NOTEBOOK SECTION

OPTION NUMBER

DRAWING SET NUMBER

PROGRAM NUMBER

DOCUMENT NUMBER   REVISION

DATE

APRIL 1978

RSX-11M PCL

SUPPORT MANUAL

YC-T012C-Z0

digital

Computer Special Systems

KANATA

TABLE OF CONTENTS

0.	INTRODUCTION
1.	OVERVIEW
1.1	Functional Description
1	Environment Description
1.3	Component Modules
2.	TRANSMITTER DRIVER - LTDRV
2.1	Functional Description
2.2	Environment
2.2.1	Characteristics Words Usage
2.2.2	UCB Work Area Usage
2.3	Initiator Service
2.3.1	SET Service
2.3.2	ATX Service
2.3.3	STA Service
2.4	Interrupt Service
2.5	Cancel Service
2.6	Power Fail Service
2.7	Time Out Service
2.8	Macro Definitions
2.8.1	FINSTS Macro
2.8.2	HNDERR Macro
3.	RECEIVER DRIVER - LRDRV
3.1	Functional Description
3.2	Environment
3.2.1	Characteristic Words Usage
3.2.2	UCB Work Area Usage
3.3	Driver States
3.3.1	Legal Functions by State
3.4	Initiator Service
3.4.1	CRX Service
3.4.2	RTF Service
3.4.3	ATF Service
3.4.4	DRX Service
3.5	Hardware Manipulation Subroutines
3.5.1	Perform CRX - PRCRX
3.5.2	Perform ATF - PRATF
3.5.3	Perform DRX - PRDRX

3.6 Interrupt Service

3.6.1 State -6
3.6.2 State -4
3.6.3 State -2
3.6.4 State 0
3.6.5 State +2
3.6.6 State +4
3.6.7 State +6

3.7 Cancel Service
3.8 Power Fail Service
3.9 Time Out Service
3.10 Macro Definitions

4. DATA STRUCTURE - USRTB

4.1 Conditional Assembly for Multiple Units
4.2 Receiver DCB
4.3 Transmitter DCB
4.4 Receiver UCB's
4.5 Transmitter UCB's
4.6 Receiver SCB's
4.7 Transmitter SCB's
4.8 Interrupt Vectors

0. INTRODUCTION

This document describes the logic and internal workings of the drivers for the Parallel Communications Link (PCL) and their associated data structure. The reader is assumed to be familiar with the following two manuals.

1. RSX-11M GUIDE TO WRITING AN I/O DRIVER
   AA-2600D-TC

2. RSX-11M PARALLEL COMMUNICATIONS LINK USER'S GUIDE
   YC-T012C-UD

This document describes the following versions of the following modules

Transmitter Driver	LTDRV V03-02
Receiver Driver	LRDRV V03-02
Transmitter Data Structure	LTTAB V03-02
Receiver Data Structure	LRTAB V03-02

1. OVERVIEW

1.1 Functional Description

Two drivers are provided to operate the PCL. One handles transmission of messages to the other processors in the network. The second handles reception of messages from them. The drivers operate independently of each other; transmission and reception may occur simultaneously in a processor.

1.2 Environment Description

The drivers operate as loadable parts of an RSX-11M Executive System. Since they have been made loadable, the user has the option of incorporating them into his system as either resident or loadable. The user initiates driver functions by means of the QIO RSX-11M directive.

1.3 Component Modules

The drivers will run in any RSX-11M system ($GTWRD and $PTWRD must be generated into the system). If the drivers are to run as loadable drivers, the operating system must have been generated to support loadable drivers. The amount of memory required by the drivers and their associated data structures is as follows: for the transmitter 1596 bytes plus 94 bytes for each unit after the first, for the receiver 1416 bytes plus 98 bytes for each unit after the first.

If the drivers are to be used on an 11/70 with extended memory support (22 bit addressing) then the core requirements are as follows: for the transmitter 1608 bytes plus 94 bytes for each unit after the first; for the receiver 1424 bytes plus 98 bytes for each unit after the first. Also, since the PCL operates as a non-processor request (NPR) device each transmitter or receiver requires one dynamically allocated UNIBUS memory register (UMR) for each 4K of its maximum data transfer.

2. TRANSMITTER DRIVER - LTDRV

2.1 Functional Descriptional

This driver allows a user to set the characteristics of the master section of the hardware, obtain the current setting of the master section and transmit messages to other processors in the PCL network. It supports the following GIO functions:

- set master section (SET)
- retrieve the current master section setting (STA)
- attempt transmission (ATX)
- cancel outstanding ATX's (KIL)

In addition, the following functions which are processed by the RSX-11M Executive, are legal for the PCL transmitter:

- attach to device (ATT)
- detach from device (DET)

All of these functions are described in the RSX-11M PCL USER'S GUIDE YC-T012C-U0.

2.2 Environment

This driver operates in the environment of the RSX-11M V3.1 Executive as described in RSX-11M GUIDE TO WRITING AN I/O DRIVER AA-2600D-TC.

It uses the data structure for device LT: as defined in the LTTAB module.

2.2.1 Characteristics Words Usage

The second and third device characteristics words (UCB offsets U.CW2 and U.CW3 respectively) are used by the drivers as follows:

U.CW2	bits 8-15	unused
	bits 0-7	number of retries to performed after current try (if necessary)
U.CW3	bits 0-15	number of driver errors detected since last system bootstrap (see 2.8.2)

2.2.2 UCB Work Area Usage

The transmitter driver uses words of UCB area workspace. They are used as follows:

OFFSET NAME	LENGTH	USAGE
U.BUF	2 bytes	1) For non-transfer functions, this contains the first parameter word from the I/O packet. 2) For transfer functions, this contains the first word of a 2 word address double word used to initiate the I/O operation. The format ofU.BUF is:

Word U.BUF

Bit 0	Go bit (=0)
Bits 1-3	Function code (=0)
Bits 4-5	Memory extension bits
Bit 6	Interrupt enable (=0)

OFFSET	LENGTH	USAGE
U.BUF+2	2 bytes	1) For non-transfer functions parameter it contains the second word from the I/O packet (if a second parameter word exists) 2) For transfer functions, it contains the second word of the 2- word address doubleword. This word contains the 16 bits of the physical address.
U.CNT	2 bytes	1) For non-transfer functions it contains the third parameter word from the I/O packet (if a third parameter word exists). 2) For transfer functions it contains the byte count of the buffer described by U.BUF and U.BUF+2.
U.LERR	2 bytes	Program counter value of last detected driver error (see 2.8.2)
U.EFN	2 bytes	Event flag number to be used when "now master" interrupt occurs.
U.TCB	2 bytes	TCB address of requestor that specified event flag number.
U.BUF1	2 bytes	The first word of the address doubleword (relocation bias) for the return register buffer (if specified).
U.BUF1+2	2 bytes	The second word of the address doubleword (the low-order 16 bits of the physical address) for the return register buffer (if specified).

2.3 Transmitter Driver Initiator Service

The transmitter driver initiator has its entry point at LTCHK. It is called by the executive prior to queueing the I/O Packet. This enables the driver to handle the additional buffer address "retadd"; (the return register buffer) as well as the buffer that the Executive address checks and relocates in a normal transfer request. If the address check succeeds, then the I/O Packet is constructed and inserted on the queue before the initiator is entered to start the transmitter. This routine is depicted in Flowchart # 1.

2.3.1 Initiator Service Routine

The initiator service routine is entered at location LTINI. It is entered in one of two ways: when an I/O request has been queued and at the end of a previous I/O operation to look for more work.

(When the initiator service routine is branched to from within the driver, the driver must not be executing its interrupt service routine unless at the $FORK level. This is because the initiator service routine assumes that it can exit from the driver by way of a RETURN Macro call).

The initiator service routine is outlined in Flowchart # 2. Its main function is to determine which QIO function is being requested and to transfer control to its service routine.

2.3.1.1 SET Service

The SET function service routine is outlined in Flowchart # 3. The function of this routine is to set the master section values as specified in the function parameter list. Before any changes are made to the current setting of the TMMR and/or the address silo, all parameters are checked for validity.

2.3.1.2 ATX Service

The ATX function service routine is responsible for validating the parameters passed to it and for initiating the transmission of the data buffer. Flowchart # 4 outlines this service routine.

2.3.1.3 STA Service

The STA function service routine retrieves and stores in IOSB+2, the current setting of the Transmitter Master/Maintenance Register (TMMR). Flowchart # 5 pertains to this routine.

2.4 Interrupt Service

The interrupt service routine is outlined in Flowchart # 6. As can be seen from the flowchart, successful transfer processing is performed starting at LTGOOD, and error processing is performed at LTERR. If a "Now Master" interrupt occurs and an event flag number has been specified, an event flag is set to notify the user that this master section is now in control.

2.5 Cancel Service

The cancel service routine, LTCAN, is called by the RSX-11M Executive whenever a KIL function is issued. The KIL function may be issued explicitly by the user task or implicitly within the Executive during task termination.

If the task issuing the KIL function (either explicitly or implicitly) is the one whose function is in progress, the cancel service routine aborts the ATX function by initializing the PCL transmitter and returning STS=IE.ABO for the ATX. It then branches to LTINI to look for more work.

If the task issuing the KIL function is not the one whose ATX is in progress, the cancel service routine just returns to the Executive.

2.6 Power Fail Service

The power fail service routine, LTPWF, is called by the RSX-11M Executive during power fail recovery. It simply does a return to the Executive.

2.7 Time Out Service

The timeout service routine, LTOUT returns an error code of 'DEVICE NOT READY, initializes the PCL and goes and looks for more work.

2.8 Macro Definitions

Two macros are defined and used in the driver:

FINSTS - signals function completion and returns specified status block values;

HNDERR - signals the detection of a driver error (which may actually be a hardware error).

2.8.1 FINSTS Macro

The FINSTS macro signals I/O function completion and specifies values for the I/O status block (IOSB) associated with the QIO request (if specified).

```
        FINSTS          stwd0,stwd1
```

where

	stwd0	is an assembler operand specifying the contents of the first word of the IOSB
and	stwd1	is an assembler operand specifying the second word

The second parameter, stwd1, is optional; it is assumed to be zero if omitted.

This macro causes the executive routine $IODON to be called. (Its alternate entry point, $IOALT, will be called if stwd1 is omitted.) The routine is described in Chapter 4 of RSX-11M Guide to Writing an I/O Driver (AA-2600D-TC).

2.8 HNDERR Macro

The HNDERR macro signals that the driver has encountered a supposedly impossible circumstance. The circumstance might be the result of an error in the PCL hardware or in the driver's own code.

If the symbol DEBUG is not defined, indicating that the driver is in production use, the macro generates a call to the subroutine HNDERR, which increments U.CW3 (the count of HNDERR calls since the last system bootstrap operation) and saves the return address in U.LERR.

If DEBUG is defined, indicating that debugging is in progress, the macro generates a breakpoint trap instruction (BPT) so that a debugging aid such as XDT will be entered.

Regardless of the status of the DEBUG symbol, the macro always returns to the driver so that some sort of recovery may be attempted.

3. RECEIVER DRIVER - LRDRV

3.1 Functional Description

This driver allows one proussor to receive messages from other processors in the PCL network. It supports the following QIO functions:

- connect for reception (CRX)
- reject transfer (RTF)
- accept transfer (ATF)
- disconnect from reception (DRX)
- cancel outstanding requests (KIL)

Attach (ATT) and detach (DET) device functions are not legal for this driver. Their need is obliviated by CRX and DRX.

The above functions are described in the RSX-11M PCL User's Guide YC-T012C-UD.

3.2 Environment

The driver operates in the environment of the RSX-11M executive as described in RSX-11M Guide to Writing an I/O Driver (AA-2600D-TC). It uses the data structure for device LR: defined in the LRTAB module.

3.2.1 Characteristics Words Usage

The second and third device characteristics words (UCB offsets U.CW2 and U.CW3 respectively) are used by the driver as follows:

U.CW2	bits 8-15	unused;
	bits 0-7	index of current driver state (see 3.3)
U.CW3	bits 0-15	number of errors detected since last system bootstrap (see 3.10)

3.2.2 UCB Work Area Usage

The driver uses 5 words of UCB workspace. They are used as follows:

OFFSET	LENGTH	USAGE
U.LERR	2 bytes	program counter value of last detected driver error (see 3.10)
U.TASK	2 bytes	TCB address of connected task (0 if no task connected)
U.TEF	2 bytes	trigger event flag number
U.TRSB	4 bytes	real address of trigger status buffer

3.3 Driver States

The driver considers a PCL unit to be in one of a number of software states during its operation. Its current state is defined in the low order byte of U.CW2 in the UCB. Driver states are defined below:

INDEX	MEANING
0	no task connected
2	task connected but not triggered
4	task triggered and awaiting ATF or RTF
6	task connected and timed out awaiting ATF or RTF

The controller is considered busy by RSX-11M while the driver is in one of the states below. Therefore, only the interrupt processor need concern itself with them.

INDEX	MEANING
-2	ATF in progress
-4	task connected, not triggered, and has ATF pending
-6	RTF in progress

3.3.1 Legal Functions by State

The states in which functions processed by the driver are legal are tabulated below:

FUNCTION	LEGAL STATES
CRX	0
RTF	4, 6
ATF	2, 4, 6
DRX	2, 4, 6

3.4 Initiator Service

The receiver driver initiator has its entry point at LRCHK. It is called by the executive prior to queueing the I/O Packet. This enables the driver to handle the additional buffer address "retadd"; (the return register buffers) as well as the buffer that the Executive address checks and relocates in a normal transfer request. If the address checks succeed, then the I/O Packet is constructed and inserted on the queue before the initiator is entered to start the receiver. This routine is depicted in Flowchart # 7.

The initiator service routine has its entry point in LRINI. It is called by the executive whenever a new I/O packet has been enqueued for the device. It is also branched to within the driver whenever it has completed an operation and is looking for more work.

(When the initiator service routine is branched to from within the driver, the driver must not be executing its interrupt service routine unless at the $FORK level. This is because the initiator service routine assumes that it can exit from the driver by means of a RETURN macro call.)

The main portion of the initiator service routine is outlined in Flowchart # 8. Its main function is to determine which QIO function is being requested and transfer control to the appropriate service routine.

With the exception of RTF, the function service routines do not communicate directly with the PCL hardware. They perform the software housekeeping aspects of their processing and then call a subroutine to manipulate the hardware. This approach has been taken because othe parts of the driver also have cause to call these subroutines. The hardware manipulation subroutines are described in 3.5.

3.4.1 CRX Service

The CRX function service routine is outlined in Flowchart # 9.

The task I/O count (at TCB offset T.IOC) is incremented to ensure that LRCAN is entered at task termination if the task neglects to disconnect from reception. LRCAN, which is described in 3.7, will disconnect the task from the device so that the device does not find itself connected to a nonexistent task. The DRX service routine decrements the I/O count.

3.4.2 RTF Service

The RTF function service routine is outlined in Flowchart # 10.

3.4.3 ATF Service

The ATF function service routine is outlined in Flowchart # 11.

3.4.4 DRX Service

The DRX function service routine is outlined in Flowchart # 12.

3.5 Hardware Manipulation Subroutines

For the CRX, AFT and DRX functions, the actual performance of the functions, from the hardware standpoint, is performed by a subroutine separate from the function service routine. This was done because these functions are also activated from other parts of the driver. For example, an implicit CRX is performed after completion of an ATF.

3.5.1 Perform CRX - PRCRX

This routine is called to connect the receiver for reception. It activates the hardware to receive a word, which will be treated as a flags word. It sets the driver state to 2. The routine is called whenever the driver wishes to receive a flags words.

Calling instruction: JSR PC,PRCRX

Inputs:
R4 = SCB address
R5 = UCB address

Outputs:
The STS byte of the trigger status block is set to zero.
The receiver is initialized (BDINIT) and RCVWD is set.
The driver state is set to 2.
RO = RSR address

3.5.2 Perform ATF - PRATF

This routine is called to initiate the reception of the message which has just been accepted. The state of the driver is set to 2, which indicates that the driver is busy performing the accept transfer function.

Calling instruction: JSR PC,PRATF

Inputs:
R4 = SCB address
R5 = UCB address

```
Outputs:        The RDBC and RDBA device registers are
                set up.
                An NPR RCVDAT function is initiated.
                The driver state is set to -2.
                RO = RSR address
                R2 is destroyed.
```

3.5.3 Perform DRX - PRDRX

This routine is called to disconnect the receiving task from reception. It decrements the task's I/O count, which was incremented by the CRX function service routine. (See 3.4.1 for an explanation of this.)

Calling instruction: JSR PC,PRDRX

```
Inputs:         R1 = TCB address
                R4 = SCB address
                R5 = UCB address

Outputs:        The task's I/O count is decremented.
                The receiver is initialized (BDINIT).
                The driver state is set to O.
                RO = RSR address.
```

3.6 Interrupt Service

The interrupt service routine processes interrupts depending upon the current driver state. The main part of the routine merely performs the following actions:

- the processor priority is lowered to 5.
- PCL receiver interrupts are disabled.
- A FORK process is created.
- The interrupt service code for the current driver state is initiated.

The interrupt service code for each state performs actions depending upon the status of the bits in the receiver device registers, particularly the RSR and RCR. The bit settings and their corresponding actions are tabulated for each state in the following subsections.

3.6.1 State -6

RECOM=1 and SUCTXF=0:

- call PRCRX
- return STS=IS.SUC
- go to LRINI

Other Settings of RECOM and SUCTXF

- signal driver error
- return STS=IE.FHE (fatal hardware error)
- go to LRINI

3.6.2 State -4

PAR=1:

- call PRCRX
- set driver state to -4
- return

PAR=0 and ERR=1:

- signal driver error
- call PRCRX
- set driver state to -4
- return

DTORDY=1:

- Throw away flags word
- call PRATF
- return

Anything else:

- signal driver error
- call PRCRX
- return

3.6.3 State -2

SUCTXF=1 and RECOM=0

- call PRCRX
- return STS=IS.SUC
- go to LRINI

SUCTXF=1 and RECOM=1:

- call PRCRX
- return STS=IS.TNC
- go to LRINI

BCOFL=1:

- set REJ and IEN
- return

PAR=1:

- call PRCRX
- return STS=IE.BBE
- go to LRINI

TXMERR=1:

- call PRCRX
- return STS=IE.BBE
- go to LRINI

TIMOUT=1:

- call PRCRX
- return STS=IE.DNR
- go to LRINI

MEMOFL=1 or NEXLOC=1:

- signal driver error
- call PRCRX
- return STS=IE.SPC
- go to LRINI

Other Setting

- signal driver error
- return STS=IE.BBE
- go to LRINI

3.6.4 State 0

All b.t. Settings:

- signal driver error (IEN should be 0)
- initialize receiver (set BDINIT)
- return

3.6.5 State +2

Interrupt processing is the same as for state -4 except for the processing of the DTORDY condition, which is as follows:

DTORDY=1:

- move trigger status of IS.SUC, the transmitter's identifier, and the flags word into the status block
- set driver state to +4
- set the trigger event flag (if specified)
- declare a significant event
- return

3.6.6 State +4

TIMOUT=1:

- call PRCRX
- set driver state to +6
- return

Anything else:

- signal driver error
- call PRCRX
- set driver state to +6
- return

3.6.7 State +6

Interrupt processing is the same as for state -4 except for the processing of the DTORDY condition, which is as follows:

DTORDY=1:

- move trigger status of IE.DAO, the transmitter's identifier, and the flags word into the status block
- set driver state to +4
- set the trigger event flag (if specified)
- declare a significant event
- return

3.7 Cancel Service

The cancel service routine, LRCAN, is called by the RSX-11M executive whenever a KIL function is issued. The function may be issued explicitly by the user task or implicitly within the executive during task termination. Because the UC.KIL bit is set in the unit control block(s) for LR: (see 4.4) the cancel service routine is called when a KIL is issued regardless of whether any other functions are pending or not.

If the task issuing the KIL function (either explicitly or implicitly) is not currently connected for reception, then the cancel service routine just returns to the executive. Otherwise, if task termination is in progress, the task is disconnected and STS=IE.ABO is returned for any outstanding request. If task termination is not in progress and a request is outstanding, the task is reconnected for reception (by calling PRCRX), and STS=IE.ABO is returned.

3.8 Power Fail Service

The power fail service routine, LRPWF, is called by the RSX-11M executive during power fail recovery. It simply returns to the executive.

3.9 Time Out Service

The timeout service routine, LROUT, is called when a software timeout occurs. This routine checks to see if a task was connected, and busy. If this is the case, DEVICE NOT READY is returned to the user, and more work is looked for. Otherwise, control is returned to the user.

3.10 Macro Definitions

The two macros defined in the receiver driver, FINSTS and HNDERR, are exactly as described in 2.8 for the transmitter driver.

4. DATA STRUCTURE - USRTB

The data structure module, USRTB, provides the RSX-11M device data structures for both the PCL transmitter and receiver. The data structures are formatted as described in the RSX-11M Guide to Writing an I/O Driver. Only fields pertaining to the PCL are described in the subsections which follow.

4.1 Conditional Assembly for Multiple Units

Through the specification of parameters defining the number of PCL transmitters and receivers, and their status register and vector addresses, the USRTB module can be assembled to provide data structures for any number of PCL transmitters and receivers on a system.

For the transmitter, the following symbol definitions are utilized:

```
L$$T11  defines the number of transmitters in the system.
LTSRnn  defines the TSR address for unit nn.
LTVTnn  defines the interrupt vector address for unit nn.
```

For the receiver, the following symbol definitions are utilized:

```
L$$R11  defines the number of receivers in the system.
LRSRnn  defines the RSR address for unit nn.
LRVTnn  defines the interrupt vector address for unit nn.
```

The unit number nn is expressed as an octal number in the range 0 through 17 (the current RSX-11M limit) with the leading zero omitted for units 0 through 7. If none of the above symbols are defined, the following default values are assumed:

```
L$$T11 = 1
LTSR0  = 164202
LTVT0  = 170
L$$R11 = 1
LRSR0  = 164222
LRVT0  = 174
```

If the defaults are not suitable, symbol definitions should be inserted into the file [200,200] RSXMC.MAC as described in the driver installation procedures given in the RSX-11M PCL User's Guide.

4.2 Receiver DCB

The device name for the PCL receiver is defined as "LR". The function masks are defined according to the following table.

LEGAL FUNCTION	OCTAL CODE	MASK BIT VALUE*	OTHER MASKS SET
KIL	0	1,0	
CRX	31	0,1000	control
RTF	32	0,2000	control
ATF	33	0,4000	
DRX	34	0,10000	control

4.3 Transmitter DCB

The device name for the PCL transmitter is defined as "LT". The function masks are defined according to the following table.

LEGAL FUNCION	OCTAL CODE	MASK BIT VALUE*	OTHER MASKS SET
KIL	0	1,0	
ATT	3	10,0	control
DET	4	20,0	control
ATX	30	0,400	

*he mask bit value is given in the from "a,b" where a is the word for codes 15, and b is for codes 16-31.

4.4 Receiver UCBs

One UCB is defined for each receiver unit. They are identical to each other except for the U.UNIT and U.SCB fields, which vary for each unit. The U.CTL field has the following bits set:

```
UC.ALG   to require word aligned buffers;
UC.KIL   to cause the driver to be entered on all KIL
         functions, even when the unit is not busy;
UC.PWF   to cause the driver to be entered after all
         power failures, even when the unit is not busy;
UC.NPR   to cause U.BUF to be formatted for NPR devices;
UC.LGH/3 to require buffer sizes which are divisible by 2
           (but not necessarily 4).
```

Each UCB contains 5 extra words of work space. These are defined in 3.2.2.

4.5 Transmitter UCBs

One UCB is defined for each transmitter unit. They are identical to each other except for the U.UNIT and U.SCB fields, which vary for each unit. The U.CTL field has the following bits set:

```
UC.ALG   to require word aligned buffers;
UC.NPR   to cause U.BUF to be formatted for NPR devices;
UC.LGH/3 to require buffer sizes which are divisible by 2
           (but not necessarily 4).
```

Each UCB contains 5 extra words of work space. It is defined in 2.2.2.

4.6 Receiver SCBs

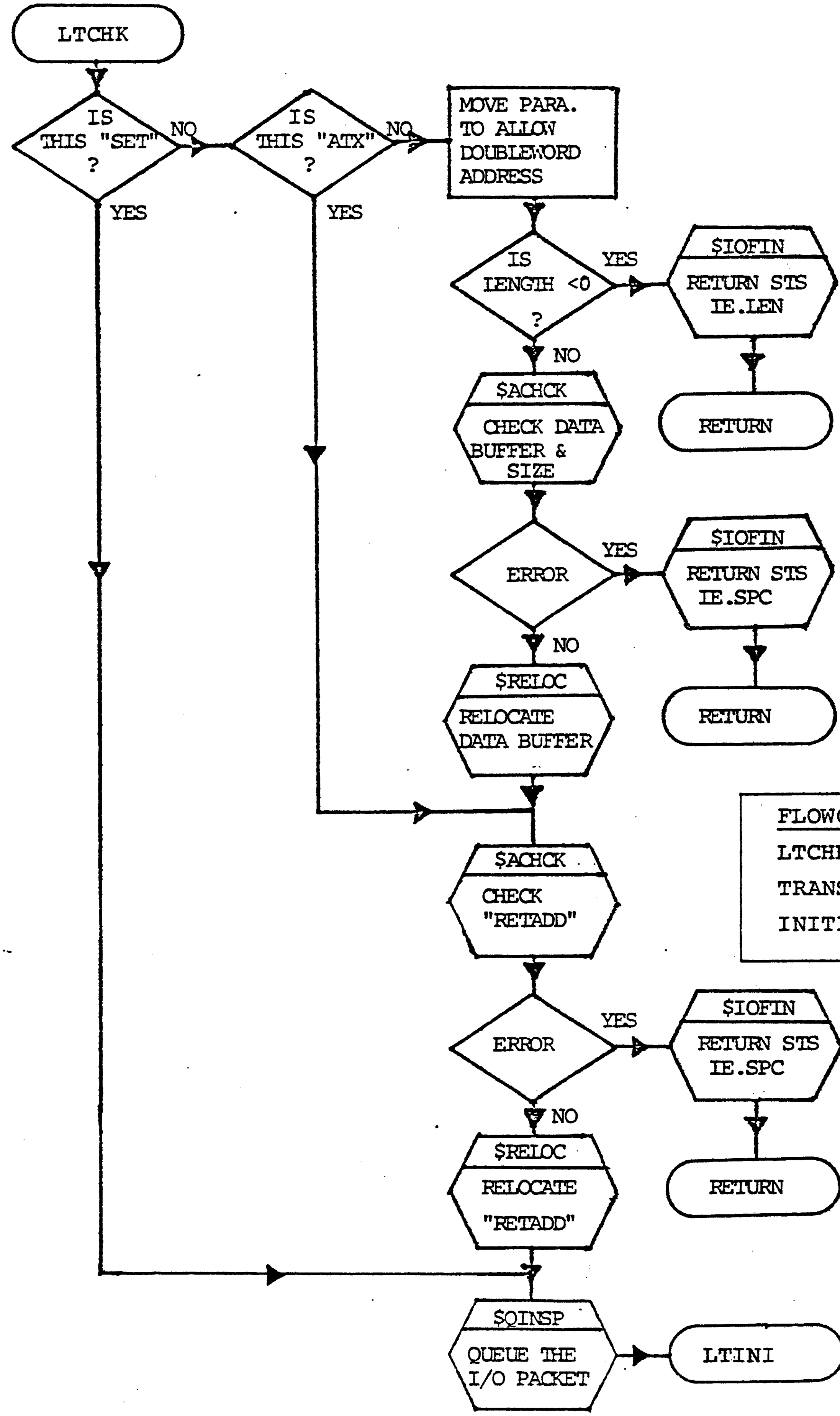
One SCB is defined for each receiver unit, since each unit will have a distinct controller. The SCB's are identical to each other except for the S.VCT, S.CON, and S.CSR fields, which vary for each unit.

4.7 Transmitter SCBs

One SCB is defined for each transmitter unit, since each unit will have a distinct controller. The SCB's are identical to each other except for the S.VTC, S.CON, and S.CSR fields which vary for each unit.

4.8 Interrupt Vectors

The interrupt vectors are set up for each unit of transmitter and receiver. The RSX-11M convention of having the low order 4 bits of the second vector word contain the unit number of the device has been followed.

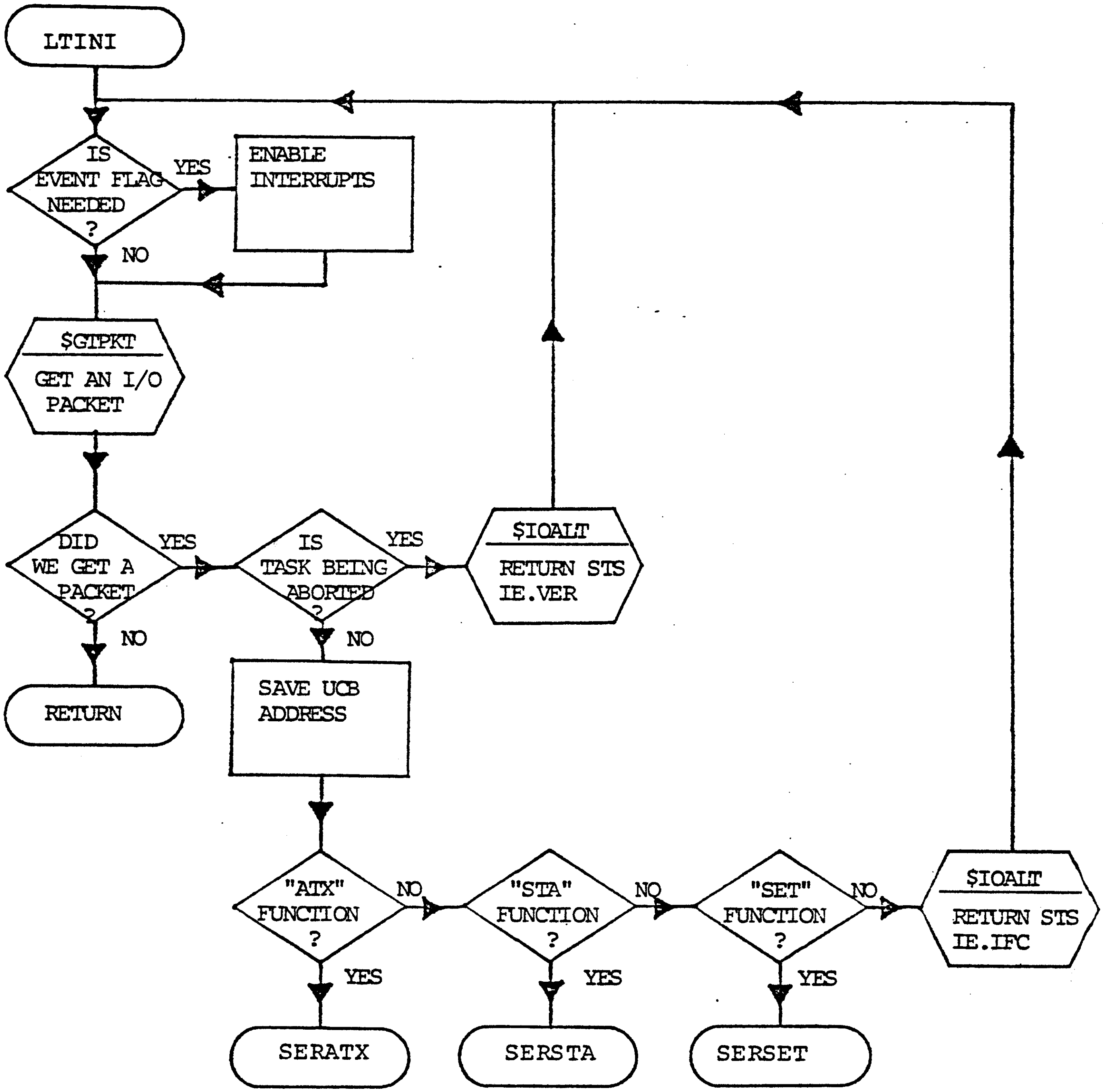


FLOWCHART 1.

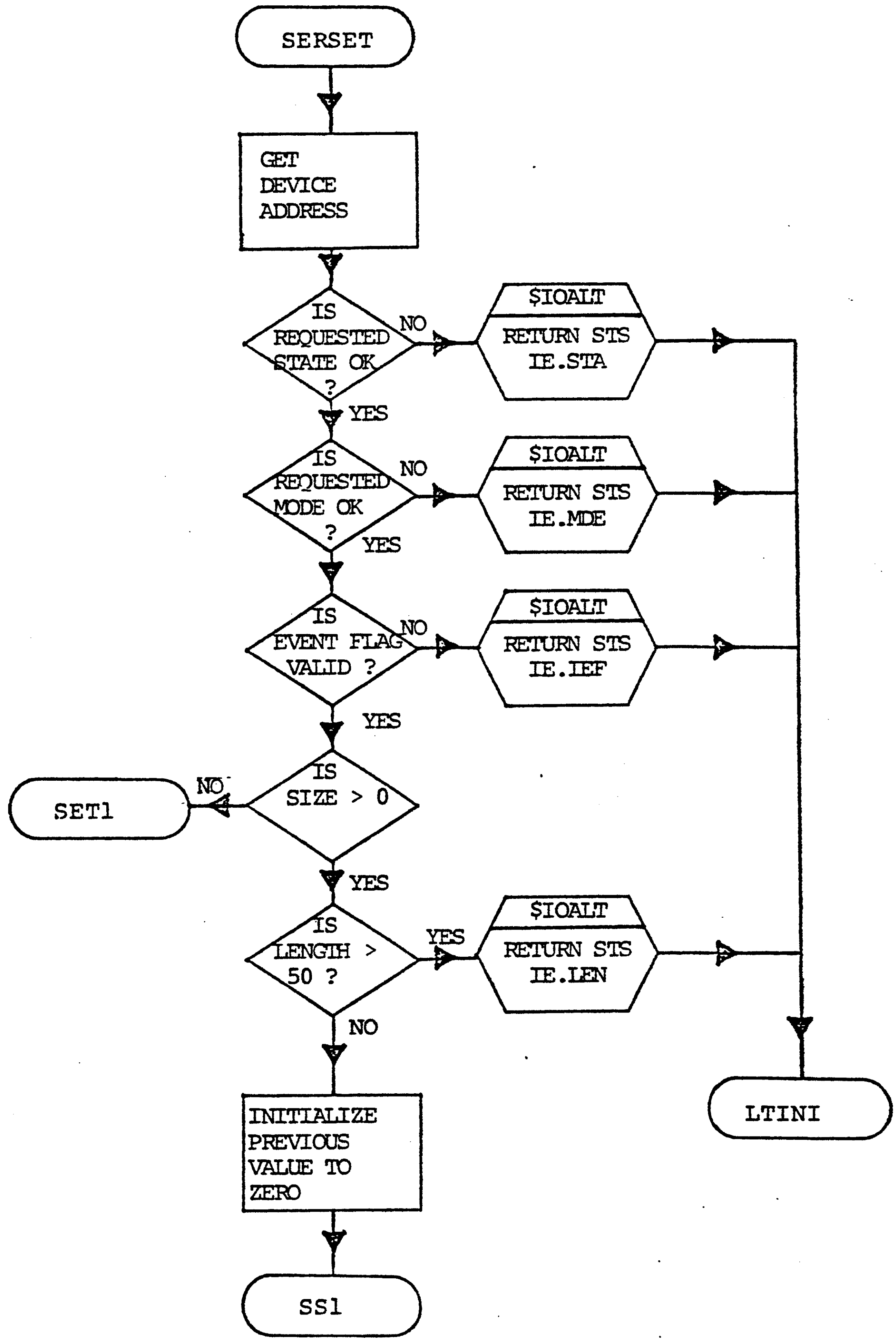
LTCHK

TRANSMITTER DRIVER

INITIATOR



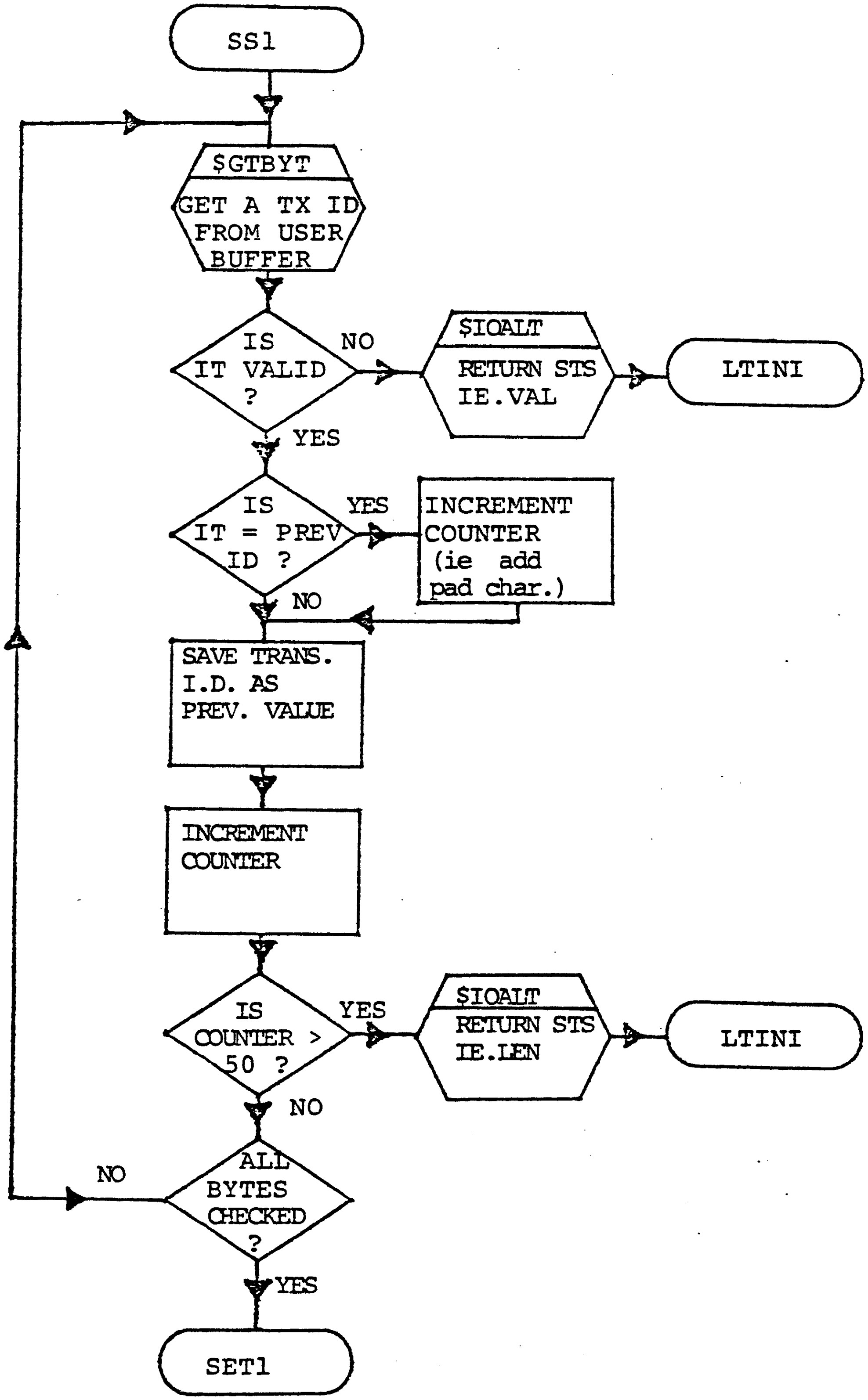
FLOWCHART 2. LTINI
INITIATOR SERVICE ROUTINE



FLOWCHART 3. SERSET

"SET" FUNCTION SERVICE ROUTINE

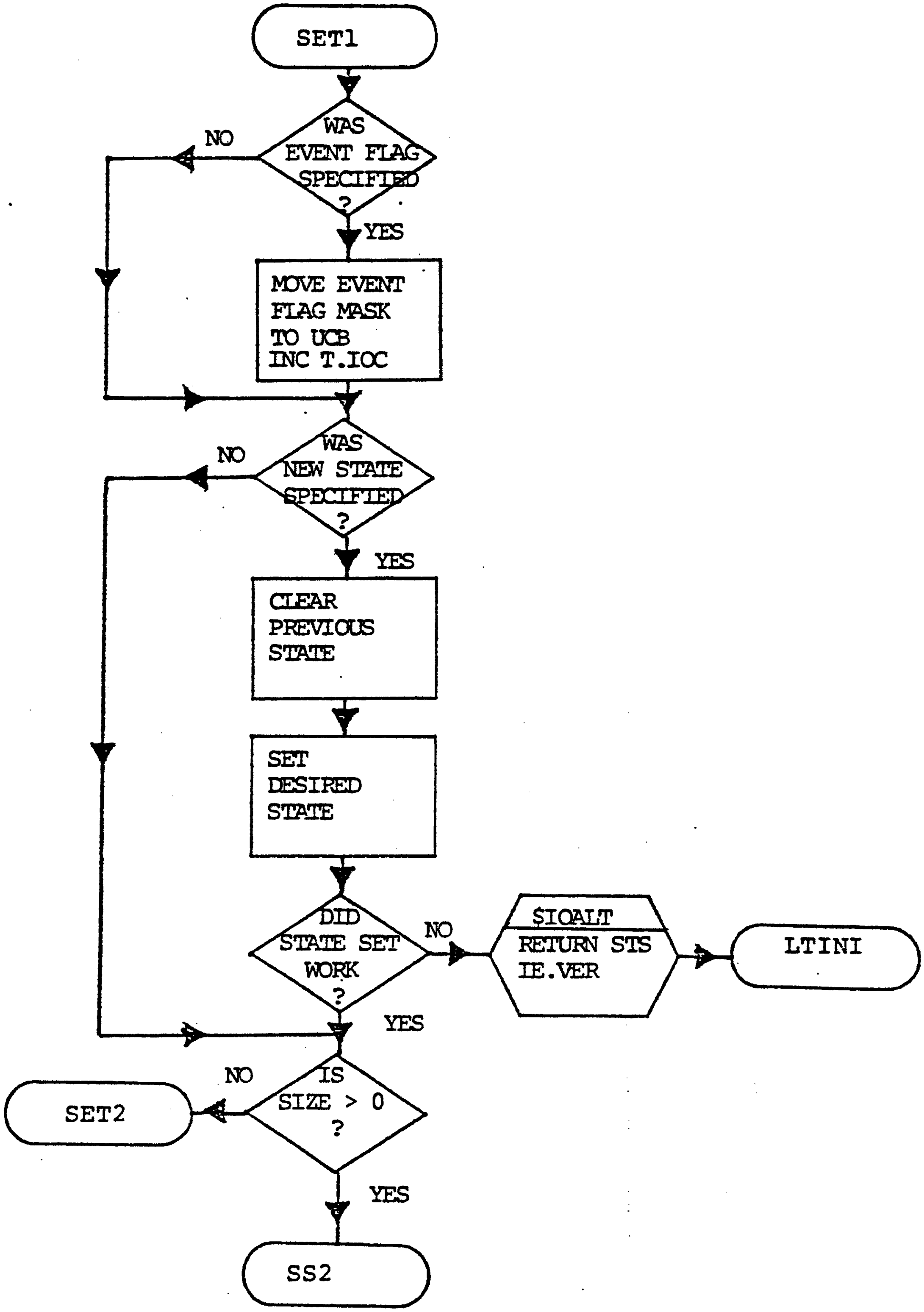
PART 1 of 5



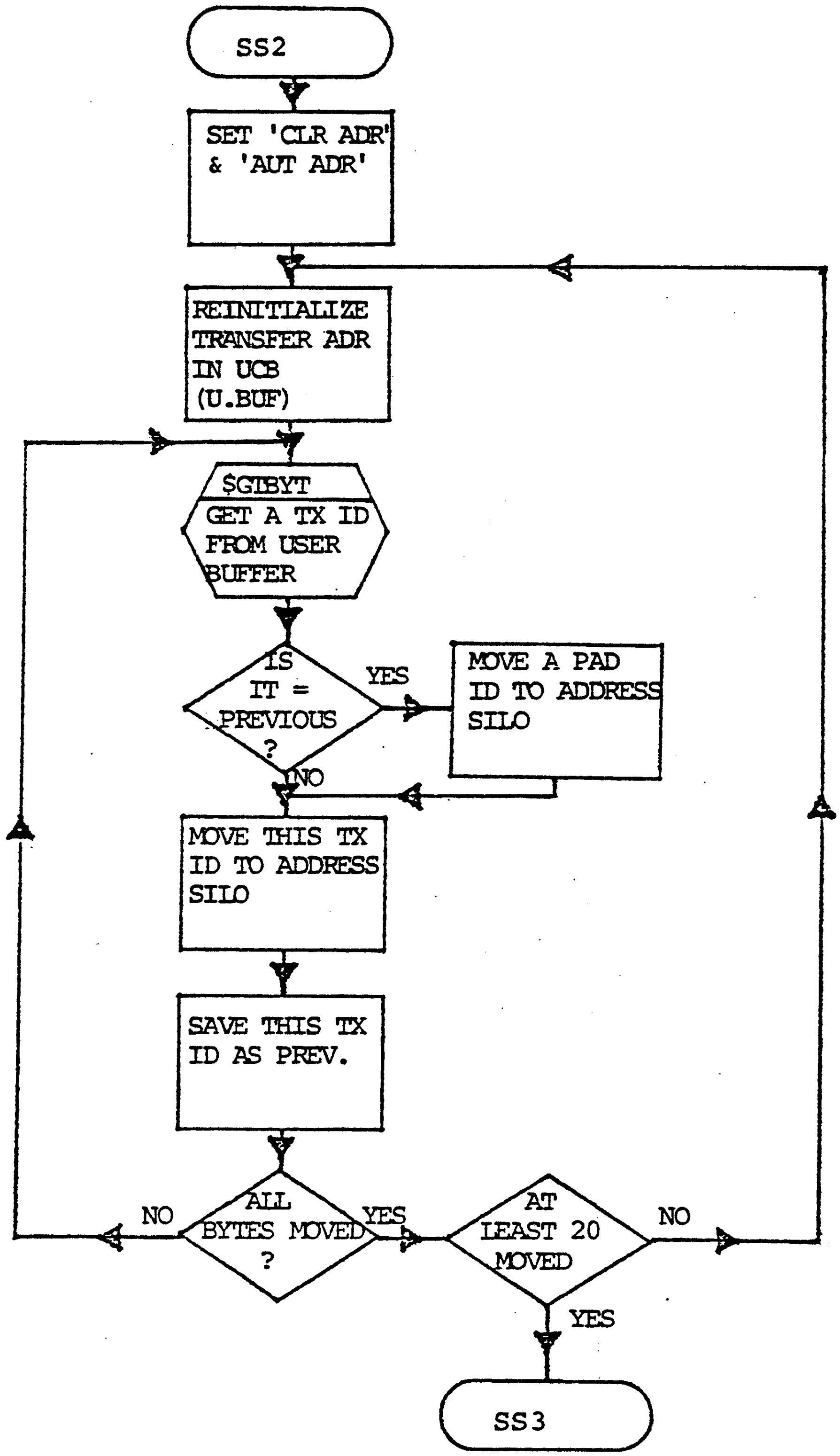
FLOWCHART 3. SERSET

"SET" FUNCTION SERVICE ROUTINE

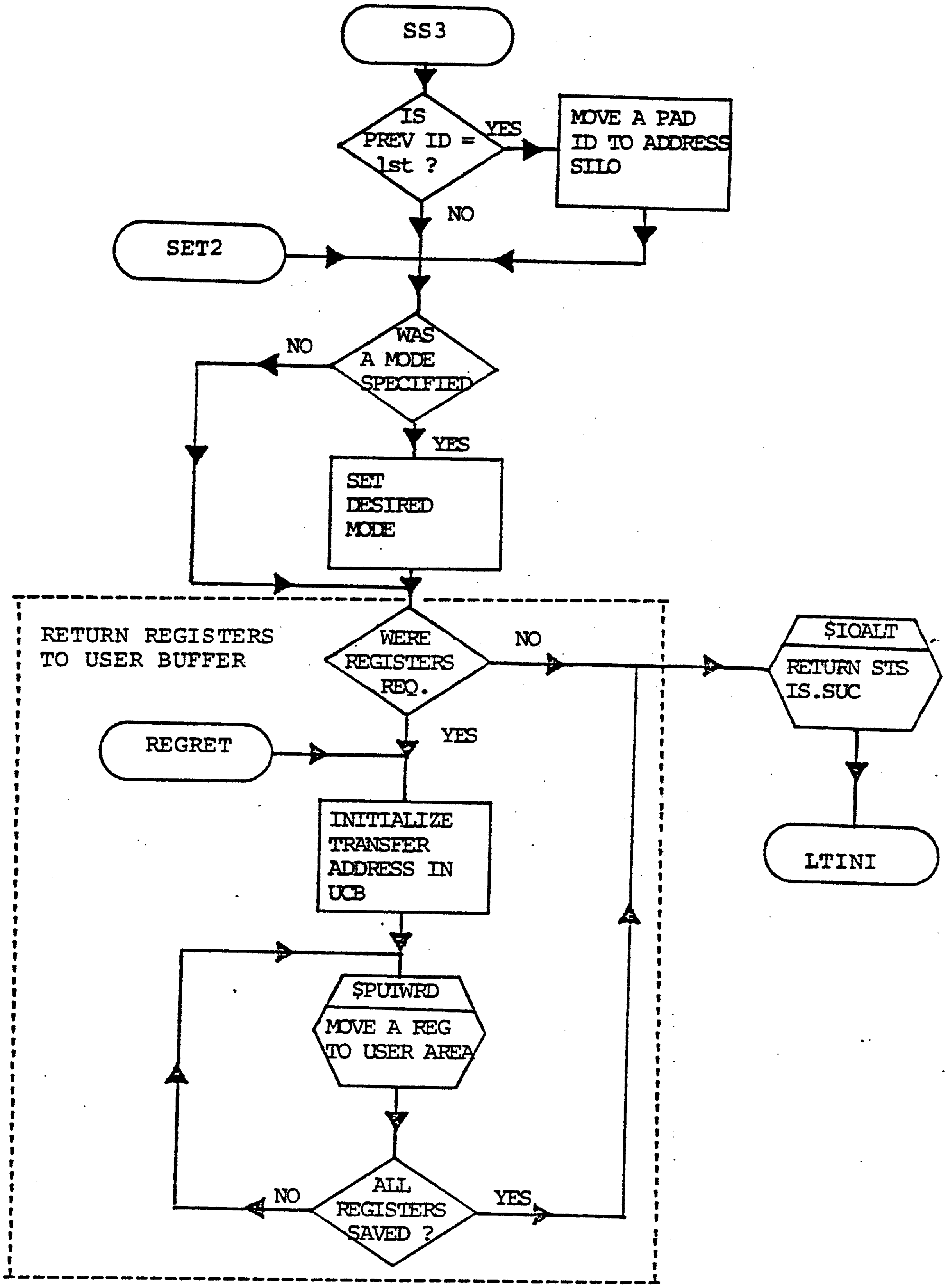
PART 2 of 5



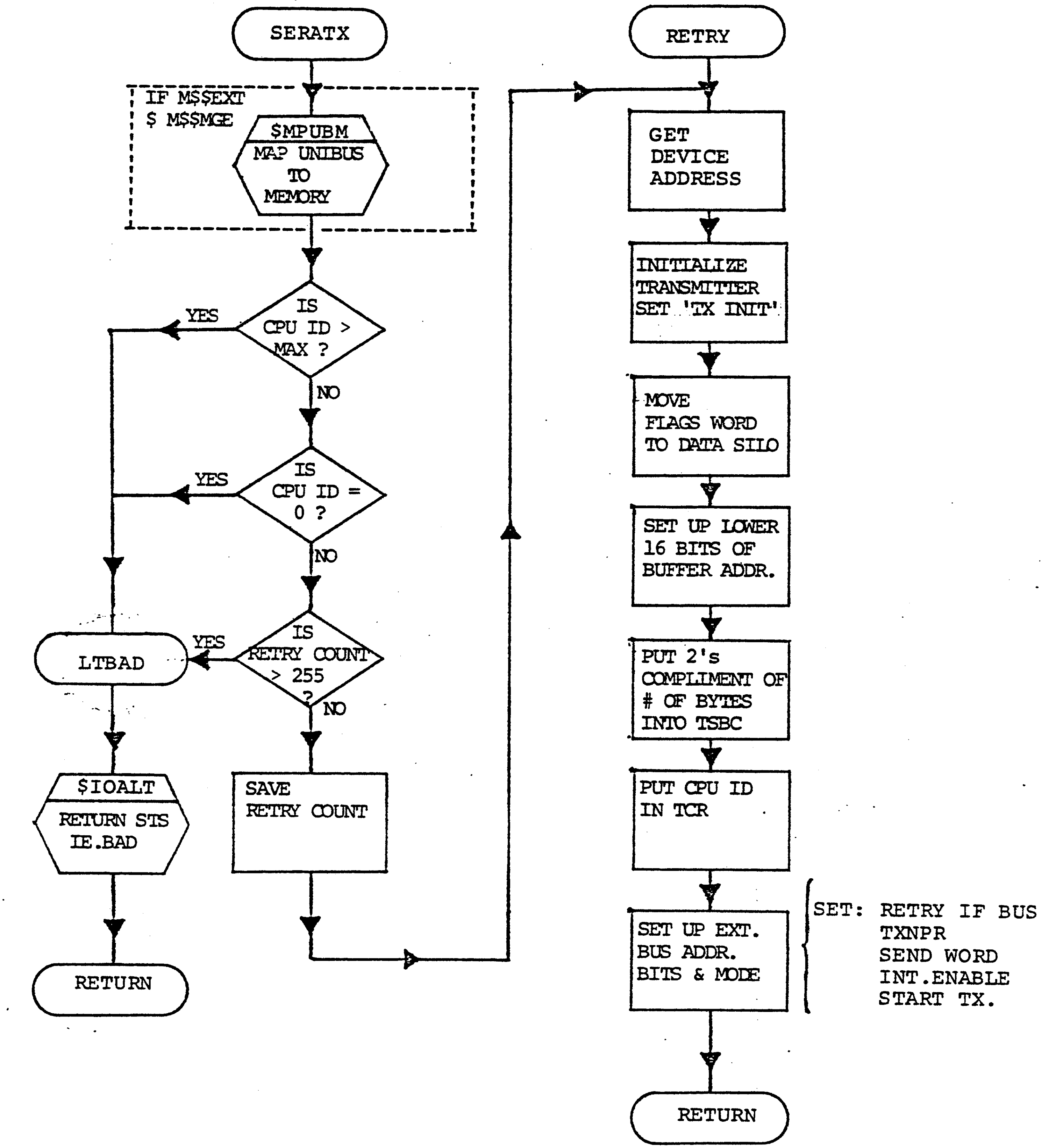
FLOWCHART 3. SERSET
"SET" FUNCTION SERVICE ROUTINE
PART 3 of 5



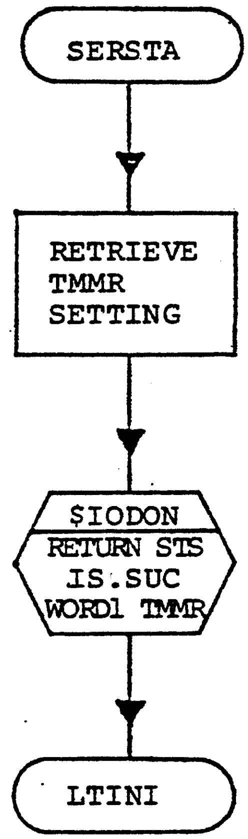
<u>FLOWCHART 3.</u> SERSET
"SET" FUNCTION SERVICE ROUTINE
PART 4 of 5



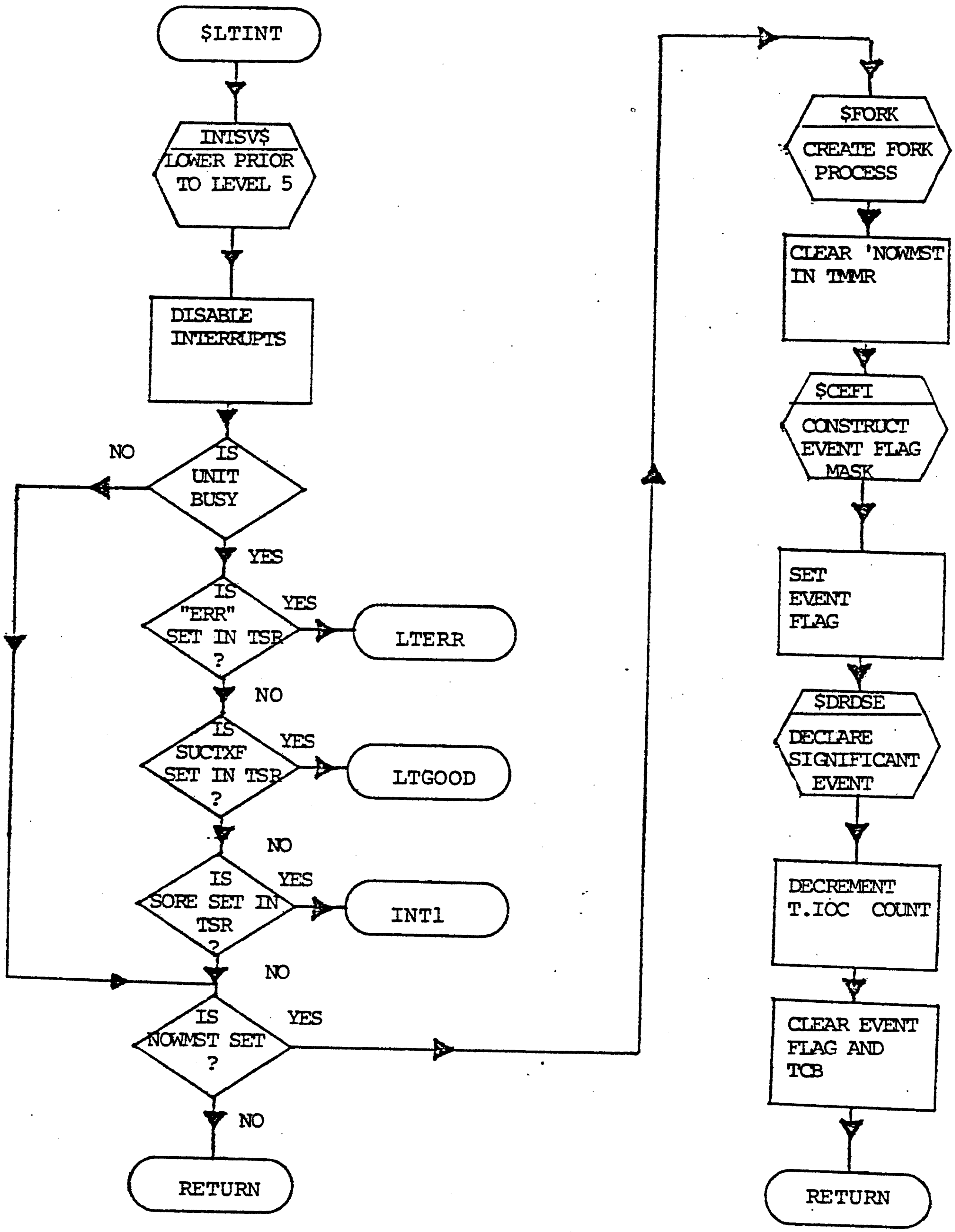
FLOWCHART 3. SERSET
"SET" FUNCTION SERVICE ROUTINE
PART 5 of 5



FLOWCHART 4. SERATX
"ATX" FUNCTION SERVICE ROUTINE



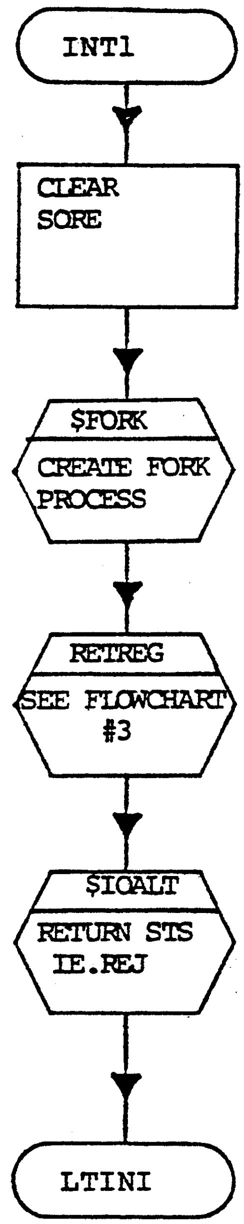
FLOWCHART 5. SERSTA
"STA" FUNCTION SERVICE ROUTINE



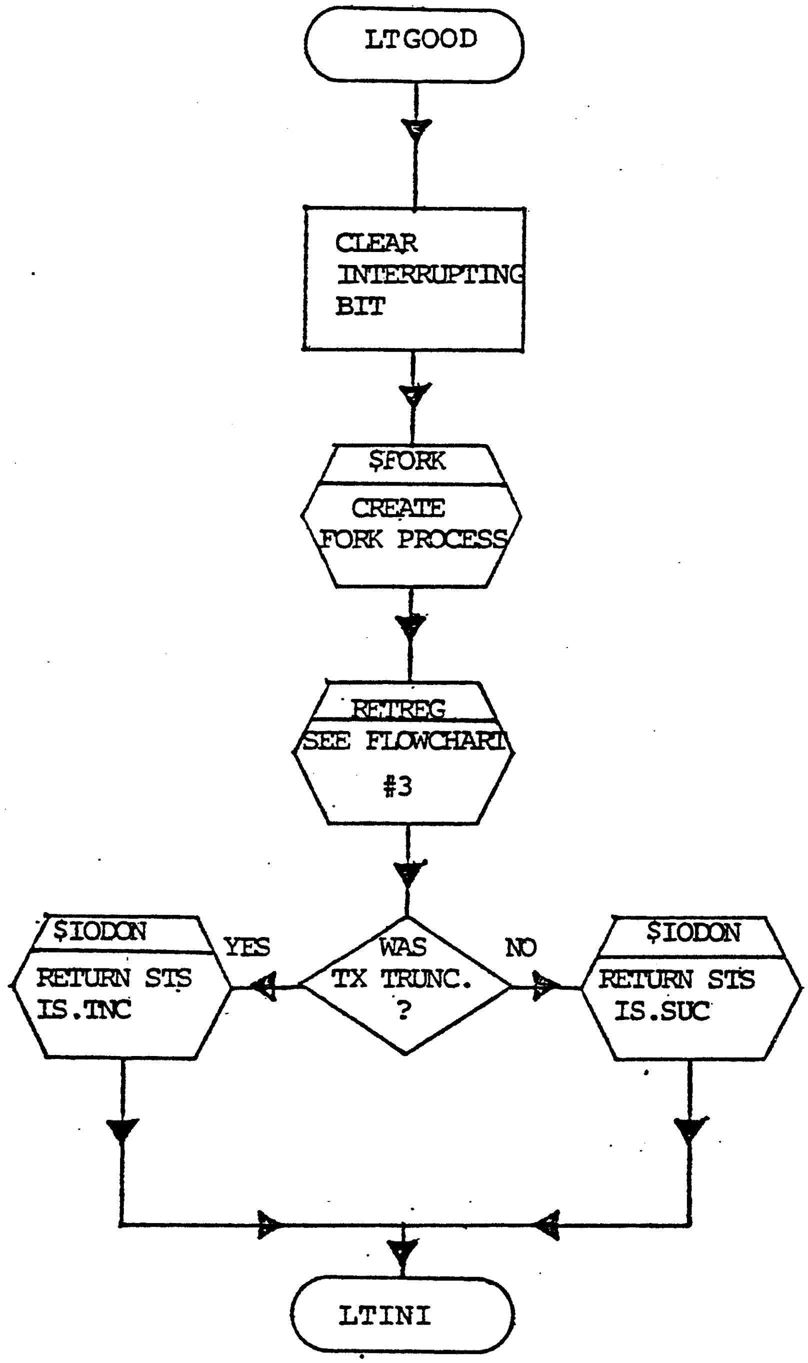
FLOWCHART 6. $LTINT

INTERRUPT SERVICE ROUTINE

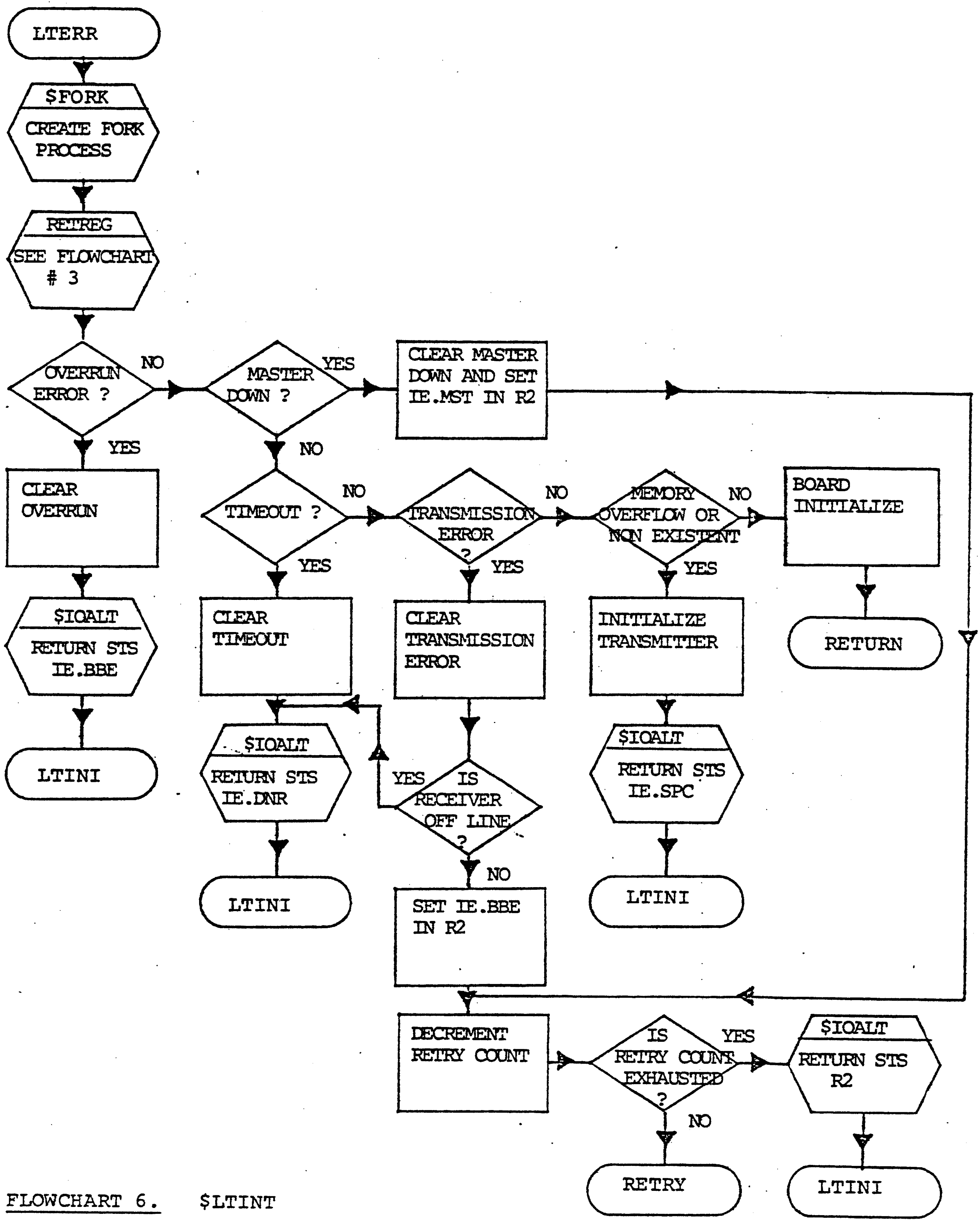
PART 1 of 4



FLOWCHART 6. $LTINT
INTERRUPT SERVICE ROUTINE
PART 2 of 4



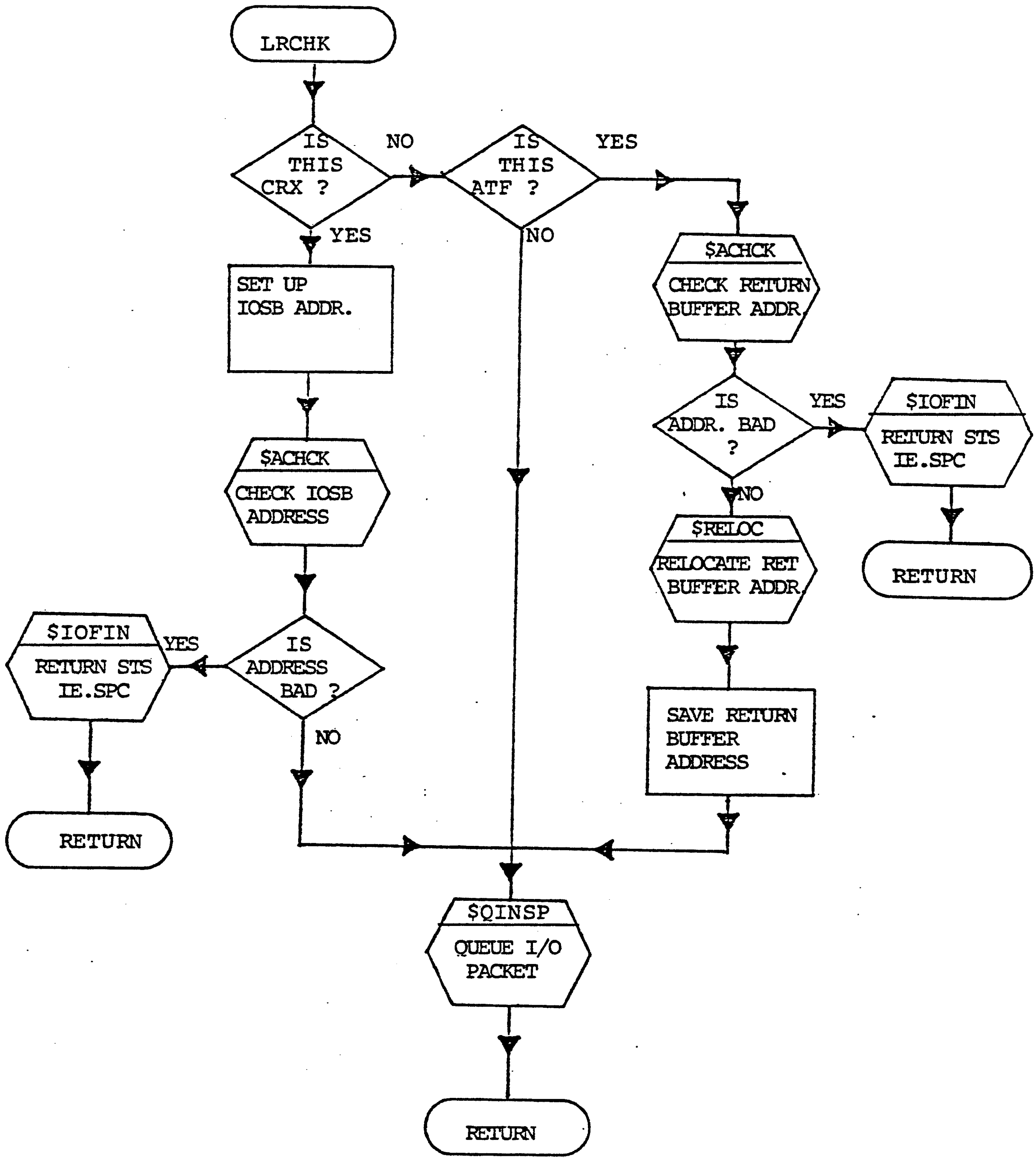
FLOWCHART 6. $LTINT
INTERRUPT SERVICE ROUTINE
PART 3 of 4



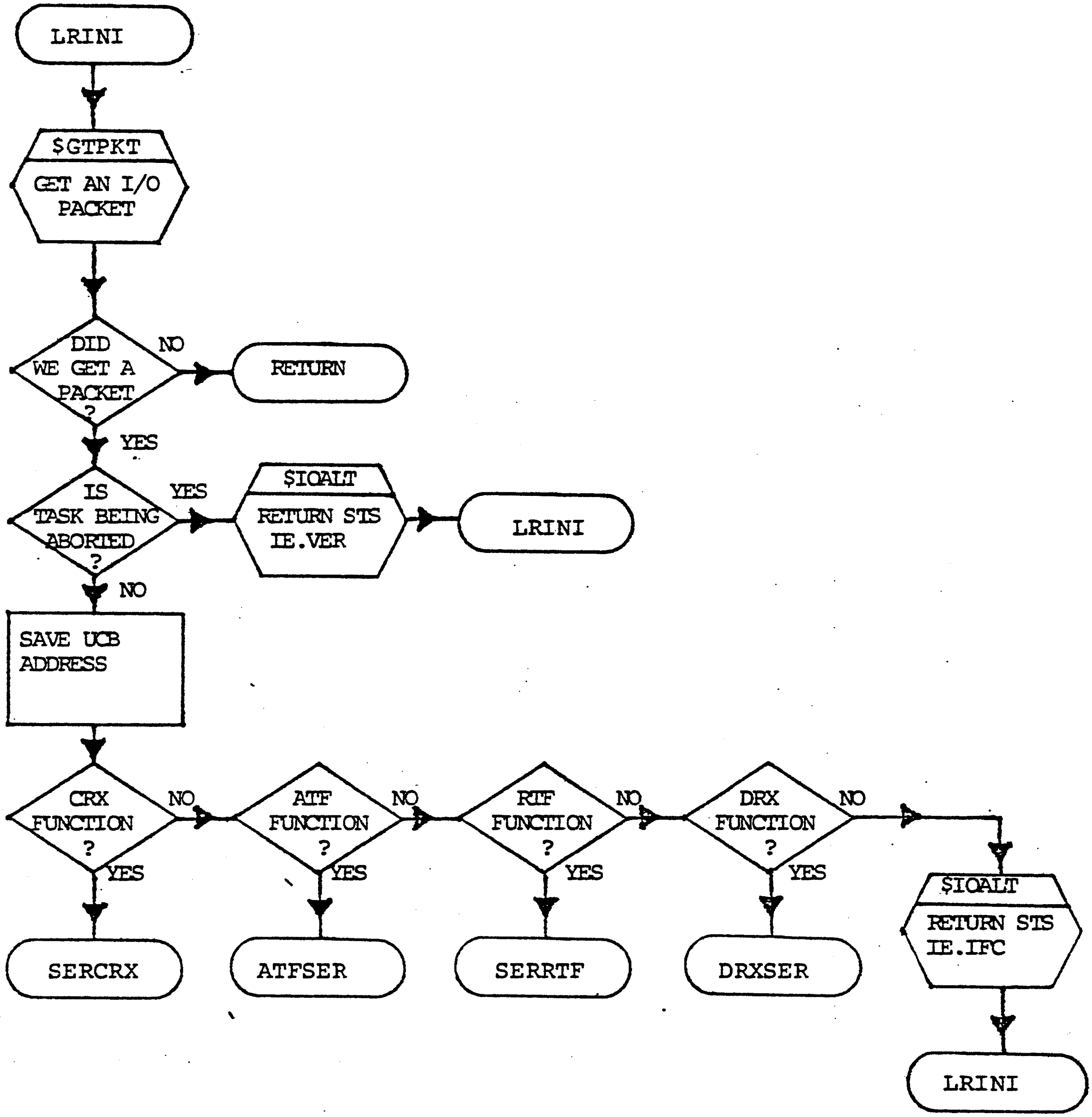
FLOWCHART 6. $LTINT

INTERRUPT SERVICE ROUTINE

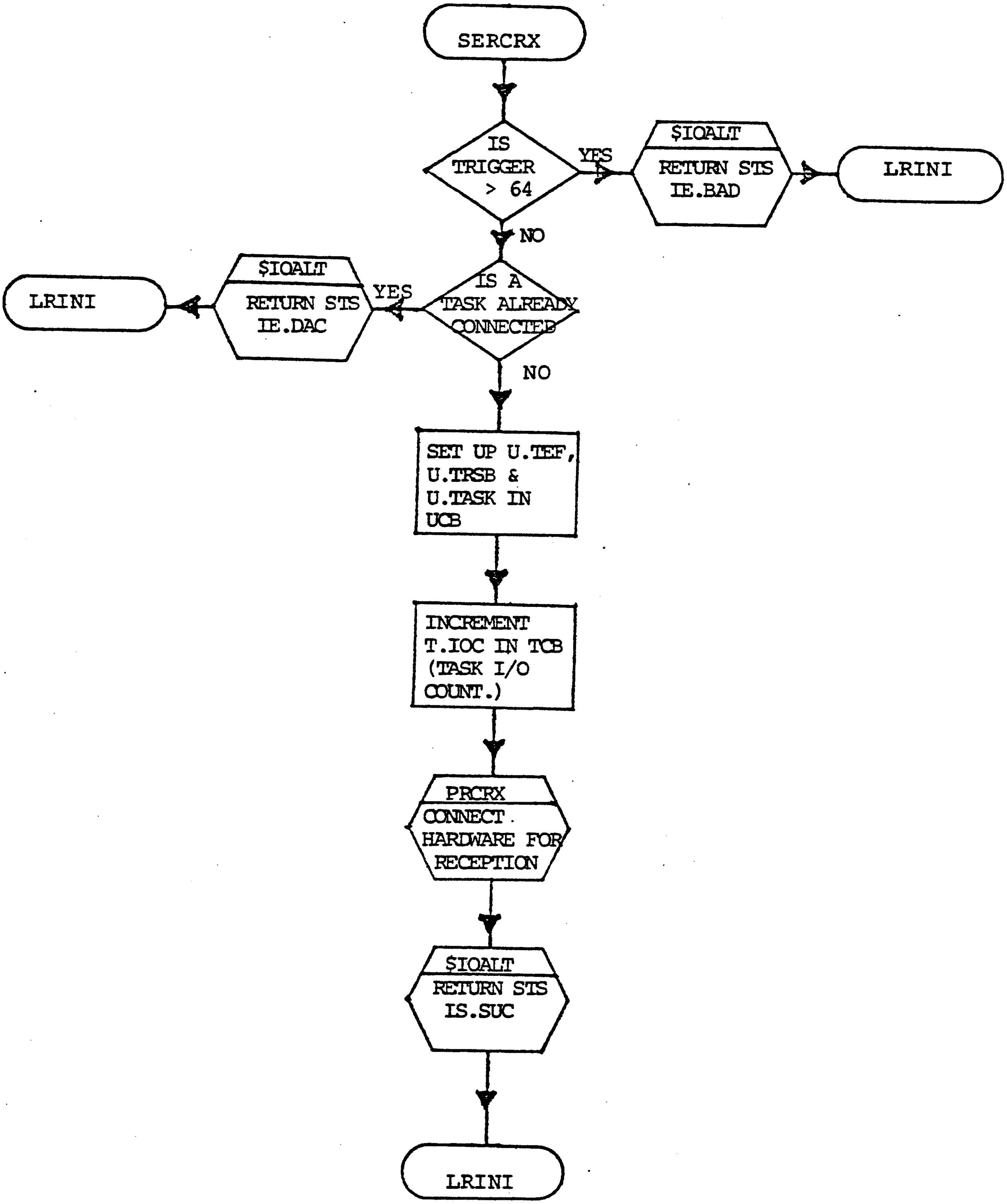
PART 4 of 4



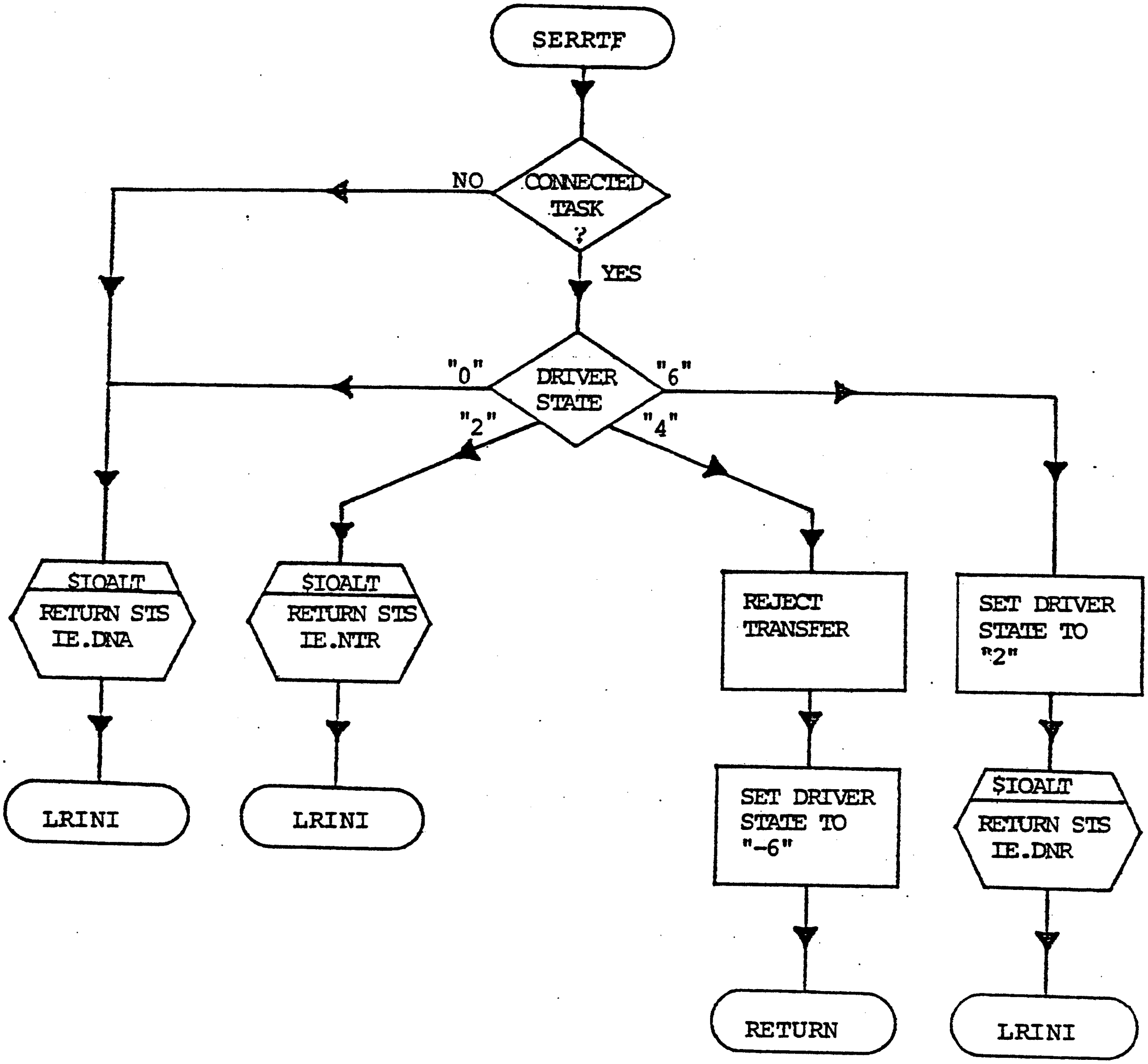
<u>FLOWCHART 7.</u> LRCHK
RECEIVER DRIVER
INITIATOR



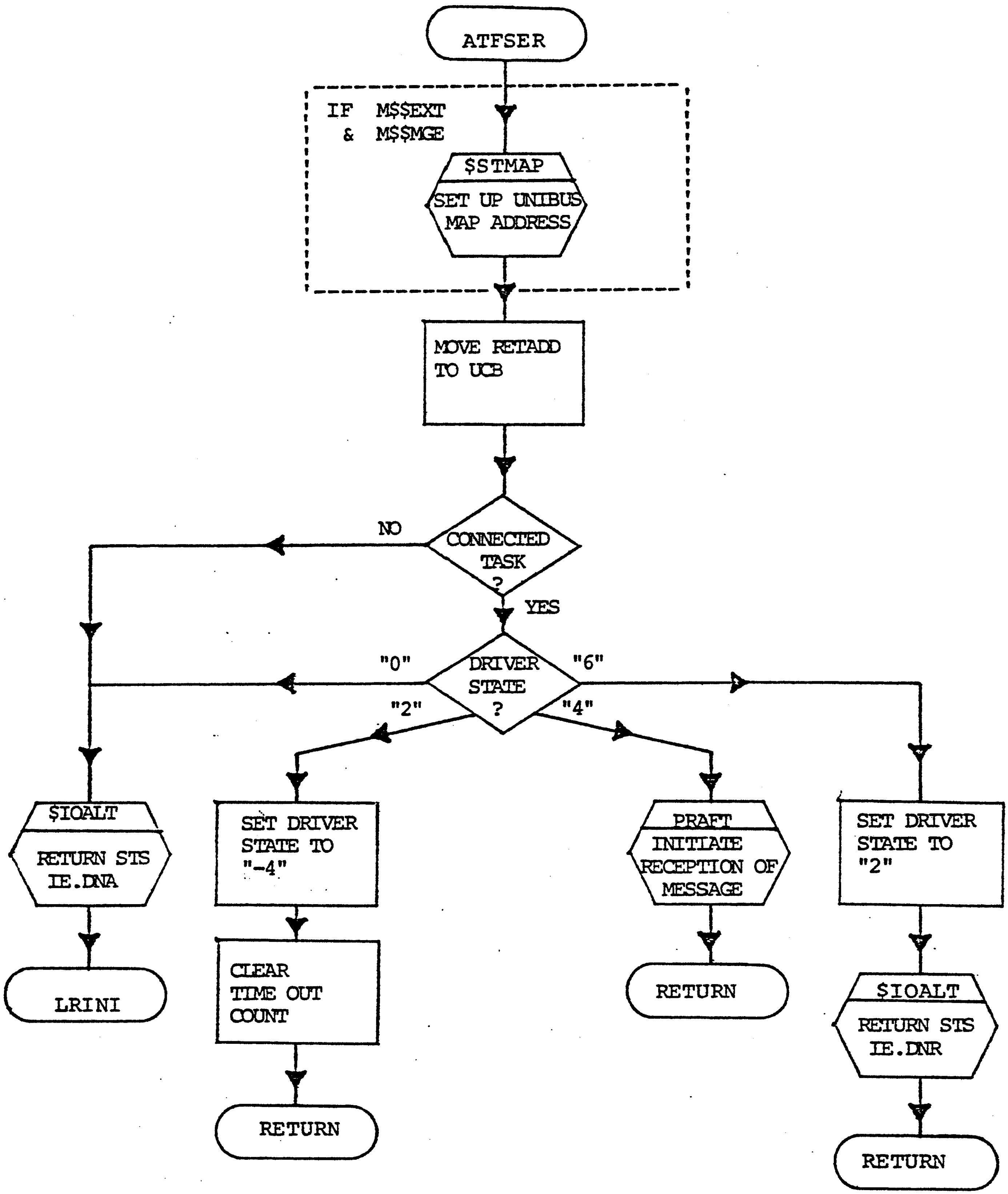
FLOWCHART 8. LRINI
INITIATOR SERVICE ROUTINE



FLOWCHART 9. SERCRX
"CRX" FUNCTION SERVICE ROUTINE

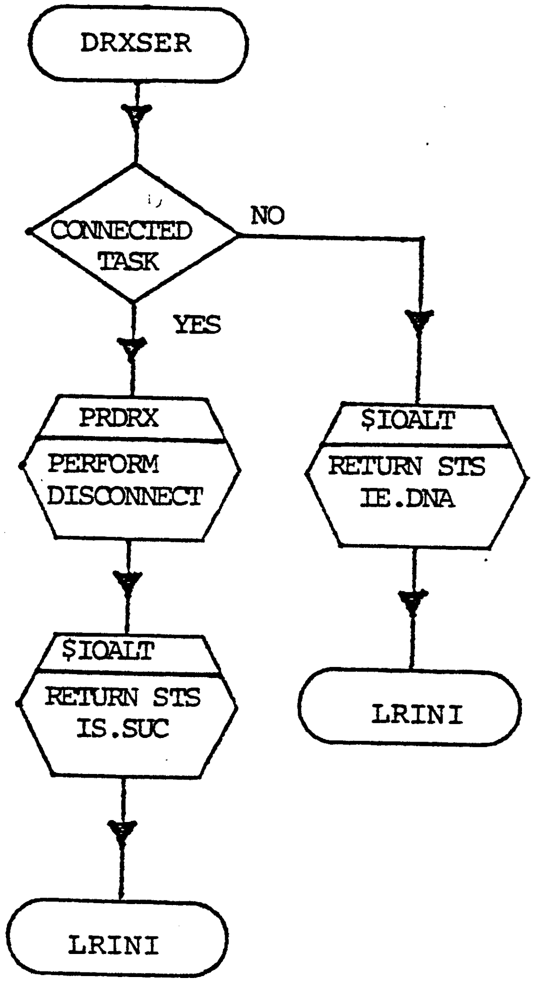


FLOWCHART 10. SERRTF
"RTF" FUNCTION SERVICE FUNCTION



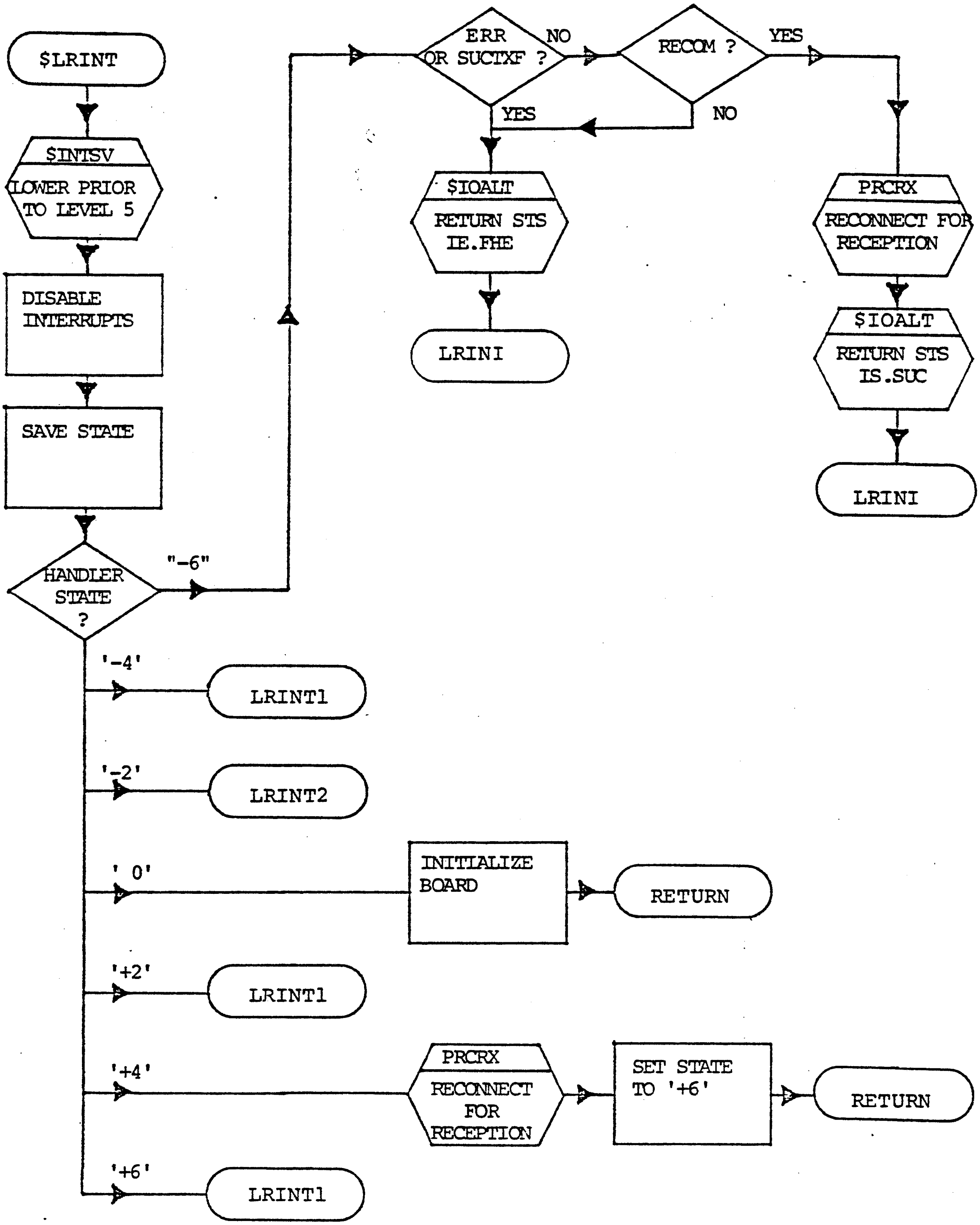
FLOWCHART 11. ATFSER

"ATF" FUNCTION SERVICE ROUTINE



FLOWCHART 12. DRXSER

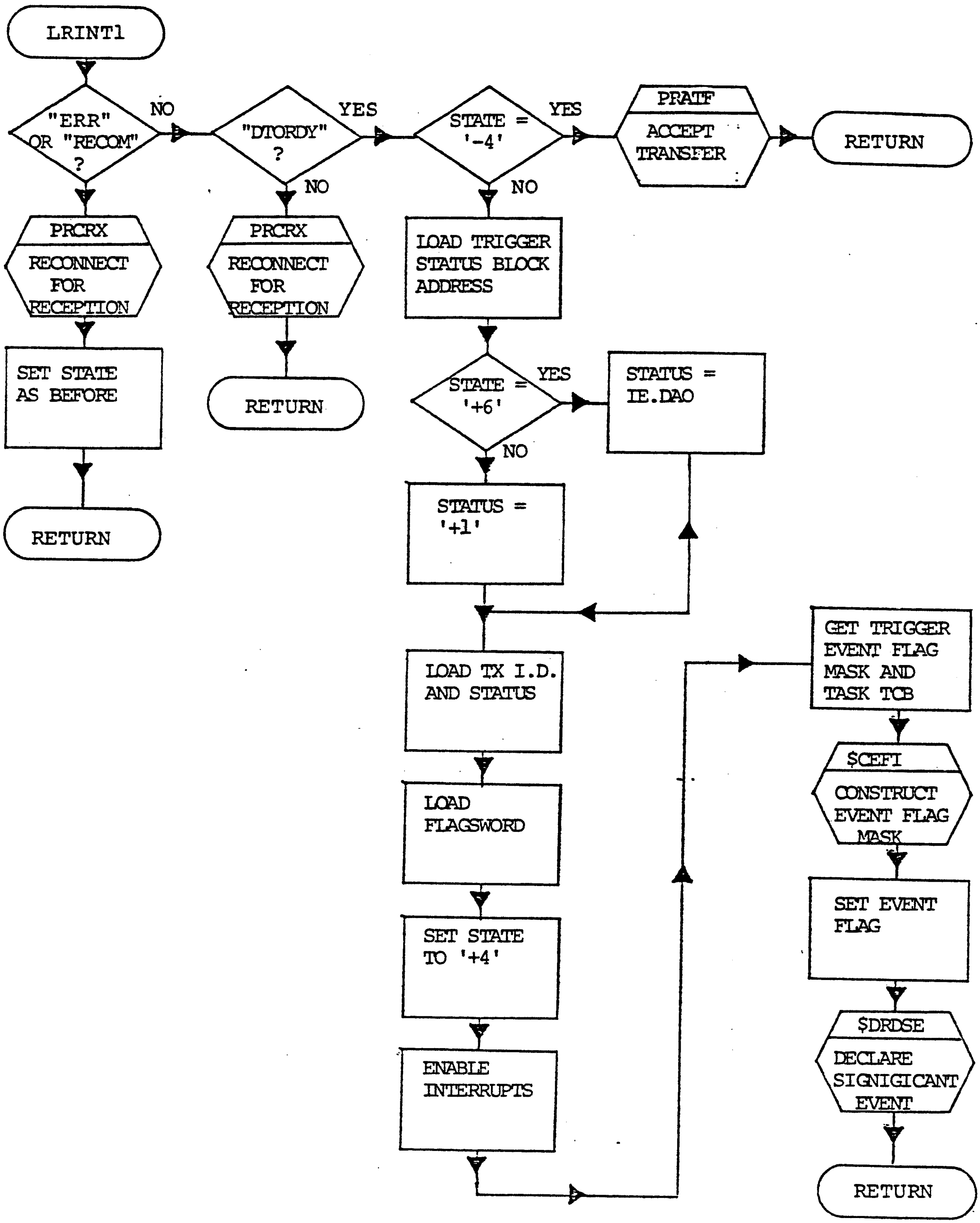
"DRX" FUNCTION SERVICE ROUTINE



FLOWCHART 13. $LRINT

INTERRUPT SERVICE ROUTINE

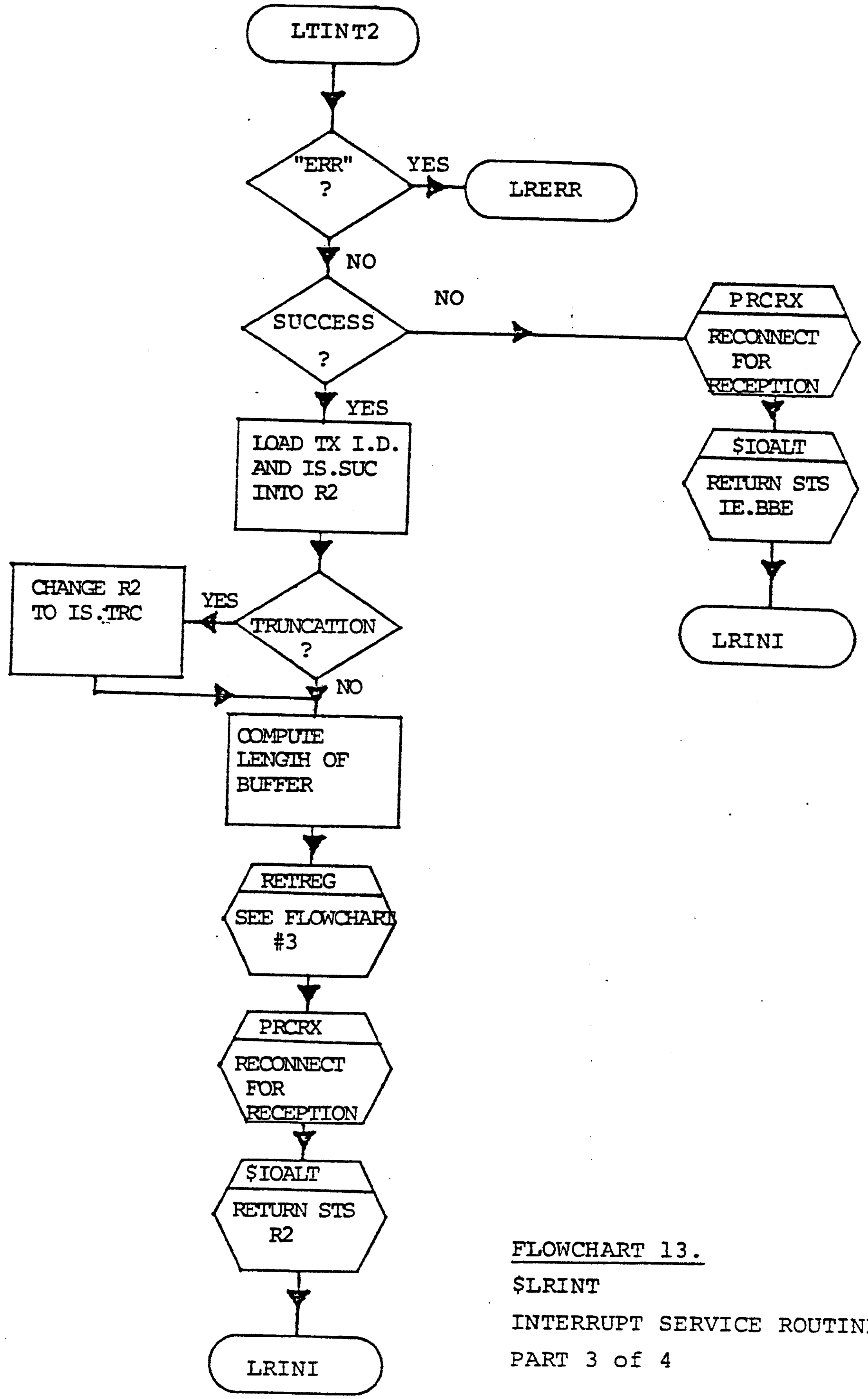
PART 1 of 4



FLOWCHART 13. $LRINT

INTERRUPT SERVICE ROUTINE

PART 2 of 4

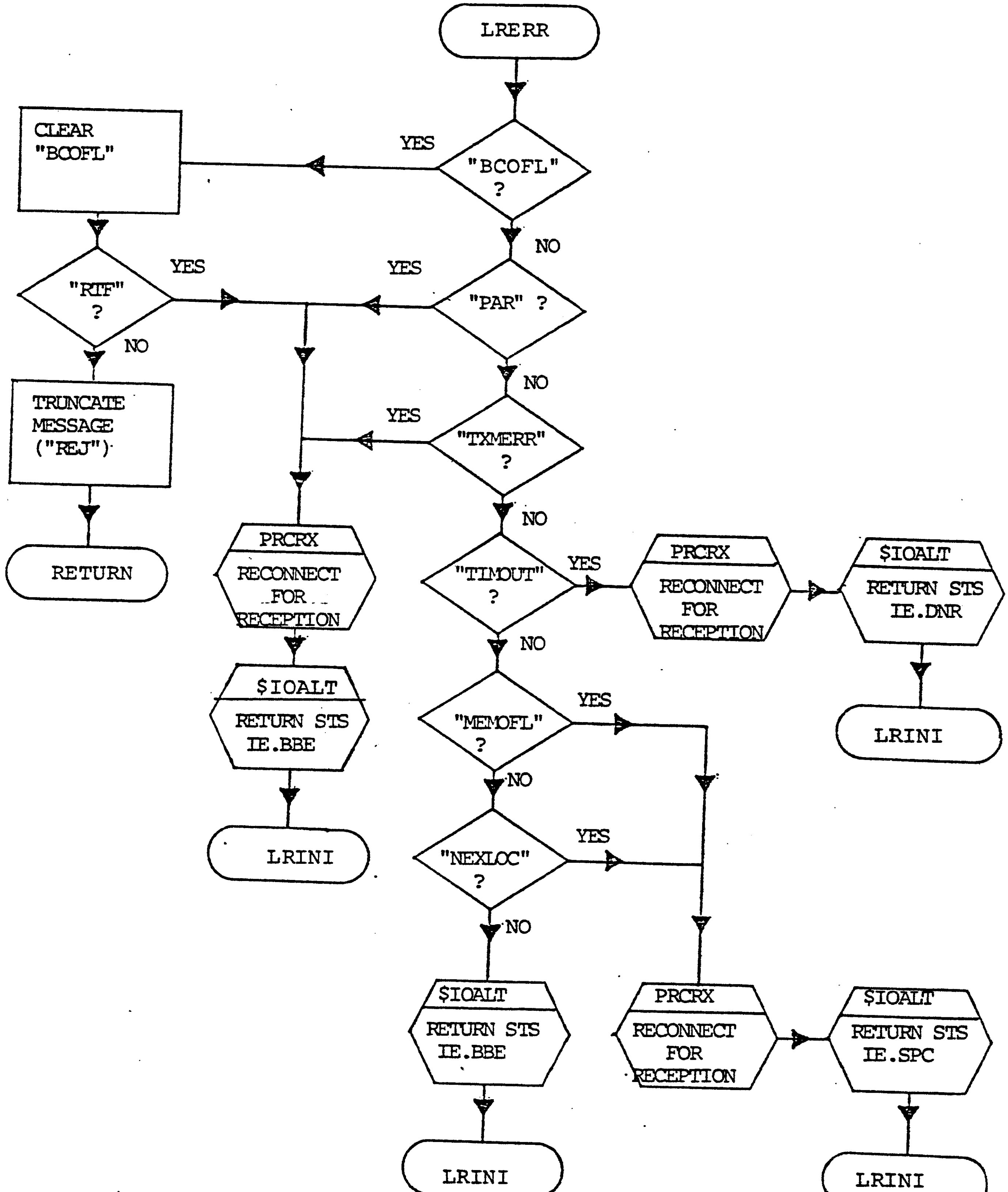


FLOWCHART 13.

$LRINT

INTERRUPT SERVICE ROUTINE

PART 3 of 4



FLOWCHART 13. $LRINT
INTERRUPT SERVICE ROUTINE
PART 4 of 4