1990 9779

Burroughs Corporation

COMPUTER SYSTEMS GROUP
PASADENA PLANT

OFFLOAD READER SORTER DLP

ENGINEERING DESIGN SPECIFICATION

COMPANY CONFIDENTIAL

REVISIONS

REV. LTR	CONTROL DOC & NO	PAGES REVISED, ADDED, DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
A	11/30/84	Initial Issue ECN 49500	L. Simpson T. Chan A. Krantz	R. Haraguchi M. Emm
B	04/22/86	Changed name to Offload Reader Sorter DLP. Corrected errors in result descriptors and updated wording on result headers, per REEs 96960, 97509, 97697, 97705, 97706 and 97708. ECN 50347	L. Simpson M. Jacobs	M. Jacobs M. Emm

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1 INTRODUCTION

This specification describes the Data Comm Data-Link Processor, ORS DLP, which controls information transfer between a host system and one to four data communication lines.

The ORS DLP is compatible with systems conforming to the requirements of SDS 2399 6028 (DCP).

This specification describes functions unique to the Data Comm DLP. Common DLP functions are described in SDS 2323 7381 (DLP subsystem).

The Data Comm Data Link Processor is subsequently referred to simply as ORS DLP or ORS DLP (Low Cost Data Comm).

2 RELATED DOCUMENTS

2323 7381 DLP Subsystem
2323 7399 Message-Level Interface

3 GENERAL DESCRIPTION

The ORS DLP is used to control 1 to 4 low to medium speed data comm lines or a single broadband data comm line of 56K bits per second. The ORS DLP is capable of supporting full or half-duplex lines. The lines may be private or switched in synchronous, asynchronous or bit-oriented modes of transmission.

The ORS DLP is connected to the data communication line via DCE (Data Circuit-terminating Equipment) and ACU (Automatic Calling Unit).

The ORS DLP firmware is composed of several concurrent processes which intercommunicate via shared data structures, mailboxes and message queues. The MLI_HANDLER together with the OP_HANDLER control the low level communication protocol between the ORS DLP and the host.

3 GENERAL DESCRIPTION (Continued)

The MLI_HANDLER also performs all of the ORS DLP I/O operations which are described in the MLI_HANDLER section. The ISC process controls the ISC level of the subsystem which is described in the Connection Protocol section. The EXECUTIVE process controls the DCP level of the subsystem. The DCP Message format section describes the DCP protocol. The EXECUTIVE performs all of the ORS DLP high-level data-communication functions and complies to the DCP protocol.

Each line is governed by an individual LINE_PROCESS which represents the data comm protocol. The LINE_PROCESS communicates with the EXECUTIVE via two message queues and shared data structure. The LINE_PROCESS code and data are loaded as needed (see section 4.6). Up to four LINE_PROCESSes may be running at the same time.

3.1 HARDWARE RESOURCES OF THE ORS DLP

3.1.1 DLI BUFFER

The DLI buffer is a 4K x 17 array used to buffer data between the host and the microprocessor. This static RAM is only word addressable and should not be accessed with odd addresses. This RAM array occupies memory addresses @80000@ - @81FFF@.

3.1.2 DLISEL REGISTER

DLISEL is a pseudo read only register that occupies memory mapped I/O address @88024@. The following chart defines each bit in this register:

DLISEL register (Read only)

a	b	c	d	e	f	g	h	i	j	k	x	l	l	l	l

a : DLP/HDP
A high on this bit indicates the ORS DLP is in DLP mode.

b : CONNECT
A low on this bit indicates that the ORS DLP is connected to the host.

c : LOCAL/
A low on this bit indicates that the ORS DLP is in maintenance mode.

d : AF
Synchronized STIOL.

e : TERM
Backplane terminate signal.

f : UPMESS
Microprocessor generated flag indicating a new DLIOP.

g : STOPBDEL
Indicates end of burst.

h : DLIMESS
DLI state machine flag indicating completion of DLIOP.

i : DLPCON/
Indicates that the ORS DLP is connected to the DLI backplane.

j : STFLAG
Indicates the state of the self test flip-flop.

3.1.2 DLISEL REGISTER (Continued)

k : IOSND/
When this bit is low and the ORS DLP is connected, it will denote that the ORS DLP is driving the DLI data bus. If this bit is high, the ORS DLP receives data from the host.

l : DLISTAT0-DLISTATA (D0-D1)
DLP status is present on the DLI backplane.

3.1.3 ADSEL REGISTER

ADSEL is a pseudo read only register that occupies memory mapped I/O address @8802C@. The following chart defines each bit in this register:

ADSEL register (Read only)

X	X	X	X	X	X	X	X	a	b	c	d	e	f	g	h

a : VPERRP
DLI buss parity error. High true.

b : DLIPARER
DLI microstore parity error. High true.

c : NEQZERO
Low indicates no LPW error. Valid only when all data and LPW words have been loaded into LPW accumulator.

d : PER/
Low indicates a dynamic RAM vertical parity error has occured.

e : LCLCLR
Local clear. High true.

f : MSTRCLR
Master clear. High true.

g : SELCLR
Selective clear. High true.

h : PSSCLR
Selective Master clear. High true.

3.1.4 RBTSEL REGISTER

RBTSEL is a pseudo read only register that occupies memory mapped I/O address @8802A@. The following chart defines each bit in this register:

RBTSEL register (Read only)

a	a	a	a	b	X	X	X	c	c	c	c	c	c	c	c

a : DLISTATnn
The four DLISTAT bits are current ORS DLP status loaded into the DLP Status Register.

b : EMREQ/
This bit, when low, indicates that a DLP is requesting emergency service.

c : BRSTCNTnn
Eight bits of the ORS DLP Burst Counter.

3.1.5 IDSEL REGISTER

IDSEL is a pseudo read only register that occupies memory mapped I/O address @88040@. The following chart defines each bit in this register:

IDSEL register (Read only)

X	X	X	X	X	X	X	X	a	a	a	a	a	a	a	a

a : IDnn
DCDLP jumper selectable type ID.

3.1.6 HPTRLD REGISTER

HPTRLD is a pseudo write only register that occupies memory mapped I/O address @88048@. The following chart defines each bit in this register:

HPTRLD register (Write only)

X	X	X	X	a	a	a	a	a	a	a	a	a	a	a	a

a : HPTRnn
Twelve bits of data that are loaded into the host Pointer Register to address the DLI buffer.

3.1.7 HPTRSEL REGISTER

HPTRSEL is a pseudo write only register that occupies memory mapped I/O address @8804C@. The following chart defines each bit in this register:

HPTRLD register (Read only)

X	X	X	X	a	a	a	a	a	a	a	a	a	a	a	a

a : HPTRnn
Twelve bits of address read from the Host Pointer Register.

3.1.8 BRSTCNTW REGISTER

BRSTCNTW is a pseudo write only register that occupies memory mapped I/O address @88046@. The following chart defines each bit in this register:

BRSTCNTW register (Write only)

X	X	X	X	X	X	X	X	a	a	a	a	a	a	a	a

a : BRSTCNTn
Eight bit value that is loaded into the DLI Burst Counter. Value loaded = Desired value + 1.

3.1.9 STSEL REGISTER

STSEL is a pseudo write only register that occupies memory mapped I/O address @88010@. The following chart defines each bit in this register:

STSEL register (Write only)

X	X	X	X	X	X	a	b	X	X	c	X	X	X	d	e

a : RDSPLY1
A high will extinguish the LED-1.

b : SDSPLY1
A high will turn on the LED-1.

c : STSEL
Output enable for condition select FPLA.DLI

NOTE: This bit must be set low for proper microsequencer operation.

d : LOOPST
This bit is used to enable loop mode test through the DLI Send/Receive registers.

NOTE: This bit must be set low for proper DLI operation.

e : DSPLY2
This bit controls the LED-2.

3.1.10 REQSEL REGISTER

REQSEL is a pseudo write only register that occupies memory mapped I/O address @8801E@. The following chart defines each bit in this register:

REQSEL register (Write only)

X	a	b	c	d	e	f	X	X	X	X	X	X	X	X	X

a : SETREQ
A high will cause the ORS DLP request to go true.

b : SETEMR
A high will set the emergency request.

c : RSTEMR
A high will reset the emergency request.

d : RSTIOSND
A high will make IOSND false.

e : CLRREQ
A high will inactivate ORS DLP request.

f : SETIOSND
IOSND will go true when this bit is set high.

3.1.11 LDDLIREG REGISTER

LDDLIREG is a pseudo write only register that occupies memory mapped I/O address @8801C@. The following chart defines each bit in this register:

LDDLIREG register (Write only)

X	X	X	X	X	X	X	X	X	X	X	X	a	1	b	X

a : VPERRCLR
Clear DLI data path parity error flag. Must be kept high to enable vertical parity checking.

b : CLRBRST
Setting this bit low will clear the Burst flip flop. Must be set high to allow burst to be enabled.

3.1.12 STATWRT REGISTER

STATWRT is a pseudo write only register that occupies memory mapped I/O address @8801A@. The following chart defines each bit in this register:

STATWRT register (Write only)

X	X	X	X	X	X	X	X	X	X	X	X	a	a	a	a

a : STATnn
A write to this register will allow data on the low four bits of the microprocessor data bus to be written into the DLI Status Register.

3.1.13 DLIOPs - DLI OPCODES

The microprocessor has control over only the 4 most significant bits of the DLI state machine. The 6 least significant bits are controlled by the state machine next address and condition logic. The MLI_HANDLER issues DLI operations (DLIOPs), which are adddresses to the DLI state machine, and sets the flag UPMESS which is fed into the DLI state machine condition logic. The DLI state machine clears the flag UPMESS and carries out the operation. The DLIOPs are loaded in the DOPSEL pseudo register, which is in the address @88042@ and has the following format:

DOPSEL Register (Write only)

x	x	x	x	x	x	x	x	D	D	D	D	I	U	R	P

x - dont care

D - DLIOPs bits. These four bits comprise the op select for the DLI state machine.

I - PRLPW. Initializes the LPW accumulator to all ones (FFFF). This bit is low true and should remain high when the LPW accumulator is to be used.

U - UPMESS. This bit, when high indicates to the DLI state machine that a new DLIOP has been loaded into the DOPSEL register.

R - RDLIMESS. This bit when set high will reset the DLI state machine message bit (DLIMESS).

P - RPARER. This high true bit will reset the microstore parity error flag.

3.1.13.1 DLI STATE MACHINE OPCODES (DLIOPs)

The following DLIOPs are avaliable to the user:

@0@ - SEND_STROBE_READ Loads data into the send register and LPW accumulator, increments RAM address and burst counter, sends a DLP strobe and sets DLIMESS.

@1@ - SEND_STROBE_WRITE Receives data from the receive register into RAM and LPW is accumulator, increments RAM address and burst counter, sends a DLP strobe and sets DLIMESS.

@2@ - SEND_STROBE Sends a DLP strobe and sets DLIMESS.

@3@ - SEND_LPW_STROBE Sends LPW via send register, sends a DLP strobe, and sets DLIMESS.

@4@ - LOAD_SEND_REG Loads send register and LPW accumulator from RAM, increments RAM address and burst counters, and sets DLIMESS.

@5@ - MOVE_RECEIVE_REG Loads RAM and LPW accumulator from receive register, increments RAM address and burst counter and sets DLIMESS.

@6@ - MOVE_LPW Loads contents of LPW into RAM, increments RAM address and burst counter and sets DLIMESS.

@7@ - LOAD_LPW Loads LPW accumulator from RAM, increments RAM address and burst counter and sets DLIMESS.

@8@ - WAIT_AF Looks for host strobe and when found sets DLIMESS.

@9@ - WAIT_CONNECT Looks for host connecting and when found sets DLIMESS.

@A@ - NO_OPERATION Idle state.

@B@ - READ_BURST Sets burst flip-flop, loads send register and LPW from RAM, waits for AF, increments RAM address and burst counter, looks for end of burst and at the end of burst sets DLIMESS.

3.1.13.1 DLI STATE MACHINE OPCODES (DLIOPs) (Continued)

@C@ - WRITE_BURST Sets burst flip-flop, loads ram and LPW from receive register, increments RAM address and burst counter, waits for AF, looks for end of burst and at the end of burst sets DLIMESS.

@D@ - READ_HOST_POINTER Enables RAM address counter so that microprocessor can read its contents.

Upon completion of the DLIOP, the state machine sets DLIMESS which provides an interrupt to the microprocessor. After the initial DLIMESS interrupt, the DLIMESS acts as a status flag to the microprocessor.

3.1.14 I/O PORT MAP

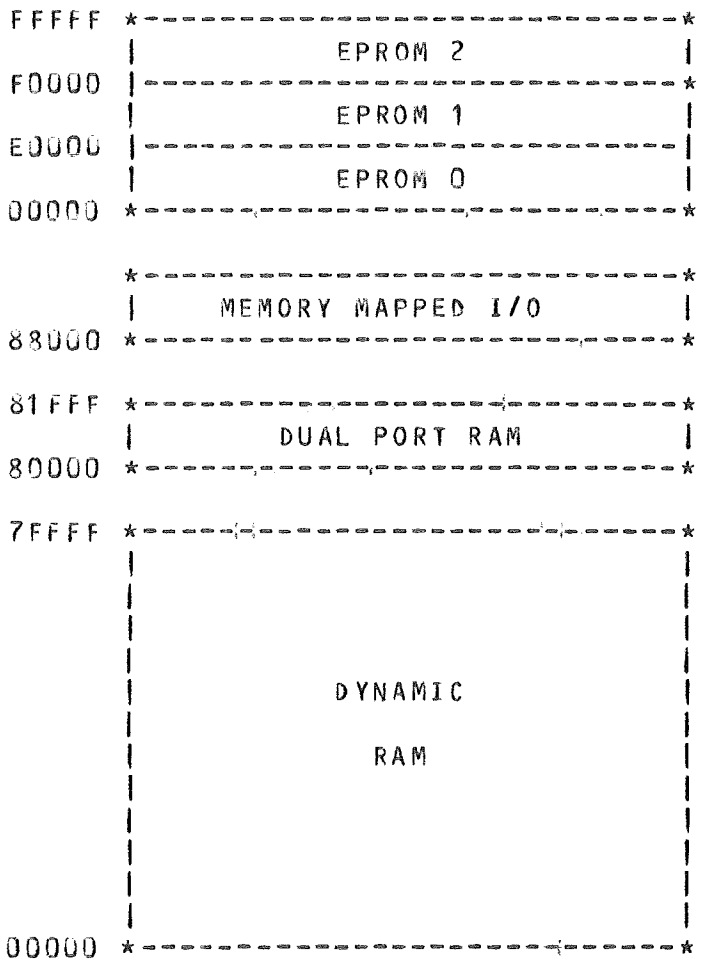
The following table shows all the ports in the ORS DLP subsystem.

I/O Port Map

ADDRESS	DEVICE	CHANNEL	FUNCTION
0040	SCC1	B	Control
0042	SCC1	B	Data
0044	SCC1	A	Control
0046	SCC1	A	Data
0048	SCC2	B	Control
004A	SCC2	B	Data
004C	SCC2	A	Control
004E	SCC2	A	Data
0050	CIO1	C	Data
0052	CIO1	B	Data
0054	CIO1	A	Data
0056	CIO1	-	Control
0058	CIO2	C	Data
005A	CIO2	B	Data
005C	CIO2	A	Data
005E	CIO2	-	Control
0060	PIC	-	A0 = 0
0062	PIC	-	A0 = 1
0068	PIT	0	Data
006A	PIT	1	Data
006C	PIT	2	Data
006E	PIT	-	Control
0070	PERCLR/	-	CLR RAM PER

3.1.15 ORS DLP MEMORY MAP

The following diagram shows the ORS DLP Memory map.



3.1.16 INTERRUPT VECTOR TABLE

The following table shows the configuration of the Interrupt Vector Table in the ORS DLP.

ADDR		INT DEC	INT HEX
000	Divide by zero	# 000	
004	Trace Trap	# 001	
008	NMI Interrupt	# 002	
00C	INTO Interrupt	# 003	
:	:	:	
080	MLI Interrupt	# 032	
084	Tick Interrupt	# 033	
088	Line 0/1 Interrupt	# 034	
08C	Line 2/3 Interrupt	# 035	
090	Line 0/1 Interrupt	# 036	
094	Line 2/3 Interrupt	# 037	
098	Reserved	# 038	
09C	Reserved	# 039	
0A0	VRTX Interrupt	# 040	
:	:	:	
200	VRTX Configuration Table	# 128	
:	:	:	

3.1.16 INTERRUPT VECTOR TABLE (Continued)

0240H	Line 1 Transmit Vector	# 144	90H
0248H	Line 1 Transmit Exception Vector	# 146	92H
0250H	Line 1 Receive Vector	# 148	94H
0258H	Line 1 Receive Exception Vector	# 150	96H
0260H	Line 0 Transmit Vector	# 152	98H
0268H	Line 0 Transmit Exception Vector	# 154	9AH
0270H	Line 0 Receive Vector	# 156	9CH
0278H	Line 0 Receive Exception Vector	# 158	9EH
0280H	Line 3 Transmit Vector	# 160	A0H
0288H	Line 3 Transmit Exception Vector	# 162	A2H
0290H	Line 3 Receive Vector	# 164	A4H
0298H	Line 3 Receive Exception Vector	# 166	A6H
02A0H	Line 2 Transmit Vector	# 168	A8H
02A8H	Line 2 Transmit Exception Vector	# 170	AAH
02B0H	Line 2 Receive Vector	# 172	ACH
02B8H	Line 2 Receive Exception Vector	# 174	AEH

3.1.16 INTERRUPT VECTOR TABLE (Continued)

0340H	Line 1 CIO Vector #0	# 208	D0H
0348H	Line 1 CIO Vector #1	# 210	D2H
0350H	Line 1 CIO Vector #2	# 212	D4H
0358H	Line 1 CIO Vector #3	# 214	D6H
0360H	Line 0 CIO Vector #0	# 216	D8H
0368H	Line 0 CIO Vector #1	# 218	DAH
0370H	Line 0 CIO Vector #2	# 220	DCH
0378H	Line 0 CIO Vector #3	# 222	DEH
0380H	Line 3 CIO Vector #0	# 224	E0H
0388H	Line 3 CIO Vector #1	# 226	E2H
0390H	Line 3 CIO Vector #2	# 228	E4H
0398H	Line 3 CIO Vector #3	# 230	E6H
03A0H	Line 2 CIO Vector #0	# 232	E8H
03A8H	Line 2 CIO Vector #1	# 234	EAH
03B0H	Line 2 CIO Vector #2	# 236	ECH
03B8H	Line 2 CIO Vector #3	# 238	EEH
	:	:	
	:	:	
	:	:	
03FCH		# 255	FFH

3.2 OPERATIONS

OVERVIEW

To initiate an operation, the host sends the ORS DLP an I/O descriptor and a descriptor link. The I/O descriptor specifies the type of operation to be performed as well as various parametric information.

After reception of an I/O descriptor and descriptor link, the ORS DLP will make a transition to one of three MLI states:

Result Desc: This state transition indicates that the ORS DLP is immediately returning a result descriptor for the operation (e.g. because it detected either a vertical or longitudinal parity error in the I/O descriptor or descriptor link).

Disconnect: This state transition indicates that the ORS DLP cannot accept any more operations at this time and that the I/O descriptor and descriptor link were received without parity errors.

Idle: This state transition indicates that the ORS DLP can accept another legal operation at this time and that the I/O descriptor and descriptor link were received without parity errors.

Note: If the ORS DLP detects a vertical parity error on the first MLI word of the I/O descriptor, which contains the "Number of Additional Words" field, it will immediately make a transition to the MLI "I/O Descriptor LPW" state.

Upon termination of an operation, the ORS DLP returns a result descriptor (which contains information indicating the status of the operation including, but not limited to, exception conditions) to the host.

3.2.1 DESCRIPTOR MANAGEMENT

When a host connects to the ORS DLP, the ORS DLP may be found in one of two distinct states, either ready to receive a new descriptor or busy. When STC=3(IDLE), the ORS DLP will cooperate in the transfer of a new I/O descriptor. When STC = 1(DISCONNECT) or STC = 5(RECONNECT), the ORS DLP is busy performing some previously initiated operation.

The ORS DLP is capable of handling only one I/O descriptor. When an I/O descriptor is accepted by ORS DLP, it will continue to accept operations, but the only valid operation at this time is CANCEL.

If a descriptor is accepted by the ORS DLP, and the descriptor does not contain an operation which requires immediate execution, it is stored in the appropriate mailbox and if the operation was a GET-MESSAGE, the ORS DLP makes itself available to receive another descriptor (CANCEL descriptor is the only one it can accept).

If the ORS DLP detects an MLI vertical parity error, an MLI longitudinal parity error, or invalid information in a descriptor sent by the host, the ORS DLP will go immediately to the result status and will return a result descriptor which indicates the nature of the error. Following the return of the result and its LPW, the ORS DLP is again ready to accept new descriptors.

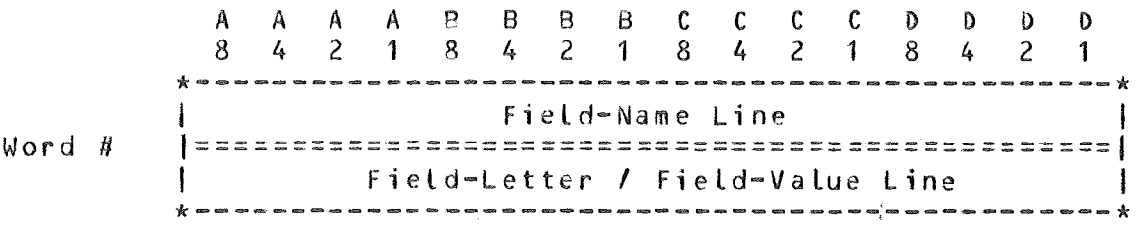
Following the result, the ORS DLP makes itself free to accept new descriptors.

When an operation is received by the ORS DLP which requires immediate execution (e.g., Test/ID), the ORS DLP will not disconnect but will return a result descriptor. In these special cases, an operation will not be stored and will be considered complete upon sending a result descriptor. After this, the ORS DLP will become available for new I/O descriptors.

If the ORS DLP finds a descriptor that needs attention, the ORS DLP checks to see if the host is still disconnected. If so, the ORS DLP goes busy by setting STC=1(DISCONNECT). The ORS DLP will be busy and will not be available for new descriptors until the current descriptor has completed execution.

3.2.2 I/O DESCRIPTOR - RESULT DESCRIPTOR NOTATION

I/O descriptors and result descriptors are data structures which contain multiple component fields. These fields are mapped into a sequence of MLI words. An MLI word is 16 bits wide. The mapping between fields and MLI words is shown below.



A particular field is defined by the following notation:

w: "Field Name": Word (x) [y:z]

w: The letter "w" is used to cross-reference to the verbal description of the field with its position in the associated diagram. Subfields of fields are not assigned individual field letters.

Field Name: The name assigned to the field.

Word (x): The field is contained within (or starts in) MLI word number "x".

[y:z]: The field starts at the bit labeled "y" and is "z" bits wide.

Portions of MLI words which are marked as either "(Not Used)" or "(NU)" must be zero.

Integer fields are binary with the left-most bit (in the sense of the diagrams in this document) being the most significant bit.

In the lists of values for encoded fields, any value which is not explicitly listed is invalid.

3.2.3 COMMON I/O-DESCRIPTOR FIELDS

Except for a few common fields, the I/O-descriptor fields are different for the individual operation types. The meaning of common fields is discussed in this section, while the actual I/O-descriptor definitions are listed in section 3.2.5 (Operation Summary) and section 3.2.8 (Individual Operation Types).

Common I/O-descriptor field definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Operation				Sub Op				Reserved							
	a	a	a	a	b	b	b	b	0	0	0	0	0	0	0	0
Word: 2	Length															
	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c

a: Operation: Word (1) [A8:4]
This enumeration specifies the nature of the data transfer which may occur as part of the operation. The actual data-transfer values which can occur for each operation type are listed in the sections dealing with the individual operation types. The encoding of this field is listed in the table below.

Data Transfer	Code
Read: ORS DLP --> Host	1000
Write: Host --> ORS DLP	0100
Test: None	0010

b: Sub-Operation: Word (1) [B8:4]
This hex field specifies the sub-operation to be performed. The actual sub-operation values which occur for each operation are listed in the section dealing with the individual operation types.

3.2.3 COMMON I/O-DESCRIPTOR FIELDS (Continued)

c: Data Length: Word (2) [A8:16]
This integer specifies either the maximum or the required number of bytes of data transfer for the operation. It does not apply to all operation types.

3.2.4 COMMON RESULT-DESCRIPTOR FIELDS

The meaning of common fields is discussed in this section, while the actual result-descriptor definitions are listed in section 3.2.5 (Operation Summary) and section 3.2.8 (Individual Operation Types).

Common result-descriptor field definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Common Exceptions															
	-	DE	VP	LP	ET	-	-	IF	IS	-	DA	AE	-	-	LM	EX
Word: 2	Common Exceptions															
	WD	-	-	-	-	-	LG	AM	-	-	AC	-	-	-	-	-

DE - Descriptor Error: Word (1) [A4:1]

A TRUE value of this boolean indicates that the ORS DLP detected an error on the I/O descriptor link. This exception is reported with at least one other exception to indicate type of error.

3.2.4 COMMON RESULT-DESCRIPTOR FIELDS (Continued)

VP - MLI Vertical Parity Error: Word (1) [A2:1]

A TRUE value of this boolean indicates that the ORS DLP detected a vertical parity error on an I/O descriptor, a descriptor link, or data sent from the host to the ORS DLP. If bad parity is detected on either an I/O descriptor or a descriptor link, DE will be indicated. If bad parity is detected on data transfer, DA will be indicated.

LP - MLI Longitudinal Parity Error (LPE): Word (1) [A1:1]

A TRUE value of this boolean indicates that the ORS DLP detected a longitudinal parity error on an I/O descriptor, a descriptor link, or data sent from the host to the ORS DLP. DE/DA will be set accordingly.

ET - Early Termination: Word (1) [B8:1]

A TRUE value of this boolean indicates that before the total number of data bytes required (generated) by the operation had been transferred from (to) the host, the host terminated data transfer to (from) the ORS DLP. This exception is reported in combination with DA.

IF - Invalid Information: Word (1) [B1:1]

A TRUE value of this boolean indicates that the ORS DLP detected invalid information within one of the fields of the I/O descriptor. If the error was detected in the I/O descriptor, DE will also be set. DA will be set if incorrect ISC header is detected.

IS - Incorrect State: Word (1) [C8:1]

A TRUE value of this boolean indicates that a vald Op was received but the Op is not valid at this time because of the state of the ORS DLP. This condition will also cause a DE result to occur.

3.2.4 COMMON RESULT-DESCRIPTOR FIELDs (Continued)

DA - MLI Data Error:: Word (1) [C2:1]

A TRUE value of this boolean indicates that an error was detected in the data.

AE - Access Error: Word (1) [C1:1]

A TRUE value of this boolean indicates that the operation could not be completed successfully because of the incorrect sequence of a ISC headers.

LM - ORS DLP Malfunction: Word (1) [D2:1]

A TRUE value of this boolean indicates that the ORS DLP has detected a firmware/hardware malfunction and unable to complete the operation successfully.

EX - Exception: Word (1) [D1:1]

A TRUE value of this boolean indicates that an exception occurred during the operation. At least one exception condition will be set.

WD - Write Denied: Word (2) [A8:1]

This bit is set when ORS DLP denies a SEND-MESSAGE operation.

LG - MLI Length Error: Word (2) [B2:1]

The number of the words transmitted by the host to the ORS DLP is greater than the value received in the length field.

3.2.4 COMMON RESULT-DESCRIPTOR FIELDs (Continued)

AM - AMR Error: Word (2) [B1:1]

This bit is set when ORS DLP denies a SEND-MESSAGE operation.

AC - AMR Cleared: Word (2) [C2:1]

This bit is set a new value loaded into the AMR register.

3.2.5 OPERATION SUMMARY

The operation types of the ORS DLP are summarized below followed by tables which summarize the ORS DLP I/O and result descriptors. The operations are summarized in op-type order.

Send Message:

An operation of this type is used both to transfer a message from the host to the ORS DLP and to acknowledge messages which were previously sent from the ORS DLP to the host. It is discussed in section SEND MESSAGE.

Send AMR:

An operation of this type is used both to transfer an AMR value from the host to the ORS DLP.

Get Message:

An operation of this type is used to transfer a message from the ORS DLP to the host. It is discussed in section GET MESSAGE.

3.2.5 OPERATION SUMMARY (Continued)

Get AMR:

An operation of this type is used to transfer an AMR value from the ORS DLP to the host. It is discussed in section GET AMR.

Get Status:

An operation of this type is used to transfer a status from the ORS DLP to the host. It is discussed in section GET STATUS.

Cancel:

An operation of this type is used to force the conditional termination of the "Get Message" operation which is currently in progress. It is discussed in section CANCEL.

Test ID:

An operation of this type is used to determine the type and configuration ID of the ORS DLP. It is discussed in section TEST ID.

Self Test:

An operation of this type is used to initiate the Self Test on the ORS DLP. See Self Test Section for more information.

Self Test Status:

An operation of this type is used to transfer a Self Test Status of the ORS DLP to the host. For more information, refer to Self Test Status Section.

3.2.6 I/O DESCRIPTOR SUMMARY

I/O Descriptors	Word 1				Word 2			
Operation Type	A	B	C	D	A	B	C	D
Send Message	4	0	0	0	l	l	l	l
Send AMR	4	1	0	0	-	-	-	-
Get Message	8	2	0	0	-	-	-	-
Get AMR	8	1	0	0	-	-	-	-
Get Status	8	E	0	0	-	-	-	-
Cancel	2	8	0	0	-	-	-	-
Test ID	2	C	0	0	-	-	-	-
Self Test	2	2	0	0	-	-	-	-
Self Test Status	2	4	0	0	-	-	-	-

0-F = The corresponding hexadecimal digit.
- = The word does not apply to the op type.
l = Length in decimal.

3.2.7 RESULT DESCRIPTORS SUMMARY

Result Descriptors	Word 1: Common Exceptions																
Operation Type	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1	Sum
Send Message	0	x	x	x	x	0	0	x	x	0	x	x	0	0	x	x	2
Send AMR	0	x	x	x	x	0	0	x	x	0	x	x	0	0	x	x	2
Get Message	0	x	x	x	x	0	0	x	x	0	x	x	0	0	x	x	2
Get AMR	0	x	x	x	x	0	0	x	x	0	x	x	0	0	x	x	2
Get Status	0	x	x	x	x	0	0	x	x	0	0	0	0	0	x	x	2
Cancel	0	x	x	x	0	0	0	x	x	0	0	0	0	0	x	x	2
Test ID	0	x	x	x	0	0	0	x	0	0	0	0	0	0	x	x	2
Self Test	0	x	x	x	0	0	0	x	x	0	0	0	0	0	0	x	2
Self Test Status	0	x	x	x	0	0	0	x	0	0	0	0	0	0	0	x	2

Sum = The total number of result-descriptor words.
x = Either a 0 or a 1.

3.2.7 RESULT DESCRIPTORS SUMMARY (Continued)

Result Descriptors	Word 2: Common Exceptions																
Operation Type	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1	Sum
Send Message	x	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	2
Send AMR	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	2
Get Message	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Get AMR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Get Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Cancel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Test ID	0	1	0	0	1	1	0	1	x	x	x	x	x	x	x	x	2
Self Test	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Self Test Status	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	2

x = Either a 0 or a 1.

3.2.8 INDIVIDUAL OPERATION TYPES

3.2.8.1 SEND MESSAGE

A "Send Message" operation is used to transfer a message from the host to the ORS DLP.

Acceptance of a "Send Message" operation causes the ORS DLP to allocate a buffer. If the ORS DLP cannot accept the message, WD and AM exceptions will be returned.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Word: 2	Length															
	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

n: Data Length: Word (2) [A8:16]
This BCD value represents the number of bytes in the message and can range in value from one to 9,999. All zeros or any value greater than 9 in any BCD digit is invalid and will return DE, IF, and LG exceptions. If the actual message length is different from the specified in the Data Length field, a LG length exception will be returned.

Note: The minimum A to B address difference should not be less than 18 bytes. This is a software requirement.

3.2.8.1 SEND MESSAGE (Continued)

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	ET	0	0	IF	IS	0	DA	AE	0	0	LM	EX
Word: 2	Exceptions															
	WD	0	0	0	0	0	LG	AM	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
ET	Early Termination	Word(1) [B8:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
DA	MLI Data Error	Word(1) [C2:1]
AE	Access Error	Word(1) [C1:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]
LG	MLI Length Error	Word(2) [B2:1]
WD	Write Denied	Word(2) [A8:1]
AM	AMR Error	Word(2) [B1:1]

3.2.8.2 SEND AMR

A "Send AMR" operation is used to transfer an AMR value from the host to the ORS DLP. Acceptance of a "Send AMR" operation causes the ORS DLP to load the Third Access Mask pseudo register with 3 word value.

I/O-descriptor definition:

	A8	A4	A2	A1	B8	B4	B2	B1	C8	C4	C2	C1	D8	D4	D2	D1
	Op Type				Sub op											
Word: 1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Result-descriptor definition:

	A8	A4	A2	A1	B8	B4	B2	B1	C8	C4	C2	C1	D8	D4	D2	D1
	Exceptions															
Word: 1	0	DE	VP	LP	ET	0	0	IF	IS	0	DA	AE	0	0	LM	EX
	Exceptions															
Word: 2	0	0	0	0	0	0	0	0	0	0	AC	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
ET	Early Termination	Word(1) [B8:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
DA	MLI Data Error	Word(1) [C2:1]
AE	Access Error	Word(1) [C1:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]
AC	AMR Cleared	Word(2) [C2:1]

3.2.8.3 GET MESSAGE

A "Get Message" operation is used to transfer a message from the ORS DLP to the host.

Acceptance of a "Get Message" operation causes the ORS DLP to set flag indicating GET MESSAGE available. This is a "No Time_Out" operation which can be terminated by either send data to the host or execution of a cancel operation.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

3.2.8.3 GET MESSAGE (Continued)

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	ET	0	0	IF	IS	0	DA	AE	0	0	LM	EX
Word: 2	Exceptions															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
ET	Early Termination	Word(1) [B8:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
DA	MLI Data Error	Word(1) [C2:1]
AE	Access Error	Word(1) [C1:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]

3.2.8.4 GET AMR

A "Get AMR" operation is used to transfer all 17 Access Mode pseudo registers to the host, total of 51 words.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
	Op Type				Sub op											
Word: 1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
	Exceptions															
Word: 1	0	DE	VP	LP	ET	0	0	IF	IS	0	DA	AE	0	0	LM	EX
	Exceptions															
Word: 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
ET	Early Termination	Word(1) [B8:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
DA	MLI Data Error	Word(1) [C2:1]
AE	Access Error	Word(1) [C1:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]

3.2.8.5 GET STATUS

A "Get Status" operation is used to transfer 3 word Firmware Status to the host. The first word contains the Self Test result value (Good Result=AAAA). The second word contains the ORS DLP configuration; each digit represents a line configuration. The third word contains firmware error information. Upon successful completion of this operation, the firmware error (third word) will be cleared.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	ET	0	0	IF	IS	0	0	0	0	0	LM	EX
Word: 2	Exceptions															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
ET	Early Termination	Word(1) [B8:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]

3.2.8.6 CANCEL

A "Cancel" operation is used to force the conditional termination of the "Get Message" operation which is currently in progress. If the "Cancel" operation is unable to cancel any operation, the "Incorrect state", "Descriptor Error" and "Exception" are reported in "Cancel" operation result descriptor, and the ORS DLP must not disconnect from the host before the result descriptor has been sent. The "Cancel" operation does not return a result descriptor if the operation is successful.

See section GET MESSAGE for a discussion of the effect of a "Cancel" on "Get Message" operation which is in progress.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	0	0	1	0	1	0	0	0	0	0	0	u	0	u	0	0

3.2.8.6 CANCEL (Continued)

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	0	0	0	IF	IS	0	0	0	0	0	LM	EX
Word: 2	Exceptions															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
IF	Invalid Information	Word(1) [B1:1]
IS	Incorrect State	Word(1) [C8:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]

3.2.8.7 TEST ID

A "Test ID" operation is used to determine the type and configuration ID of the ORS DLP. This operation type is common to all DLPs.

Upon receipt of an I/O descriptor for an operation of this type, the ORS DLP will not make a transition to the "Idle" MLI state before it has returned a result descriptor for the operation.

Acceptance of a "Test ID" operation simply causes the ORS DLP to return a result descriptor containing the desired information.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
	Op Type				Sub op											
Word: 1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0

3.2.8.7 TEST ID (Continued)

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	0	0	0	IF	0	0	0	0	0	0	LM	EX
Word: 2	Exceptions															
	0	1	0	0	1	1	0	1	c	c	c	c	c	c	c	c

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
IF	Invalid Information	Word(1) [B1:1]
LM	ORS DLP Malfunction	Word(1) [D2:1]
EX	Exception	Word(1) [D1:1]
c	Field installed Jumpers	Word(2) [C8:8]

3.2.8.8 SELF TEST

Receipt of this descriptor initiates the Self Test on a ORS DLP providing the ORS DLP is in the proper state. This operation causes the ORS DLP to return an immediate result descriptor.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

3.12.8.8 SELF TEST (Continued)

If a Self Test I/O descriptor is sent to a ORS DLP, the ORS DLP immediately returns a two word result descriptor.

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Exceptions															
	0	DE	VP	LP	0	0	0	IF	IS	0	0	0	0	0	0	EX
Word: 2	Exceptions															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE	Descriptor Error	Word(1)	[A4:1]
VP	MLI Vertical Error	Word(1)	[A2:1]
LP	MLI Longitudinal Error	Word(1)	[A1:1]
IF	Invalid Information	Word(1)	[B1:1]
IS	Incorrect State	Word(1)	[C8:1]
EX	Exception	Word(1)	[D1:1]

3.2.8.9 SELF TEST STATUS

When the ORS DLP returns to IDLE (status 3), the Self Test Status I/O Descriptor is sent to the ORS DLP. The ORS DLP returns an immediate two word result descriptor.

I/O-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
Word: 1	Op Type				Sub op											
	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

3.2.8.9 SELF TEST STATUS (Continued)

The Self Test Status operation is used to retrieve immediate information on the result of the Self Test. That information may have been generated by any of the means to initiate Self Test which are specified. See Self Test Section.

Result-descriptor definition:

	A 8	A 4	A 2	A 1	B 8	B 4	B 2	B 1	C 8	C 4	C 2	C 1	D 8	D 4	D 2	D 1
	Exceptions															
Word: 1	0	DE	VP	LP	0	0	0	IF	0	0	0	0	0	0	0	EX
	Exceptions															
Word: 2	0	0	0	0	0	0	0	0	p	b	b	b	p	b	b	b

This field contains the same information specified in Section 5, Self Test result reporting.

DE	Descriptor Error	Word(1) [A4:1]
VP	MLI Vertical Error	Word(1) [A2:1]
LP	MLI Longitudinal Error	Word(1) [A1:1]
IF	Invalid Information	Word(1) [B1:1]
EX	Exception	Word(1) [D1:1]
b =	Board Failure	Word(2) [D8:4]
p =	Parity Error	

3.3 CONNECTION PROTOCOL

The protocol used over the ISC connection between the host and the ORS DLP is a variation of the BNA ISC Station group protocol. The main difference between the two is due to the assumption that the ISC channel has a high reliability factor. In order to be MCS and MCP compatible with the B974, the same connection protocol will be implemented between the ORS DLP and the host.

3.3.1 FRAMES

The minimum amount of information passed across the ISC channel is called a frame. Each frame has a control part and some frames have a text part. The following tables list all the valid frames and their control portion for each system.

3.3.1 FRAMES (Continued)

FRAMES SENT BY THE HOST

FRAME	FRAME TYPE	ACK REQUIRED	ACK	DESTINATION	N(R)	N(S)
INFO	0001	1	0/1	00/01	N(R)	0000
RR	0010	0	1	00	N(R)	0000
LR	0011	0	0	00	0000	0000
DISC	0101	0	0	00	0000	0000
TEST	0110	0/1	0/1	00	0000	0000

FRAMES SENT BY THE DC_DLP

FRAME	FRAME TYPE	ACK REQUIRED	ACK	DESTINATION	N(R)	N(S)
INFO	0001	1	0	00/01	0000	N(S)
LR	0011	0	0	00	0000	0000
DISC	0101	0	0	00	0000	0000
TEST	0110	0/1	0/1	00	0000	0000

The fields within the control portion of the frame have the following meanings:

FRAME TYPE

This field distinguishes between the different frames. The different values of this field are: INFORMATION, RECEIVER READY, LINK RESET, DISCONNECT, and TEST. These frames are described later in this document.

3.3.1 FRAMES (Continued)

ACKED-REQUIRED BIT

This bit is used to indicate whether this frame is requesting an acknowledgement (bit = 1) or not (bit = 0). However, on some frames it has a slightly different meaning. It indicates whether additional frames will be sent (bit = 0) or not (bit = 1). In other words, if the bit is set, then the sending side will be turning the connection around. If the bit is reset, then additional frames will be sent. This bit is also known as the A/R bit.

ACK BIT

This bit is used to indicate whether this frame is a specific acknowledgement (bit = 1) or not (bit = 0). If a frame was received with the ACK-REQUIRED bit = 1, an acknowledgement must follow.

DESTINATION BITS

This field is used only on INFORMATION frames and may be one of two values (00 or 01). It is used to indicate whether the information is initialization type data (value = 00) or station type data (value = 01).

N(R)

N(R), RECEIVE SEQUENCE NUMBER, indicates the sequence number of the next expected INFORMATION frame. This field is only valid on frames sent from the host. It implicitly acknowledges all INFORMATION frames numbered up to but not including N(R). N(R) ranges from 0 to 7. It is only valid on RECEIVER READY and INFORMATION frames.

N(S)

N(S), SEND SEQUENCE NUMBER, indicates the sequence number of the INFORMATION frame. It is valid only on INFORMATION frames sent from the ORS DLP. N(S) ranges from 0 to 7. N(S) for INFORMATION frame sent from the host always equals 0.

3.3.1.1 INFORMATION FRAME

The INFORMATION frame (I frame) is used to efficiently transfer data across a link. The information field of the I frame starts in byte 3 and may be any multiple of bytes long. Data in the information field is unrestricted with respect to code (it is the responsibility of the two end-points to decide). The length of the information field is not needed because it can be determined from the length specified in the IO descriptor. An information field length of zero is permitted, however it is useless.

The maximum amount of data of one opcode is 8192 bytes. Since the ISC contol field is two bytes, the maximum I frame data length is 8190 bytes.

An I frame, sent by host, is acknowledged by a good I/O descriptor. Thus, the window size is one and the need for send numbers is eliminated. This also eliminates the need for the ORS DLP to send receive numbers or RR frames to acknowledge received frames.

These differences in the host and the ORS DLP protocol cause the ACK-REQUIRED and the ACK bits to take on different meanings. In all the I frames originated by the ORS DLP, the ACK bit is always reset. If the ORS DLP set the ACK-REQUIRED bit to turn on an INFORMATION frame, the host must respond with an ACK, otherwise the ORS DLP will issue a Link-Reset frame to the host.

All possiole combinations of the ACK-REQUIRED and ACK bits for an I frame and their meanings for each side are described in the following charts:

FRAMES SENT BY THE HOST

Ack Req	Ack	Meaning
0	0	INVALID
0	1	INVALID
1	0	A frame being sent and not responding to an ack request.
1	1	A frame being sent and responding to an ack request.

3.3.1.1 INFORMATION FRAME (Continued)

FRAMES SENT BY THE ORS DLP

Ack Req	Ack	Meaning
0	0	INVALID
0	1	INVALID
1	0	Requesting an ACK.
1	1	INVALID

The following figure represents the structure of the I frame.

I FRAME

CONTROL FIELD			
0 0 0 1 R A 0 0	N(R)	N(S)	INFO

BYTE 1 2 3
Bit 8 4 2 1 8 4 2 1 8 4 2 1 8 4 2 1

3.3.1.2 RECEIVER READY

The RECEIVER READY (RR) frame is only sent by the host and it is used to respond to an acknowledgement whenever no data is to be sent. All valid combinations of the ACK-REQUIRED and ACK bits for the RR frame are shown in the chart below.

Ack Req	Ack	Meaning
0	0	INVALID.
0	1	Not requesting an ack but responding to an ack request.
1	0	INVALID.
1	1	INVALID.

The following figure represents the structure of the RR frame.

RR FRAME

0 0 1 0 0 1 0 0	N(R)	0 0 0 0

CONTROL FIELD

BYTE	1								2							
Bit	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

3.3.1.3 LINK RESET

The LINK RESET (LR) frame is used to reset the logical link. LR always has the ACK-REQUIRED bit = 0 and the ACK bit = 0. Upon acceptance of this frame the sequence number N(S) and N(R) are set to zero. The receiving side does NOT send a response to the LR frame. All previously unacknowledged I frames remain unacknowledged.

Note: The length of two or more bytes is accepted. However, if the length is two bytes, the third byte is assumed to be zero.

The following figure represents the structure of the LR frame.

LR FRAME

0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 n n 0 0 0 0
CONTROL FIELD		
BYTE 1	2	
Bit 8 4 2 1 8 4 2 1	8 4 2 1 8 4 2 1	

x - Reserved

nn = 00 - No action

nn = 01 - All MCS's status change to "Open and Blocked"

nn = 10 - All MCS's status change to "Closed"

nn = 11 - Invalid

3.3.1.4 DISCONNECT

The DISCONNECT (DISC) frame is used to perform a logical disconnect; i.e., inform the receiving side that the transmitting side is suspending operation. The DISC frame always has the ACK-REQUIRED bit = 0 and the ACK bit = 0. The receiving side accepts the DISC and ceases all transmissions. All previously unacknowledged I frames remain unacknowledged. The sending side must issue a Link_Reset frame to resume transmission.

The following figure represents the structure of the DISC frame.

DISC FRAME

0 1 0 1 0 0 0 0	0 0 0 0 0 0 0 0
CONTROL FIELD	

BYTE	1								2							
Bit	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

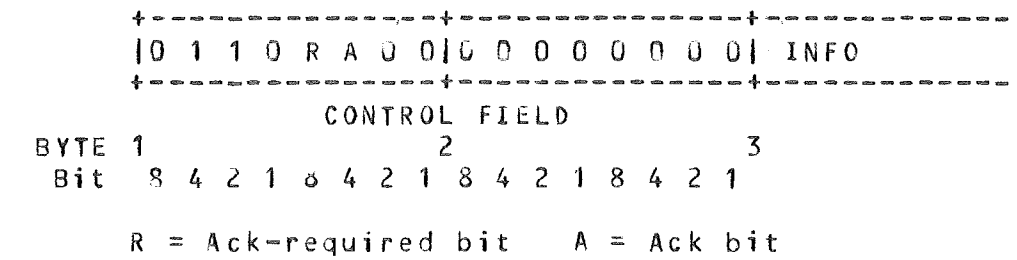
3.3.1.5 TEST

The TEST frame is used to determine the status of the link. The system initiating the TEST will have the ACK-REQUIRED bit = 1 and the ACK bit = 0. The system receiving the TEST will send a TEST with the ACK-REQUIRED bit = 0 and the ACK bit = 1. All other combinations of the ACK-REQUIRED and ACK bits are invalid.

An information field is optional with the TEST frame and if present, the information field will be returned (echoed) by the receiver with a TEST frame. The sending side considers the test terminated upon the receipt of the TEST response or by exceeding the retry limit. The TEST frame has no effect on the mode or the sequence counts, N(R) and N(S), maintained by either side.

The following figure represents the structure of the TEST frame.

TEST FRAME



R = Ack-required bit A = Ack bit

3.4 DCP MESSAGE FORMAT

This section describes the interface between the host and the ORS DLP subsystem. All messages to be written by the host to the ORS DLP are preceded by an 80-digit MESSAGE HEADER. Some special DCP messages consist of a header only. RESULT HEADERS are formatted by the ORS DLP and returned to the host.

The message header is an 80-digit information block used to provide communication control. The result header is used to report conditions concerning physical and logical entities within the system. Result headers are not reported for every function. (The Result headers are described in section 3.5.1).

The message header occurs at the beginning of the text field, if there is one, and it is used to carry control information. The 80-digits of the header (both message headers and result headers) are divided into 20 words. The header words are further subdivided into fields using software bit numbering scheme. Message header layout is defined in the following table:

Word	00	FFFFFFFF........
	01	LLLLLLLLLLLLLLLL
	02	VVVVVVVVVVVVVVVV
	03	TTTTTTTTTTTTTTTT
	04	....
	05	....
	06	........RRRRRRRR
	07	xxxxxxxxxxxxxxxx
	08	xxxxxxxxyyyyyyyy
	09	..iq............
	10	....
	11	NNNNNNNNNNNNNNNN
	12	SSSSSSSSSSSSSSSS
	13	SSSSSSSSSSSSSSSS
	14	....
	15	....
	16	....
	17	....
	18	....
	19	....

3.4 DCP MESSAGE FORMAT (Continued)

F - Communication function. The requested/returned code indicates action to-be-taken/completed.

L - Local Station Number (LSN). A hexadecimal number from @0000@ - @00FF@ is used to specify a station in the network.

V - Function Variants. Describes variation of the original function.

T - Text Length in bytes. Defines the actual size of the message text in hexadecimal.

R - Retry Count.

x - User Station Tallies.

y - User Station Toggles (8 boolean values).

i - Immediate bit

q - Queue bit (top=1, bottom = 0).

N - Message Number. A four-digit number used to ensure that no messages are lost in or out of the system.

S - Sequence number. This field may be used by programs for sequencing information.

3.4.1 DCP MESSAGES

3.4.1.1 WRITE (FUNCTION 30)

The Write function allows output to be sent to the station indicated by the inserted Logical Station Number (LSN). Carriage control is performed as specified by the variant field (Word 2) bits if it is implemented in the Line Process. If the specified text length is zero, only the carriage control is accomplished. If the text length is greater than zero, the text, along with the carriage control, is passed to the station. A result header is returned.

WRITE HEADER FORMAT

Header (0)	[1:8]	= Function No. (30)
Header (1)	[1:16]	= Logical Station No
Header (2)	[1:8]	= SKIPCONTROL byte
	[9:1]	= 1 SKIP
	[10:1]	= 1 SPACE
	[11:1]	= 1 PAPER MOTION
	[12:1]	= 1 TAB
	[13:1]	= 1 LINEFEED
	[14:1]	= 1 CARRIAGE RETURN
	[15:1]	= 1 PAGE
	[16:1]	= 1 BLOCK
Header (3)	[1:16]	= Character Count (Bytes)
Header (6)	[9:8]	= Retry Value
Header (7)	[1:8]	= Tally (0)
Header (7)	[9:8]	= Tally (1)
Header (8)	[1:8]	= Tally (2)
Header (8)	[9:8]	= Toggles (7-0)
Header (9)	[4:1]	= 0 = Bottom queue 1 = Top queue
Header (11)	[1:16]	= Message number

3.4.1.1 WRITE (FUNCTION 30) (Continued)

NOTE:

This entire 16-bit field (i.e., header word (2) is an additional field that can be passed to the Line Process to be used for special devices.

A MCS may elect to assign a Message Number (Header (11) [1:16] to all messages that it creates for output. This provides some ability to audit the flow of messages and aids in any system recovery procedure. The systems software preserves the integrity of this field and does not use it for other purposes.

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. An ACK is received from the station.

10,01 Error retry count expired. Station made not ready.

10,02 A BREAK was detected during transmission with no provision in the Line Process to handle BREAK. Station made not ready.

10,05 The Line Process did not detect CTS within 7 seconds of raising RTS. Line made not ready.

10,06 The Line Process detected loss of DSR while transmitting. Line made not ready.

10,17 The TERMINAL TYPE specified in the station table cannot process output messages.

3.4.1.2 ENABLEINPUT (FUNCTION 32)

The Enableinput function causes the STATION.INPUT_ENABLED to be set. The Enableinput function (32) affects the readiness of a station to receive input.

If both STATION.ENABLED and the STATION.INPUT_ENABLED are set, the Line Process will start accepting incoming messages from the station. Consequent result messages are attached to the Dispatcher Result Queue for return to the host system. Error Results are dealt with in the same way. The EXECUTIVE module only sets the STATION.INPUT_ENABLED boolean variable in the station table.

The Line Process may use STATION.FREQUENCY to affect the rate at which a station is polled.

ENABLEINPUT HEADER FORMAT

Header (0) [1:8]	= Function No. 32.
Header (1) [1:16]	= Logical Station Number.
Header (2) [1:1]	= If true, take the two-digit poll frequency value specified in Header (2) [9:8] and store it in the specified stations table in place of the FREQUENCY. A poll frequency of "00" represents the highest rate.

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. INPUT_ENABLED is set.

10,17 The TERMINAL TYPE specified in the Station table cannot process output messages.

3.4.1.3 DISABLEINPUT (FUNCTION 33)

The Disableinput function is the opposite of Enableinput. Following a Disableinput request, polling ceases if the station is a polled station. A result header is returned.

DISABLEINPUT HEADER FORMAT

```
Header (0) [1:8]     = Function No. 33
Header (1) [1:16]    = Logical Station Number
Header (9) [4:1]     = 0 = Bottom queue
                       1 = Top queue
```

If the function is top queued, no more inputs are accepted from the specified station. If the function is bottom queued, inputs are still accepted from the station until all queued output requests are completed and Disableinput is performed.

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. The Station is now not ENABLED.

3.4.1.4 MAKE STATION READY (FUNCTION 34)

The make station ready function supplies the MCS with the ability to control the readiness of a station. No result header is returned.

MAKE STATION READY HEADER FORMAT

Header (0) [1:8] = Function No. 34
Header (1) [1:16] = Logical Station Number

Due to error recovery requirements, it may become necessary for the MCS to perform a Make Station Ready request. Refer to error result values which require this function as a response.

For example, if the result message from the ORS DLP indicates a station error, the MCS must perform a Make Station Ready request. This is required because the ORS DLP makes the station not ready if an irrecoverable error is encountered for a station.

The ORS DLP does not analyze the queue flag in the header for this function; it is performed immediately by the ORS DLP firmware.

3.4.1.5 MAKE STATION NOT READY (FUNCTION 35)

The Make Station Not Ready function supplies the MCS the ability to control the readiness of a station. For example, if a station is being polled because it is input-enabled and the MCS desired to have polling and/or input temporarily suspended, it may issue a Make Station Not Ready request. A result header is returned.

MAKE STATION NOT READY HEADER FORMAT

Header (0) [1:8] = Function No. 35
Header (1) [1:16] = Logical Station Number
Header (9) [4:1] = 0 = Bottom queue
1 = Top queue

Because of error recovery requirements, it may become necessary for the MCS to perform a Make Station Not Ready request. The ORS DLP will check if the line of this station is ready and if it is, place this function in the station's message queue. If the line is not ready, the message will be placed in the top of the station's message queue regardless the queue bit.

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. The Station is now not READY.

3.4.1.6 SET CHARACTER (FUNCTION 37)

The Set Character function allows a MCS to change/restore certain characters used for polling, end-of-message, backspacing, etc. The particular character(s) for which the change is being requested must have been specified as "Dynamic" in the Line Process translation table. Note that the control character is always dynamic and the address characters, if any, are always changeable. A result header is returned.

Set Character Header Format:

```
Header (0) [1:8]      = Function No. (37)
Header (1) [1:16]     = Logical Station No.
Header (2) [1:8]      = 0 = Set Control Character
                        1 = Set End-of-Message Character
                        2 = Set Backspace Character
                        3 = Set Line Delete Character
                        4 = Set WRU Character
                        5 = Set Address Character
                            (See Header (2) [13:2])
Header (2) [13:2]     = Number of Transmit Address Characters
Header (2) [15:2]     = Number of Receive Address Characters
Header (3) [1:16]     = Character Count (Bytes)
Header (9) [4:1]      = 0 = Bottom queue
                      = 1 = Top queue
Text   (0) [1:8]      = First Transmit Address
Text   (0) [9:8]      = Character Ctrl/First Receive Address
Text   (1) [1:8]      = Second Transmit Address
Text   (1) [9:8]      = Second Receive Address
Text   (2) [1:8]      = Third Transmit Address
Text   (2) [9:8]      = Third Receive Address
```

The Set Character function always requires a text portion.

If the Set Character function cannot be performed (that is, character not declared dynamic), an error result of type 10, (word (19 [1:8] = 17) an inappropriate request is returned.

3.4.1.6 SET CHARACTER (FUNCTION 37) (Continued)

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result.

10,16 Specified character is invalid for this Line Process.

3.4.1.7 SET TRANSMISSION NUMBER (FUNCTION 38)

The Set Transmission Number function allows a MCS to reinitialize the Transmit or Receive (or both) transmission number. The number must be represented in digit format. A result header is returned.

Set Transmission Number Header Format:

```
Header (0) [1:8]     = Function No. (38)
Header (1) [1:16]    = Logical Station No.
Header (2) [15:1]    = 1 = Set Transmit Transmission No.
Header (2) [16:1]    = 1 = Set Receive Transmission No.
Header (3) [1:16]    = Character Count (Bytes)
Header (9) [4:1]     = 0 = Bottom queue
                     = 1 = Top queue
Text   (0) [5:12]    = Transmit Transmission No.
Text   (1) [5:12]    = Receive Transmission No.
```

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result.

3.4.1.8 RECALL (FUNCTION 40)

The Recall function allows a MCS to request the return of all messages in the station output queue for the specific Logical Station Number. The station or line must be in a Not Ready state at the time the function is issued by the MCS. The Not Ready state may either have been initiated by the MCS or it may be the result of a previous station error (terminate error). If both the line and station are ready, a result header type 10 (header word (10) [1:8] = 15) (i.e., unable to initiate function) is returned.

Recall Header Format:

Header (0) [1:8] = Function No. (40)
Header (1) [1:16] = Logical Station No.

Each recalled message will have a Recall Result header (type 12) returned with it. The last recall will be followed by a result header (type 02) and an original function of 40.

The "Recall" result header format for the last message in the queue is as follows:

Header (0) [1:8] = 02
Header (0) [9:8] = 40
Header (1) [1:16] = Logical Station Number

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. This result indicates messages recalled from the station have been returned to the host.

10,15 Recall not performed. Either the station or line must be not ready prior to a Recall.

3.4.1.9 SET LOGICALACK (FUNCTION 41)

The Set Logicalack function allows the MCS to set the station Logicalack flag true for disposition by the ORS DLP. Following the successful reception of input from the station, the Line Process sends a Good Input Received message to the system, sets WAIT_ACK boolean in the Station Table and suspends itself with no timeout until an Acknowledge (function 43) is issued to the ORS DLP by the MCS. The Executive module will resume the suspended Line Process. A result header is returned.

Set Logicalack Header Format:

Header (0) [1:8] = Function No. (41)
Header (1) [1:16] = Logical Station No.
Header (9) [4:1] = 0 = Bottom queue
1 = Top queue

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. The Station attribute LOGICALACK is now true.

3.4.1.10 RESET LOGICALACK (FUNCTION 42)

The Reset Logicalack function allows the MCS to reset the station Logicalack flag. This function permits the ORS DLP to acknowledge a station without host system or MCS action. A result header is returned.

Reset Logicalack Header Format:

Header (0) [1:8] = Function No. (42)
Header (1) [1:16] = Logical Station No.
Header (9) [4:1] = 0 = Bottom queue
1 = Top queue

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. The Station attribute LOGICALACK is now true.

3.4.1.11 ACKNOWLEDGE (FUNCTION 43)

The Acknowledge function causes the ORS DLP to resume execution of the Line Process where execution has been suspended waiting for acknowledgement. If Acknowledge function is issued when not waiting on an acknowledge, a result header (type 10, result byte index = 15) (i.e., unable to initiate function) is returned. Otherwise, no result header is returned for this function.

Acknowledge Header Format:

Header (0) [1:8] = Function No. (43)
Header (1) [1:16] = Logical Station No.

POSSIBLE RESULTS (Result Type, RBI)

10,15 The line was not waiting for acknowledge from the host.

3.4.1.12 NULL STATION REQUEST (FUNCTION 44)

The Null Station Request function causes a message to be queued at the ORS DLP in the appropriate Station Queue. Upon encountering this message, the ORS DLP performs no action other than to return the "Good Result" (type 02) result header having an original function of 44.

This function affords the MCS the ability to insert a marker to signal the end of a series of outputs or a batch within a series of outputs to a station. A result header is returned.

Null Station Request Header Format:

Header (0) [1:8] = Function No. (44)
Header (1) [1:16] = Logical Station No.
Header (9) [4:1] = 0 = Bottom queue
1 = Top queue

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good result.

3.4.1.13 SET SEQUENCE MODE (FUNCTION 45)

The Set Sequence Mode function allows a MCS to set the Sequence toggle for use in the Line Process and provides the base (or starting) sequence, the increment value, and the maximum number of digits allowed for the sequence number. All values are expressed in digit format. A result header is returned.

Set Sequence Mode Header Format:

```
Header (0) [1:8]       = Function No. (45)
Header (1) [1:16]      = Logical Station No.
Header (2) [13:4]      = Maximum Number of Digits (1-8)
Header (3) [1:16]      = Character count (Bytes)
Header (9) [4:1]       = 0 = Bottom queue
                         1 = Top queue
Header (12)[1:16]      = Beginning Sequence Number MS
Header (13)[1:16]      = Beginning Sequence Number LS
Text   (0) [1:16]      = Increments MS
Text   (1) [1:16)      = Increments LS
```

POSSIBLE RESULTS (Result Type, RBI)

10,17 This Station is not a "sequence" type terminal.

3.4.1.14 RESET SEQUENCE MODE (FUNCTION 46)

The Reset Sequence Mode function allows the MCS to request the ORS DLP to reset the "Sequence" toggle for use in the Line Process. A result header is returned.

Reset Sequence Mode Header Format:

```
Header (0) [1:8]       = Function No. (46)
Header (1) [1:16]      = Logical Station No.
```

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good result.

3.4.1.15 MAKE LINE READY (FUNCTION 47)

For a MCS to be able to effectively accomplish error recovery, it may become necessary for the MCS to perform a Make Line Ready. Certain values, returned in an error result header from the ORS DLP, require that the MCS subsequently responds with a Make Line Ready function; until that time, the line remains in a Not Ready state. No result header is returned if operation is successful.

Make Line Ready Header Format:

Header (0) [1:8] = Function No. (47)
Header (1) [1:16] = Logical Station No.

If the line is already in a Ready state, this function acts as a no-operation. If line is Not Ready, it is then marked "Ready" and the ORS DLP checks to see if there is a Line Process present in the line to which the function is issued. If no Line Process is present, a result header type 10, original function = 99 and result byte index = 8 is returned.

3.4.1.16 MAKE LINE NOT READY (FUNCTION 48)

The Make Line Not Ready function allows a MCS to cause a line to be made Not Ready.

Make Line Not Ready Format:

Header (0) [1:8] = Function No. (48)
Header (1) [1:16] = Logical Station No.
Header (9) [3:1] = 1 = Immediate (Always 1)

If the Immediate bit (i.e., header (9) [3:1]) is set, the line is made Not Ready immediately. No check is made of the "Line's Busy" status. Any messages currently in the process of being transmitted or received on the line are lost.

POSSIBLE RESULTS (Result Type, RBI)

05,00 Line is not ready.

3.4.1.17 DIALOUT (FUNCTION 49)

The Dialout function causes the ORS DLP Dialout firmware initiation and supplies the up-to-15-digit phone number. This function is avaliable only to line 0 of the ORS DLP. There must also be a line declaration respecting a type of line in the Line Data Structure.

Dialout Header Format:

```
Header (0) [1:8]    = Function No. (49)
Header (1) [1:16]   = Logical Station No.
Header (3) [1:16]   = Number of Digits
Text   (0) [9:8]    = Four Digits
Text   (1) [1:8]    = Four Digits
Text   (2) [9:8]    = Four Digits
Text   (3) [1:8]    = Four Digits
```

Telephone digits must be left-justified in the text and consist of numeric digits. For example, 912152698575, where the "91" is necessary to obtain an "outside" line.

POSSIBLE RESULTS (Result Type, RBI)

03,00 Good Result. The dial operation was successful and the line is connected.

10,15 Line is not ready, busy or ringing.

10,17 Line is not declared as "dialout".

10,18 Unable to complete call; DSR is not present after dial operation.

3.4.1.18 DISCONNECT (FUNCTION 50)

The Disconnect function causes the ORS DLP to hang up a line connected through a telephone switchboard or switching network. The line must have been declared as "Dialin" or "Dialout" and must be ready, connected, and not busy. After completion of the Disconnect function, the ORS DLP monitors the line for subsequent incoming calls. A Result Header is returned.

Disconnect Header Format:

Header (0) [1:8] = Function No. (50)
Header (1) [1:16] = Logical Station No.

POSSIBLE RESULTS (Result Type, RBI)

03,00 Good Result. The line now is disconnected.

10,15 Line is not connected.

10,17 Line is not declared as "dialout" or "dialin".

10,18 Unable to complete disconnect; DSR is present after this operation.

3.4.1.19 ANSWER THE PHONE (FUNCTION 51)

The Answer The Phone function causes the ORS DLP to answer an incoming call from a station or a line that was not marked as a line to be automatically answered. The line must have been declared as "Dialin" and must be in a Ready and Disconnected but Ringing state.

Answer the Phone Header Format:

Header (0) [1:8] = Function No. (51)
Header (1) [1:16] = Logical Station No.

The Logical Station Number for this function is supplied to the MCS through the use of a Line Status Result (type 03) in header word (1) [1:16]. The ringing indicator is Result header word (2) [9:1]. Refer to the Line Status Result (type = 03) paragraph in Section 3.5.1.4.

POSSIBLE RESULTS (Result Type, RBI)

03,00 Good Result. The phone has been answered and the line is connected.

10,15 Line is not ringing.

10,17 Line is not declared as "dialin".

10,18 Unable to complete connection; DSR is not present after this operation.

3.4.1.20 INTERROGATE LINE STATUS (FUNCTION 52)

In response to this function, the ORS DLP generates a Status Result (type 03) message.

The ORS DLP interrogates the current status register of the line for such data set interface levels as Data Set Ready (CC), Data Terminal Ready (CD), and Ring Indicator (CE). Refer to Line Status Result (type = 03) paragraph.

Interrogate Line Status Header Format:

Header (0) [1:8] = Function No. (52)
Header (1) [1:16] = Logical Station No.

POSSIBLE RESULTS (Result Type, RBI)

03,00 Good Result.

3.4.1.21 SET AUTOANSWER (FUNCTION 53)

The Set Autoanswer function is allowed for stations on lines declared as "Dialin". When these "Dialin" lines are in a disconnected state, the ORS DLP monitors the Data Set Ring Indicator (CE) for incoming calls.

When the ringing is detected (if Autoanswer is true for the line), the ORS DLP returns a Status Result (type 03) message that shows the line's new status as "disconnected", but Switched Busy. The ORS DLP will answer the phone automatically, exactly as an explicit Answer function (No. 51). No result header is returned for the function itself.

Set Auto Answer Header Format:

Header (0) [1:8] = Function No. (53)
Header (1) [1:16] = Logical Station No.

POSSIBLE RESULTS (Result Type, RBI)

10,17 Line is not declared as "dialin".

3.4.1.22 RESET AUTOANSWER (FUNCTION 54)

The Reset Autoanswer function is allowed for stations on lines declared as "Dialin". This function causes the Autoanswer flag in the appropriate line table to be set to false.

When ringing is detected, the ORS DLP generates a Status Result (type 03) message and passes it to the MCS to indicate the line status as Disconnected and Ringing. No result header is returned for the function itself.

Reset Autoanswer Header Format:

Header (0) [1:8] = Function No. (54)
Header (1) [1:16] = Logical Station No.

POSSIBLE RESULTS (Result Type, RBI)

10,17 Line is not declared as "dialin".

3.4.1.23 MEMORY DUMP (FUNCTION 61)

The Memory Dump function provides the MCS with the ability to dump the ORS DLP main memory to the host. The MCS can dump either an absolute memory location or any declared Line/Station table. Result header (type 11) is returned with this function.

Memory Dump Header Format:

```
Header (0) [1:8]      = Function No. (61)
Header (1) [1:16]     = Logical Station No.
Header (2) [13:4]     = 0 = Dump memory
                        1 = Dump Line Table
                        2 = Dump Station Table
Header (3) [1:16]     = Character Count (Bytes)
Header (15)[1:16]     = ORS DLP Memory Address (Offset)
Header (16)[1:16]     = ORS DLP Memory Address (Segment)
```

The MCS will either process this information or pass it to some other program for processing.

The text size must not be greater than the MCS record area in which input is processed. Multiple requests will be required to dump the entire memory.

POSSIBLE RESULTS (Result Type, RBI)

11,00 Memory Dump Result.

10,15 Memory Dump function contains incorrect information.

3.4.1.24 MCS STATUS CHANGE (FUNCTION 66)

The MCS Status Change is a MCP initiated function used for flow control to inuicate to the ORS DLP the Ready or Not Ready status of the MCS. When a MCS opens a DCP file, the MCP will send this function header to the ORS DLP with the "open-and-ready" option. When the MCS queue becomes full, the MCP will send this header with the "open-and-blocked" option. When space in the MCS's queue becomes avaliable, the MCP will send "open-and-ready" option to allow messages to be sent to the MCS's queue again.

The "closed" option will be sent to the ORS DLP when a MCS closes its DCP file. The ORS DLP will remove all the result messages for that MCS and purge all the Station send queues for all stations attached to that MCS.

No result header is returned with this function.

MCS Status Change Format:

```
Header (0) [1:8]      = Function No. (66)
Header (1) [1:16]     = Binary MCS number
Header (2) [15:2]     = 0 = Closed
                        1 = Open and Ready
                        2 = Open and Blocked
```

3.4.1.25 MEMORY LOAD (FUNCTION 71)

The Memory Load function allows the MCS to load the ORS DLP main memory. This function is primarily used to load Line and Station data stuctures and the Line Process code. The address where the information will be loaded will be determined by the ORS DLP firmware.

Memory Load Header Format:

```
Header (0) [1:8]      = Function No. (71)
Header (1) [1:16]     = Logical Station Number
Header (2) [5:4]      = 0 = Nominal size Line Process
                        1 = First block of enlarged
                            Line Process
                        2 = Unprecedented, but not last block
                            of enlarged Line Process
                        3 = Last block of enlarged
                            Line Process
Header (2) [9:4]      = 0 = Invalid
                        1 = Load LINE_TABLE
                        2 = Load STATION_TABLE
                        3 = Load LINE_PROCESS
                        4 = Load DIAL_PROCESS
                        5 = Start LINE_PROCESS
Header (2) [13:4]     = Relative line number
Header (3) [1:16]     = Character count (Bytes)
```

The ORS DLP firmware allocates and links the data structures or code accordingly. The Load Line Table variant does not use the Logical Station Number field.

Header (2) [5:4] is valid only in case of Load Line Process (Header (2) [4:4] = 3), and it controls the load sequence of the Line Process. Line Process below 8000 bytes in size could be loaded in one block (Header (2) [5:4] = 0). Line Processes above 8000 bytes and up to 16000 bytes should be loaded using Header (2) [5:4] as a load control. Only a complete load of a Line Process will return a result header.

POSSIBLE RESULTS (Result Type, RBI)

02,00 Good Result. Successful Load operation.

10,15 Unsuccessful attempt to perform Memory Load function.

3.4.1.26 ECHO (FUNCTION 72)

The Echo function allows the MCS or ORS DLP test/confidence routines to exercise the ORS DLP firmware and hardware. The variant bits word 2 [1:16] will determine the depth where the Echo function will be performed. Result header (type 11) is returned with this function.

Echo Header Formart:

Header (0)	[1:8]	= Function No. (72)
Header (1)	[1:16]	= Logical Station Number
Header (2)	[1:16]	= 0 = Executive Echo 1 = Line Process Echo 2 = Line USART Echo 3 = Line Interface Echo
Header (3)	[1:16]	= Character Count (Bytes)
Header (9)	[4:1]	= 0 = Bottom queue 1 = Top queue

POSSIBLE RESULTS (Result Type, RBI)

11,00 Good Echo Result.

10,15 Unsuccessful attempt to perform Echo function.

3.5 RESULT MESSAGES FORMAT

The Result Headers and Message Headers have the same general format. A more detailed description of each pertinent Result Header is provided in the following sections.

The following figure shows the Result Header Block Layout:

```
Word 00  RRRRRRRRFFFFFFFF
     01  LLLLLLLLLLLLLLLL
     02  VVVVVVVVVVVVVVVV
     03  TTTTTTTTTTTTTTTT
     04  EEEEEEEEEEEEEEEE
     05  EEEEEEEEEEEEeeee
     06  ........rrrrrrrr
     07  xxxxxxxxxxxxxxxx
     08  xxxxxxxxyyyyyyyy
     09  ....tttttttttttt
     10  BBBBBBBB........
     11  NNNNNNNNNNNNNNNN
     12  SSSSSSSSSSSSSSSS
     13  SSSSSSSSSSSSSSSS
     14  MMMMMMMMMMMMMMMM
     15  ................
     16  ................
     17  ................
     18  ................
     19  ................
```

3.5 RESULT MESSAGES FORMAT (Continued)

R - Result type. The result type field contains one of a possible number of values that determines the way in which the remainder of the Result Header is interpreted. The possible values are as follows:

VALID DCP RESULTS

Header Code	Header Description
00	Good Input
01	Station Event
02	Good Results
03	Line Status
05	Line Status Change
10	Error Result
11	Memory Dump Result
12	Recall Result

F - Original Function. This field (Word 0 [9:8]) contains the original function number that was issued to the ORS DLP. In the case of an unsolicited result, a BCD 99 is returned in this field.

L - LSN. This field (Word 1 [1:16]) contains the Logical Station Number of the station which the result is from.

3.5 RESULT MESSAGES FORMAT (Continued)

V - Variants. This field (Word 2 [1:16]) contains information that further qualifies the result header. This field is valid if the result type is 03, 05, or 10. The meaning of the bits in this field and the mapping done to get this information is as follows:

LINE CONDITION FIELD

Bit	Description
1	Line Control Mode
2	Busy
3	Line Queued
4	Terminate Logicalack executed
5	Acknowledge Logicalack required
6	Line Ready
7	Direct Connect
8	Reserved
9	Ringing
10	Switched Busy
11	Connected
12	Autoanswer
13	Dialout
14	Dialin
15	Disconnect Pending
16	Output Request set active

I - Text Size. For result headers that imply input from the ORS DLP, this field (Word 3 [1:16]) contains the number of bytes of text that are returned with the result. This field is in hexadecimal and will be converted by the Medium System MCP to decimal.

3.5 RESULT MESSAGES FORMAT (Continued)

E - Exception Flags. The exception flags field consists of 28 bits (Word 4 [1:16] and Word 5 [1:12]). It is used to indicate any errors or situations encountered by the ORS DLP while attempting to perform a particular request. This field is valid on 00, 01, 02, 05, and 10 results. The meaning of the bits and the mapping done to get this information is as follows:

Note: All Result Header fields except Result Type 10, Original Function, and Result Byte Index, may contain invalid information if a Valid Line Table, Station Table, or Line Process have not been successfully loaded.

EXCEPTION FLAGS

Word	Bit	Description
4	1	Time out
	2	Stop bit error
	3	Buffer overflow
	4	Break on input
	5	Vertical parity error
	6	Loss of carrier
	7	Break on output
	8	Horizontal parity error
	9	Address character error
	10	Transmission number error
	11	Format error
	12	Output was NAK'd
	13	Control character received
	14	WRU received
	15	Sequence number overflow
	16	Message to be acknowledged
5	1	NAK-ON-SELECT
	2	End of buffer
	3	Disconnect
	4	Event1
	5	Line not ready
	6	Cluster not ready
	7	Station not ready
	8-12	Reserved

3.5 RESULT MESSAGES FORMAT (Continued)

r - Retry Count. This field (Word 6 [9:8]) indicates the number (in binary) of retries remaining at the time the ORS DLP finished the request. This field contains this value only for result types of 00, 01, or 02. If the result type is 03, 05 or 10, then this field contains a report of the adapter status. The meaning of these bits and the mapping done to get this information is as follows:

ADAPTER STATUS

Bit	Description
9	Parity error
10	CB true (CTS)
11	CE true (Ring indicator)
12	Line adapter present
13	CF true (Carrier)
14	CC true (Data set ready)
15	SCF true (Secondary carrier)
16	(Reserved)

x,y - Tally and Tog Fields. These fields are used as a means of communication between the MCS and the Line Process. They are only valid for 00, 01, 0230, or 10 headers. The location of these fields and the mapping done to get this information is as follows:

TALLYS AND TOGS

Location	Field
Word (7) [1:8]	TALLY (0)
Word (7) [9:8]	TALLY (1)
Word (8) [1:8]	TALLY (2)
Word (8) [9:8]	TOGS (7-0)

3.5 RESULT MESSAGES FORMAT (Continued)

t - Transmission Number. This field (Word 9 [5:12]) contains the transmission number received from the station. This field is only valid for 00 and 01 result headers. The mapping done to get this information is as follows:

TRANSMISSION NUMBER

Location	Field
Word (9) [5:4]	MSD
Word (9) [9:4]	Middle
Word (9) [13:4]	LSD

B - Result Byte Index (RBI). This field (Word 10 [1:8]) is a further extension of the result header definition. Non-error result headers have a RBI of zero. The values are expressed in decimal digit format. RBI values are as follows:

RBI VALUES

RBI	Description
00	Good results.
01	Terminate Error. (Station made not ready)
02	Terminate No Label. (Station made not ready)
03	Reserved.
04	Reserved.
05	Transmit Abort. (Line not ready)
06	Receive Abort. (Line made not ready)
07	Terminate Nospace. (Line made not ready)
08	Line Failure at Request Initialization. (Line made not ready)
09	Reserved.
10	Cluster Parity Error. (Cluster made not ready)
11	Cluster Timeout Error. (Cluster made not ready)
12	Cluster Scanner Error. (Cluster made not ready)
13	Reserved
14	Reserved
15	Inappropriate MCS request. (No action)
16	Receiving Location Occupied. (No action)
17	Not applicable to station or line. (No action)
18	Unable to complete. (No action)

3.5 RESULT MESSAGES FORMAT (Continued)

N - Message Number. This field (Word 11 [1:16]) is a 4 digit value that may be used by the MCS to ensure that no messages are lost in or out of the system. The value placed in this field is untouched by the ORS DLP and is returned in the result header exactly as received in the message header. This field has no significance in a result type 00 or 01.

M - Original Variant. This field (Word 14 [1:16]) contains a copy of the original Variant field in the message header (Word 2 [1:16]). It is valid on all result types.

3.5.1 RESULT MESSAGES

A description of the various types of result messages is detailed in the following paragraphs.

3.5.1.1 GOOD INPUT RECEIVED (Type 00)

Receipt of a message of this type is a direct result of a successful reception of a data comm message by the ORS DLP or as a result of a successful execution of an "Enableinput" message. Text, if any, appears following the header, and is valid information for a total of the text size field (header 3) characters. The Error Flag field may have one or more bits set in this message, although the input was successful.

The result format is as follows:

Result type = 00, Original function = 32, LSN, Text Size, Exceptions (for result type = 00, exceptions will be zero), Retry, Tallies, Togs, Tran numbers, and Sequence numbers. Result Byte Index will be = 00. Original Variant is also present.

When a MCS receives a Station Event result from the system, it indicates a noteworthy event has occurred to or at one of the stations under the control of the MCS. The "station event" which occurred is indicated in the Exception Flags, header (4) bits 1-16 and header (5) bits 1-7.

3.5.1.2 STATION EVENT RESULT (Type 01)

The Station Event result is the same as a result type 00 (Good Input), but with some error/situation conditions reported in the Exception Flags field. In addition, a result type 01 is returned to the MCS when the Line Process waits for a Logicalack from the MCS when the logicalack option of the station is set. In this case a result type 01, Original exception 32, is returned with header word 4, bit 16 set (i.e., message to be acknowledged). The MCS must respond with an Acknowledge function (function 43) to the station before the Line Process of that station or any other stations attached to that same line.

More than one exception may be flagged in the Exception Flag field for any given Station Event result, but this is determined to some extent by the Line Process.

The action, taken by the MCS upon receipt of a Station Event result, depends upon particular installation requirements. For example, upon receiving a Station Event message in which header word 4, bit 13 is 1 (message contains station's "Control" character), the MCS might treat the message text in a different manner than other input messages.

Note: If the Text Length field, header (3), is non-zero, a valid message text is returned with the Station Event result. For example, a vertical parity (character) error may have been detected by the Line Process while receiving an input message from a station. However, on a subsequent retry of the input message, the Line Process was able to successfully receive the message from the station. In this case, a Station Event result is returned to the MCS complete with the station's input message and an indication that a vertical parity error occurred prior to the message being successfully received.

The result format is as follows:

Result type = 01. The rest of the header is the same format as Result type 00.

3.5.1.3 GOOD RESULT (Type 02)

The type and variant fields of the original function are found in each result header in header word 0, bits 9-16 and header word 14, bits 1-16, respectively. If the Original function type were one of the switched line requests (Dialout, Disconnect, Answer or Interrogate Switched Status), the message format is instead a Status message format, type 03.

The Error Flag field may have one or more bits on in the result header but the request itself was honored by the ORS DLP. That is, the ORS DLP was able to recover successfully from any and all of the conditions encountered.

Good Result messages indicate the operation requested from the ORS DLP (e.g., Make Station Not Ready, Write) has been successfully completed.

The result format is as follows:

Result type = 02, Original function, LSN, Text size if Original function = 30, Exception Flags, Retry, Tallies and Togs if Original function = 30, Message number, Sequence number if Original function = 30 and implemented and Original variant.

3.5.1.4 LINE STATUS RESULT (Type 03)

The origin of the "Line Status Result" message is either:

a. The result of
The "Dialout" (Function = 49) MCS request,
The "Disconnect" (Function = 50) MCS request,
The "Answer The Phone" (function = 51) MCS request,
The "Interrogate Line Status" (Function = 52) MCS request, or

b. An unrequested change of (line) status automatically reported by the ORS DLP (for example, an unexpected disconnect or connection by the "Autoanswer" capability).

The result format is as follows:

```
Header (0)  [1:8]   = Result Type 03
Header (0)  [9:16]  = Original Function
Header (1)  [1:16]  = Logical Station Number
Header (2)  [1:16]  = Line Condition Report
                      (Refer to description of Variant)
Header (6)  [9:16]  = Adapter Status Report
Header (10) [1:8]   = Result Byte Index
```

The Result Byte Index (RBI) may have the digit value with different meanings for different dialout functions. Table 4-2 lists the switched status result values. If the result type is 03, the RBI will always be 00.

3.5.1.5 LINE STATUS CHANGE RESULT (Type 05)

A Line Status Change Result message is generated by the ORS DLP as a part of the response to the Make Line Not Ready as a result of a Line Abort.

The message format is as follows:

```
Header (0)  [1:8]   = Result Type 05
            [9:16]  = 48
Header (5)  [5:1]   = 1 = Line Note Ready
Header (6)  [9:16]  = Adapter Status Report
```

3.5.1.6 ERROR RESULT (Type 10)

The Error Result message is used by the "Data Comm" System to report errors on lines and stations over which the MCS has control. The Error Result message is recognized by the value 10 in the type field: header word 0, bits 1-8.

The format is as follows:

Header (0)	[1:8]	= Result Type 10
Header (0)	[9:16]	= Original Function
Header (2)*	[1:16]	= Condition Report. (Refer to Variant description)
Header (3)*	[1:16]	= Text size if Original Function=30
Header (4)*	[1:16]	= Exception Flag Field
& Header (5)*	[1:7]	
Header (6)*	[9:16]	= Adapter Status Report
Header (10)	[1:8]	= Result Byte Index

*Note: All Result Header fields except Result Type 10, Original Function, and Result Byte Index, may contain invalid information if a Valid Line Table, Station Table, or Line Process have not been successfully loaded.

Note: For Error results (type 10) with Original Function = 30, the message text associated with the function 30 request is returned with the Error Result.

The following RBIs require MCS error recovery action if traffic on a line or with a station is to continue.

01 = The Line Process marks a station "Not Ready"

02 = Reserved

03 = Reserved

04 = Reserved

05 = Transmit Abort. Error condition (loss of Data Set Ready or Clear to Send signal) encountered while attempting to transmit information (line made not ready).

06 = Receive Abort. Error condition (loss of Data Set Ready signal) encountered while receiving information (line made not ready).

3.5.1.16 ERROR RESULT (Type 10) (Continued)

07 = Terminate No Space. The ORS DLP has no memory to process this function.

The following RBIs do not require MCS error recovery since they do not cause a station or line to be made "not ready". They do report that the MCS is misdirecting requests which are being ignored by the ORS DLP. A list of these non-critical results appears below.

15 = Inappropriate Request. Logical prerequisites to perform operation are not met.

16 = Reserved.

17 = Not Applicable to Station or Line.

18 = Unable to Complete.

Non-Critical Error Results (Type 10)

Original Function	Result Byte Index	Interpretation
30	17	Station TERMINAL_TYPE set to input only.
32	17	Station TERMINAL_TYPE set to input.
36	17	Application number error zero or too high.
40	15	Station must be made Not Ready before recall.
43	15	Station not awaiting acknowledgement.
45	15	Sequence Mode already set.
	17	Not "Sequence" type terminal or greater than 8 digits were specified for sequence number.
49	15	Not Ready.
	17	Not Dialout.
	18	Unable to complete call.
50	15	Not Connected.
	17	Not Switched Line.
	18	Unable to Complete disconnect.
51	15	Line Not Ringing.
53	17	Not Dialin type.
54	17	Not Dialin type.

3.5.1.7 MEMORY DUMP RESULT (Type 11)

The Memory Dump Result message is always generated by the ORS DLP in response to a successfully completed Memory Dump function (61).

The message format is as follows:

```
Header (0)  [1:8]   = Result Type 11
Header (3)  [1:16]  = Text length in bytes
Header (15) [1:16]  = Memory address (Offset)
Header (16) [1:16]  = Memory address (Segment)
```

3.5.1.8 RECALL RESULT (Type 12)

Any messages queued in the system will be returned with a type 12 header as a result of a Recall (40) function. The MCS will receive a type 02/40 header after the last recalled message has been passed to the MCS.

The message format is as follows:

```
Header (0)  [1:8]   = Type 12
Header (0)  [9:16]  = Original function
Header (2)  [1:16]  = Original LSN
Header (14) [1:16]  = Original variant
```

3.5.1.9 UNSOLICITED RESULTS

Unsolicited results are Result Messages generated by the Line Process as a result of a Line/Station condition change. Unsolicited results have always "99" in Original Function field (Header 0 [9:16]). The folowing table lists possible Unsolicited Results:

Result	RBI	Description
10	02	The Line Process encountered one of the following error conditions: Parity, Buffer Overflow, Stopbit error, Timeout, Break or Loss of Carrier, but there was no provision to handle these errors within the Line Process.
10	05	The Line Process attempted to transmit but aborted because: "clear to send" signal was not detected within 7 seconds of raising the "request to send" signal, or the Line Process detects a loss of "data set ready" signal (Line made not ready).
10	06	The Line process attempted to receive but did not detect "data set ready" signal (Line made not ready).
10	07	The Line Process attempted unsuccessfully to allocate a memory block for a new input buffer or result header (Line made not ready).
10	08	Initialize abort. During the initialization phase, the Line Process detects absence of a Paddle card or "data set ready" in a non switched line.
03	00	Status change for a switched line. See Line Condition Report Header 2.

4 OPERATING FIRMWARE

OVERVIEW

The ORS DLP firmware is divided into two major parts; the PROM firmware and the RAM firmware. The PROM firmware consists of the following modules:

a. MLI HANDLER - This module is designed to assist the DLI interface logic to conform to the MLI protocol, to accept and to parse I/O descriptors from the host. This module is structured as an Interrupt handler and has the highest priority in the system.

b. OP HANDLR - OPERATION HANDLER is a task that is designed to assist the MLI HANDLER to conform to the MLI protocol. This task manages at the I/O descriptor level and at the ISC protocol level. The priority of this task is below line processes.

d. REQUESTOR - This module is called by OP HANDLR to issue a request to the host to start a Poll Request sequence.

e. EXECUTIVE - This module is made out of two seperate tasks; the ROUTER and the DISPATCHER. The ROUTER functions are to route DCP messages from/to Line Processes and to handle some of the DCP messages. The DISPATCHER functions are to dispatch DCP messages from the Line Processes or the EXECUTIVE to the OP HANDLER. Both the ROUTER and the DISPATCHER will get invoked when a Mailbox queue has a message from either OP HANDLR or LINE PROCESS. This module has the lowest priority in the subsystem.

f. VRTX - Operating system.

g. SELF TEST - Self test routine.

4 OPERATING FIRMWARE (Continued)

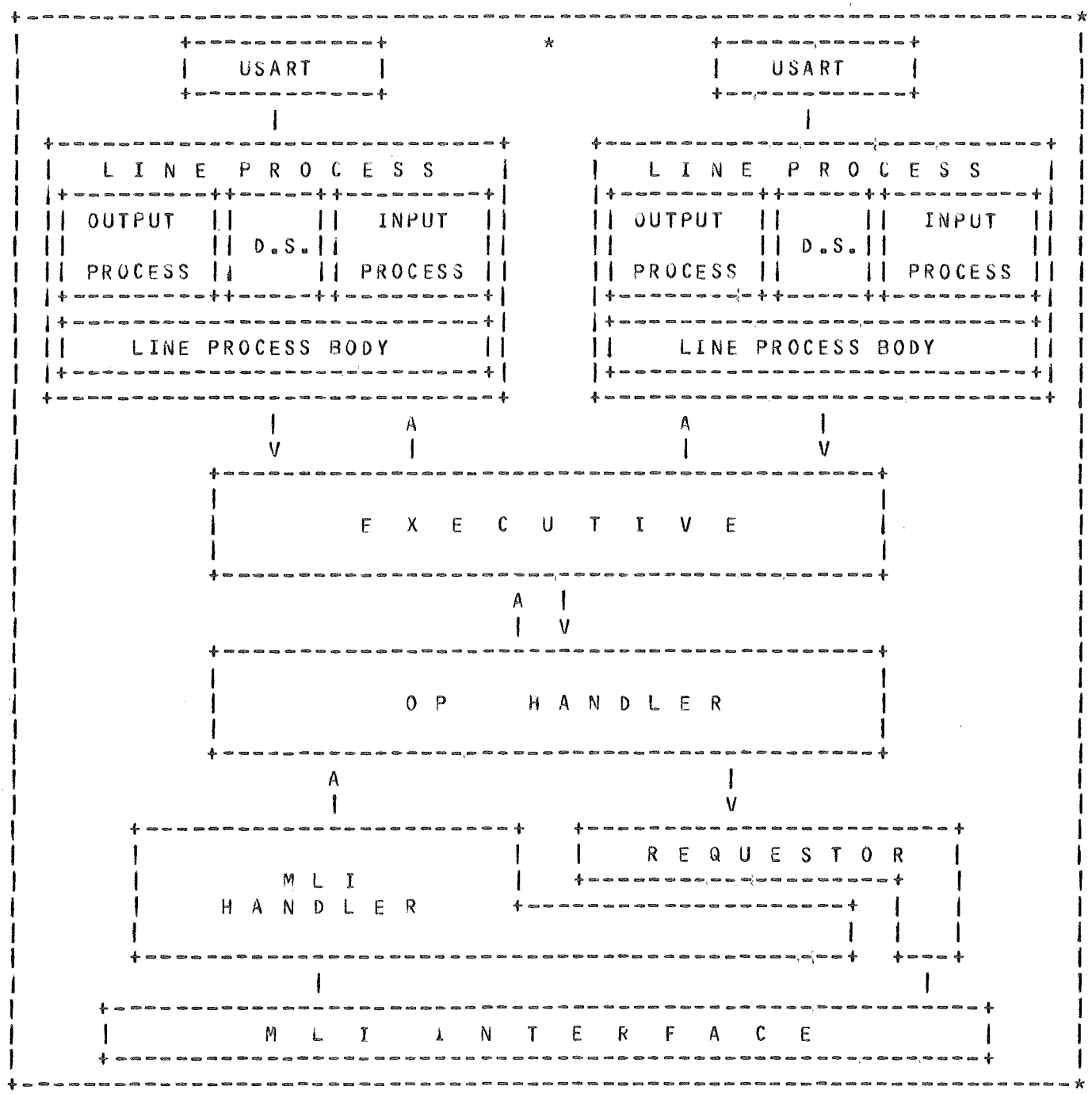
The RAM firmware is loaded from the host using the Load (71) message. The RAM firmware consists of up to 4 Line Processes. Each Line Process has a control over 1 line. Four submodules form a Line Process. The submodules are:

a. LINE PROCESS BODY - This submodule is the main procedure of the Line Process. Its major functions are establishing connection, initialization of the data comm chips, conforming to the operational protocol, and to maintain the station queues.

b. INPUT and OUTPUT PROCESS - These submodules are the two Interrupt driven submodules. The submodules will get invoked when a data comm interrupt occurs. The submodules determine the overall data comm performance of the ORS DLP, therefore the priority of these modules is second only to the MLI HANDLER.

c. LINE/STATION Data Structure.

The following block diagram gives an overall firmware structure.



4.1 MLI_HANDLER

The MLI_HANDLER is an Interrupt Handler which is invoked when the DLI State machine signals DLIMESS to the 8086 Microprocessor. The MLI_HANDLER performs two main tasks:

a. POLL TEST - Host connects to the ORS DLP to send an I/O Descriptor. This module implements the MLI protocol to accept and parse I/O descriptors from the host. Operations such as TEST ID, and CANCEL are handled immediately. Other operations are hanled by OP HANDLR module. The MLI HANDLER is capable of handling only one descriptor at a time. GET_MESSAGE operation is capable of accepting CANCEL during non active stages of the operation.

b. POLL REQUEST - The ORS DLP connects to the host to do one of the following tasks:

   1. Send Result Descriptor to the host
   2. Request data from the host for Write operation
   3. Send data to the host for Read operation

The DLI interface logic comes to assist the MLI HANDLER to perform the above tasks. The DLI interface logic consists of clear and self test initialization elements, DLI send/receive registers, burst counter, burst end logic, longitudinal parity word (LPW) generator, vertical parity generation and routing, request, emergency request logic and a 2K x 24 bit state machine.

The MLI_HANDLER is designed to run as fast as possible, since the the DLP cannot tie-up the MLI bus for too long and no other interrupts can be serviced when the MLI_HANDLER is executing. The MLI_HANDLER is designed not to use any form of procedure calls and returns since these instructions use considerable time. Macros are used in place of procedures. The paths that are usually executed are considered to be primary paths. The jump instructions will be taken to non-primary paths only.

The MLI_HANDLER should never stay in an infinite loop waiting for an event since the wait loop will tie-up the Microprocessor from doing other work.

4.1 MLI_HANDLER (Continued)

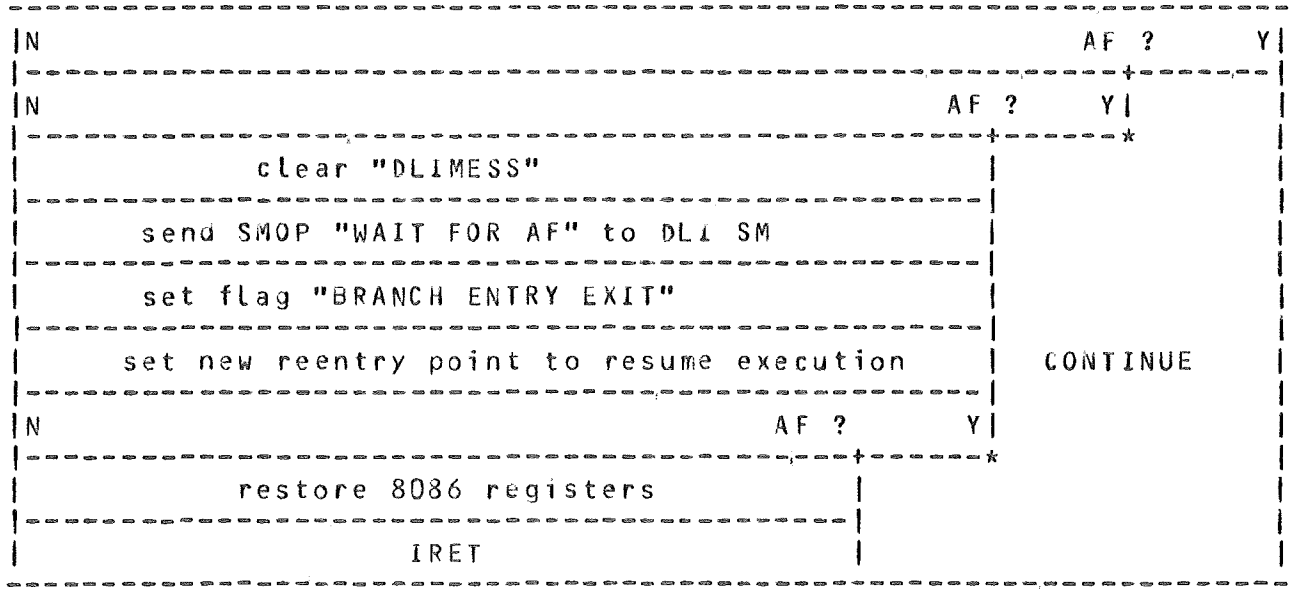
When the ORS DLP waits for a signal from the host, the MLI_HANDLER checks the signal 3 times before giving up. If the signal does not arrive the first 2 times, the MLI_HANDLER will set all the conditions to leave and return with interrupt. If the signal does not arrive at the third check, the MLI_HANDLER will force the DLI state machine to wait for the signal. When the signal finally comes, the DLI state machine will cause an interrupt. The MLI_HANDLER will resume execution from where it left off.

The Macros are:

A. SEND_DLIOP_WAIT_AF (SDWA)
B. SEND_DLIOP_WAIT_CONNECT (SDWC)
C. SEND_DLIOP_WAIT_DISCONNECT (SDWD)
D. INVOKE_OPHNDL

SEND_DLIOP_WAIT_AF

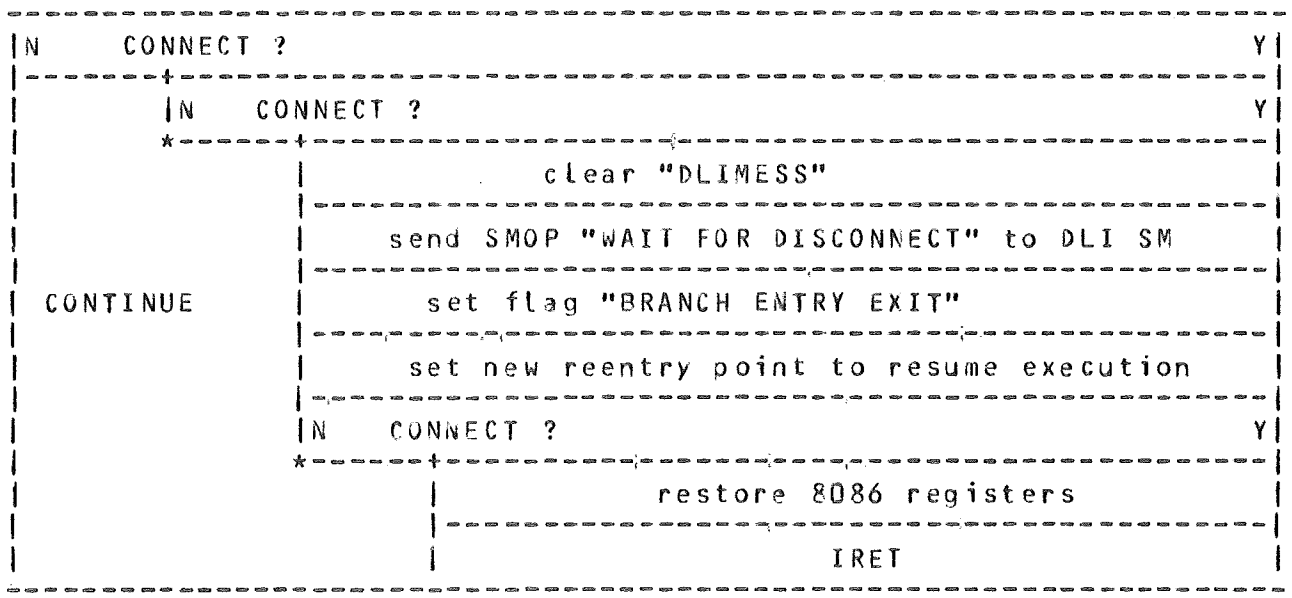
This Macro is used to wait for the Asyncronous Flag from the host while doing data transfers between the host and the ORS DLP. This macro checks the AF three times before giving up. If the AF signal does not arrive, it sets up WAIT_AF DLIOP in the DLI State Machine, sets flag and branch entry point to resume execution later.



4.1 MLI_HANDLER (Continued)

SEND_DLIOP_WAIT_DISCONNECT

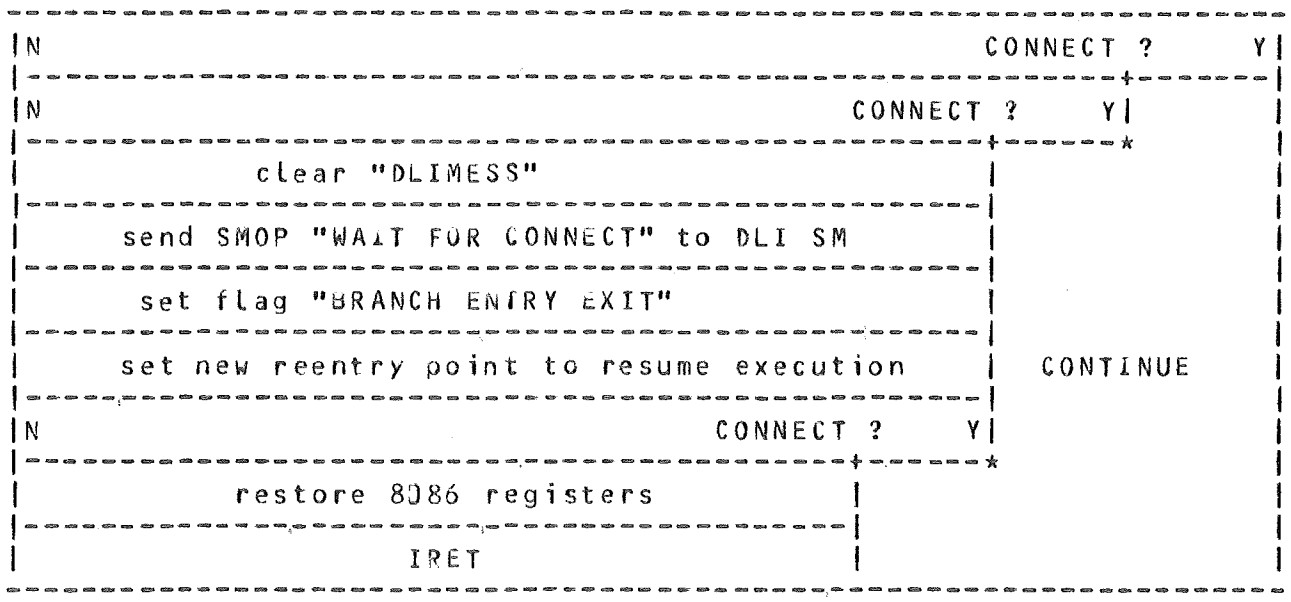
This Macro is used to wait for the host DISCONNECTs from the ORS DLP. This macro checks the CONNECT signal three times before giving up. If the CONNECT signal does not go off, it sets up WAIT_DISCONNECT DLIOP in the DLI State Machine, and sets the flag and branch entry point to resume execution later.



4.1 MLI_HANDLER (Continued)

SEND_DLIOP_WAIT_CONNECT

This Macro is used to wait for the host CONNECTs to the OCDLP. This macro checks the CONNECT signal three times before giving up. If the CONNECT signal does not arrive, it sets up WAIT_CONNECT DLIOP in the DLI State Machine, sets the flag and branch entry point for resuming execution later.



4.1 MLI_HANDLER (Continued)

INVOKE_OPHNDL

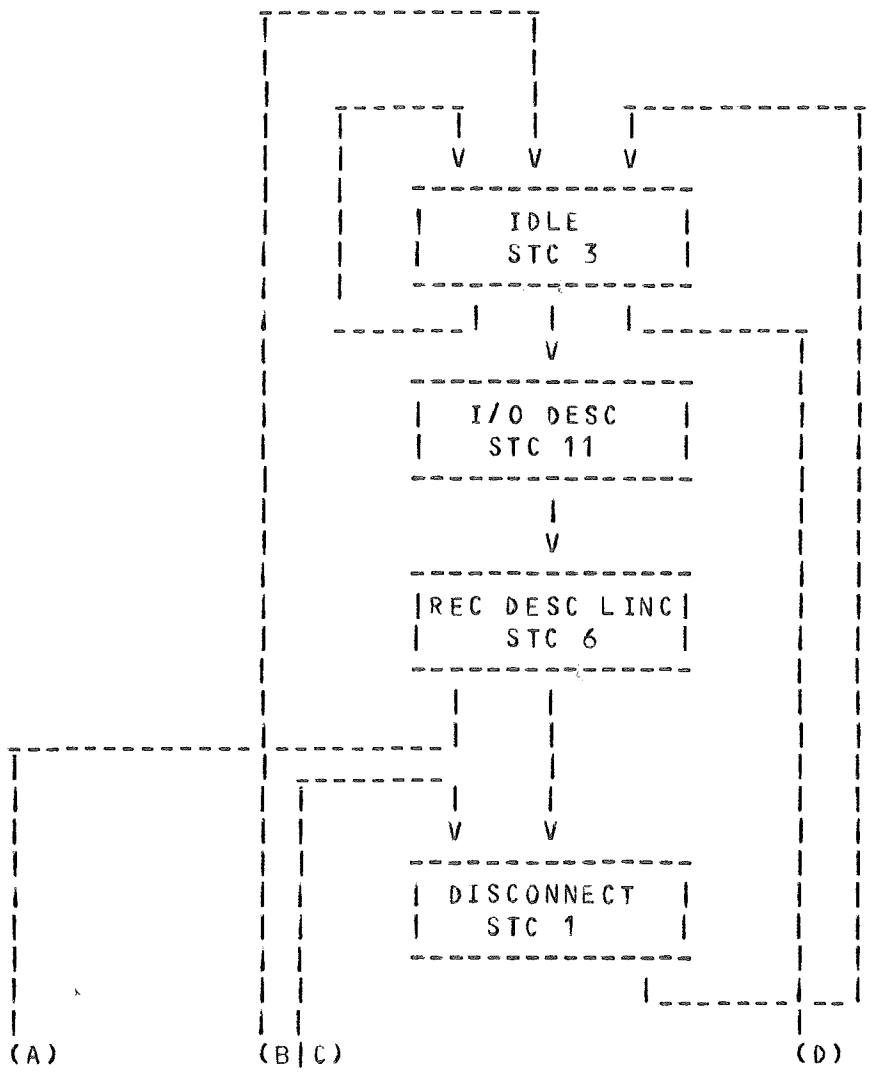
This macro enables the OP HANDLER INIT interrupt. The OP HANDLER INIT module will send a message to the OP HANDLER mailbox. The OP HANDLER task at this time is in suspended state waiting for a message. Arrival of a message will reinstate the OP HANDLER.

Enable interrupts
Post message to the OP HANDLER mailbox.
Macro SDWC
Restore 8086 contents
I R E T

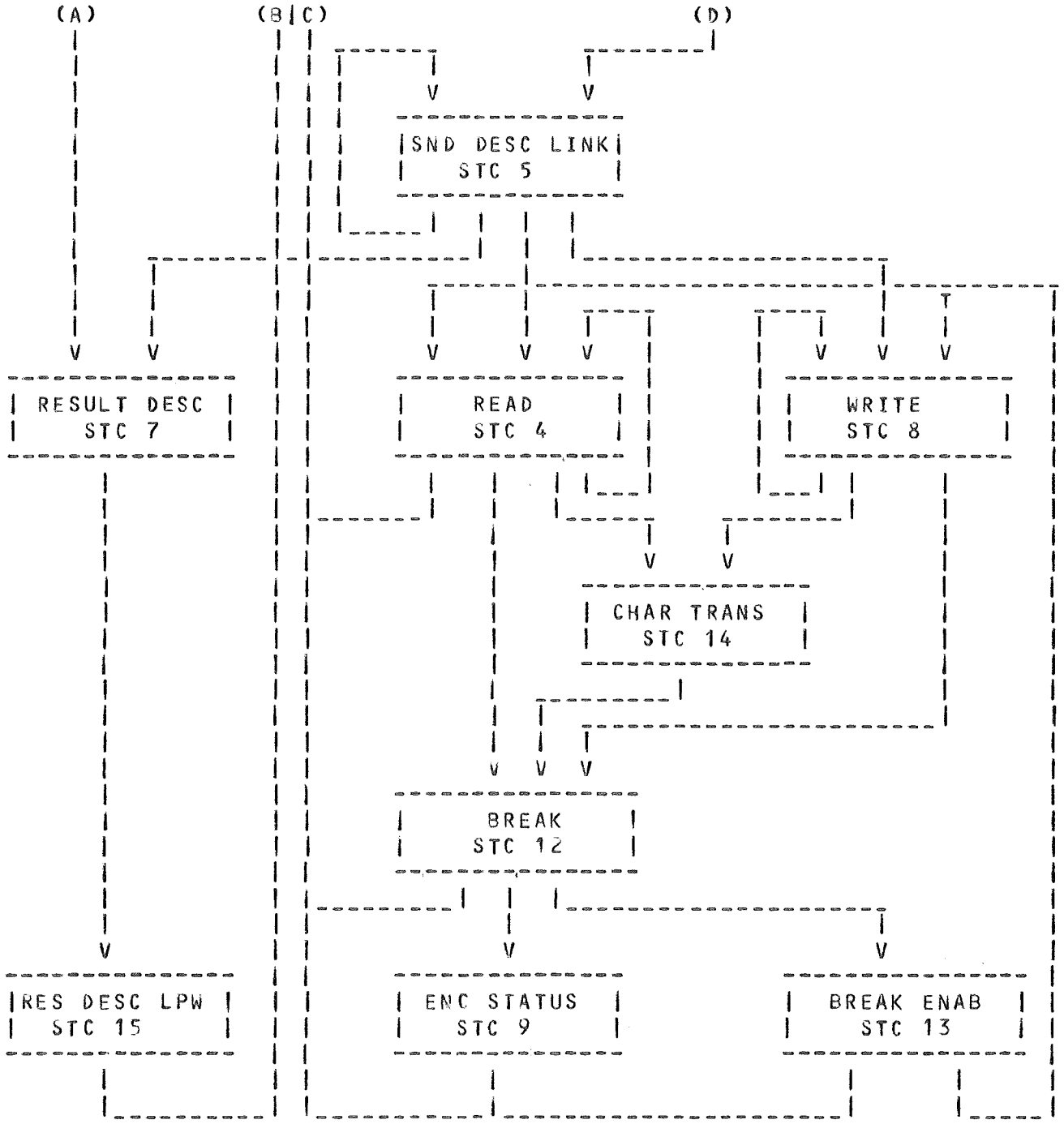
4.1.1 ORS DLP STATUS FLOW

The ORS DLP status flow is designed to enable message transfer to perform in an orderly manner. Status counts (STC) are transmitted continuously by the ORS DLP once it is connected to the host. The value of the status count lines are valid when the DLP STROBE is sent. The host can disconnect when the ORS DLP is in any status. If the ORS DLP is in any status except IDLE(STC 3) or DISCONNECT(STC 1), the ORS DLP must remain at that status and cannot alter the ORS DLP state until the host reconnects to the ORS DLP. If the ORS DLP is at IDLE status (STC 3) when the host disconnects, the ORS DLP can go to SEND DESCRIPTOR LINK (STC 5) only if the ORS DLP has descriptor link. ORS DLP is required to send from STC 6, Receive descriptor link, to STC 7, Result descriptor without disconnecting from the host.

4.1.1 ORS DLP STATUS FLOW (Continued)



4.1.1 ORS DLP STATUS FLOW (Continued)



4.1.1 ORS DLP STATUS FLOW (Continued)

This section includes a very brief description of each valid DLP status.

BREAK - STC=12

Indicates the end of a data message.

BREAK ENABLE - STC=13

Indicates the desire by the DLP to transmit another message to the system. The system may accept or refuse this request.

CHARACTER TRANSFER - STC=14

Used by DLPs which can handle single character per transfer mode to resolve the contents of this final data word.

CLEARED - STC=0

Entered by the DLP when cleared.

DISCONNECT - STC=1

Used by the DLP to indicate that no more transfers are possible during the connection, or to indicate that the DLP is unable to accept a new I/O descriptor.

ENCODED STATUS - STC=9

Indicates that the DLP needs to send special status information on the data lines.

4.1.1 ORS DLP STATUS FLOW (Continued)

IDLE - STC=3

Indicates that the DLP can accept a new I/O descriptor, or that a DLP receiving this new descriptor requests additional descriptor information.

I/O DESCRIPTOR LPW - STC=11

Indicates that the DLP requires only one more word of I/O descriptor followed by the I/O descriptor LPW.

RECEIVE DESCRIPTOR LINK - STC=6

Indicates that the DLP needs to receive, or is receiving the descriptor link. If the DLP detects a vertical or logitudinal parity error during transfer of the descriptor link, the DLP is transferred immediately from status six to seven.

READ - STC=4

Indicates that data is being transferred to the system by the DLP.

RESULT DESCRIPTOR - STC=7

Indicates that the result descriptor is to be sent, or is being sent, to the system.

RESULT DESCRIPTOR LPW - STC=15

Indicates that the final word of the result descriptor is being sent to the system and will be followed by the approoriate LPW.

4.1.1 ORS DLP STATUS FLOW (Continued)

SEND DESCRIPTOR LINK - STC=5

Indicates that the descriptor link is to be sent, or is being sent, to the system.

WRITE - STC=8

Indicates that data is being transferred from the system to the DLP.

STC=2

Reserved for expansion. Detection is an error.

PORT BUSY OVERRIDE - STC=10

Indicates, during Wait-For-Port-Busy-not, that a LEM detected an interrupt request and is forcing a system disconnect. Detection of this status from a DLP is an error.

4.1.12 MLI_HANDLER FLOWS

MLI_HANDLER MAIN DLIMESS INTERRUPT ENTRY

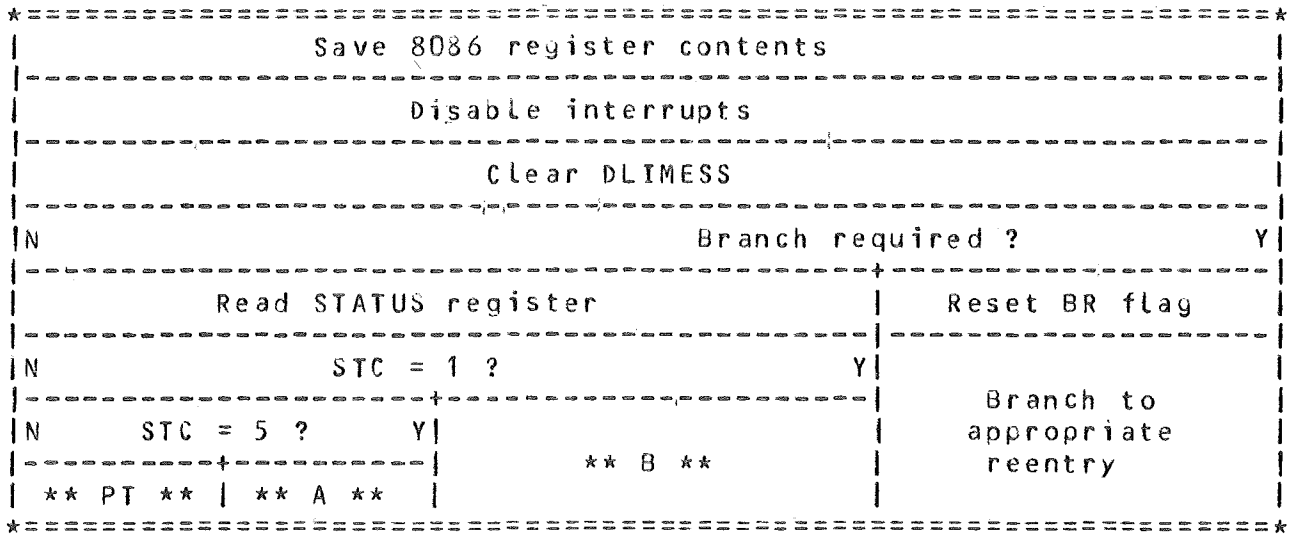
This is the top level of the MLI_HANDLER which branches to different parts of the module depending on the following conditions:

a. BRANCH_FLAG = TRUE
When this flag is on, go to continuation point stored in BRANCH_ENTRY_LABEL.

b. STC = 3
If DLI Status Count seen by the host is 3 (IDLE), go to POLL_TEST routine.

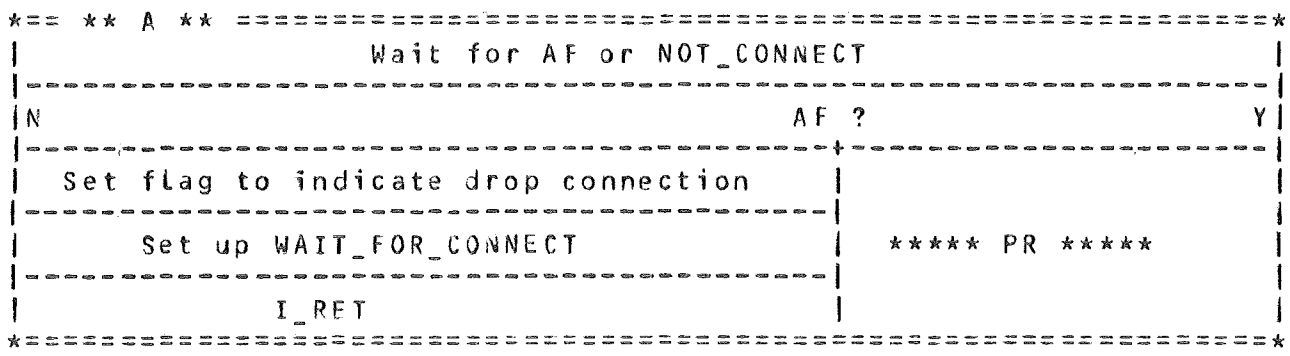
c. STC = 5
If DLI Status Count seen by the host is 5 (RECONNECT), go to POLL_REQUEST routine.

NOTE: STC 5 (RECONNECT) is forced by REQUESTOR module.

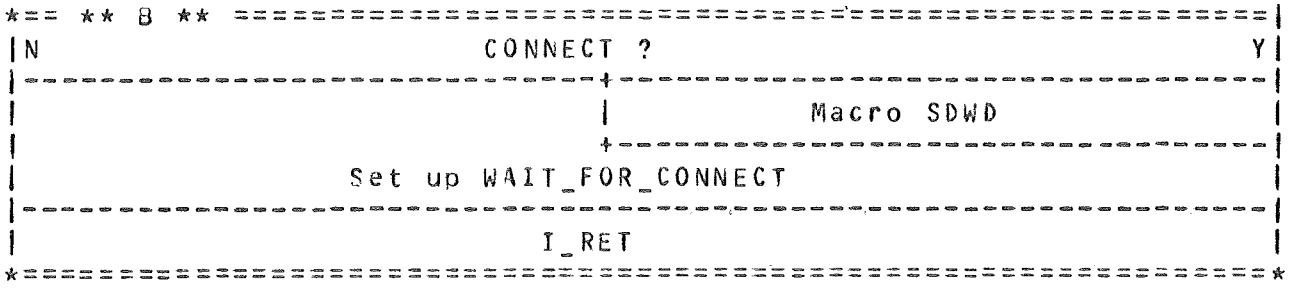


Before entering POLL_REQUEST, check to see if the host changes its mind and drops connection. Also wait for the host acknowledgement of Descriptor_Link_1 sent by REQUESTOR.

4.1.2 MLI_HANDLER FLOWS (Continued)

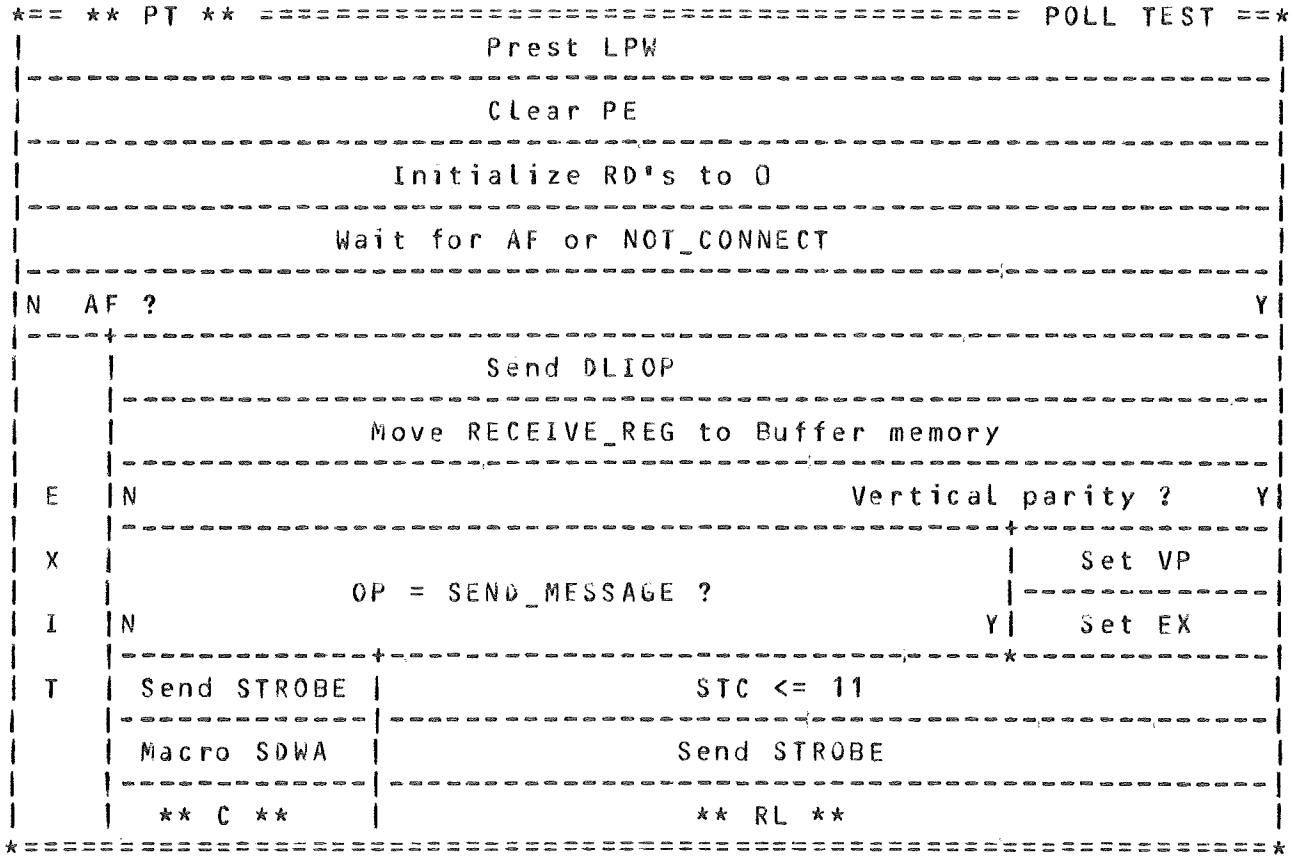


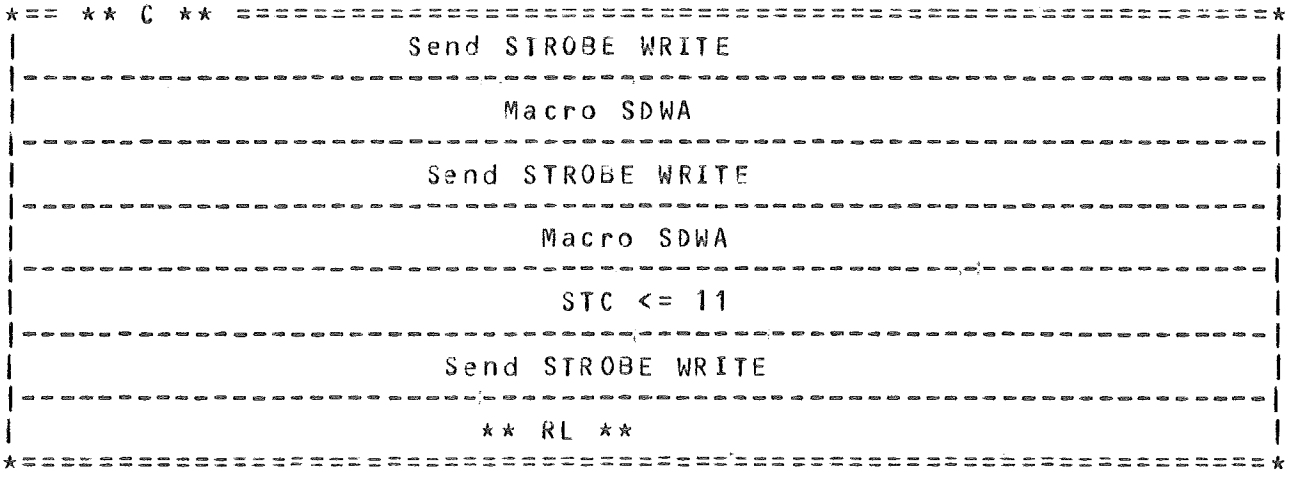
STC 1. Wait until the host disconnects and then wait for Connect (See REQUESTOR module).



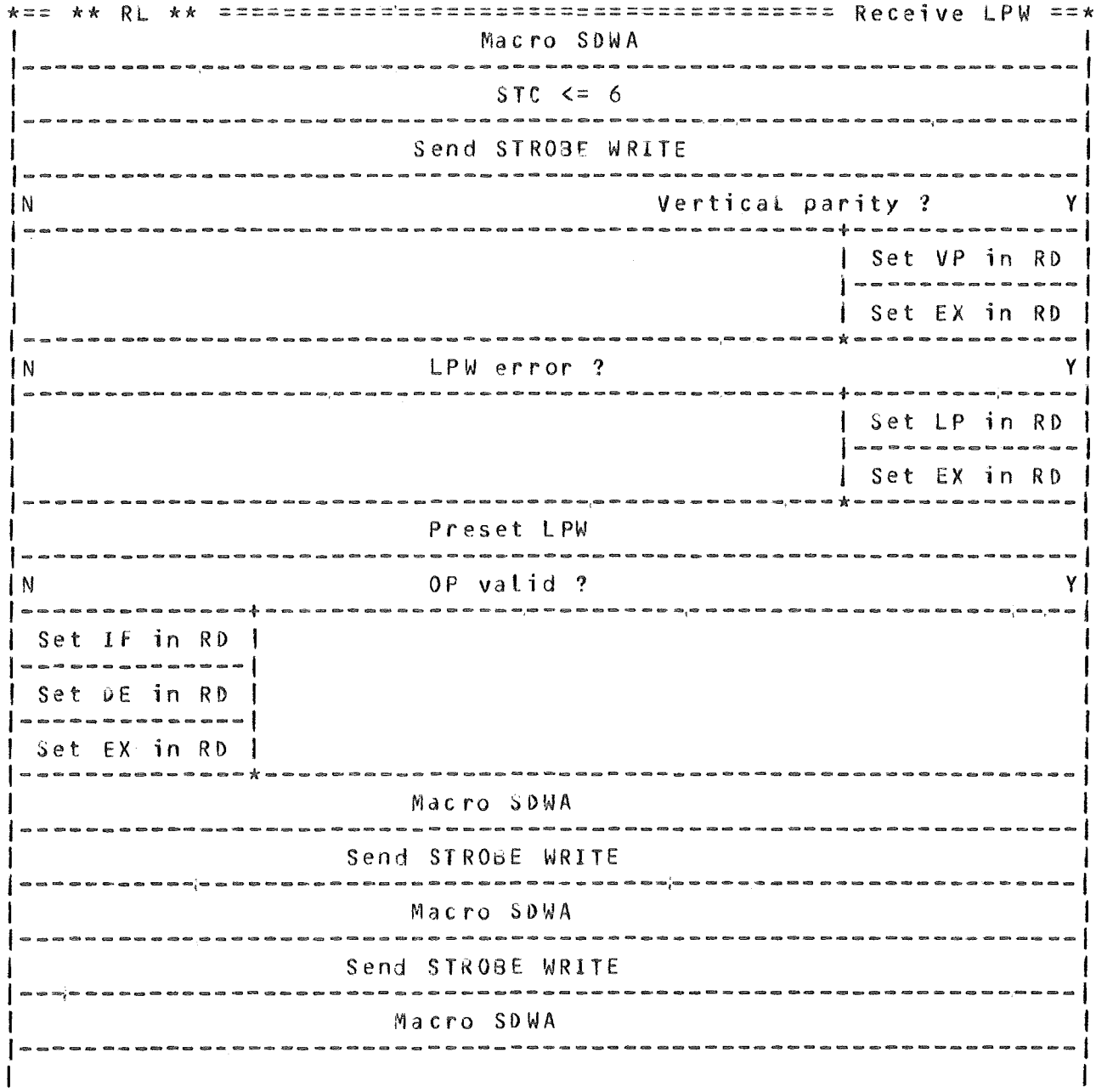
At the beginning of the Poll Test, if AF has not arrived, it waits for AF signal or Disconnection. This is the first strobe that the host sends with I/O Descriptor word 1.

4.1.2 MLI_HANDLER FLOWS (Continued)

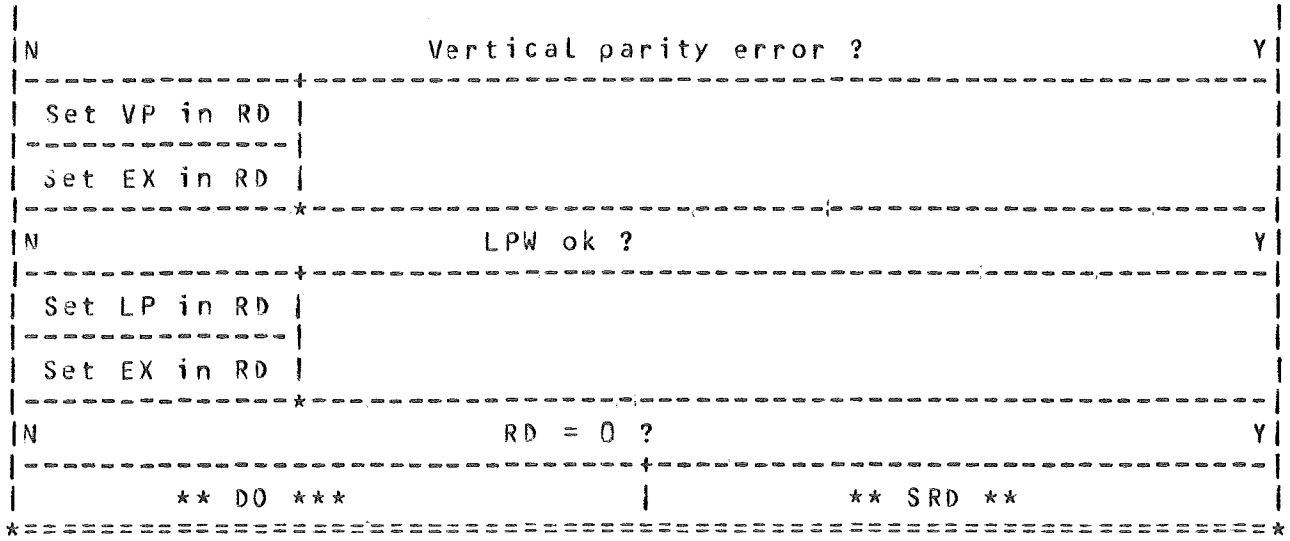




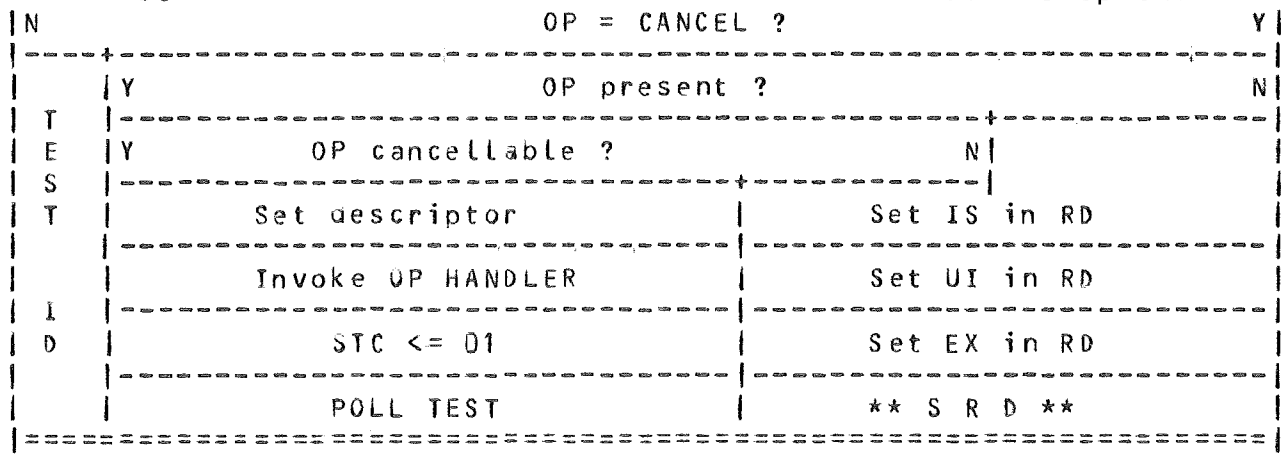
4.1.2 MLI_HANDLER FLOWS (Continued)

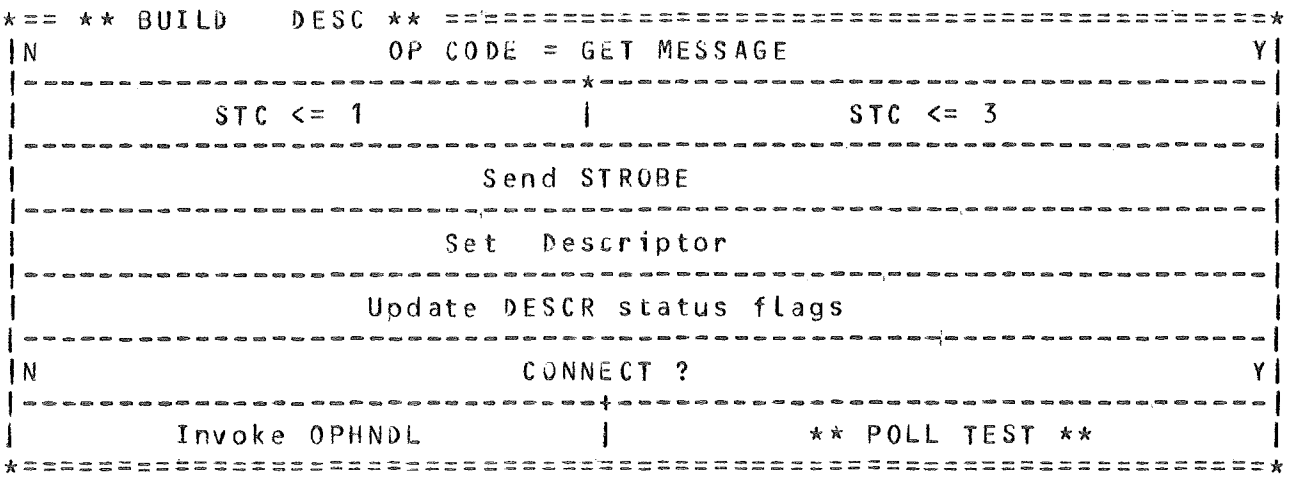


4.1.2 MLI_HANDLER FLOWS (Continued)

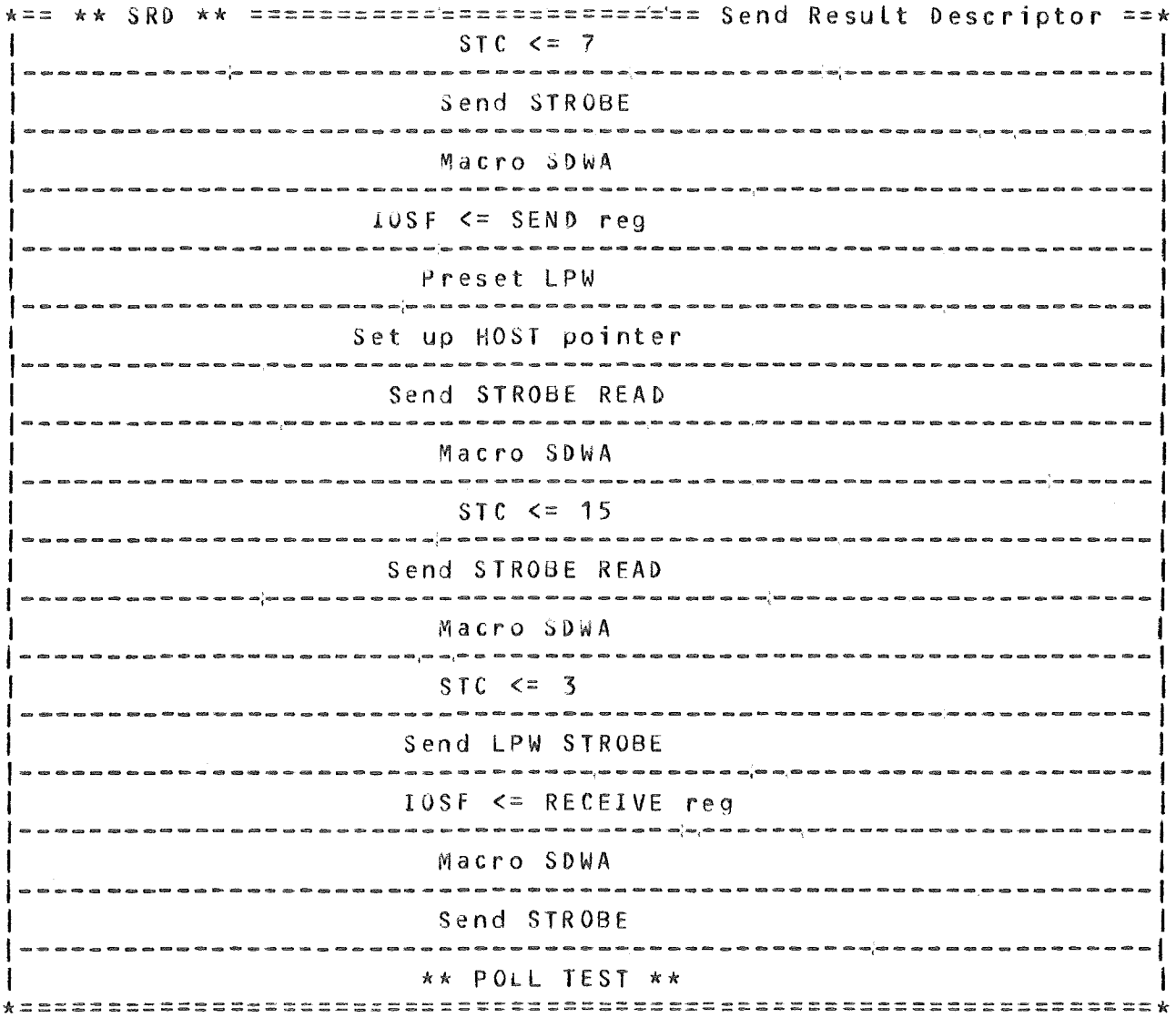


4.1.2 MLI_HANDLER FLOWS (Continued)

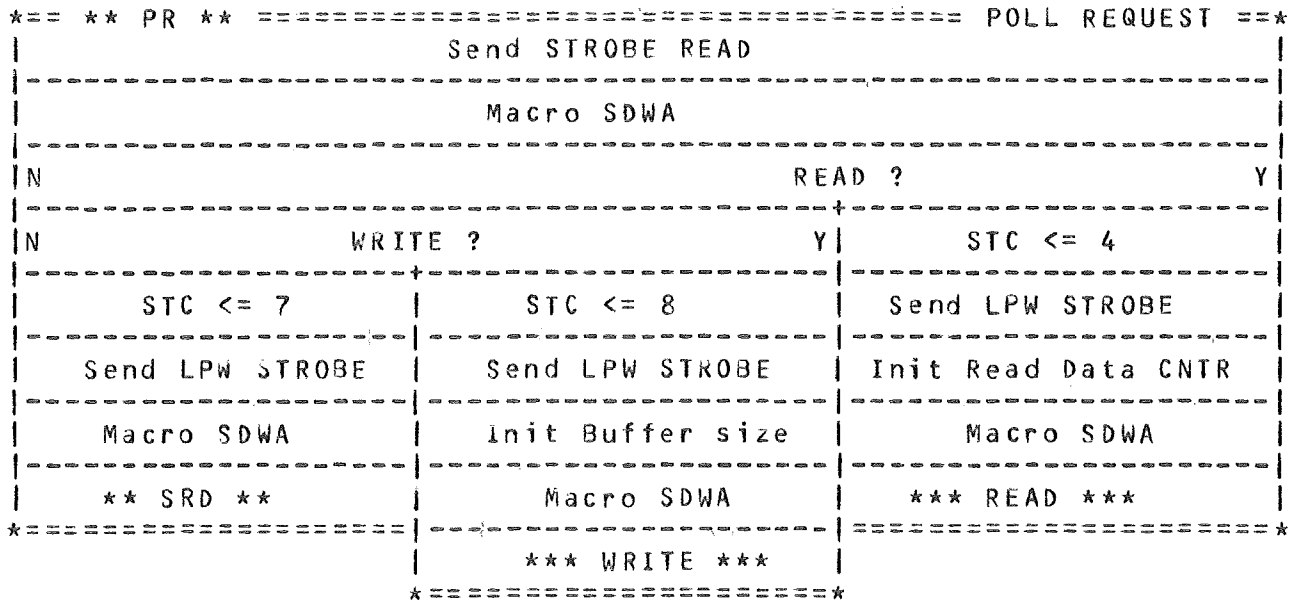


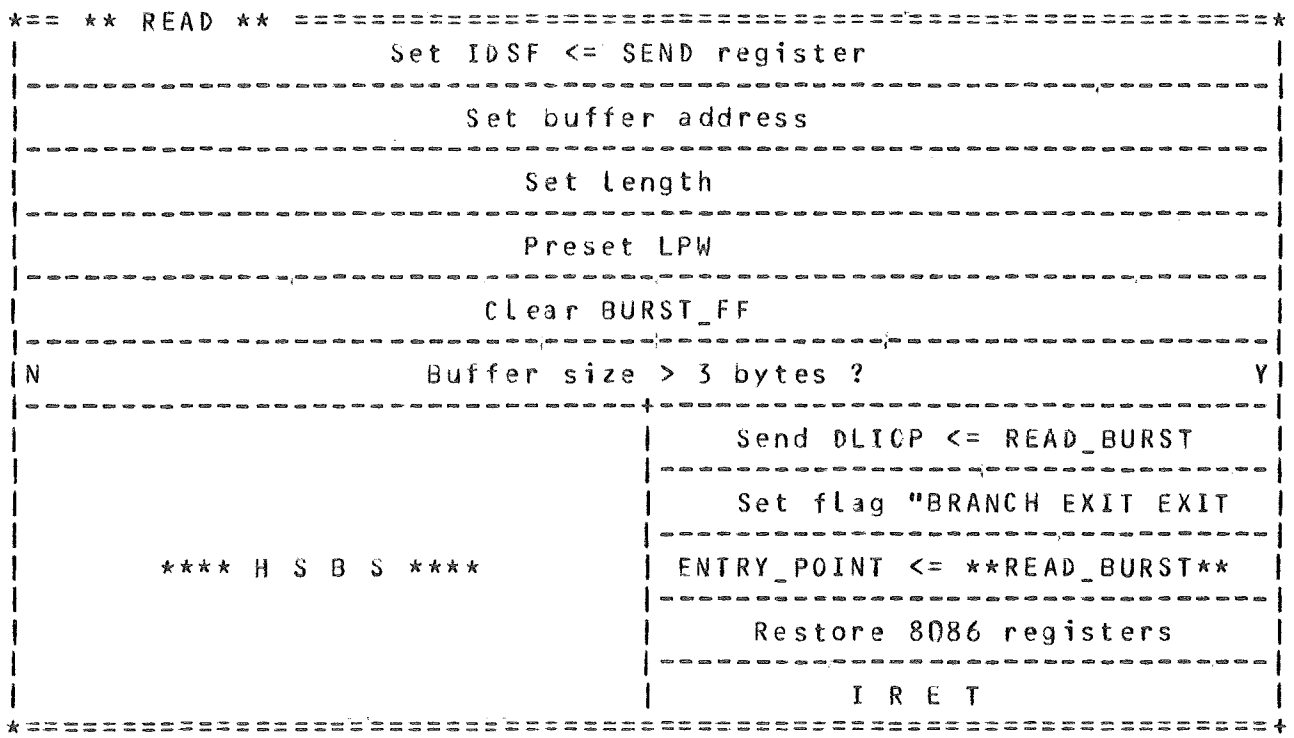


4.1.2 MLI_HANDLER FLOWS (Continued)

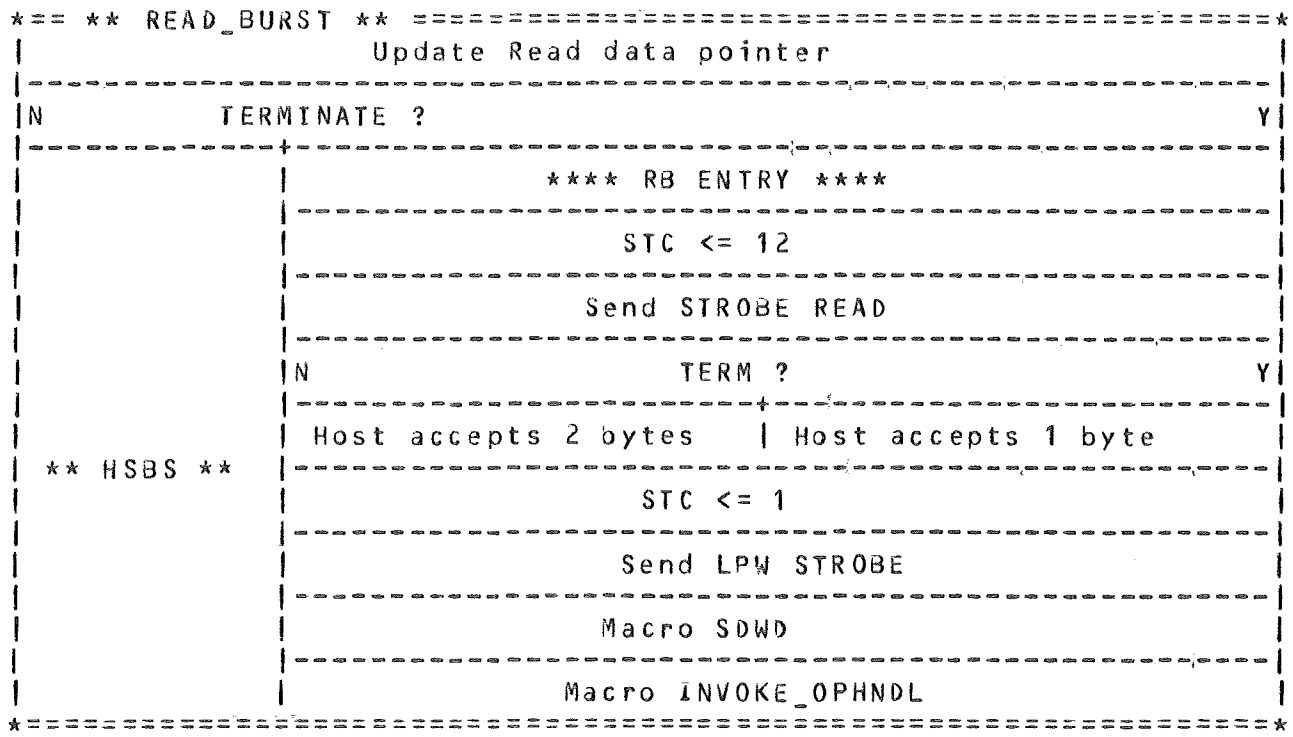


4.1.2 MLI_HANDLER FLOWS (Continued)

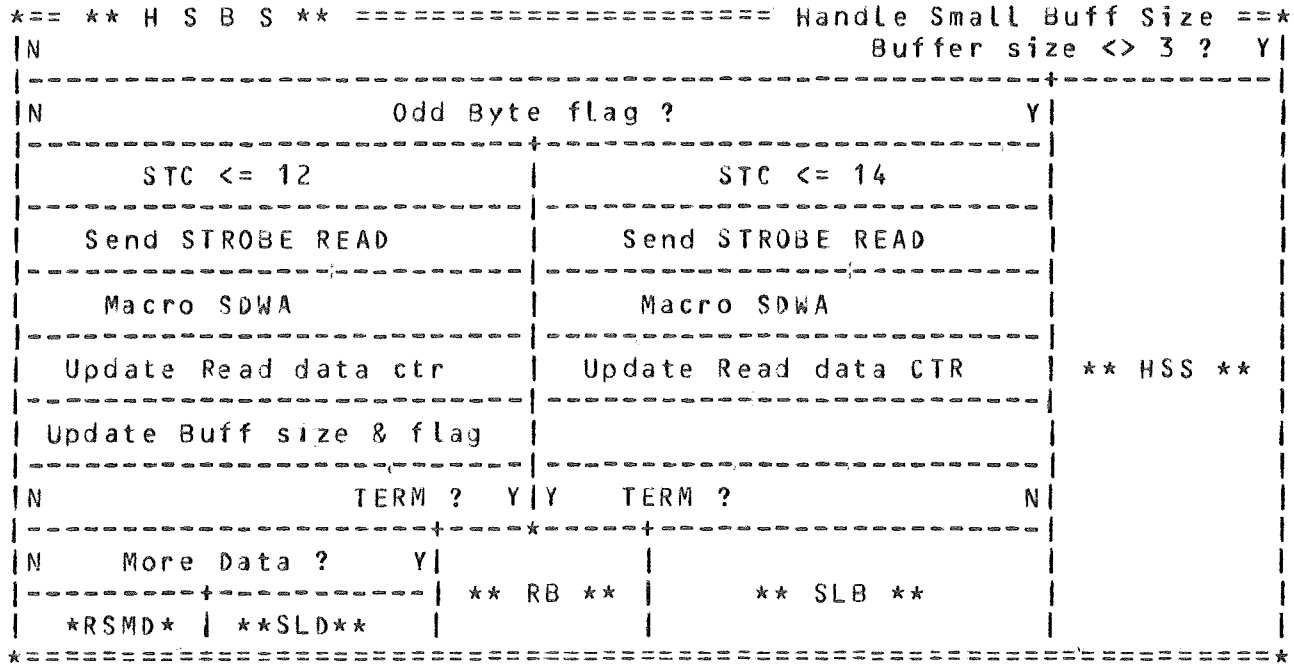


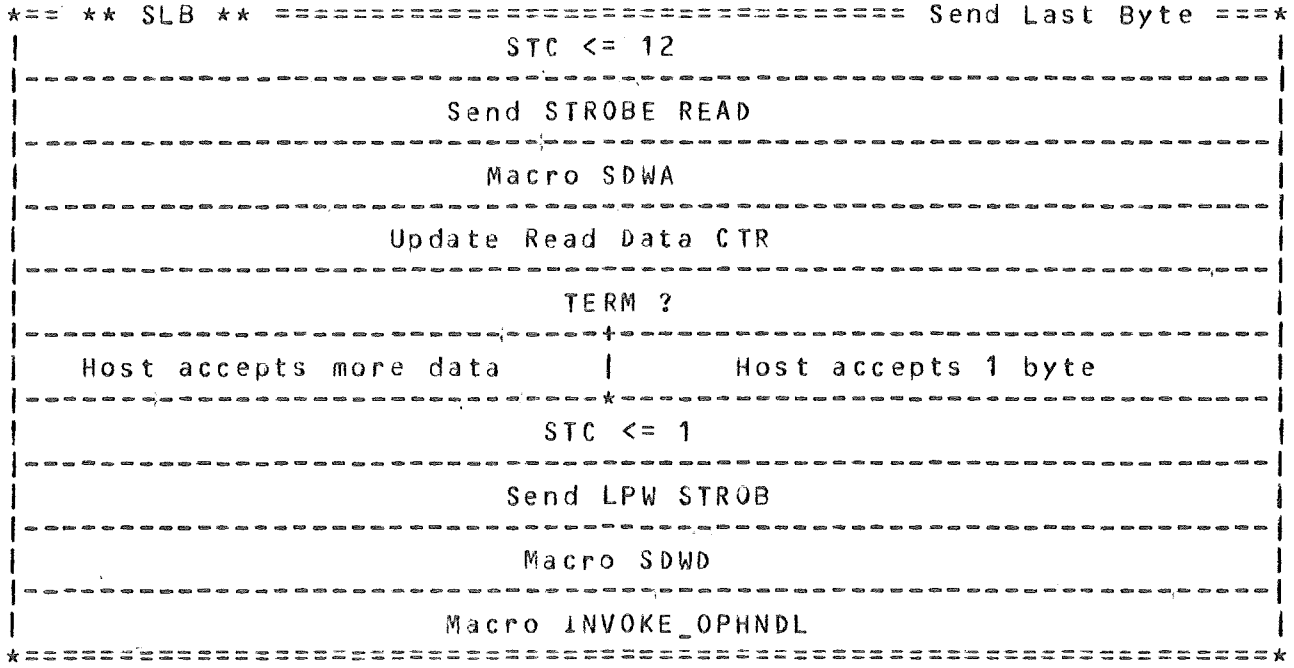


4.1.2 MLI_HANDLER FLOWS (Continued)

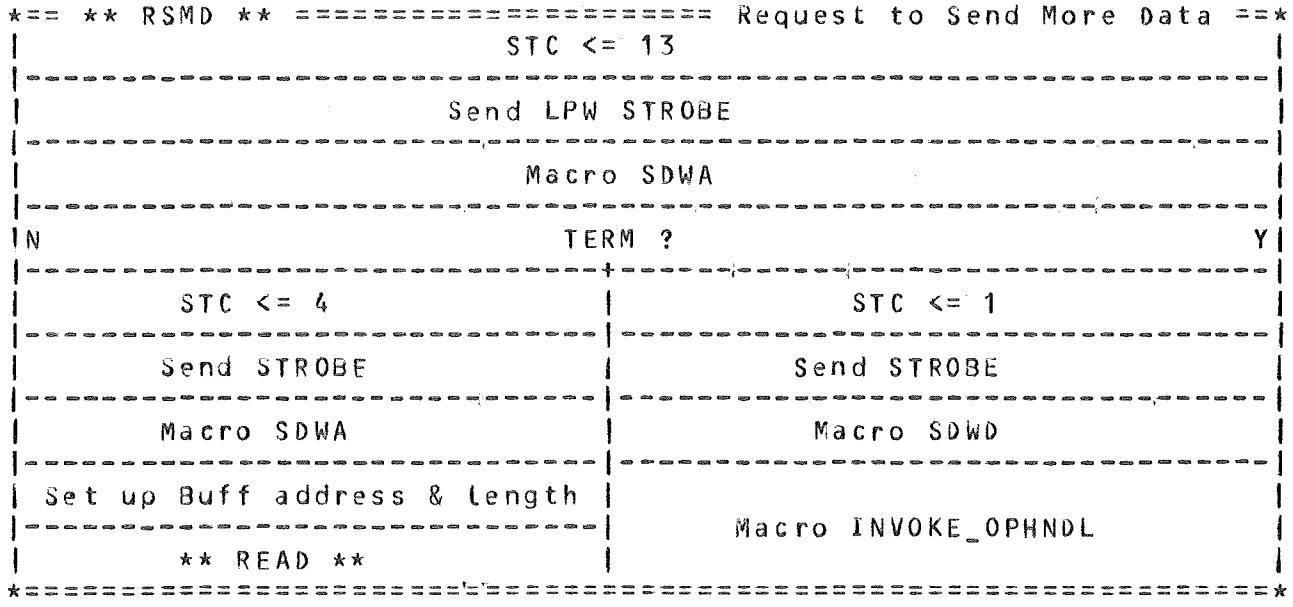


4.1.2 MLI_HANDLER FLOWS (Continued)

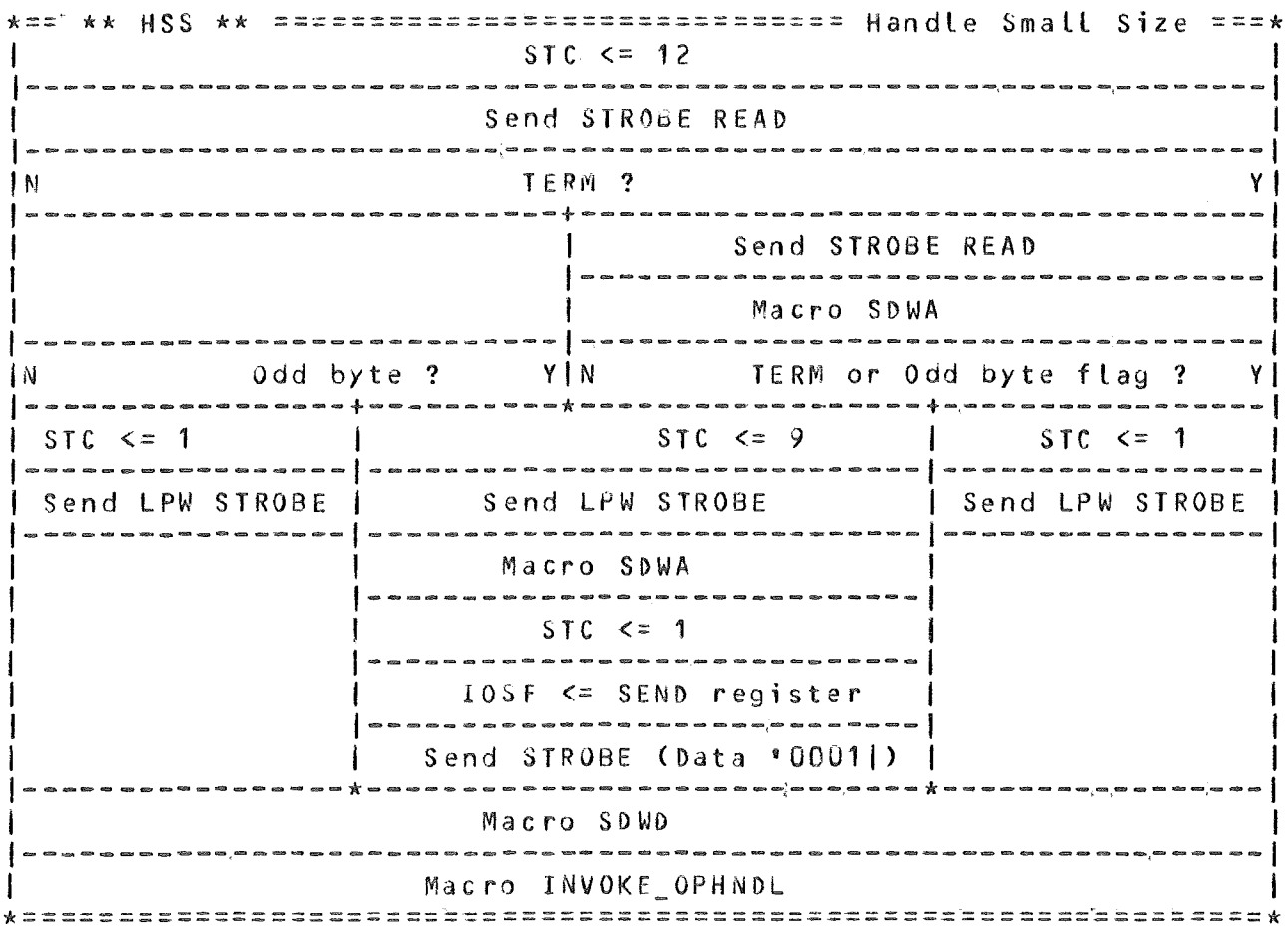




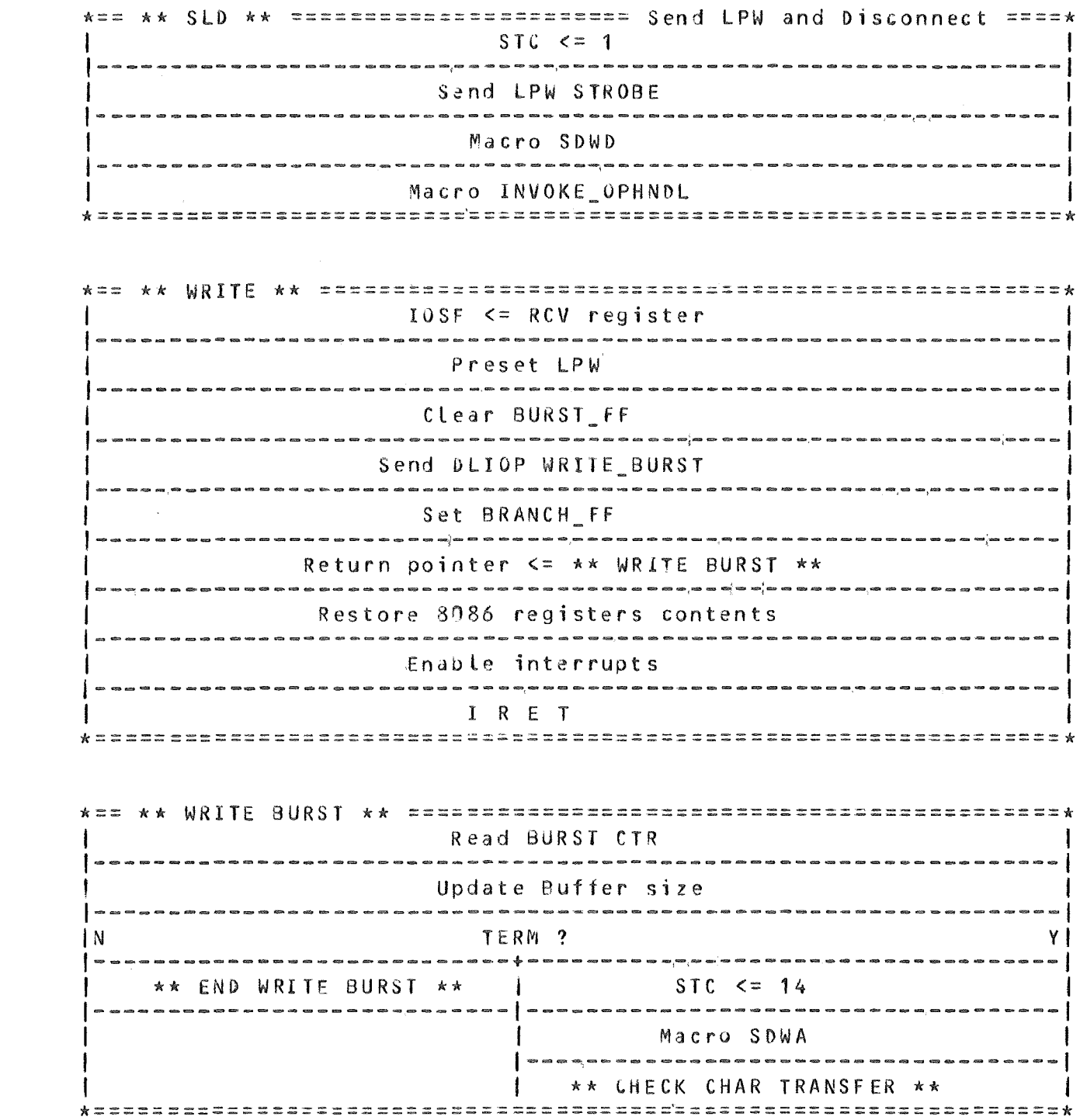
4.1.2 MLI_HANDLER FLOWS (Continued)



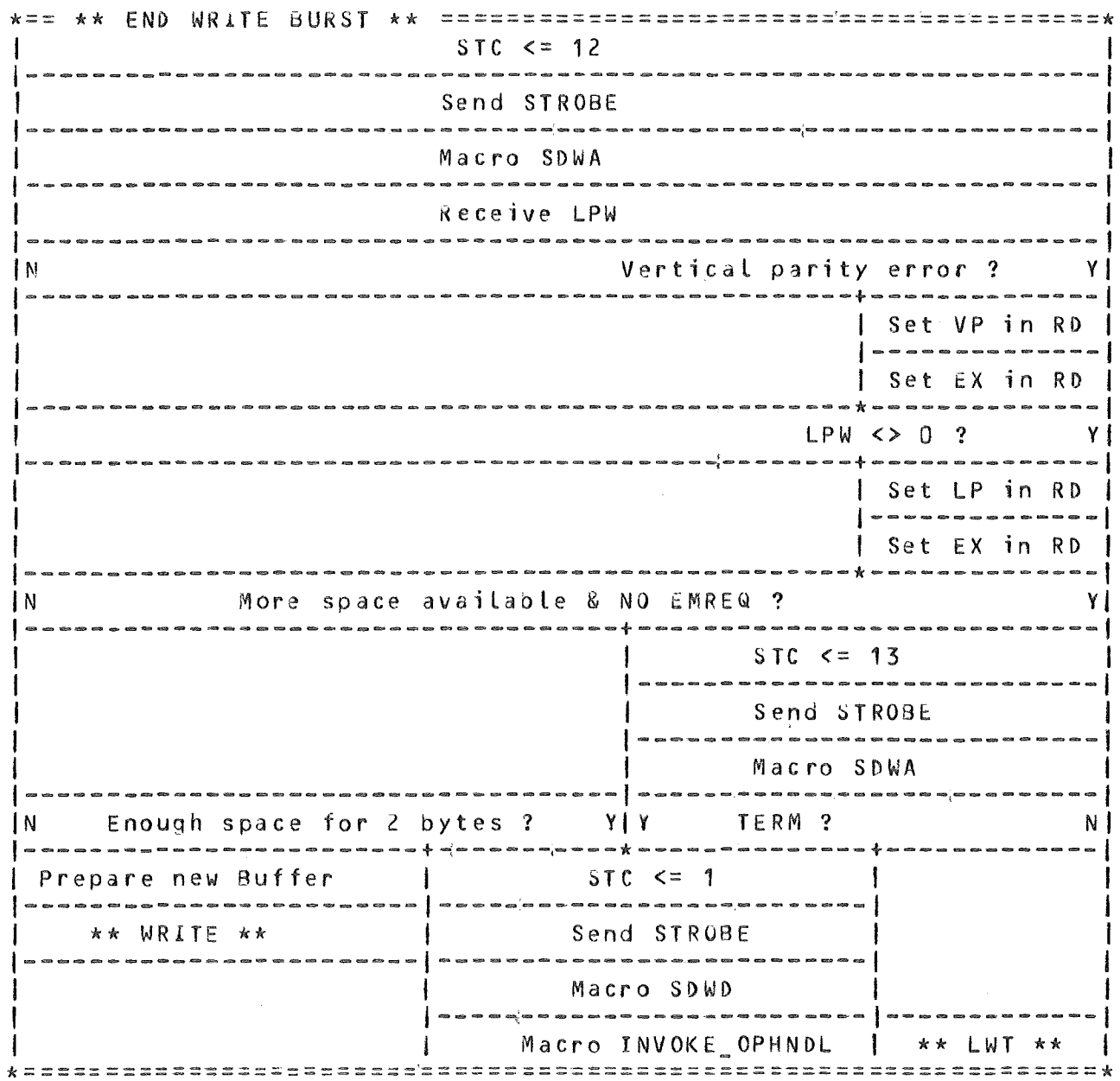
4.1.2 MLI_HANDLER FLOWS (Continued)



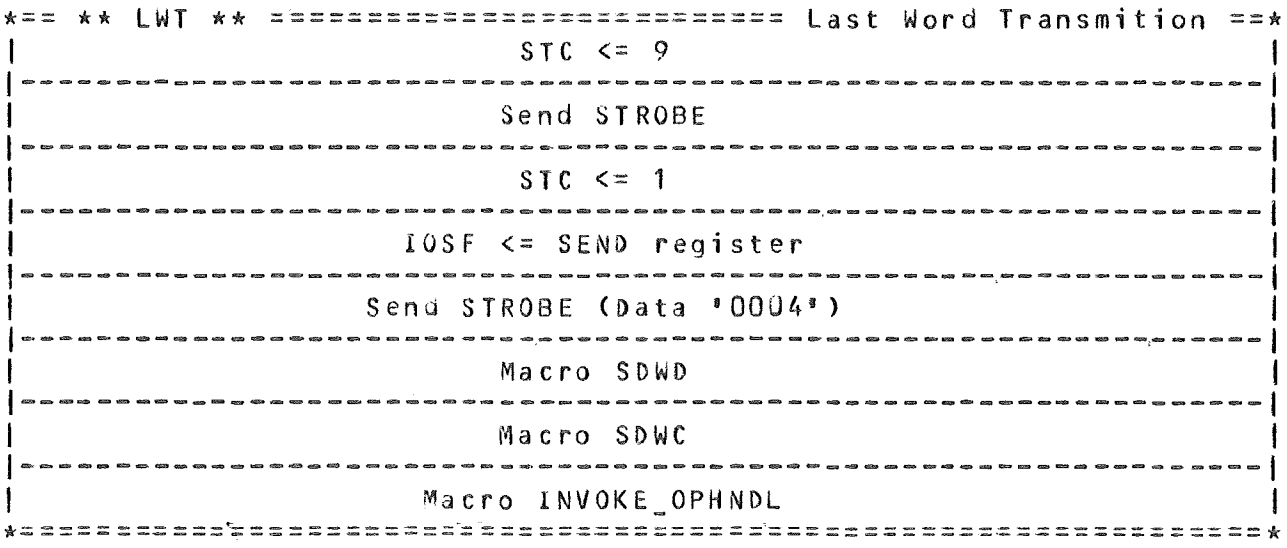
4.1.2 MLI_HANDLER FLOWS (Continued)



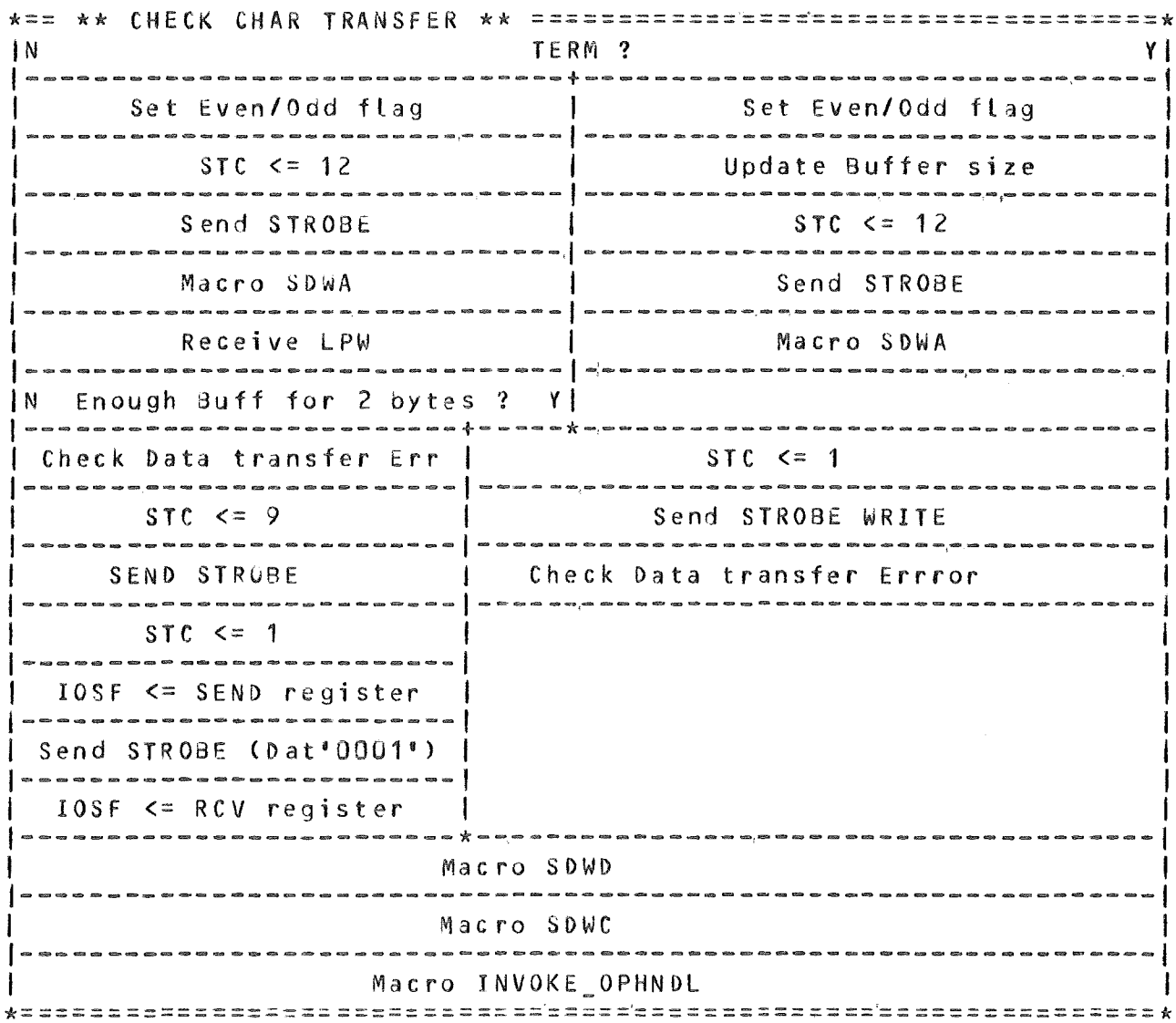
4.1.2 MLI_HANDLER FLOWS (Continued)



4.1.2 MLI_HANDLER FLOWS (Continued)



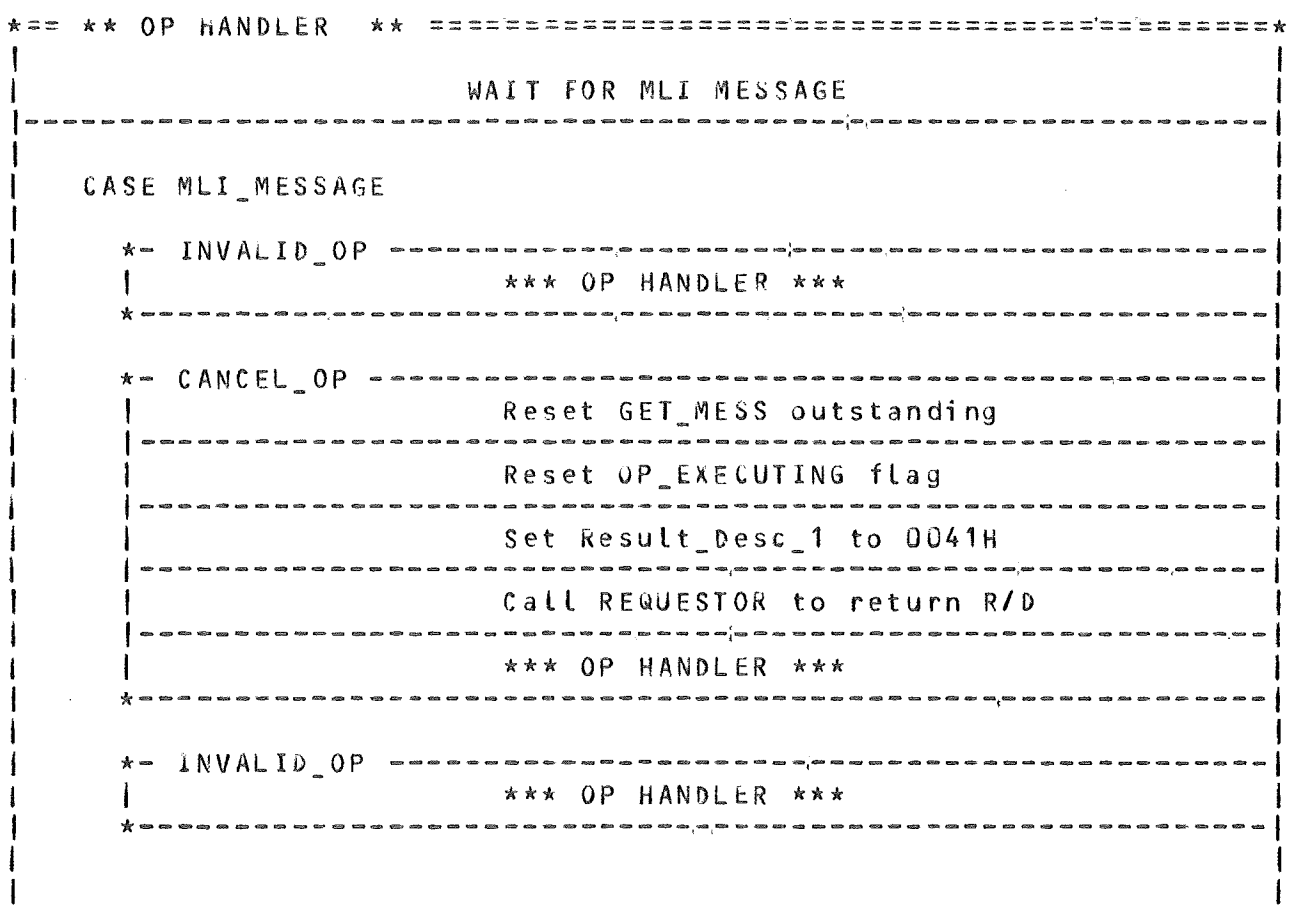
4.1.2 MLI_HANDLER FLOWS (Continued)



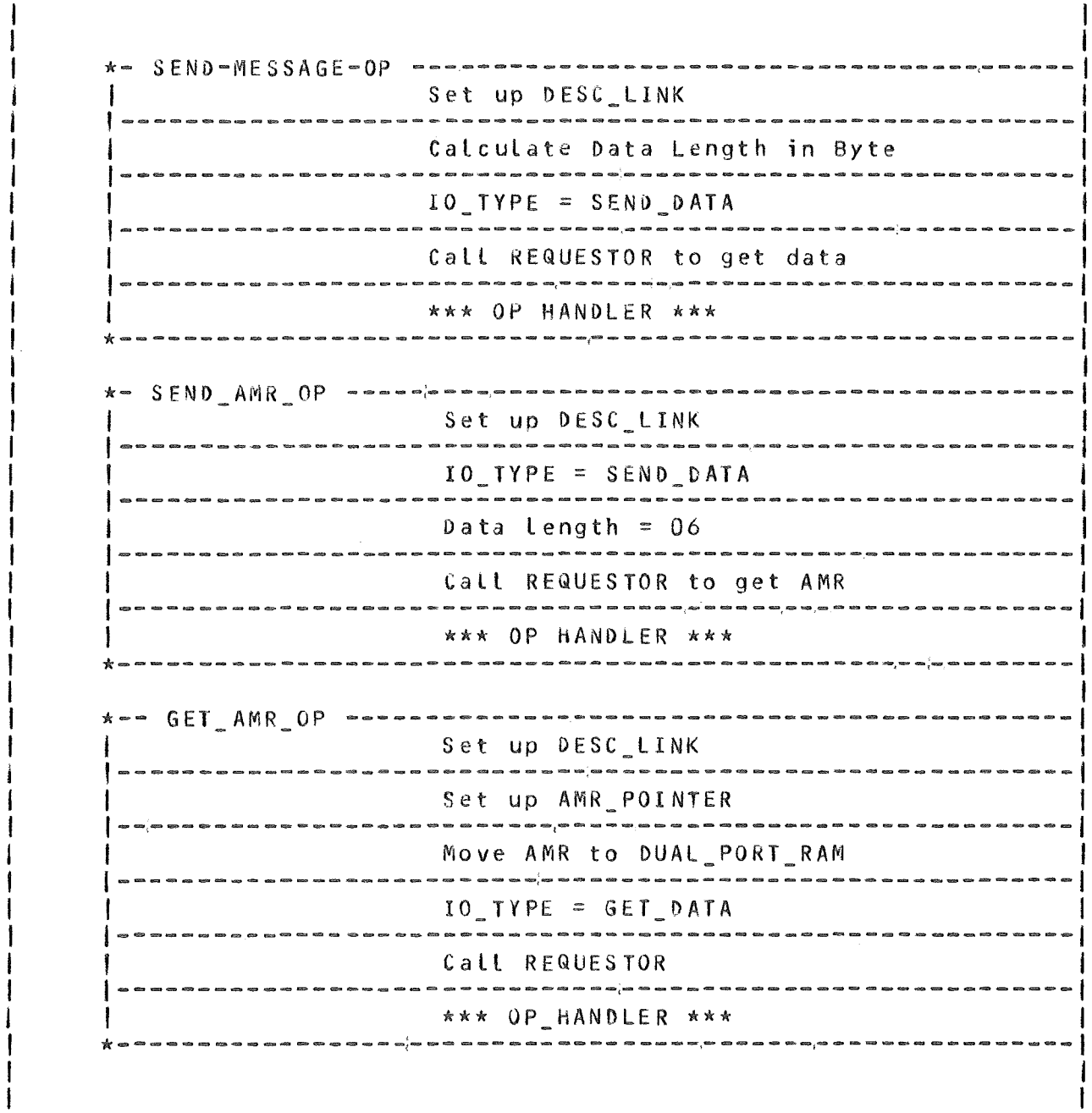
4.2 OPHNDLR FLOW

OP HANDLR - OPERATION HANDLER is a task that is designed to assist the MLI HANDLER to conform to the MLI protocol. This task manages an I/O descriptor level protocol and the ISC level protocol. The following diagram describes the functional flow of the OP_HANDLER task.

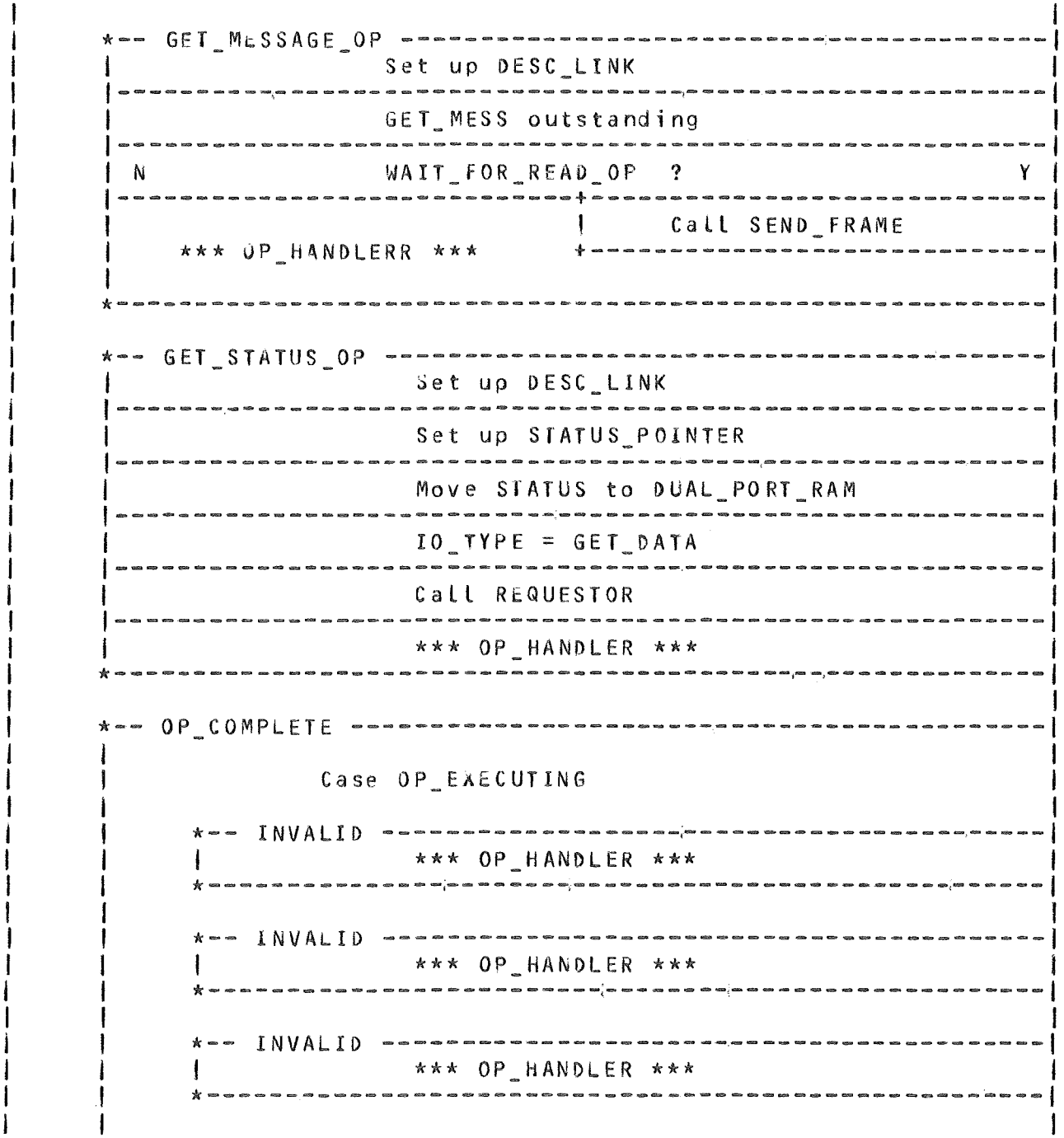
OPHANDLER FLOW



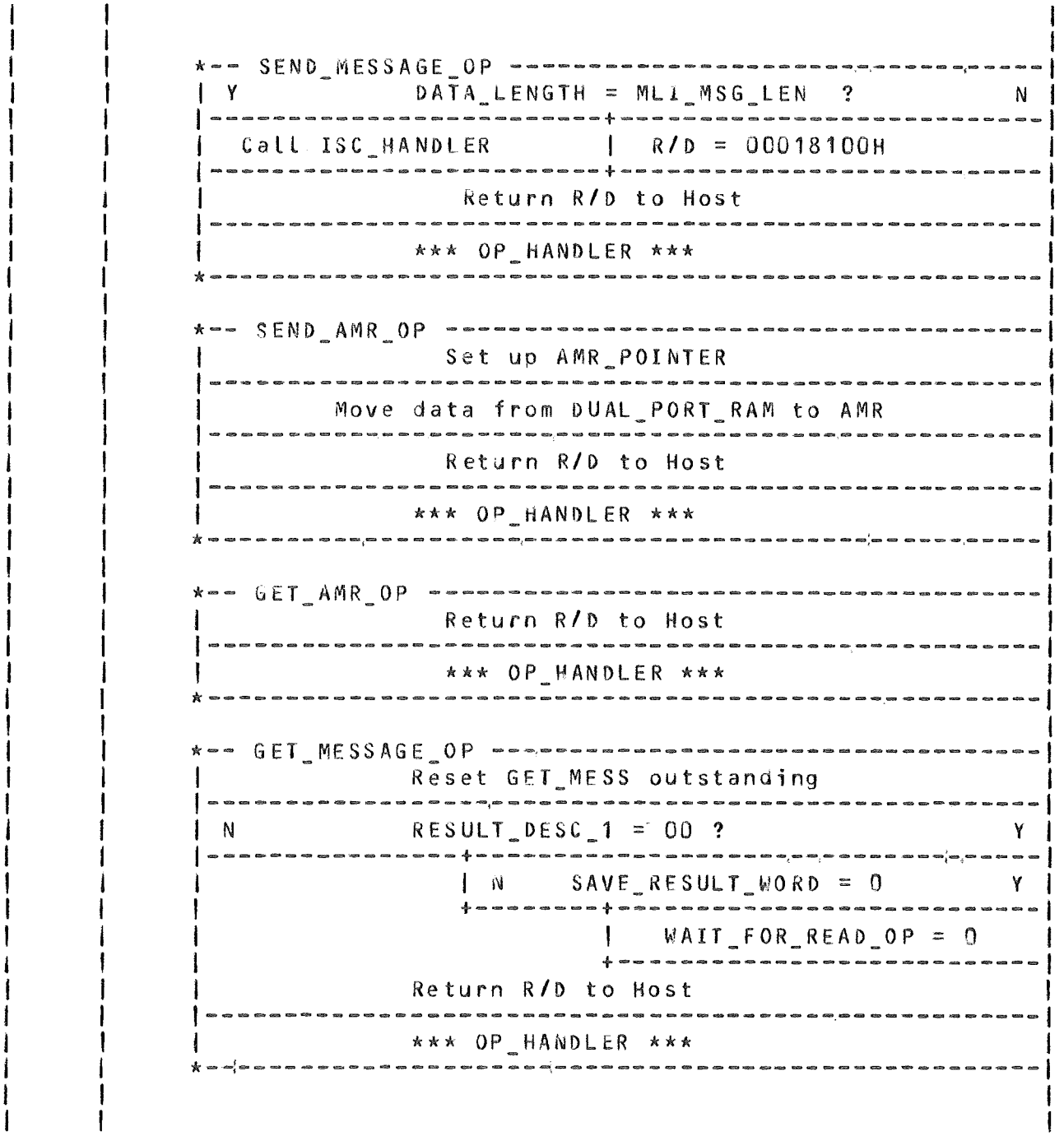
4.2 OPHNDLR FLOW (Continued)



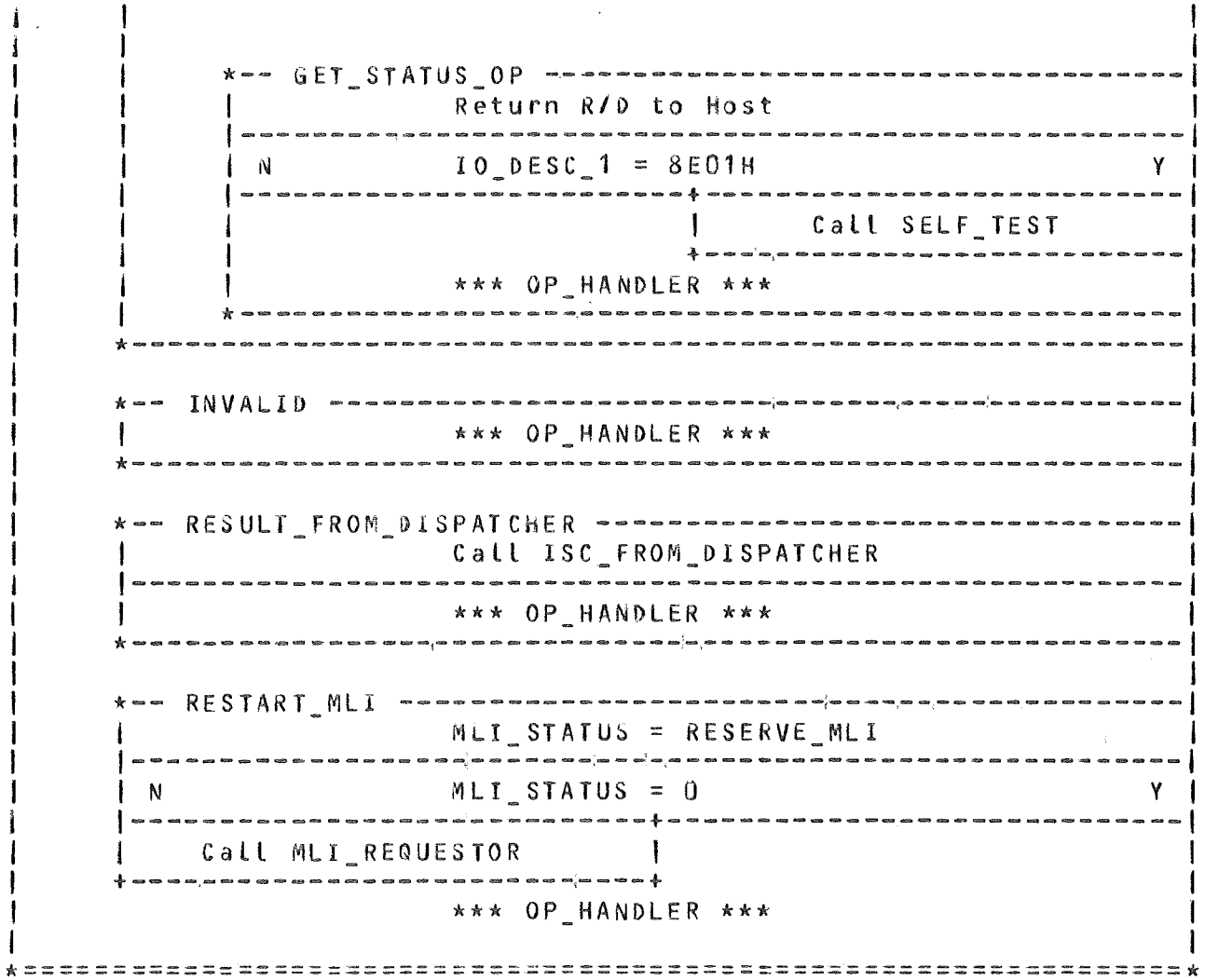
4.2 OPHNDLR FLOW (Continued)



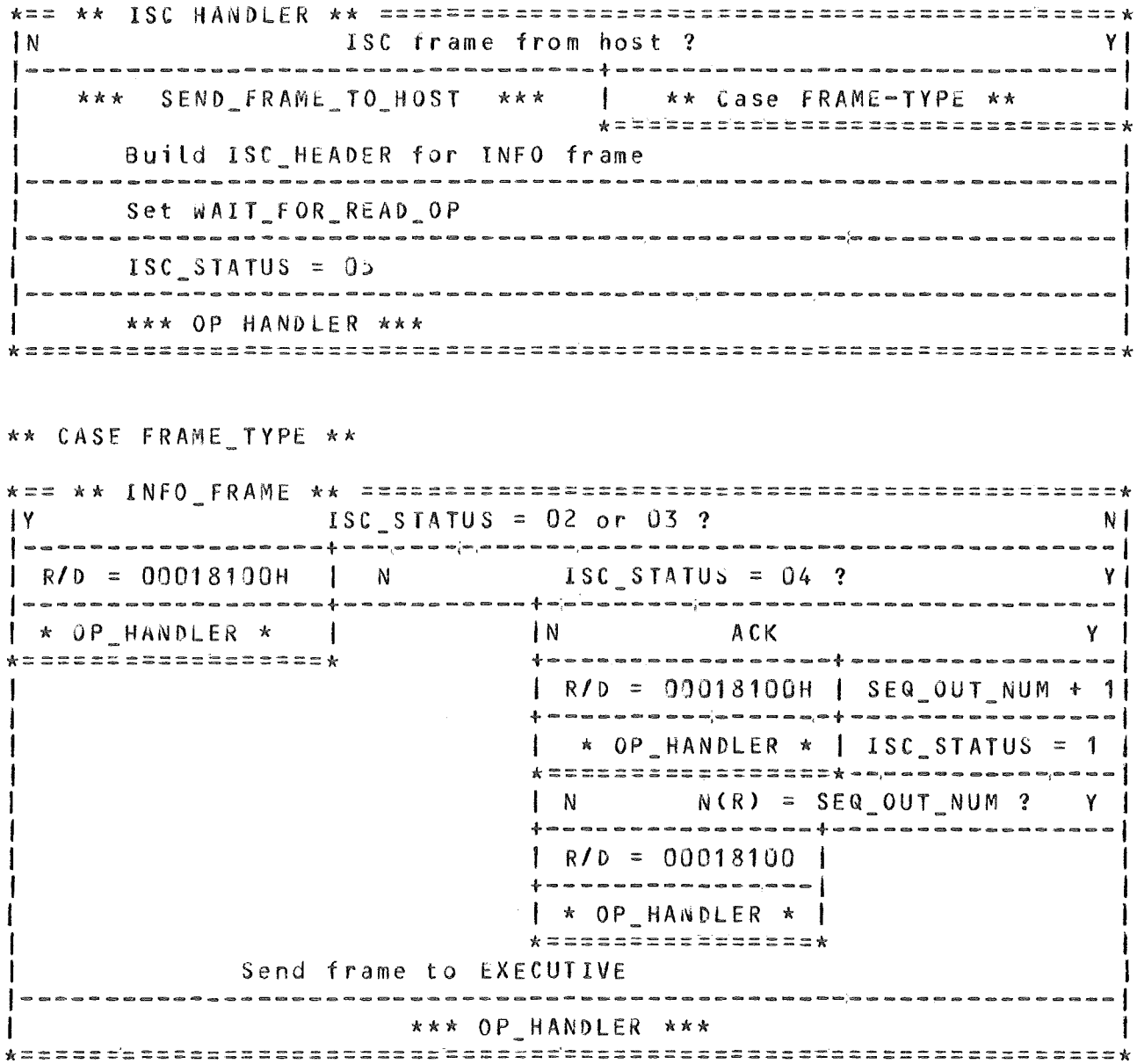
4.2 OPHNDLR FLOW (Continued)



4.2 OPHNDLR FLOW (Continued)

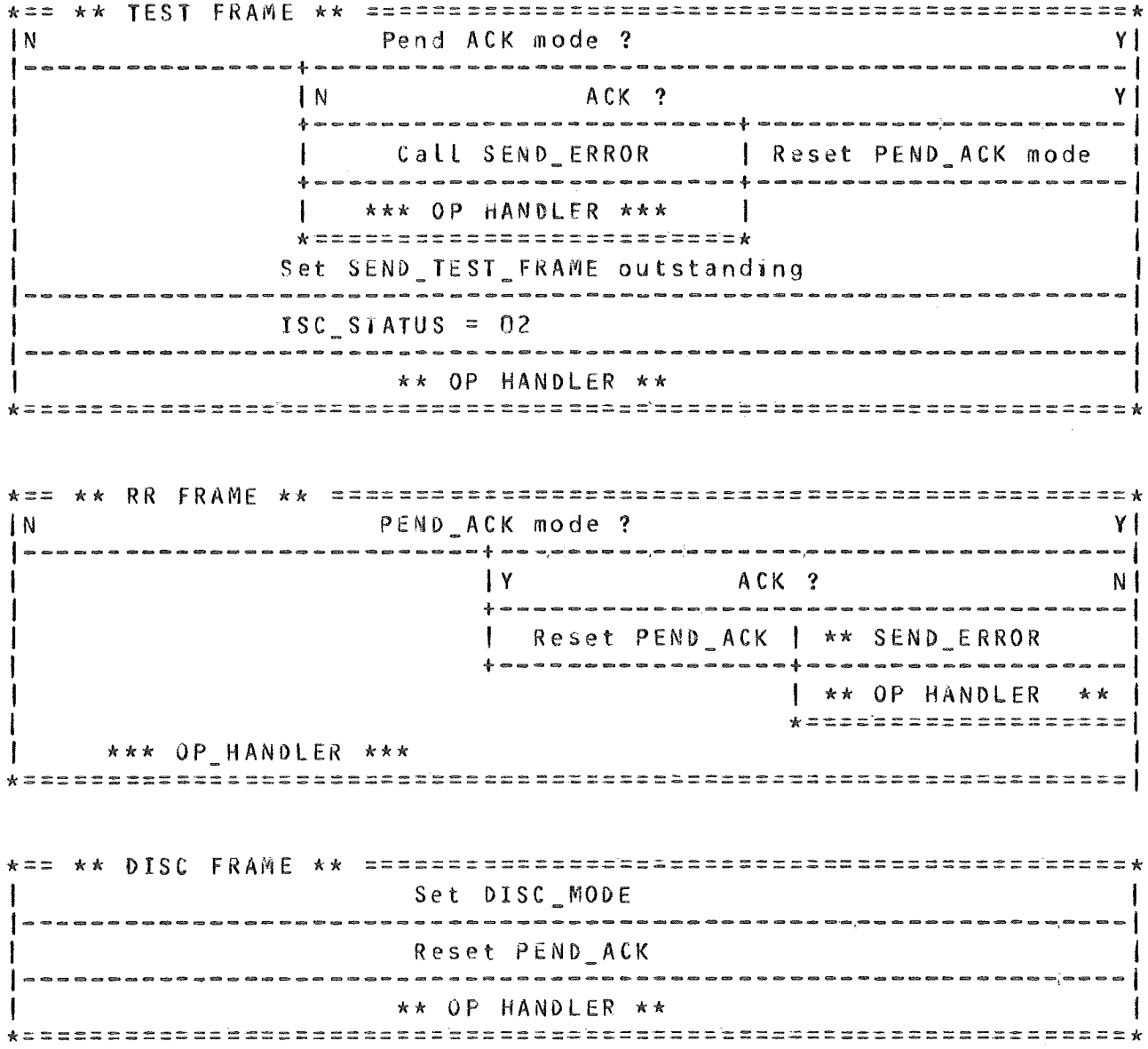


4.2 OPHNDLR FLOW (Continued)

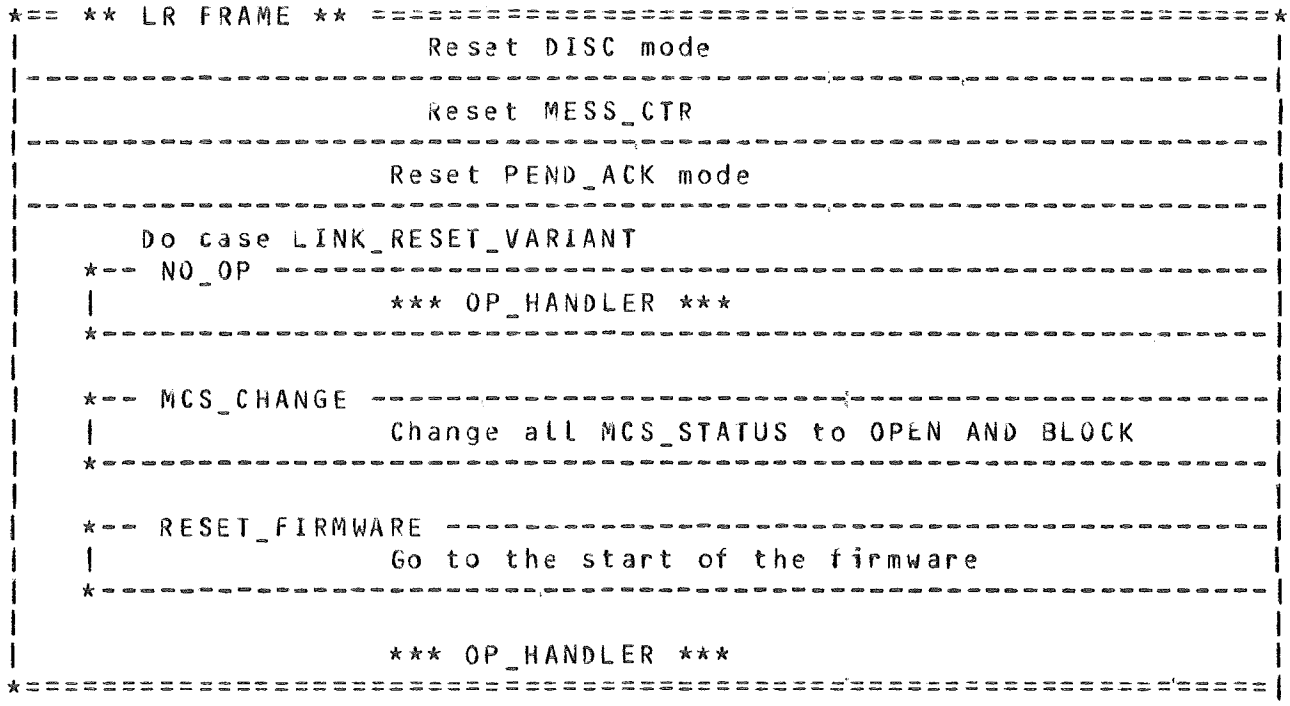


** CASE FRAME_TYPE **

4.2 OPHNDLR FLOW (Continued)



4.2 OPHNDLR FLOW (Continued)



4.3 EXECUTIVE

The EXECUTIVE module is the router of the subsystem. Its functions are to route DCP messages from/to Line Processes and to handle some of the DCP messages. The EXECUTIVE module consists of two tasks; ROUTER task and DISPATCHER task. The ROUTER task will get invoked when a MESSAGE_QUEUE receives a message from OP HANDLER. The DISPATCHER task will get invoked when a RESULT_QUEUE receives a message from either LINE PROCESS or ROUTER. The ROUTER module has the lowest priority in the subsystem and one of its major functions is to route the DCP message from the OP_HANDLER to the Station Output Queues. The word 9 [4:1] bit in a DCP Header will determine the position within the Station Output Queue; 0 - bottom queue, 1 - top queue. Some of the DCP Headers have to be performed immediately or the function that has to be performed is for a suspended Line Process. These messages are handled entirely by the ROUTER. The messages are:

1. Enableinput (33)
2. Make Station Ready (34)
3. Recall (40)
4. Acknowledge (43)
5. Make Line Ready (47)
6. Make Line Not Ready (48)
7. Dialout (49)
8. Disconnect (50)
9. Answer the Phone (51)
10. Interrogate Line Status (52)
11. Set Autoanswer (53)
12. Reset Autoanswer (54)
13. Memory Dump (61)
14. Load (71)
15. Echo (72)

The DISPATCHER task waits for a message in the RESULT_QUEUE and attempts to send it to the host. The EXECUTIVE PROCEDURES section describes the way the EXECUTIVE handles the messages.

4.3.1 EXECUTIVE DATA STRUCTURE

The EXECUTIVE starts execution of a LINE_PROCESS task and passes a pointer of the first element in the LINE_TABLE through a line mailbox. From the LINE_TABLE data structure, the LINE_PROCESS is capable of finding all the station data structure attached to this line.

The EXECUTIVE procedures described in section 4.3.2 are provided to use and share data structure described in the next section.

4.3.1.1 LINE_TABLE VARIABLES

```
DECLARE
     CURRENT_LINE_PTR                    POINTER;
DECLARE
     LINE   BASED CURRENT_LINE_PTR
     STRUCTURE(
          NUMBER                              BYTE,
          TYPE                                BYTE,
          BAUD_RATE                           WORD,
          RECEIVE_DELAY                       WORD,
          TRANSMIT_DELAY                      WORD,
          INITIAL_TIME_OUT                    WORD,
          TEXT_TIME_OUT                       WORD,
          VERTICAL_PARITY                     BYTE,
          STOP_BITS                           BYTE,
          NEW_SYNC                            BYTE,
          BREAK_ENABLE                        BYTE,
          CONTINUOUS_CARIER                   BYTE,
          CCITT                               BYTE,
          SELECT_RATE                         BYTE,
          SELECT_STANDBY                      BYTE,
          DIAL_IN                             BYTE,
          DIAL_OUT                            BYTE,
          ADDRESS                        (4)  BYTE,
          CPCLOT                              BYTE,
          PROTOCOL                            BYTE,
          PROTOCOL_PTR                        POINTER,
          FIRST_STATION_PTR                   POINTER,
```

4.3.1.1 LINE_TABLE VARIABLES (Continued)

```
CURRENT_STATION_PTR              POINTER,
BUFF_IN_PTR                      POINTER,
CNTRL_BUFF_PTR                   POINTER,
TASK_ID                          BYTE,
DC_VECTOR                        BYTE,
USART_PORT                       WORD,
READY                            BYTE,
BUSY                             BYTE,
QUEUED                           BYTE,
FILLER_A                         BYTE,
CONDITION                        WORD,
ADAPTER_STATUS                   BYTE,
FILLER_B                         BYTE,
EXCEPTION_A                      WORD,
EXCEPTION_B                      WORD,
DISCONNECT_PENDING               BYTE,
ANSWER_PHONE                     BYTE,
CRC                              WORD,
TIME_OUT                         WORD,
TALLY                      (10)  BYTE,
BLOCK_TABLE_PTR                  POINTER);
```

4.3.1.1 LINE_TABLE VARIABLES (Continued)

NUMBER

This byte contains binary Line number.

TYPE

This byte indicates the type of line. See table below.

Line Type	Description
00	Invalid
01	TTY
02	TDI
03	BDAA
04	Async, RS232, Leased
05	Async, RS232, Leased, C.C
06	Async, RS232, Switched
07	Async, RS232, Switched, C.C
08	Sync, RS232, Leased
09	Sync, RS232, Leased, C.C
0A	Sync, RS232, Switched
0B	Sync, RS232, Switched, C.C
0C	Sync, RS449, Leased
0D	Sync, RS449, Leased, C.C
0E	Sync, RS449, Switched
0F	Sync, RS449, Switched, C.C

BAUD_RATE

In Asyncronous operation, this word indicates the clock divisor to be used by the Line Process for timer initializations. The clock divisor values for the desired baud rates are given below:

4.3.1.1 LINE_TABLE VARIABLES (Continued)

RATE	CONSTANT
50	9C3E
75	6828
110	4703
150	3414
300	1A08
600	D03
1200	680
1800	455
2000	3E6
2400	33F
3600	229
4800	19E
7200	113
9600	CE
19200	66

RECEIVE_DELAY

This word contains the Receive Delay value in milliseconds.

TRANSMIT_DELAY

This word contains the Transmit Delay value in milliseconds.

INITIAL_TIME_OUT

This word contains the Initial Time Out value in milliseconds.

TEXT_TIME_OUT

This word contains the Text Time Out value in milliseconds.

4.3.1.1 LINE_TABLE VARIABLES (Continued)

VERTICAL_PARITY

This field specifies the type of vertical parity to be used according to:

Sync = 0 - Even
1 - Odd

Async = 0 - Odd
1 - Even

Vertical parity testing and generation is suppressed if either the protocol used is Binary Syncronous or any Bit Oriented.

STOP_BITS

This field contains the number of Stop Bits to be transmitted.

00 = reserved
01 = 1 bit
10 = 1.5 bits
11 = 2 bits

NEW_SYNC

This byte contains the New Sync boolean value. If set, it allows the Line Process to use the AUX.1 driver to control the New Sync lead of a data set such as WE201 or TA714 for minimum re-sync time in receive mode.

BREAK_ENABLE

This byte contains the Break Enable boolean value. If this byte is set, the Line Process is enabled to interpret 15 ms of received spacing while in write as a break and terminate the write.

4.3.1.M LINE_TABLE VARIABLES (Continued)

CONTINUOUS_CARIER

This byte indicates that the CARRIER signal is to be continuously held high.

CCITT

This parameter allows the Line Process to control the Select Rate and Select Standoy leads according to the SELECT_RATE and SELECT_STANDBY parameters.

SELECT_RATE

When true, it will indicate to the Line Process to set the Select Rate. CCITT parameter must be set.

SELECT_STANDBY

When true, it will indicate to the Line Process to set the Select Standby. CCITT parameter must be set.

DIAL_IN

When true, it indicates that the this line is capable of Autoanswer.

DIAL_OUT

When true, it indicates that the this line is capable of Autodial.

ADDR

This field contains the address of this line.

CPCLOT

When true, it indicates to the Line Process to enable clock element to the paddle card.

4.3.1.1 LINE_TABLE VARIABLES (Continued)

PROTOCOL

This byte contains the Protocol number used on this Line.

PROTOCOL_PTR

This field contains a pointer of the Line Process task for this line.

FIRST_STATION_PTR

This field contains a pointer of the first Station Table.

CURRENT_STATION_PTR

This field contains a pointer af the station table currently in use.

BUFF_IN_PTR

This field contains a pointer of the Receive Buffer.

CNTRL_BUFF_PTR

This field contains a pointer of the Control Buffer.

TASK_ID

This field contains the Task Identification of the Line Process.

DC_VECTOR

This field contains the datacomm interrupt vector associated with this line.

USART_PORT

This field contains a Port number of the USART to be used on this line.

4.3.1.1 LINE_TABLE VARIABLES (Continued)

READY

This field determines the readyness of this line.

BUSY

This field contains a status of a Line Process for this line.

QUEUED

This field indicates the number of messages to one or more stations on this line.

CONDITION

This field contains the Line Conditions (See section 3.5, variant V).

EXCEPTION_A

This field contains the Line exceptions (See section 3.5, variant E).

EXCEPTION_B

This field contains the Line exceptions (See section 3.5, variant E).

ADAPTER_STATUS

This field contains the status of the DC interface of this line.

DISCONNECT_PENDING

This field indicates that this line is in process of being disconnected.

4.3.1.1 LINE_TABLE VARIABLES (Continued)

ANSWER_PHONE

This field enables the Line Process to answer the phone.

CRC

CRC

TALLY

General purpose 10 tallies.

BLOCK_TABLE_PTR

This field contains the pointer of the Block table. The Block table should contain:

a. PID of this Table

b. n entries of pointers and PID's of blocks allocated by this Line process

It is the responsibility of the Line Process to maintain this table. The format of the table is as follows:

```
DECLARE     BLOCK_TABLE_PTR       POINTER;
DECLARE     BLOCK_TABLE BASED BLOCK_TABLE_PTR
            STRUCTURE(
            SELF_PID              BYTE,
            BLOCK_PTR    (24)     POINTER,
            BLOCK_PID    (24)     BYTE);
```

4.3.1.2 STATION_TABLE VARIABLES

```
DECLARE
     CURRENT_STATION_PTR                    POINTER;

DECLARE
     STATION   BASED CURRENT_STATION_PTR
     STRUCTURE(
          LSN                                WORD,
          LINE                               BYTE,
          MCS                                BYTE,
          RECEIVE_ADDR                 (4)   BYTE,
          TRANSMIT_ADDR                (4)   BYTE,
          DEVICE_ADDR                  (4)   BYTE,
          MAX_INPUT                          WORD,
          MAX_OUTPUT                         WORD,
          MAX_RETRY                          BYTE,
          FREQUENCY                          BYTE,
          TERMINAL_TYPE                      BYTE,
          TRANSMISSION_NO                    BYTE,
          PREV_STATION_PTR                   POINTER,
          NEXT_STATION_PTR                   POINTER,
          PREV_MCS_PTR                       POINTER,
          NEXT_MCS_PTR                       POINTER,
          OUT_Q_HEAD_PTR                     POINTER,
          OUT_Q_TAIL_PTR                     POINTER,
          NEXT_MSG_PTR                       POINTER,
          MSG_COUNT                          BYTE,
          STATUS                             BYTE,
          READY                              BYTE,
          ENABLED                            BYTE,
          INPUT_ENABLED                      BYTE,
          LOGICAL_ACK                        BYTE,
          RETRY_COUNT                        BYTE,
          RETRY_INTERVALS                    BYTE,
          EVENT                              BYTE,
          WAIT_ACK                           BYTE,
          TIME_STAMP                         WORD,
          SEND_TRAN_NO                       WORD,
          RECEIVE_TRAN_NO                    WORD;
          SEQUENCE_MODE                      BYTE,
          SEQUENCE_SIZE                      BYTE,
          SEQUENCE_INC                 (2)   WORD,
          SEQUENCE_NO                  (2)   WORD,
          TALLY                        (10)  BYTE);
```

4.3.1.2 STATION_TABLE VARIABLES (Continued)

LSN

Logical Station Number.

LINE

This field contains binary Line Number.

MCS

This field contains binary MCS number to which this station belongs.

RECEIVE_ADDR

This field contains the Receive Address of this station.

TRANSMIT_ADDR

This field contains the Transmit Address of this station.

DEVICE_ADDR

This field contains the Device Address of this station (IBM 3270).

MAX_INPUT

This field contains the largest input size for this station.

MAX_OUTPUT

This field contains the largest output size for this station.

MAX_RETRY

This field contains the maximum retry count. When the actual retry count reaches this value, the Line Process will report an error.

4.3.1.2 STATION_TABLE VARIABLES (Continued)

FREQUENCY

This field contains time intervals between pollings of this station.

TERMINAL_TYPE

This field contains Terminal Type of this station. The 2 least significant bits of the TERMINAL_TYPE contains the "MYUSE" of this station;

00 - Transmit/Receive
01 - Receive only
10 - Transmit only

TRANSMISSION_NO

This field contains the type of transmission number used on this station.

00 - Not used.
01 - 0 / 1.
02 - 0, 1, 2...9.
03 - 3 byte count.

PREV_STATION_PTR

This field contains a pointer of the previous station in the line chain. If this is a first station in the line chain, this field will contain "0"s.

NEXT_STATION_PTR

This field contains a pointer of the next station in the line chain. If this is a last station in the line chain, this field will contain "0"s.

PREV_MCS_PTR

This field contains a pointer of the previous station in the MCS chain. If this is a first station in the MCS chain, this field will contain "0"s.

4.3.1.2 STATION_TABLE VARIABLES (Continued)

NEXT_MCS_PTR

This field contains a pointer of the next station in the MCS chain. If this is a last station in the MCS chain, this field will contain "0"s.

OUT_Q_HEAD_PTR

This field contains a first message in the Station Out Queue.

OUT_Q_TAIL_PTR

This field contains a pointer of a last message in the Station Out Queue.

NEXT_MSG_PTR

Reserved.

MSG_COUNT

This field contains a count of messages in the Station Out Queue.

STATUS

This field contains a status of this station.

READY

This field contains a Station Ready boolean value.

ENABLED

This field contains a Station Enabled Boolean value. It will be set to "0" when a MCS of this station changes status to "Closed" or "Open-and-Blocked".

4.13.1.12 STATION_TABLE VARIABLES (Continued)

INPUT_ENABLED

This field contains a Station Input Enable boolean value. This field will be set by an "ENABLEINPUT" function.

LOGICAL_ACK

This field indicates that the Logical Ack for this station is set.

RETRY_COUNT

This field contains actual retry count. On each successful I/O this field will be loaded by the Line Process with MAX_RERTY value.

RETRY_INTERVALS

This field contains the Retry Intervals in seconds between polls when the station is physically not ready.

EVENT

This field is used by the Line Process.

WAIT_ACK

This field contains a Station Wait for Logical Ack boolean value.

TIME_STAMP

This field contains a Time Stamp used by the Line Process for general purposes.

SEND_TRAN_NO

This field contains Send Transmission Number.

4.3.1.2 STATION_TABLE VARIABLES (Continued)

RECEIVE_TRAN_NO

This field contains Receive Transmission Number.

SEQUENCE_MODE

This field indicates whether this Sequence Mode of this station is set.

SEQUENCE_SIZE

This field contains the Max. Sequence Size.

SEQUENCE_INC

This field contains Sequence Number increments.

SEQUENCE_NO

This field contains the actual Sequence Number.

TALLY

General purpose tallies.

4.13.1.3 MESSAGE DECLARATION

```
DECLARE
     ACTUAL_MESSAGE_PTR                    POINTER;

DECLARE
     MESSAGE BASED ACTUAL_MESSAGE_PTR
     STRUCTURE(
     FUNCTION                              BYTE,
     FILLER                                BYTE,
     LSN                                   WORD,
     VARIANT                               WORD,
     TEXT_LENGTH                           WORD,
     FILLER                          (5)   BYTE,
     RETRY_COUNT                           BYTE,
     TALLY                           (4)   BYTE,
     QUEUE_CONTROL                         BYTE,
     FILLER                          (3)   BYTE,
     MESSAGE_NUMBER                        WORD,
     SEQUENCE_NUMBER                 (2)   WORD,
     FILLER                          (6)   WORD);


DECLARE
     RESULT BASED ACTUAL_MESSAGE_PTR
     STRUCTURE(
     TYPE                                  BYTE,
     ORIGINAL_FUNCTION                     BYTE,
     LSN                                   WORD,
     VARIANT                               WORD,
     TEXT_LENGTH                           WORD,
     EXCEPTION                       (4)   BYTE,
     FILLER                                BYTE,
     RETRY_COUNT                           BYTE,
     TALLY                           (4)   BYTE,
     TRANS_NUMBER                    (2)   BYTE,
     RBI                                   BYTE,
     FILLER                                BYTE,
     MESSAGE_NUMBER                        WORD,
     SEQUENCE_NUMBER                 (2)   WORD,
     ORIGINAL_VARIANT                      WORD,
     FILLER                          (5)   WORD);
```

4.3.1.4 STATION DIRECTORY DECLARATION

```
DECLARE
     STATION_DIR (256) POINTER;
```

4.3.1.5 MCS DIRECTORY DECLARATION

```
DECLARE
     MCS_DIR     (256)     STRUCTURE(
     FIRST_STATION_PTR     POINTER
     STATUS                BYTE) PUBLIC;
```

4.3.2 EXECUTIVE PROCEDURES

4.3.2.1 ROUTER_MAIN_LOOOP

This procedure is the main loop in the ROUTER module. The execution starts when the START module issues a "Resume" system call to the operating system.

```
ROUTER_LOOP PROCEDURE: PUBLIC;
  CALL INITIATE_SUBSYSTEM;
  DO FOREVER;
    CALL SC_QPEND (MESSAGE_QUEUE)
    CASE MESSAGE.FUNCTION;
              %
              %
       This case statement handles
       each DCP message individually.
       Invalid Functions will be
       detected.
              %
              %
    END CASE;
  END;
END ROUTER_LOOP;
```

4.3.2.2 DISPATCHER_LOOOP

This procedure is the main loop in the DISPATHER task. The execution of this task starts when START module issues a "Resume" system call.

```
DISPATCHER_LOOP PROCEDURE: PUBLIC;
  DO FOREVER;
    CALL SC_QPEND (RESULT_QUEUE);
    CALL SEND_HOST
  END;
END DISPATCHER-LOOP;
```

4.13.2.3 QUEUE_FUNCTION PROCEDURE

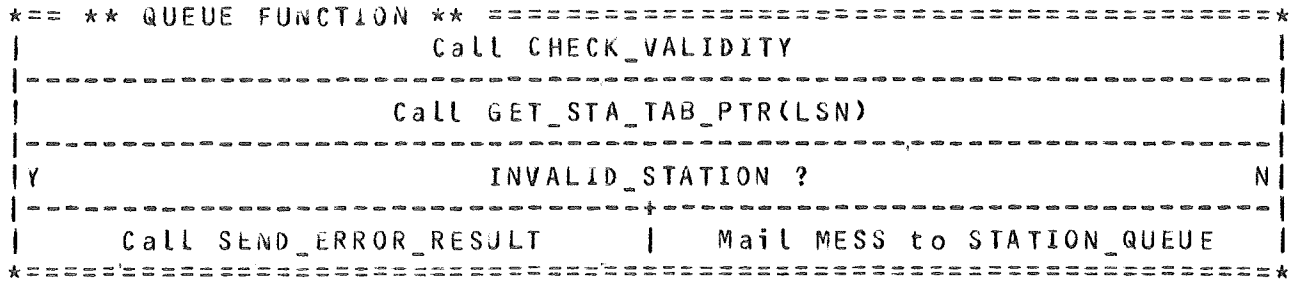
This procedure will be called when the host sends a DCP message that requires Line Process handling. This procedure will perform the following:

a. Check the validity of the header.

b. Find the station table.

c. Check if the station is available for output.

d. Put the message in the station queue (top or bottom)



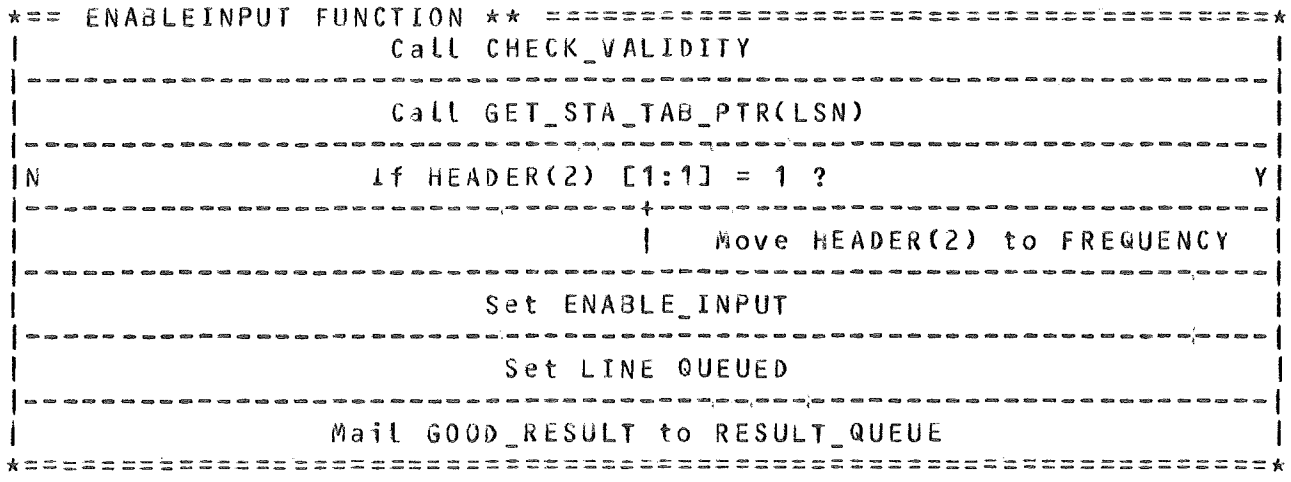
4.3.2.4 IMMEDIATE FUNCTION PROCEDURES

This section defines all the procedures handling all the DCP Headers that have to be performed immediately, or the function that has to performed is not running Line Process. The message procedures are:

1. Enableinput (33)
2. Make Station Ready (34)
3. Recall (40)
4. Acknowledge (43)
5. Make Line Ready (47)
6. Make Line Not Ready (48)
7. Dialout (49)
8. Disconnect (50)
9. Answer the Phone (51)
10. Interrogate Line Status (52)
11. Set Autoanswer (53)
12. Reset Autoanswer (54)
13. Memory Dump (61)
14. Load (71)
15. Echo (72)

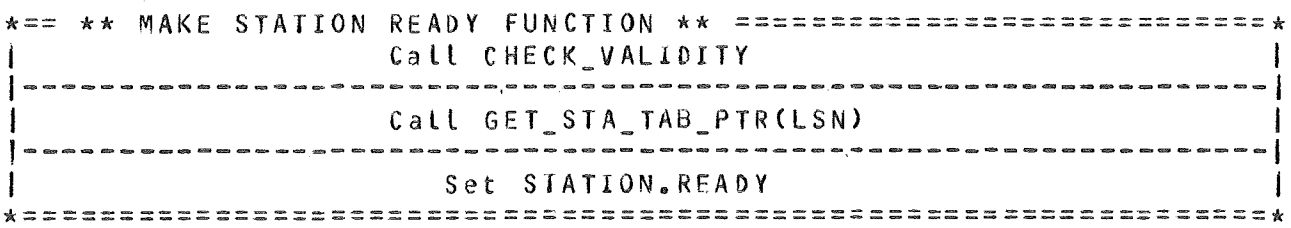
4.3.2.5 ENABLEINPUT MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Enableinput" message procedure:



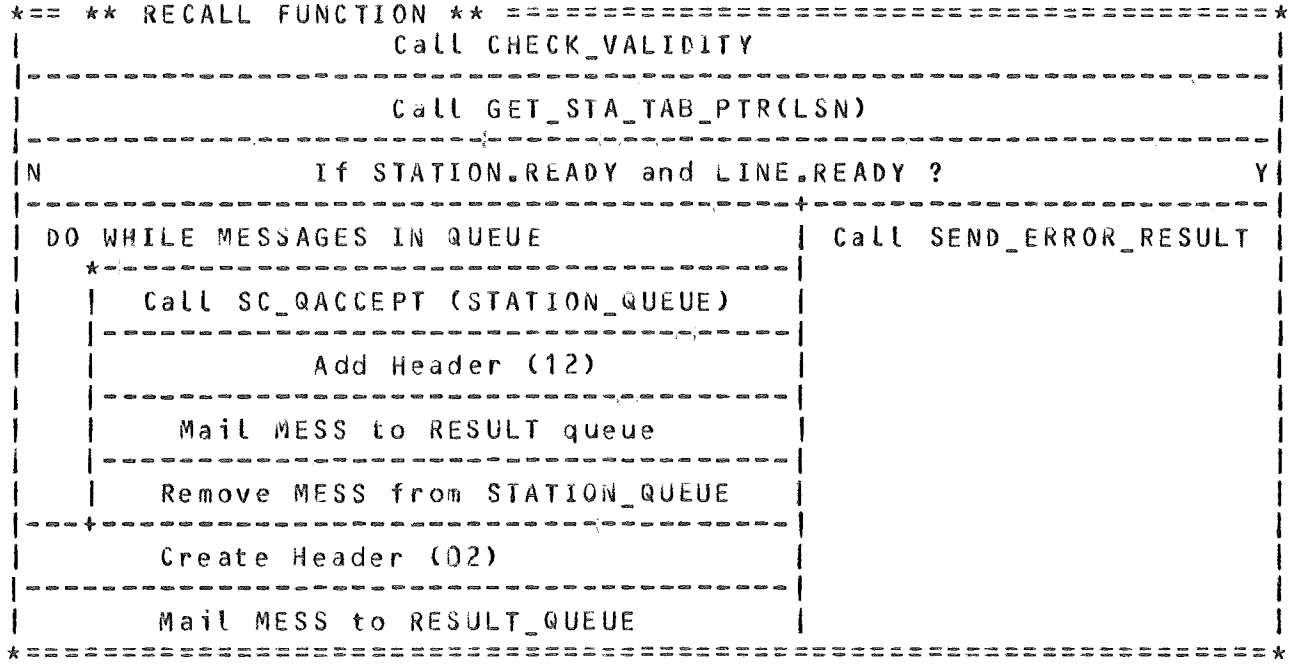
4.3.2.6 MAKE STATION READY MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Make Station Ready" message procedure:



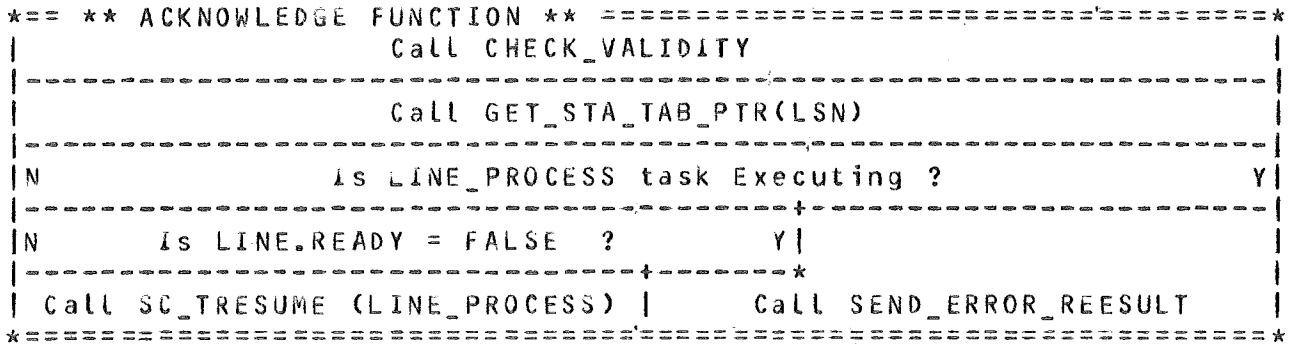
4.3.2.7 RECALL MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Recall" message procedure:



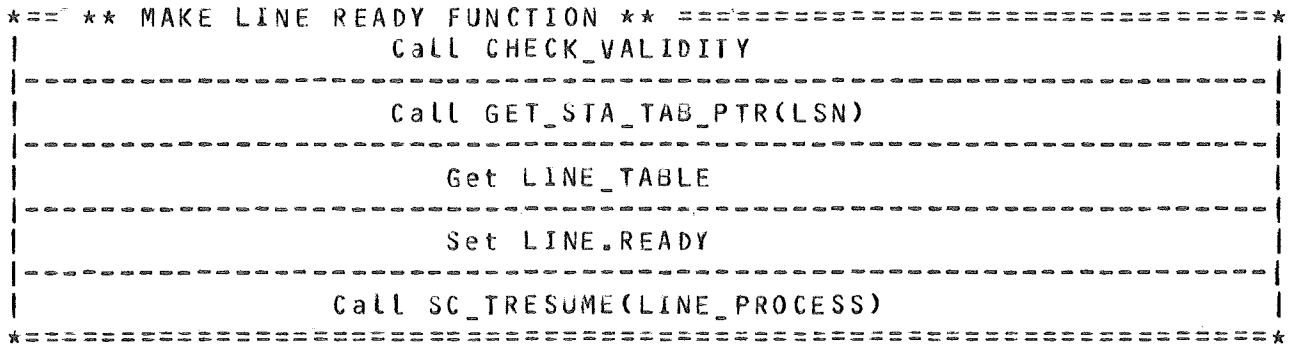
4.3.2.8 ACKNOWLEDGE MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Acknowledge" message procedure:



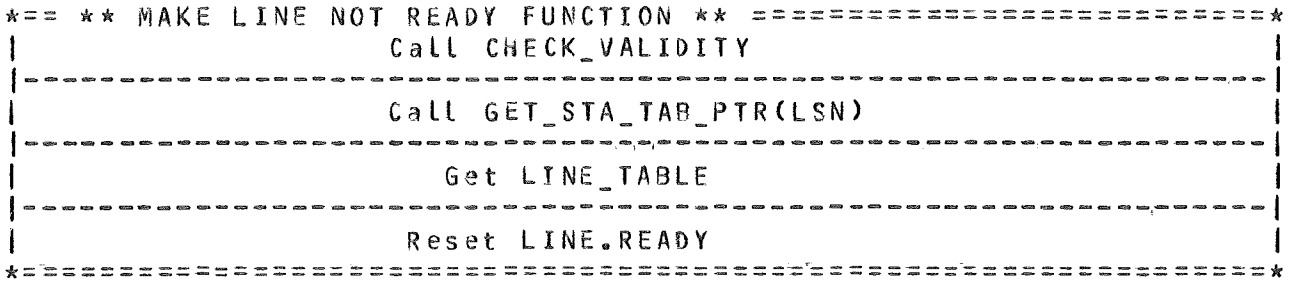
4.3.2.9 MAKE LINE READY MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Make Line Ready" message procedure:



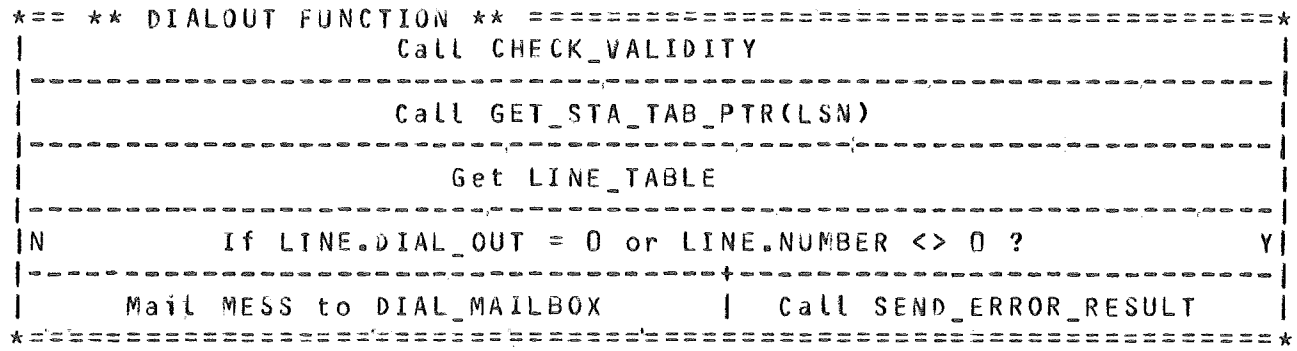
4.3.2.10 MAKE LINE NOT READY MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Make Line Not Ready" message procedure:



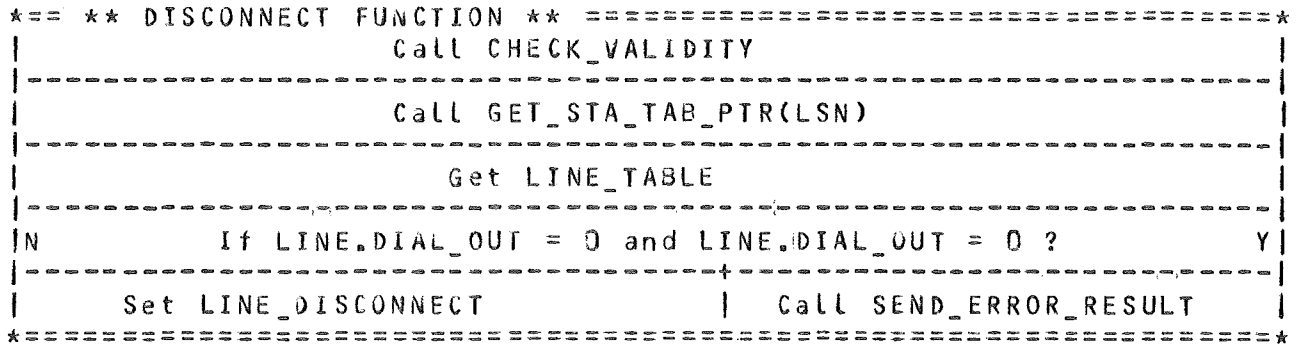
4.3.2.11 DIALOUT MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Dialout" message procedure:



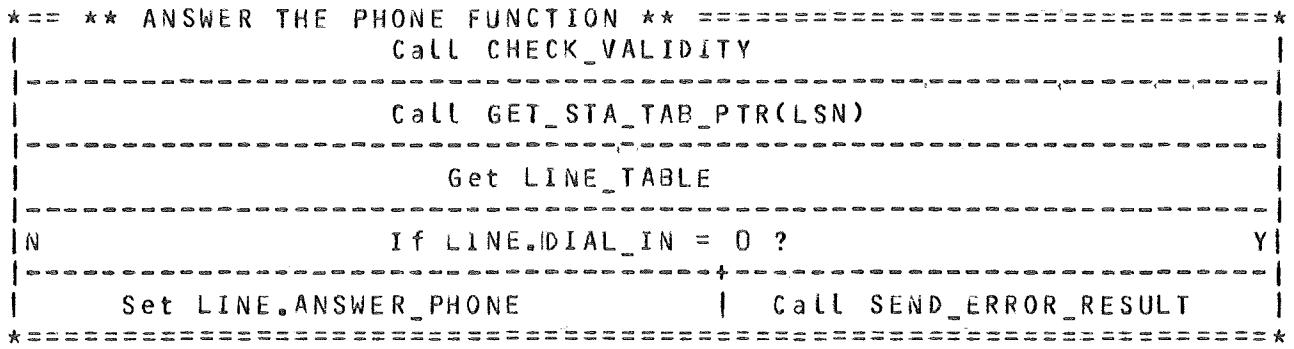
4.3.2.12 DISCONNECT MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Disconnect" message procedure:



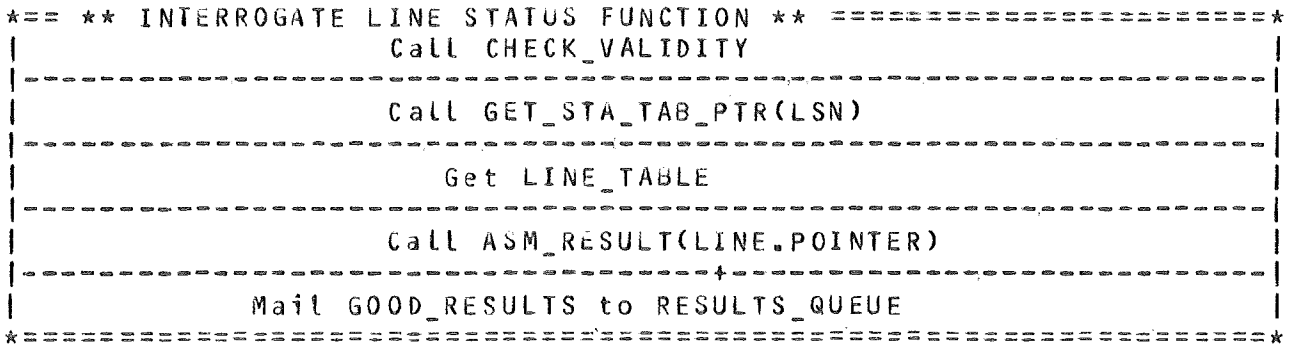
4.3.2.13 ANSWER THE PHONE MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Answer the Phone" message procedure:



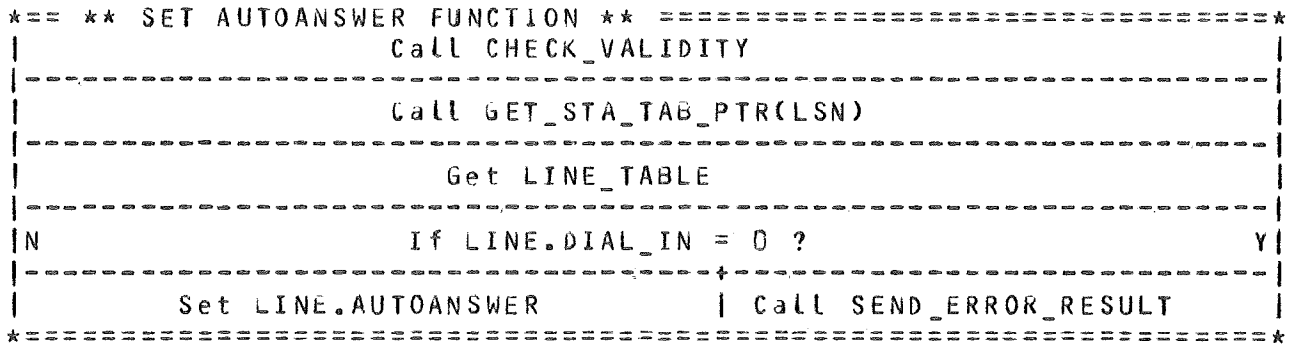
4.3.2.14 INTERROGATE LINE STATUS MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Interrogate Line Status" message procedure:



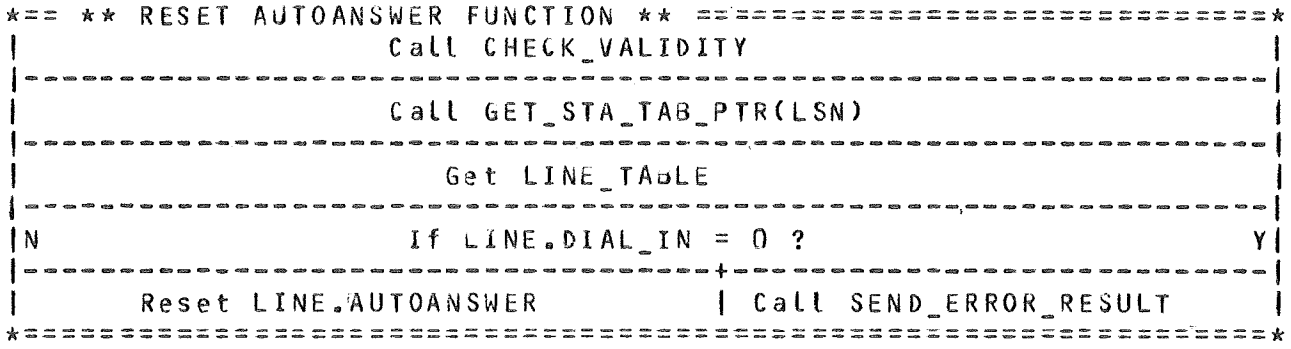
4.3.2.15 SET AUTOANSWER MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Set Autoanswer" message procedure:



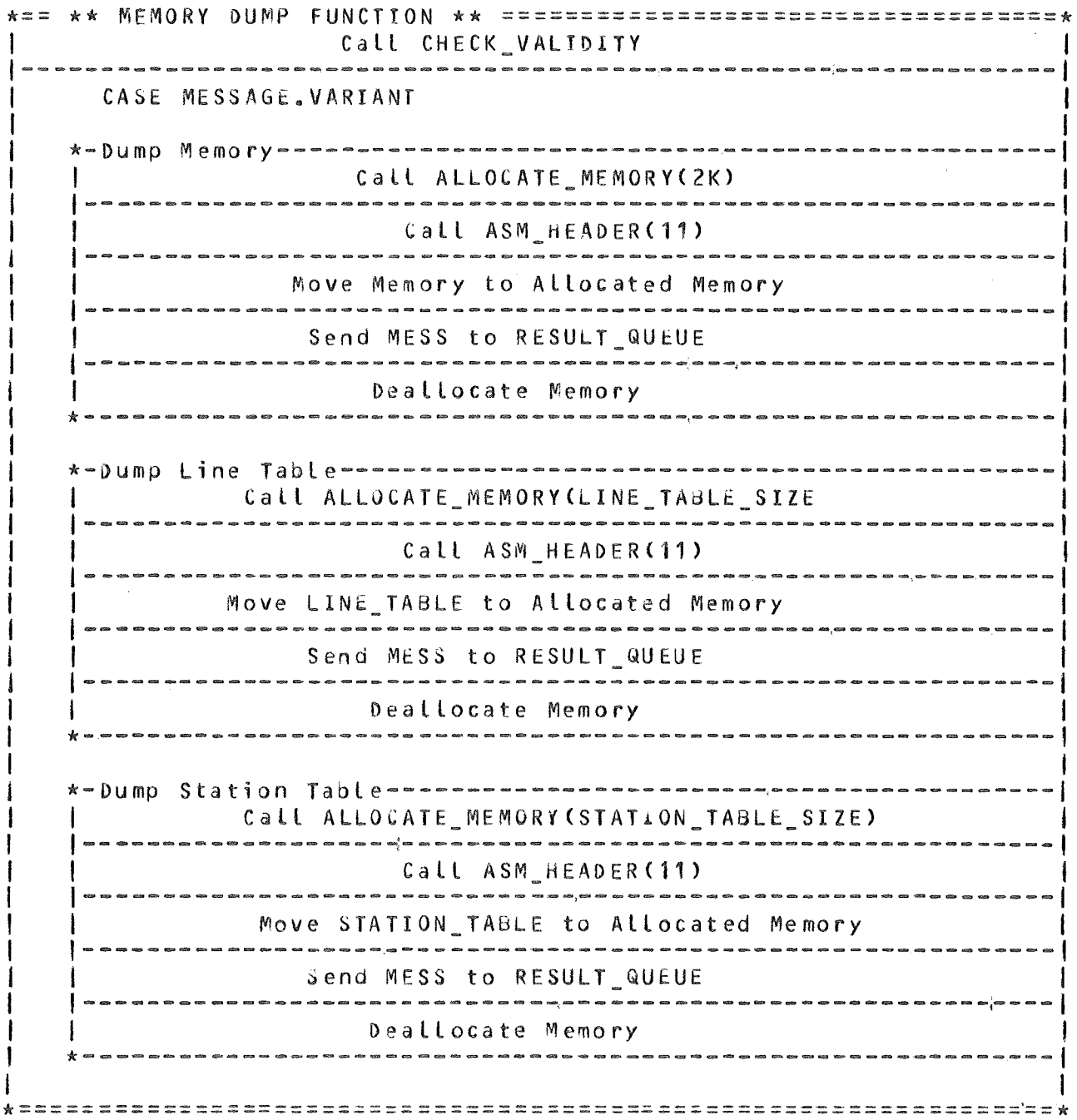
4.3.2.16 RESET AUTOANSWER MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Reset Autoanswer" message procedure:



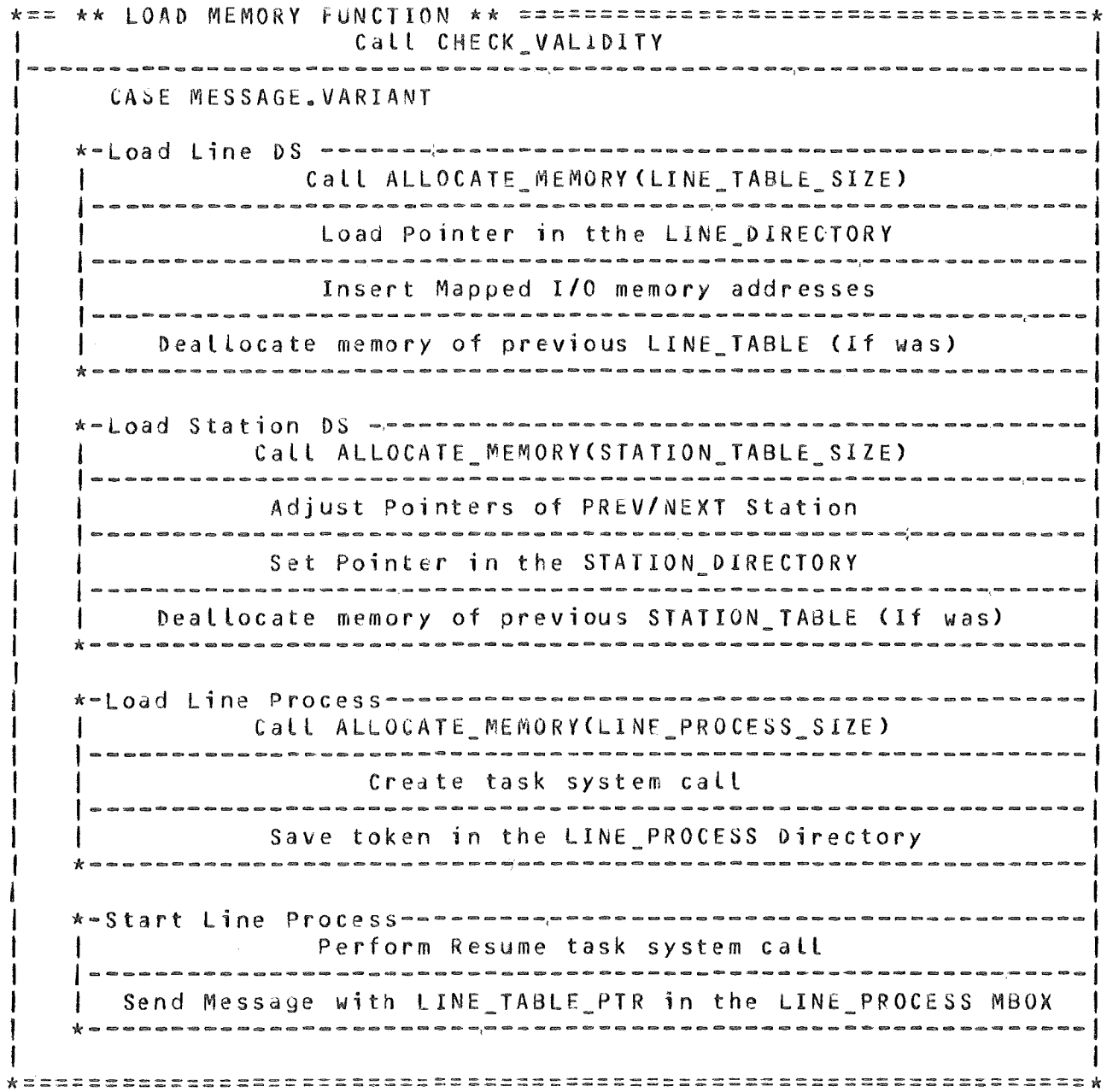
4.3.2.17 MEMORY DUMP MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Memory Dump" message procedure:



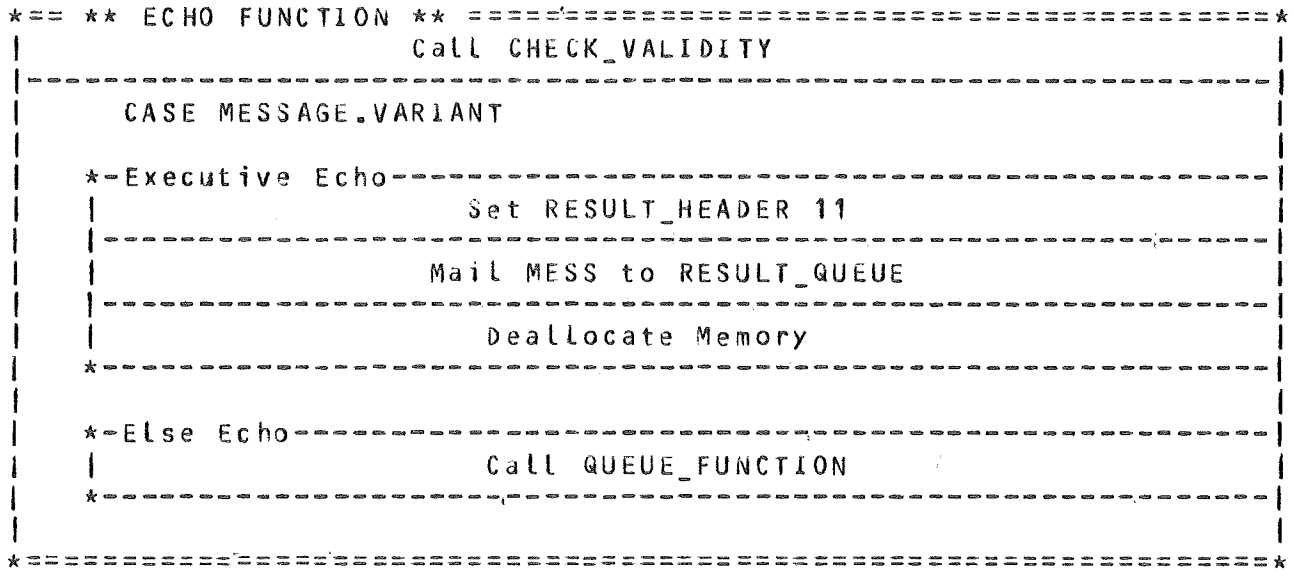
4.3.2.18 LOAD MEMORY MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Load Memory" message procedure:



4.3.2.19 ECHO MESSAGE PROCEDURE

The following diagram describes the functional flow of the "Echo" message procedure:



4.4 REQUESTOR

The REQUESTOR is a module called by OP HANDLER or EXECUTIVE and can be considered as a submodule in the MLI HANDLER. The function of the REQUESTOR is to issue a request to the host to start a Poll Request.

This module first shows STC 1 (DISCONNECT) and presents Descriptor Link word 1. If connecting, the host is supposed to disconnect from the ORS DLP immediately upon seeing STC 1. It is possible that while loading STC 1, the host connects to the ORS DLP to start a Poll Test sequence for the CANCEL operation. If MLI_HANDLER ever sees connection with STC 1, it waits until host disconnects from the ORS DLP.

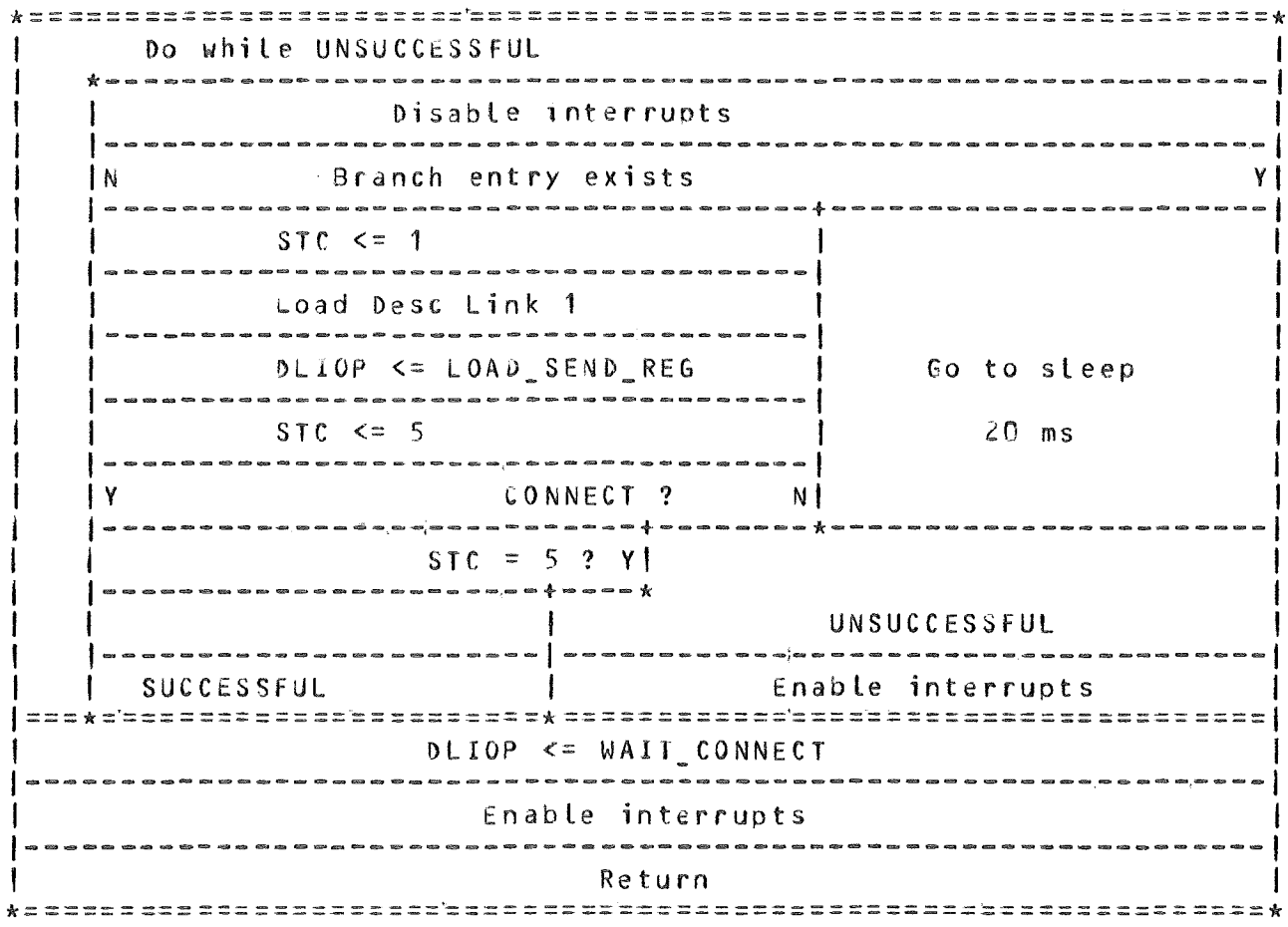
The MLI_HANDLER always has a WAIT_CONNECT DLIOP outstanding in the DLI state machine so that any time the host connects to the ORS DLP to start a Poll Test sequence, the DLI state machine will interrupt the processor and control will be passed to the MLI_HANDLER. When the REQUESTOR loads a new DLIOP, the previous outstanding WAIT_CONNECT DLIOP is canceled.

After loading Descriptor Link word 1, the REQUESTOR then changes to STC 5 and sets REQUEST. If the host is disconnected, the REQUESTOR then sends WAIT_CONNECT DLIOP to DLI state machine and exits.

The ACTIVE_FLAG parameter will set the type of action the Poll Request module has to perform (Read, Write or Send Result). Upon completion of the requested function, the Poll Request signals the OPHANDLER that the operation is complete.

4.4 REQUESTOR (Continued)

REQUESTOR FLOW



4.5 LINE_PROCESS

4.5.1 OVERVIEW

A LINE_PROCESS is downloaded from the host via a "Memory Load" (71) message. The role of the LINE_PROCESS is to process output messages which are found in the STATION_OUTPUT_QUEUE, accept spontaneous input from data comm lines, and to place the message in the RESULT_QUEUE. The LINE_PROCESS is also capable of handling the following DCP messages that are found in the STATION_OUTPUT_QUEUE:

1. Write (30)
2. Disableinput (33)
3. Make Station Not Ready (33)
4. Set Character (37)
5. Set Transmission (38)
6. Set Logical Ack (41)
7. Reset Logical Ack (42)
8. Null Station Request (44)
9. Set Sequence (45)
10. Disconnect (50)

The Line Process interfaces with the Executive in 4 ways:

a. Common Data Structure - Line/Station tables can be accessed by the Executive and the Line Process. See the Line & Station Tables section.

b. Line Process Start Mailbox - This Mailbox located at segment 0090H offset 0046H and is used to pass Line Table pointer to the Line Process during the initialization time.

c. Station Output Queue - This Queue holds all the messages for each station. Four variables in the Station Table support this queue. See station table for details.

d. Result Queue - The Result Queue (QID = 0050) is the only way to get messages out to the host. The Dispatcher module accepts the messages from this queue and sends them to the ISC HANDLER.

4.5.2 LINE_PROCESS DATA STRUCTURE

The pointer of the LINE TABLE passed to the Line Process via Start Mailbox located at segment 0090H offset 0046H during initialization time.

The LINE_TABLE is used as the primary method which the Line process finds the configuration of the line. A pointer within the LINE_TABLE points to the first STATION_TABLE. Each STATION_TABLE has a pointer to the next STATION_TABLE. At the last STATION_TABLE the pointer is "0".

4.16 INITIALIZATION SEQUENCE

This section describes the initialization sequence of the ORS DLP. In order to bring the ORS DLP from a state where firmware is fully operational, the following functions and sequence of functions have to be performed.

4.6.1 RESET SEQUENCE

Master Clear signal from the DLI bus will invoke the Reset sequence. Reset sequence consists of five major functions:

a. Unconditional suspension of any operation in progress.

b. Invoke Self Test.

c. Store the result of the Self Test in the diagnostic register.

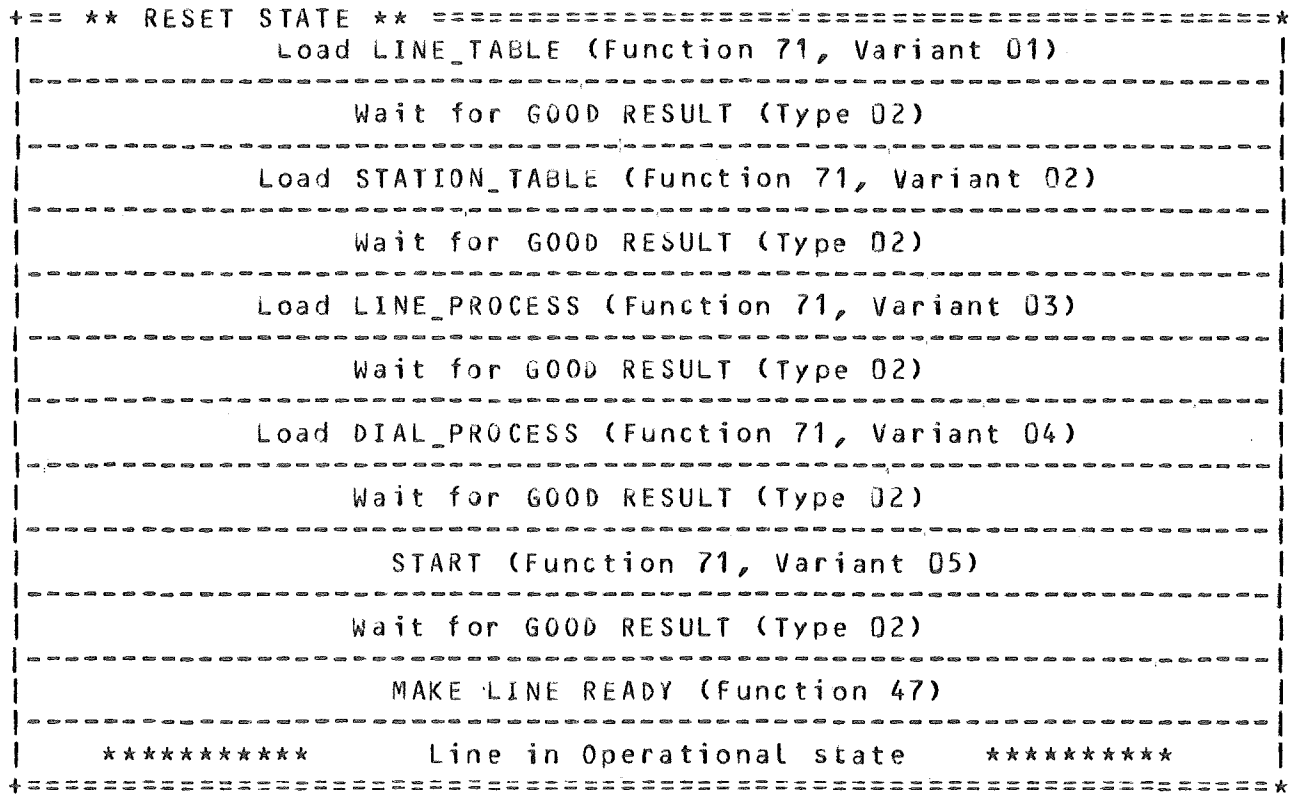
d. Initialization of EXECUTIVE data structure and peripheral chips.

e. Wait for first message from the host.

A succesful completion of the first four functions brings the ORS DLP into a Reset State.

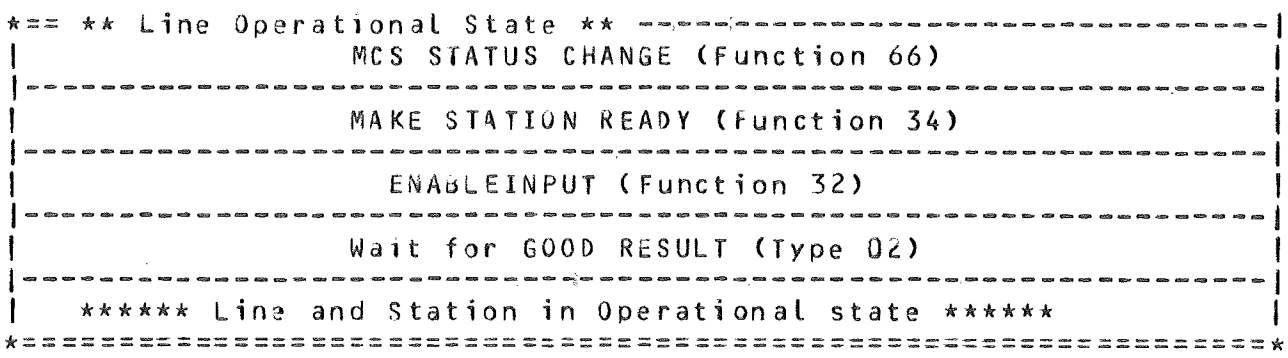
4.6.2 START SEQUENCE

Start sequence is defined as a sequence of events that will bring one of the ORS DLP lines from a Reset State to a Line Operational State. The sequence of events are supervised by the host. The events are various messages initiated by the host. The following chart gives the sequence of events and their responses:



4.6.2 START SEQUENCE (Continued)

In order to bring a Station from non operational state to an operational state, the following sequence of events have to be followed.



4.7 CLEAR SIGNALS

This section defines the Operating Firmware behavior after receiving any of the following clear signals:

PUPCLR - Power Up Clear

This Clear will force the Microprocessor into reset mode. The Microprocessor will initiate Self Test and upon successful completion of the Self Test, the Operating Firmware will be initiated.

MASTRCLR - Master Clear

If the "Selective Clear" jumper is in, then the Clear will force the Microprocessor into reset mode. The Microprocessor will initiate Self Test and upon successful completion of the Self Test, the Operating Firmware will be initiated. If the "Selective Clear Jumper" is out, then the Clear will force a Non maskable interrupt. The Operating Firmware will be initiated.

LCLCLR - Local Clear

This Clear will force the Microprocessor into reset mode. The Microprocessor will initiate Self Test and upon successful completion of the Self Test, the Operating Firmware will be initiated.

SELCLR - Selective Clear

This Clear will force a Non maskable interrupt. The Operating Firmware will be initiated.

5 SELF TEST

The ORS DLP employs a Self Test function with two methods of initiation and two methods of reporting the functionality of the ORS DLP. The two initiation methods are:

a. Reset
b. Execution of Self Test I/O opcode.

The ORS DLP will generate reset on receipt of any of the following signals:

a. Frontplane Clear (pushbutton)
b. Power-Up Clear
c. Master Clear (Jumper selected)
d. DLI State Machine Parity Error
e. Local Clear

The reset will initialize the DLI state machine, reset the 8086 microprocessor, SCC and CIO chips. The 8086 will restart execution from the bootstrap loader (address FFFF:0000).

The ORS DLP will also receive a Selective Clear which is fed to the CLEAR PAL, and upon reception of this clear, the microprocessor will be interrupted by a non-maskable interrupt. The non-maskable interrupt vector points to the Operational Firmware entry.

All clears are latched by the CLEAR PAL. The CLRCLRS register clears all the active latches in the PAL.

Master clear is configured as a jumper option. When the jumper is installed, it will cause a reset. When it is not installed, a non-maskable interrupt will take place.

After initialization of the Self Test, the ORS DLP will disable its peripheral and the DLI interfaces until such time when the ORS DLP passes its Self Test. A status of zero and LCPCON/0 is presented to the backplane whenever the ORS DLP is addressed by the Host.

5 SELF TEST (Continued)

Upon successful or unsuccessful completion of the Self Test, the Operational Firmware will be restarted and the results will be loaded into ORS DLP status register. There are two I/O opcodes avalaible to read the Self Test results:

a. Self Test Status I/O descriptor
b. Get Status I/O descriptor

The top frontplane red LED indicates the status of the Self Test. The LED is on when the Self Test is in progress or a failure is detected. When the LED is off, it will indicate successful completion of the Self Test. The operating firmware examines the the result of Self Test. If succsessful, the 500 ms cycle olinking will be initiated.

The execution time of the ORS DLP Self Test shall not exceed 5 seconds.

See Get Status and Self Test Status operations. The following table shows the ORS DLP Self Test result codes.

Code	Description
0001	8086 test failure
0002	CIO #1, 2 test failure
0003	SCC #1, 2 test failure
0004	PIC test failure
0005	PIT test failure
0006	RAM first 64K failure
0007	RAM failure
0008	Interrupts failure
0009	HDP/DLP init failure
001U	DLI state machine failure
0011	Dual port RAM failure
0012	DLI Host pointer failure
0013	DLI LPW failure
0014	DLI V.P. failure
0015	DLI burst counter failure
0020	RAM refresh failure
0021	Unexpected int failure
0022	Non Muskable interrupt failure
007A	Self test successful complete