

**NETWORK FABRIC EXERCISE FUNCTION  
SOFTWARE SYSTEM DESCRIPTION  
NO. 3 ELECTRONIC SWITCHING SYSTEM**

	CONTENTS	PAGE	
	<b>1. GENERAL</b> . . . . .	<b>1</b>	<b>1.04</b> The following Bell System Practices provide background information related to No. 3 ESS network fabric exercise processing:
	<b>2. NFEX PROGRAM</b> . . . . .	<b>5</b>	(a) TTY Software Description, Section 233-152-120
	<b>3. GLOSSARY</b> . . . . .	<b>9</b>	(b) Peripheral Control Software Description, Section 233-151-155
			(c) Tape Operations Software Description, Section 233-152-130
			(d) Basic Call Processing Software Description, Section 233-151-130
			(e) Network Path Hunt Description, Section 233-151-140
			(f) Programmed Maintenance Aids, Section 233-153-105
			(g) EPL Instruction Set, Section 233-154-116.
<b>Figures</b>			
	<b>1. Loading Of Office Function (Magnetic Tape) To On-Line Memory (Temporary Store)</b> . . . . .	<b>2</b>	
	<b>2. Network Fabric Exercise Program</b> . . . . .	<b>3</b>	
	<b>3. Network Fabric Exercise Program Flow Diagram</b> . . . . .	<b>6</b>	
	<b>4. Request Buffer</b> . . . . .	<b>7</b>	
	<b>5. Example of a Network Fabric Exercise Failure Message</b> . . . . .	<b>7</b>	
<b>1. GENERAL</b>			
	<b>1.01</b> This section describes the network fabric exercise function (performed by program NFEX) for the No. 3 Electronic Switching System (ESS) offices. The NFEX program is nonresident (stored on magnetic tape) and must be loaded into temporary memory before testing the input switch, output switch, test vertical, and junctor switch crosspoints.		<b>1.05</b> The following programs contain codes and comments that detail network fabric exercise processing in the No. 3 ESS.
	<b>1.02</b> When this section is reissued, the reason for reissue will be listed in this paragraph.		(a) Network Fabric Exercise Program (NFEX), PR-3H313, is used to test all 2-wire (tip and ring) talk path connections in the No. 3 ESS network.
	<b>1.03</b> Part 3 is a glossary containing terms and definitions used in this section.		(b) Common Base Level Monitor Program (CBLM), PR-1C950, is the focal point of the base level system and determines the sequencing of all programs.

**NOTICE**

Not for use or disclosure outside the  
Bell System except under written agreement

(c) Application Portion of the Base Level Monitor (BLMMA), PR-3H004, performs two basic functions:

- (1) BLMMA provides tables that are used by CBLM.
- (2) BLMMA invokes routines that are executed once per base level loop.

(d) Peripheral Order Interpreter Program (POINT), PR-3H168, interprets data words from the peripheral catalog (PCAT) and translates these words into peripheral orders.

(e) Network Path Hunt, Busy, and Idle Program (PATHNT), PR-3H166, provides the No. 3 ESS algorithm for maintaining the proper busy/idle status of the network links by means of correct updating of a network map in call store (NETMAP).

(f) Base Level TCR Scan Program (TCRSCN), PR-3H174, is the scan monitor for the transient call record (TCR) in the base level loop.

(g) Scan Point Number Translation Program (XSLSPN), PR-3H179, translates the scan point number into line or trunk features.

**1.06** The software document PK-3H301-01 contains procedures for locating network fabric shorts and guidelines for the network group-check bypass mode.

**1.07** Interrelationships between NFEX and other programs are shown in Fig. 1. NFEX is a nonresident program that must be transferred from magnetic tape to temporary store (Fig. 2). A manual request from the teletypewriter (TTY) or an automatic request from the BSYHRTRFC routine in the traffic and plant measurement program (TRAFIC) causes NFEX to be loaded in memory. Since other nonresident multiscan functions may be active, the multiscan function controller (MSFC) in the common base level monitor (CBLM) program determines whether the request will be granted or denied. When the request is granted, the MSFC initiates the loading of NFEX into temporary memory. When NFEX is resident in temporary memory, MSFC enables NFEX during the next base level loop. After completing the instructions for this entry, NFEX calls the WAIT subroutine in CBLM. CBLM returns to NFEX from the WAIT subroutine on the next base level loop. Network orders are sent when the network controller queues are empty or at the end of four base level loops ( $\approx 800$  milliseconds).

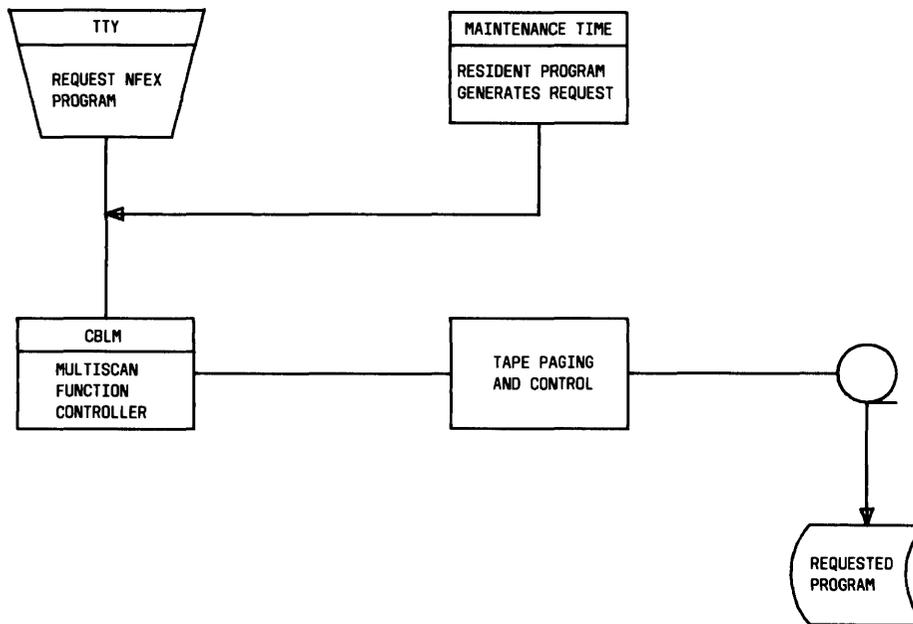


Fig. 1—Loading of Office Function (Magnetic Tape) to On-Line Memory (Temporary Store)

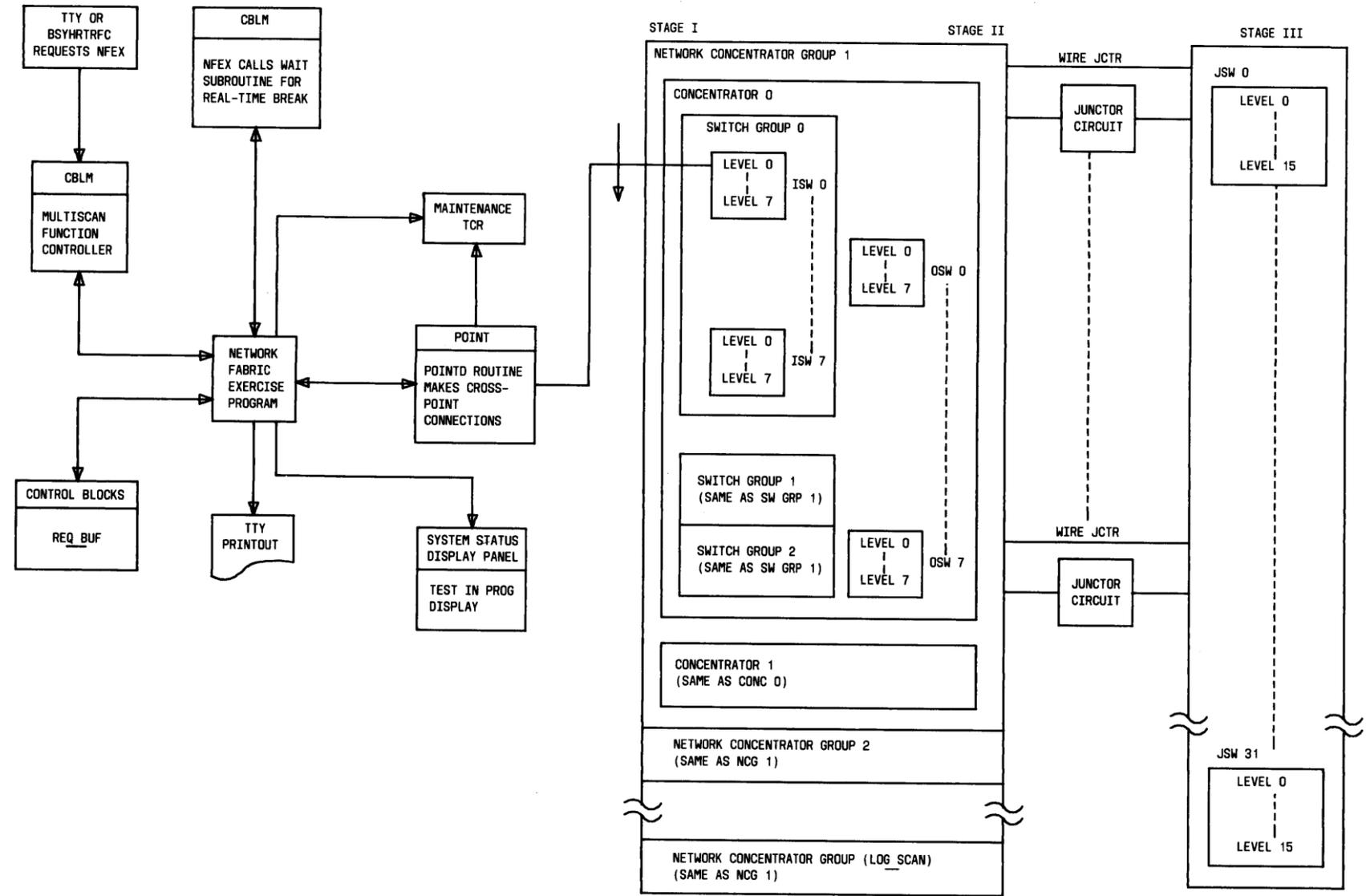


Fig. 2—Network Fabric Exercise Program



**1.08** NFEX constructs a path through the crosspoint to be tested and transfers the connection information to the POINTD routine in program POINT. This routine makes the network connection and returns on the next base level loop.

**1.09** The path is tested, and failure information is transferred to the TTY and system status display panel programs.

## 2. NFEX PROGRAM

**2.01** The input message **EX:NW!** or **EX:NW** (options)! generates an NFEX request to the MSFC. The first message tests all crosspoints in the office. Several hours are required to test all crosspoints; therefore, this message should only be used when necessary. The second message tests the portion of the network that is contained in the options. The smallest unit that can be tested is a simple switch (64 crosspoints). A group of switches also can be tested depending on the options.

**2.02** An automatic request for NFEX is generated by the BSYHRTRFC routine during the early-hour maintenance period. This request checks a concentrator group each day; therefore, several days are required for the complete network to be tested.

**2.03** If the entire network is requested to be tested or when testing a concentrator group automatically, NFEX will test the crosspoints in the following sequence:

- (a) Test verticals (TV)
- (b) Output switches (OSW), second-stage crosspoints
- (c) Input switches (ISW), first-stage crosspoints
- (d) Junctor switches (JSW), third-stage crosspoints.

**2.04** When the entire network is tested, the program begins with concentrator group 1 and exercises the TV crosspoints, then the OSW or second-stage crosspoints, and finally the ISW or first-stage crosspoints. The program repeats this pattern for each succeeding concentrator group until the last one equipped is exercised. At this point, the program finishes by exercising the JSW or third-stage crosspoints.

**2.05** The program may be run in one of several modes. In the default mode, upon the failure of a test, a TTY failure message is generated and the running of the exercise is terminated. In the unconditional (UCL) mode, the running of the exercise is not terminated upon the failure of a test. Instead, a failure message is generated, and the next sequential crosspoint is tested. When all tests have been run, a completion message is generated. In the repeat (RPT) mode, the exercise request is run repetitively until cleared by an input message. An output message is generated by the first run; therefore, a message is generated only if the result changes. The status of each run is displayed by the pass/fail lights of the system status panel (SSP). The repetition may be stopped by depressing the EXECUTE key on the SSP and may be restarted by depressing the EXECUTE key again. The step (STEP) mode is the same as the repeat mode except the exercise stops after the first run. Thereafter, the request is run once each time the EXECUTE key is depressed.

**2.06** When the MSFC receives the request for NFEX, it performs a priority check to determine if NFEX can be loaded without interference with other nonresident functions that may be currently in temporary memory. When NFEX is loaded from tape into temporary memory, CBLM will enable NFEX during the next base level loop.

**2.07** A flow diagram of the NFEX program is shown in Fig. 3. Upon entry from CBLM, NFEX loads the request buffer (REQ\_BUF) (Fig. 4) with the input parameters. These parameters determine whether NFEX tests all crosspoints or selected crosspoints as directed by the TTY input message. As the REQ\_BUF is loaded, the input data is screened and invalid data will cause an abort.

**2.08** When the REQ\_BUF has been loaded with valid input data, various counters located in the paging buffer are cleared. These counters keep track of the number of crosspoints that have passed (TESTS\_PASS), that have failed (TESTS\_FAIL), and that are not tested due to busy lines, paths, etc. (NO\_LINES, LINES\_BUSY, NO\_PATH, NO\_TV). An additional counter (FAIL\_MSG) counts failure messages that have been printed; and when the number of failure messages reaches 50, the network exercise is terminated. At the end of the exercise, these counters are dumped to provide a record of the network fabric health.

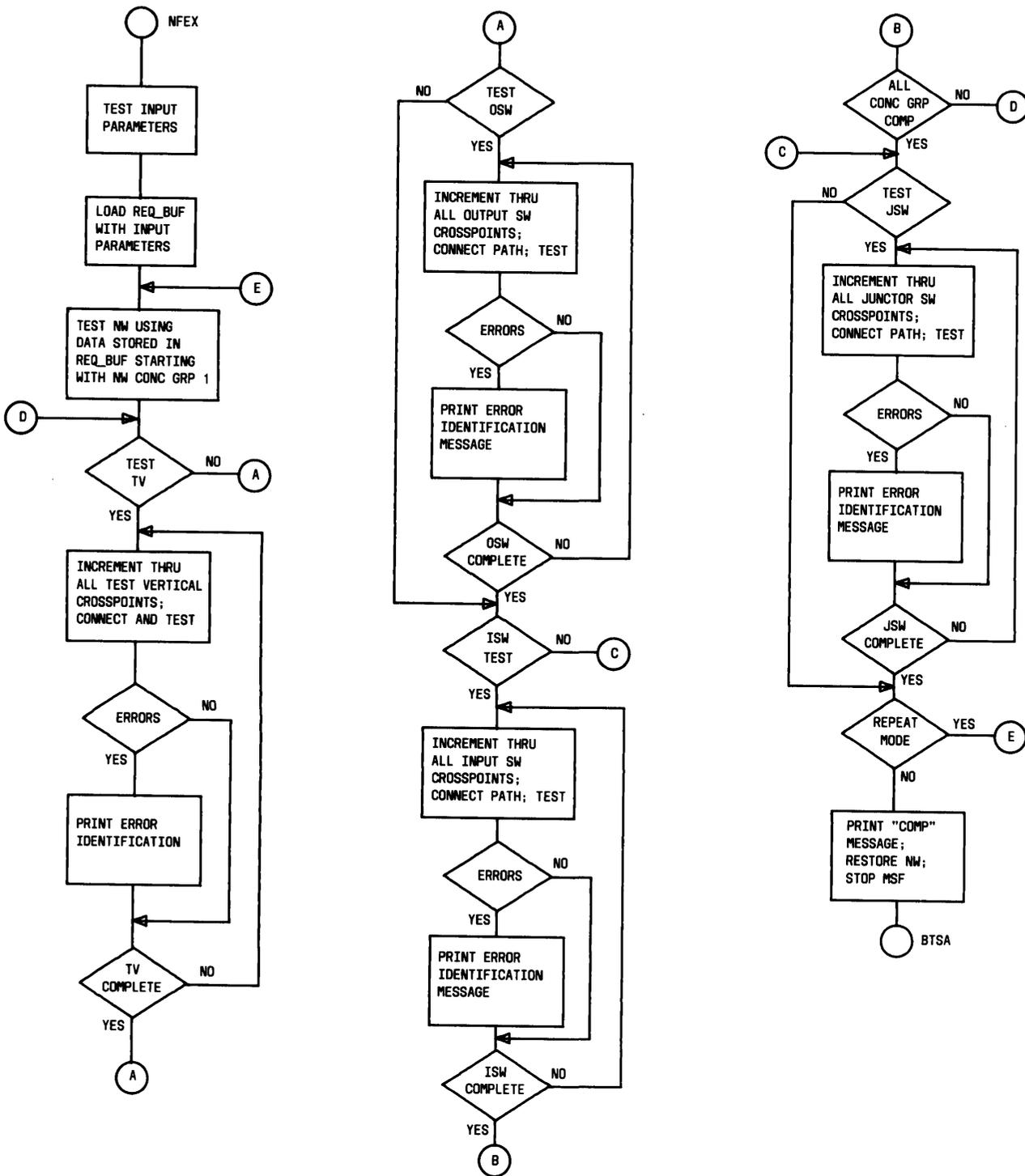


Fig. 3—Network Fabric Exercise Program Flow Diagram

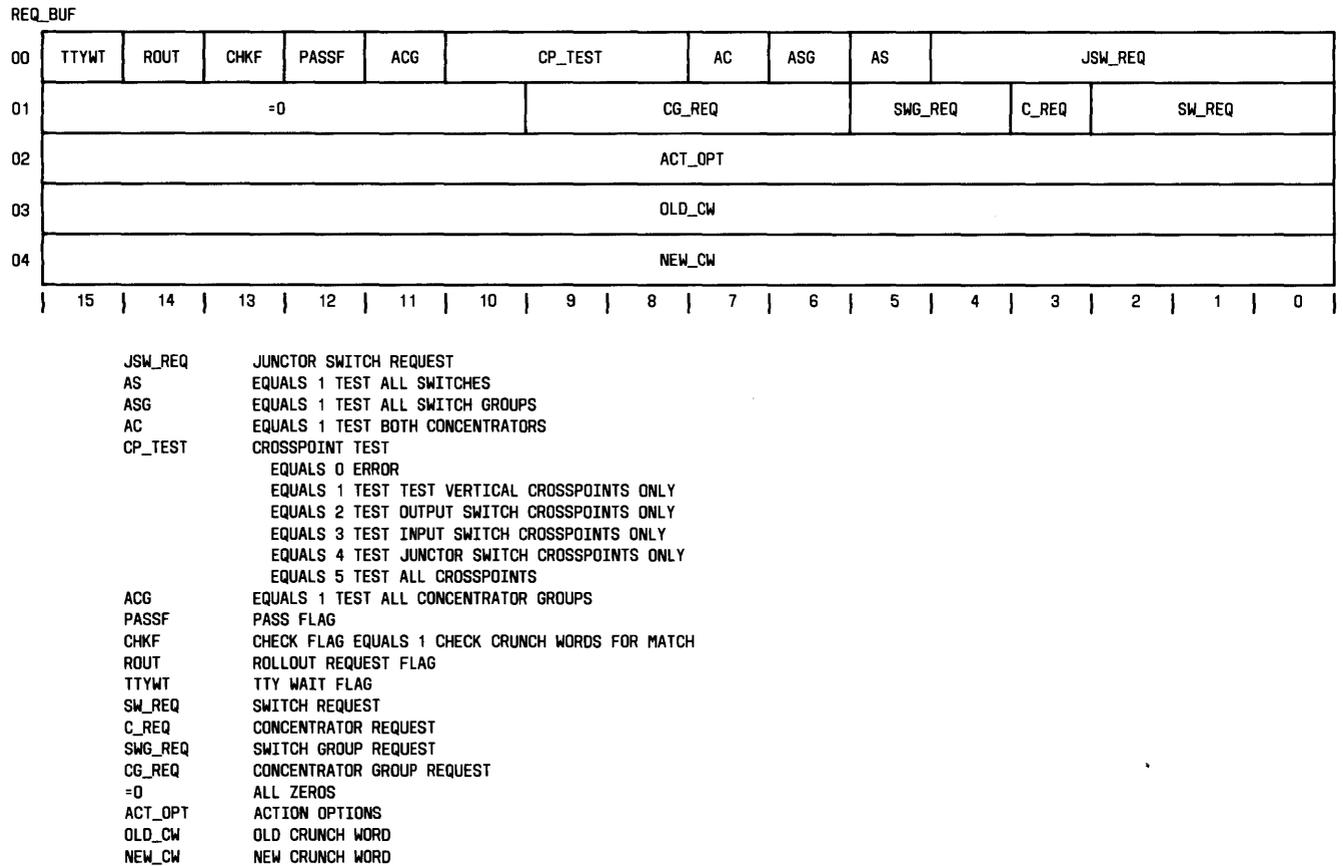


Fig. 4—Request Buffer

2.09 Three types of output messages can be generated from the TTY while performing NFEX. Refer to the output manual (OM-3H300-01) for an explanation of the following output messages and variables.

(a) **Test Complete:** The following output message is used to indicate that NFEX has run to completion.

**EX NW COMP status**  
n1 n2 n3 n4 n4 n6

(b) **Test Abort:** The following output message is used to indicate that NFEX has been aborted.

**EX NW ABT a**  
n1 n2 n3 n4 n5 n6

(c) **Test Failure:** The following output message is used to indicate a failure of the NFEX test. The crosspoint being tested, the type of failure occurring, and the path used are identified.

**EX NW** status switch location failure path information

An example of a failure message from the NFEX program is shown in Fig. 5.

```
40 EX NW FALT TV N1-51-37 NWC
TEN 1 0000 ALNK 1 070 JC 1 29
```

Fig. 5—Example of a Network Fabric Exercise Failure Message

**2.10** The basic approach to network testing is for one crosspoint at a time to be tested in the following sequence:

- (a) Find a nonbusy path from an idle loop-start line or unassigned terminal, through the crosspoint to be tested, to the nearest test vertical.
- (b) Close the path, except the crosspoint under test, and perform a false cross and ground (FCG) test, which will fail if the crosspoint is stuck closed or the path is shorted to battery or ground.
- (c) Close the entire path, and perform a restore verify (RVF) test, which will fail if the crosspoint is stuck open or the path is open or if the crosspoint contains a tip/ring reversal.
- (d) Tear down the path.
- (e) Repeat Steps (a) through (d) for the next crosspoint.

**2.11** Since there are more input switches (first-stage crosspoints) than any other type switch, the method of testing the input switch will be described in the following paragraphs. The other tests are similar.

**2.12** Each input switch (first-stage crosspoints) has eight levels with eight crosspoints on each level. There are eight input switches per switch group, three switch groups per concentrator, and two concentrators per network concentrator group. A representation of the network concentrator groups is shown in Fig. 2. To implement scanning of the crosspoints, the following nested loops are used:

- (a) **Loop 6:** Increments the crosspoint number by one from zero to seven (eight crosspoints) for each entry.
- (b) **Loop 5:** Increments the input level by one from zero to seven (eight levels) for each completion of loop 6.
- (c) **Loop 4:** Increments the switch number by one from zero to seven (eight switches) for each completion of loop 5.

- (d) **Loop 3:** Increments the switch group number by one from zero to two (three switch groups) for each completion of loop 4.

- (e) **Loop 2:** Increments the concentrator number by one from zero to one (two concentrators) for each completion of loop 3.

- (f) **Loop 1:** Increments the network concentrator group by one from one to LOG\_SCAN (the number of network concentrator groups depends on office size and is stored in translation data at the location labeled LOG\_SCAN) for each completion of loop 2.

The program will exit this looping when the network concentrator group is equal to the maximum number in the office. When the loops are completed, an attempt has been made to test each crosspoint of all switches.

**2.13** Similar loops are made to test the test vertical crosspoints, output switch (second-stage) crosspoints, and junctor switch (third-stage) crosspoints. These crosspoints are exercised as shown in Fig. 2 and 3.

**2.14** Each time the loops are incremented to the next crosspoint to be tested, a path through the network must be constructed to include the selected crosspoint.

**2.15** When the path is selected, the connecting information is transferred to routine POINTD in the POINT program. This routine completes the path and returns with a success code or an error code. The error code causes the proper error counter to be incremented. The FAIL\_CHECK routine waits for completion of the test and handles the success or failure code. The failure code branches to TTY\_FAIL\_MSG and prints the path used when the failure occurred. The exercise terminates unless the unconditional, step, or repeat mode is present.

**2.16** The success code will increment the loop to the next crosspoint to be tested, and this sequence continues until all crosspoints have been tested or until an error occurs that terminates the exercise.

**2.17** As the program is executed, NFEX utilizes the following subroutines to perform the task described.

(a) **ISW\_CP\_PATHNT, OSW\_CP\_PATHNT, JSW\_CP\_PATHNT, TV\_CP\_PATHNT:**

These routines use the specified crosspoint to establish a path through the network that must include the selected crosspoint. The selected path is marked busy and placed in the maintenance TCR. At the end of the test, all path components are returned to their previous state (idle or out-of-service). The selected crosspoint is tested even though marked out of service. It is subsequently returned to the out-of-service state when testing is completed.

(b) **FAIL\_CHECK:** This routine waits for the selected path (including the crosspoint to be tested) completion and tests the path. When the test fails, a TTY message is generated. After the test is completed, the path is restored. The return code for this routine is a continue (test passed), an abort (peripheral error), or a failure (FCG or RVF). A failure return terminates NFEX except when the unconditional (UCL) option is selected. When the NFEX option is UCL, the failure prints the failure message and returns a continue code.

(c) **SLOWDOWN:** This routine is called when the network controller queues are busy. It returns when the network controller queues are empty or at the end of four base level loops.

(d) **RESTORE:** This routine restores the selected path to its previous state and clears the maintenance TCR except for the active bit and timer word.

(e) **TTY\_FAIL\_MSG:** This routine is entered when a test failure occurs and prints the identity of the "failed" path and the type of failure.

(f) **TEN\_HNT:** This routine attempts to find an idle loop-start terminal associated with a given A-link.

(g) **TEN\_CONST:** This routine is an alternate for TEN\_HUNT. TEN\_CONST constructs a terminal equipment number (TEN) associated with a given A-link.

(h) **A LNK\_HNT, B LNK\_HNT:** For the given A-link (or B-link), these routines try to select an idle B-link (or A-link) on the same switch.

### 3. GLOSSARY

**3.01** The following terms and definitions are used frequently in this section:

**Algorithm**—A procedure for solving a mathematical problem

**Base Level**—Major software loop including all functions not done during interrupt level

**Loop Start**—A signaling method that uses the metallic loop formed by the line tip and ring leads

**Transient Call Record**—A 16-word block of writable storage assigned to a call in the transient state.

