CT
ACD
34094
14

DOT/FAA/CT-ACD34094/14

PAGE CHANGES TO

ARTS IIIA PROGRAM CODING SPECIFICATIONS

VOLUME III OF III

FOR

ARTS IIIA FINAL APPROACH SPACING TOOL PHASE 1 CONFIGURATION CONTROL DECISIONS (CCD) T15712 AND T16295 CASE FILES CD300-CPT-OO3A and CD300-AR3A-003

A3.05

SEPTEMBER 13, 1993

FEDERAL AVIATION ADMINISTRATION
ATC TECHNOLOGY BRANCH, ACD-340
ATLANTIC CITY INTERNATIONAL AIRPORT, N.J., 08405

TABLE OF CONTENTS (Continued)

Section			Page
	3.3.84	Calculate DBAT Hashed Index Location (DBATHASH)	3.3.84–1
	3.3.85	Push DBAT Insert Entry Onto the DBAT Processing Stack (DBATSTKI)	3.3.85–1
	3.3.86	Push DBAT Delete Entry Onto the DBAT Processing Stack (DBATSTKD)	3.3.86–1
	3.3.87	Delete TI Beacon Code From DBAT (CHKDBAT)	3.3.87–1
	3.3.88	Delete Assigned Beacon Code From DBAT (DBATTST)	3.3.88–1
	3.3.89	Test TI Beacon Code/ABC Request (DBATCOM)	3.3.89–1
	3.3.90	TUD Sensor Stack Handler (TUDSENSTK)	3.3.90–1
	3.3.91	Home Trackball (HOMETB)	3.3.91–1
	3.3.92	Pack Ghosting Application Buffer (PGABF)	3.3.92–1
	3.3.93	Pack Delta (PKDELTA)	3.3.93–1
	3.3.94	Pack Late HO (PKHO)	3.3.94–1
	3.3.95	Re-Output CENRAP Data (CPREOUT)	3.3.95–1
3.4	DATABASE		3.4.1–1
	3.4.1	Central Track Store Table (CTS)	3.4.1–1
	3.4.2	System Database 1	3.4.2–1
	3.4.3	System Database 2 — SDB2	3.4.3–1
	3.4.4	Multiprocessor Temporary Storage (TMP) (i = 1–5)	3.4.4–1
	3.4.5	MSAW Database (MSAWD)	3.4.5–1
	3.4.6	Altitude Tracking Database	3.4.6–1
	3.4.7	Interfacility Data (IDAT)	3.4.7–1
	3.4.8	Remote Display Database (RDBMDAT)	3.4.8–1
	3.4.9	CRDA Database (CRDAD)	3.4.9–1
	3.4.10	MDBM Output Chain Buffer (MBUFi)	3.4.10–1
	3.4.11	Teletype Database (TTYD)	3.4.11–1
	3.4.12	CDR Database (CDRD)	3.4.12–1
	3.4.13	Critical Database (CRITD)	3.4.13–1
	3.4.14	MSAW Map Tables (TMAPi) (i = 1,2)	3.4.14–1
	3.4.15	Performance Monitor Data (PERDA)	3.4.15–1
	3.4.16	Track Record Tables (TRT and TRT1)	3.4.16–1
	3.4.17	A027	3.4.17–1
	3.4.18	MT28 Database	3.4.18–1
	3.4.19	Keyboard Input Processing Data (KIPD)	3.4.19–1
	3.4.20	Local Display Output Variables (DOPV)	3.4.20–1
	3.4.21	Scratch Pad Timing Database (SDAT)	3.4.21–1
	3.4.22	Purposely Removed	3.4.22–1
	3.4.23	Console Typewriter Input Processing Data (CTIPSI)	3.4.23–1
	3.4.24	Tracking Output Database (TRDAT)	3.4.24–1
	3.4.25	Tracking Tables	3.4.25–1
	3.4.26	Data Type Record Table (DTRT)	3.4.26–1
	3.4.27	Conflict Alert Data	3.4.27–1
	3.4.28	System Database 3 — SDB3	3.4.28–1
	3.4.29	System Database 4 — SDB4	3.4.29–1
	3.4.30	CNFGD	3.4.30–1

INSERT ATTACHMENT 1 →

TABLE OF CONTENTS (Continued)

Section		*Page*
3.5	SITE ADAPTATION PARAMETERS	3.5.1-1
3.5.1	System Parameters (TI)	3.5.1-1
3.5.2	System Parameters (SYSEQ)	3.5.2-1
3.5.3	General Site Parameters (SITEQ)	3.5.3-1
3.5.4	CDR Site Parameters (CDR)	3.5.4-1
3.5.5	MSAW Parameters (MSAWQ)	3.5.5-1
3.5.6	Conflict Alert System Parameters	3.5.6-1
3.6	THIS SECTION HAS BEEN PURPOSELY REMOVED	3.6.1-1

ATTACHMENT 1

		Page
3.3.96	FAST Output Chain Buffer Request Subroutine (FSTRBUF)	3.3.96-1
3.3.97	FAST Transmit Buffer Subroutine (FSTRSND)	3.3.97-1
3.3.98	FAST Input Interrupt Handler (FSTREC) ...	3.3.98-1
3.3.99	FAST Interrupt Error Handler (FSTREC) ...	3.3.99-1

3.3.96 FAST Output Chain Buffer Request Subroutine (FSTRBUF)

3.3.96.1 Purpose. This subroutine allocates high and low priority output chain buffer space for the Final Approach Spacing Tool (FAST). This subroutine is called by KOF, TROUT, TPRED, MAT, CDR, AUT, KIP, and COMA.

3.3.96.2 Inputs/Outputs

Inputs

Register Preset:

S0	- Set to 0
S1	- Set to FASTD
B3	- Contains the callers buffer priority (index)
	- Zero (0) equals low priority buffer request
	- One (1) equals high priority buffer request
A	- Requested buffer space

FAST Global Data:

FSTBEGBF	- FAST Output Buffer Beginning Address
FSTCRBUF	- FAST Output Chain Buffer Current Address
FSTRBUFR	- FAST Output Buffer Request Subroutine Lock
FSTSBADD	- FAST Next Available Sub-Buffer Address
FSTUSER	- FAST Output Chain Buffer User Counter
IOJUMP	- FAST I/O Jump Instruction Mask
IONOPMSK	- FAST I/O NOOP Instruction Mask

FAST Local Data:

FSTBUFB1	- FAST Buffer Request Subroutine B1 Temporary Storage
FSTBUFRT	- FAST Output Chain Buffer Reset Flag
FSTCSIZ	- FAST Output Chain Buffer Size Request
FSTENDBF	- FAST Output Chain Buffer Ending Address
FSTSBSBA	- FAST Current Output Chain User Buffer Address

Outputs

A	- Contains the user requested output chain buffer address
FSTCRBUF	- FAST Output Chain Buffer Current Address
FSTRBUFR	- FAST Output Buffer Request Subroutine Lock
FSTSBADD	- FAST Next Available Sub-Buffer Address
FSTUSER	- FAST Output Chain Buffer User Counter

Table Locks:

FSTRBUFR	- FAST Output Buffer Request Subroutine Lock

System Parameters:

RTCQ	- IOP Internal RTC Memory Location

Site Parameters:

FASTPQ	- FAST Relative Peripheral Number

3.3.96.3 Design Logic. The FAST Output Chain Buffer Request Subroutine (FSTRBUF) is called from various FAST extraction routines for the purpose of allocating buffer space within the FAST high (FSTOTBF1) or low (FSTOTBF2) priority output chain buffers. This reentrant subroutine can process both high and low priority buffer requests simultaneously, but requires table locks for multiple requests for the same priority buffers.

The FAST Output Chain Buffer User Counter (FSTUSER) is incremented by the FSTRBUF subroutine as each user requests buffer space. FSTUSER is decremented by the FAST Transmit Buffer Subroutine (FSTRSND), after each user has finalized their buffer.

The FAST Next Available Sub-Buffer Address (FSTSBADD) is saved in the FAST Current Output Chain User Buffer Address (FSTSBSBA).

If the requested space will not cause the selected buffer to overflow its physical size the following will occur:

1. the next available sub-buffer address is computed by adding the requested space to FSTSBADD;
2. an IONOOP instruction is built containing the 18-bit address of the selected buffer and is stored at the requester's last buffer address;
3. the A-register is loaded with the starting sub-buffer address contained in FSTSBSBA;
4. if FSTRSND has requested a new sub-buffer starting address, FSTCRBUF is set to FSTSBSBA;
5. the FSTRBUF subroutine is unlocked for the requested buffer and the subroutine is exited with the starting sub-buffer address in the A-register.

If the requested space results in an overflow of the selected buffer, the following will occur:

1. the FAST Output Chain Buffer Reset Flag (FSTBUFRT) is incremented. (Note: this flag is used solely for Buffer sizing).

2. if FSTRSND has not requested a new sub-buffer starting address, an IOJUMP instruction is built containing the 18-bit beginning address of the selected buffer and is stored at the last users ending buffer address;

3. FSTSBADD and FSTSBSBA are reset to the beginning address of the physical buffer;

4. the next available sub-buffer address is computed by adding the requested space to FSTSBADD;

5. an IONOOP instruction is built containing the 18-bit address of the selected buffer and is stored at the requester's last buffer address;

6. the A-register is loaded with the starting sub-buffer address contained in FSTSBSBA;

7. if FSTRSND has requested a new sub-buffer starting address, FSTCRBUF is set to FSTSBSBA;

8. the FSTRBUF subroutine is unlocked for the requested buffer and the subroutine is exited with the starting sub-buffer address in the A-register.

3.3.96.4 Flowcharts. See figure 3.3.96-1.

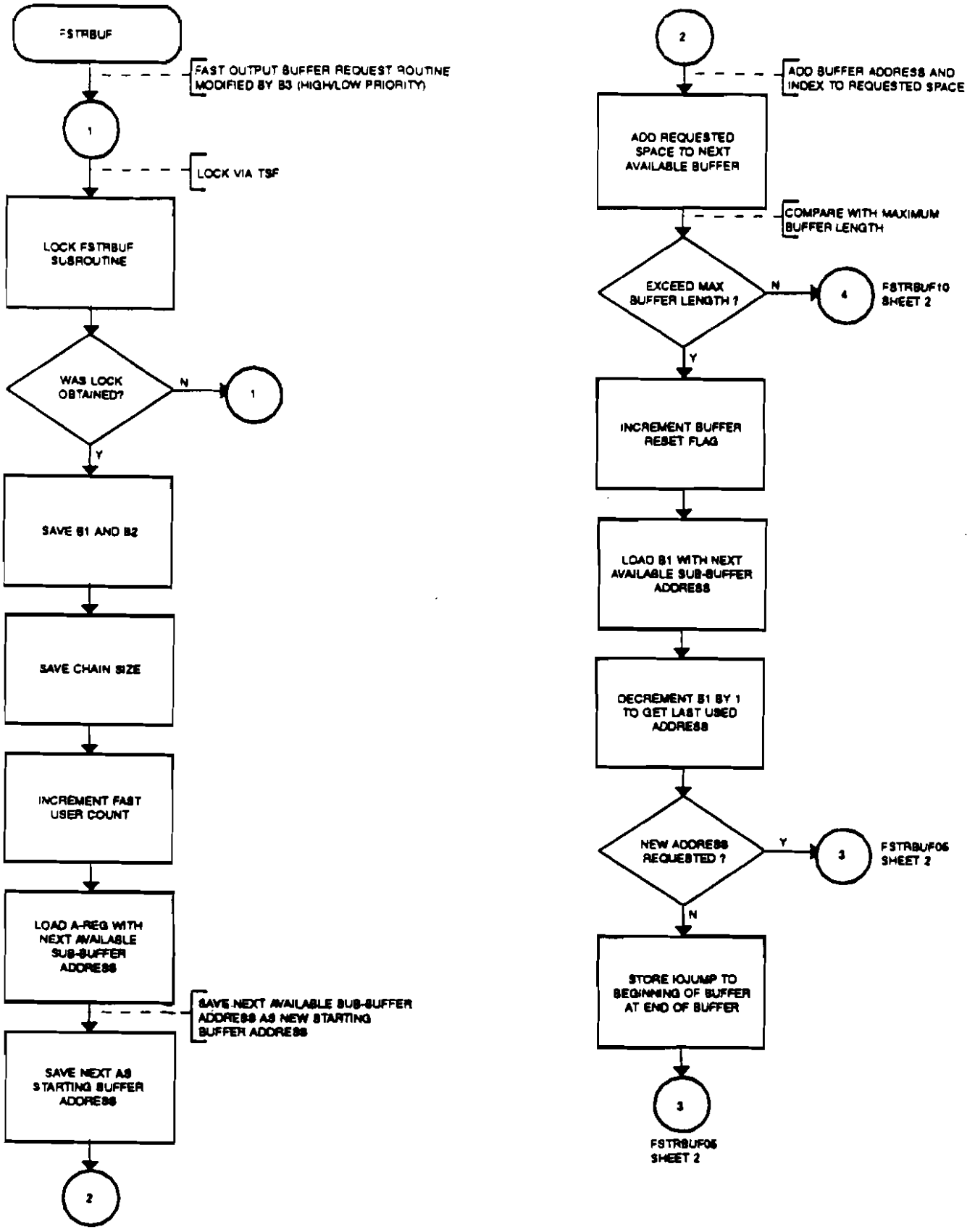


FIGURE 3.3.96-1 FAST OUTPUT BUFFER REQUEST FLOWCHART (SHEET 1)

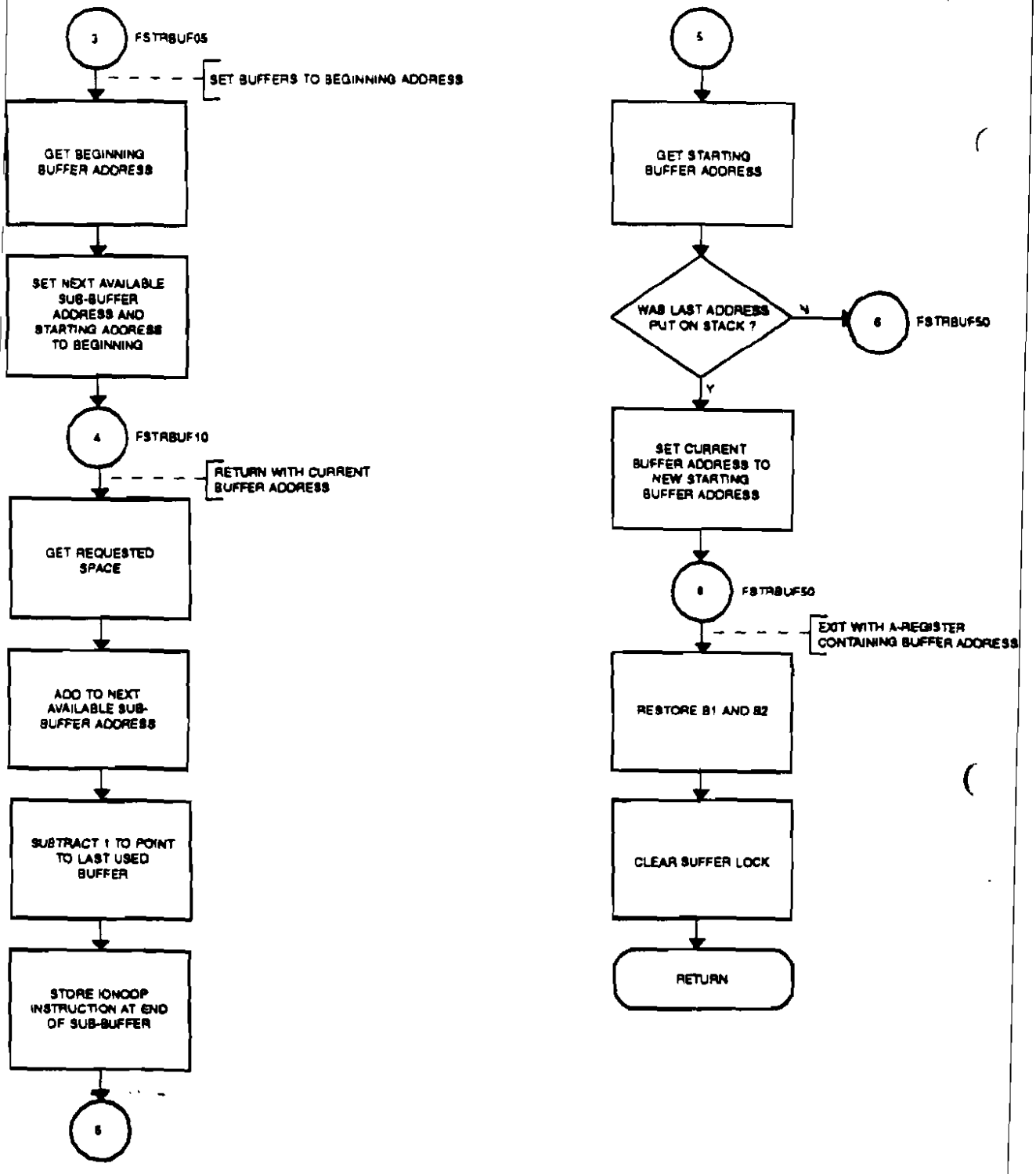


FIGURE 3.3.96-1 FAST OUTPUT BUFFER REQUEST FLOWCHART (SHEET 2)

3.3.97 FAST Transmit Buffer Subroutine (FSTRSND)

3.3.97.1 Purpose. This subroutine finalizes output chain buffers, updates output stack pointer tables, and initiates output chains for the Final Approach Spacing Tool (FAST). This subroutine is called by KOF, TROUT, TPRED, MAT, CDR, AUT, KIP, and COMA.

3.3.97.2 Inputs/Outputs

Inputs

Register Preset:

S0	- Set to 0
S1	- Set to FASTD
B3	- Contains the caller's buffer priority
	- Zero (0) equals low priority buffer request
	- One (1) equals high priority buffer request

FAST Global Data:

ADFSTPPS	- FAST Stack Pointer Address Table
FASTFAIL	- FAST Failure Flag
FSTBEGBF	- FAST Output Buffer Beginning Address
FSTBIT27	- RTCQ Clock Wrap Mask
FSTCRBUF	- FAST Output Chain Buffer Current Address
FSTIME	- FAST Output Buffer Initiated Time
FSTMCERP	- FAST Interrupt Error Flag
FSTOUTA	- FAST Output Channel Active Flag
FSTPPS1	- FAST High Priority Output Stack
FSTPPS2	- FAST Low Priority Output Stack
FSTRBUFR	- FAST Output Buffer Request Subroutine Lock
FSTSBADD	- FAST Next Available Sub-Buffer Address
FSTTSF01	- FAST Output Stack Lock
FSTUSER	- FAST Output Chain Buffer User Counter
IOJUMP	- FAST I/O Jump Instruction Mask

IONOPMSK - FAST I/O NOOP Instruction Mask

FAST Local Data:

FSTECBF - FAST Last Used Address Current Buffer

FSTCHNER - TIU Output Channel Failure Time-out

FSTIMF - FAST Output Buffer Active Fail Time

FSTRBSND - FAST Buffer Send Subroutine Lock

FSTSNDB1 - Temporary Storage B1

FSTSNDB2 - Temporary Storage B2

FSTSNDB4 - Temporary Storage B4

FSTSTKER - TIU Output Stack Overflow

FSTSTOPM - I/O Stop with Monitor Mask

Outputs

S0 - Set to 0

S1 - Set to FASTD

B3 - Contains the caller's buffer priority

- Zero (0) equals low priority buffer request

- One (1) equals high priority buffer request

A - Contains Error Condition:
Positive = No Error
Negative = Stack Overflow or Unable to Initiate Output

FASTFAIL - FAST Failure Flag

FSTECBF - FAST Last Used Address Current Buffer

FSTCHNER - TIU Output Channel Failure Time-out

FSTCRBUF - FAST Output Chain Buffer Current Address

FSTIME - FAST Output Buffer Initiated Time

FSTMCERP - FAST Interrupt Error Flag

FSTOUTA - FAST Output Channel Active Flag

FSTPPS1 - FAST High Priority Output Stack

FSTPPS2 - FAST Low Priority Output Stack

FSTSBADD - FAST Next Available Sub-Buffer Address

FSTTSF01 - FAST Output Stack Lock

FSTUSER - FAST Output Chain Buffer User Counter

Table Locks:

FSTOUTA - FAST Output Channel Active Flag

FSTRBSND - FAST Buffer Send Subroutine Lock

FSTRBUFR - FAST Output Buffer Request Subroutine Lock

FSTTSF01 - FAST Output Stack Lock

System Parameters:

RTCQ - IOP Internal RTC Memory Location

Site Parameters:

FASTPQ - FAST Relative Peripheral Number

3.3.97.3 Design Logic. The FAST Transmit Subroutine (FSTRSND) is a non-reentrant subroutine called by various FAST extraction routines to initiate FAST buffered output chains. Upon entrance, the subroutine is locked and pertinent registers are saved. The FAST Output Chain Buffer Request Subroutine (FSTRBUF) is locked, for the specified buffer, to allow for the decrementing/interrogating of the FAST Output Chain Buffer User Counter (FSTUSER). FSTUSER 'lower' is cleared to initialize the PUSH fail flag and the 'upper' is decremented/interrogated for other buffer users.

The following processing flow is dependent on the buffer user count:

A. If there are additional users the following will occur:

1. the FSTRBUF Subroutine is unlocked for the specified buffer;

2. the caller will bypass all logic to finalize the current sub-buffer by proceeding to FSTRSD10 (see section 3.3.97.3.C) to attempt output for buffers already on the FAST Output Stack (FSTPPS1/FSTPPS2).

B. If there are no other users the following will occur:

1. the FAST Output Stack Lock (FSTTSF01/FSTTSF02) is set for the specified priority buffer;

2. B4 is set to the address of either the FAST High Priority Output Stack (FSTPPS1) or the FAST Low Priority Output Stack (FSTPPS2);

3. if an output chain buffer address is already on the specified stack the following will occur:

a). if the FAST Last Used Address Current Buffer (FSTECBF) is greater than the FAST Next Available Sub-buffer Address (FSTSBADD), then the physical buffer has wrapped and an IOJUMP instruction is built to replace the IOSTOP instruction located at the end of the previous buffer;

b). if the buffer did not wrap, the IOSTOP is replaced with an IONOOP instruction;

c). B4 is set to the contents of FSTSBADD minus one, and then stored in FSTECBF as the new ending buffer address;

d). an IOSTOP with monitor instruction is stored at the new ending buffer address;

e). Bit 29 of FSTCRBUF is set, requesting the FSTRBUF Subroutine to update FSTCRBUF to a new starting sub-buffer address;

f). the FSTTSF01 and FSTRBUFR locks are cleared;

g). processing proceeds to FSTRSD10 (see section 3.3.97.3.C) to attempt output for buffers already on the FAST Output Stack (FSTPPS1/FSTPPS2).

4. the following process occurs when the stack is empty:

a). in preparation to place the current buffer address onto the specified stack, B1 is set to the FAST Output Chain Buffer Current Address (FSTCRBUF);

b). Bit 29 of FSTCRBUF is set, requesting the FSTRBUF Subroutine to update FSTCRBUF to a a new starting sub-buffer address;

c). the last user's buffer address is computed by subtracting 1 from the FAST Next Available Sub-Buffer Address (FSTSBADD) and the result is stored in the FAST Last Used Current Buffer (FSTECBF);

d). an IOSTOP instruction with monitor is stored at the last user's buffer address;

e). the B1 register is PUSHed onto the selected stack, and if the PUSH fails, FSTUSER (Upper) is set negative;

f). the FSTTSF01 and FSTRBUFR locks are cleared;

g). if the PUSH failed (FSTUSER Upper is negative) the error exit FSTRSD32 (see section 3.3.97.3.F) is taken;

h). if the FAST Output Channel Active Flag (FSTOUTA, Lower) is set, continuation of processing will not be allowed until the flag is set to zero. This is to insure that a monitor clock does not occur, because the processor in the FAST Input Interrupt Handler, could be the same processor as the one in FSTRSND;

i). processing proceeds to FSTRSD10 (see section 3.3.97.3.C) to attempt output for buffers already on the FAST Output Stack (FSTPPS1/FSTPPS2).

C. The FSTRSD10 routine performs the following output initiation functions:

1. if the FAST Output Channel Active Flag (FSTOUTA) indicates that the channel is active, then the FAST output initiation logic is bypassed and processing proceeds to FSTRSD30, the FAST channel time-out error logic (see section 3.3.97.3.E);

2. if the channel is not active, FSTOUTA and FSTTSF01+1 are locked via Test and Set instructions;

3. an attempt is made to PULL a buffer address from the high priority stack;

4. if the high priority stack is empty, the following processing occurs:

    a). the high priority stack is unlocked and the FAST Low Priority Stack Lock (FSTTSF01) is set via TSF;

    b). FSTTSF01 is cleared after an attempt is made to PULL a buffer address from the low priority stack;

    c). if a low priority buffer address was obtained, processing proceeds to the FSTRSD25 task to initiate the output chain at the address contained in the B1-register (see section 3.3.97.3.D);

    d). if the stack is empty, FSTOUTA is cleared and processing proceeds to the exit return FSTRSD35 (see section 3.3.97.3.G).

5. if a high priority buffer address was obtained, the high priority stack lock is cleared, and processing proceeds to the FSTRSD25 task to initiate the output chain at the address contained in the B1-register (see section 3.3.97.D);

D. The FSTRSD25 routine performs the following output functions:

1. obtain the Real Time Clock (RTCQ) and set the time of the initiation of the output chain in the FAST Output Buffer Initiated Time (FSTIME);

2. initiate the output chain via an ESICOUT Executive Service Request from the location identified by the B-register;

3. if an ESICOUT ESR packet error occurs, processing proceeds to the error exit FSTRSD32 (see section 3.3.97.3.F);

4. normal processing proceeds to the exit return FSTRSD35 (see section 3.3.97.3.G).

E. The FSTRSD30 routine performs the following channel time-out logic:

1. the FAST Output Buffer Initiated Time (FSTIME) is subtracted from the value of RTCQ and adjusted for bit 27 time wrap around;

2. if the time difference does not exceed the FAST Output Buffer Time-out parameter (FSTIMFQ), processing proceeds to the normal exit return FSTRSD35 (see 3.3.97.3.G);

3. the TIU Output Channel Failure Time-out (FSTCHNER) counter is incremented;

4. processing proceeds to the error exit FSTRSD32 (see 3.3.97.3.F).

F. The FSTRSD32 routine performs the following error handling logic:

1. if FSTUSER (Upper) is negative, indicating that a PUSH onto an Output Chain Buffer Stack has failed, the TIU Output Stack Overflow (FSTSTKER) counter is incremented;

2. the FAST Interrupt Error (FSTMCERP) and FAST Failure (FASTFAIL) Flags are set;

3. the FAST Output Channel Active Flag (Upper) is cleared.

G. The FSTRSD35 routine performs the following normal exit processing:

1. all saved registers are restored to their previous values;

2. the A-Register is returned to the caller with the error condition code contained in FSTUSER upper (0 = No Error, Negative = PUSH Failure);

3. the FSTRSND Subroutine is unlocked and control is returned to the caller.

3.3.97.4 Flowcharts.

See figure 3.3.97-1.

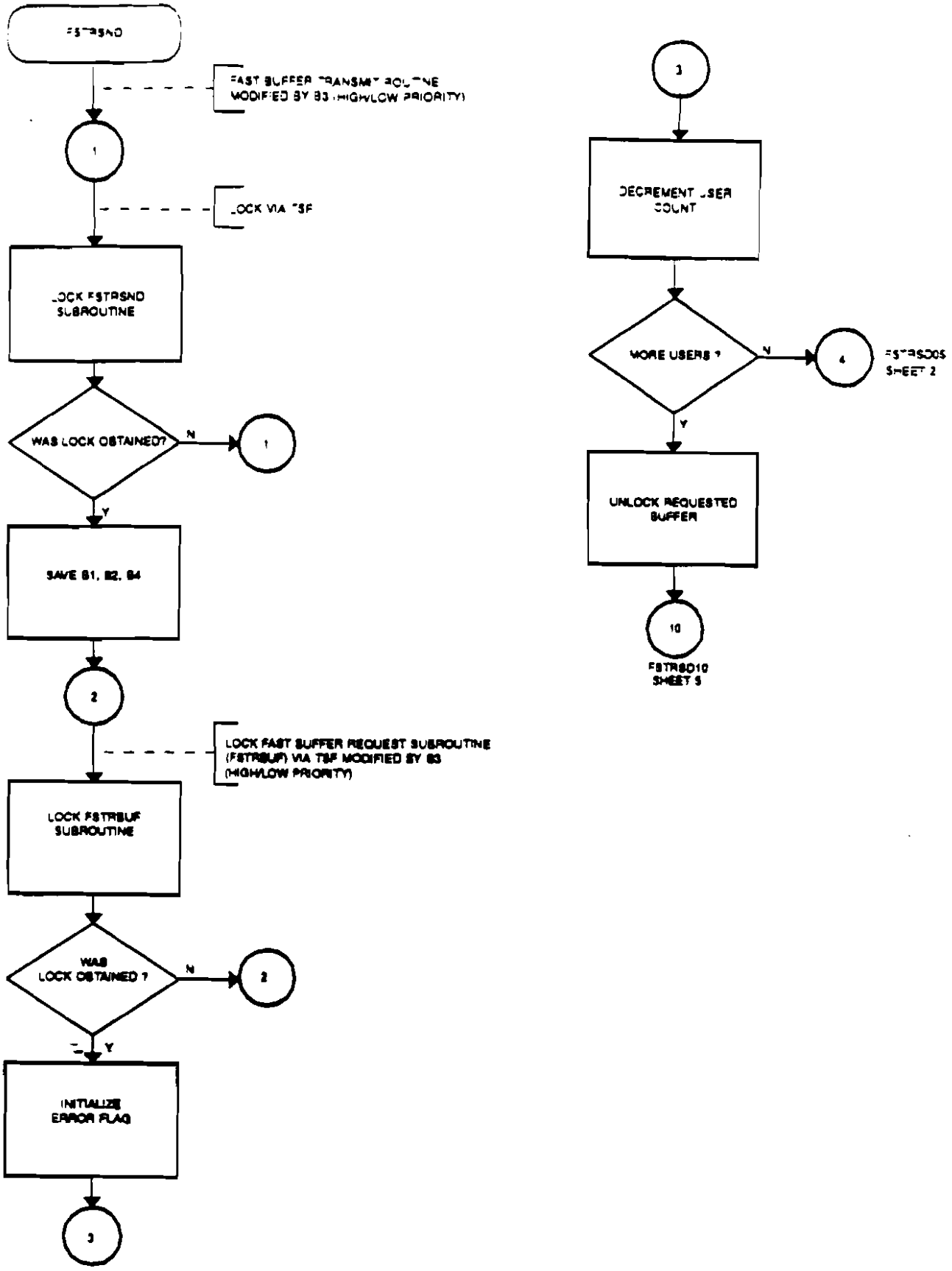


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 1)

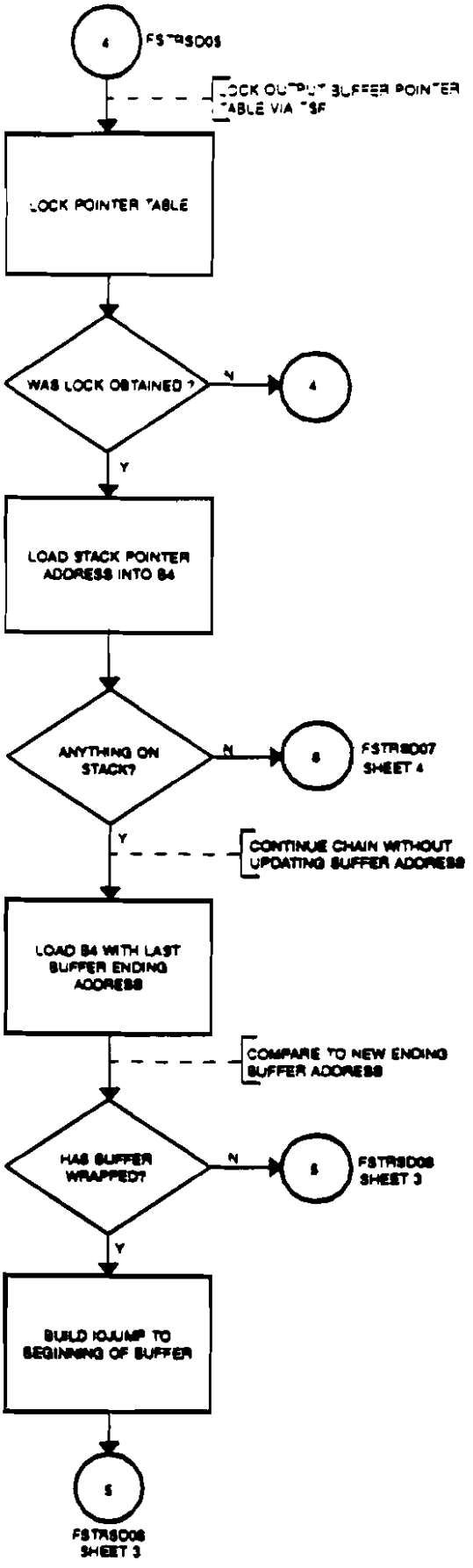


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 2)

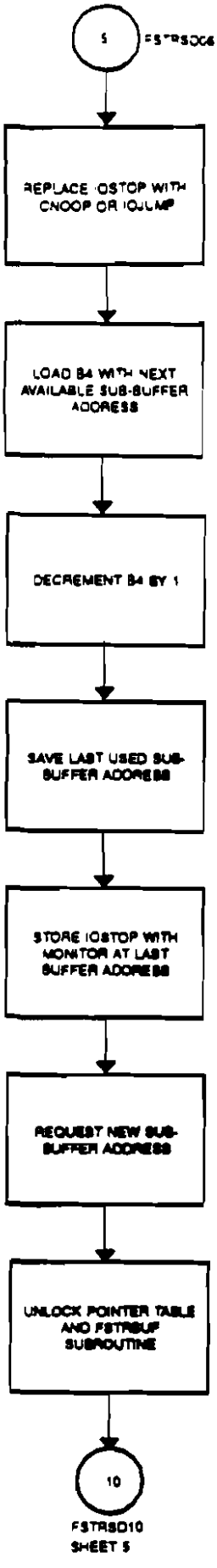


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 3)

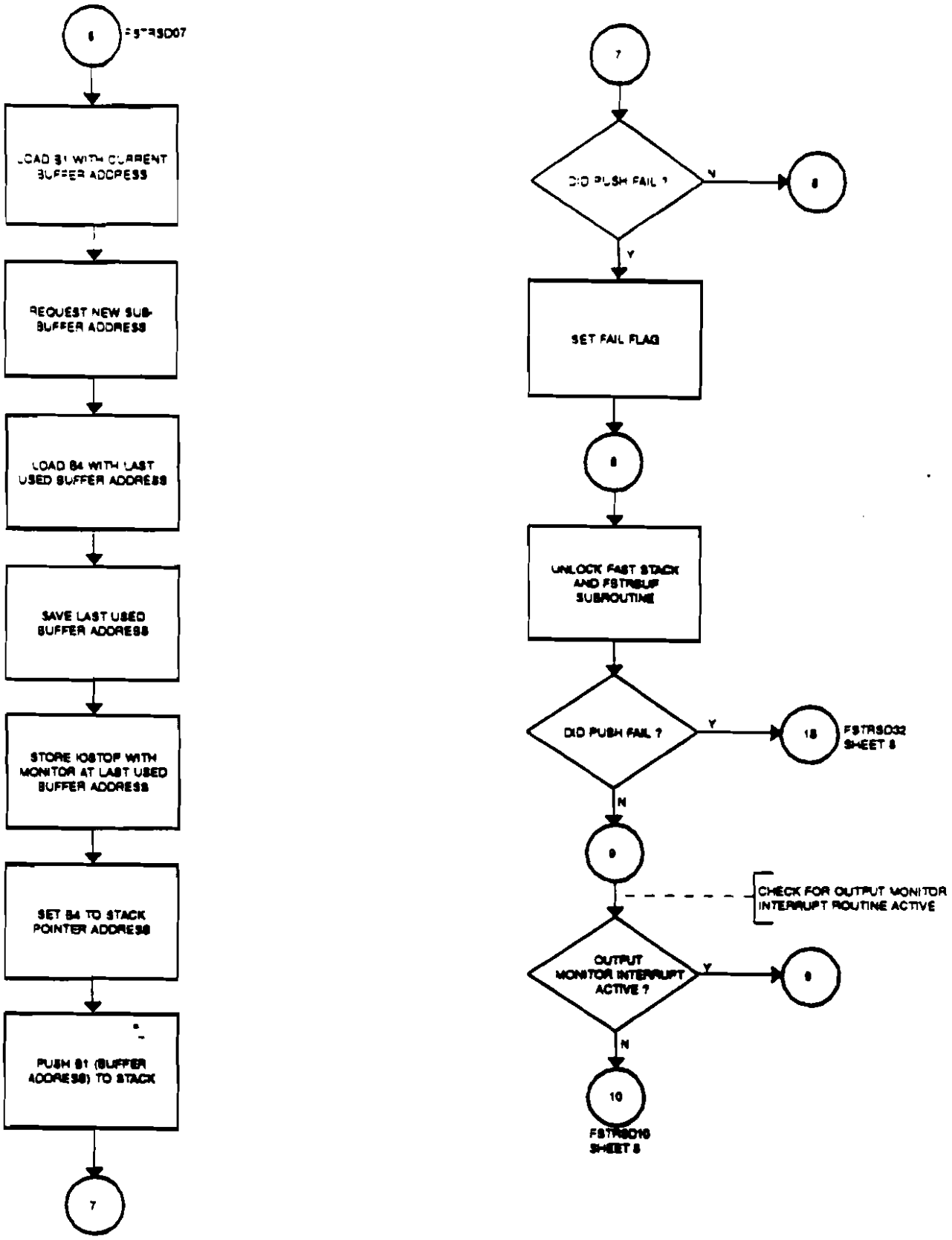


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 4)

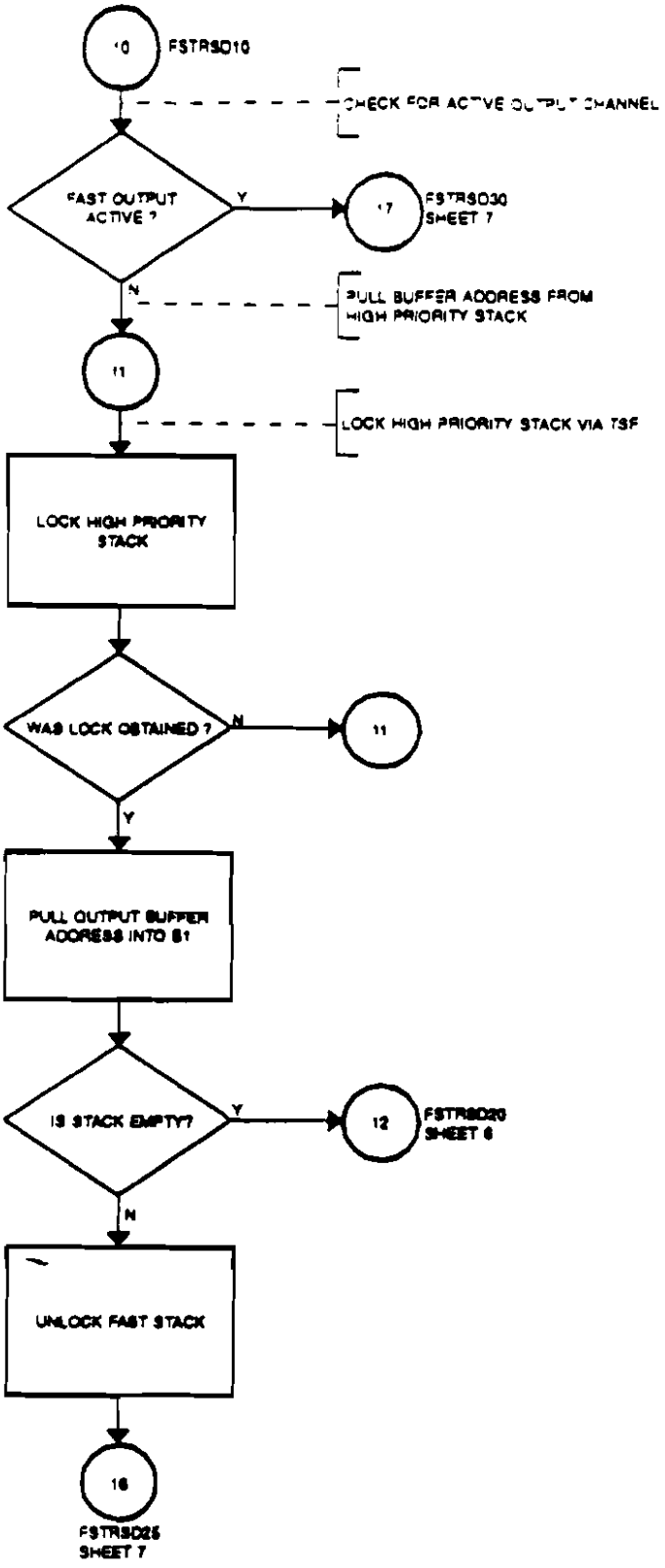


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 5)

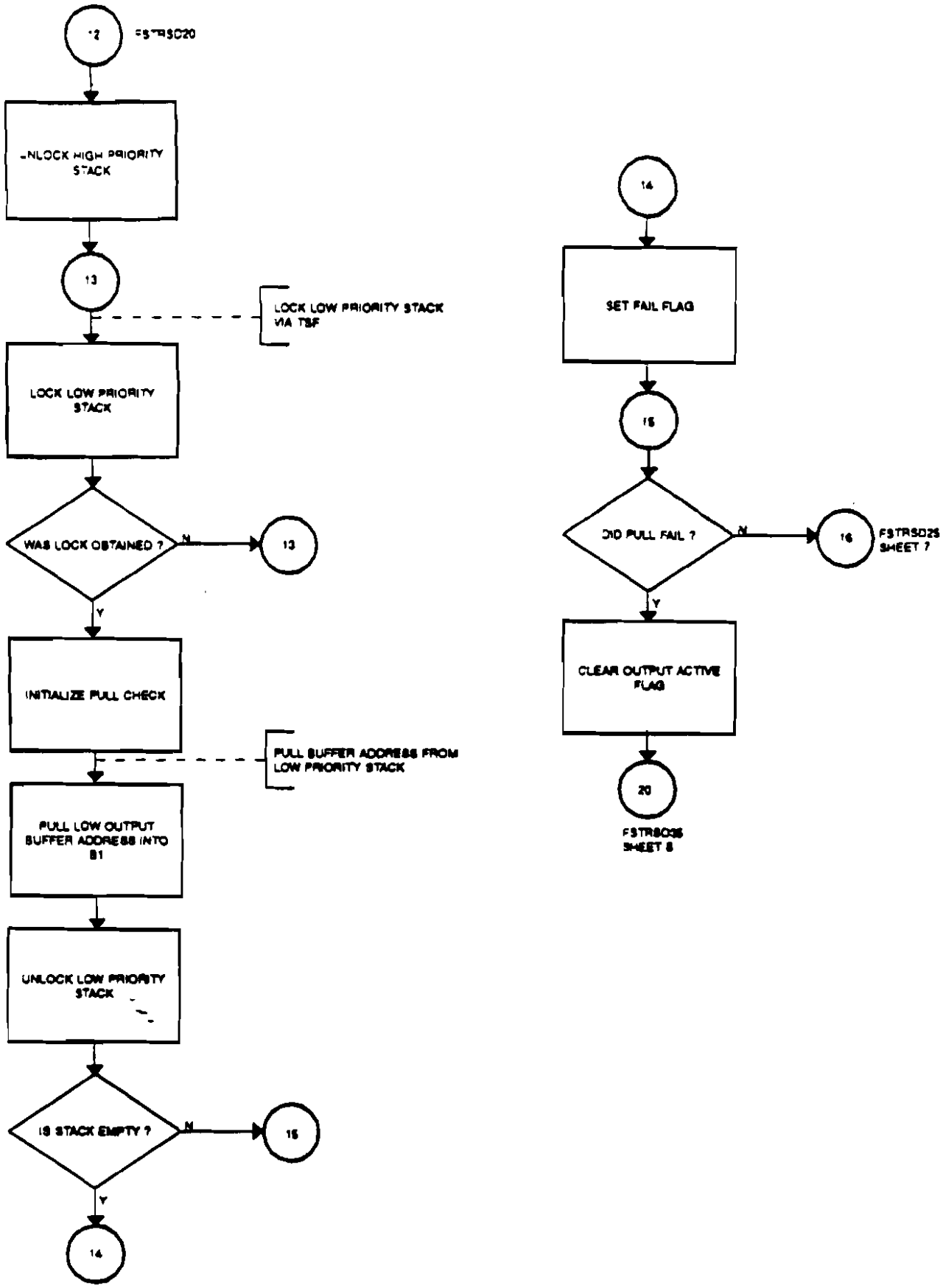


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 6)

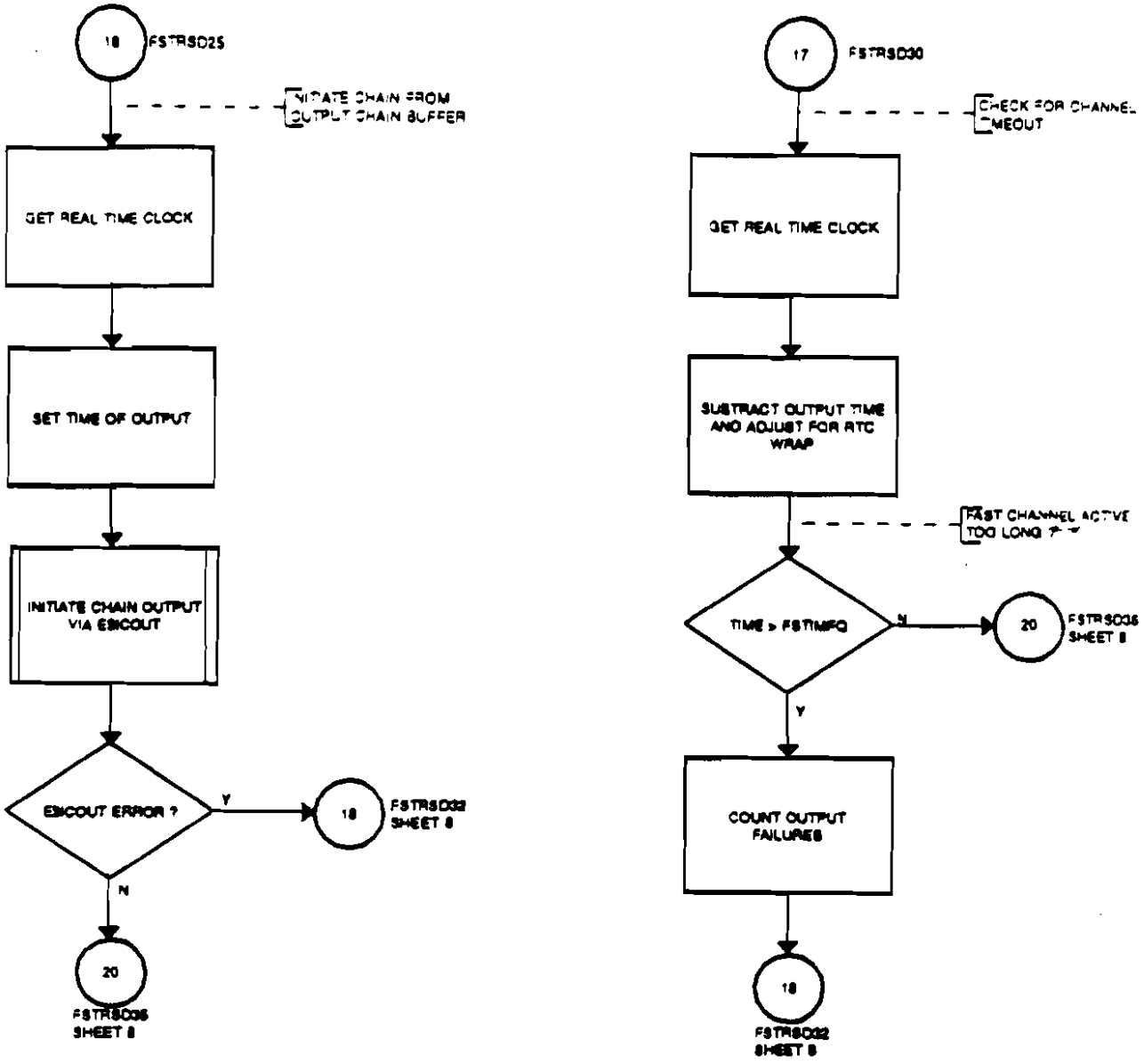


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 7)

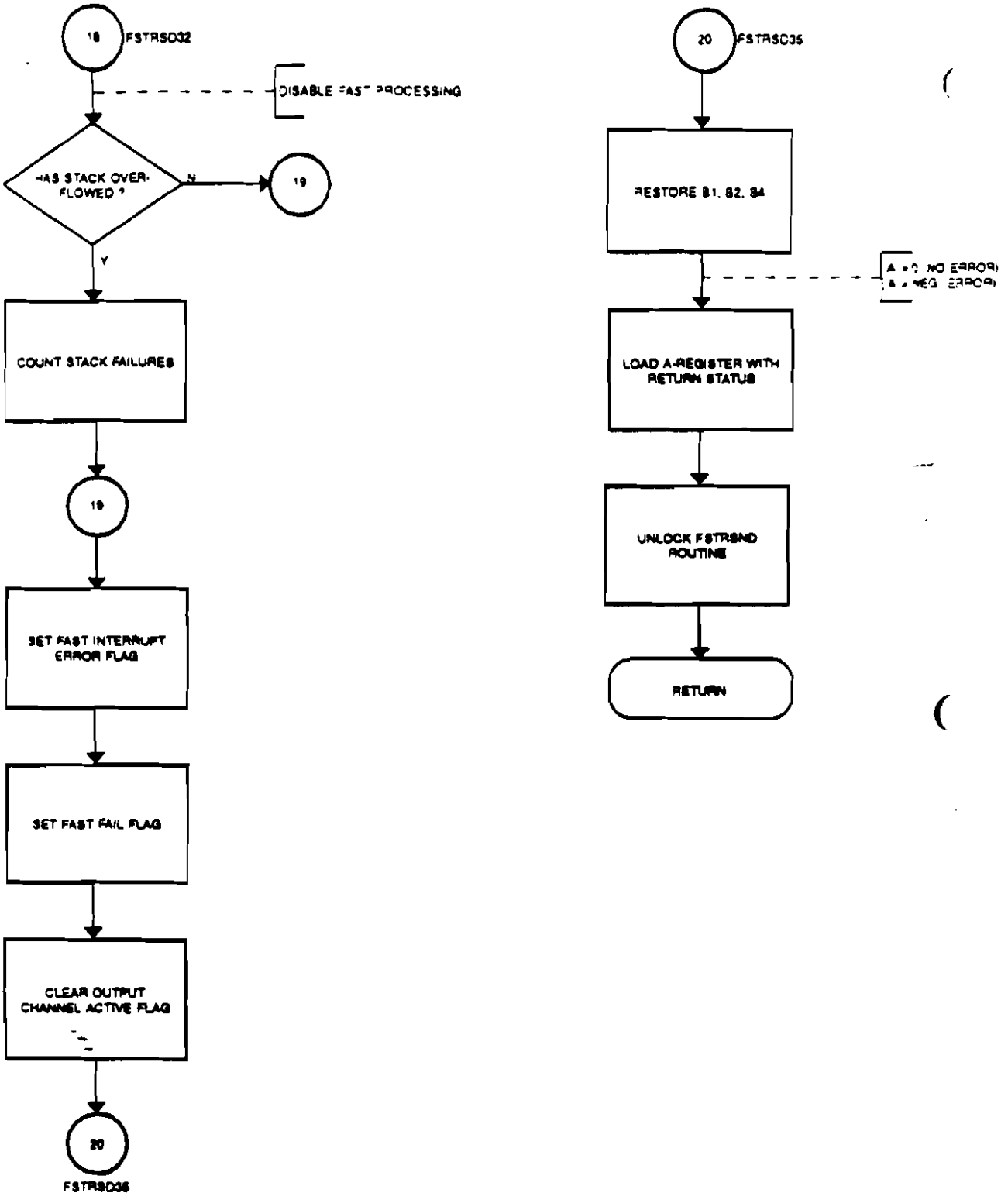


FIGURE 3.3.97-1 FAST OUTPUT BUFFER TRANSMIT FLOWCHART (SHEET 8)

3.3.96 FAST Output Chain Buffer Request Subroutine (FSTRBUF)

3.3.96.1 Purpose. This subroutine allocates high and low priority output chain buffer space for the Final Approach Spacing Tool (FAST). This subroutine is called by KOF, TROUT, TPRED, MAT, CDR, AUT, KIP, and COMA.

3.3.96.2 Inputs/Outputs

Inputs

Register Preset:

S0	- Set to 0
S1	- Set to FASTD
B3	- Contains the caller's buffer priority
	- Zero (0) equals low priority buffer request
	- One (1) equals high priority buffer request
A	- Requested buffer space

FAST Global Data:

FSTBEGBF	- FAST Output Buffer Beginning Address
FSTCRBUF	- FAST Output Chain Buffer Current Address
FSTRBUFR	- FAST Output Buffer Request Subroutine Lock
FSTSBADD	- FAST Next Available Sub-Buffer Address
FSTUSER	- FAST Output Chain Buffer User Counter
IOJUMP	- FAST I/O Jump Instruction Mask
IONOPMSK	- FAST I/O NOOP Instruction Mask

FAST Local Data:

FSTBUFB1	- FAST Buffer Request Subroutine B1 Temporary Storage
FSTBUFRT	- FAST Output Chain Buffer Reset Flag
FSTCSIZ	- FAST Output Chain Buffer Size Request
FSTENDBF	- FAST Output Chain Buffer Ending Address
FSTSBSBA	- FAST Current Output Chain User Buffer Address

Outputs

A	- Contains the user requested output chain buffer address
FSTCRBUF	- FAST Output Chain Buffer Current Address
FSTRBUFR	- FAST Output Buffer Request Subroutine Lock
FSTSBADD	- FAST Next Available Sub-Buffer Address
FSTUSER	- FAST Output Chain Buffer User Counter

Table Locks:

FSTRBUFR	- FAST Output Buffer Request Subroutine Lock

System Parameters:

RTCQ	- IOP Internal RTC Memory Location

Site Parameters:

FASTPQ	- FAST Relative Peripheral Number

3.3.96.3 Design Logic. The FAST Output Chain Buffer Request Subroutine (FSTRBUF) is called from various FAST extraction routines for the purpose of allocating buffer space within the FAST high (FSTOTBF1) or low (FSTOTBF2) priority output chain buffers. This reentrant subroutine can process both high and low priority buffer requests simultaneously, but requires table locks for multiple requests for the same priority buffers.

The FAST Output Chain Buffer User Counter (FSTUSER) is incremented by the FSTRBUF subroutine as each user requests buffer space. FSTUSER is decremented by the FAST Transmit Buffer Subroutine (FSTRSND), after each user has finalized their buffer.

The FAST Next Available Sub-Buffer Address (FSTSBADD) is saved in the FAST Current Output Chain User Buffer Address (FSTSBSBA).

If the requested space will not cause the selected buffer to overflow its physical size the following will occur:

1. the next available sub-buffer address is computed by adding the requested space to FSTSBADD;
2. an IONOOP instruction is stored at the requester's last buffer address;
3. the A-register is loaded with the starting sub-buffer address contained in FSTSBSBA;
4. if FSTRSND has requested a new sub-buffer starting address, FSTCRBUF is set to FSTSBSBA;
5. the FSTRBUF subroutine is unlocked for the requested buffer and the subroutine is exited with the starting sub-buffer address in the A-register.

If the requested space results in an overflow of the selected buffer, the following will occur:

1. the FAST Output Chain Buffer Reset Flag (FSTBUFRT) is incremented. (Note: this flag is used solely for Buffer sizing).

2. if FSTRSND has not requested a new sub-buffer starting address, an IOJUMP instruction is built containing the 18-bit beginning address of the selected buffer and is stored at the last users ending buffer address;

3. FSTSBADD and FSTSBSBA are reset to the beginning address of the physical buffer;

4. the next available sub-buffer address is computed by adding the requested space to FSTSBADD;

5. an IONOOP instruction is stored at the requester's last buffer address;

6. the A-register is loaded with the starting sub-buffer address contained in FSTSBSBA;

7. if FSTRSND has requested a new sub-buffer starting address, FSTCRBUF is set to FSTSBSBA;

8. the FSTRBUF subroutine is unlocked for the requested buffer and the subroutine is exited with the starting sub-buffer address in the A-register.

3.3.96.4 Flowcharts. See figure 3.3.96-1.

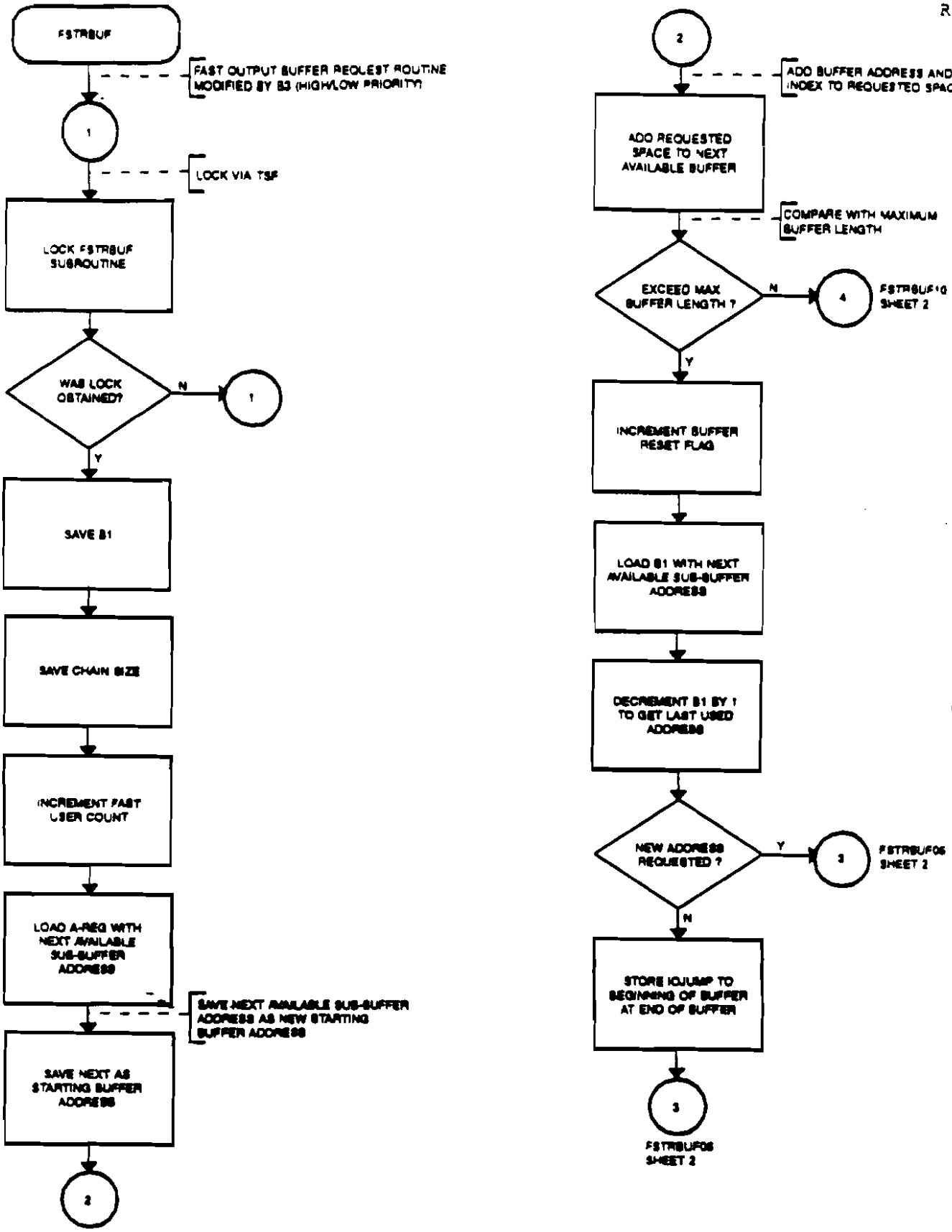


 FIGURE 3.3.96-1 FAST OUTPUT BUFFER REQUEST FLOWCHART (SHEET 1)

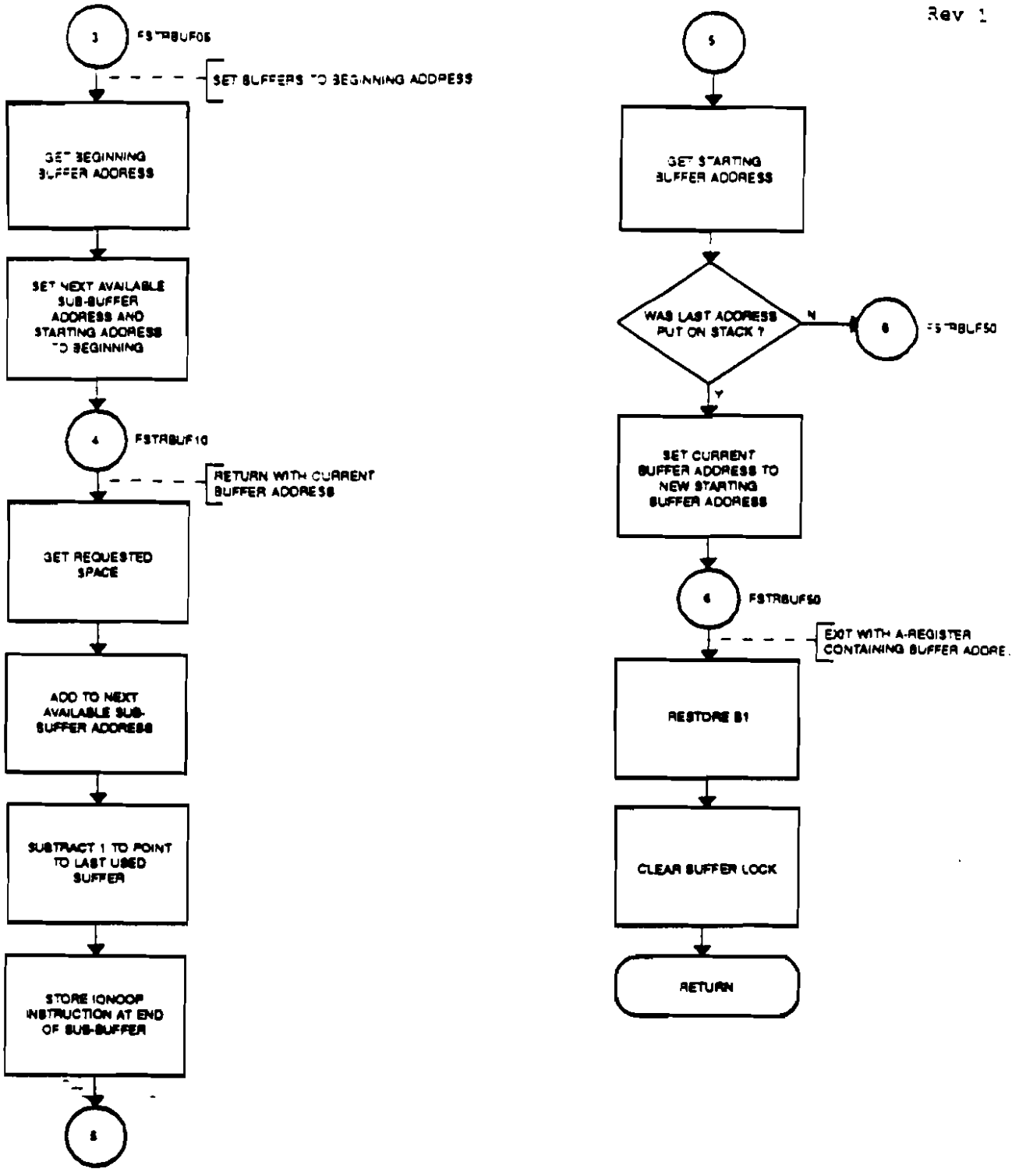


FIGURE 3.3.96-1 FAST OUTPUT BUFFER REQUEST FLOWCHART (SHEET 2)

3.3.98 Final Approach Spacing Tool (FAST) Input Interrupt Handler (FSTREC)

3.3.98.1 Purpose. The Final Approach Spacing Tool (FAST) interrupt handler consists of two routines, the FAST Input Interrupt Handler (FSTREC) and the FAST Interrupt Error Handler (FSTERRP). FSTREC performs the initial processing for each interrupt received on the FAST channel, and FSTERRP performs channel time-out failure and error message output processing for the FAST function.

3.3.98.2 Inputs/Outputs

Inputs

Register Preset:

None

FAST Global Data:

FASTATUS - FAST Interrupt Handler Status Word Table

FASTD - FAST Data Base

FASTTMP - FAST Input Handler Absolute Base Address

FSTEXTPR - FAST External Interrupt Parity Error Counter

FSTILERR - FAST Illegal Interrupt Counter

FSTIME - FAST Output Buffer Initiated Time

FSTINPER - FAST Input Buffer Parity Error Counter

FSTMCERP - FAST Interrupt Error Flag

FSTMSGCD - TIU Received Bad Message Code Counter

FSTOPAR - TIU Received Bad Data With Parity Counter

FSTOTBF1 - FAST High Priority Output Chain Buffer

FSTOTBF2 - FAST Low Priority Output Chain Buffer

FSTOUTA - FAST Output Channel Active Flag

FSTPPS1 - FAST High Priority Output Stack

FSTPPS2 - FAST Low Priority Output Stack

FSTTSF01 - FAST Output Stack Lock

FAST Local Data:

FASTMPB1 - FAST Temporary Save B1

FASTMPB2 - FAST Temporary Save B2

FASTMPB3 - FAST Temporary Save B3

FASTMPB4 - FAST Temporary Save B4

FASTMPSA - FAST Temporary Save A-Register

FASTMPSQ - FAST Temporary Save Q-Register

FASTMPS0 - FAST Temporary Save S0

FASTMPS1 - FAST Temporary Save S1

FSTAMSK - FAST External Interrupt Mask

Outputs

FASTATUS - FAST Interrupt Handler Status Word Table

FSTEXTPR - FAST External Interrupt Parity Error Counter

FSTILERR - FAST Illegal Interrupt Counter

FSTIME - FAST Output Buffer Initiated Time

FSTINPER - FAST Input Buffer Parity Error Counter

FSTMCERP - FAST Interrupt Error Flag

FSTMSGCD - TIU Received Bad Message Code Counter

FSTOPAR - TIU Received Bad Data with Parity Counter

FSTOTBF1 - FAST High Priority Output Chain Buffer

FSTOTBF2 - FAST Low Priority Output Chain Buffer

FSTOUTA - FAST Output Channel Active Flag

FSTPPS1 - FAST High Priority Output Stack

FSTPPS2 - FAST Low Priority Output Stack

FSTTSF01 - FAST Output Stack Lock

Table Locks:

FSTOUTA - FAST Output Channel Active Flag

FSTTSF01 - FAST Output Stack Lock

System Parameters:

FASTLGTH - FAST Interrupt Status Word Table Length

RTCQ - IOP Internal RTC Memory Location

Site Parameters:

FASTPQ - FAST Relative Peripheral Number

FSTEXTQ - FAST External Interrupt Parity Error

FSTIBPQ - FAST Input Buffer Parity Error

FSTILERQ - FAST Illegal Interrupt Error

FSTPARQ - FAST Output Parity Error

The following is a definition of the FAST Interrupt Status Word:

Bits 29-28 Undefined

Bits 27-25 Interrupt Type

000 External Interrupt Line Parity Error -

A parity error has been detected on a status word sent by the TATCA Interface Unit (TIU).

001 Input Buffer Parity Error -

A parity error is detected on an input word from the TIU, parity is corrected and the data is stored in the input buffer.

010 Not Used - Illegal Interrupt

011 Output Chain Monitor -

Output chain has completed and the status word is stored in the status word table.

100 External Interrupts -

TIU Initiated Interrupt

101 - 111 Not Used - Illegal Interrupt

Bits 24-15 Undefined

Bits 14-02 Not Used

Bits 01-0 TIU Initiated Interrupts -

01 TIU detected parity error in data received from the IOP.

10 TIU detected a bad message code from the IOP

3.3.98.3 Design Logic. The FAST Input Interrupt Handler (FSTREC) is the first stage of a two stage interrupt process. FSTREC performs the initial interrupt processing for each interrupt received on the FAST channel. The FAST Interrupt Error Handler (FSTERRP) performs all second stage interrupt processing, including error message output and channel time-out processing.

Upon entrance, all pertinent registers are saved. The B1-Register is initialized and used as a status word counter to determine if the status word table has overflowed. The B2-Register is loaded with the FAST Input Handler Absolute Base Address (FASTTMP) for the FAST channel.

The following processing flow is dependent on the type of interrupt:

A. The FSTREC10 routine attempts to PULL a status word from the FAST Interrupt handler Status Word Table (FASTATUS), if the table is empty, processing proceeds to the exit routine FSTEXT20 (see section 3.3.98.3.J).

B. If a status word has been successfully PULLed from the stack the following processing occurs:

1. The status word is saved in the FAST Interrupt Handler Status Word Table (FASTATUS).

2. If the number of status words PULLed is equal to the (SP) FAST Interrupt Status Word Table Length (FASTLGTH) then a status word table overflow has occurred and processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I).

3. The status word's interrupt type field is aligned and loaded into the B3-Register. The B3-Register is used as an index into the following interrupt routine jump table via an Execute Remote instruction:

Status Word Bits 27-25	Interrupt Type	Processing Routine
000	External Interrupt Line Parity Error	FSTINT01
001	Input Buffer Parity Error	FSTINT02
010	Not Used	FSTILEGL
011	Output Chain Monitor	FSTOMN10
100	External Interrupts	FSTEXT01
101	Not Used	FSTILEGL
110	Not Used	FSTILEGL
111	Not Used	FSTILEGL

C. The External Interrupt Line Parity Error routine is entered at label FSTINT01 whenever an external interrupt line parity error is detected. Upon entering, the FAST External Interrupt Parity Error Counter (FSTEXTPR) is incremented. If the count exceeds the (SV) External Interrupt Parity Error (FSTEXTQ) Parameter, processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I). Otherwise, processing proceeds to routine FSTREC20 (see section 3.3.98.3.H) to process the next status word.

D. The Input Buffer Parity Error routine is entered at label FSTINT02 whenever an input buffer parity error is detected. Upon entering, the FAST Input Buffer Parity Error Counter (FSTINPER) is incremented. If the count exceeds the (SV) FAST Input Buffer Parity Error (FSTIBPQ) Parameter, processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I). Otherwise, processing proceeds to routine FSTREC20 (see section 3.3.98.3.H) to process the next status word.

E. The Output Chain Monitor Interrupt routine is entered at label FSTOMN10 whenever an Output Chain Monitor Interrupt is detected. This routine performs the following functions:

1. A lock is placed between TASK Processing and Interrupt Processing by setting the lower of the FAST Output Channel Active Flag (FSTOUTA) to a negative value. This allows FAST stacks to be locked via Test and Set Flag (TSF) instructions, by both the interrupt routine as well as a TASK routine without causing a monitor clock time-out.

2. An attempt is made to lock the FAST High Priority Output Stack Lock (FSTTSF01+1). If the lock is not obtained, processing proceeds to label FSTOMN30 to process any low priority buffers. If the lock is obtained, an attempt is made to PULL a buffer address into the B1-Register from the FAST High Priority Output Stack (FSTPPS1). If successful, the FSTTSF01+1 lock is cleared and processing proceeds to label FSTOMN40 to initiate an output chain from the FAST High Priority Output Chain Buffer (FSTOTBF1). If the stack is empty, processing proceeds to label FSTOMN20 where the FSTTSF01+1 lock is cleared and an attempt is made to lock the FAST Low Priority Output Stack Lock (FSTTSF01).

3. If the low priority lock is not obtained processing proceeds to label FSTOMN60 to clear FSTOUTA and the next status word is processed. If the lock is obtained, an attempt is made to PULL a buffer address into the B1-Register from the FAST Low Priority Output Stack (FSTPPS2). If the stack is empty, processing proceeds to label FSTOMN50 to clear both the low priority stack lock and the active channel flag prior to processing the next status word. If a low priority buffer address is obtained, the low priority stack is unlocked and processing proceeds to label FSTOMN40 to initiate an output chain from the FAST Low Priority Output Chain Buffer (FSTOTBF2).

4. FSTOMN40 is performed when an output chain buffer address is PULLed from either the high or low output priority stacks. Upon entrance, FSTOUTA (lower) is cleared to allow FAST TASK Processing to continue. The Real Time Clock (RTCQ) is obtained and stored into the FAST Output Buffer Initiated Time (FSTIME). An Initiate Chain Out (ESICOUT) Executive Service Request (ESR) is performed, activating the chain at the buffer location specified in the B1-Register. If an ESICOUT ESR packet error occurs, processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I). Normal processing proceeds to FSTREC10 (see section 3.3.98.3.A) to process the next status word.

F. The External Interrupt routine is entered at label FSTEXT01 whenever an External Interrupt is detected. This routine performs the following functions:

1. The status word is obtained from FASTATUS and all bits are cleared except the TIU Initiated Interrupts (Bits 0-1). If the TIU interrupt code is not equal to 1, processing proceeds to label FSTINT03B. If the code is 1, indicating the TIU has detected a parity error in data received from the Input/Output Processor (IOP), the TIU Received Bad Data With Parity Counter (FSTOPAR) is incremented. If the count exceeds the (SV) FAST Output Parity Error (FSTPARQ) Parameter, processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I). Otherwise, processing proceeds to FSTREC10 (see section 3.3.98.3.A) to process the next status word.

2. FSTINT03B is performed when the TIU Interrupt Code is greater than 1. If the code is equal to 2, indicating that the TIU detected a bad message code from the IOP, the TIU Received Bad Message Code Counter (FSTMSGCD) is incremented and processing proceeds to the FAST Failure routine FSTEXIT (see section 3.3.98.3.I). Otherwise, processing proceeds to FSTILEGL (see section 3.3.98.3.G) to process the illegal interrupt.

G. The Illegal Interrupt routine is entered at label FSTILEGL whenever an illegal interrupt is detected. Upon entering, the FAST Illegal Interrupt Counter (FSTILERR) is incremented. If the count exceeds the (SV) FAST Illegal Interrupt Error (FSTILERQ) Parameter, the FAST Interrupt Error Flag (FSTMCERP, Lower) is set to request TTY output. Processing then proceeds to FSTREC10 (see section 3.3.98.3.A) to process the next status word.

H. FSTREC20 is performed when TTY output is required. This routine sets the FSTMCERP (Lower) to request the FAST Interrupt Error Handler (FSTERRP) to process an error request. Processing proceeds to FSTREC10 (see section 3.3.98.3.A) to process the next status word.

I. FSTEXIT is performed when a fatal error condition occurs and FAST must be disabled. Upon entrance, FSTMCERP is set negative, indicating a FAST failure has occurred.

J. FSTEXT20 is the normal interrupt handler exit. All pertinent registers are restored and control is returned to the interrupted routine while releasing the interrupt lockout.

3.3.98.4 Flowcharts. See figure 3.3.98-1.

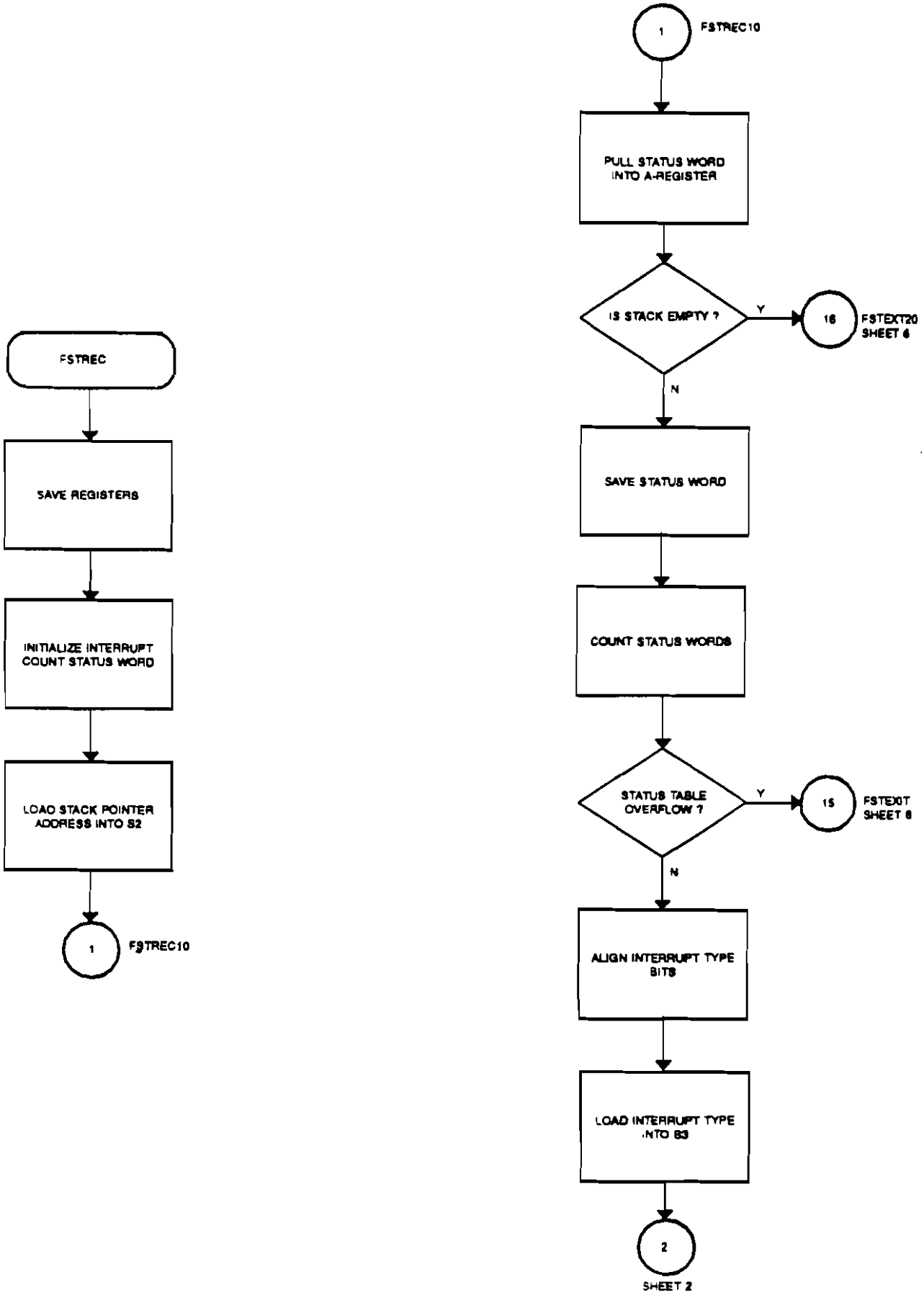


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 1)

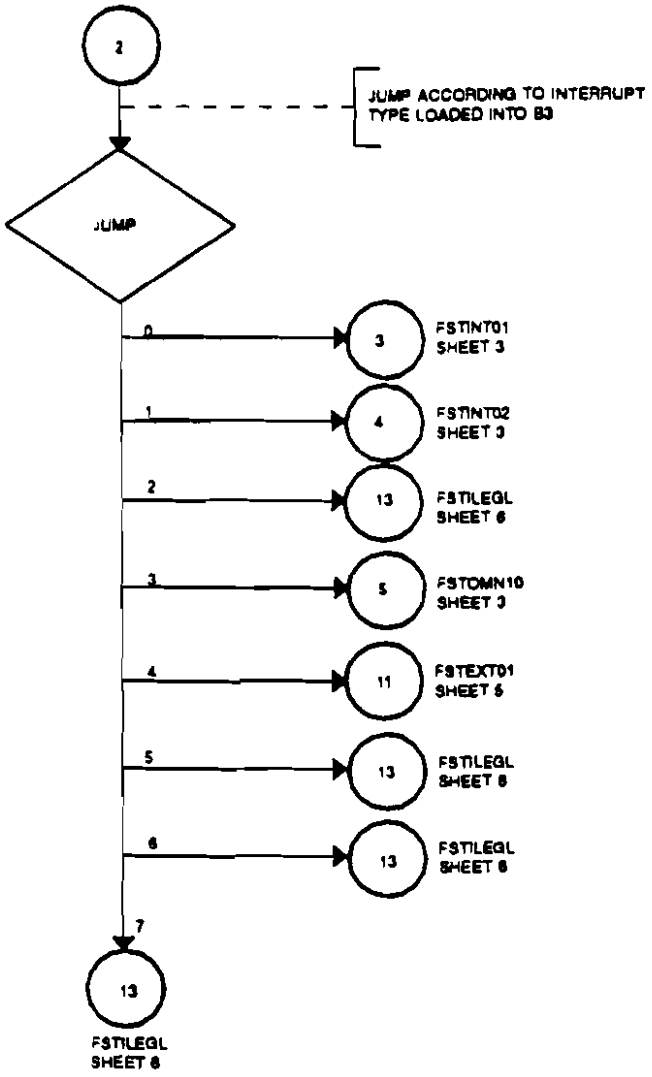


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 2)

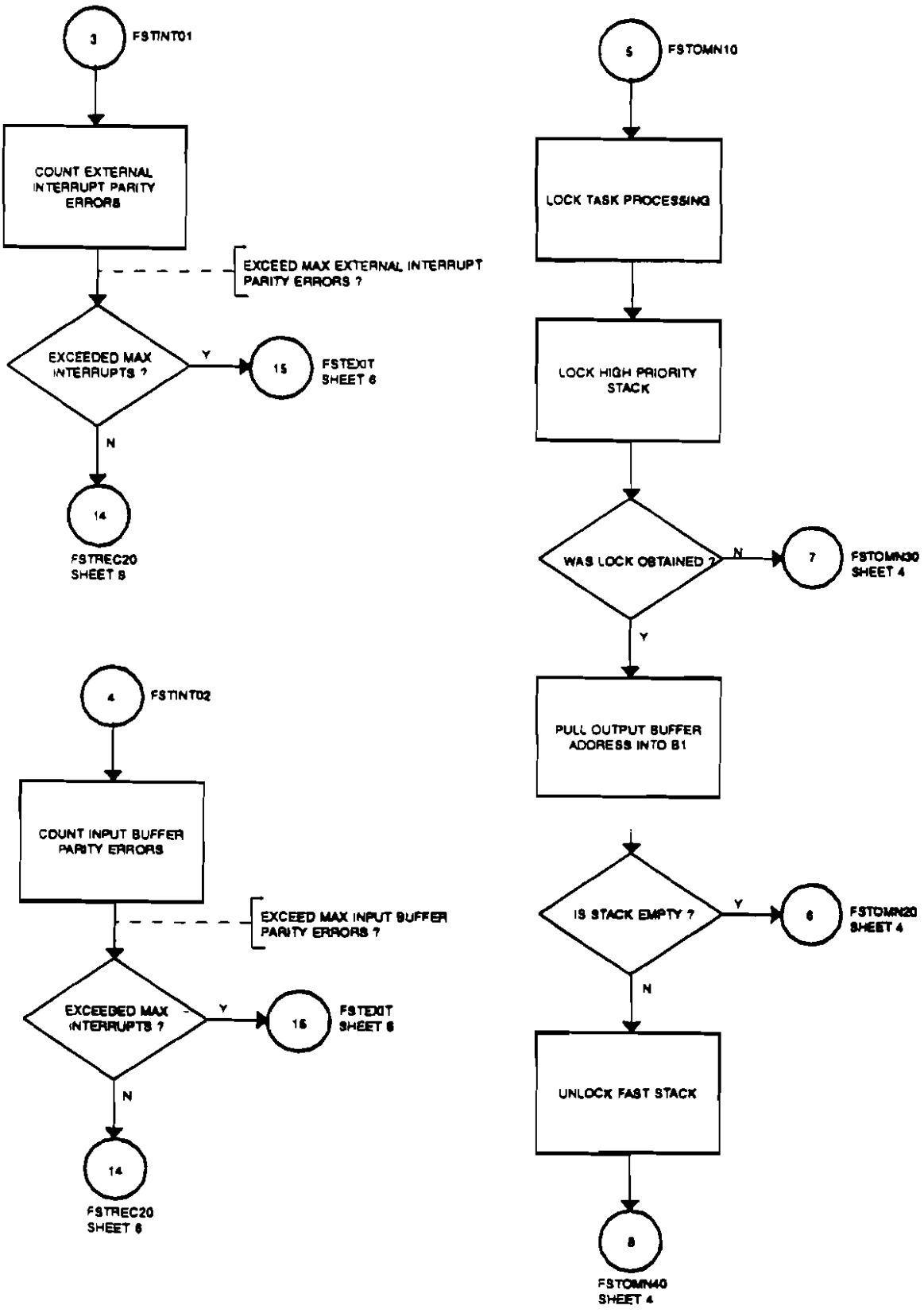


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 3)

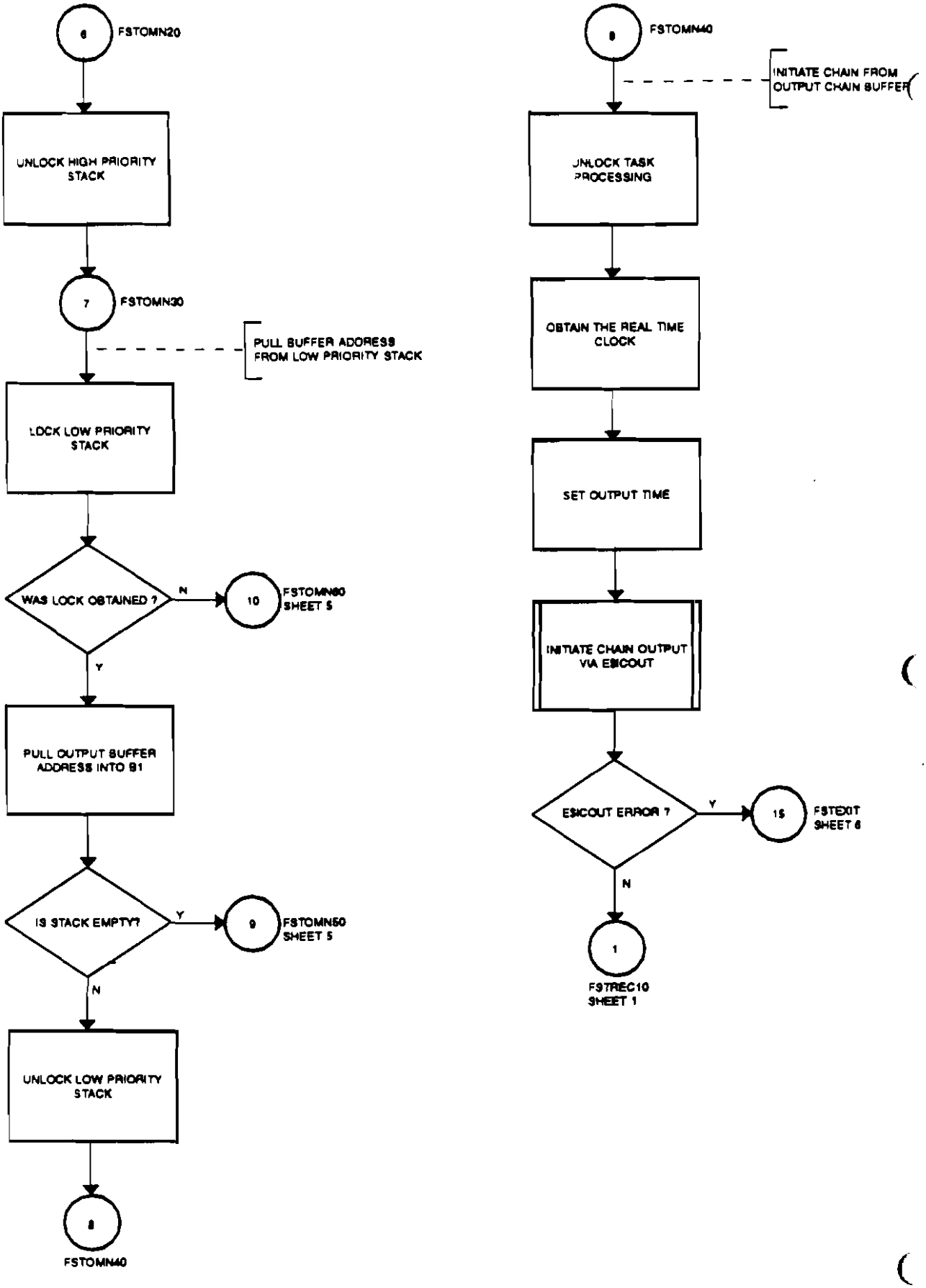


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 4)

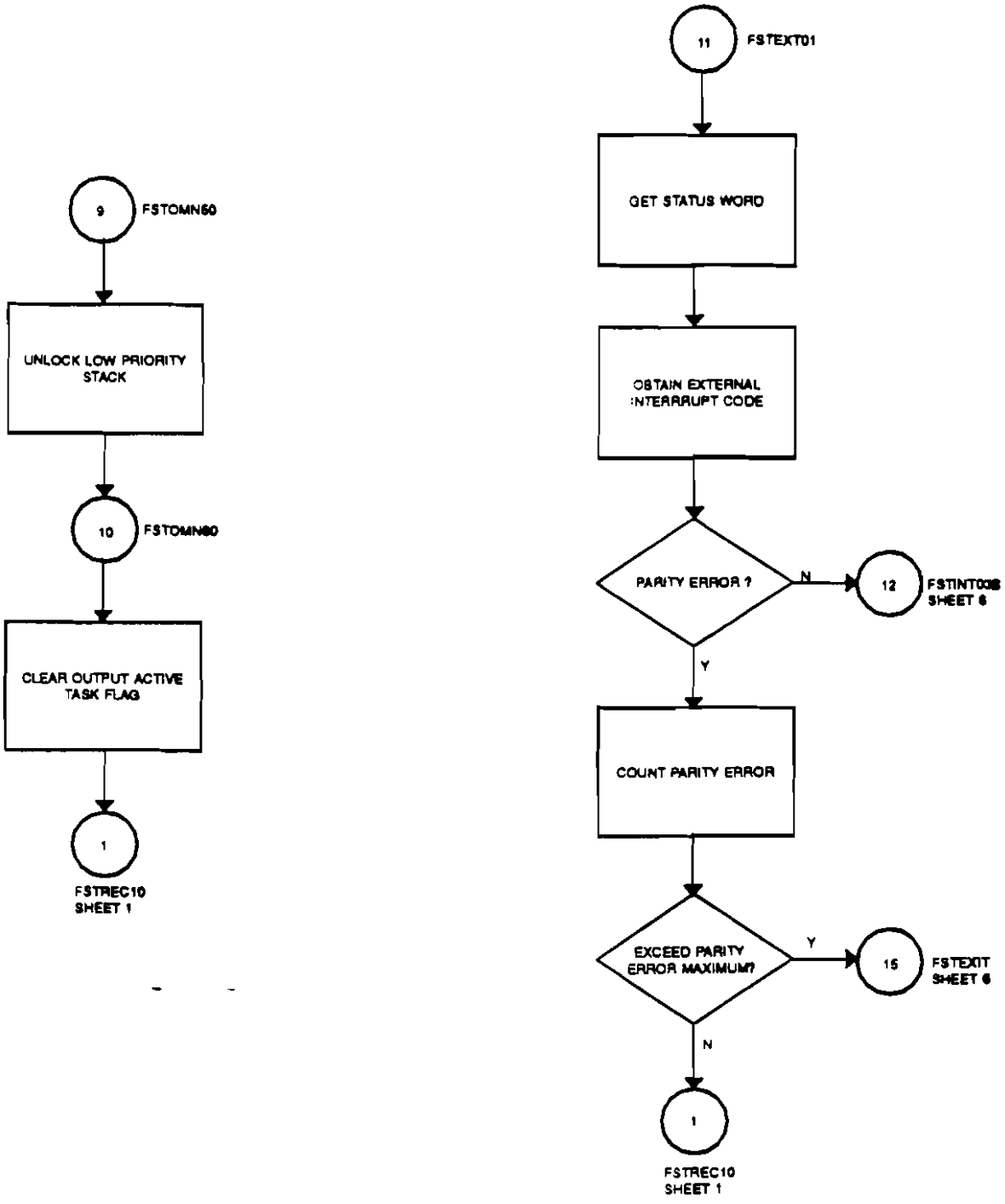


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 5)

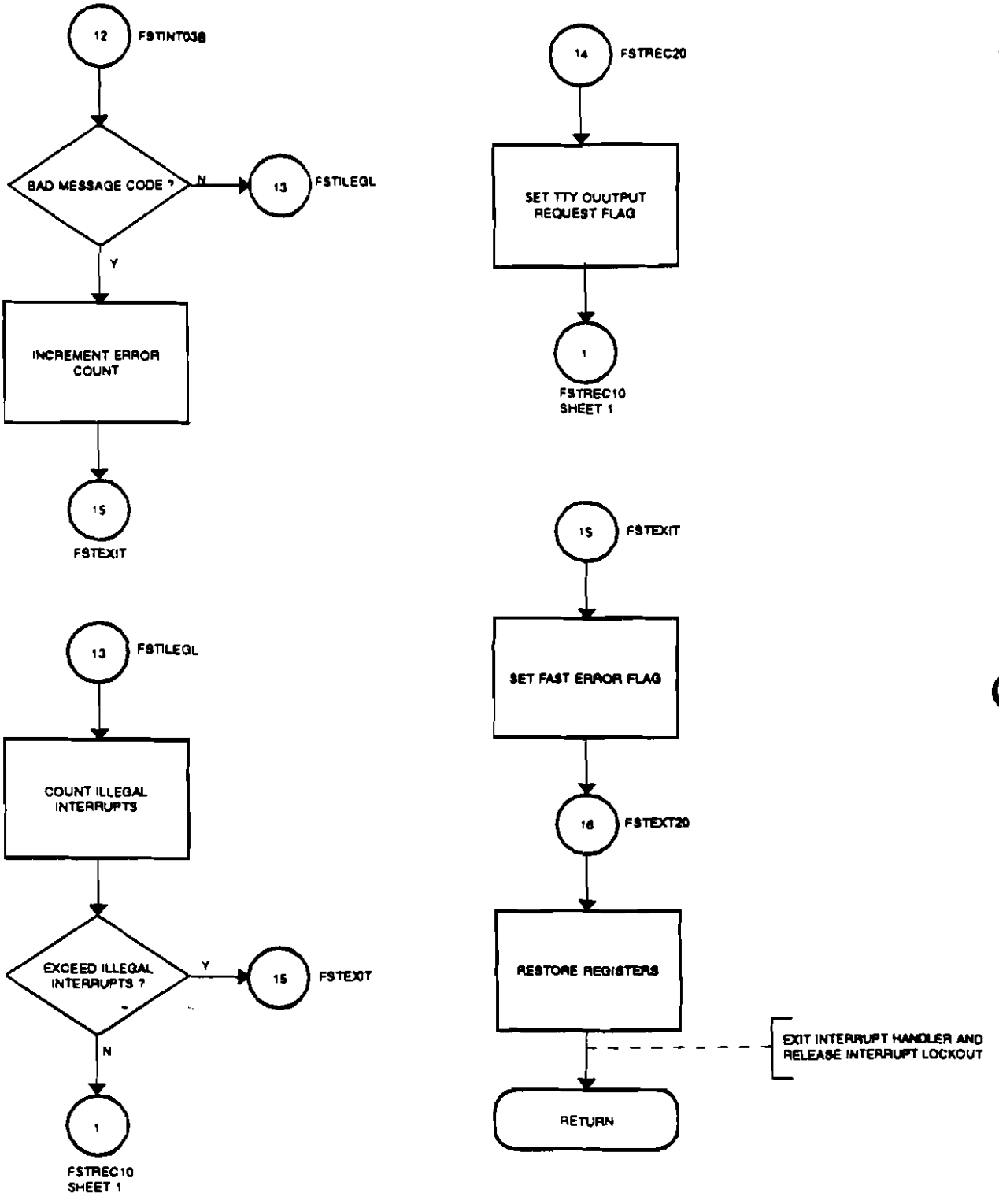


FIGURE 3.3.98-1 FAST INPUT INTERRUPT ROUTINE (SHEET 6)

3.3.99 Final Approach Spacing Tool (FAST) Interrupt Error Handler (FSTERRP)

3.3.99.1 Purpose. The Final Approach Spacing Tool (FAST) interrupt handler consists of two routines, the FAST Input Interrupt Handler (FSTREC) and the FAST Interrupt Error Handler (FSTERRP). FSTREC performs the initial processing for each interrupt received on the FAST channel, and FSTERRP performs channel time-out failure and error message output processing for the FAST function.

3.3.99.2 Inputs/Outputs

Inputs

Register Preset:

S0 - Set to 0

FAST Global Data:

FAILTO - FAST Display Fail Time-out

FASTD - FAST Data Base

FASTFLG - FAST On/Off Flag

FASTFAIL - FAST Processing Loss

FINITFLG - FAST Initialization Flag

FINITTME - TIU Initialization Start Time

FSTBIT27 - FAST Bit 27 Mask

FSTEXTPR - FAST External Interrupt Parity Error Counter

FSTINBCW1 - FAST Old Input Buffer Pointer

FSTINBUFA - FAST Input Buffer Address

FSTMCERP - FAST Interrupt Error Flag

FSTOUTA - FAST Output Channel Active Flag

HBFAST - TIU Heartbeat Receipt Time

HBSEQLK - TIU Heartbeat Sequence Lock

FAST Local Data:

CONVFDXR - FAST Execute Remote Instruction Table

FSTBCWMSK - FAST Last Input Buffer Address

FSTERTM - FAST Error Message Output Parameter Table

FSTFAILT - FAST Fail Display Time

FSTFLPKT - TIU System Failure Packet

FSTILMES - FAST Illegal Input Message Code Counter

FSTINBCW - FAST Current Input Buffer Pointer

FSTINBUF - FAST Input Buffer

FSTINLST - FAST Last Input Buffer Address

FSTMANY - FAST Field Data Mask Many

FSTMASK - FAST Field Data Mask Digit

FSTMERR - FAST Error Message Output Time

FSTTYPKT - FAST Interrupt Error TTY Packet Address

FTYP - FAST Fail Print Flag

HBFAILCNT - TIU Heartbeat Failure Counter

HBSEQCNT - TIU Heartbeat Sequence Counter

HBSEQFAIL - TIU Heartbeat Sequence Number Failure

Outputs

FAILTO - FAST Display Fail Time-out

FASTFLG - FAST On/Off Flag

FASTFAIL - FAST Processing Loss

FINITFLG - FAST Initialization Flag

FINITTME - TIU Initialization Start Time

FSTEXTPR - FAST External Interrupt Parity Error Counter

FSTFAILT - FAST Fail Display Time

FSTFLPKT - TIU System Failure Packet

FSTILMES - FAST Illegal Input Message Code Counter

FSTINBCW - FAST Current Input Buffer Pointer

FSTINBCW1 - FAST Old Input Buffer Pointer

FSTMCERP - FAST Interrupt Error Flag

FSTMERR - FAST Error Message Output Time

FSTOUTA - FAST Output Channel Active Flag

FSTTYPKT - FAST Interrupt Error TTY Packet Address

FTYP - FAST Fail Print Flag

HBFAILCNT - TIU Heartbeat Failure Counter

HBFAST - TIU Heartbeat Receipt Time

HBSEQLK - TIU Heartbeat Sequence Lock

HBSEQCNT - TIU Heartbeat Sequence Counter

HBSEQFAIL - TIU Heartbeat Sequence Number Failure

The following FAST status messages are output to the TTY:

where; HH = hours MM=minutes SS=seconds and XXXX= 1 to 99 or 'MANY'

HH:MM:SS TIU EXTERNAL INTERRUPT PARITY - XXXX
HH:MM:SS TIU INPUT BUFFER PARITY - XXXX
HH:MM:SS TIU ILLEGAL INTERRUPT - XXXX
HH:MM:SS TIU DETECTED BAD MESSAGE CODE - XXXX
HH:MM:SS TIU HEARTBEAT FAILURE - XXXX
HH:MM:SS TIU LOW PRIORITY STACK OVERFLOW - XXXX
HH:MM:SS TIU HIGH PRIORITY STACK OVERFLOW - XXXX
HH:MM:SS TIU RECEIVED DATA WITH BAD PARITY - XXXX
HH:MM:SS TIU OUTPUT CHANNEL TIMEOUT - XXXX
HH:MM:SS TIU AUTO DISABLE SECTORS BEHIND - XXXX
HH:MM:SS TIU HEARTBEAT SEQUENCE FAILURE - XXXX
HH:MM:SS TIU ILLEGAL INPUT MESSAGE CODE - XXXX

HH:MM:SS TIU SYSTEM FAILURE

Table Locks

FSTOUTA - FAST Output Channel Active Flag

HBSEQLK - TIU Heartbeat Sequence Lock

System Parameters

RTCQ - IOP Internal RTC Memory Location

Site Parameters

FASTPQ - FAST Relative Peripheral Number

FINTMEQ - FAST Initialization Fail Time

FSTILMPQ - FAST Illegal Input Message Code

FSTFAILQ - FAST Fail Display Time-out

HBFAILQ - FAST Heartbeat Fail Time-out

HBSEQNQ - FAST Heartbeat Sequence Fail Count

HBSEQMAX - TIU Maximum Heartbeat Sequence Number

3.3.99.3 Design Logic. The Final Approach Spacing Tool (FAST) Interrupt Error Handler (FSTERRP) is the second stage of the two stage FAST interrupt process. FSTERRP performs all second stage interrupt processing, including channel time-out failure and error message output. The following processing logic is performed by FSTERRP:

A. Upon entrance, a Store Input BCW (ESSBIN) Executive Service Request (ESR) is performed, storing the FAST channel input Buffer Control Word (BCW) into the FAST Current Input Buffer Pointer (FSTINBCW) table. If an ESSBIN ESR packet error occurs, processing proceeds to label FSTERPA1 (see section 3.3.99.3.G) to declare a FAST failure. FSTINBCW is cleared of all extraneous data other than the current 18-bit buffer address. The FAST Old Input Buffer Pointer (FSTINBCW1) is loaded into both the A-Register and B4-Register. The Current Input Buffer Pointer is subtracted from the A-Register to determine if an input message has been received on the FAST channel. If the pointers are equal, indicating that a FAST message has not been received, processing proceeds to label FSTERPA (see section 3.3.99.3.F) to perform FAST Heartbeat/Channel Failure processing. FSTINBCW1 is updated to reflect the new Current Input Buffer Pointer by storing the value of FSTINBCW into FSTINBCW1.

B. FSTERRP1 performs input buffer message processing. If the FAST On/Off Flag (FASTFLG) indicates that the FAST function is 'OFF' processing proceeds to label FSTERRPA (see section 3.3.99.3.H) to process any possible interrupt routine failures. Utilizing the Old Input Buffer Pointer (B4-Register) as the start of message pointer, the message code field is checked for a value of one (1), indicating that the message is a TIU Heartbeat. If the message is a Heartbeat, processing proceeds to label FSTERPC (see section 3.3.99.H) to perform heartbeat wrap around logic. Otherwise, the FAST Illegal Input Message Code Counter (FSTILMES) is incremented. If the count exceeds the (SV) FAST Illegal Input Message Code (FSTILMPQ) parameter, the FAST Interrupt Error Flag (FSTMCERP) is set to request FAST system failure and TTY output processing. Processing then proceeds to label FSTERPA (see section 3.3.99.3.F) to perform FAST Heartbeat/Channel Failure processing.

C. The FSTERPC routine is performed when a Heartbeat message is received from the TIU. If the TIU Heartbeat Sequence Lock (HBSEQLK) is not locked, processing proceeds to label FSTERPC1 (See section 3.3.99.3.D) to count the new Heartbeat. If the TIU Heartbeat Sequence Counter (HBSEQCNT) is greater than the (SP) TIU Maximum Heartbeat Sequence Number (HBSEQMAX) parameter the sequence lock is cleared to reset HBSEQCNT. Normal processing proceeds to label FSTERPC1 (See section 3.3.99.3.D) to count the new Heartbeat.

D. The FSTERPC1 routine performs Heartbeat count validation by obtaining the Heartbeat Sequence Count field from the TIU Heartbeat Message (upper 15 bits of the address specified by B4-Register). If HBSEQLK is not set, then this is either the first Heartbeat Message received or the Heartbeat Sequence Count has wrapped around. In either case, HBSEQLK is set via a Test and Set Flag (TSF) instruction and the count from the Heartbeat Message is stored in HBSEQCNT. If HBSEQLK is set, HBSEQCNT is incremented and the TIU Heartbeat Message Sequence Count is subtracted from the new count in HBSEQCNT. If the counts are equal, processing proceeds to label FSTERPD (See section 3.3.99.3.E) to process the heartbeat. Otherwise, the TIU Heartbeat Sequence Number Failure (HBSEQFAIL) is incremented. If HBSEQFAIL is greater than the (SV) FAST Heartbeat Sequence Fail Count (HBSEQNQ) parameter, FSTMCERP is set to request FAST system failure and TTY output processing. Processing then proceeds to label FSTERPA (see section 3.3.99.3.F) to perform FAST Heartbeat/Channel Failure processing.

E. The FSTERPD routine is performed when a valid TIU Heartbeat Message is received. The Real Time Clock (RTCQ) is stored into HBFAST to mark the time of receipt. The input buffer pointer (B4-Register) is incremented to the next input buffer location, unless the FAST Last Input Buffer Address (FSTINLST) is reached. In this case, the B4-Register is reset to the FAST Input Buffer Address (FSTINBUFA). If no other messages are to be processed, processing proceeds to label FSTERPA (see section 3.3.99.3.F) to perform FAST Heartbeat/Channel Failure processing. Otherwise, the next message is processed at lab

FSTERRP1 (see section 3.3.99.3.B).

F. The FSTERPA routine performs heartbeat failure and channel time-out logic. If FASTFLG indicates that the FAST function is 'OFF', processing proceeds to label FSTERRPA (see section 3.3.99.3.H) to process possible interrupt routine failures. Otherwise, HBFAST is subtracted from RTCQ and adjusted for clock wrap around. If the time difference does not exceed the (SV) FAST Heartbeat Fail Time-out (HBFAILQ) parameter, normal processing proceeds to label FSTERRPA. If the parameter is exceeded, the current value of RTCQ is stored into HBFAST and the TIU Heartbeat Failure Counter (HBFAILCNT) is incremented. Error processing continues to label FSTERPA1 (see section 3.3.99.3.G) to declare a FAST system failure.

G. The FSTERPA1 routine is performed when either a TIU Heartbeat Failure occurs or the FAST E$SBIN ESR packet error exit is taken. FSTMCERP is set to request FAST system failure and TTY output processing. Processing continues at label FSTERRPA (see section 3.3.99.3.H) to process the failure.

H. The FSTERRPA routine performs FAST channel and interrupt termination logic. If a FAST failure is not detected, processing proceeds to label FSTERP40 (see section 3.3.99.3.I). If a failure has been declared (FSTMCERP, Lower = 077777) the following actions are taken on the FAST channel:

1. All Input/Output (I/O) is terminated, via a Terminate Input/Output ESR (E$TIO).
2. A Disable Channel (E$D) ESR is performed to disable channel interrupts and external status word storage.
3. Label FSTERP10 is the ESR packet error exit for the FAST E$TIO and E$D ESR packets.
4. The FAST Output Channel Active Flag (FSTOUTA) is cleared.
5. The FAST Processing Loss (FASTFAIL) flag is set.
6. The FAST Initialization Flag (FINITFLG) is cleared.
7. The FAST On/Off Flag (FASTFLG) is set to 'OFF'.
8. The FAST Interrupt Error Flag (FSTMCERP, Lower) is cleared.

I. The FSTERP40 routine performs initial FAST TTY output request processing if FSTMCERP 'Upper' is set. If not set, processing proceeds to label FSTERP62 (see section 3.3.99.3.M) to perform FAST time-out processing. FSTMCERP 'Upper' is set to zero, clearing the TTY processing request. The B4-Register is initialized to the total number of FAST error message. Processing continues to label FSTERP50 (see section 3.3.99.3.J).

J. The FSTERP50 routine performs a search for FAST TTY error messages marked for output. If an entry into the FAST Error Counter Table (FSTEXTPR to FSTILMES) is none zero, the following processing occurs:

1. The FAST Error Message Output Time (FSTMERR) is subtracted from RTCQ and adjusted for clock wrap around. If the time difference does not exceed the (SV) FAST Error Message Output Parameter Table (FSTERTM), the output of the error message is delayed and processing proceeds to label FSTERP55 (see section 3.3.99.3.K).
2. A call is made to the local subroutine Convert Octal to Field Data (CONVFD) (see section 3.3.99.3.Q) to convert and store the number of errors obtained from the specified FAST Error Counter Table entry.
3. The B7-Register is loaded with the FAST Interrupt Error TTY Packet Address (FSTTYPKT) table entry for the specified error message.
4. An Indirect ESR Request With B7 Index Register (E$IDRB) is performed for the FAST Error Message Packet addressed by Register B7. If an ESR packet error occurs processing proceeds to label FSTERP62 (see section 3.3.99.3.M).

5. The following error messages are output to the TTY, utilizing B4 as the index into the FAST Error Counter Tables Error Message Packets:

where; HH = hours MM=minutes SS=seconds and XXXX= 1 to 99 or 'MANY'

Index	Error Counter Label	Message Content	Number of Occurrences
0	FSTEXTPR	HH:MM:SS TIU EXTERNAL INTERRUPT PARITY	- XXXX
1	FSTINPER	HH:MM:SS TIU INPUT BUFFER PARITY	- XXXX
2	FSTILERR	HH:MM:SS TIU ILLEGAL INTERRUPT	- XXXX
3	FSTMSGCD	HH:MM:SS TIU DETECTED BAD MESSAGE CODE	- XXXX
4	HBFAILCNT	HH:MM:SS TIU HEARTBEAT FAILURE	- XXXX
5	FSTSTKER	HH:MM:SS TIU LOW PRIORITY STACK OVERFLOW	- XXXX
6	FSTSTKER+1	HH:MM:SS TIU HIGH PRIORITY STACK OVERFLOW	- XXXX
7	FSTOPAR	HH:MM:SS TIU RECEIVED DATA WITH BAD PARITY	- XXXX
8	FSTCHNER	HH:MM:SS TIU OUTPUT CHANNEL TIMEOUT	- XXXX
9	FSTSECBH	HH:MM:SS TIU AUTO DISABLE SECTORS BEHIND	- XXXX
10	HBSEQFAIL	HH:MM:SS TIU HEARTBEAT SEQUENCE FAILURE	- XXXX
11	FSILMES	HH:MM:SS TIU ILLEGAL INPUT MESSAGE CODE	- XXXX

6. The current value of RTCQ is stored into FSTMERR to mark the output time of the specified error message.

7. The specified FAST Error Counter Table entry (FSTEXTPR to FSTILMES) is cleared and processing proceeds to label FSTERP60 (see section 3.3.99.3.L) to process the next error message.

K. The FSTERP55 routine reschedules TTY processing for those error message that did not meet the elapsed output time interval by setting the FSTMCERP (upper) negative. If the FAST function is 'ON' (FASTFLG) processing proceeds to label FSTERP60 (see section 3.3.99.3.L). Otherwise, The FAST Error Counter Table entry and the TTY Request (FSTMCERP) are cleared. Processing proceeds to label FSTERP60 (see section 3.3.99.3.L) to process the next error message.

L. The FSTERP60 routine performs error message loop control processing. The B4-Register is decremented after each error message counter is interrogated for processing requirements. If all error counters have been checked, processing proceeds to FSTERP62 (see section 3.3.99.3.M). Otherwise, processing proceeds to label FSTERP65 (see section 3.3.99.3.O) to process the next error message.

M. The FSTERP62 routine performs initialization time-out processing. If the FAST Initialization Flag (FINTFLG) is not set, processing proceeds to label FSTERP63 (see section 3.3.99.3.N). If it is set, the TIU Initialization Start Time (FINTTME) is subtracted from the RTCQ and adjusted for clock wrap around. If the time difference exceeds the (SV) FAST Initialization Fail Time (FINTMEQ) parameter, the FAST Processing Loss (FASTFAIL) flag is set. Processing continues to label FSTERP63 (see section 3.3.99.3.N).

N. The FSTERP63 routine performs FAST failure message processing. If FASTFAIL is not set, processing proceeds to label FSTERP88 (see section 3.3.99.3.P). Otherwise the following processing will occur:

1. If the FAST Fail Print Flag (FTYP) is set indicating that the system failure message has already been queued for printing, processing proceeds to label FSTERP65 (see section 3.3.99.3.O).

2. An Indirect ESR Request (ESIDR) is performed for the TIU System Failure Message Packet (FSTFLPKT). If an ESR packet error occurs, processing proceeds to label FSTERP88 (see section 3.3.99.3.P).

3. The current value of RTCQ is stored into the FAST Fail Display Time (FSTFAILT) table to mark the start time for display of 'FAST FAIL' in the System Data Area (SDA).

4. The FAST Display Fail Time-out (FAILTO, Lower) flag is set to notify the PDOP task program to display 'FAST FAIL' in the SDA.

5. Processing proceeds to label FSTERP88 (see section 3.3.99.3.P).

O. The FSTERP65 routine performs FAST failure time-out logic. FSTFAILT is subtracted from the RTCQ and adjusted for clock wrap around. If the time difference does not exceed the (SV) FAST Fail Display Time-out (FSTFAILQ) parameter, processing proceeds to label FSTERP88 (see section 3.3.99.3.P). Otherwise the following logic is performed:

1. The FAST Display Fail Time-out (FAILTO, Upper) flag is set to notify PDOP to stop the display of 'FAST FAIL' in the SDA.

2. The FASTFLG, FINTFLG, FASTFAIL and FTYP flags are cleared and processing continues to label FSTERP88 (see section 3.3.99.3.P).

P. The FSTERP88/FSTERP90 exit task processing via an ESEXIT ESR.

Q. The CONVFD local subroutine performs Octal to Field Data conversions. Upon entrance, the A-Register is compared for a value greater than 99. If the input error count is not greater than 99 decimal, the octal number is converted to Field Data. If the count is greater than 99, then the Field Data mask 'MANY' is obtained. The Field Data mask or converted number is then stored into the buffer specified by the B4-Register and utilizing an Execute Remote instruction contained in the FAST Execute Remote Instruction Table (CONVFDXR).

3.3.99.4 Flowcharts.

See figure 3.3.99-1.

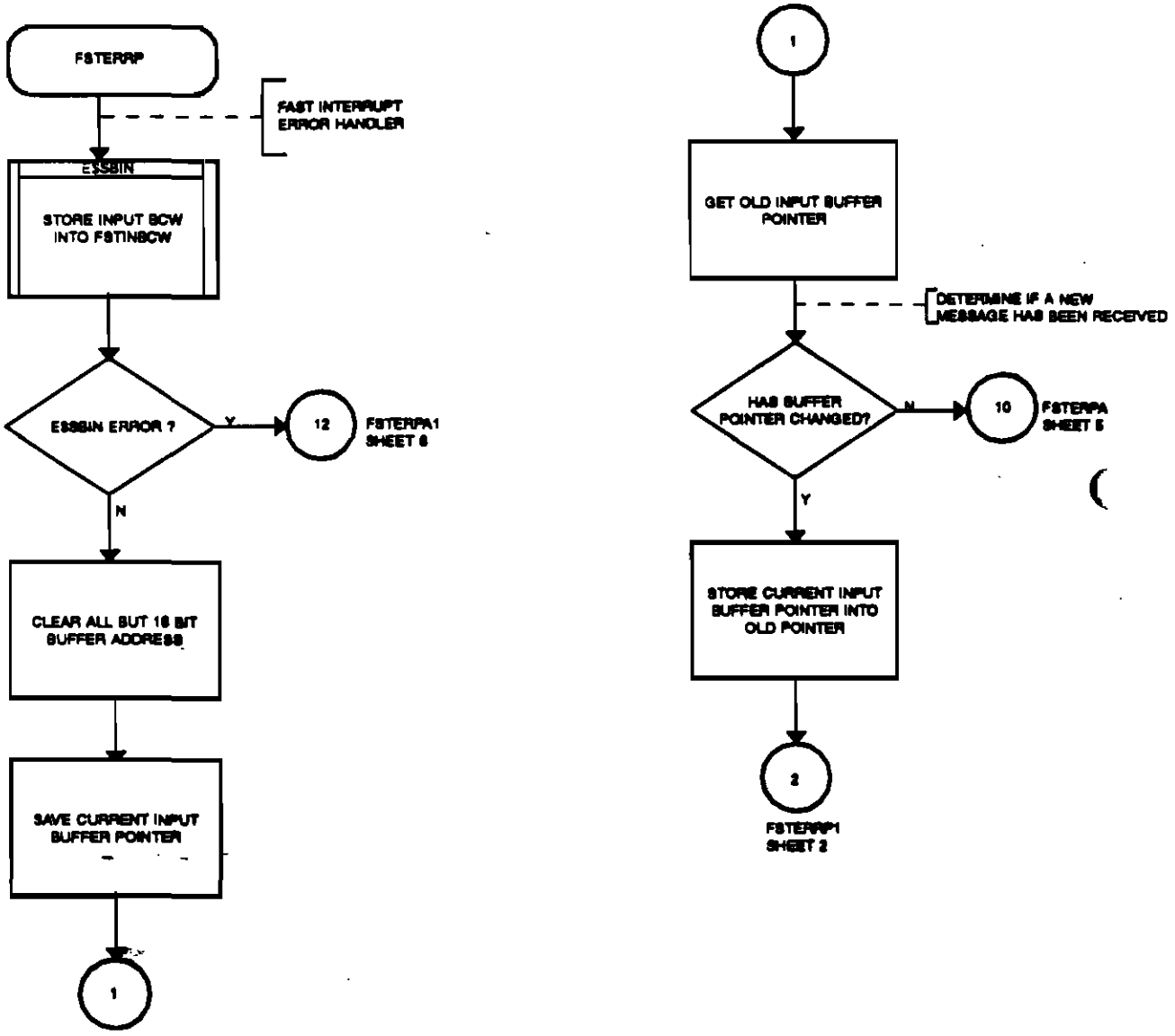


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 1)

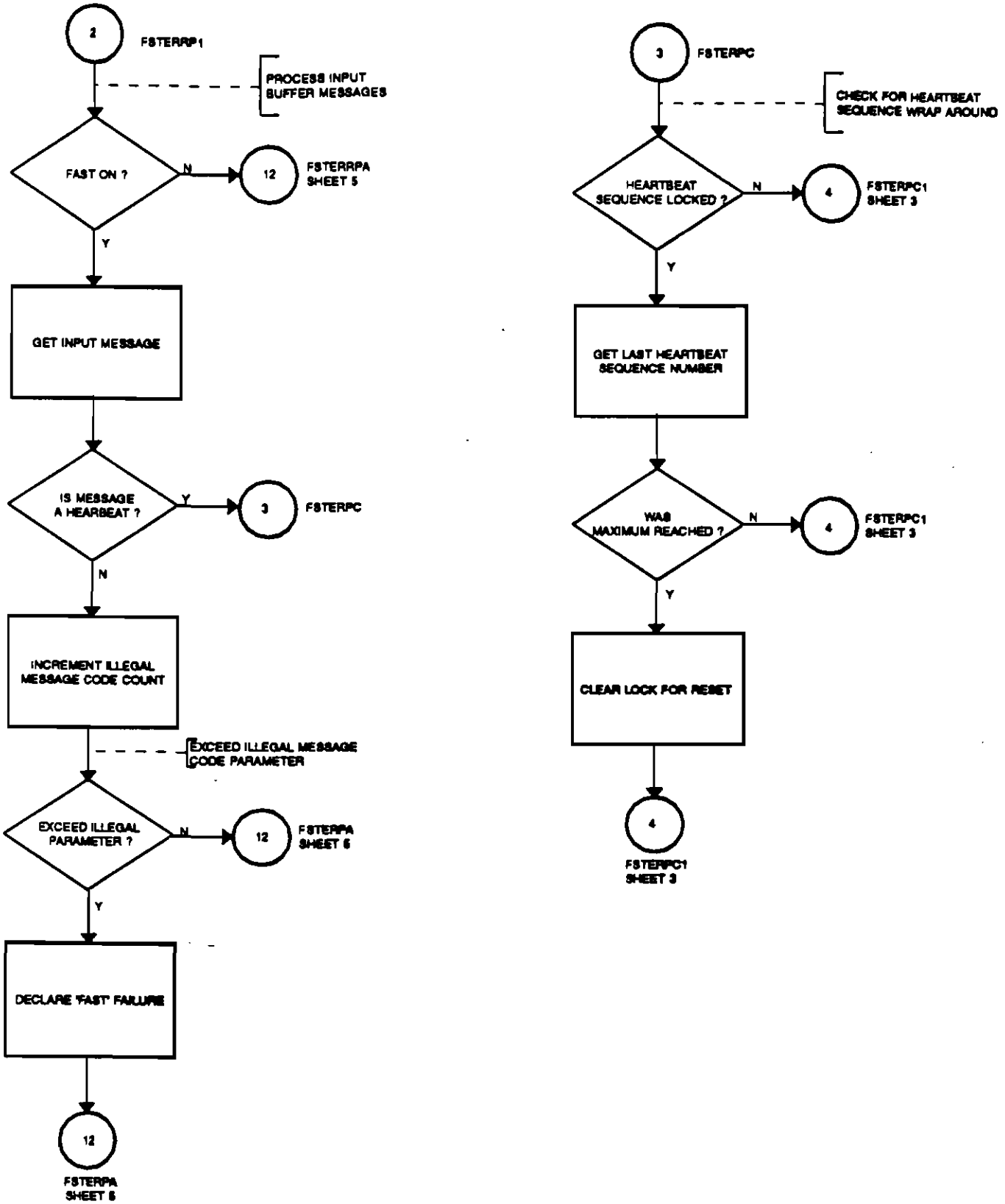


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 2)

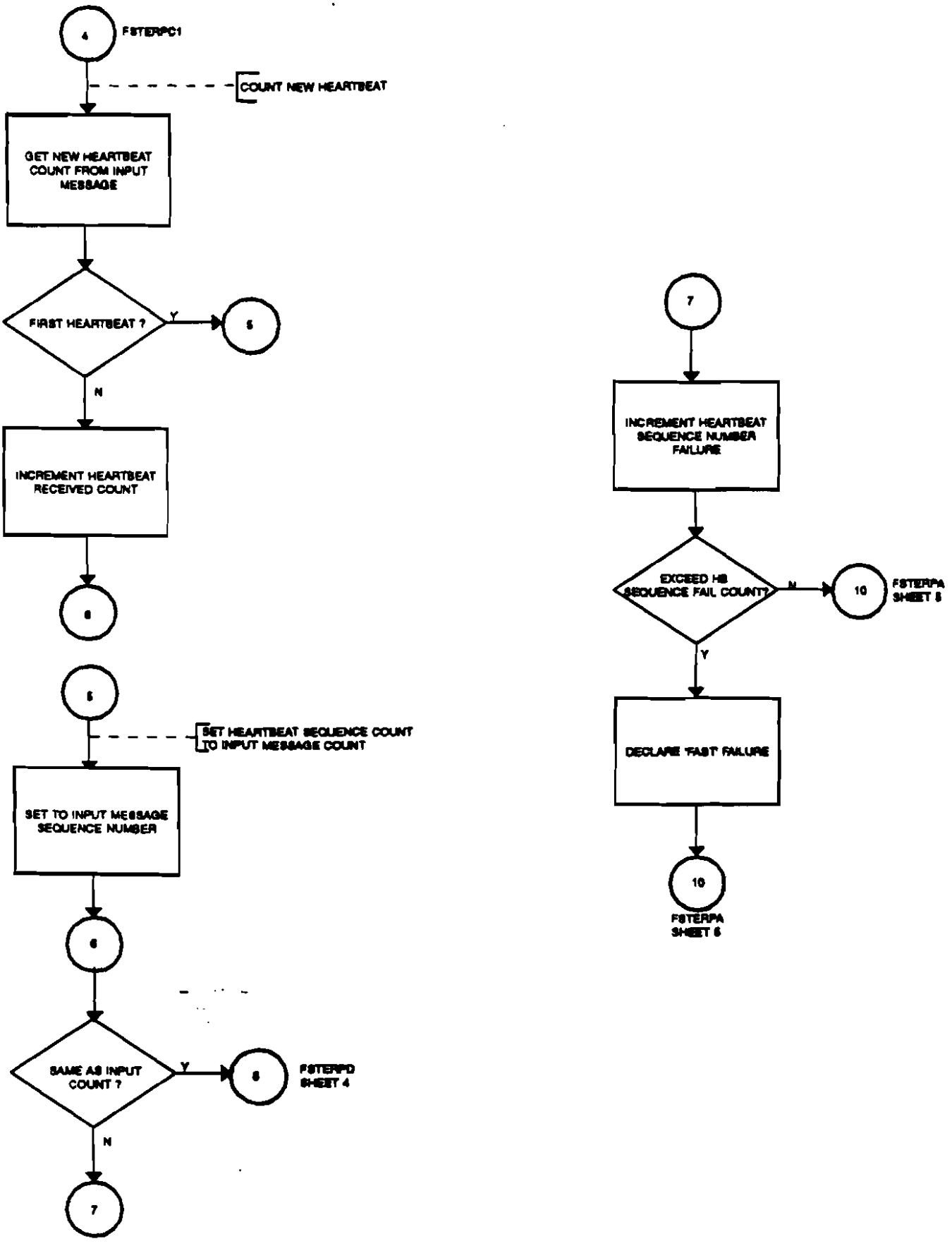


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 3)

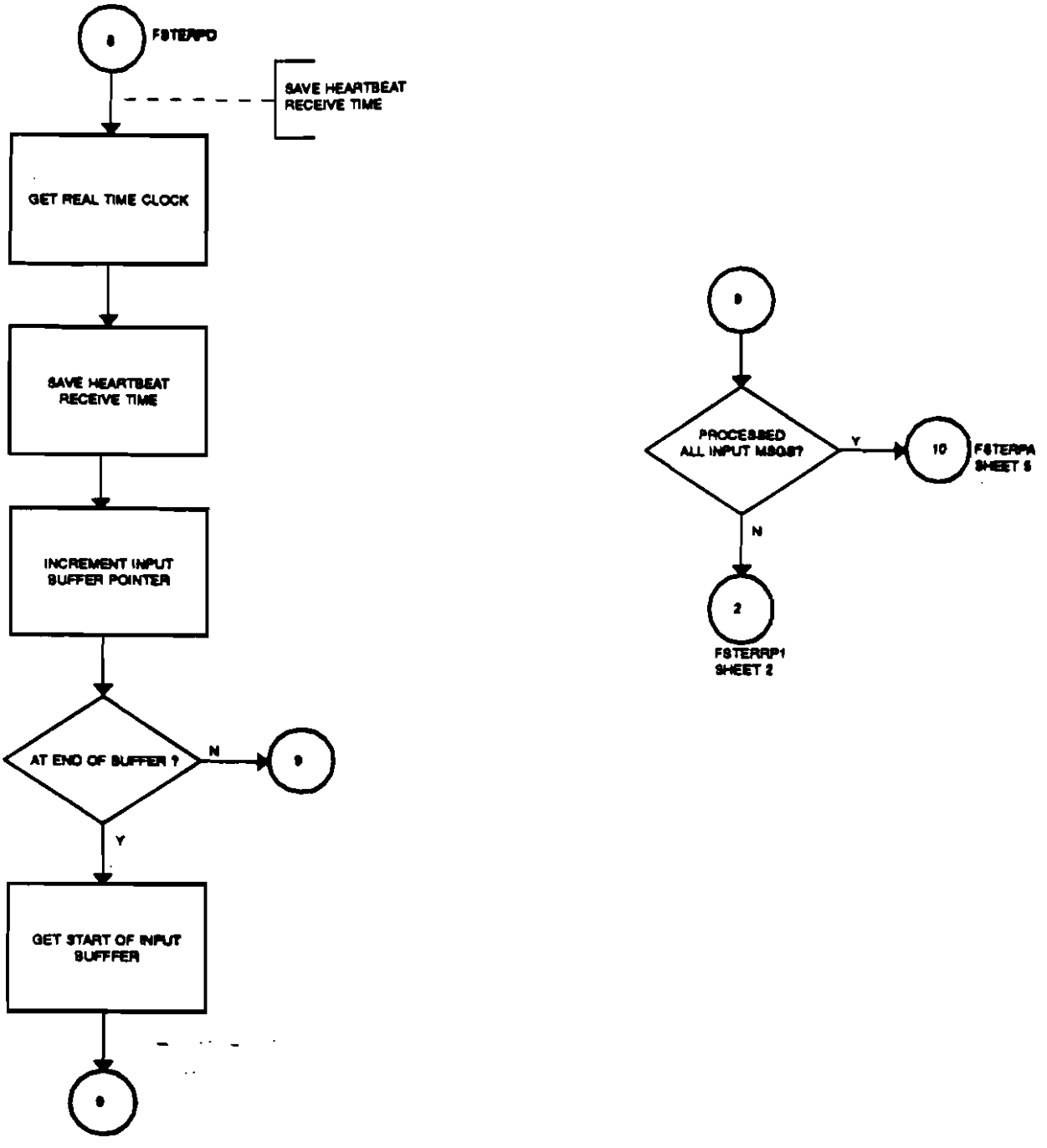


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 4)

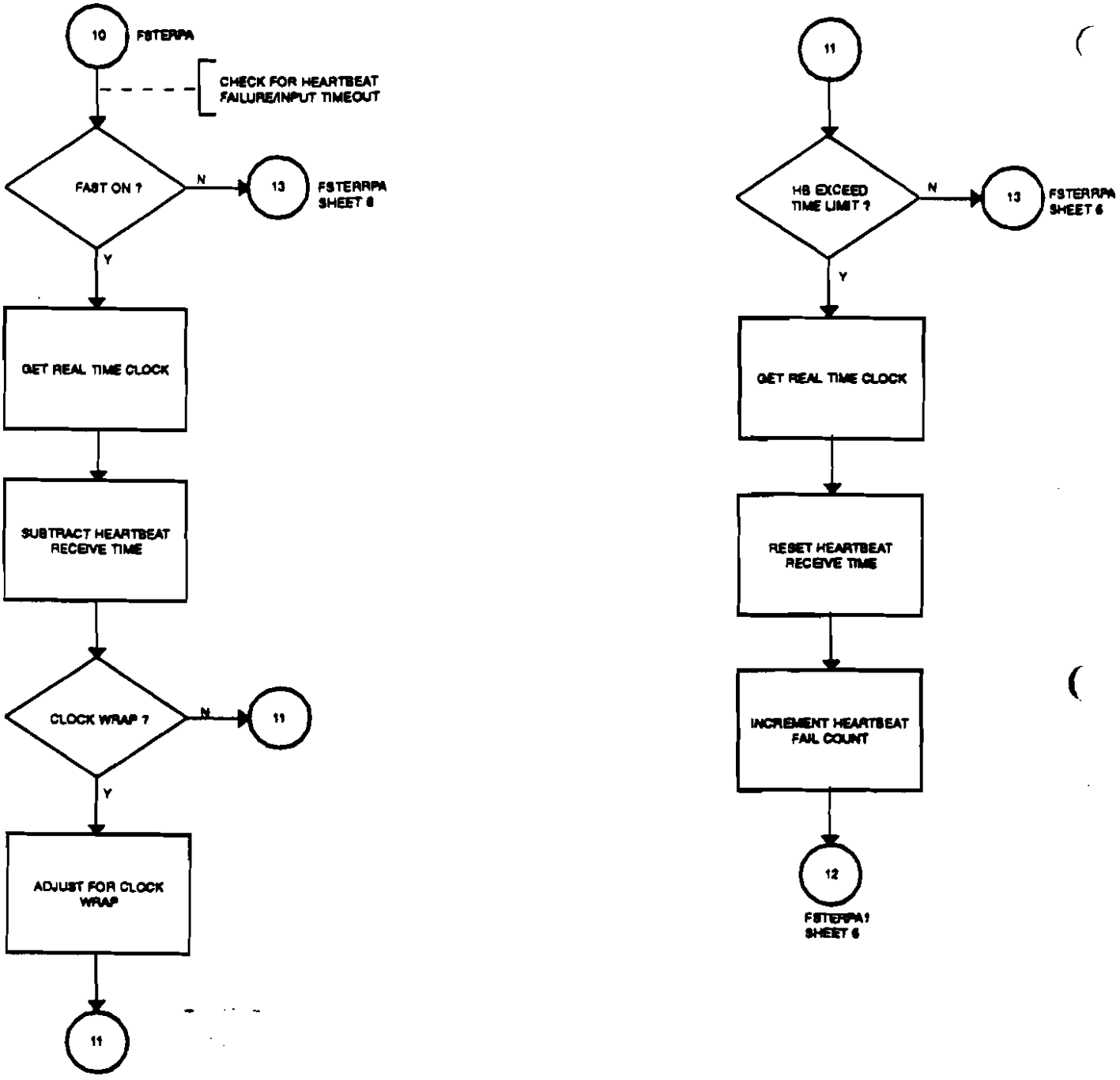


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 5)

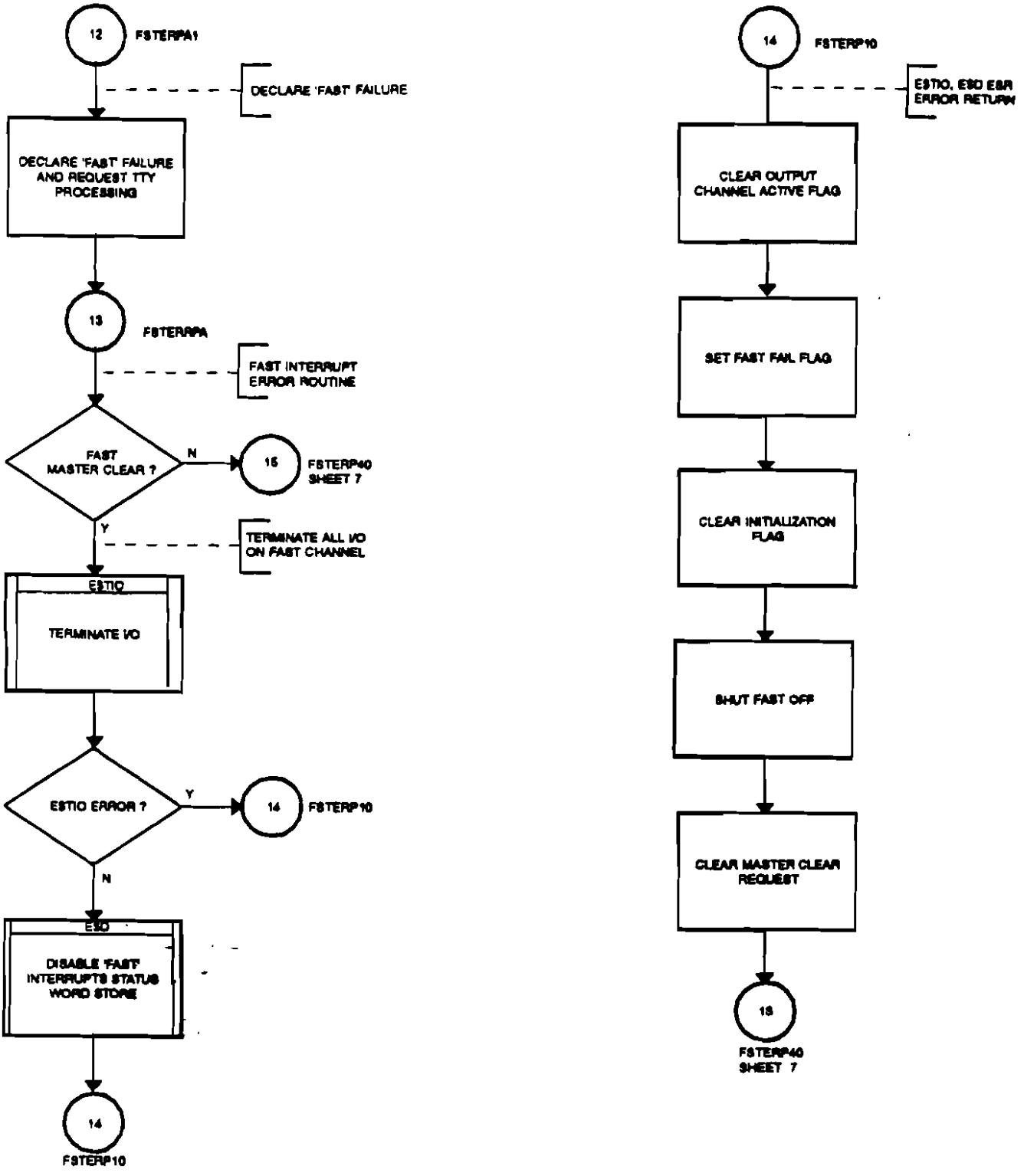


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 6)

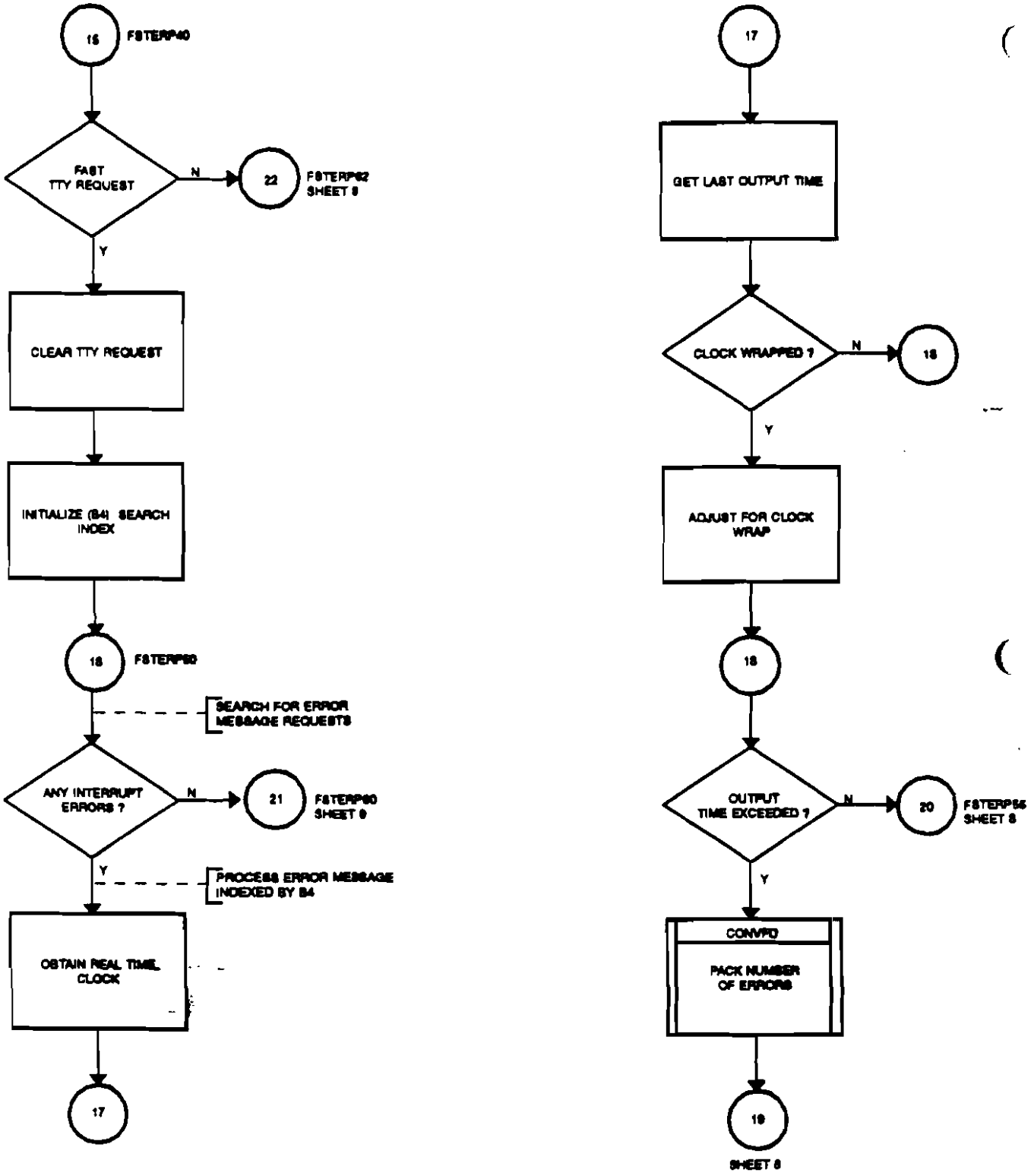


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 7)

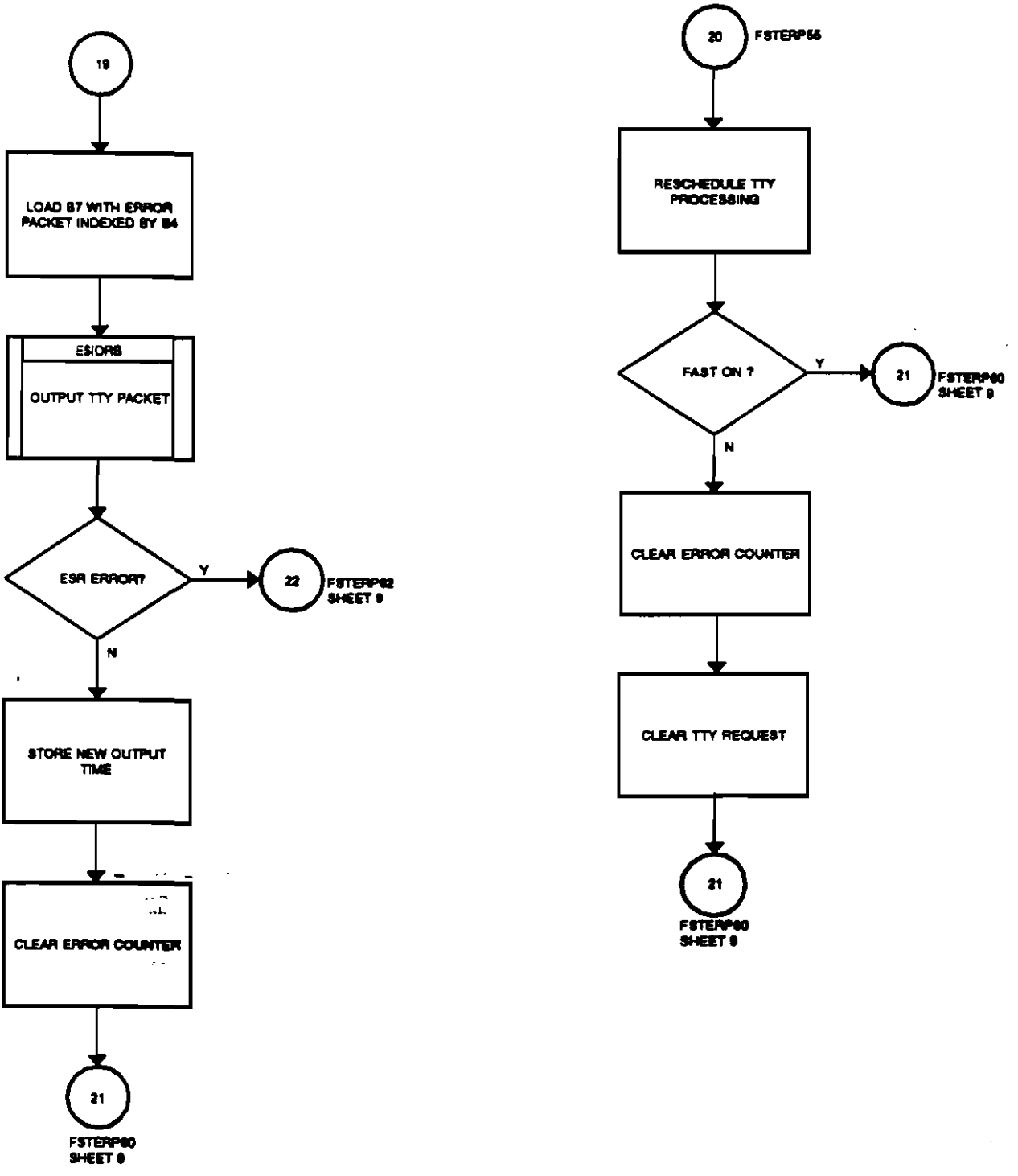


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 8)

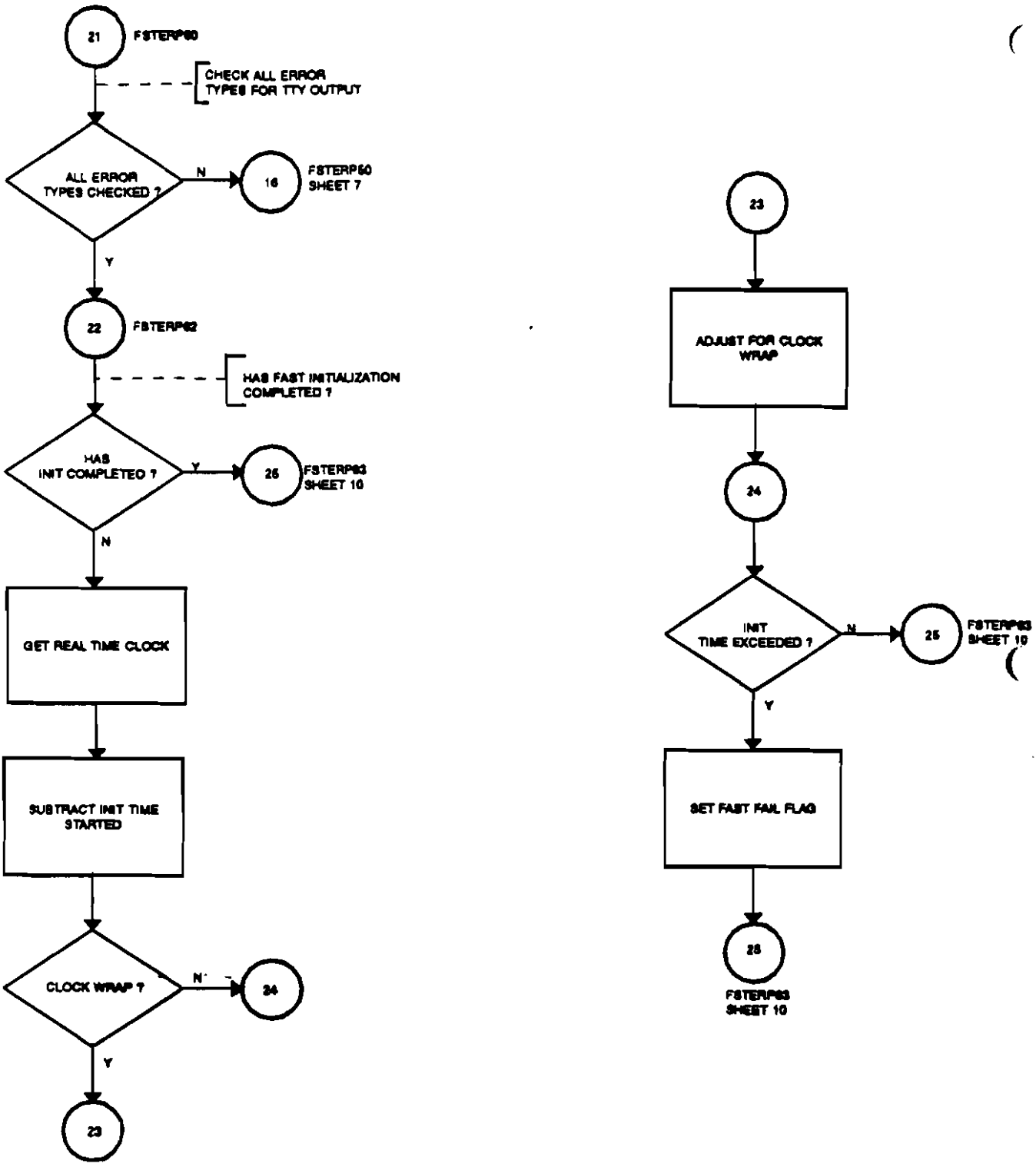


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 9)

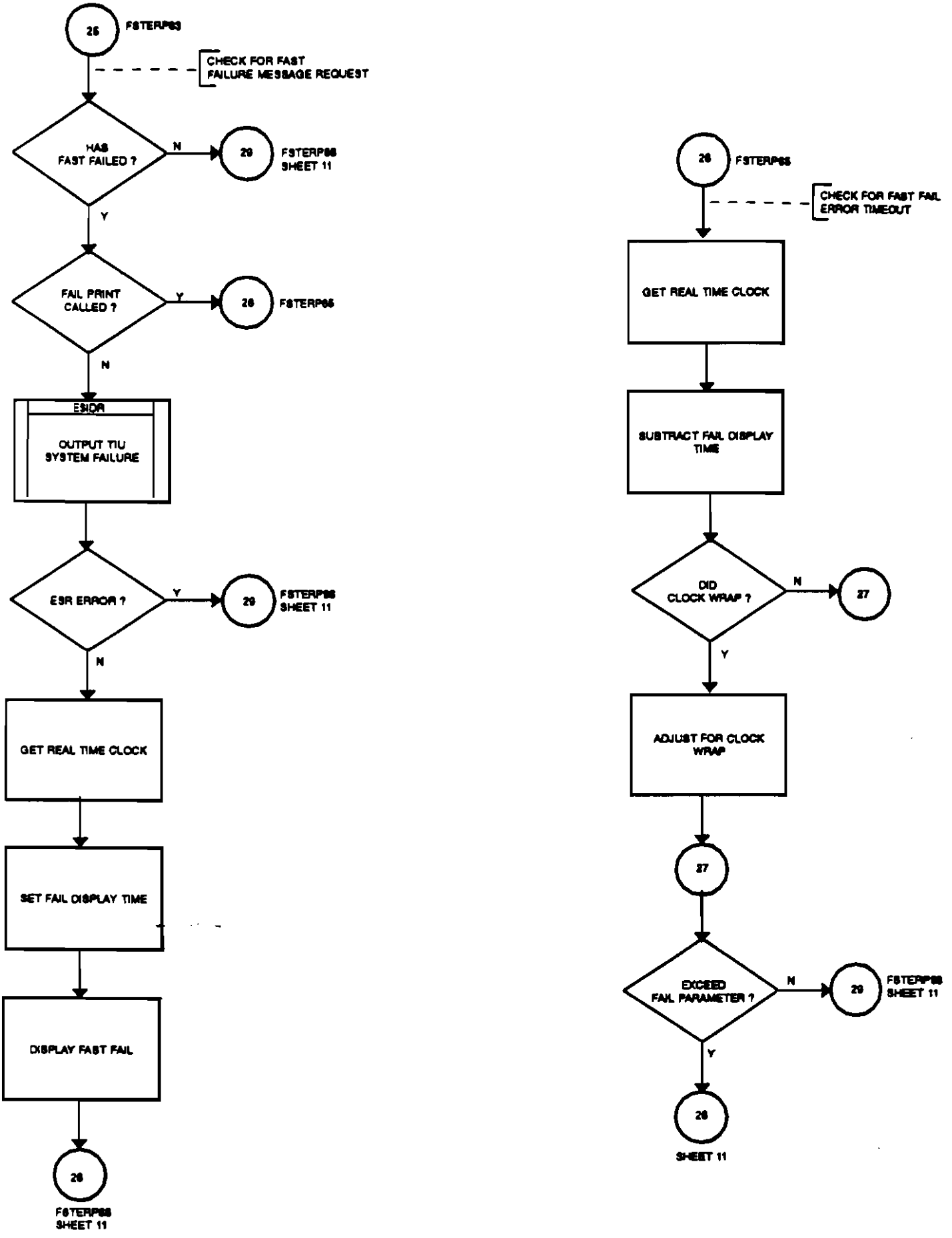


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 10)

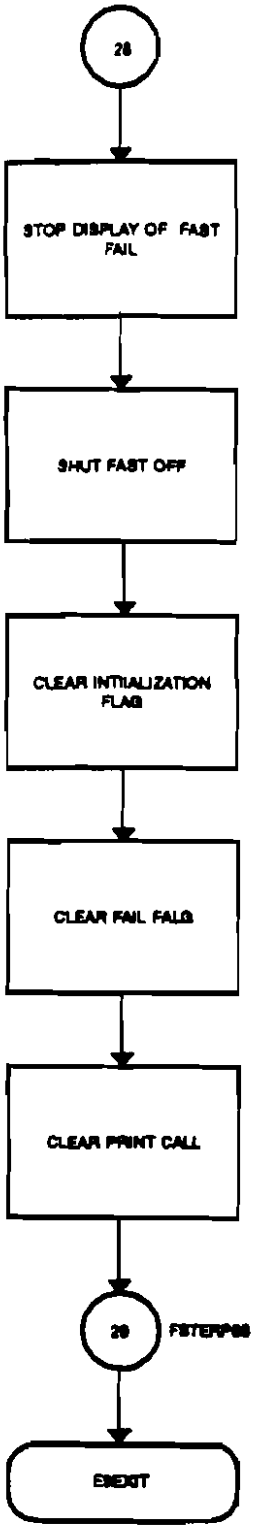


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 11)

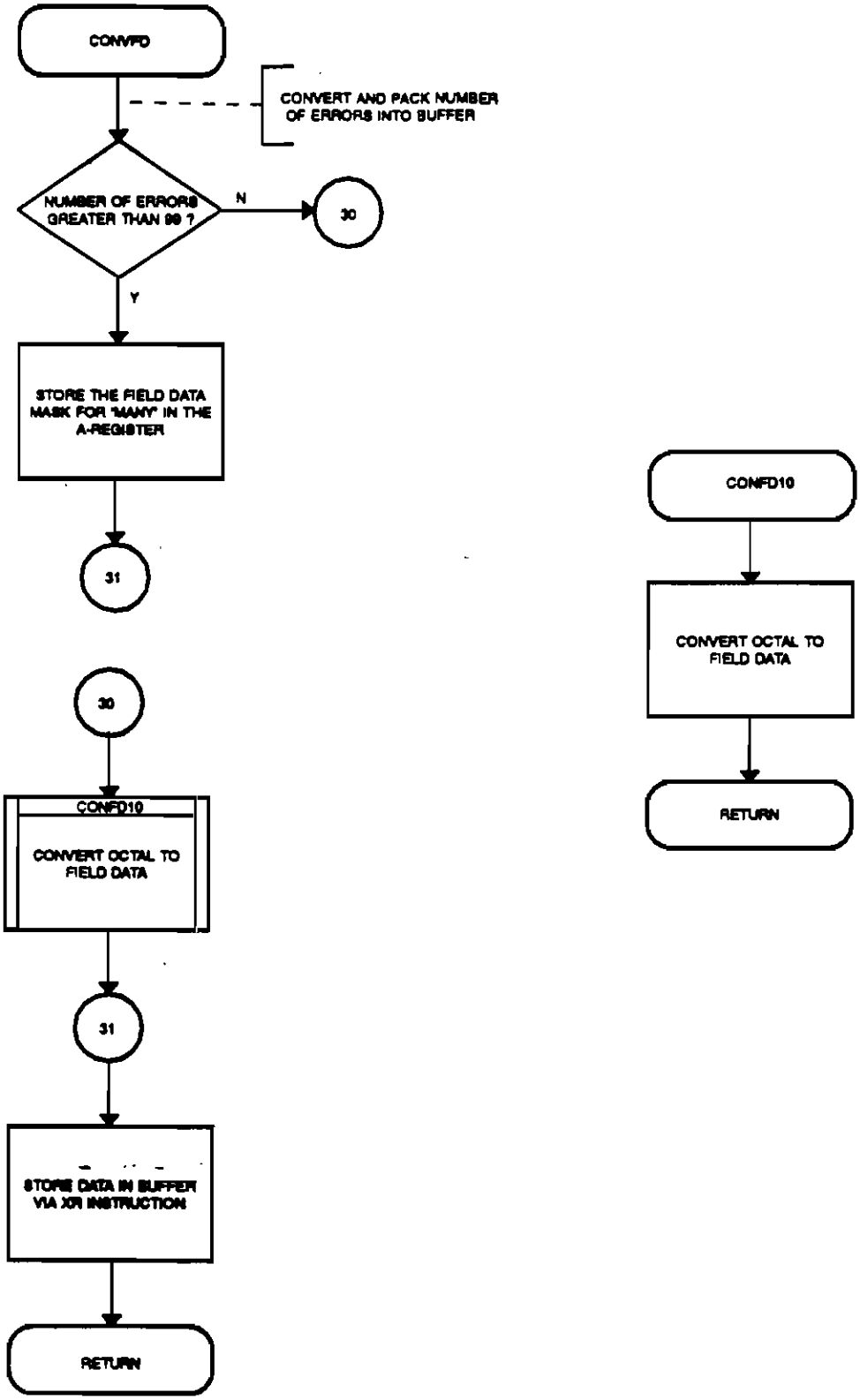


FIGURE 3.3.99-1 FAST INTERRUPT ERROR PROCESSING FLOWCHART (SHEET 12)