.

Figure C-1 shows how control information and data are passed between any application program, line procedure and CMS driver. Initially all buffers are linked together to form a pool. Each buffer is composed of a five-word header (which

forms a Control Block) and a contiguous portion (for data). The application program first gets a buffer from the pool (step 1) and provides its address to the .CMSD and/or .CMRD routines (step 2). These routines insert the Control Block address into the appropriate line table, and then start the operation (step 3). The line procedure inserts or extracts characters into or from the data portion (step 4). When the procedure determines that the operation is complete (or that an error has occurred), the procedure enqueues the Control Block on the Control Block queue by means of a call to CENQ or CPENQ (step 5).

The task readied by the enqueuing operation then dequeues a Control Block by issuing either a .CMNQ or .CMDQ call (step 6). The Control Block is then returned to the buffer pool (step 7) or is reused for a send operation (step 8).

The QMTST listing is reproduced in figure C-4; BLOK and QCONFIG are illustrated in figures C-5 and C-6 respectively.

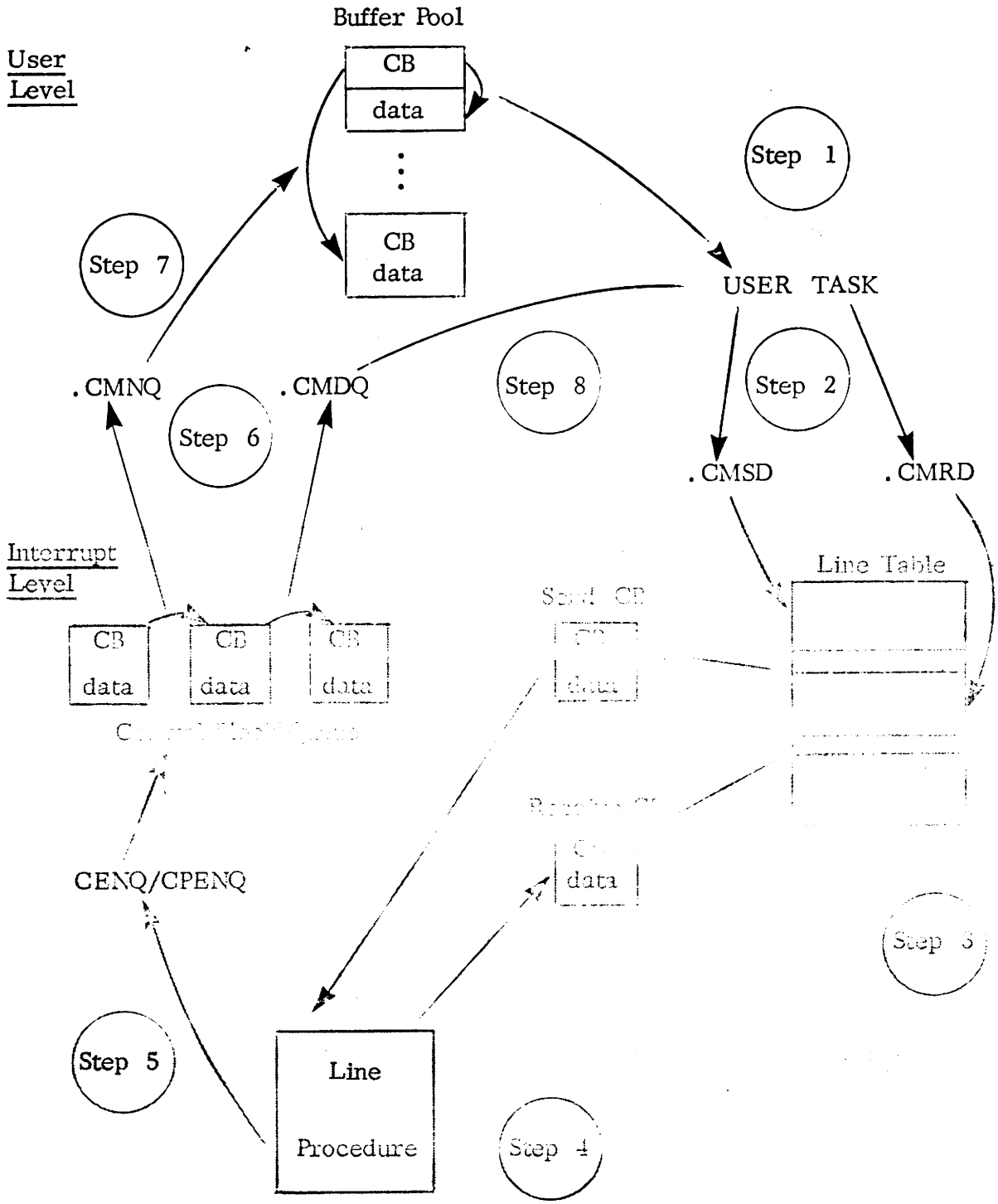


Figure C-1 Data and Control Flow Between Program, Procedure and Driver

NOTE: "CB" means
"Control Block"

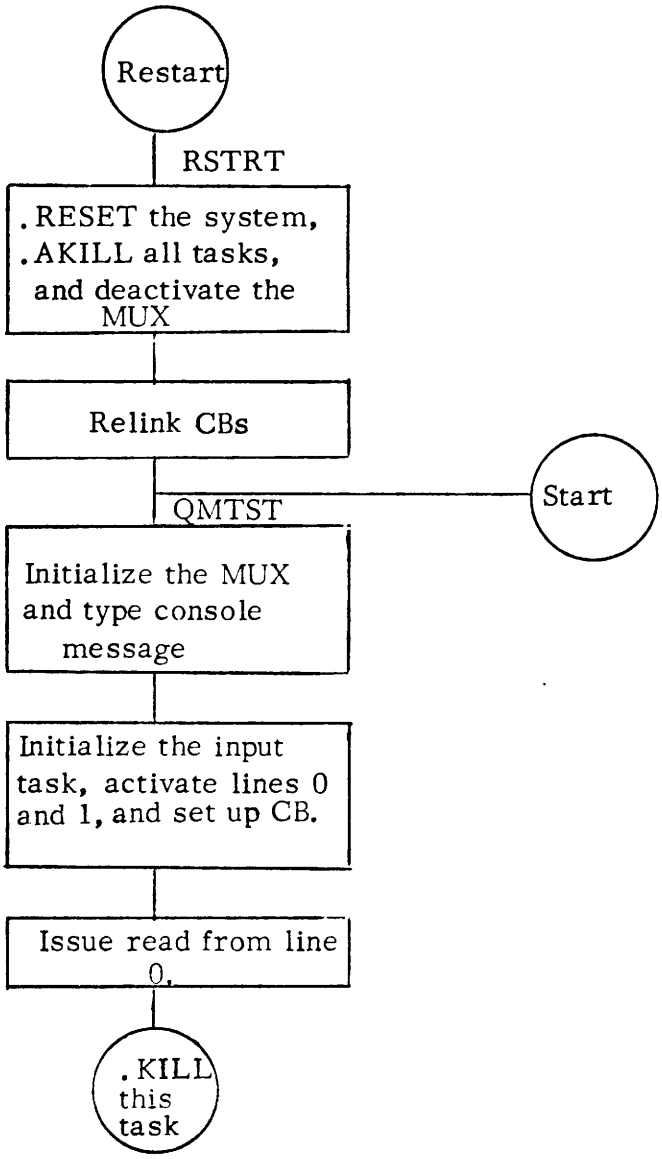


Figure C-2 QMTST Initialization and Restart Flowchart

NOTE: "CB" means
"Control Block"

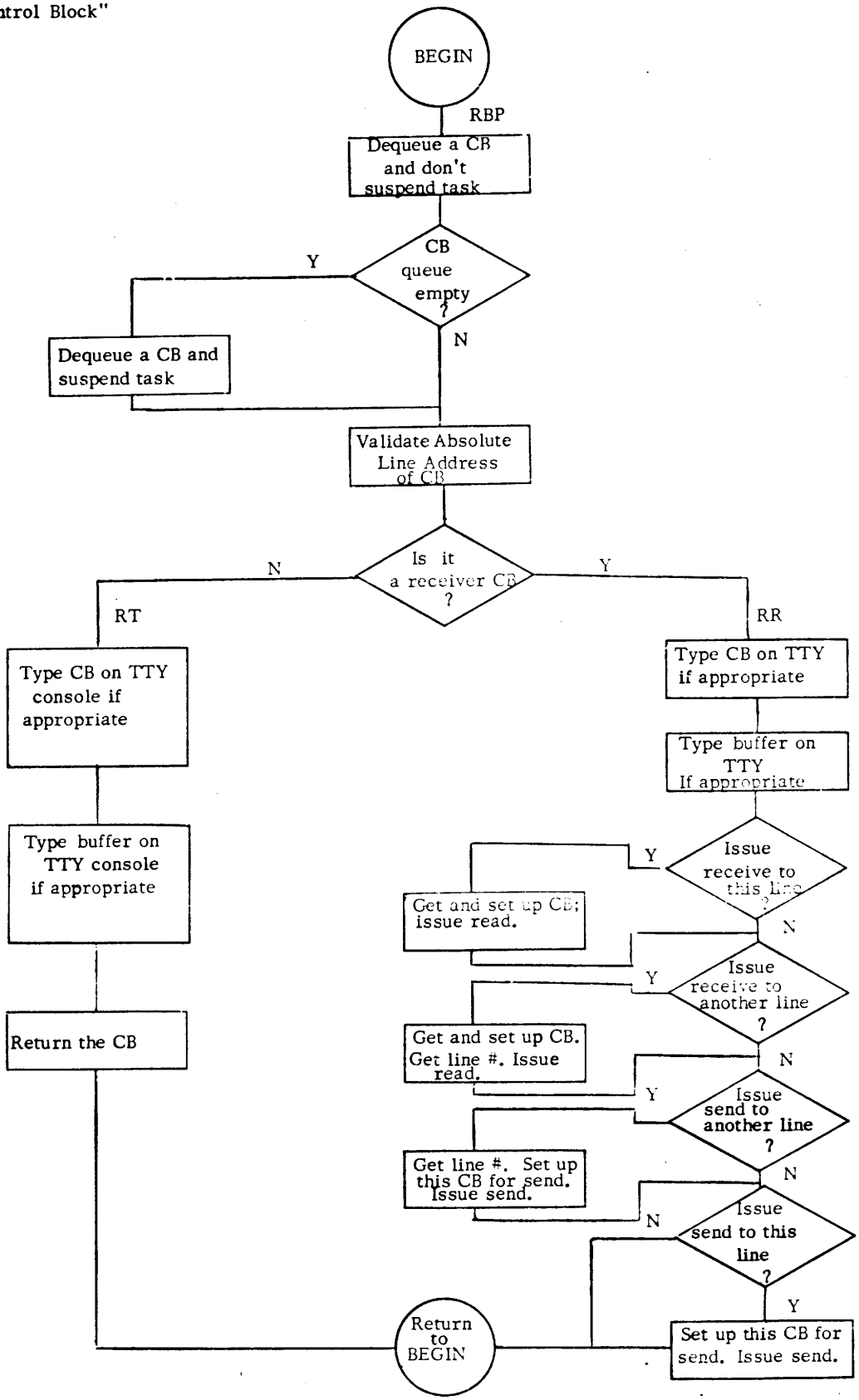


Figure C-3 QMTST Main Task Flowchart

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0001 QMTST
01
02
03
                        .TITL   QMTST   ;DEMONSTRATE 4000(CMS)
05
00                  INITIAL TASK FOR RUNNING CMS AND LINE PROCEDURE
07
06                      ;OPEN TTY CHNL
09                      ;INITIALIZE THE MUX
10                      ;WRITE INITIAL MSG TO TTY
11                      ;SCHEDULE INPUT TASK
12                      ;ACTIVATE LINES 0,1 & ISSUE RECEIVE ON LINE 0
13
14                      .ENT    RBP
15                      .ENT    QMTST,RSTRT
16                      .EXTN   .TASK,.PRI,.KILL,.AKILL
17                      .EXTN   .CMSO,.CMRD,QMLIS,.CMDA,BSOC,BEOC,BNOC
18                      .EXTN   .CMOC,.CMNC,BENC,BOEC,YLL,CB0,BFSZ
19                      .EXTN   TAB,.CMIN
20                      .NREL.
21      0000001         .TXTM   1
22 000000'020450 QMTST: LDA     0,TPNM          ;GET TTY FILE NAME
23 000001'125400        SUB     1,1
24 000002'000017        .SYSTM
25 000003'014003        .OPEN   0               ;OPEN CHL 0
26 000004'000442        JSR     ERROR           ;ERROR RETURN
27 000005'177777        .CMIN                   ;INITIALIZE CMS
28 000006'000435        JSR     ERR             ;ERROR RETURN
29 000007'020445        LDA     0,MSG1          ;WRITE MSG TO TTY
30 000010'000017        .SYSTM
31 000011'017000        .WRL    0               ;WRITE TO CHL 0
32 000012'000434        JSR     ERROR           ;ERROR RETURN
33 000013'020410        LDA     0,ITP           ;SCHEDULE
34 000014'020410        LDA     1,.RBP          ;INPUT TASK
35 000015'177777        .TASK
36 000016'000430        JSR     ERROR           ;ERROR RETURN
37 000017'000030        JSR@    .ODEQ           ;GET A CTRL BLK
38 000020'000033        JSR     ERR             ;ERROR RETURN
39 000021'020400        LDA     0,STLN          ;LINE NUMBER TO AC0
40 000022'020400        LDA     1,OBBC          ;MAX BYTE CNT
41 000023'040002        STA     1,CBCT,2        ;INTO CB
42 000024'177777        .CMRD                   ;ISSUE A RECEIVE FOR THIS LINE
43 000025'000420        JSR     ERR             ;ERROR RETURN
44 000026'177777        .KILL                   ;DELETE TASK
45 000027'000000 STLN:  0                       ;STARTING LINE NUMBER
46 000030'177777 OBBC:  YLL                     ;BUFFER BYTE COUNT
47 000031'000001 ITP:   1                       ;INPUT TASK PRIORITY  - 1
48 000032'000127'.RBP:  RBP                     ;INPUT TASK
```

Figure C-4 Multiplexer Test Program (QMTST) Listing

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
[0002 QMTST]
01
02
03                ;                 RESTART PROCEDURES
04
05 000036'000017 RSTRT:  .SYSTM                    ;TTY RESTART ENTRY POINT
06 000034'000000         .RESET                    ;CLOSE  ALL OPEN CHLS
07 000035'004411         JSR      ERROR
08 000036'020773         LDA      0,ITP
09 000037'177777         .AKILL                    ;KILL PRIOR 1 TASKS
10 000040'177777         .CMDA                     ;DEACTIVATE MUX
11 000041'004512         JSR      ERR              ;ERROR RETURN
12 000042'004426         JSR      RLCB             ;RE-LINK THE CONTROL BLOCKS
13 000043'102400         SUB      0,0              ;PRIORITY 0
14 000044'177777         .PRI                      ; SET TASK PRIORITY TO 0
15 000045'000733         JMP QMTST
16
17 000046'000017 ERROR:  .SYSTM                    ;ERROR RETURN
18 000047'000000         .ERTN                     ;IDLE SYSTEM
19 000050'000122"TPNM:   .+1*2
20 000051'022124         .TXT     'RTTO'           ;TTY FILE NAME
21        052117
22        000000
23 000054'000132"MSG1:   .+1*2
24 000055'041515         .TXT     'CMS DEMO INITIALIZED<15>'
25        051440
26        042105
27        046517
28        020111
29        047111
30        052111
31        044514
32        044532
33        042004
34        000400
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10003 QMTST
01
02
03                ;       RE-LINK THE CONTROL BLOCKS
04
05 000726'053430 RLCB:   STA     3,RTMP+1 ;SV RTN PTR
06 000727'024431         LDA     1,K4
07 000730'044425         STA     1,RTMP
08 000731'123400         INC     1,1      ;CB CTRL CELLS
09 000732'046431         STA#    1,.BNOQ  ;(RESTORE NMBR OF BLKS IN Q CNT)
10 000733'026431         LDA     0,.BFSZ  ;PLUS DATA BUFFER SIZE
11 000734'123000         ADD     1,0      ;FORM CB SIZE IN AC0
12 000735'030422         LDA     2,.CBM   ;PTR TO 1ST CB IN CHAIN TO AC2
13 000736'160000 RLC1:   MOV     0,1      ;FORM PTR TO
14 000737'147000         ADD     2,1      ;NEXT CB
15 000740'054000         STA     1,0,2    ;INSERT LINK
16 000741'014414         DSZ     RTMP     ;FINISHED
17 000742'000402         JMP     .+2      ;NO
18 000743'000403         JMP     RLC2     ;ALMOST
19 000744'151000         MOV     1,2      ;LINK THE
20 000745'000771         JMP     RLC1     ;NEXT CB
21 000746'151000 RLC2:   MOV     1,2      ;INSERT
22 000747'120402         SUB     1,1      ;LINK 0
23 000750'054000         STA     1,0,2    ;IN THE LAST CB IN THE CHAIN
24 000751'052411         STA#    2,.BEOQ  ;RESTORE EOQ PTR
25 000752'024405         LDA     0,.CBM   ;RESTORE
26 000753'042400         STA#    0,.BSOQ  ;SOQ PTR
27 000754'000202         JMP#    RTMP+1   ;RETURN
28
29 000755'000002 RTMP:   .BLK    2
30 000756'177777 .CBM:   CBM             ;PTR TO 1ST CB IN CHAIN
31 000757'000004 K4:     4
32 000760'177777 .BSOQ:  BSOQ            ;PTR TO START OF CB QUEUE
33 000761'177777 .BEOQ:  BEOQ            ;PTR TO END OF CB QUEUE
34 000762'177777 .BNOQ:  BNOQ            ;PTR TO NUMBER OF BLKS IN CB QUEUE
35 000763'177777 .BFSZ:  BFSZ            ;DATA BUFFER SIZE
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
10004 QMTST
01
02
03
04                ;       RECEIVE TASK FOR MUX TEST
05
06                ;       WILL CHECK TO SEE WHAT TO DO WITH EACH CONTROL BLOCK(CB)
07                ;       ON INPUT QUE AND DEPENDING ON LINE NO.
08                ;       OPTIONS ARE:    B0 - PRINT CB
09                ;                       B1 - PRINT BUFFER
10                ;                       B2 - ISSUE A RECV ON THIS LINE (RT)
11                ;                       B3 - ISSUE A RECV ON OTHER LINE (RO)
12                ;                       B4 - SEND BLK TO OTHER LINE (SO)
13                ;                       B5 - SEND BLK TO SAME LINE (ST)
14                ;                       B10-B15 - LINE NO. FOR SEND
15
16 00127'177777 RBP:    .CMNQ                   ;GET QUE ENTRY W/O SUSPEND
17 00130'177777          .CMDQ                   ;MT RETURN, GET ENTRY WITH SUSPE
18 00131'000401          JMP     .+1
19 00132'151000          MOV     1,2             ;CB PTR TO AC2(LN IN AC0)
20 00133'034441          LDA     3,AMK
21 00134'117400          AND     0,3
22 00135'054446          STA     3,ALN           ;SAVE LINE NO.
23 00136'022440          LDA0    0,.MLIS         ;CHECK IF GR HIGHEST LN IN SYS
24 00137'115032          ADCZ#   0,3,SZC
25 00140'000413          JSR     ERR                     ;YES- HALT
26 00141'054471          STA     2,SAV2          ;SAVE QUEUED CB ADDR
27 00142'020435          LDA     0,.TAB          ;TAB ADDR INTO AC3
28 00143'117000          ADD     0,3
29 00144'021400          LDA     0,0,3           ;OPTION IN AC0
30 00145'025003          LDA     1,CBDL,2        ;CHECK CB TYPE
31 00146'125113          MOVL#   1,1,SNC
32 00147'000021          JMP     RI              ;SEND
33 00150'000405          JMP     RR              ;RECEIVE CTRL BLK - CHECK OPTION
34
35 00151'177777 .BENQ:   BENQ                    ;ENQUEUE CB
36 00152'177777 .BDEQ:   BDEQ                    ;DEQUEUE CB
37
38                ;               ERROR EXIT ROUTINE
39 00153'003077 ERR:     HALT                    ;HALT TO EXAMINE REGISTERS
40 00154'001400          JMP     0,3             ;RETURN
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
[0000 QMTST
01
02
03
04                ;   RECEIVE CONTROL BLOCK - CHECK OPTIONS
05
06 000155'024423 RR:      LDA     1,RMSK          ;GET RECEIVE MASK
07 000156'123400          AND     1,0
08 000157'101102          MOVL    0,0,SZC         ;TYPE CB?
09 000160'004432          JSR     PRCB            ;YES
10 000161'101102          MOVL    0,0,SZC         ;TYPE BUFFER?
11 000162'004432          JSR     PRT             ;YES
12 000163'101102          MOVL    0,0,SZC         ;ISSUE RECV ON THIS LINE?
13 000164'004415          JSR     RL              ;YES
14 000165'101102          MOVL    0,0,SZC         ;ISSUE RECV ON OTHER LINE?
15 000166'004457          JSR     RNL             ;YES
16 000167'101102          MOVL    0,0,SZC         ;SEND BLK TO ANOTHER LINE
17 000170'004454          JSR     SNDL            ;YES
18 000171'101102          MOVL    0,0,SZC         ;SEND BLK TO SAME LINE
19 000172'004442          JSR     SND             ;YES
20 000173'000750          JMP     RDP             ;GET NEXT ITEM ON QUE
21
22 000174'000077 AMK:     77                      ;LINE NO. MASK
23 000175'000000 ALN:     0                       ;LINE NO. SAVE
24 000176'177777 .MLIS:   QMLIS                   ;HIGHEST LINE NUMBER
25 000177'177777 .TAB:     TAB                    ;LINE OPTION TABLE
26 000200'177777 RMSK:    177777                  ;RECEIVE OPTION MASK
27 000201'000000 MBCT:    YLL                     ;MAX BYTE COUNT FOR RECV OR SEQ
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
[0000 QMTST
01
02
03
04                ;                 RECEIVE THIS LINE
05
06 00202'054427 RL:       STA     3,USPT          ;SV RTN
07 00203'040425           STA     0,ZSV           ;SV AC0 (TAB)
08 00204'000746           JSR@    .BDEQ           ;GET A CB
09 00205'004746           JSR     ERR                     ;ERROR RTN (CK AC0)
10 00206'020767           LDA     0,ALN           ;LOAD LN IN AC0
11 00207'024772 RX:       LDA     1,MBCT          ;INIT. COUNT
12 00210'040002           STA     1,CBCT,2
13 00211'000024'          .CMRD                   ;ISSUE RECEIVE
14 00212'004741           JSR     ERR                     ;ERROR HALT
15 00213'000427           JMP     XIT             ;RETURN
16                ;                 TYPE BUFFER CONTENTS
17 00214'040414 PRT:      STA     0,ZSV           ;SAVE AC0
18 00215'054414           STA     3,USP1          ;SAVE AC3 IN UST WORD
19 00216'034414           LDA     2,SAV2          ;GET CB ADDR
20 00217'021401           LDA     0,CBBF,2        ;GET BUFFER ADDR
21 00220'000017 PW:       .SYSTM
22 00221'017000           .WRL    0               ;WRITE TO TTY
23 00222'004024           JSR     ERROR           ;ERROR RETURN
24 00223'020410           LDA     0,WCR           ;WRITE CR
25 00224'000017           .SYSTM
26 00225'017000           .WRL    0               ;WRITE TO TTY
27 00226'004024           JSR     ERROR           ;ERROR RETURN
28 00227'000413           JMP XIT                 ;RETURN
29 00230'000000 ZSV:      0                       ;SAVE AC0
30 00231'000000 USPT:     0                       ;SAVE RETURN
31 00232'000000 SAV2:     0                       ;SAVE CB ADDR
32 00233'000702"WCR:      SPC+1*2                 ;BUFFER BYTE PTR FOR CR
33                ;                 SEND THIS LINE
34 00234'054775 SND:      STA     3,USPT          ;SV RTN
35 00235'040773           STA     0,ZSV           ;SV AC0 (TAB)
36 00236'020767           LDA     0,ALN           ;ALA TO AC0
37 00237'034773 SX:       LDA     2,SAV2          ;CB TO AC2
38 00240'177777           .CMSD                   ;SEND SUBR
39 00241'004712           JSR     ERR                     ;ERROR RETURN=AC0=REASON
40 00242'020766 XIT:      LDA     0,ZSV           ;RESTORE AC0
41 00243'002766           JMP@    USPT            ;RETURN
42                ;                 SEND ANOTHER LINE
43 00244'054755 SNDL:     STA     3,USPT          ;SAVE RETURN
44 00245'040753           STA     0,ZSV           ;SV AC0 (TAB)
45 00246'034727           LDA     3,ALN           ;GET LINE NUMBER (LN)
46 00247'024730           LDA     1,.TAB          ;FROM OPTION TABLE
47 00250'137000           ADD     1,3             ;TAB ADDR TO AC3
48 00251'021400           LDA     0,0,3           ;LINE NO. FROM TAB
49 00252'034722           LDA     3,AMK
50 00253'163400           AND     3,0             ;LINE NO. TO AC0
51 00254'000763           JMP     SX              ;GO TO COMMON SEND
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
;0007 QMTST
01
02                      ;               RECEIVE OTHER LINE
03 00255'054754 RNL:    STA     3,USPT          ;SAVE RETURN PTR
04 00256'040752         STA     0,ZSV           ;SV AC0 (TAB)
05 00257'000673         JSR@    .BDEQ           ;GET A CB
06 00260'004673         JSR     ERR                     ;ERROR RTN (CK AC0)
07 00261'034714         LDA     3,ALN           ;GET LINE NUMBER (LN)
08 00262'024715         LDA     1,.TAB          ;FROM LPTION TABLE
09 00263'137000         ADD     1,3             ;TAB ADDR TO AC3
10 00264'021400         LDA     0,0,3           ;LINE NO. FROM TAB
11 00265'034707         LDA     3,AMK           ;LINE MASK
12 00266'153400         AND     3,0             ;LINE NO. TO AC0
13 00267'000720         JMP     RX              ;GO TO COMMON RECV
14                      ;               TEST SEND OPTIONS
15 00270'024421 RT:     LDA     1,TMSK          ;LOAD TRANSMIT MASK
16 00271'123400         AND     1,0
17 00272'101102         MOVL    0,0,SZC ;TYPE CB?
18 00273'004417         JSR     PRCB            ;YES
19 00274'101102         MOVL    0,0,SZC         ;TYPE BUFFER?
20 00275'004717         JSR     PRT             ;YES
21 00276'101102         MOVL    0,0,SZC         ;ISSUE A RECV ON THIS LINE?
22 00277'004703         JSR     RL                      ;YES
23 00300'101102         MOVL    0,0,SZC         ;ISSUE A RECV ON OTHER LINE?
24 00301'004754         JSR     RNL             ;YES
25 00302'101102         MOVL    0,0,SZC         ;SEND BLK TO ANOTHER LINE (SO)
26 00303'004741         JSR     SNOL            ;YES
27 00304'101102         MOVL    0,0,SZC         ;SEND BLK TO THIS LINE (ST)
28 00305'004727         JSR     SND             ;YES
29 00306'020724 RTX:    LDA     0,SAV2
30 00307'000642         JSR@    .BENQ           ;RETURN CB TO FREE STORAGE
31 00310'000617         JMP     RBP             ;GET NEXT ITEM ON QUE
32 00311'140000 TMSK:   140000                  ;TRANSMIT OPTION MASK
33                      ;               TYPE CB CONTENTS
34 00312'034717 PRCB:   STA     3,USPT          ;SAVE RETURN
35 00313'040715         STA     0,ZSV           ;SAVE AC0 AND AC2
36 00314'000422         STA     2,CB            ;INITIAL CB PTR
37 00315'020420         LDA     0,C5            ;WORD CT
38 00316'040421         STA     0,PCT
39 00317'030423         LDA     2,PBUF          ;TYPE BUFFER TO AC2
40 00320'151220         MOVZR   2,2             ;MAKE WORD PTR
41 00321'022415 P1:     LDA@    0,CB            ;GET NEXT WORD
42 00322'004421         JSR     OCTC            ;CONVERT SUBR
43 00323'020415         LDA     0,SPC           ;STORE SPACES
44 00324'041400         STA     0,0,2
45 00325'101430         INC     2,2             ;STEP BUF PTR
46 00326'010410         ISZ     CB              ;STEP WORD CT
47 00327'010410         ISZ     PCT             ;TEST IF END
48 00330'000771         JMP     P1              ;NO
49 00331'020416         LDA     0,SPC+1         ;YES STORE CR
50 00332'041000         STA     0,0,2
51 00333'020407         LDA     0,PBUF
52 00334'000634         JMP     PW              ;COMMON TYPE
53 00335'177773 C5:     -5                      ;CB SIZE
54 00336'000000 CB:     0                       ;CB PTR
55 00337'000000 PCT:    0                       ;WORD COUNT
56 00340'020040 SPC:    .TX1    *  <15>*        ;SP,SP,CR,0
57              000400
58 00342'001014'PBUF:   PBK+2
```

Figure C-4 QMTST Listing (Continued)

```
        LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
[0008 QMTST
01
02
03
04                ;                OCTAL TO ASCII CONVERSION SUBR
05
06                ;                        AC0=WORD TO CONVERT
07                ;                        AC2=BUFFER PTR
08
09 000340'054440 OCTC:   STA      3,ORTN   ;SAVE RETURN
10 000341'034433         LDA      3,C3     ;WORD CT
11 000342'024434         LDA      1,MSK1   ;MASK FOR BIT 0
12 000343'101100 OC1:    MOVL     0,0      ;SHIFT TWO FOR BITS
13 000344'101100         MOVL     0,0      ;NOTE:C BIT MUST BE PRESERVED
14 000345'040434         STA      0,SAV    ;SAVE WORD
15 000346'123400         AND      1,0      ;SAVE OCTAL
16 000347'151424         ISZ      OEVEN    ;TEST IF ODD BYTE
17 000350'000417         JMP      OC4      ;NO
18 000351'125000         LDA      1,0,2    ;YES- LOAD AND PACK WORD
19 000352'123300         MOVS     1,1
20 000353'123000         ADD      1,0
21 000354'024421         LDA      1,C60    ;GET ASCII MASK
22 000355'123000         ADD      1,0
23 000356'041000         STA      0,0,2    ;STORE INTO BUFFER
24 000357'151400         INC      2,2      ;STEP BUFFER PTR
25 000360'175400         INC      3,3      ;STEP WORD CT
26 000361'175000         MOV      3,3,SNR  ;TEST IF END
27 000362'000416         JMP@     ORTN     ;YES, RETURN
28 000363'024414 OC2:    LDA      1,MSK2   ;TRIAD MASK
29 000364'020415         LDA      0,SAV    ;GET WORD
30 000365'101100         MOVL     0,0      ;SHIFT BYTE TO RIGHT HALF
31 000366'000735         JMP      OC1      ;FINISH
32
33 000367'024413 OC4:    LDA      1,M1     ;SET ODD/EVEN FLAG TO -1
34 000370'044433         STA      1,OEVEN
35 000371'041000         STA      0,0,2    ;STORE BYTE
36 000372'000771         JMP      OC2      ;GET NEXT
37
38 000373'000000 OEVEN:  0                ;ODD/EVEN FLAG(ENDS UP =0)
39 000374'177775 C3:     -3               ;WORD CT
40 000375'000060 C60:    60000            ;ASCII MASK
41 000376'000001 MSK1:   1                ;BIT 0 MASK
42 000377'000007 MSK2:   7                ;TRIAD MASK
43 000400'000000 ORTN:   0                ;SAVE RETURN
44 000401'000000 SAV:    0                ;SAVE WORD
45 000402'177777 M1:     -1               ;MINUS ONE FOR FLAG
46
47
48 000403'000025 PBK:    .BLK     25      ;TYPE BUFFER STORAGE
49
50                       .END     QMTST
```

Figure C-4 QMTST Listing (Continued)

ALN	000175'		4/22	5/23	6/10	6/30	6/45	7/07	
AMK	000174'		4/20	5/22	6/49	7/11			
BUEU	000152'	XN	1/16	4/36					
BENU	000151'	XN	1/16	4/35					
BEU0	000124'	XN	1/17	3/33					
BFSZ	000120'	XN	1/16	3/35					
BNUU	000125'	XN	1/17	3/34					
BSUU	000123'	XN	1/17	3/32					
C3	000377'		8/10	8/39					
C5	000335'		7/37	7/53					
C00	000400'		8/21	8/40					
C0	000330'		7/36	7/41	7/40	7/54			
CB0	000121'	XN	1/16	3/30					
C0BF	000001		6/20						
C0C1	000002		1/41	0/12					
C0DL	000000		4/30						
ERR	000155'		1/20	1/30	1/43	2/11	4/25	4/39	6/09
			0/14	0/39	7/00				
ERRUR	000046'		1/20	1/32	1/30	2/07	2/17	6/23	6/27
ITP	000031'		1/33	1/47	2/00				
K4	000122'		3/00	3/31					
M1	000400'		8/33	8/45					
M0C1	000201'		5/27	6/11					
MSG1	000054'		1/29	2/23					
MSK1	000401'		8/11	8/41					
MSK2	000402'		8/28	8/42					
OBBC	000000'		1/40	1/40					
OC1	000340'		8/12	8/31					
OC2	000300'		8/28	8/36					
OC4	000372'		8/17	8/33					
OCTC	000343'		7/42	8/09					
OEVEN	000370'		8/10	8/34	8/36				
ORTN	000403'		8/09	8/27	8/43				
P1	000021'		7/41	7/46					
P0K	000400'		7/50	8/46					
P0UF	000342'		7/39	7/51	7/56				
PCT	000337'		7/38	7/47	7/55				
PRC0	000312'		5/09	7/18	7/34				
PRT	000214'		5/11	0/17	7/20				
PW	000220'		0/21	7/52					
QML10	000170'	XN	1/17	5/24					
QMTST	000000'	EN	1/04	1/15	1/22	2/15	8/50		
R0P	000127'	EN	1/14	1/40	4/10	5/20	7/31		
RL	000202'		5/13	0/00	7/22				
RLC1	000100'		3/13	3/20					
RLC2	000110'		3/10	3/21					
RLC0	000070'		2/12	3/00					
RMSK	000200'		5/00	5/20					
RNL	000200'		5/10	7/03	7/24				
RR	000150'		4/33	5/00					
RSTRT	000000'	EN	1/15	2/05					
RT	000170'		4/32	7/15					
RTMP	000111'		3/00	3/07	3/10	3/27	3/29		
RTX	000300'		7/29						
RX	000207'		0/11	7/13					
SAV	000404'		8/14	8/29	8/44				
SAV2	000232'		4/20	0/19	6/31	6/37	7/29		
SND	000234'		5/19	6/34	7/20				
SNDL	000244'		5/17	0/43	7/20				

Figure C-4 QMTST Listing (Continued)

```
                LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
  0010 QMTST

SPC     000340'          6/32    7/43    7/49    7/56
STLN    000627'          1/39    1/45
SX      000237'          6/37    6/51
TAB     000177'  XN      1/19    5/25
TMSK    000311'          7/15    7/32
TPNM    000050'          1/22    2/19
USPT    000231'          6/06    6/18    6/30    6/34    6/41    6/43    7/03
                         7/34
WCR     000233'          6/24    6/32
XIT     000242'          6/15    6/28    6/40
YLL     000201'  XN      1/18    1/46    5/27
ZSV     000230'          6/07    6/17    6/29    6/35    6/40    6/44    7/04
                         7/35
.AKIL   000057'  XN      1/16    2/09
.BDEQ   000152'          1/37    4/36    6/08    7/05
.BENQ   000151'          4/35    7/30
.BEOQ   000124'          3/24    3/33
.BFSZ   000125'          3/10    3/35
.BNOQ   000120'          3/09    3/34
.BSOQ   000123'          3/20    3/32
.CBA    000121'          3/12    3/25    3/30
.CMDA   000040'  XN      1/17    2/10
.CMDQ   000130'  XN      1/18    4/17
.CMIN   000050'  XN      1/19    1/27
.CMNQ   000127'  XN      1/16    4/16
.CMRQ   000211'  XN      1/17    1/42    6/13
.CMSQ   000240'  XN      1/17    6/38
.MLIS   000170'          4/23    5/24
.KILL   000020'  XN      1/16    1/44
.PRI    000044'  XN      1/16    2/14
.RBP    000032'          1/34    1/46
.TAB    000177'          4/27    5/25    6/46    7/08
.TASK   000015'  XN      1/16    1/35
```

Figure C-4 QMTST Listing (Continued)

```
 0001 BLOK
01
                          .TITL   BLOK
03                      USED TO:  GET / RETURN CONTROL BLOCKS
04                        .ENT    BENQ     ;RTN CB
05                        .ENT    BDEQ     ;GET CB
06                        .EXTN   BSOQ,BEOQ,BNOQ
07                        .NREL
08
09                  ;     ENQUEUE CB (RTN CB)
10                  ;     AC0 = CB ADDRESS
11                  ;     JSR   BENQ
12                  ;     NORMAL RETURN
13  0000'126400 BENQ:     SUB     1,1
14  0001'060277           INTDS
15  0002'054420           STA     2,BSV2           ;SV AC2
16  0003'054420           STA     3,BRTN           ;SV RTN PTR
17  0004'115000           MOV     0,3              ;MOVE CB TO AC3
18  0005'054400           STA     1,CBLK,3         ;SET CB LINK = 0
19  0006'032417           LDA     2,.BEOQ          ;OLD EOQ TO AC2
20  0007'050416           STA     3,.BEOQ          ;NEW EOQ = CB
21  0010'101005           MOV     2,2,SNR          ;WAS THE Q EMPTY
22  0011'000407           JMP     BMT              ;YES
23  0012'050000           STA     3,0,2            ;UPDATE AND LINK CB TO END
24  0013'012413 BEXT:     ISZ     .BNOQ            ;INCREMENT NOQ COUNT
25  0014'000401           JMP     .+1
26  0015'030405           LDA     2,BSV2           ;RESTORE AC2
27  0016'060177           INTEN
28  0017'000204           JMP     BRTN             ;RETURN TO CALL+1
29  0020'050404 BMT:      STA     3,.BSOQ          ;SOQ = CB
30  0021'000772           JMP     BEXT             ;RTN TO CALL+1
31  0022'000000 BSV2:     0                        ;SV CELL FOR AC2
32  0023'000000 BRTN:     0                        ;SV CELL FOR RTN PTR
33  0024'177777 .BSOQ:    BSOQ                     ;PTR TO SOQ
34  0025'177777 .BEOQ:    BEOQ                     ;PTR TO EOQ
35  0026'177777 .BNOQ:    BNOQ                     ;PTR TO NOQ
36                  ;     DEQUEUE CB    (GET CB)
37                  ;     JSR     BDEQ     GET A CB
38                  ;     EMPTY RETURN     AC2 = 0
39                  ;     NRML RTN         AC2 = CB
40                  ;                      AC1 = NOQ
41                  ;                      AC0 = NXT Q ENTRY
42  0027'126400 BDEQ:     SUB     1,1
43  0030'060277           INTDS
44  0031'032773           LDA     2,.BSOQ          ;GET SOQ
45  0032'101005           MOV     2,2,SNR          ;TST FOR MT
46  0033'000412           JMP     MTQ              ;NO CB AVAILABLE
47  0034'030000           LDA     0,0,2            ;UPDATE
48  0035'042767           STA     0,.BSOQ          ;SOQ
49  0036'010770           DSZ     .BNOQ            ;DECREMENT NOQ COUNT
50  0037'000402           JMP     .+2
51  0040'042765           STA     0,.BEOQ          ;EOQ = 0 IF Q MT
52  0041'050001           STA     1,0,2            ;INSURE 0 LINK
53  0042'032764           LDA     1,.BNOQ          ;NUMBER OF BLKS ON THE Q TO AC1
54  0043'060177           INTEN                    ;ENABLE INTERRUPTS
55  0044'001401           JMP     1,3              ;NRML RETURN
56  0045'060177 MTQ:      INTEN
57  0046'001400           JMP     0,3              ;NO BLOCKS, RTN TO CALL+1
58                        .END
```

Figure C-5 Control Block Processing Routine Listing (BLOK)

```
0002 BLOK

BUEW  000027' EN    1/05    1/42
BENW  000000' EN    1/04    1/13
BEOW  000025' XN    1/06    1/34
BEX1  000013'       1/24    1/30
BMT   000020'       1/22    1/29
BNOW  000020' XN    1/00    1/35
BRTN  000023'       1/10    1/28    1/32
BSOW  000024' XN    1/06    1/33
BSV2  000022'       1/15    1/26    1/31
CBLK  000000        1/18
MTW   000045'       1/46    1/56
.BEUU 000025'       1/19    1/20    1/34    1/51
.BNUU 000020'       1/24    1/35    1/49    1/53
.BSUU 000024'       1/29    1/33    1/44    1/48
```

Figure C-5 BLOK Listing (Continued)

```
 0001 QCONF
01
02
03
                        .TITL   QCONFIG ;CONFIGURATION MODULE FOR 4060
05
06                      .ENT    QLTDP ,BSOQ,BEOQ,BNOQ
07                      .ENT    YDL,YSL, YLL, YCM, YSUB,TAB,CB0,BFSZ
08                      .EXTN   QTTLP   ; 4060 TELETYPE LINE PROCEDURE TABLE
09
10                      .NREL
11
12          000110      YLL=    72.     ;MAX LINE LENGTH(72-TTY)
13          000177      YCM=    177     ;CHARACTER MASK(ASCII-7 BIT)
14          000077      YSUB=   77      ;QUESTION MARK(SUBSTITUTE CHARACTER)
15          000177      YDL=    177     ;DEL(RUBOUT)=CHARACTER DELETE
16          000134      YSL=    134     ;BACKSLASH  =LINE DELETE
17
18                TAB:              ;TAB ENTRIES
19
20  000000'154001       154001          ;180 =>TYPE THE CONTROL BLOCK
21                                      ;181 => TYPE THE BUFFER
22                                      ;082 => DONOT ISSUE RECV. FOR THIS LINE
23                                      ;183 => ISSUE A RECEIVE FOR LINE 1
24                                      ;184 => ISSUE A SEND FOR LINE 1
25                                      ;085 => DO NOT SEND BLOCK TO SAME LINE
26                                      ;B10-B15 => LINE NUMBER FOR SEND
27
28  000001'154000       154000          ;180 => TYPE THE CONTROL BLOCK
29                                      ;181 => TYPE THE BUFFER
30                                      ;082 => DONOT ISSUE RECV. FOR THIS LINE
31                                      ;183 => ISSUE A RECEIVE FOR LINE 0
32                                      ;184 => ISSUE A SEND FOR LINE 0
33                                      ;085 => DONOT SEND BLOCK TO SAME LINE
34                                      ;B10-B15 => LINE NUMBER FOR SEND
35
36
37                QLTDP:                ;LINE TABLE DIRECTORY
38
39  000002'000000'      LTK             ;POINTER TO LINE TABLE FOR LINE 0
40  000003'000224'      LT1             ;POINTER TO LINE TABLE FOR LINE 1
41  000004'177777       -1              ;LAST TABLE ENTRY FLAG
42
```

Figure C-6 Configuration Module Listing (QCONFIG)

```
10002 QCONF
01
02
03                ;                LINE TABLES:
04                LT0:             ;LINE TABLE ENTRIES FOR LINE ZERO
05
06                ;   FOR THE DRIVER INTERFACE - ENTRIES 0 THRU 4
07
08                                 ;ENTRY 0
09 00005'000000           0        ;LINE CHARACTERISTICS
10                                 ;B0 = ASYNCHRONOUS LINE
11                                 ;B1-B15 = NOT USED
12                                 ;ENTRY 1
13 00006'177777           QTTLP    ;POINTER TO TTY LINE PROCEDURES
14                                 ;ENTRY 2
15 00007'000000           0        ;CELL RESERVED FOR  LINE ADDRESS (LN)
16                                 ;ENTRY 3
17 00010'000000           0        ;TRANSMIT CELL
18                                 ;ENTRY 4
19 00011'000000           0        ;LINE STATUS WORD
20
21                ;   FOR THE PROCEDURES INTERFACE - ENTRIES 5 THRU 16
22                                 ;ENTRY 5
23 00012'000000           0        ;RECEIVE CONTROL BLOCK (CB) POINTER
24                                 ;ENTRY 6
25 00013'000000           0        ;RECEIVE BUFFER BYTE POINTER
26                                 ;ENTRY 7
27 00014'000000           0        ;TRANSMIT CB POINTER
28                                 ;ENTRY 10
29 00015'000000           0        ;TRANSMIT BUFFER BYTE POINTER
30                                 ;ENTRY 11
31 00016'000000           0        ;RECEIVE PARAMETER (STATUS)
32                                 ;ENTRY 12
33 00017'000000           0        ;RECEIVE BYTE COUNT
34                                 ;ENTRY 13
35 00020'000000           0        ;TRANSMIT PARAMETER (STATUS)
36                                 ;ENTRY 14
37 00021'000000           0        ;TRANSMIT BYTE COUNT
38                                 ;ENTRY 15
39 00022'001040           1040     ;CHARACTERISTICS WORD
40                                 ;B0 = 0 (LINE MODE)
41                                 ;B6 = 1 (LINE FEED FLAG)
42                                 ;B10 = 1 (ECHO MODE FLAG)
43                                 ;ENTRY 16
44 00023'000000           0        ;LINE FEED FLAG
45
```

Figure C-6 QCONFIG Listing (Continued)

```
!0003 QCONF
01
02                LT1:              ;LINE TABLE ENTRIES FOR LINE 1
03  0002A'000000                    0         ;ENTRY 0
04  0002B'000006'                   QTTLP     ;ENTRY 1
05  0002C'000003                    .BLK 3    ;ENTRIES 2-4
06  00031'000010                    .BLK 8.   ;ENTRIES 5-14
07                                            ;ENTRY 15
08  00041'100000                    100000    ;CHARACTERISTICS WORD
09                                            ;B0 = 1 (SEQUENTIAL MODE)
10  00042'000000                    0         ;ENTRY 16
11
12
13                ;                 LINKED CONTROL BLOCKS
14
15        000050                    BFSZ = 40.        ;DATA BUFFER STORAGE CELLS
16
17  00043'000120'CB0:               CB1               ;CBLK - LINK
18  00044'000120"                   CB0+5*2           ;CBBF - BUFFER BYTE PTR
19  00045'000000                    0                 ;CBCT - BUFFER BYTE CTR
20  00046'000000                    0                 ;CBDL - TYPE/LINE NUMBER (LN)
21  00047'000000                    0                 ;CBRT - WORK CELL
22  00050'000050                    .BLK     BFSZ     ;DATA BUFFER
23
24  00120'000175'CB1:               CB2
25  00121'000252"                   CB1+5*2
26  00122'000000                    0
27  00123'000000                    0
28  00124'000000                    0
29  00125'000050                    .BLK     BFSZ
30
31  00175'000252'CB2:               CB3
32  00176'000404"                   CB2+5*2
33  00177'000000                    0
34  00200'000000                    0
35  00201'000000                    0
36  00202'000000                    .BLK     BFSZ
37
38  00252'000327'CB3:               CB4
39  00253'000536"                   CB3+5*2
40  00254'000000                    0
41  00255'000000                    0
42  00256'000000                    0
43  00257'000050                    .BLK     BFSZ
44
45  00327'000000 CB4:               0                 ;LAST CB - LINK 0
46  00330'000670"                   CB4+5*2
47  00331'000000                    0
48  00332'000000                    0
49  00333'000000                    0
50  00334'000050                    .BLK     BFSZ
51
52  00434'000043'BSOQ:              CB0               ;START OF CB QUEUE
53  00435'000327'BEOQ:              CB4               ;END OF CB QUEUE
54  00436'000005 BNOQ:              5                 ;NUMBER OF CONTROL BLOCKS
55
56                                  .END
```

Figure C-6 QCONFIG Listing (Continued)

```
0004 QCONF

BEOU  000405' EN    1/06    3/53
BFSZ  000056  EN    1/07    3/15    3/22    3/29    3/36    3/43    3/50
BNOU  000400' EN    1/06    3/54
BSOU  000404' EN    1/06    3/52
CB0   000043' EN    1/07    3/17    3/18    3/52
CB1   000120'       3/17    3/24    3/25
CB2   000175'       3/24    3/31    3/32
CB3   000252'       3/31    3/38    3/39
CB4   000327'       3/38    3/45    3/46    3/53
LT0   000005'       1/39    2/04
LT1   000024'       1/40    3/02
QLTOP 000002' EN    1/00    1/37
QTTLP 000025' XN    1/08    2/13    3/04
TAB   000400' EN    1/07    1/18
YCM   000177  EN    1/07    1/13
YOL   000177  EN    1/07    1/15
YLL   000110  EN    1/07    1/12
YSL   000134  EN    1/07    1/16
YSUB  000077  EN    1/07    1/14
```

Figure C-6 QCONFIG Listing (Continued)

*****

APPENDIX D

COMMUNICATIONS TIMER PACKAGE

Each communications device driver includes two communications timer modules as part of its library. These modules contain all routines necessary to implement a communication timeout process. The RTOS/RDOS operating systems allow the definition of one user clock driven by the system Real Time Clock (RTC). If the communications timer package is used, the timer routine is defined as the one allowable user clock routine, but the user may still gain access to the timing mechanism provided by the user clock by using an optional feature called the secondary timer function.

Throughout this appendix reference is made to primary and secondary timer functions. The primary function is defined as the action taken by the timer routine with respect to all active communications lines in the system. This includes the decrementing of line timer counts, recognizing timeout conditions and calling line-specific time-out procedures when required. The secondary timer function is defined as the action taken by the timer routine with respect to a user-defined secondary timer function task. The secondary function could be used to start a system sub-function periodically e.g., calling the communications auto-answer/disconnect routines. Both timer functions are described below in more detail.

The names of the two timer modules are TMIN and TIMER. TMIN contains the initialization and deactivation routines and TIMER contains the user defined clock routine.

These modules contain three timer subroutines with the following entry points:

Entry Point	Routine Description
.TOIN	Initialize timeout functions.
.TODA	Deactivate timeout functions.
TIMER	User-defined clock routine that performs primary and secondary timer functions.

Each routine and its function is described below in a greater detail.

The requirements for implementing the communications timer functions fall into four areas:

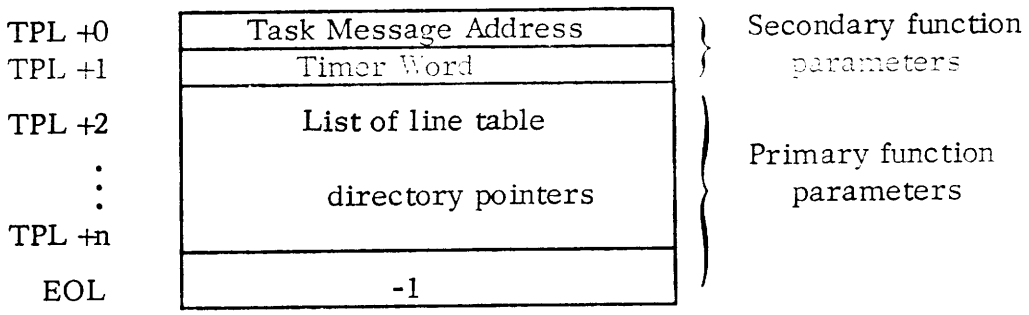
- Data Structure
- Timer Subroutines
- Line procedure interface
- Application programming

The primary timer function requires that the user make available three (3) additional line table entries. These entries are mentioned in the Chapter 3 line table description. They are located in the three words immediately preceding the line table and are referenced by means of negative index values as follows:

Mnemonic	Index Value	Entry Description
LTTTO	-3	Transmitter timer word
LTTRO	-2	Receiver timer word
LTTLK	-1	Timer link word

In addition, each line that is to be included in the timeout chain must have the time-out enable bit (B14 of LTSTW in the line table) set to a one.

The user must also provide a Timer Parameter List (TPL) formatted as follows:



The first word of the TPL is the task message address used by TIMER to communicate with the user-defined secondary timer function task and the second word (TPL +1) is the timer count associated with the secondary timer function. Both words are defined and maintained by the user. If the system does not require the secondary timer function, both words must exist and must be set to zero.

Starting at TPL +2 and continuing to EOL -1 is a list of table directory pointers. A pointer must exist in this list for each line table directory (LTD) which has any time-out-enabled lines in its directory. The TPL is terminated by a minus one (EOL = -1).

Timer Initialization (.TOIN) (Continued)

COMMUNICATIONS TIMER ROUTINES

The modules, as previously described, contain three subroutines. The following sections describe each routine and list the function it performs, and, where appropriate, give the calling sequence and error returns.

Timer Initialization (.TOIN)

This routine is contained in the TIMIN module and must be called by the user to start the communications timer functions. Depending on the parameters passed, .TOIN performs the following functions:

- Creates a timeout chain by linking the timeout-enabled lines (B14 of word LTTLK in the line table (primary function only).
- Clears timer words (LTTTO & LTTRO) in the line table (primary function only).
- Initialized secondary function routine indicator and clears secondary timer (secondary function only).

Timer Deactivation (.TODA)

This routine is contained in the TIMIN module and allows the user to deactivate one or both of the primary or secondary timer functions. Depending on the parameters passed, .TODA performs the following functions:

- Terminates the primary timer function.
- Terminates the secondary timer function.
- Removes the user clock from the system by issuing a .RUCLK system call (only if no timer functions are to remain active after this call).

The calling sequence for this non-reentrant subroutine is as follows:

```
        AC1     -   Clock interval specifier
                    0 = one second.
                    1 = one half second.
                    2 = one-quarter second.
AC0     -   Function code:
            0 = Deactivate both primary and secondary timer
                functions.
            1 = Deactivate primary timer function only.
AC2         2 = Deactivate the secondary timer function only.
            Timer parameter list address (TPL).

.TODA
normal return
```

Timer Initialization (.TOIN) Message

This routine is contained in the TIMIN module and must be called by the user to start the communications timer functions. Depending on the parameters passed, .TOIN performs the following functions:

- Creates a timeout chain by linking the timeout-enabled lines (B14 of word LTTLK in the line table (primary function only).
- Clears timer words (LTTTO & LTTRO) in the line table (primary function only).
- Initialized secondary function routine indicator and clears secondary timer (secondary function only).

This routine is contained in the TIMIN module and allows the user to deactivate one or both of the primary or secondary timer functions. Depending on the parameters passed, .TODA performs the following functions:

- Terminates the primary timer function.
- Terminates the secondary timer function.
- Removes the user clock from the system by issuing a .RUCLK system call (only if no timer functions are to remain active after this call).

The calling sequence for this non-reentrant subroutine is as follows:

```
        AC1     -   Clock interval specifier
                    0 = one second.
                    1 = one half second.
                    2 = one-quarter second.
AC0     -   Function code:
            0 = Deactivate both primary and secondary timer
                functions.
            1 = Deactivate primary timer function only.
AC2         2 = Deactivate the secondary timer function only.
            Timer parameter list address (TPL).

.TODA
normal return
```

On the normal return AC1 contains 0. If the returned value is non-zero (i.e. the TPL address), the user must start a secondary timer task that issues a .REC call on the Task Message (TLP +0) as the final step in initializing a secondary timer function.

AC2 - TPL

Possible error returns from this subroutine call are the following:

AC0 Error code

1 = No lines timeout enabled.
2 = No clock in system.
3 = Secondary function not properly defined.
4 = User clock already defined.
5 = Address outside address space (mapped systems).

Determines the TCB address by issuing system call .GHRZ.

Defines the following (TPLER) to system as the user-defined clock routine by issuing a .DUCLK call.

0 = initialize both primary and secondary functions.
1 = initialize primary function only.
2 = initialize secondary function only.

```
.TOIN
normal return
```

User-Defined Clock Routine (TIMER)

This is the routine that is called by the system each time the user clock interval expires and a user clock interrupt occurs. TIMER performs both the timing and subroutine calls for both the primary and secondary timer functions.

If the primary function was enabled by means of a .TOIN call, TIMER uses the timer chain to scan both timer word entries in all chained line tables in the system. If a timer word is found to be non-zero, TIMER decrements the value and checks the result for zero, which indicates a timeout condition. When a timeout occurs, the timeout line procedure is called (see next section). If no timeout occurs, the decremented timer value is restored to the timer word in the line table.

If the secondary timer function is enabled, TIMER decrements the secondary function timer word in the TPL and checks for a zero result. When the secondary function timer reaches zero, TIMER issues a .IXMT call with a message address of TPL +0 (the Task Communications Cell) to start the secondary timer task.

When TIMER has completed its activity, it exits by executing a .UCEX task call.

LINE PROCEDURE INTERFACE

If the primary timer function is used, the user must provide a timeout line procedure. The address of this routine is located in the fourth word (LPTLP) of each line procedure table (see the line procedure table description in Chapter 3).

When a line timeout is detected by the TIMER routine, the timeout line procedure is called as follows:

AC0 - mode indicator
0 = receiver timeout.
1 = transmitter timeout.

AC2 - Line table address.

AC3 - Base address of line procedure table.

```
JSR@ LPTLP, 3                ;CALL THE TIMEOUT LINE PROCEDURE
normal return                ;CONTINUE SCAN
terminate scan return
```

On the normal return, AC2 must contain the line table address.

LINE PROCEDURE INTERFACE (Continued)

The terminate scan return is provided to allow the user to limit the number of line timeouts that are processed at each clock interval. By taking this return, the user forces TIMER to stop scanning the timer chain and to proceed immediately to the secondary timer function activity if it is enabled. On the next clock interrupt, the line table scan is resumed from the point of termination.

APPLICATION PROGRAMMING

This section describes the steps that must be taken by the user in writing an application program that correctly implements both the primary and secondary timer functions. Also included is an application example (TIMAP) that shows how the timer routine's calls and data structures might be implemented.

The Communications Timer Package, as described in previous sections, has been designed to be easily incorporated into the user's communications system. Since the timeout modules are part of a library, they will be loaded only once for the entire system, regardless of how many library names appear in the RLDR command line. This satisfies the requirement that there exist only one timeout routine in the system which, in turn, will handle any combination of communications devices with any number of attached lines. To force the inclusion of the timeout modules in a communications save file, the user must ensure that either the initialization or deactivation routine calls, .TOIN or .TODA respectively, is referenced by a .EXTN statement in one of the load modules preceding the library (XXX.LB) module in the RLDR command line.

The TIMIN module is also supplied as a separate relocatable binary file (TIMIN.RB). Therefore, the initialization and deactivation routines could be run as overlays by specifying TIMIN.RB as an overlay in the RLDR command line. In this case, only the TIMER module would be loaded from the library.

The user must first create the timer parameter list (TPL) as previously specified. If the primary timer function is used, the timeout line procedures must be provided with their addresses stored in the fourth word of all appropriate line procedure tables.

The primary timeout routine is called at interrupt level, so it should do as little processing as possible. Thus, major processing should be deferred to a task-level routine, perhaps using the .IXMT/.REC task communications calls. The secondary function is required to run as a user task.

The user may then incorporate .TOIN or .TODA calls into an application program to initialize or deactivate timer functions. Once the primary timer function has been initialized, the user is responsible for setting and clearing the timer words in the line tables.

Even though a line is linked on the timer chain, it does not commence timeout activity until its timer word has been set to a non-zero value. This value is equal to the number of user clock intervals to expire before the line times out, and this value should be set into the timer word immediately following the initialization of a transfer on a line.

Similarly, the user must clear the timer word in the line table as soon as possible following a normal completion of the transfer. This terminates timeout activity on the line and eliminates the possibility of a line timing out after normal completion has occurred.

Included here is an application example in subroutine form, TIMAP, that shows how the primary and secondary timer functions may be activated or deactivated. Also included is an example of a secondary function routine. This particular routine shows how the auto-answer/disconnect functions for all lines in the system could be implemented so that all switched lines in the system would be scanned periodically.

Since TIMAP is supplied in source form to a variety of communications users, minor editing of this source must be performed by each user to tailor the illustrated example to a specific communications device application. Consequently throughout this illustration commented references to entries for each of the appropriate device types will be made; those references which apply to the device being used in an application must have the semicolon deleted, raising the comment to active source code status. The following discussion elaborates upon the TIMAP assembly listing presented as figure D-1. Comments upon this illustration will refer to sections of code by assembly listing page and line number.

The first page of the illustration in figure D-1 contains several distinct elements. Page 1 line 6, 7, 12, and 13 contain external references and external definitions required to run the application. Page 1 lines 8 - 11 contain external references to line table directory pointers for the following respective devices: 4100 (Multiline Asynchronous Controller), 4060/4073 (Communications Multiplexer System), 4015 (Synchronous Communications Controller), and 4029 (Asynchronous Line Adapter). The order of these devices is preserved in the other portions of this page where these devices are referenced in comment groupings.

APPLICATION PROGRAMMING (Continued)

Lines 19 through 25 contain the Timer Parameter List. The first two entries in this list contain the address of the secondary timer function routine and secondary timer word respectively. The next four lines constitute the comment grouping for all four devices. Each device has a line table directory pointer. The list is terminated by a -1. Page 1 lines 27 - 33 contain definitions for INTVL, STCOD, and STCNT. INTVL specifies that the user clock interval is set to one half second. STCOD specifies a full timer initialization, i.e., activation of both the primary and secondary timer functions. STCNT is the value used to reset the secondary timer function word. The value two (2) in this example causes the secondary timer task to be started every second, since the user clock interval (INTVL) was previously specified as one half second.

Any of these three parameters may require changes to reflect the user's individual requirements. These parameters are merely used as input to the .TOIN and .TODA calls in this example.

Page 1 lines 36 - 40 contain a list of auto-answer subroutines used by the secondary function. Note that this comment grouping omits any reference to the 4100 (MAC), since this device provides no auto-answer function. As with the TPL, this list is terminated by a -1.

Page 2 of the illustration contains the two user-level routines which would be called to initiate or stop the system timing functions. The user calling sequences are described on lines 5 - 8 and 30 - 33 respectively. These calling sequences require no input parameters, because the parameters to .TOIN and .TODA are obtained in-line on this page of code. These parameters are as described earlier in this appendix for .TOIN and .TODA. Note that these routines are not reentrant (the return is saved in RETN).

Page 3, the last page of code in this illustration, contains an implementation of a secondary timer task. If the return from the .TOIN call indicates that a secondary task should be started, the code on page 3 lines 7 - 11 is executed to initialize a secondary timer task that has the entry point SECFR (page 3 line 19). The first time that SECFR runs, it issues a .REC call using TPL +0 as the task message address. When the TIMER routine finds that the secondary timer count (TPL +1) has expired, it issues an .IXMT call using TPL +0 as the task message address. As a result, the secondary function task resumes operation following the .REC call (page 3 line 24).

This routine would call any auto-answer routine specified in the secondary function list on page 1 (lines 36 - 40). In this example, each .xxAA call in the list is placed in relative location 55 (line 29) and is then executed. After the successful execution of this routine, the secondary timer function count is reset in the Timer Parameter List using the value STCNT described previously, and the secondary function task issues another .REC call on TPL +0 and awaits another .IXMT call from TIMER.

```
 0001 TIMAP
01
02
                        .TITLE  TIMAP   ;USER DEFINED CMS TIMER APPLICATION
04
05
06                      .EXTN   .TDIN,.TDDA
07                      .EXTN   .TASK,.REC,.KILL
08              ;       .EXTN   LTDP
09                      .EXTN   QLTDP,.CMAA
10              ;       .EXTN   SLTDP,.SCAA
11              ;       .EXTN   ALTDP,.ALAA
12                      .ENT    STRTT,STOPT
13                      .ENT    TPL
14
15                      .NREL
16
17              ;TIMER PARAMETER LIST
18
19 000000'000000 TPL:   0               ;SEC. TIMER FUNCTION COMM CELL
20 000001'000002         2               ;SECONDARY TIMER FUNCTION TIMER WORD
21              ;       LTDP            ;4100 LTD
22 000002'177777        QLTDP           ;4060/4073 LTD
23              ;       SLTDP           ;4015 LTD
24              ;       ALTDP           ;4029 LTD
25 000003'177777        -1              ;EOL
26
27 000004'000001 INTVL:  1               ;TIMER INTERVAL=1/2 SECOND
28
29 000005'000000 STCOD:  0               ;FULL TIMER INITIALIZATION INDICATOR
30
31              ;SECONDARY TIMER FUNCTION PARAMETERS.
32
33 000006'000002 STCNT:  2               ;SECONDARY TIMER COUNT(1 SECOND)
34
35
36              AALST:          ;SECONDARY FCN LIST(AUTO-ANSWER)
37 000007'177777        .CMAA           ;CMS-AUTO-ANSWER FUNCTION
38              ;       .SCAA           ;SCC-AUTO-ANSWER FUNCTION
39              ;       .ALAA           ;ALA-AUTO ANSWER FUNCTION
40 000010'177777        -1              ;EOL
```

Figure D-1 TIMAP Illustration

```
10002 TIMAP
01
02
03                ;SUBROUTINE TO START THE SYSTEM TIMING FUNCTIONS
04                ;                    CALL
05                ;                            JSR@    .STRTT
06                ;                            ERROR RETURN
07                ;                            NORMAL RETURN
08                ;                    .STRTT: STRTT
09
10
11 00011'054415 STRTT:  STA     3,RETN          ;SAVE RETURN
12 00012'020774         LDA     0,STCNT         ;GET SEC TIMER COUNT
13 00013'040756         STA     0,TPL+1         ;SET SEC FUNCTIN COUNT
14 00014'020771         LDA     0,STCOD         ;AC0=FUNCTIN CODE
15 00015'024767         LDA     1,INTVL         ;AC1=TIMER INTERVAL
16 00016'030467         LDA     2,.TPL          ;AC2=SEC. FUNCTION PARAM. TABLE
17 00017'177777         .TOIN                   ;INITIALIZE THE TIMER ROUTINE
18 00020'000404         JMP     STEXT           ;INIT. ERROR AC0=ERROR CODE
19 00021'151004         MOV     1,2,SZR         ;SHOULD SEC TASK BE STARTED
20 00022'000414          JMP    SECTK           ;YES-START TASK
21 00023'010403         ISZ     RETN            ;MAKE FOR NORMAL RETURN
22 00024'002402 STEXT:  JMP     @RETN           ;EXIT
23
24 00025'000000'.TPL:   TPL                     ;POINTER TO TIMER PARAMETER LIST
25 00026'000000 RETN:   0
26
27
28                ;SUBROUTINE TO STOP THE SYSTEM TIMER FUNCTIONS
29                ;                    CALL
30                ;                            JSR@    .STOPT
31                ;
32                ;                            NORMAL RETURN
33                ;                    .STOPT: STOPT
34
35
36 00027'175400 STOPT:  INC     3,3             ;STEP FOR NORMAL RETURN
37 00030'054776         STA     3,RETN          ;SAVE RETURN
38 00031'020754         LDA     0,STCOD         ;AC0=FUNCTION CODE
39 00032'177777         .TODA                   ;DEACTIVATE TIMER FUNCTION
40 00033'024753         LDA     1,STCNT         ;GET SECONDARY TIMER COUNT
41 00034'044745         STA     1,TPL+1         ;RESET TIMER
42 00035'000767         JMP     STEXT           ;EXIT
```

Figure D-1 TIMAP Illustration (Continued)

```
TIMAP
;START THE SECONDARY FUNCTION TASK
;       ON ENTRY:
;          AC2=POINTER TO TPL
;
SECTK:  LDA     1,.SECF         ;GET STARTING ADDR OF TASK
        LDA     0,STTP          ;GET TASK PRIORITY
        .TASK                   ;INITIATE THE SEC TIMER FUNCTION
        JMP     .+1             ;IGNORE ERROR RETURN
        JMP     STEXT-1         ;EXIT


;SECONDARY TIMER FUNCTION TASK
;
;CALLED EVERY 1 SECOND BY THE TIMER ROUTINE
;
;
SECFR:  SUB     0,0
        STA     0,TPL           ;CLEAR TASK COMM CELL
        LDA     0,SFLST         ;SETUP FOR A=A SCAN
        STA     0,SFLTP
        LDA     0,.TPL          ;LOAD ADDRESS OF MESSAGE LOCATIO
        .REC                    ;ISSUE REC FOR SEC TIMEOUT
NXTSF:  LDA@    0,SFLTP         ;ENTRY FROM SEC. FUN. LIST
        COM#    0,0,SNR         ;END OF LIST (=1)
         JMP    ENDSF           ;YES=RETURN
        STA     0,.+1           ;STORE THE SUBROUTINE CALL FOR E
        0                       ; EXECUTE .XXAA CALL FROM HERE
        ISZ     SFLTP
        JMP     NXTSF

ENDSF:  SUB     0,0
        STA     0,TPL           ;CLEAR TASK COMM CELL
        LDA     0,STCNT         ;RESET SECONDARY TIMER WORD
        STA     0,TPL+1         ;WITH TIMER COUNT
        JMP     SECFR           ;REISSUE REC TASK CALL


SRRET:  0                       ;SAVE RETURN
SFLST:  AALST                   ;ADDRESS OF SECONDARY FCN. LIST
SFLTP:  0                       ;CURRENT AALST PTR.
SECF:   SECFR .                 ;POINTER TO SEC FUNCTION ROUTINE
STTP:   2                       ;SECONDARY FUNCTION TASK PRIORITY


        .END
```

Figure D-1 TIMAP Illustration (Continued)

```
00.4 TIMAP

AALST 000007'          1/36    3/41
ENDSP 000006'          3/27    3/33
INTVL 000004'          1/27    2/15
NXTSP 000051'          3/25    3/31
ULTOP 000002' XN       1/09    1/22
RETN  000020'          2/11    2/21    2/22    2/25    2/37
SECFR 000043'          3/19    3/37    3/43
SECTR 000036'          2/20    3/07
SFLST 000060'          3/21    3/41
SFLTP 000007'          3/22    3/25    3/30    3/42
SRRET 000005'          3/40
STCNT 000000'          1/33    2/12    2/40    3/35
STCOU 000005'          1/29    2/14    2/36
STEXT 000024'          2/16    2/22    2/42    3/11
STOPT 000027' EN       1/12    2/36
STRTT 000011' EN       1/12    2/11
STTP  000071'          3/08    3/44
TPL   000000' EN       1/13    1/19    2/13    2/24    2/41    3/20    3/34
                       3/36
.CMAA 000007' XN       1/09    1/37
.KILL 177777  XN       1/07
.REC  000050' XN       1/07    3/24
.SECF 000070'          3/07    3/43
.TASK 000040' XN       1/07    3/09
.TODA 000032' XN       1/06    2/39
.TOIN 000017' XN       1/06    2/17
.TPL  000025'          2/16    2/24    3/23
```

Figure D-1 TIMAP Illustration (Continued)

*****

APPENDIX E

CORE REQUIREMENTS

Program Name	Subroutines	Function	Octal	Decimal
IQMS	.CMIN .CMDA	Initialize/deactivate CMS	262	178
QLPIF	CPENQ CENQ .CMDQ .CMNQ	Line Procedure Interface	76	63
QISR	QISR	Interrupt Service	120	80
QMOD	.CMMS QF1 QF3 QF5 QF7	Modem Subroutines	213	139
QFCN	QF0 QF2 QF4 QF9	CMS Functions	73	59
QRECV	.CMRD	Receive Data	103	67
QSEND	.CMSD	Send Data	136	94
QTTYLP	QIP QOP QTTLP	Teletype line procedure	365	248
QMAA	.CMAA	Auto answer	70	56

*****

APPENDIX F

CMS - 4060/4073 CALLING SEQUENCE AND RETURN SUMMARY

Call	Input	Normal Return	Error Return
.CMIN .CMDA	None	AC1 - Line Table Directory Address AC3 - USP	AC0 - Error Code AC1 - Line Table Directory Address AC3 - USP
.CMSD .CMRD	AC0 - Line Number AC2 - Control Block Address	AC1 - Line Table Address AC3 - USP	AC0 - Error Code AC1 - Line Table Address AC3 - USP
.CMMS	AC0 - Line Number	AC0 - Modem Status Address AC3 - USP	AC0 - Error Code Address AC3 - USP
.CMNQ		word of CS	AC1 - 0 AC3 - USP

APPENDIX F

CMS - 4060/4073 CALLING SEQUENCE AND RETURN SUMMARY

Call	Input	Normal Return	Error Return
.CMIN .CMDA	None	AC1 - Line Table Directory Address AC3 - USP	AC0 - Error Code AC1 - Line Table Directory Address AC3 - USP
.CMSD .CMRD	AC0 - Line Number AC2 - Control Block Address	AC1 - Line Table Address AC3 - USP	AC0 - Error Code AC1 - Line Table Address AC3 - USP
.CMMS	AC0 - Line Number	AC0 - Modem Status AC1 - Line Table Address AC3 - USP	AC0 - Error Code AC1 - Line Table Address AC3 - USP
.CMDQ .CMNQ	None	AC0 - Contents of CBDL word of CB AC1 - Control Block Address AC3 - USP	no errors AC1 - 0 AC3 - USP
.CMAA	None	AC3 - USP	AC3 - USP

*****