NEW SYSTEM TEST

(NST)

USER'S GUIDE

Preface

The New System Test (NST) provides a rapid means of testing the processor and exercising peripherals in IBM 4300 Systems.

NST should be considered as an additional resource supplementing existing diagnostic aids, such as the functional diagnostics and EREP.

The "Introduction" assists an IBM CE, or customer, in the normal running of NST. If execution of the tests result in error messages, the results should be analyzed by a CE trained in-depth on the system being tested. The remainder of the User's Guide is for use during analysis and program installation or update.

The remaining sections provide:

- a description of the various tests executing under the NST Control Program (NSTCP).
- a description of the format of commands and options for modifying the tests.
- instructions for the running of PGENXX, PATDAT, EPAGE1, and IODRVR.
- Service Aids gives some additional hints and examples of how to invoke and modify the tests.
- a familiarization with some of the resulting output from the previously-described tests.
- an introduction and description of the UPDATE, UDASDI, and EMUFMT utilities, and their options.
- Appendix A describes the steps necessary for APAR submission.
- Appendix B provides instructions for installation on a customer disk pack.
- Appendix C provides a glossary.
- an Index.

Comments regarding this document should be directed to:

NST Coordinator
Department B44
P.O. Box 390
Poughkeepsie, N.Y. 12602

This document is prepared for Department B44 by Product Publications, Department B97, Poughkeepsie, N.Y., 12602.

Contents

Figure List

Introduction

NST provides a rapid and extensive checkout of 4300 systems. (.5 meg 4331s are excluded - a minimum storage of 1 meg is required.) The processor is tested and attached I/O is operated in a system environment.

This section assists an IBM CE, or customer, in the normal running of NST. If execution of the tests result in error messages, the results should be analyzed by a CE trained on the system being tested. The remainder of the User's Guide is for use during this analysis and program installation or update.

Program	Description
NSTCP	NST control program
PGENXX	Basic processor tests
PATDAT	370-mode paging tests
EPAGE1	VSE-mode paging tests
IODRVR	I/O tests
UPDATE UDASDI EMIFMT	Utility programs

Figure 1. NST Programs

Command Procedures (CPROCs)

The NST tests for 4300 processors are started with command procedures (CPROCs). The following table lists the CPROC IDs, the tests started by each CPROC, and the required processor mode (370 or VSE). For additional information, see the Notes following the table. For CPROC listings, see the next page.

Tests	370 mode			VSE mode			370orVSE	
Run:	370A	370B	370C	VSEA	VSEB	VSEC	COMA	IODR
PGENXX	X	X		X	X		X	
PATDAT	X	X	X					
EPAGE1				X	X	X		
IODRVR	X		X	X		X	X	X

Notes:

1. *Only one CPROC can run at a time. Stop a CPROC before starting another by entering:* STOP,ID-SYSTEM
2. *To test the basic processor and attached I/O on a 1 meg 4331, select CPCOMA (runs in either 370 or VSE mode). 1 meg may not be enough storage to run CP370A or CPVSEA, depending on the features installed and the number of attached I/O devices.*

CPROC Listings

The CPROCs listed here are available for 1 meg 4331 and all 4341 Processors. (See Note 2 on the preceeding page.)

```
CESD CP370A 025 CP370A
*CP CPROC
*CP COM CP370A WILL RUN THESE TESTS
*CP COM PATDAT00 PGENXX00 IODRVR00
*CP STA,ID-PATDAT
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000,NOP-E6,PER-00
*CP STA,ID-IODRVR
*CP TERM
 END

CESD CP370B 025 CP370B
*CP CPROC
*CP COM CP370B WILL RUN THESE TESTS
*CP COM PATDAT00 PGENXX00
*CP STA,ID-PATDAT
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000,NOP-E6,PER-00
*CP TERM
 END

CESD CP370C 025 CP370C
*CP CPROC
*CP COM CP370C WILL RUN THESE TESTS
*CP COM PATDAT00 IODRVR00
*CP STA,ID-PATDAT
*CP STA,ID-IODRVR
*CP TERM
 END

CESD CPCOMA 025 CPCOMA
*CP CPROC
*CP COM CPCOMA WILL RUN THESE TESTS
*CP COM PGENXX00 IODRVR00
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000,NOP-E6,PER-00
*CP STA,ID-IODRVR
*CP TERM
 END
```

```
CESD CPVSEA 025 CPVSEA
*CP CPROC
*CP COM CPVSEA WILL RUN THESE TESTS
*CP COM EPAGE100 PGENXX00 IODRVR00
*CP STA,ID-EPAGE1
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000,NOP-E6,PER-00
*CP STA,ID-IODRVR
*CP TERM
 END

CESD CPVSEB 025 CPVSEB
*CP CPROC
*CP COM CPVSEB WILL RUN THESE TESTS
*CP COM EPAGE100 PGENXX00
*CP STA,ID-EPAGE1
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000,NOP-E6,PER-00
*CP TERM
 END

CESD CPVSEC 025 CPVSEC
*CP CPROC
*CP COM CPVSEC WILL RUN THESE TESTS
*CP COM EPAGE100 IODRVR00
*CP STA,ID-EPAGE1
*CP STA,ID-IODRVR
*CP TERM
 END

CESD CPIODR 025 CPIODR
*CP CPROC
*CP COM CPIODR WILL RUN THESE TESTS
*CP COM IODRVR00
*CP STA,ID-IODRVR
*CP TERM
 END
```

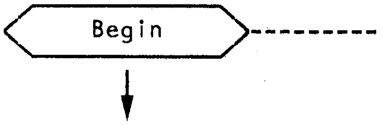
Basic Setup Procedures

- Tapes: Drives should be clean. Mount scratch tapes. Tapes must have tape marks (the NST EXIO command may be used).
- Disk/Drum: Scratch media is required for write testing.
- Reader: Make ready with at least 10 cards (blank or punched). With less than 10 cards, unpredictable results may occur.
- Punch: Ready, with blank cards.
- Printer: UCS buffer must be loaded. Forms control buffer, if applicable, must be loaded. The NST LPB command may be used for both buffers.

For setup procedures for all other devices, see "Detailed Setup Procedures" in this section.

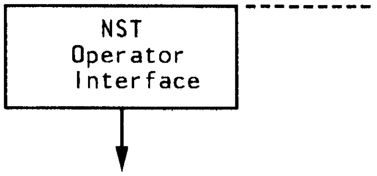
Invoking NST

The following steps are required to invoke NST:



Clear storage. System should be in process mode. Mount NST tape or disk and IPL. System should stop with wait code of 000048.

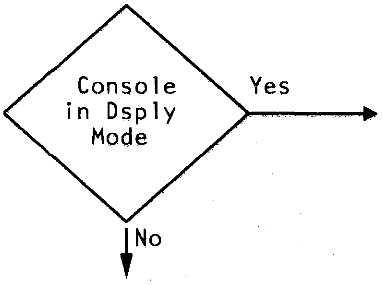
NOTE: The remainder of this section covers console display mode. If your console is in printer/keyboard mode, press REQUEST before entering information.



Press ENTER key and wait for console message:

NST-X.X: OPERATOR INTERFACE ACTIVE

X.X = Release Number



There are two modes of console frame control, scroll stop mode and scroll continuous mode. The default is scroll continuous mode (in this mode, some messages may be lost). Because less operator intervention is required, this mode may be more convenient.

Scroll stop mode is selected by entering: +++

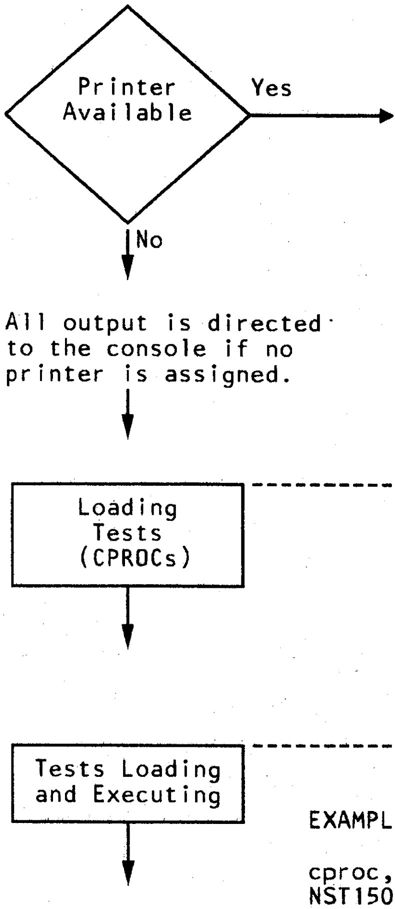
Press ENTER to continue.

Messages continue to be displayed until the +++ leaves the top of the screen and reappears at the bottom as:

+++PRESS ENTER TO CONTINUE

In this way, scroll stop mode saves each frame until it is released by the operator.

To change back to scroll continuous mode, use the space key to blank out the +++ characters when they are at the bottom of the screen. Then press ENTER.


Printer Available
Yes
No
All output is directed to the console if no printer is assigned.
Loading Tests (CPROCs)
Tests Loading and Executing

NOTE:

ALL OPERATOR ENTRIES ARE PRINTED IN THIS TYPE STYLE

Ensure that the printer is ready. Then assign the printer using this format:

AS,S,UA-cuu,DT-tttt

cuu = Printer address
tttt = Printer type

NOTE: The printer must be ready before attempting to assign it.

To begin testing, key in:

CPROC,CP-CPxxxx

(where xxxx is the proper CPROC for the processor; 370 or VSE).
Use either a - or an = when making entries, whichever is more convenient on a particular console.

Observe console messages as a result of previous entry:

EXAMPLE:

```
cproc,cp-cp370A
NST150R-(CP370AFF/$SYCPX) CLOCK=10/25/78-13.14.58.375400
CP370A WILL RUN THESE TESTS
PATDAT00 PGENXX00 IODRVR00
--
 PATDAT00-$SDINT :1000 10/25/78-13.14.59.575400
PATDAT00(09/18/78-00-0102)LOADED AT 00048840, 00477E BYTES
 PGENXX00-$SDINT :1000 10/25/78-13.15.05.407242
PGENXX00(09/12/78-00-0403)LOADED AT 0004CFC0, 005ABE BYTES
 IODRVR00- SDINT :1000 10/25/78-13.15.59.455400
IODRVR00(09/12/78-00-0402)LOADED AT 00004D000, 004770 BYTES
*PGENXX00-PGENXX :1000 SUSPENDED 10/25/78-3.15.25.044121
SUSPENDED AT LOAD TIME
NST15C-(CP370AFF-$SYCPX):R150 CLOCK-13.16.22.09.234567
THE OPERATIONS SPECIFIED IN CP370A HAVE BEEN INITIATED.
```

Note: *If this message is displayed:*

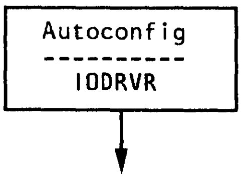
REQUESTED STORAGE NOT AVAILABLE. CREATE AVAILABLE STORAGE AND RESUME OR CANCEL.

Cancel one of the programs that is running and resume the remaining programs.

Example: Assume that PATDAT00, PGENXX00, and IODRVR00 are running. Enter:

```
CANCEL,ID-PATDAT00
RESUME,ID-PGENXX00
RESUME,ID-IODRVR00
```

(To end EPAGE1, enter STOP,ID-EPAGE100*)*



IODRVR starts only when it is included in the selected CPROC. See "Command Procedures (CPROCs)."

IODRVR goes into preconfiguration.

```
IOD508I (IODRVR00/IODDFN)              01/01/80/00.00.20.345678
PRE-CONFIGURATION STARTED
```

A list of devices that are available, both ready and not ready, are displayed on the console under the following heading:

```
TABLE OF I/O ADDRESSES TO BE CONFIGURED ON THIS SYSTEM

0009    000C * 000D    000E    0165    0170    0173
0175    0178   0179    017F    0190    0191    019E

* INDICATES OPERATOR INTERVENTION REQUIRED ON DEVICE
**CAUTION - TAPE OR DASD DEVICES SHOWN AS NOT READY MAY NOT
            EXIST
```

If a device is not to be tested, drop it at this time.

```
ENTER DROP-LCCUU-HCCUU (UP TO 5 UNIT RANGES) OR
DROP-CCUU (UP TO 10 UNITS) OR DROP-NONE TO CONTINUE
```

Key in (for example):

```
MOD-IODRVR00,DROP-00C-00D-180-181-182
```

Press ENTER to continue.
Entering a DROP message results in the table being displayed again with the updated entries.

Notes:

1. *It is not necessary to drop non-existent or not-ready addresses. The configurator automatically drops them.*
2. *If DASD is shared with another processor, the shared DASD is reserved by IODRVR only when a CCW string is executing. All other device types are permanently reserved by IODRVR.*
3. *When a tape drive is shared on two channels, IODRVR configures the address for both channels. However, the tape drive is tested on the lower-address channel only. To test the tape drive on the higher-address channel, drop the lower address using the procedure that follows.*

If a range of addresses is to be dropped from the table, key in (for example):

```
MOD-IODRVR00,DROP-L100-H300
```

Entering this DROP message results in any devices with addresses falling in the range 100 to 300, inclusive, being dropped from the table.

When all devices are made ready or DROPped, key in:

```
MOD-IODRVR00,DROP-NONE
```

Press the ENTER key.

This message is displayed:

```
IOD502I-(IODRVR00-IODCNF  * SUSPENDED 12/31/79/23.59.58.345678
*** WARNING ***** WARNING ***** WARNING ***** WARNING ***

IODRVR WILL DESTROY DATA ON NON FILE PROTECTED TAPES.
LABEL INFORMATION ON CUSTOMER TAPES WILL BE PROTECTED.
ENSURE ALL DRIVES TO BE TESTED HAVE SCRATCH OR BLANK TAPES MOUN

BLANK TAPES (NO RECORDS) MUST BE TAPE MARKED BEFORE CONTINUING.
USE 'EXIO,UA-CCUUU,WTM' TO WRITE A TAPE MARK.

FOR CORRECT AUTO CONFIGURATION ALL OTHER I/O MUST BE HALTED.

*** WARNING ***** WARNING ***** WARNING ***** WARNING ***
ENTER 'CNFG' TO CONTINUE.
```

The only NST I/O program running at this time should be this copy of IODRVR.

Key in:

```
MOD-IODRVR00,CNFG
```

Press ENTER key

The unit availability table is then displayed on the console and printed on the printer, if assigned. For example:

```
 UNIT  UNIT   ADDRESS     TEST
 TYPE  CLASS       CUU    LEVEL  FEATURES
 3278   GD    00000011     2
 2540   RD    0000012C     0
 2540   PU    0000012D     0     COL BIN
 3420   TP    00000180     2     FILE PROTECTED REEL
 3420   TP    00000181     0     3420 MOD3, 9 TRK
 3310   DS    00000240     0
```

To VARY a device to a different testing level, the following is needed:

```
MOD-IODRVR00,VARY-cuu-l
```

(where: cuu is the address of the unit to be VARY'd, and l is the desired test level.)
For example:

```
MOD-IODRVR00,VARY-02C-1
```

This example causes the reader at address 02C to read cards instead of performing a level 0 test.

Note: *Only one device may be varied at a time.*

The following lists summarize the functions performed at the three test levels:

Test Level 0

- DASD - seek, read only.
- Printers - buffer test only.
- Tape - read/write test.
- Display - read/write test.
- TP - wrap only, read/write test.
- Channel-to-channel adapter - no testing.
- All other devices - sense, no-op.

Test Level 1

- DASD - seek, read/write test. Mount CE pack or CE cell. *Level 1 testing destroys format.*
- Printers - print test (buffer must be loaded).
- Reader/Punch - read/write test. Ready with blank cards.
- Channel-to-channel adapter - Tested.
- All other devices - same as test level 0.

Test Level 2

- No testing is performed.

IODRVR Go

Key in:

```
MOD-IODRVR00,GO
```

Press ENTER key.
This causes the command blocks for each device type to be loaded and executed.

Tests are Running

The above console messages complete the IPL phase, and the test are beginning to execute.
If you wish to confirm that test execution is progressing, wait a few minutes and proceed to the next step. This wait period is load device-dependent.

Display Current Results

To check the status of testing, key in:

```
DIS,COUNTR
```

or

```
DIS,COU
```

Press ENTER key.
The following is an example of the resulting console display:

```
PROGRAM   PASS      ERROR     SEED

PGENXX00  00000745  00000000  ED234xxx
PATDAT00  00000054  00000000  55423xxx
IODRVR00  00000043  00000000  4572Axxx
```

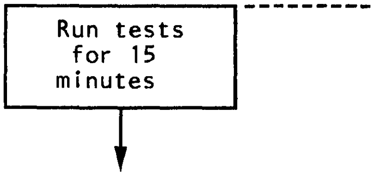
where:

PROGRAM - is the name of the test

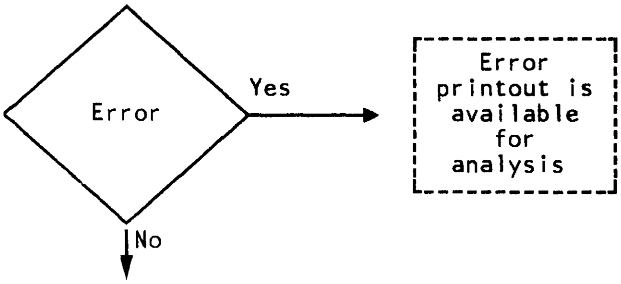
PASS - is the total number of times executed

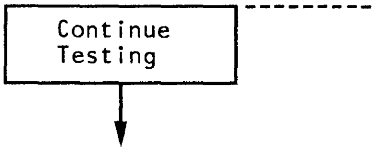
ERROR - is the number of times an error occurred

SEED - represents the particular set random instructions generated and executed



Unless tests stop execution because of an error condition, allow tests to run for at least 15 minutes. The start of this run time is determined when, after keying in "DIS,COUNTR," the pass column of the display program status indicates at least a "00000001" for each testing program.

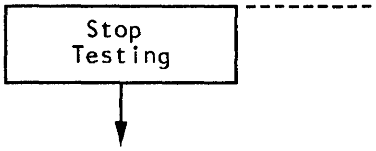




To continue testing after an error is encountered, key in:

RESUME,ID-xxxxxxxx

Press ENTER key.
ID-xxxxxxxx is the name of the test which detected the error and was suspended.
IODRVR does not stop after an error. All other programs must be RESUMEd after an error.



After running test for 15 minutes, key in:

STOP,ID-SYSTEM

Press ENTER key.
All programs are stopped and the run results are displayed.
The output printer lists all opcodes and interrupts tested.

Error Messages

An error message from the test programs is displayed with an asterisk (*) and one of the following test program names:

EPAGE100
PATDAT00
PGENXX00
IODRVR00

An exception to this is a message stating that a program had been suspended at load time. This is a normal message and is the result of specifying the HALT option when invoking a test. The HALT option allows the user to define a variety of test parameters. The message appears as follows:

```
*PGENXX00-PGENXX   :I000 SUSPENDED    06/08/78-12.51.39.334671
SUSPENDED AT LOAD TIME
```

An example of an error message appears below. The details of an error are printed on the assigned printer, or else displayed on the console.

Example:

```
*PGENXX00-PGENXX       :E701 ERROR DETECTED AT INTERRUPT PSW
```

Note: *Messages prefixed with an A, I, or E, (action, information, and error) followed by three digits result from improper selection of various options and various error conditions.*

Wait States

The following wait states may occur when running NST. The address portion of the PSW displays:

000048 Initial IPL wait PSW. Press ENTER key to continue.

DEAD01 Insufficient storage available during IPL initialization.

DEAD28 NST detected a program check within the control program. Take a dump. Contact IBM CE for APAR initiation. Refer to Appendix A for dump procedures.

DEAD39 The system detected no more control block space for I/O-related control blocks.

00DEAD IPL of NST on a system not supported by NST.

DEADxx and any other wait state may indicate an error condition and must be investigated by an IBM CE.

Any other wait state indicates an error condition and must be investigated by IBM personnel.

The following are enable wait states:

D218 - Level 2.0 Interim Dormant PSW
D718 - Level 2.1 Interim Dormant PSW

Note: *Enable wait state will vary from release to release.*

Detailed Setup Procedures

Successful configuration depends on a fully-functional basic processor and console device (such as keyboard/display for input, and hard-copy device for output), and making ready all I/O devices to be tested. To ensure a successful configuration, follow these steps:

1. Tape Drives - be sure that:
   a. Drives are clean, and no pins are in the red area, CE panel, or tape control unit.
   b. Good, 2400-ft reels of tape should be used to avoid false errors and allow complete isolation.

   Caution: Only scratch tapes should be used; IODRVR will write over customer tapes.

2. DASD - check that disks/drums are correctly formatted.

   Caution: Level 1 testing writes on disks. Use scratch disk for testing.

3. UCS Printer - ensure that the buffer is correctly loaded. Folding must be off.
4. Card Reader/Punch - make sure that at least ten cards are in the hoppers, and where multiple reader/punches have the same control unit (2596), all feeds must be ready. Do not set EOF.
5. 3705/3704 - if included in your system, its storage should be cleared and the LOAD key pressed before starting IODRVR. If operating on dual channels, one of the channels must be disabled.
6. It is normal operation to have non-supported devices in off-line mode when running IODRVR.

Channel-to-Channel Adapter

1. When running channel-to-channel testing, one system (slave) must have its channel-to-channel varied to test level 1, and all other I/O varied to level 2 (no testing).
2. The other system (master-test system) must have its channel-to-channel varied to test level 1 also. However, normal testing can proceed on this second processor's I/O.

Non-supported Devices

Non-supported devices should be off-line when running IODRVR.

The remaining sections of this document contain a more detailed description of the tests, and additional information regarding options and other programs.

NST Overview

The purpose of this section is to describe the NST programs. Refer to Figure 1 for the overall program structure.

Chapters following this one provide information on how to invoke individual programs and enter the desired options, and also to familiarize the user with the various types of output.

PGENXX - Basic Processor Test

Caution: The running of PGENxx on a processor with RPQ(s) installed may yield unpredictable results.

PGENXX performs the following steps:

1. Generates a random instruction stream (identified by a SEED number assigned by the program).
2. Simulates execution of the instruction stream.
3. Performs actual execution of the instruction stream.
4. Compares any SVC (Supervisor Call) or program interrupt to the results of the simulation. If an error is detected, the program attempts to isolate the detected error to a failing instruction or minimum combination of instructions that produced the error. Error message output is to the console, and the details of the error analysis are printed on the printer.
5. Continues execution to the end of the instruction stream (unless halted by the operator). When PGENXX is RESUMEd, it generates and executes a new random instruction stream.
6. Up to 14 copies of PGENXX can be run simultaneously. However, one or two copies of PGENXX is the most efficient number of copies to be run, depending on the storage size of the processor. More than two copies significantly increase the program overhead.

   The number of random instruction streams executed in a given period of time increases with the amount of buffer space available to the program. There is a minimum requirement of X'1800' bytes of storage buffer. See "BUFSZ" in "PGENXX: Use of Command Line Entries," for more detail.

PGENXX Test Components
PGENXX performs the following:

- Tests Standard, Decimal, and Floating Point Instructions in a pseudo-random fashion.
- Tests Move Character Long (MVCL) and Compare Character Long (CLCL) instructions with large data counts.
- Tests Monitor Call (MC).
- Exercises the Program Event Recording (PER) feature.
- Exercises the Storage Protection feature.

Functions Tested:
The following functions are either tested or exercised. Testing consists of generating the op code and forcing all possible exceptions. Exercising consists of generating the op code with restrictions; that is, there is no comparison of results, and interrupts are not forced.

Standard Instruction Set -	System control op codes exercised only, all others are tested.

Decimal Instruction Set
Floating-Point Instruction Set
Extended Precision Floating Point Instruction Set
Monitoring
Dynamic Address Translation
PSW Key Handling
Conditional Swapping
Protection
Program Event Recording
Byte-Oriented Operands
Tracing
Program Level

Functions Excluded:
The program does not attempt to verify the following processor functions:

I/O: All Forms
Direct Control
Emulation: All Forms
Timer Functions:
  Interval Timer
  Time of Day Clock
  Clock Comparator
  CPU Timer

PATDAT - Dynamic Address Translation (DAT) Test

Functions Tested

- Load Real Address (LRA)
  Translation specification exception interrupts occur as they do for implicit translation. Page/segment exceptions and valid translations result in condition code settings.
- Test Protection (TPROT)
  The instruction accesses translation tables in DAT mode only. Translation exception interrupts are handled in the same manner as they are for the LRA instruction.
- Invalidate Page Table Entry (IPTE)
  Access exception interrupts for addressing, and protection are possible.

EPAGE1 - Paging Test

EPAGE1 is a program that runs under control of the New System Test control program (NSTCP). EPAGE1 consists of tests that verify VSE mode paging operations on 4300 processors only.

The EPAGE1 tests verify operation of new 4300 control instructions and control instructions that have been changed from 370 mode. The tests change the page status and description bits, in both random and sequential patterns, to check the interaction between paging functions, control functions, and other system operations.

The two types of EPAGE1 tests are:

- Instructions tests
- Function tests

Instruction Tests:
Each EPAGE1 instruction test executes one page-control instruction and checks for correct results. The test page is used as the operand of the instruction, and may be addressable, connected, disconnected, invalid, or page 0.

Function tests:
Each EPAGE1 function test performs a 4300-VSE-mode paging function and checks for correct results. Page 0 is *not* used in any of the function tests.

EPAGE1 tests run in a random sequence and results are predicted. After each test has run, the condition code, general registers, control registers, capacity counters, interrupt condition, page state, and page descriptor bits are checked against the predicted results. An error is recorded when the actual results are not the same as the predicted results. EPAGE1 continues running after recording an error and an error message is displayed.

IODRVR - I/O Exerciser

IODRVR is a program executing under NSTCP; it executes 370 I/O command blocks. All supported I/O attached to the system and in a ready state will be exercised.

The instructions for invoking IODRVR and configuring the I/O for testing are shown next to the block "Autoconfig/IODRVR" in the first chapter. Refer to "IODRVR Program Options" for details about modifying IODRVR.

Utility Programs

UPDATE

UPDATE is a utility program that can be executed under NSTCP, and provides for the following functions:

1. Listing the contents of the NST data set.
2. Copying the NST data set to tape or disk.
3. Changing (modifying) CPROCs.
4. Replacing data or modules in an NST data set.

For details, see "UPDATE - Description and Functions."

UDASDI

UDASDI is used to format a disk pack to allow NST to be copied to it. Always format a disk pack before copying the NST data set to it. For details, see "UDASDI - Description and Function."

EMUFMT

EMUFMT is used to format a 3310 disk for 2314 emulation. A 3310 disk must be formatted for 2314 emulation before the NST data set can be copied to it. After running EMUFMT, run UDASDI to create the VTOC before copying the NST data set to it with UPDATE. For details, see "EMUFMT - Description and Function."

NSTCP Program Control and Modification

After the NST Control Program (NSTCP) is loaded, it is controlled by user commands with parameters. An exception to this was described in "Introduction," where default parameters were selected by invoking a command procedure (CPROC). This chapter describes the operator selected functions in terms of Commands, Keywords, Values, and Options.

Console Scroll Control

There are two modes of console frame control: scroll stop mode and scroll continuous mode. The default is scroll continous mode (in the continuous mode, messages may be lost). Because less operator intervention is required, this mode may be more convenient if a secondary printer is not assigned.

Scroll stop mode can be selected by entering:

```
+++
```

Press ENTER to continue.

Messages continue to be displayed until the +++ leaves the top of the screen and reappears at the bottom as: +++ PRESS ENTER TO CONTINUE. In this way, scroll stop mode saves each frame until it is released by the operator.

To change back to scroll continous mode, use the space key to blank out the +++ characters when they are at the bottom of the screen. Then press ENTER.

Time-of-Day Clock

A single optional keyword may be specified for the NSTCP: TODC.

The TODC keyword provides the ability to enter the current current time-of-day clock value. The value reflects the current date and time, and is included in message output headings. Valid input ranges are shown in parenthesis.

```
TODC-m1-dd-yy-hh-m2-ss
```

Where:

m1 = Month of year (01-12)

dd = Day of month (01-31)

yy = Year (00-99)

hh = Hour of day (00-23)

m2 = Minute of hour (00-59)

ss = Seconds of minute (00-59)

For example, key in:

```
TODC-08-01-79-16-30-00
```

Operating Notes:

1. *All characters contained in the TODC keyword are positional and must be entered exactly as described.*
2. *All characters are entered as two digits; that is, 01, not 1.*
3. *The TODC value is not set until the TODC keyword is entered. At initialization time the TODC value is unpredictable.*
4. *The operator is prompted to enable the set TODC switch when the TODC keyword is entered, to enable the time-of-day clock.*
5. *Messages include a time stamp. The right-most field is the decimal fraction of a second.*

Test Program Controls

Test programs are invoked and controlled by operator commands. The operator passes information to test programs by entering program keywords and program options with the MODIFY command (see section on "Commands."

Keywords, Values, and Options

- A KEYWORD is one to six characters in length. It associates operator input VALUEs with a particular parameter entry in a test program.
- VALUE is the operator input placed in a particular KEYWORD parameter entry for a test program.
- A KEYWORD is always separated from its associated VALUEs by an equal sign (=) or a dash (-).
- An OPTION is a single character string of one to six characters in length for system OPTIONS, and one to four characters in length for any program OPTIONS.
- There are no positional requirements for any KEYWORD or OPTION except TODC.
- KEYWORDs and OPTIONS on a command input line are processed as a group without any sequential or positional meanings, except that the last VALUE entered for any KEYWORD overrides any previous VALUEs for that particular KEYWORD.
- A comma (,) indicates that a KEYWORD or OPTION follows. If a comma is the last non-blank character on a line, it indicates that more parameter input will be entered on the next line.
- Imbedded blanks are not allowed on command input lines unless they are part of a quoted string.
- A not (¬) immediately before any option causes a reset condition of the option. Character values are reset to blanks, and hex values are reset to zeroes.

Delimiters

Delimiters to be used on command input lines with their respective meanings are as follows:

Delimiter	*Name*	*Description*
,	COMMA	Keyword or Option follows.
-	DASH	Separates keywords from values.
=	EQUAL	Separates keywords from values.
;	SEMI-COLON	Ends a command line.
	BLANK	Ends a command line.
(	LEFT PARENTHESIS	Keyword array index follows.
)	RIGHT PARENTHESIS	Ends keyword array index designation.
:	COLON	Separates keyword array indexes.
'	QUOTE	Begins and ends a quoted character. string.

[ ] Brackets indicate optional entries. The brackets are not entered as part of the command statement.

Operator Command Line Format

The general format of the NST operator commands includes its syntax, long and abbreviated (italicized) forms, and options. The following is an example of a command line input:

*EXA*MPL,KEY1-xkey1,OPTION1

Where:

*EXA*MPL - is the name of the NST command to be executed. The operator can use the short form for the command, which is the first two or three characters shown italicized.

KEY1 - is the name of a KEYWORD.

xkey1 - is the VALUE of operator input that is placed in the particular parameter entry associated with KEY1.

OPTION1 - is the name of an OPTION that has a specific predefined value. It is placed in the parameter entry for its associated KEYWORD.

Commands

ASSIGN - NST Secondary Output Device

The ASSIGN command defines a device and its characteristics to NST. The device must be ready *before* it is assigned.

```
ASSIGN,S,UA-cuu,DT-tttttt
```

Where:

S - specifies a secondary device.

cuu - is the unit address of the device ASSIGNed.

tttttt - is a four-to-six character device type.
DEFAULT is the type previously defined for the device address.

Example:

To assign the secondary output to a 1403 printer with an address of 00E, enter:

```
AS,S,UA-00E,DT-1403
```

To revert back to the console with an address of 01F, enter:

```
AS,S,UA-01F,DT-3277
```

Operating Note: *Assigning a device whose address has already been assigned causes any new characteristics to override the previous information.*

The output will go to a printer or to a console, not both. If the secondary output printer gets a print check, the output will be diverted to the console. The following are valid secondary devices:

1403
1404
1443
3203
3211
3213
32771 (12 lines)
32772 (24 lines)
3278 (24 lines)*
3278E (20 lines)*
3284
3286
3287
3288
3289

* Use 3278E for the 4331 primary console (3278-2A), and also for *all* 3278-2A displays attached to the 4341 display/printer adapter. Use 3278 for 3278-2 displays other than the primary console attached to the 4331 display/printer adapter.

CANCEL - Terminate Program Immediately

The CANCEL command causes the immediate termination of a program. It also can cause the immediate termination of output on the system primary or secondary output device. Brackets indicate an optional parameter.

```
        [UA-cuu ]
CANCEL, [       ]
        [ID-xid ]
```

Where:

xid - is the name of the program to be immediately terminated. The name is eight characters in length.

cuu - is the device unit address that is to have output CANCELed.

Examples:

```
CANCEL,ID-PGENXX00
```

Description: Terminate the execution of program PGENXX00 immediately. Release all the resources for program PGENXX00, and release any I/O devices assigned to the program.

```
CANCEL,UA-280
```

Description: Terminate the I/O operation currently executing on unit address 280.

Operating Note: *Do not use CANCEL to halt execution of EPAGE100. The STOP command should be used for EPAGE100. See chapter "EPAGE1 Program Options and Restrictions" for more detail.*

DISPLA and PRINT

The DISPLA and PRINT commands allow the operator to examine the status of the hardware, test programs, I/O resources, and software system. The DISPLA command output is sent to the system primary output device. The PRINT command output is sent to the system secondary output device, and is more detailed than the DISPLA output. Parameters shown in brackets are optional input.

```
DISPLA ⎡,IO(,UA-cuu)  ⎤
       ⎣,COU(,ID-xid) ⎦
```

```
PRINT  ⎡,IO(,UA-cuu)  ⎤
       ⎣,COU(,ID-xid) ⎦
```

Where:

The type of output to be DISPLAyed or PRINTed is indicated by:

IO[,UA-cuu] = program I/O in system
COU[,ID-xid] = program counters

xid - is an eight-character program name about which information is DISPLAyed or PRINTed.
DEFAULT is to use the xid value from the last command line on which xid was entered, or use SYSTEM if xid was not previously entered.

cuu - is the unit address of the I/O device for which information is DISPLAyed or PRINTed.
DEFAULT is all I/O device addresses active in the system.

Example:

The following example may be used to determine the status of all programs which have been started under NSTCP. Key in:

```
DIS,COU
```

This results in program, pass, error, and seed information being displayed at the console.

Operating Notes:

1. *Any DISPLA or PRINT of information causes a suspension of the program during output; that is, the data is not changing as it is being outputted.*
2. *The CANCEL command can be used to terminate any output started by DISPLA or PRINT.*

DISPLA, COUNTR (Output)

```
PROGRAM   PASS      ERROR     SEED
PGENXX00  00000003  00000001  AB1345671
```

Where:

PROGRAM - is an eight-character program name.

PASS - is the total number of times a program has generated a new random seed for test.

ERROR - is the number of errors encountered by the program.

SEED - is the current random seed of the program.

PRINT,COUNTR (Output)

```
PROGRAM  PASS     ERROR    SEED     COUNTER  VALUE    MODE
PGENXX00 00000003 00000001 EFA17892 ----     ----     REAL
PGENXX00 ----     ----     ----     RESTARTS 00000001 ----
```

Where:

PROGRAM - is an eight-character program name.

PASS - is the number of times a program has generated a new random seed for test.

ERROR - is the number of errors encountered by the program.

SEED - is the current random seed of the program.

COUNTER - is the name of a counter internally defined by the program.

VALUE - is the number of times the counter has been incremented.

MODE REAL - TO BE ADDED WHEN AVAILABLE

EXIO - I/O Commands

The EXIO command causes the immediate execution of a specified I/O command to a specified device. All resulting status is reported to the operator. Brackets indicate an optional parameter.

```
EXIO,UA-cuu[,CCW-xccw][,INSTR-xinstr]
     [,LNG-xlng][,DATA-xdata][-xdata-...xdata]
```

Where:

cuu - is the unit address of the device to which the I/O command is executed.

xccw - is the two-digit CCW command code to be executed. A list of options which can be used is found in the Operating Note 1 which follows. DEFAULT is to execute the I/O instruction xinstr, or if INSTR is not specified, to perform a sense to device at address cuu.

xinstr - is the four-digit I/O instruction operation code to be executed. A list of options which can be used is found in Operating Note 2 which follows. DEFAULT is to execute the CCW command xccw, or if CCW is not specified, to perform a sense to device at address cuu.

xlng - is the length of data in hex to be read if xccw is a read or read backward command. DEFAULT is to read 1 byte.

xdata - is the data to be written if xccw is a write command. The data must be hexadecimal (0-F) and must be grouped into 2-byte fields (4 hex digits). DEFAULT is to write 1 byte of zeroes.

Write Tape Mark

The EXIO command can write a tape mark on a tape to prevent a tape from running away during configuration of IODRVR. The command format is:

```
EXIO,UA-cuu,WTM
```

Example:

```
EXIO,UA-180,WTM
```

This writes a tape mark on the tape at address 180.

Block Data Check

The EXIO command can be used to block data checks. This can be done after loading the print buffer. Refer to the following section "LPB - Load Print Buffer."
The command format is:

```
EXIO,UA-cuu,CCW-73
```

Example:

```
EXIO,UA-00E,CCW-73
```

This blocks data checks on the printer at address 00E.

Operating Notes:

1. *The following table lists some of the OPTIONS which can be substituted for CCW-xccw values:*

OPTIONS	*xccw Value*	*Command Description*
BSF	2F	Backspace tape file
BSR	27	Backspace tape record
FSF	3F	Forward space tape file
FSR	37	Forward space tape record
REW	07	Rewind tape
RUN	0F	Rewind and unload tape
SNS	04	Sense device (DEFAULT)
WTM	1F	Write tape mark
NOP	03	No operation
READ	02	Read
WRIT	01	Write

2. *The following table lists some of the OPTIONS which can be substituted for INSTR-xinstr values:*

OPTIONS	*xinstr Value*	*Command Description*
HIO	9E00	Halt I/O
HDV	9E01	Halt device
CIO	9D01	Clear I/O
TIO	9D00	Test I/O
TCH	9F00	Test channel
SIO	9C00	Start I/O
SIOF	9C01	Start I/O Fast Release

3. *A maximum of 100 hex (256 decimal) bytes may be specified to be read on a read or read backward command.*

4. *A maximum of A0 hex (160 decimal) bytes may be specified to be written on a write command.*

5. *The length of data to be written on a write command is determined by the length of xdata input.*

6. *A sense command will display 64 (hex) bytes of data containing sense information.*

LPB - Load Print Buffer

The LPB command loads print and/or forms buffers for a specified device. Parameters shown in brackets are optional input.

```
LPB,UA-cuu,DT-tttttt[,xucb][,FOLD][,xfcb]
```

Where:

cuu - is the unit address of the device to which the print or forms buffer is loaded.

tttttt - is a four- or six-character device type.

xucb - is the name of the print buffer to be loaded, as shown on the next page. DEFAULT is the AN alphanumeric character train/chain (if no xfcb value is entered.

FOLD - indicates that lowercase alphabetic characters are printed as uppercase characters only.

xfcb - is the name of the Forms Control Buffer to be loaded, as shown on the next page. DEFAULT is to not load a forms control.

Example:

```
LPB,UA-00E,DT-1403,AN
```

Description: Load the print buffer for an "AN" alphanumeric chain into the printer at unit address 00E.

```
LPB,UA-00F,DT-3211,STD1
```

Description: Load the forms control buffer into the printer at unit address 00F. This will set up the printer for 8.5-inch paper at 6 lines per inch.

Operating Notes:

1. *If no print or forms control buffer is entered, the AN standard alphanumeric AN print buffer is loaded as a default.*
2. *Use the EXIO command to block data checks if required (see section "Block Data Check" for more detail).*
3. *The device type entry for a 3203 must include the model number. For example, the device type entry is DT-320305 for a 3203 Model 5.*

Print and Forms Buffer Names

Print Buffer Name	*Train/Chain Definition*
1403/3203 Printers:	
AN	Standard alphanumeric
GN	Standard ASCII
HN	Standard scientific
PCAN	Preferred alphanumeric AN
PCHN	Preferred alphanumeric HN
PN	PL1
QN	PL1
QNC	PL1
RN	Fortran-Cobol
SN	Standard test printing
TN	Scientific text printing
XN	Extended text printing
YN	High-speed alphanumeric
3211 Printers:	
A11	Standard alphanumeric
G11	Standard ASCII
H11	Standard scientific
P11	PL1
T11	Scientific text printing
3262 Printers:	
A62	Standard 48 character - US EBCDIC
B62	Standard 64 character - US EBCDIC ASCII
C62	Universal 63 character - US EBCDIC ASCII
D62	Standard 96 character - US EBCDIC ASCII
E62	52 character - Austria/Germany
F62	116 character - French/Canada
G62	128 character - Katakana
3289 Printers:	
A89	116 character - French/Canada
B89	128 character - Katakana
C89	Universal 48 character - US EBCDIC
D89	Universal 64 character - US EBCDIC
E89	Universal 96 character - US EBCDIC

Forms Buffer Name	*Forms Control Definition*
HIPO	22 inch paper, 8 lines per inch
STD1	8.5 inch paper, 6 lines per inch
STD2	11 inch paper, 6 lines per inch
STD3	8.5 inch paper, 8 lines per inch
STD4	11 inch paper, 8 lines per inch
STD5	9 inch paper, 6 lines per inch
STD6	12 inch paper, 6 lines per inch
STD7	9 inch paper, 8 lines per inch
STD8	12 inch paper, 8 lines per inch

MODIFY - Pass Information to Program

The MODIFY command allows the operator to pass parameter information to a program. It is the only means of passing information to a program. The KEYWORDs and OPTIONS are determined by the program to be modified. The specific keywords and options are listed with the particular program.

```
MODIFY-xid,[KEY-xkey],[OPTION],[SEED-xseed]
          ,[LOOP-xloop],[RESET-xreset]
```

Where:

xid - is the eight-character name of the program to which information is to be passed. The xid field must be eight characters in length.

KEY - is a KEYWORD defined in the parameter entry table of program xid.

xkey - is the VALUE to be associated with program keyword KEY. Any KEYWORD defined in the program is valid input, and each MODIFY line can have many program (xid) KEYWORDs on it.

OPTION -is an OPTION value defined in the program (xid). Any OPTION defined in the program is valid input, and each MODIFY line can have many program OPTIONS on it.

xseed - is the value of the random base seed for program xid, one to eight hex digits in length. DEFAULT is to allow internal algorithms to generate the next base seed.

xloop - is one to four hex digit numbers that control the number of pseudo random base seeds generated in a repeating cycle. The same seeds are repeatedly generated. Use with "SEED" to repeat a failure. The value 1 repeats only that seed. A value of 2 or more adds random seeds to the "SEED" until that number is reached, and then the program loops on that group. DEFAULT is zero. This indicates no repeated seed-generated cycles.

xreset - is a one- to six-character keyword that has its current value reset to its initial value. DEFAULT is not to reset the keyword value.

Operating Notes:

1. *The parameters are checked by the system for syntax errors only. The operator is prompted to re-enter any parameters in error, and any following parameters on that input line.*
2. *Multiple MODIFY commands with the same xid are permitted. However, if a keyword is used more than once, only the last value of the keyword is passed to the program.*
3. *A MODIFY with no parameters indicates parameter processing completion, and allows the program to continue.*
4. *The three keywords common to all programs are RESET, SEED, and LOOP.*

Examples:

```
MODIFY-PGENXX00,BUFSZ-3000,TR
```

Description: Modify the value of keyword BUFSZ in program PGENXX00 to the value 3000. In addition, place the value for OPTION TR into its associated keyword in program PGENXX00.

```
MODIFY-PGENXX00,RESET-TRACE
```

Description: Reset the value of keyword TRACE in program PGENXX00 to its initial value.

RESUME - Continue Program Execution

The RESUME command removes a program from a wait state and allows it to continue execution. Brackets indicate an optional parameter.

*RES*UME [,ID-xid]

Where:

xid - is the name of the program(s) to be RESUMEd. DEFAULT is to use the xid value from the last command line on which xid was entered, or use SYSTEM if xid was not previously entered.

Operating Note:

1. *If xid is eight characters in length, program xid is RESUMEd. No other programs are resumed.*
2. *If xid is six characters in length, all programs that have xid values which match the first six characters of xid are RESUMEd.*
3. *If ID-SYSTEM is specified, all programs that were SUSPENded with ID-SYSTEM are RESUMEd. Any programs SUSPENded with xid not equal to SYSTEM are not resumed.*
4. *A program that is in a wait state for I/O completion is RESUMEd with a return code from the I/O operation indicating that the device is still busy.*

Examples:

```
RESUME,ID-PGENXX00
```

Description: Program PGENXX00's execution is resumed from a wait state.

```
RESUME,ID-SYSTEM
```

Description: Any programs suspended with a SUSPEND,ID-SYSTEM are resumed.

```
RESUME,ID-PGENXX
```

Description: Any programs halted in the system with the first six characters PGENXX are resumed.

START - Load and Begin Programs

The START command loads a module into the system, or loads and begins a program. Brackets indicate an optional parameter.

```
START ,ID–xid ,[HALT][,REP–xrep]
```

Where:

xid - is the name of a program to be loaded and begun. The name can be six or eight characters in length, where the last two characters are the program's copy identifier. If six characters are input, the control program will add a copy identifier to the name.

HALT - indicates that the program is to be loaded but not begun.

xrep - is the number of copies of program xid in hex which are to be STARTed. The maximum number of copies in hex is 'E' for PGENXX. *Only a single copy of all other tests should be run.*

Operating Notes:

The copy identifiers of multiple copies of programs STARTed with the REP parameter are determined by:

1. *The available copy identifiers in the system for program xid if the last two characters of xid are not entered.*
2. *The copy identifier entered in the last two characters of xid. Each additional copy of xid has a copy identifier one greater than the previous copy. If any of the copy identifiers conflict with a copy identifier already in the system, an error occurs, and that copy of the program is not STARTed.*

Examples:

```
START,ID-PGENXX
```

Description: Load and begin test program PGENXX. The program has a two-character copy identifier appended to it. For example, if PGENXX is STARTed for the first time, the program id is PGENXX00. The second START of PGENXX results in a program id of PGENXX01.

```
START,ID-PGENXX,HALT
```

Description: Load program PGENXX and do not start execution. A MODIFY or RESUME to this program starts its execution.

```
START,ID-PGENXX,REP-2
```

Description: Load and begin two copies of program PGENXX. The copy identifiers for the two copies of PGENXX are determined by the copies of PGENXX already running in the system. For example, if PGENXX00 is running in the system, then the program ids of the two copies of PGENXX are PGENXX01 and PGENXX02.

STOP - Terminate Programs

The STOP command notifies the test program to terminate at the end of its current pass. Brackets indicate an optional parameter.

```
STOP [,ID-xid]
```

Where:

xid - is the eight-character name of the program to be STOPped, or use SYSTEM if all programs in the system are to be STOPped. DEFAULT is to use the xid value from the last command line on which xid was entered, or use SYSTEM if xid was not previously entered.

Operating Note: *If program xid continues to execute after a STOP is issued (which indicates a program loop or hang), the operator can issue a CANCEL to terminate the program.*

Examples:

```
STOP,ID-PGENXX00
```

Description: Program PGENXX00 terminates when the program reaches the end of the current pass. All resources assigned to that program are released.

```
STOP,ID-SYSTEM
```

Description: All programs currently running in the system are terminated as each program reaches the end of its current pass. All resources and I/O devices assigned to each program are released as the program is terminated.

SUSPEN - Temporarily Halt Program Execution

The SUSPEN command halts the execution of a program. Brackets indicate an optional parameter.

```
SUSPEN [,ID–xid]
```

Where:

xid - is the name of the program(s) to be SUSPENded. DEFAULT is to use xid value from the last command line on which xid was entered, or use SYSTEM if xid was not previously entered.

Operating Notes:

1. *If xid is eight characters in length, program xid is SUSPENded. No other program is suspended.*
2. *If xid is six characters in length, all programs that have xid values matching the first six characters of xid are SUSPENded.*
3. *If ID–SYSTEM is specified, all programs in the system are SUSPENded.*
4. *A program that is suspended may be restarted using the RESUME command.*

Examples:

```
SUSPEN,ID-PGENXX00
```

Description: Program PGENXX00's execution is halted.

```
SUSPEN,ID-SYSTEM
```

Description: Halt all programs in the system. Any program already halted remains halted; however, the status of the programs halted by a SUSPEND, ID-SYSTEM will differ from programs already halted because of the general suspend bit in its STATUS field.

```
SUSPEN,ID-PGENXX
```

Description: Any programs running in the system with the first six characters PGENXX are halted.

PGENXX Program Control and Modification

Detailed PGENXX program keyword and control option information follows. For a summary, see "PGENXX Easy Reference Table."

Use of Command Line Entries

Caution:

1. **If PGENXX is started standalone (not from a CPROC), MODify PGENXX after starting, by entering:**

   ```
   STA,ID-PGENXX,HALT
   MOD-PGENXX00,NOP-E6
   ```

2. **The running of PGENXX, whether from a CPROC or invoked standalone, on a processor with RPQ(s) installed, may yield unpredictable results.**

Program Keywords

Operator input to the program is defined in the form of keyword parameters and associated values. These parameters are provided to the program in any of the following formats:

Format:

```
KEYWORD-VALUE
KEYWORD-VALUE-VALUE
OPTION
OPTION,OPTION
```

KEYWORD: is a mnemonic defined for specific program services.

VALUE: is a value associated with a keyword.

OPTION: is the symbolic value defined for program input. Refer to MODIFY for operator input definition.
To cancel the effect of an option for the keywords CLASS, STATE, and INTR, enter an ¬ before the option.

Example:

```
MODIFY-PGENXX00,¬EX
```

This will stop the interrupt testing for execute exceptions.

The following is a description of currently defined keywords and associated values:

BUFSZ - Buffer Size

Defines the number of hex bytes of storage to be used for program workspace.

KEYWORD:

BUFSZ

OPTION:

None

VALUE:

Any hex number that is equal to or greater than X'1800' and less than the amount of available storage left in the system. For additional information, see note 1.

DEFAULT VALUE:

X'F800'

Notes:

1. *If the requested size is not available, the program requests whatever is available and indicates the result to the operator. If the amount of storage available does not meet a minimum requirement of X'1800' bytes, the program is suspended. The user should note that the number of streams executed per unit of time is inversely proportional to the work space size.*
2. *The program does not use the protection feature with storage allocations that are less than X'3000' bytes. Protection is included with storage allocations greater than X'3000' bytes. If extensive testing of move character long (MVCL) or compare logical long (CLCL) is required, increase the BUFSZ above X'F800'.*

Example:

```
MODIFY-PGENXX00,BUFSZ-8000
```

Specifies the buffer storage area to be allocated.

CLASS - Instruction Classes

The following table defines the available instruction classes that are used in the generation of instruction streams.

OPTION	*DESCRIPTION*
STN	Standard instruction set
FLT	Floating point instruction set
DEC	Decimal instruction set
STP	Instructions associated with storage protect
XFL	Extended floating point instruction set
DAT	Instructions associated with dynamic address translation
MON	Instructions associated with monitor call

DEFAULT VALUE:
If no option (OPTION) or combination of options is selected, a default value is assigned. This value consists of all instruction classes supported by the system.

Notes:

1. *Classes selected are ORed with any previously selected classes.*
2. *Each program pass uses a random selection of the specified or default instruction classes.*
3. *A reset causes the default value to be assigned.*
4. *The monitor call instruction is generated only if control over at least one monitor class is available from the control program.*

Examples:

```
MODIFY-PGENXX00,¬FLT,¬MON
```

Specifies that floating point instructions and monitor call instructions are not to be tested.

```
MODIFY-PGENXX00,RESET-CLASS
```

Specifies that CLASS is to be set to the default value.

INTR - Interrupts

The following table defines those interrupts which are forced or allowed in the generated instruction stream:

OPTION	*DESCRIPTION*
ADR	Addressing Exception
DCD	Decimal Divide Exception
DCO	Decimal Overflow Exception
DE	Data Exception
EX	Execute Exception
EXO	Exponent Overflow Exception
EXU	Exponent Underflow Exception
FPD	Floating Point Divide Exception
FXD	Fixed Point Divide Exception
FXO	Fixed Point Overflow Exception
IPER	Program Event Recording Exception
MEE	Monitor Event Exception
OPCK	Operation Exception
PRT	Protection Exception
PRV	Privileged Operation Exception
SIG	Significance Exception
SPC	Specification Exception
STD	Special Operation Exception
SVC	Supervisor Call Interrupts

DEFAULT VALUE:
If no option (OPTION) or combination of options is selected, a default value is assigned. This value consists of all possible interrupts.

Notes:

1. *Interrupts selected are ORed with any previously selected interrupts.*
2. *Each program pass does a random selection of the specified or default interrupts.*
3. *A reset causes the default value to be assigned.*
4. *The supervisor call instruction forces interrupts at known intervals in the instruction stream. The SVC codes generated are those allowed by the control program. If SVC codes are not available, the program suppresses SVC interrupt generation.*
5. *Privileged operation exceptions are suppressed if the random stream is executed in the privileged state.*
6. *Protection exceptions are suppressed when the program executes with a buffer size that is less than X'3000' bytes in length.*
7. *Addressing exceptions are suppressed if the program's address space does not permit that exception.*
8. *Fixed point overflow, decimal overflow, significance, and exponent underflow are controlled by the program mask.*
9. *Fixed point divide, decimal divide, exponent overflow, and floating point divide exceptions are not controllable in a random environment. If any of these exceptions are suppressed, the instruction that causes the exception is not generated.*
10. *If monitor event exceptions are suppressed, the program generates monitor call instructions with unmasked monitor classes. If no such class exists, the monitor call instruction is not generated.*
11. *If PER exceptions are suppressed and PER state is desired, the program executes the random instruction stream with the PER event mask reset.*

Examples:

```
MODIFY-PGENXX00,EX,EXO
```

Execute and Exponent Overflow Exceptions are ORed with any previous value.

```
MODIFY-PGENXX00,¬EX,¬EXO
```

Specifies no Execute and no Exponent Overflow Exception.

```
MODIFY-PGENXX00,RESET-INTR
```

Specifies INTR to be set to the default value.

LOOP - Repeat a Seed

When entered with a specific SEED, this repeats that set of of instructions.

Examples:

```
MODIFY-PGENXX00,SEED-45FE00B2,LOOP-1
```

This starts with this seed and continues to generate the same stream.

```
MODIFY-PGENXX00,SEED-45FE00B2,LOOP-3
```

This starts with this seed and two other random seeds, and continues looping on those three seeds.

```
MODIFY-PGENXX00,RESET-LOOP
```

This resets the loop value and random generation of seeds continues.

N - Number Instructions/Stream

Defines the number of instructions generated for each instruction stream.

KEYWORD:

N

OPTION:

None

VALUE:

Any hex number in the range X'000' to X'FFF'. If zero is specified, the minimum number is assumed to be one.

DEFAULT VALUE:

A pseudo randomly selected number in the specified range.

The program attempts to build the user-selected or randomly-selected number of instructions. But, if the program encounters space problems, it only builds the number of instructions necessary to fill the allotted space.

Examples:

```
MODIFY-PGENXX00,N-50
```

Specifies generation of 80 instructions (X'50') in each stream.

```
MODIFY-PGENXX00,RESET-N
```

Specifies N be set to a default value that is a random number of instructions in each stream.

NIS - Number of Instructions Between Interrupts

Defines the number of instructions to be generated between interrupts. If the selected number of instructions is generated without a forced interrupt, the program generates an SVC instruction.

KEYWORD:

NIS

OPTION:

None

VALUE:

Any hex number in the range of X'01' - X'FF'.

DEFAULT VALUE:

Random number between X'01' - X'20'.

Note: *The supervisor call instruction is used to force interrupts at known intervals in the instruction stream. The SVC codes generated are those allowed by the control program. If no SVC codes are available, the program suppresses SVC interrupt generation.*

Examples:

```
MODIFY-PGENXX00,NIS-2
```

Specifies generation of 2 instructions between interrupts in the stream.

```
MODIFY-PGENXX00,RESET-NIS
```

Specifies NIS to be set to default value which is a random number of instructions between interrupts.

Note: *The value entered for NIS will be the average number of instructions generated between interrupts in the stream over a long period of time. There will be instances in instruction stream generation where interrupts will not be generated exactly after the number of instructions specified by NIS.*

NOP - Not Generate

Indicates that the listed operation codes are not to be generated (not tested).

KEYWORD:

NOP

OPTION:

None

VALUE:

Any hex operation code.

DEFAULT VALUE:

None

Notes:

1. *Only single-byte or two-byte extended op codes, which are legal and valid, may be entered. The extended op codes 'B2xx' and 'E5xx' (where xx = the second byte of the op code) may be entered. A maximum of 40 op codes are allowed. If more than 40 op codes are entered, be aware that the 41st op code replaces the 1st op code, with no notification to the user.*
2. *If the same op code is found in the OP and NOP parameters, the NOP parameter overrides.*

Examples:

```
MODIFY-PGENXX00,NOP-44-5A
```

Specifies the op codes which should not be generated (not tested).

```
MODIFY-PGENXX00,RESET-NOP
```

Specifies that No Operation Code Array is reset. Random generation of op codes is resumed.

OP - Generate

Indicates that *only* the operation codes listed are to be generated.

KEYWORD:

OP

OPTION:

None

VALUE:

Any hex operation code.

DEFAULT VALUE:

All operation codes are generated.

Note: *Only single-byte or two-byte extended op codes may be entered. The extended op codes 'B2xx' and 'E5xx' (where xx = the second byte of the op code) may be entered. The minimum number of op codes allowed is defined as the minimum combination of op codes which will build a random instruction stream. In most cases, the minimum number of op codes is three. A maximum of 40 op codes are allowed. Illegal or invalid operation codes are not used.*

Examples:

```
MODIFY-PGENXX00,OP-44-0E-0F
```

Specifies instruction operation codes to be generated.

```
MODIFY-PGENXX00,RESET-OP
```

Specifies that the Operation Code Array is reset. Random generation of op codes is resumed.

PER - Program Event Recording

The following table defines which events are monitored when the PER option of the STATE parameter is selected.

KEYWORD:

PER *OPTION*	*VALUE*	*DESCRIPTION*
BR	80	Monitor on successful execution of a branch instruction.
IF	40	Monitor on fetching of an instruction from main storage.
ST	20	Monitor on alteration of contents of main storage.
GR	10	Monitor on alteration of contents of general purpose register.

Input for the PER option is either in the form OPTION or PER-VALUE. For example: MOD-PGENXX00,BR and MOD-PGENXX00,PER-80 result in the same input to the PGENXX00 program. Any input of the form PER-OPTION (e.g. PER-BR) is invalid and will not be accepted.

DEFAULT VALUE:

If no option (OPTION) or combination of options is selected, a default value is assigned. This value consists of all possible events.

Notes:

1. *PER events selected are ORed with any previously selected event.*
2. *Each program pass uses a random selection of the specified or default PER events.*
3. *The reset value is the default value.*
4. *If PER state is not set or if PER state is set and PER exceptions are not requested, this parameter is ignored. In the latter case, the random stream executes with this PER mask reset.*
5. *The PER start address, PER end address and PER GPR mask are program-selected.*

Examples:

```
MODIFY-PGENXX00,ST
```

Specifies PER Storage Alteration Event Recording is ORed with any previous value.

```
MODIFY-PGENXX00,PER-00
```

Specifies that no PER events are to be monitored.

```
MODIFY-PGENXX,PER-FO
```

Specifies that all possible events are to be tested.

SEED - Current Seed

The value of the random-based seed for PGENXX, one to eight hex digits in length. Specifying a particular seed number permits the re-execution of a particular instruction stream on the same system, if no configuration or feature changes have been made.

KEYWORD:

SEED

OPTION:

None

VALUE:

Any eight-digit current seed.

DEFAULT VALUE:

Random SEEDS are generated.

Example:

```
MODIFY-PGENXX00,SEED-45FE00B3
```

This starts with this seed and then continues with random seeds.

Note: *Random-based seeds must always be odd. Therefore, if an even random-based seed is entered, the value will be incremented by one to make it odd.*

STATE - Execution States

The following table defines the states or facilities used in the execution of the random instruction stream:

OPTION	*DESCRIPTION*
SPRV	Execution includes privileged state
SPER	Program event recording facility
SBOO	Byte-oriented operands

DEFAULT VALUE:

If no option (OPTION) or combination of options is selected, a default value is assigned. This value consists of all execution states supported by the system and the implemented diagnostic interface.

Notes:

1. *States selected are ORed with any previously selected states.*
2. *Each program pass uses a random selection of the specified or default execution states.*
3. *A reset causes the default value to be assigned.*
4. *The selection of privileged state implies the suppression of privileged exception.*
5. *If PER state is selected and PER exceptions are suppressed, the random instruction stream executes in PER mode with the PER Event Mask reset.*
6. *The byte-oriented operand symbval (SBOO) is used only to determine whether or not to generate operands off their preferred boundaries. The program predicts specification exceptions for off-boundary operands based on whether the feature is on the system.*
7. *Privileged and PER states are reset if control over these states can not be obtained.*
8. *The program does not attempt to attain basic control, extended control, or dynamic address translation. The test program executes in whatever mode it is loaded into.*

Examples:

```
MODIFY-PGENXX00,SPER
```

Specifies PER is to be ORed with any previous value.

```
MODIFY-PGENXX00,PER-00
```

Specifies that no PER events are to be monitored.

PGENXX - Program Control Options

A series of parameters are provided for additional operator control of program operation and output. These are entered with the MODIFY Command.

Examples:

```
MODIFY-PGENXX00,NEH
```

Do not halt on error.

```
MODIFY-PGENXX00,BUFSZ-3000,SEED-54326789,LOOP-1
```

Loop on the selected seed with a buffer size of X'3000'.

```
MODIFY-PGENXX00,BUFSZ-3000,SEED-CDA01235,LOOP-1,NEH,NEP,QLP
```

Loop on the selected seed with a buffer size of X'3000'. Do not halt on error, do not print on error, go into the quick loop function after completion of isolation.

BSC - Mode of Operation

Defines whether or not the Move Long (MVCL) and Compare Long (CLCL) instructions are to be used in the manipulation of data by the program. This option is intended to enhance the program's performance.

OPTION	*VALUE*	*DESCRIPTION*
BSC	Y	Manipulate data using Move Character and Compare Character.
NBSC	N	Manipulate data using Move Long and Compare Long.

DEFAULT VALUE:

NBSC

CNT - Counter Display

Indicates that the op code and SVC program interrupt counters are to be displayed.

OPTION	*DESCRIPTION*
CNT	Display counters.

DEFAULT VALUE:

No display of counters.

Notes:

1. *This parameter causes a display of the counters and is then reset by the program. The parameter must be selected for each subsequent counter display.*
2. *The count represents the number of times an op code is generated or a program interrupt is anticipated, as predicted by the simulator modules of the test program.*

3. No warning is provided if the count exceeds the maximum size. The counters wrap and continue as normal.

DBG - Debug Mode

Provides the system operator with an indication that a serious or unusual situation has occurred in the execution of the test program.

OPTION	*VALUE*	*DESCRIPTION*
DBG	Y	Provide program execution warning message.
NDBG	N	Do not provide program execution warning messages.

DEFAULT VALUE:

NDBG

EH - Halt On Error

Defines whether or not the program is to be suspended on a detected error.

OPTION	*DESCRIPTION*
EH	Suspend the program on error.
NEH	Do not suspend the program on error.

DEFAULT VALUE:

EH

EP - Print

Defines whether or not an error printout is desired.

OPTION	*DESCRIPTION*
EP	Print on detected error.
NEP	Do not print on detected error.

DEFAULT VALUE:

EP

HST - Program History

Defines whether or not a display of the program history is desired with any error output.

OPTION	*DESCRIPTION*
HST	Print program history.
NHST	Do not print program history.

DEFAULT VALUE:

NHST

ISO - Isolate

Defines whether or not an attempt is to be made to isolate on the instruction or combination of instructions which caused an error.

OPTION	*DESCRIPTION*
ISO	Isolate on error.
NISO	Do not isolate on an error.

DEFAULT VALUE:

ISO

Note: *This parameter must be indicated before an error is detected.*

PDC - Print Decimal Instruction Data Area

Defines whether or not the data area referenced by the decimal instruction set is to be printed with any error output.

OPTION	*DESCRIPTION*
PDC	Print the decimal instruction data area.
NPDC	Do not print the decimal instruction data area.

DEFAULT VALUE:

PDC

If an error is detected in a data area where printing is suppressed, that area *is printed.*

PFL - Print Floating Point Instruction Data Area

Defines whether or not the data area referenced by the floating point instruction set is to be printed with any error output.

OPTION	*DESCRIPTION*
PFL	Print the floating point instruction data area.
NPFL	Do not print the floating point instruction data area.

DEFAULT VALUE:

PFL

If an error is detected in a data area where printing is suppressed, that area *is printed* (even though printing had been suppressed).

PFS - Print Instruction Stream

Defines whether or not the entire instruction stream or that portion from the last good interrupt comparison to the error interrupt comparison is to be printed with any error output.

OPTION	*DESCRIPTION*
PFS	Print the instruction stream entirely.
NPFS	Do not print the entire instruction stream.

DEFAULT VALUE:

NPFS

Note: *In the event of a program detected error while the above default is in effect, only that part of the instruction stream from the last comparison point to the error comparison point is displayed. For restart, trace, and trace on interrupt, this parameter has no effect.*

PST - Standard Data Area Dump

Defines whether or not the standard instruction data area associated with the random stream is to be printed.

OPTION	*DESCRIPTION*
PST	Print standard instruction data area.
NPST	Do not print the standard instruction data area.

DEFAULT VALUE:

PST

If an error is detected to be in a data area where printing is suppressed, that area *is printed* (even though printing had been suppressed).

PSW - Print Stream Work Data Area

Defines whether or not the data work area referenced by the standard instruction set is to be printed with an error output.

OPTION	*DESCRIPTION*
PSW	Print Stream Work Area
NPSW	Do not Print Stream Work Area

DEFAULT VALUE:

PSW

QLP - Quick Loop

Defines whether or not an attempt is made to Quick Loop on the instructions or combination of instructions generated by isolation.

OPTION	*DESCRIPTION*
QLP	Quick Loop on isolated stream
NQLP	Do not Quick Loop on isolated stream

DEFAULT VALUE:

NQLP

Example:

```
MODIFY-PGENXX00,SEED-CDA01234,LOOP-1,NEH,NEP,QLP
```

Loop on the selected seed. Do not halt on error, do not print on error, go into the quick loop function after completion of isolation.

Address Sync:

If the program is made to do a Quick Loop on an isolated stream, the address within this stream as listed with the error may be used to trigger the Address Sync Probe.

TR - Trace

Defines whether or not a TRACE is desired after instruction stream execution.

OPTION	*DESCRIPTION*
TR	Print TRACE after stream execution.
NTR	Do not print TRACE after stream execution.

DEFAULT VALUE:

NTR

TRI - Trace on Interrupt Comparison

Defines whether or not a TRACE is desired after an interrupt comparison.

OPTION	*DESCRIPTION*
TRI	Print TRACE after interrupt compare.
NTRI	Do not print TRACE after interrupt comparison.

DEFAULT VALUE:

NTRI

PGENXX Easy Reference Table (part 1 of 2)

Keyword	Option	Default	Description
BUFSZ	-	x'F800'	Size of program workspace
CLASS	-	Note 1	Instruction classes in stream
	STN		Standard instruction set
	FLT		Floating point instruction set
	DEC		Decimal instruction set
	STP		Storage protection instruction set
	XFL		Extended floating point instruction set
	DAT		DAT instruction set
	MON		Monitor call instruction set
INTR	-	Note 1	Interrupts forced or allowed in stream
	ADR		Addressing exception
	DCD		Decimal divide exception
	DCO		Decimal overflow exception
	DE		Data exception
	EX		Execute exception
	EXO		Exponent overflow exception
	EXU		Exponent underflow exception
	FPD		Floating point divide exception
	FXD		Fixed point divide exception
	FXO		Fixed point overflow exception
	IPER		Programming event recording exception
	MEE		Monitor event exception
	OPCK		Operation exception
	PRT		Protection exception
	PRV		Priviledged operation exception
	SIG		Significance exception
	SPC		Specification exception
	STD		Special operation exception
	SVC		Supervisor call interrupts
LOOP	-	0	Number of random streams to loop on
N	-	Note 2	Number of instructions per stream
NIS	-	Note 3	Number of instructions between interrupts
NOP	-	None	Op codes not to be gene ated
OP	-	None	Op codes only to be generated
PER	-	x'F0'	PER events monitored
	BR	Note 4	Branch instructions
	IF		Instruction fetching
	ST		Storage alteration
	GR		GPR alteration
SEED	-	Note 5	Random-base seed

PGENXX Easy Reference Table (part 2 of 2)

Keyword	Option	Default	Description
STATE	-	Note 1	States used for instruction stream
	SPRV	↓	Priviledged state
	SPER		PER facility
	SBOO		Byte-oriented operands
-	BSC/NBSC	NBSC	Manipulate data using MVCL and CLCL
-	CNT	None	Op code and SVC counters to be displayed
-	DBG/NDBG	NBDG	Do not provide program warning messages
-	EH/NEH	EH	Halt on error
-	EP/NEP	EP	Print errors
-	HST/NHST	NHST	Do not print history on error
-	ISO/NISO	ISO	Isolate on error
-	PDC/NPDC	PDC	Print decimal instruction area on error
-	PFL/NPFL	PFL	Print floating point instr. area on error
-	PFS/NPFS	NPFS	Do not print instruction stream on error
-	PST/NPST	PST	Print standard data area on error
-	PSW/NPSW	PSW	Print stream work area on error
-	QLP/NQLP	NQLP	Do not quick loop on isolated instructions
-	TR/NTR	NTR	Do not trace after stream execution
-	TRI/NTRI	NTRI	Do not trace after interrupt comparison

Notes:

1. *Defaults are determined by system configuration.*
2. *Default is a random number in the range x'1' to x'FF'.*
3. *Default is a random number in the range x'1' to x'20'.*
4. *Default is all PER events (PER = x'F0').*
5. *Default is a random number.*

PATDAT

Program Options

PATDAT is invoked using the following command:

```
START,ID-PATDAT
```

PATDAT tests the DAT features, as follows: The first section of PATDAT generates pseudo-random test cases that verify the DAT interrupt priority scheme.

1. Valid segment tables and page tables are built for three random addresses selected for each test case.
2. Errors are then injected into the test case by setting invalid bit combinations in CR0, CR1, the segment table, and page table entries.
3. The test case runs, and as each interrupt occurs the condition that caused it is corrected. The test case is rerun until all errors have been corrected and the test case is successful.
   Any unexpected interrupts are printed as errors.

One pass through the PATDAT program causes the repetitive execution of one of the random test cases, which is followed by the execution of each of the special tests (described below).

SPECIAL TESTS:

These three special tests are done while in DAT mode and disabled for interrupt. Before execution of the following tests, a Purge Table Lookaside Buffer (PTLB) instruction is issued.

1. PTLB Function:
   This test case does a valid translation, turns on the page invalid bit, does a PTLB, and then another valid translation. A page fault exception interrupt is expected on the last translation.
2. IPTE Function:
   This test case does a valid translation, an IPTE instruction, and another valid translation. A page fault exception interrupt is expected on the last translation. Because neither of the instructions affect the condition code, a random condition code is set up for the start of the test instruction stream and checked for a "no change" when the interrupt occurs.
3. Common Segment Bit Function:
   This test case is done in two parts. Part one has the common bit ON in an entry in the first segment table, and OFF in an entry in the second segment table. Part two has the common bit OFF in the first segment table, and ON in the second.
   Both parts of PATDAT use the same test stream. The test stream does a valid translation, turns on the page invalid bit, loads Control Reg 1 with the second segment table pointer, then does another valid translation. The first case should not cause a page translation, while the second case should.

Should an error message occur from the PATDAT program, the current seed may be entered and looped. For example:

```
MOD-PATDAT00,SEED-123XXXXX,LOOP-1
```

For any confirmed PATDAT error, functional diagnostics covering the the DAT feature should be used.

To stop looping and continue normal execution, key in:

```
MOD-PATDAT00,RESET-LOOP
```

EPAGE1

Program Options

Notes: *Before starting EPAGE1:*

1. *EPAGE1 runs on 4300 processors only, and only in VSE mode. EPAGE1 will not run in 370 mode.*
2. *Start only one copy of EPAGE1. Results can not be predicted when more than one copy of EPAGE1 is active.*
3. *Use the STOP command (STOP,ID-EPAGE100) to end EPAGE1. Pages may be left disconnected if the CANCEL command is used.*
4. *Perform a clear/reset after running EPAGE1 to ensure that pages are configured and addressable.*

Starting EPAGE1

EPAGE1 is started as a default when the CPVSEA or CPVSEB command procedure (CPROC) is selected. For details, see "Invoking Tests - Selecting CPROCS" and "Invoking NST" in the first chapter.

EPAGE1 can also be started, after the NST control program (NSTCP) is loaded, by entering:

```
START,ID-EPAGE100
```

To end EPAGE1, enter: `STOP,ID-EPAGE100`

Optional Tests

All EPAGE1 tests run automatically except for two:

- DEP - Deconfigure Page
- IO ROUTINE

The DEP and IO ROUTINE tests are started with EPAGE1 active using the MODIFY command. Details follow.

DEP - Deconfigure Page
DEP is an EPAGE1 instruction test. For test details, see the DEP test description.

To start the DEP test, enter: `MOD-EPAGE100,DEP`

To end the DEP test, enter: `MOD-EPAGE100,NDEP`

IO ROUTINE
IO ROUTINE is an EPAGE1 function test. For test details, see the IO ROUTINE test description.

To start the IO ROUTINE test:

1. Ready a card reader with blank cards or a tape drive with a scratch tape. Card reader and tape drive types accepted by EPAGE1 are:

Card Readers	*Tape Drives*
1442	2400
2501	3410
2540	3420
3505	8809

2. Enter:

```
MOD-EPAGE100,UA-cuu,DT-tttt
```

where:

cuu = Device and unit address

tttt = Device type

If the device type entered is a card reader, the IO ROUTINE test starts running.

If the device type entered is a tape drive, a message is displayed requesting that IO=ON be entered after ensuring that a scratch tape is mounted. (IO ROUTINE writes an 80-byte record on the tape during initialization of the test.) After entering IO=ON, the IO ROUTINE test starts running.

To end the IO ROUTINE test, enter:

```
MOD-EPAGE100,IO-OFF
```

Run Options

The two EPAGE1 run options are:

- SIZE
- TCNT

The EPAGE1 run options are specified as MODIFY command parameters with EPAGE1 active. The format is:

```
MOD-EPAGE100,SIZE-xx
```

or

```
MOD-EPAGE100,TCNT
```

SIZE

The size parameter specifies the number (in hex) of 2K storage pages available for EPAGE1. The default is 8 pages (16K). The minimum value that can be specified is 2 (4K), and the maximum is 80 (256K).

Note: *The following message may be displayed when the size parameter is changed:*

```
UNABLE TO SET STORAGE TO ADDRESSABLE
STORAGE AREA HELD UNTIL RESET IPL
```

This message indicates that EPAGE1 can not make all previously-defined pages available. The storage area that was defined for these pages is not released to the control program. EPAGE1 assigns a new storage area of the requested size and continues running.

After the above messages are displayed, EPAGE1 can not be stopped, and the previously-defined storage is held until a clear/reset and IPL is performed.

TCNT

The TCNT (test counter) parameter causes an EPAGE1 run summary to print. Each of the 16 EPAGE1 tests is listed with totals of the number of times they have run with each type of page.

Following is an example of the printout that results from entering:

```
MOD-EPAGE100,TCNT.

ROUTINE  ADDRESS   CONNECTED  DISCONNECT INVALID    PAGE0
CLRP     00000000  00000000   00000000   00000000   00000000
CTP      00000000  00000000   00000000   00000000   00000000
DEP      00000000  00000000   00000000   00000000   00000000
DCTP     00000000  00000000   00000000   00000000   00000000
IPB      00000000  00000000   00000000   00000000   00000000
LFI      00000000  00000000   00000000   00000000   00000000
MAD      00000000  00000000   00000000   00000000   00000000
MUN      00000000  00000000   00000000   00000000   00000000
RSP      00000000  00000000   00000000   00000000   00000000
SPB      00000000  00000000   00000000   00000000   00000000
STCAP    00000000  00000000   00000000   00000000   00000000
SIO      00000000  00000000   00000000   00000000   00000000
ALL      00000000  00000000   00000000   00000000   00000000
IFETCH   00000000  00000000   00000000   00000000   00000000
ACCESS   00000000  00000000   00000000   00000000   00000000
STORE    00000000  00000000   00000000   00000000   00000000
```

Test Description

The two types of EPAGE1 tests are:

- Instruction Tests
- Function Tests

EPAGE1 executes the instruction and function tests in a random sequence.

Instruction Tests

There are eleven EPAGE1 instruction tests. Except for the deconfigure page (DEP) test, all EPAGE1 instruction tests run automatically. DEP is an optional test that runs only when selected with the MODIFY command.

Each EPAGE1 instruction test executes one page-control instruction and checks for correct results. The test page is used as the operand of the tested instruction and may be addressable, connected, disconnected, invalid, or page 0.

The EPAGE1 instruction tests are:

CLRPI- Clear Page: This test forces addressing and page state exceptions. The test page is checked for all binary zeros when it is addressable. Page 0 is never used with this test. (Clearing page 0 would destroy the control program.)

CTP - Connect Page: This test connects disconnected pages when the free-frame capacity counter is greater than zero. An invalid-page address causes an addressing exception and an invalid addressable page causes a page transition exception. The assigned index is checked to verify that no entries are the same.

DCTP - Disconnect Page: This test disconnects connected pages. Addressing, specification, and page transition exceptions are forced.

DEP - Deconfigure Page: This test is optional. DEP is selected when EPAGE1 is active using the MODIFY command. For information on how to run the DEP test, see "Optional Tests."

The DEP test may force addressing, specification, and page-transition exceptions, or it may deconfigure page frames and disconnect pages. The frame index is recorded and the number of deconfigured frames is checked. A warning message is displayed when the number of deconfigured pages equals the number allocated for EPAGE1. This usually means that no page frames are available for reconnecting so the tests run on disconnected pages only. (More page frames can be made available by changing the SIZE parameter. For details, see "Run Options.")

IPB - Insert Page Bits: This test compares the programmable, reference, and change bits with their specified value in the page table. Page 0 is not used. An invalid address forces an addressing exception.

LFI - Load Frame Index: This test verifies that the page-frame index equals the index value in the page table. It also checks that no other page-frame index is the same. Exceptions are not forced.

MAD - Make Addressable: This test forces page-transition exceptions for disconnected pages and addressing exceptions for invalid pages. Page 0 and connected pages are addressable.

MUN - Make Unaddressable: This test changes addressable pages to connected. Specification, addressing, and page transition exceptions are forced.

RSP - Retrieve Status and Page: This test uses two pages, neither of which may be page 0. Page-access and addressing exceptions are forced. The retrieved status and page data are checked if a machine save was performed before starting EPAGE1.

SPB - Set Page Bits: This test changes the page descriptor bits to random values, and validates the condition code. Page 0 is not used. An invalid address forces an addressing exception.

STCAP - Store Capacity Counts: This test stores the capacity-counter value on the selected page. The values are checked when the page is addressable. Addressing and page-access exceptions are forced. Page 0 is not used.

Function Tests

There are five EPAGE1 function tests. Except for the IO ROUTINE test, EPAGE1 runs all the function tests automatically. The IO ROUTINE test runs when EPAGE1 is active only when selected with the MODIFY command.

Each EPAGE1 function test performs a 4300-VSE-mode paging function and checks for correct results. Page 0 is not used in any of the function tests.

The EPAGE1 function tests are:

IO ROUTINE

This test is optional and is selected with the MODIFY command. For information on how to run the IO ROUTINE test see "Optional Tests."

The IO ROUTINE test reads a record and stores it on the test page. Data is checked when the test page is addressable. The test page may be addressable, connected, disconnected, or invalid. EPAGE1 verifies that the test ran successfully or than an expected channel program check occurred.

All Instruction Routine

This test executes a string of system control instructions and checks the final page status. The PSW key, PSW mask, and the page description are changed and then restored. The page is disconnected and made addressable when a page frame is available. Page 0 and invalid address are not used in this test.

The following system control instructions are executed by the all instruction routine:

CLRP	- Clear Storage Page
CTP	- Connect Page
DCTP	- Disconnect Page
DEP	- Deconfigure Page
IPB	- Insert Page Bits
IPK	- Insert PSW Key
ISK	- Insert Storage Key
LFI	- Load Frame Index
MAD	- Make Page Addressable
MUN	- Make Page Unaddressable
SPB	- Set Page Bits
STCAP	- Store Capacity Counts
LCTL	- Load Control Registers
LPSW	- Load PSW
RRB	- Reset Reference Bit
SPKA	- Set PSW Key from Address
SSK	- Set Storage Key
SSM	- Set System Mask
STCTL	- Store Control Registers

Access Routine

This test tries to access the test page as the operand of a compare instruction. Page-access and addressing exceptions are forced when the selected page is not addressable.

IFETCH Routine
This test causes a branch to an instruction in the test page. The instruction in the page is a branch-back to the IFETCH routine. Addressing and page-access exceptions are forced.

STORE Routine
This test tries to change four bytes in the test page. A connected or disconnected page causes an access exception. An invalid address causes an addressing exception.

Running Restrictions

1. EPAGE1 runs on 4300 processors only, and only in VSE mode. EPAGE1 will not run in 370 mode.
2. Start only one copy of EPAGE1. Results can not be predicted when more than one copy of EPAGE1 is active.
3. Use the stop command (STOP,ID=EPAGE100) to end EPAGE1. Pages may be left disconnected if the cancel command is used.
4. Perform a clear/reset after running EPAGE1 to ensure that pages are configured and addressable.
5. The reference and change page-descriptor bits are changed when a page from the EPAGE1 paging pool is printed, displayed, or altered. If this is done while EPAGE1 is running, EPAGE1 cannot predict the reference and change bits.
6. Capacity counter errors and incorrect results in the CTP (connect page) test occur when more than one copy of EPAGE1 is active. See item 2 in the above list.
7. The RSP (retrieve status and page) test fails when a machine save (MSAVE) is performed with EPAGE1 active.

EPAGE1 prints an error message when an error is sensed. For an example of an EPAGE1 error printout, see "Sample Error Printouts."

IODRVR

Program Options

CNT - Display Internal Counters

In addition to the DROP and VARY control options described in the first section, it is possible to obtain a list of the number of test passes by device. To obtain the listing, key in:

```
MOD-IODRVR00,CNT
```

The following is a sample of the detail presented in the printout (depending on the I/O configuration available for testing):

```
 IODRVR00-IODPDT-CLOCK=14-06.51.23.123456
             INTERNAL COUNTERS
DT        UA              COUNT     ERROR
3278      00000010        0005      0000
3289      00000013        000C      0000
2540      0000002C        0003      0000
1403      0000002E        0006      0000
2314      00000191        0003      0000
3310      00000240        000B      0000
3310      00000241        000B      0000
```

Note: *Entering MOD-IODRVR00,CNT will resume testing if entered after a VARY command (in the same manner that MOD-IODRVR00,GO resumes testing.*

UAT - Unit Availability Table

After pre-configuration is completed and the command

```
MOD-IODRVR00,CNFG
```

has been entered, a unit availability table is printed:

```
UNIT    UNIT      ADDRESS        TEST
TYPE    CLASS     CUU            LEVEL     FEATURES
3278     GR       00000011         2
2540     RD       0000012C         0
2540     PU       0000012D         0       COL BIN
3420     TP       00000180         2       FILE PROTECTED REEL
3420     TP       00000181         0       3420 MOD3, 9 TRK
3310     DS       00000240         0
```

If additional devices are VARY'd after IODRVR is started, the UAT can be printed again by entering:

```
MOD-IODRVR00,PUT
```

The new printout lists the current configuration, including changes.

Stopping IODRVR

The STOP command is not effective until after the GO parameter has been manually entered (not part of the CPROC). To terminate before the GO parameter, use the CANCEL command.

Example:

```
CANCEL,ID-IODRVR00
```

Diagnostic and Service Aids

An example of a diagnostic approach would be:

1. Run PGENXX and record the current seed for each error printout and the op codes that are flagged.
2. MODIFY the program with the NOP option to not run these failing op codes. When you can get the program to run error-free, you will have a list of failing op codes.
3. MODIFY the program with the OP option to run just these op codes to verify the failures.
4. MODIFY the program with the SEED option to loop on a particular set of op codes.
5. You may want to invoke the Quick Loop option (QLP) to place the system into a small scope loop.
6. If errors indicate that PER is failing, you may want to drop PER testing to verify that all other testing is working. To do this enter:

   ```
   MOD-PGENXX00,PER-00
   ```

   Note: *The delimiters = and - may be used interchangeably.*

Processor Testing (Example)

1. Assigning a secondary output device is recommended.
   Enter:

   ```
   ASSIGN,S,UA-00E,DT-1403
   ```

   This puts the bulk printouts on the printer, leaving the messages on the console.

2. Enter:

   ```
   START,ID-PGENXX,HALT
   ```

3. Refer to CPROC Service Aids (Section on "IODRVR Service Aids.") to check for op codes that are not tested for your system.
4. After the message stating that the program has been loaded, enter:

   ```
   MODIFY-PGENXX00,BUFSZ-4000
   ```

5. After a few minutes to display the number of correct and error passes, enter:

   ```
   DISPLA,COUNTR
   ```

6. To display the number of times each op code has been tested, enter:

   ```
   MODIFY-PGENXX00,CNT
   ```

This op code count information is also printed on the output printer when the program is STOPped.

7. If an error is detected, the program is suspended. You can continue by entering:

   ```
   RESUME,ID-PGENXX00
   ```

   Or, by entering the base seed number you can repeat the failing set of instructions. After it repeats, the program continues with random test cases.

   ```
   MODIFY-PGENXX00,SEED-xxxxxxxx
   ```

   To LOOP on a particular SEED, enter:

   ```
   MODIFY-PGENXX00,SEED-xxxxxxxx,LOOP-1
   ```

8. To drop op codes from testing, enter:

   ```
   MODIFY-PGENXX00,NOP-44
   ```

   This deletes op code 44 (Execute) in the testing.

9. To select specific op codes, you should first reset any NOP options and then add those desired.

   ```
   MODIFY-PGENXX00,RESET-NOP,OP-5A
   ```

10. To enter a tight loop after an error is detected, enter:

    ```
    MODIFY-PGENXX00,QLP
    ```

Service Aids

1. In starting a test, the test is loaded into the system by a 6-digit name (START,ID-PGENXX). However, after it is loaded, use an 8-digit name (MODIFY-PGENXX00) to modify the test. This is because you have the ability to bring in several copies of the same test to load down the system, and you must be able to modify each copy separately.
2. Most commands require that the name of the test be identified using (ID-xxxxxx). However, with the MODIFY command, the (ID) is not required; only (-xxxxxx) is necessary.
3. To list the commands within a CPROC do not include the parameter (,NCOM) when doing a LIST.
4. If the secondary output device gets a unit check while printing, the output automatically switches to the primary output device.
5. The output printer requires a loaded forms control buffer or a "1" hole punch per page.
6. Errors encountered within the processor tests during auto-configuration of IODRVR can cause unpredictable indications. Run the processor tests alone for a longer period of time (more than 3 minutes). Stop the processor tests and STArt IODRVR alone, to isolate a failure.
7. When executing IODRVR, a tape without a tape mark causes the tape to run away. Refer to EXIO to write a tape mark.

8. Commands entered incorrectly are flagged with an asterisk (*). Re-enter only from the * using MODify to save entry time.
9. If extensive testing of MVCL (move character long) or CLCL (compare logical long) is required, increase the BUFSZ of PGENXX above X'F800".
10. The maximum number of copies of PGENXX is 14 (X'E'). One or two copies is recommended, depending on storage size (one copy with a BUFSZ of 4000, and one copy with the default BUFSZ of F800). All other programs must be run with only one copy.
11. All values are in hex, unless specifically stated otherwise.

Check Stop Mode

Setting the system in check stop mode can be done to force red-light errors.

Program Function Keys (PFK)

When a 4300 processor is in display mode, PFK keys may be set to eliminate keying repetitive information:

1. Enter the desired information; for example:

```
+mod-update00,
```

2. Press ALT key and the desired PF key (PF1 for example).

The PF1 key is now set to enter:

```
MOD-UPDATE00,
```

on the console whenever the ALT and the PF1 key is pressed. The + before the command causes the command to be retained on the screen so that it may be modified before pressing the ENTER key.

CPROC Service Aids

CPROCs for each system type have been customized for correct operation of all programs. Part of the customization requires certain op codes to be "No-op'd" depending on the system involved, features, and options. Before attempting to execute individual programs using the STArt command, check the appropriate CPROC for recommended no-ops. Failure to follow this procedure can result in invalid error indications.

To see the commands within a CPROC on the console, add the parameter CPCOM; for example:

```
CPROC,CP-CP370A,CPCOM
```

when invoking the CPROC, or use UPDATE to LIST the NST tape or disk *without* the parameter NCOM; for example:

```
STA,ID-UPDATE
AS,S,UA-00E,DT-1403
MOD-UPDATE00,UAO-180,TAO,LIST
```

To alter existing CPROCs, see CPROCs in UPDATE section.

Sample Error Printouts

This chapter provides information regarding error output. Because of the amount of potential output, it is impossible to cover the detail of all possible error conditions.

A CPROC which defines defaults has been provided to minimize the initial output. Manual selection of specific options using the MODIFY command provides the user with more detailed output concerning specific tests.

If sufficient storage is available, two copies of PGEN provide an optimum test for detecting processor failures. One copy of PGEN should be run with a BUFSZ of X'4000', and the other copy with the default value of X'F800'. The maximum number of copies of PGEN to be run is 14 (X'E'). However, the most efficient is 2 copies.

PGENXX Error Printout Sample

Steps A through G are keyed to the sample error printout reproduced as Figures 2, 3, and 4.

Immediately following Figure 4, additional information is provided regarding the error printout.

A Error - MISCOMPARE AT INTERRUPT PSW - ISOLATION COMPLETED:

This message indicates that an interrupt *was forced* by the program, and the expected and actual results did not compare. The error was successfully recreated, and a summary provided.

B Program Control Flags:

This area indicates the status of options at the time of failure (refer to PGEN Program Control Options for explanation). The amount of detail printed as a result of an error being encountered can be controlled by changing the status of these options. If testing is invoked by means of a CPROC, the default options selected are displayed. The important point to *keep in mind* is that, in most instances, PGENXX generates an isolation summary. If a summary cannot be provided, a full printout of the instruction stream and work area in use at the time of the error is provided.

C PGEN-Random Instruction Stream - Actual: The three lines of data following this heading comprise the error isolation summary.

- LAST indicates the last successful instruction completed, which was an SVC.
- The third statement indicates that the failure was encountered in this instruction. This is an EXECUTE instruction (X'44') to perform the operation at address 0005D8A4. The operation at that address, shown as TARGET, is another EXECUTE (X'44'). This is an invalid instruction sequence which resulted in the mismatched INTR CODEs in the sample printout.

D The headings PGEN - Interrupt History Actual and Interrupt History Simulated contain the following abbreviations:

NSI - next sequential instruction (the ERROR shown under this heading shows the address as incremented at the time of the failing instruction. To determine the address of the failing instruction, this address must be decremented by the IL (in this case, by 4 bytes.)

IL - instruction length.

CC - condition code (set as a result of executing the failing instruction).

PGM MSK - program mask.

INTR CODE - interrupt code.

ECM - extended control mode.

PRB - problem state.

PER - program event recording.

TRC - trace facility.

DAT - dynamic address translation.

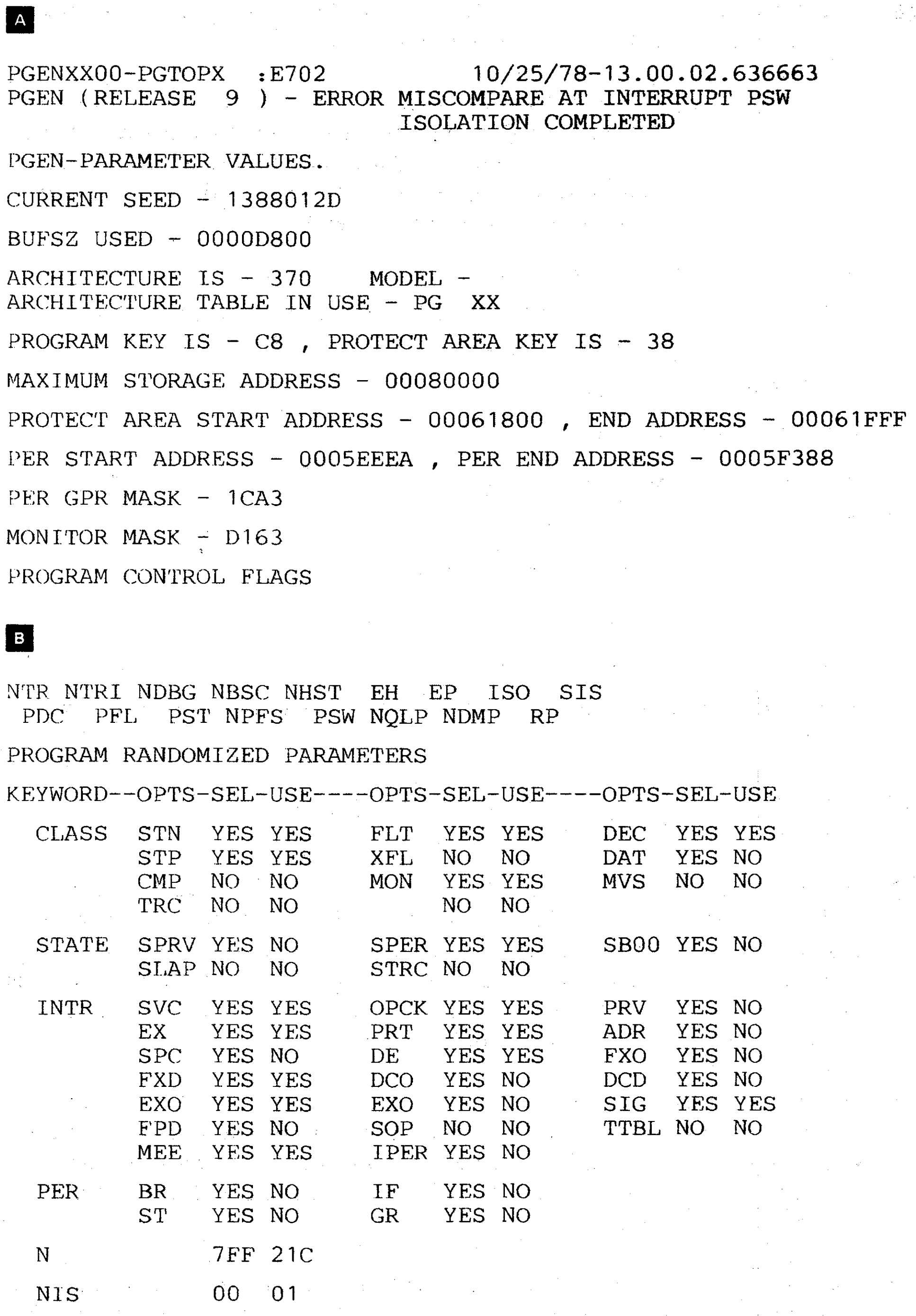
The simulated (expected) INTR code is 0003. The actual INTR code is 0006.

E The contents of the general purpose, control, and floating point registers at the time of the failing instruction.

F Instruction Stream Work Area: If PGENXX had been unable to isolate and summarize the error data, this is the work area involved with the error that was encountered. As can be seen at the first address, the first instruction encountered after the original X'44' is another X'44', which is an invalid sequence. Additional information regarding the instruction stream work area follows the sample printout.

G Decimal Instruction Data Area: In this example, the error that occurred was not contained in the decimal work area; had the option NPDC been selected, this output would have been suppressed. If an error did occur in the decimal work area and the NPDC option was selected, the option would be overridden.

The remainder of the printout, beginning with the decimal instruction data area and continuing through the floating point areas, would not have been printed had different options been selected. The additional material is shown to assist in analysis, by pointing out that in most instances all the required detail is summarized in the isolation report section (Items A through F). It also familiarizes you with the small, but most important, section of the printout that could be overlooked if the options selected result in a large amount of data.



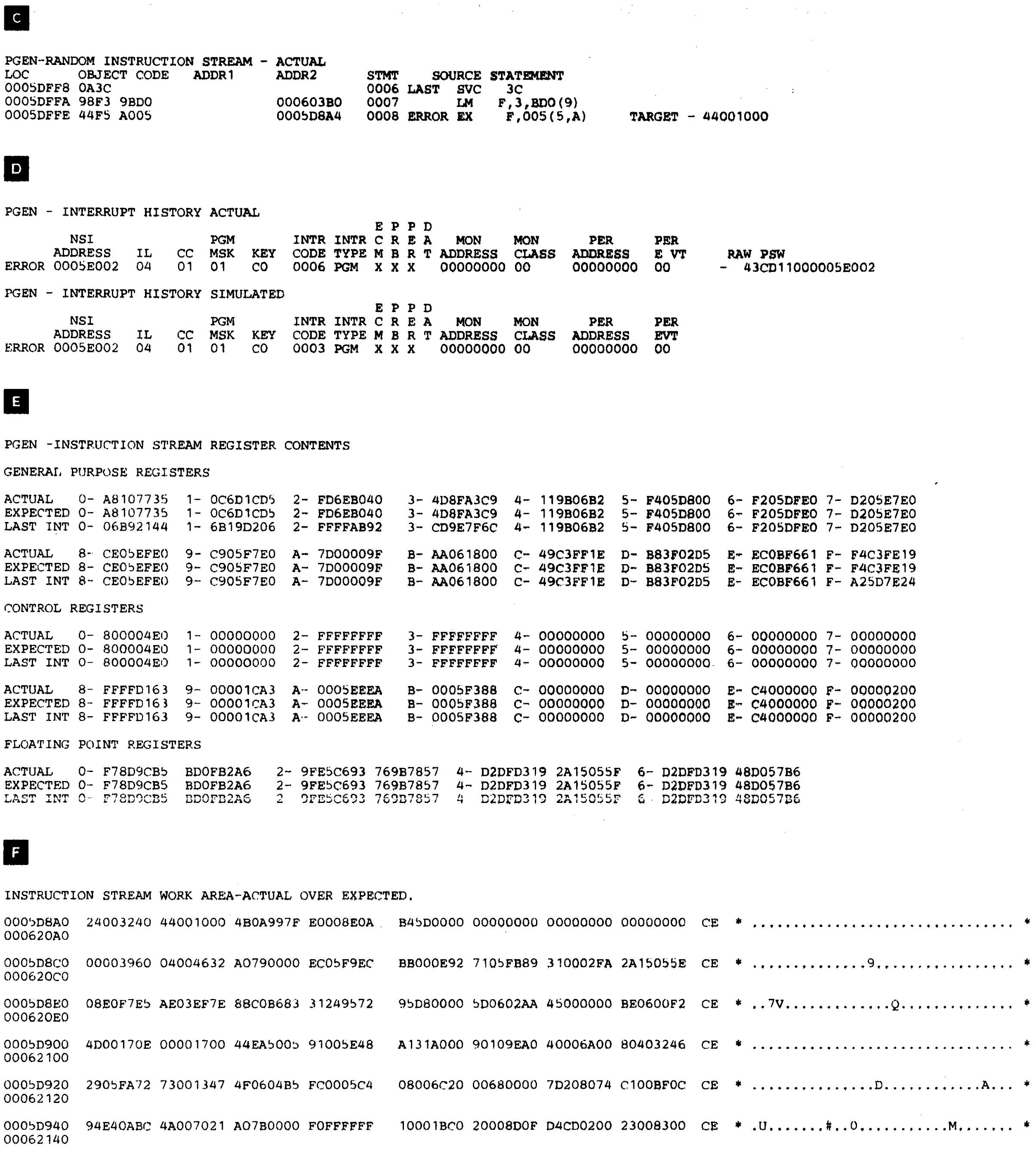
A

```
PGENXX00-PGTOPX   :E702          10/25/78-13.00.02.636663
PGEN (RELEASE  9 ) - ERROR MISCOMPARE AT INTERRUPT PSW
                           ISOLATION COMPLETED

PGEN-PARAMETER VALUES.

CURRENT SEED - 1388012D

BUFSZ USED - 0000D800

ARCHITECTURE IS - 370    MODEL -
ARCHITECTURE TABLE IN USE - PG  XX

PROGRAM KEY IS - C8 , PROTECT AREA KEY IS - 38

MAXIMUM STORAGE ADDRESS - 00080000

PROTECT AREA START ADDRESS - 00061800 , END ADDRESS - 00061FFF

PER START ADDRESS - 0005EEEA , PER END ADDRESS - 0005F388

PER GPR MASK - 1CA3

MONITOR MASK - D163

PROGRAM CONTROL FLAGS
```

B

```
NTR NTRI NDBG NBSC NHST  EH   EP   ISO  SIS
 PDC  PFL  PST NPFS  PSW NQLP NDMP  RP

PROGRAM RANDOMIZED PARAMETERS

KEYWORD--OPTS-SEL-USE----OPTS-SEL-USE----OPTS-SEL-USE

  CLASS  STN  YES YES    FLT  YES YES    DEC  YES YES
         STP  YES YES    XFL  NO  NO     DAT  YES NO
         CMP  NO  NO     MON  YES YES    MVS  NO  NO
         TRC  NO  NO          NO  NO

  STATE  SPRV YES NO     SPER YES YES    SB00 YES NO
         SLAP NO  NO     STRC NO  NO

  INTR   SVC  YES YES    OPCK YES YES    PRV  YES NO
         EX   YES YES    PRT  YES YES    ADR  YES NO
         SPC  YES NO     DE   YES YES    FXO  YES NO
         FXD  YES YES    DCO  YES NO     DCD  YES NO
         EXO  YES YES    EXO  YES NO     SIG  YES YES
         FPD  YES NO     SOP  NO  NO     TTBL NO  NO
         MEE  YES YES    IPER YES NO

  PER    BR   YES NO     IF   YES NO
         ST   YES NO     GR   YES NO

  N           7FF 21C

  NIS         00  01
```

Figure 2. PGENXX Error Printout (1 of 3)



C

```
PGEN-RANDOM INSTRUCTION STREAM - ACTUAL
LOC        OBJECT CODE    ADDR1       ADDR2        STMT     SOURCE STATEMENT
0005DFF8 0A3C                                      0006 LAST  SVC    3C
0005DFFA 98F3 9BD0                    000603B0     0007       LM    F,3,BD0(9)
0005DFFE 44F5 A005                    0005D8A4     0008 ERROR EX     F,005(5,A)      TARGET - 44001000
```

D

```
PGEN - INTERRUPT HISTORY ACTUAL
                                                 E P P D
        NSI                PGM          INTR INTR C R E A    MON      MON       PER      PER
      ADDRESS    IL    CC  MSK   KEY    CODE TYPE M B R T ADDRESS  CLASS   ADDRESS   E VT        RAW PSW
ERROR 0005E002   04    01  01    C0     0006 PGM  X X X    00000000 00      00000000  00        -  43CD11000005E002

PGEN - INTERRUPT HISTORY SIMULATED
                                                 E P P D
        NSI                PGM          INTR INTR C R E A    MON      MON       PER      PER
      ADDRESS    IL    CC  MSK   KEY    CODE TYPE M B R T ADDRESS  CLASS   ADDRESS   EVT
ERROR 0005E002   04    01  01    C0     0003 PGM  X X X    00000000 00      00000000  00
```

E

```
PGEN -INSTRUCTION STREAM REGISTER CONTENTS

GENERAL PURPOSE REGISTERS

ACTUAL    0- A8107735   1- 0C6D1CD5   2- FD6EB040   3- 4D8FA3C9   4- 119B06B2   5- F405D800   6- F205DFE0 7- D205E7E0
EXPECTED 0- A8107735   1- 0C6D1CD5   2- FD6EB040   3- 4D8FA3C9   4- 119B06B2   5- F405D800   6- F205DFE0 7- D205E7E0
LAST INT 0- 06B92144   1- 6B19D206   2- FFFFAB92   3- CD9E7F6C   4- 119B06B2   5- F405D800   6- F205DFE0 7- D205E7E0

ACTUAL    8- CE05EFE0   9- C905F7E0   A- 7D00009F   B- AA061800   C- 49C3FF1E   D- B83F02D5   E- EC0BF661 F- F4C3FE19
EXPECTED 8- CE05EFE0   9- C905F7E0   A- 7D00009F   B- AA061800   C- 49C3FF1E   D- B83F02D5   E- EC0BF661 F- F4C3FE19
LAST INT 8- CE05EFE0   9- C905F7E0   A- 7D00009F   B- AA061800   C- 49C3FF1E   D- B83F02D5   E- EC0BF661 F- A25D7E24

CONTROL REGISTERS

ACTUAL    0- 800004E0   1- 00000000   2- FFFFFFFF   3- FFFFFFFF   4- 00000000   5- 00000000   6- 00000000 7- 00000000
EXPECTED 0- 800004E0   1- 00000000   2- FFFFFFFF   3- FFFFFFFF   4- 00000000   5- 00000000   6- 00000000 7- 00000000
LAST INT 0- 800004E0   1- 00000000   2- FFFFFFFF   3- FFFFFFFF   4- 00000000   5- 00000000   6- 00000000 7- 00000000

ACTUAL    8- FFFFD163   9- 00001CA3   A- 0005EEEA   B- 0005F388   C- 00000000   D- 00000000   E- C4000000 F- 00000200
EXPECTED 8- FFFFD163   9- 00001CA3   A- 0005EEEA   B- 0005F388   C- 00000000   D- 00000000   E- C4000000 F- 00000200
LAST INT 8- FFFFD163   9- 00001CA3   A- 0005EEEA   B- 0005F388   C- 00000000   D- 00000000   E- C4000000 F- 00000200

FLOATING POINT REGISTERS

ACTUAL    0- F78D9CB5   BD0FB2A6   2- 9FE5C693 769B7857   4- D2DFD319 2A15055F   6- D2DFD319 48D057B6
EXPECTED 0- F78D9CB5   BD0FB2A6   2- 9FE5C693 769B7857   4- D2DFD319 2A15055F   6- D2DFD319 48D057B6
LAST INT 0- F78D9CB5   BD0FB2A6   2  9FE5C693 769B7857   4  D2DFD319 2A15055F   6  D2DFD319 48D057B6
```

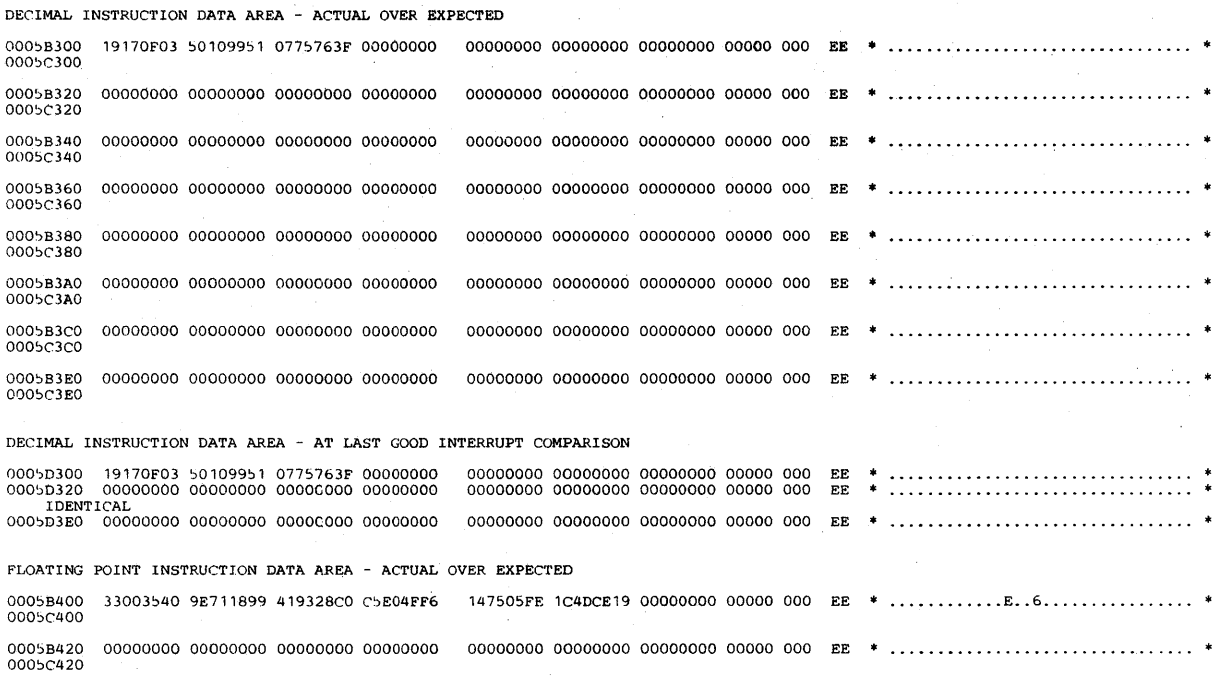
F

```
INSTRUCTION STREAM WORK AREA-ACTUAL OVER EXPECTED.

0005D8A0   24003240 44001000 4B0A997F E0008E0A   B45D0000 00000000 00000000 00000000   CE  * ................................ *
000620A0

0005D8C0   00003960 04004632 A0790000 EC05F9EC   BB000E92 7105FB89 310002FA 2A15055E   CE  * ..............9................. *
000620C0

0005D8E0   08E0F7E5 AE03EF7E 88C0B683 31249572   95D80000 5D0602AA 45000000 BE0600F2   CE  * ..7V.............Q.............. *
000620E0

0005D900   4D00170E 00001700 44EA5005 91005E48   A131A000 90109EA0 40006A00 80403246   CE  * ................................ *
00062100

0005D920   2905FA72 73001347 4F0604B5 FC0005C4   08006C20 00680000 7D208074 C100BF0C   CE  * ...............D............A... *
00062120

0005D940   94E40ABC 4A007021 A07B0000 F0FFFFFF   10001BC0 20008D0F D4CD0200 23008300   CE  * .U.......#..0...........M....... *
00062140
```

Figure 3. PGENXX Error Printout (2 of 3)

The terms actual-over-expected shown on the printout depict only the actual data. Had there been a difference between the actual/expected results, the data on both lines would have been printed. The actual results are printed immediately above the expected.

G



```
DECIMAL INSTRUCTION DATA AREA - ACTUAL OVER EXPECTED

0005B300   19170F03 50109951 0775763F 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C300

0005B320   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C320

0005B340   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C340

0005B360   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C360

0005B380   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C380

0005B3A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C3A0

0005B3C0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C3C0

0005B3E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C3E0

DECIMAL INSTRUCTION DATA AREA - AT LAST GOOD INTERRUPT COMPARISON

0005D300   19170F03 50109951 0775763F 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005D320   00000000 00000000 0000C000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
    IDENTICAL
0005D3E0   00000000 00000000 0000C000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *

FLOATING POINT INSTRUCTION DATA AREA - ACTUAL OVER EXPECTED

0005B400   33003540 9E711899 419328C0 C5E04FF6   147505FE 1C4DCE19 00000000 00000 000  EE  * ............E..6................ *
0005C400

0005B420   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000 000  EE  * ................................ *
0005C420
```

Figure 4. PGENXX Error Printout (3 of 3)

Additional Output Analysis Information

Parameter Display: The operator input parameters must be displayed so that an error environment can be recreated successfully. This is the first display produced by the test program. The display includes:

- Title, identifying current program release level, and type of display being presented, either error or trace.
- Current seed, or the random number which was in use at the completion of the program pass when this output was produced.
- Instruction operation codes selected for testing (OP).
- Instruction operation codes specified so that they are not generated (NOP).
- Buffer size (BUFSZ) which was operator supplied or default.
- Architecture and model of system under test. If the architecture or model is not supported by the test program, the test program indicates this, and provides a default table.
- Program storage key and the storage key used for protection checking, as well as the protection area start and end addresses.
- The starting address, ending address and GPR mask used for program event recording. These values are supplied by the test program.
- The monitor mask, also supplied by the program.
- The program control flags which were selected by the operator or defaulted by the program are listed.
- The program-randomized parameters for CLASS, STATE, INTR, and PER ere listed. Next to each option is an indication as to whether it is selected, and whether it is used by the program for the pass which just completed.
- N - the number of instructions per stream. If the values are different, the indication is that the parameter was allowed to default. If identical, the operator specified a value to be used. The actual number of instructions built may be greater than this value because of prerequisite or post-requisite instructions forced into the stream for simulation purposes.
- NIS - the number of instructions between interrupts, which can be operator-supplied or defaulted. If the first group is 00, then the second group is a random number in 01-20 hex. If the operator specified the number of instructions between interrupts, then the two groups should be equal and show the value specified. The actual number of instructions between interrupts may vary somewhat because of the interrupt that is to be forced and whether that interrupt can occur with the op code selected.

Figure 5 (following) is another example highlighting the information shown in the previous error printout.

```
PGEN (RELEASE X) - ERROR MISCOMPARE AT INTERRUPT PSW

PGEN - PARAMETER VALUES

CURRENT SEED - CFD1559D
OP - 44 - OE -  OF
NOP -
BUFSZ USED - 00003000

ARCHITECTURE IS - 370

PROGRAM KEY IS - 60, PROTECT AREA KEY IS - 90
PROTECT AREA START ADDRESS - 0003EA0, END ADDRESS - 0003FA0

PER START ADDRESS - 0003D8A7, PER END ADDRESS - 0003DE52

PER GPR MASK - A26F

MONITOR MASK - 971D

PROGRAM CONTROL FLAGS:
NTR  NTRI  NDBG  NBSC  NHST  EH  EP  ISO
PDC  PFL  PST  NPFS  PSW  NQLP  NDMP

PROGRAM RANDOMIZED PARAMETERS:

KEYWORD--OPTS-SEL-USE----OPTS-SEL-USE----OPTS-SEL-USE

CLASS    STN  YES YES    FLT  YES YES    DEC  YES YES
         STP  NO  NO     XFL  NO  NO     DAT  NO  NO
         CMP  NO  NO     MON  YES YES    TRC  NO  NO

STATE    SPRV YES NO     SPER YES YES    SB00 YES YES
         STRC NO  NO

INTR     SVC  YES YES    OPCK YES NO     PRV  YES NO
         EX   YES YES    PRT  YES NO     ADR  YES NO
         SPC  YES NO     DE   YES YES    FXO  YES YES
         FXD  YES YES    DCO  YES NO     DCD  YES YES
         EXO  YES YES    EXU  YES YES    SIG  YES NO
         FPD  YES NO     SOP  NO  NO     TTBL NO  NO
         MEE  YES NO     IPER YES NO

PER      BR   YES NO     IF   YES NO
         ST   YES NO     GR   YES NO

N             7FF 038
NIS           00  0E
```

Figure 5. Test Parameter Display

Instruction Stream Display: The display shows a formatted listing of the random instruction stream which previously executed. All addresses are from the particular test copy used by the processor for execution. An example is given in Figure 6. The display includes:

- Page title and column titles identifying the output following.
- The location address for each instruction within the random stream.
- The object code representation for each instruction in halfword increments.
- The effective address for each operand of an instruction which references a storage location. If the instruction is not susceptible to a fetch or store access, only the operand displacement is identified. Otherwise, the effective address is a result of the calculation of base plus index plus displacement.
- A statement sequence field for each instruction in the stream.
- A field which contains either "LAST," indicating the location where an interrupt was forced and all comparisons between actual and expected results were determined to be equal, or "ERROR," indicating the location where an interrupt was forced and a mismatch was detected in comparison.
- The random instruction in a symbolic format containing:
  - The instruction op code mnemonic
  - The formatted instruction operands
- If the random instruction is determined to be an EXECUTE instruction, the target or subject of the EXECUTE is displayed on the same line with the following considerations:
  - The target is identified in object code format if it is accessible by the EXECUTE.
  - If the target is not accessible, the indication is:
    - PROTECTED if the target is in an area which is storage-protected.
    - OUT OF ADDR SPACE if the target is at a location address that is greater than the machine storage.
    - OUT OF WORK SPACE if the target is at a location address that is greater than the workspace allocated to test program.

```
PGENXX - RANDOM INSTRUCTION STREAM
LOC     OBJECT CODE           ADDR1  ADDR2  STMT         SOURCE STATEMENT
4F200   4350 A0E0                    4F0E0  0001         IC    5,0E0(0,A)
4F204   445F A07D                    4F0E2  0002         EX    5,07D(F,A)
                                                               (NOTE 1)
4F208   3A22                                0003         AER   2,2
4F20A   7A40 D000                    4F400  0004         AE    4,000(0,D)
4F20E   F364 E272 E23C 4F772         4F73C  0005         UNPK  272(6,E)
                                                               23C(4,E)
4F214   6F40 D018                    4F418  0006         SW    4,018(0,D)
4F218   4191 F2C9                    2C9    0007         LA    9,2C9(1,F)
4F21C   1B32                                0008         SR    3,2
4F21E   0A05                                0009 LAST    SVC   05
4F220   4431 F2EF                    4FAA4  000A ERROR   EX    3,23F(1,F)
                                                               (NOTE 2)
4F224   1F60                                000B         SLR   6,0
4F226   4E0F                         4F6D0  000C         CVD   0,16B(F,E)
4F228   8D62 D58D                    58D    000D         SLDL  6,258D(D)
4F22C   3040                                000E         LPER  4,0
4F22E   0000                                000F         ILLG2 0,0
```

Notes:

1. *Target - FDA1C000C00C*
2. *Target - PROTECTED*

Figure 6. Random Instruction Stream Display

Interrupt History Display - The interrupt history, which is a result of simulation and execution of the random instruction stream, provides the basis for determination of errors. The interrupts forced within the stream, therefore, become the checkpoints for comparisons. Figure 7 is an example of an interrupt history. The display includes:

- Title of the history being displayed
- Next sequential instruction address
- Instruction length (in bytes)
- Condition code
- Program mask
- Key
- Interrupt code
- Interrupt type
- Flags, if on, indicating program execution states in:
    - EC Mode
    - Problem State
    - Program Event Recording
    - Dynamic Address Translation
    - Trace
- Monitor address and monitor class
- PER address and PER event

- The raw PSW if the history is the actual interrupt history

The program indicates, in the first field of the history, whether the actual or simulated, if that particular interrupt is the "LAST" forced within the stream, or an "ERROR" resulting from a mismatch in PSWs or data comparisons.

```
PGENXX - INTERRUPT HISTORY ACTUAL

                                                        E P P D  T
            NSI                  PGM  INTR INTR         C R E A  R
            ADDRESS    IL    CC  MSK  KEY  CODE TYPE    M B R T  C
            4F224      02    00  0D   B0   0080 PGM     X   X
            4F228      04    00  0D   B0   0080 PGM     X   X
            4F22A      02    00  0D   B0   0089 PGM     X   X
            4F22E      04    02  0D   B0   0080 PGM     X   X
            4F234      06    02  0D   B0   0080 PGM     X   X
ERROR       4F238      04    02  0D   B0   0004 PGM     X   X

PGENXX - INTERRUPT HISTORY SIMULATED

                                                        E P P D  T
            NSI                  PGM  INTR INTR         C R E A  R
            ADDRESS    IL    CC  MSK  KEY  CODE TYPE    M B R T  C
            4F224      02    00  0D   B0   0080 PGM     X   X
            4F228      04    00  0D   B0   0080 PGM     X   X
            4F22A      02    00  0D   B0   0089 PGM     X   X
            4F22E      04    02  0D   B0   0080 PGM     X   X
            4F234      06    02  0D   B0   0080 PGM     X   X
ERROR       4F238      04    02  0D   B0   0084 PGM     X   X

CPU - INTERRUPT HISTORY ACTUAL

MON          MON       PER              PER
ADDRESS      CLASS     ADDRESS          EVT   RAW PSW
0000         00        4F222            40    43BC0D000004F224
0000         00        4F224            40    43BC0D000004F228
0000         00        4F228            40    43BC0D000004F22A
0000         00        4F22A            40    43BC0D000004F22E
0000         00        4F22E            60    43BC0D000004F234
0000         00        00000            00    43BC0D000004F238

CPU - INTERRUPT HISTORY SIMULATED

MON          MON       PER              PER
ADDRESS      CLASS     ADDRESS          EVT
0000         00        4F222            40
0000         00        4F224            40
0000         00        4F228            40
0000         00        4F22A            40
0000         00        4F22E            60
0000         00        4F234            40
```

Figure 7. Interrupt History Display

Register Display - To correctly interpret the random instruction stream, the register contents at the time the stream executed are preserved and displayed. Each of the register groups, general purpose, control, or floating-point, has a register value for the actual or processor execution (expected or simulator result), and in the case of an error, the register value at the last interrupt comparison point.

Data Area Display - A data area display is provided for each area that is referenced by the random instruction stream. The display is made in an actual-over-expected fashion for ease of comparison. If a particular data area was not utilized by the stream, no display is made. In the event that the output is a result of an error, a data display is provided of the area at the last good interrupt comparison. The areas displayed are:

- Instruction Stream Work Area
- Decimal Instruction Data Area
- Floating Point Instruction Data Area
- Standard Instruction Data Area

Isolation - Output provided as a result of isolation is the same as described in the previous sections with the following exceptions:

- The error message is followed by either:
  - ERROR ISOLATION COMPLETE

    or
  - PARTIALLY ISOLATED STREAM DID NOT FAIL WHEN REPEATED
- The Instruction Stream Display should contain only the instruction necessary to recreate the failure.
- The Interrupt History should contain only the history of interrupts necessary to re-create the failure.

Counter Display - Internal counters are maintained by the test program for each interrupt forced within the random instruction stream, and for each op code found within the architecture table. This count represents the number of times an operation code is generated or an interrupt is anticipated, as predicted by the simulator routines of the test program. In addition, the op code counters provide two counts. The first counter represents the total number of times an op code was used, either with interrupts forced or without interrupts, and the second count represents the total number of times the op code was built and used correctly.

PATDAT Error Printout Sample

```
 PATDAT00-PATCMP  :I701            10/25/78-12.51.39.401522
PGM PATDAT00 UNEXPECTED INTERRUPT RECEIVED
RELEASE LEVEL-05

PGM OPTIONS-             **CNTL**     **MODE**     **PAGE**     **SEG**     **RAND**     **SPEC**
                           NTR          DAT          4K           1M         MVC          TCA
                           EP           EC           2K           64K        LRA          NTCB
                           EH           NBC                                  NIPT         NTCC
                           NHST                                              NTPR         NTCD
                           NEL                                               EXE          NTCE
                           RP                                                             NTCF

** OPTIONS THIS TEST**
ARCH-MODEL    370
MAX STORAGE    00080000
CURRENT SEED  2B1E4653
ADDRESSING       24-BIT
MODE                DAT
SEGMENT SIZE        64K
PAGE SIZE            2K
INSTRUCTION    D20720003000   MVC
RAND TEST      CRO BITS (8-12) INVALID
TRANS SPEC EXCEPTION

PROGRAM STATUS WORD

EXPECTED   INT CODE-0012  ADDR-00243912 CC-11 ILC-XX  TRANS ADDR-XXXXXXXXXX
ACTUAL     INT CODE-0012  ADDR-0024390E CC-11 ILC-10  TRANS ADDR-XXXXXXXX RAW PSW-07FC30000024390E

GENERAL PURPOSE REGISTERS

EXPECTED  0-00000000  1-00000000  2-0BEC1F50  3-4F584457  4-83243906  5-00000000  6-00000000  7-00000000
ACTUAL    0-00000000  1-00000000  2-0BEC1F50  3-4F584457  4-83243906  5-00000000  6-00000000  7-00000000

EXPECTED  8-00000000  9-00000000  A-00000000  B-00000000  C-00000000  D-00000000  E-00000000  F-00000000
ACTUAL    8-00000000  9-00000000  A-00000000  B-00000000  C-00000000  D-00000000  E-00000000  F-00000000

CONTROL REGISTERS

ACTUAL    0-806004E0  1-0D076800  2-FFFFFFFF  3-FFFFFFFF  4-00000000  5-00000000  6-00000000  7-00000000
ACTUAL    8-FFFFFFFF  9-00000000  A-00000000  B-00000000  C-00000000  D-00000000  E-C4000000  F-00000200

STORAGE ADDRESS MAP
            VALUE    VIRTUAL    REAL      STO       STE       PTO       PTE      PAGE      PAGE
             USED     EQUIV     EQUIV     ADDR      ADDR      ADDR      ADDR     ADDR      DISP
TEST INST 83243906  00243906  00078906  00076800  00076890  00077040  0007704E  0007880  000000106
OPERAND 1 0BEC1F50  00EC1F50  00078750  00076800  00076BB0  00077000  00077006  0007800  000000750
OPERAND 2 4F584457  00584457  00079457  00076800  00076960  00077080  00077090  0007900  000000457
EXEC INST 8324390E  0024390E  0007890E  00076800  00076890  00077040  0007704E  0007880  00000010E

SEGMENT TABLE

00076800  00000001 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  * ................................ *
    IDENTICAL
00076880  00000001 00000001 00000001 00000001   F0077040 00000001 00000001 00000001  FE  * ................0............... *
000768A0  00000001 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  * ................................ *
    IDENTICAL
00076960  F0077080 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  *0............................... *
00076980  00000001 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  * ................................ *
    IDENTICAL
00076BA0  00000001 00000001 00000001 00000001   040F7000 00000001 00000001 00000001  FE  * ................................ *
00076BC0  00000001 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  * ................................ *
00076BE0  00000001 00000001 00000001 00000001   00000001 00000001 00000001 00000001  FE  * ................................ *

PAGE TABLES

00077000  00040004 00040782 00040004 00040004   00040004 00040004 00040004 00040004  FE  * ................................ *
00077020  00040004 00040004 00040004 00040004   00040004 00040004 00040004 00040004  FE  * ................................ *
00077040  00040004 00040004 00040004 00040788   00040004 00040004 00040004 00040004  FE  * ................................ *
00077060  00040004 00040004 00040004 00040004   00040004 00040004 00040004 00040004  FE  * ................................ *
00077080  00040004 00040004 00040004 00040004   07900004 00040004 00040004 00040004  FE  * ................................ *
000770A0  00040004 00040004 00040004 00040004   00040004 00040004 00040004 00040004  FE  * ................................ *

TEST INSTRUCTION AND OPERAND PAGES

00078000  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  0C  * ................................ *
    IDENTICAL
00078740  00000000 00000000 00000000 00000000   40404040 40404040 00000000 00000000  0C  * ................................ *
00078760  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  0C  * ................................ *
    IDENTICAL
00078900  00000000 0000D207 20003000 00004400   40000000 00000000 00000000 00000000  FE  * ......K......................... *
00078920  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  FE  * ................................ *
    IDENTICAL
00079440  00000000 00000000 00000000 00000000   00000000 000000D7 C1E3C4C1 E3404000  FE  * .......................PATDAT... *
00079460  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  FE  * ................................ *
    IDENTICAL
000797E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  FE  * ................................ *
```

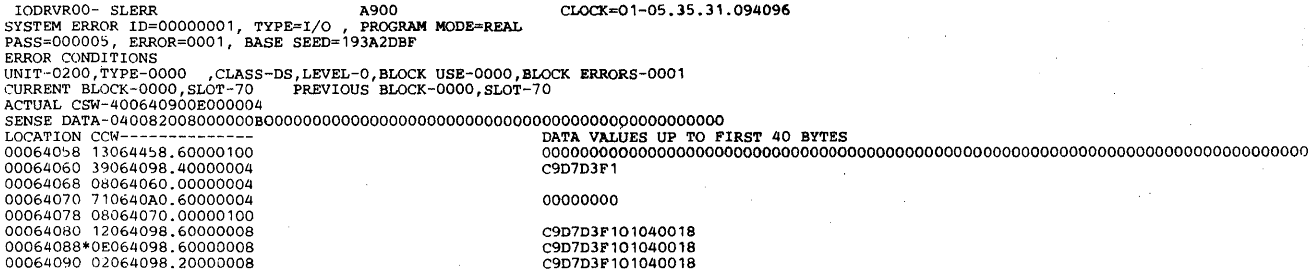
Figure 8. PATDAT Sample Error Printout

EPAGE1 Error Printout Sample

```
 EPAGE100- SLERR   E814           CLOCK=01-01.28.57.802912
SYSTEM ERROR ID=00000002, TYPE=INST, PROGRAM MODE=REAL
PASS=000051, ERROR=0002, BASE SEED=35CFC281

 EPAGE100-XPAGA0   E801           CLOCK=01-01.28.58.322592
ERROR TYPE INST OCCURRED IN THE ROUTINE TO TEST DCTP
ADDRESS OF PAGE USED FOR TEST IS 00000000
INSTRUCTION IN ERROR   B21C4000      DCTP  0(4)
PAGE KEY WAS NOT AS PREDICTED
PREDICTED AND ACTUAL RESULTS FOLLOW

PSW
  ACTUAL      -  000C100000084844
  PREDICTED   -  000C100000084844
  BEFORE TEST -  000C000000084840

GENERAL REGISTERS
  ACTUAL       0-00000000  1-00047CD4  2-00000004  3-00053000
  PREDICTED    0-00000000  1-00047CD4  2-00000004  3-00053000
  BEFORE TEST  0-00000000  1-00047CD4  2-00000004  3-00053000

  ACTUAL       4-00000000  5-22B9255A  6-000475A0  7-000479AC
  PREDICTED    4-00000000  5-22B9255A  6-000475A0  7-000479AC
  BEFORE TEST  4-00000000  5-22B9255A  6-000475A0  7-000479AC

  ACTUAL       8-00034000  9-00086040  A-00085040  B-00047B40
  PREDICTED    8-00034000  9-00086040  A-00085040  B-00047B40
  BEFORE TEST  8-00034000  9-00086040  A-00085040  B-00047B40

  ACTUAL       C-00084040  D-00047B60  E-00047B70  F-000455C0
  PREDICTED    C-00084040  D-00047B60  E-00047B70  F-000455C0
  BEFORE TEST  C-00084040  D-00047B60  E-00047B70  F-000455C0

CAPACITY COUNTERS
  ACTUAL     - 00000200 000001A3 000001A3 00000000
  PREDICTED  - 00000200 000001A3 000001A3 00000000
  BEFORE TEST- 00000200 000001A3 000001A3 00000000

PROGRAM CHECK    ACTUAL  PREDICTED
                 0006    0006

CONTROL REGISTERS
  ACTUAL       0-8000E460  1-00000000  2-FFFFFFFF  3-00000000
  PREDICTED    0-8000E460  1-00000000  2-FFFFFFFF  3-00000000
  BEFORE TEST  0-8000E460  1-00000000  2-FFFFFFFF  3-00000000

  ACTUAL       4-00000000  5-00000000  6-00000000  7-00000000
  PREDICTED    4-00000000  5-00000000  6-00000000  7-00000000
  BEFORE TEST  4-00000000  5-00000000  6-00000000  7-00000000

  ACTUAL       8-00000000  9-00000000  A-00000000  B-00000000
  PREDICTED    8-00000000  9-00000000  A-00000000  B-00000000
  BEFORE TEST  8-00000000  9-00000000  A-00000000  B-00000000

  ACTUAL       C-00000000  D-00000000  E-EF000000  F-00000200
  PREDICTED    C-00000000  D-00000000  E-EF000000  F-00000200
  BEFORE TEST  C-00000000  D-00000000  E-EF000000  F-00000200

PAGE STATUS  (00000000)- PREDICTED      ACTUAL         BEFORE TEST
  PAGE STATE             PAGE ZERO      PAGE ZERO      PAGE ZERO
  PAGE KEY FROM ISK      D7             06             06
  PAGE BITS FROM IPB     D7             06             06
  PAGE INDEX             0000           0000           0000
```

Figure 9. EPAGE1 Sample Error Printout

IODRVR Error Printout Sample



```
 IODRVR00- SLERR                     A900                 CLOCK=01-05.35.31.094096
SYSTEM ERROR ID=00000001, TYPE=I/O , PROGRAM MODE=REAL
PASS=000005, ERROR=0001, BASE SEED=193A2DBF
ERROR CONDITIONS
UNIT-0200,TYPE-0000  ,CLASS-DS,LEVEL-0,BLOCK USE-0000,BLOCK ERRORS-0001
CURRENT BLOCK-0000,SLOT-70    PREVIOUS BLOCK-0000,SLOT-70
ACTUAL CSW-400640900E000004
SENSE DATA-040082008000000B000000000000000000000000000000000000000000000000
LOCATION CCW--------------                              DATA VALUES UP TO FIRST 40 BYTES
00064058 13064458.60000100                              00000000000000000000000000000000000000000000000000000000000000000000000000000000
00064060 39064098.40000004                              C9D7D3F1
00064068 08064060.00000004
00064070 710640A0.60000004                              00000000
00064078 08064070.00000100
00064080 12064098.60000008                              C9D7D3F101040018
00064088*0E064098.60000008                              C9D7D3F101040018
00064090 02064098.20000008                              C9D7D3F101040018
```

Figure 10. IODRVR Sample Error Printout

BLANK PAGE

Utility Programs

The utility programs contained in the NST data set are:

- UPDATE NST general utility program.
- UDASDI NST disk pack format program (see Note).
- EMUFMT 2314 emulation format program for 3310 Disk Drives. (3310 is supported by the NST utility programs *only* in 2314 emulation mode.)

Note: *The NST utility programs do not support 3370 Disk Drives. Although IODRVR tests 3370 Disk Drives, NST can not reside on a 3370.*

Run Information

A summary of the NST utility program run procedures is shown under "Utilities Summary." For details, see the NST utility program descriptions.

Start

After IPLing NST, start the desired NST utility program by entering:

```
         ⎧UPDATE ⎫
START,ID-⎨UDASDI ⎬
         ⎩EMUFMT ⎭
```

Execute

The selected NST utility program begins execution when the NST MODIFY command is entered at the console keyboard. The function performed is determined by the MODIFY command parameters. The MODIFY command format is:

```
    ⎧UPDATE⎫
MOD-⎨UDASDI⎬00,param1,param2,...,paramN
    ⎩EMUFMT⎭
```

param1,param2,...,param*N* = MODIFY command parameters.

For parameter details, see "MODIFY Command Parameters for Utility Programs" and "Device Types for MODIFY Command."

For additional information about the NST utility programs and the functions they perform, see the program descriptions.

Terminate

To terminate UPDATE *after* it has completed the requested function(s), enter: MOD-UPDATE00,QUIT

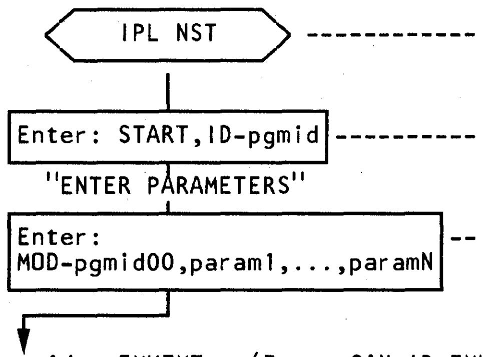
To terminate UPDATE *before* it has completed a requested function, enter: CAN,ID-UPDATE00

EMUFMT must be terminated when it has completed (or can be terminated before completion) by entering: CAN,ID-EMUFMT00.

UDASDI terminates automatically after completing the disk format, but can be terminated before completion by entering: CAN, ID-UDASDI00

Utilities Summary

The following flowchart shows a summary of the NST utility run procedures. For details, see the program descriptions: UPDATE, UDASDI, or EMUFMT.



See steps 1 through 3 of the "Run Procedures Summary" chart in the NST QUICK REFERENCE GUIDE.

Load the desired NST utility program. (pgmid = UPDATE, UDASDI, or EMUFMT)

Use the MODIFY command to perform the desired utility function. For details, see the UPDATE, UDASDI, or EMUFMT program descriptions.

pgmid = EMUFMT (Enter CAN,ID-EMUFMT to terminate.)

Emulation Format for 3310	MOD-EMUFMT00,UA-cuu,DT-3310,EDT-2314,E#-1 Required when preparing to copy the NST data set to a 3310 disk. Formats a 3310 disk for 2314 emulation. Do this before running UDASDI.

pgmid = UDASDI (Terminates automatically when formatting is completed.)

Format Disk	MOD-UDASDI00,UA-cuu,DT-yyyy,VOLID-volid [,CLIP] The disk must be formatted before the NST data set can be copied to it. To save time when the disk is known to have no defects, include the optional CLIP parameter. NOTE: On a 3310 disk, run EMUFMT before running UDASDI.

pgmid = UPDATE (Enter MOD-UPDATE00,QUIT to terminate.)

Copy to Tape	MOD-UPDATE00,UAO-cuu,DTO-xyyyy,UAN-cuu,DTN-xyyyy [,MSET-z] [,NMSG] If the MSET parameter is not specified, density is set to 1600 bpi (z = C). For 6250 bpi, z = D. For 8809 Tape Drive, z = 8.
Copy to Disk	MOD-UPDATE00,UAO-cuu,DTO-xyyyy,UAN-cuu,DTN-xyyyy,CYL-zz [,NMSG] Number of cylinders required (zz): 2314 = 20; 3330 = 12; 3340 = 32; 3350 = 10 NOTE: Format the disk before copying the data set to it - always required.
List	MOD-UPDATE00,UAO-cuu,DTO-xyyyy,LIST [,NCOM] If NCOM not specified, commands in CPROCs also list.
Change CPROC	See "Change (Modify) CPROC."
Replace Data	See "Replace Data."

MODIFY Command Parameters for Utility Programs

Parameter	Program	Description
CLIP	UDASDI	Allows formatting a disk with no surface analysis. (Saves time when the disk is known to have no defects.)
CD	UPDATE	Specifies that the corresponding UAU- entry is a card reader that accepts 2540 Reader commands. (Example: UAU-00C,CD)
CN	UPDATE	Specifies that additional console input is desired.
CYL-zz	UPDATE	Used to specify the required number of cylinders (in hex) when copying the NST data set to a disk. Replace zz with these values: 2314 = 20 3330 = 12 3340 = 32 3350 = 10
DAN DAO DAU	UPDATE	Disk address new, old, and update. Specifies that corresponding unit address (UA) entry is a 3330 disk. (Example: UAO-190,DAO)
DT-yyyy	UDASDI EMUFMT	Disk type. For a list of accepted disk types, see "Device Types for MODIFY Command." (Do not specify class.)
DTN-xyyyy DTO-xyyyy DTU-xyyyy	UPDATE	Device type new, old, and update. Specifies the device class and type for the corresponding UAN, UAO, or UAU entry. Replace xyyyy with the correct device class and type as shown under "Device Types for MODIFY Command." (Example: UAN-290,DTN-D334001)
E#-1 EDT-2314	EMUFMT	Required parameters when formatting a 3310 disk for 2314 emulation. (EDT = emulated device type)
IPL NIPL	UPDATE	IPL is the default setting and specifies that an IPL record be created. NIPL specifies no IPL record.
LIST	UPDATE	Specifies the list function.
MSET-z3	UPDATE	Sets tape density when copying the NST data set to tape. C3 = 1600 bpi, the default. D3 = 6250 bpi. 83 = 8809 Tape Drive.
MSG NMSG UMSG	UPDATE	MSG = Messages are listed or displayed. NMSG = No messages are listed or displayed. UMSG = Only UPDATE messages are listed or displayed.
NCOM PCOM	UPDATE	NCOM = CPROC commands are not listed (printed) when doing a LIST or COPY. PCOM = Default setting. CPROC commands are listed (printed) when doing a LIST or COPY.
PU	UPDATE	Specifies that the corresponding UAN- entry is a 2540 Card Punch.
TAN TAO TAU	UPDATE	Tape address new, old, and update. Specifies that the corresponding unit address (UA) entry is a 3420. (Example: UAO-280,TAO)
UAN-cuu UAO-cuu UAU-cuu	UPDATE	Unit address new, old, or update. Replace cuu with the channel and unit address.
UIP-cuu	UPDATE	Update in place. Applies to disk only. Allows making changes to the NST data set without creating a new data set. See "Update in Place (UIP)."

Device Types for MODIFY Command

To specify device type parameters (DTO, DTN, or DTU) in a MODIFY command, enter the device class and type codes as shown in the following chart.

Enter: $\left\{\begin{array}{l}\text{DTO}\\ \text{DTN}\\ \text{DTU}\end{array}\right\}\text{-xyyyy}$

x = Device class, the first character shown in the chart.

yyyy = Device type, the remaining characters shown in the chart. The fifth and sixth characters (when present) indicate model number. Include these characters when specifying yyyy.

The numbers enclosed in parenthesis () refer to the notes below the chart.

Disk	Tape	Reader	Punch
D230501	T2400	C1442	P1442
D230502	T2415	C2501	P2540P(4)
D2311	T3400	C2520	P2560
D2314	T3410	C2540 (3)	P3525
D3330 (1)	T3420 (2)	C2596	P5424
D333011	T8809	C3504	
D334001		C3505	
D334002			
D3350			

Notes:

1. *D3330 is the default disk type.* DAN *or* DAO *may be substituted for* DTN-xyyyy *or* DTO-xyyyy.
2. *T3420 is the default tape type.* TAN, TAO, *or* TAU *may be substituted for* DTN-xyyyy, DTO-xyyyy, *or* DTU-xyyyy.
3. *C2540 is the default card reader type.* CD *may be substituted for* DTU-xyyyy.
4. P2540P *is the default card punch type.* PU *may be substituted for* DTN-xyyyy.

UPDATE - Description and Functions

UPDATE is a program in the NST data set that performs all NST utility functions except for disk formatting. These functions include:

- Listing the contents of the NST data set
- Copying the NST data set to tape or disk
- Changing (modifying) CPROCs
- Replacing data or modules in an NST data set

Data Set Types

The NST data sets used when performing UPDATE functions are defined as old (O), new (N), or update (U).

Old (O)

The old NST data set is the one that is loaded during IPL.

When performing an UPDATE function, the address and device type of the old data set are defined by specifying `UAO-cuu,DTO-xyyyy` as MODIFY command parameters. `cuu` is the channel and unit address of the old data set and `xyyyy` is the class and device type.

New (N)

A new data set is created from an old data set (such as when copying the NST data set to tape or disk), or from an old data set on tape and an update data set (such as when adding a CPROC or replacing a module).

The address and device type of the new data set are defined by specifying `UAN-cuu,DTN-xyyyy` as MODIFY command parameters. `cuu` is the channel and unit address of the new data set and `xyyyy` is the class and device type.

Update (U)

An update data set consists of one or more NST CPROCs or modules. The update data set can be merged with an old data set to form a new data set. When the old data set resides on disk, update in place (UIP) may be used to add or replace modules directly on the disk. (For additional information, see "Update in Place (UIP).")

An update data set may reside on cards or tape - never on disk. The address and device type of the update data set are defined by specifying `UAU-cuu,DTU-xyyyy` as MODIFY command parameters. `cuu` is the channel and unit address of the update data set and `xyyyy` is the class and device type.

Update in Place (UIP)

Update in place (UIP) may be specified as a MODIFY command parameter when adding a modified CPROC or replacing a module directly to an old NST data set that resides on disk.

The UIP parameter may be used *only* when the data set is on disk. UIP allows making changes directly to the old data set, which eliminates the need for creating a new data set.

Enter the UIP parameter in this format: `UIP-cuu`, where `cuu` is the disk address. For details on the use of the UIP parameter, see "Change (Modify) CPROC" and "Replace Data."

List

This UPDATE function prints a list of the NST data set contents. Command procedure (CPROC), program, and module names (IDs) are printed. CPROC commands can also be printed when desired.

Note: *For information on displaying the commands in a CPROC without a full list, see "CPROC Service Aids."*

Preparation

1. Ensure that a printer is assigned (`AS,S,UA-cuu,DT-yyyy`) and ready. (`cuu` is the printer address and `yyyy` is the printer type.)
2. Enter: `START,ID-UPDATE`

Command Format

To list an *old* data set, enter:

```
                         (TAO        )
MOD-UPDATE00,UAO-cuu,{ DAO        },LIST[,NCOM]
                         (DTO-xyyyy)
```

(For an example, see the next page.)

To list an *update* data set (from card or tape), enter:

```
                         (TAU        )
MOD-UPDATE00,UAU-cuu,{ CD         },LIST[,NCOM]
                         (DTU-xyyyy)
```

CD — Card device. Indicates that the update data set is on a 2540 Card Reader.

DAO — Disk address old. Indicates that the old data set is on a 3330 Disk Drive.

DTO-xyyyy — Device type old. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

DTU-xyyyy — Device type update. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

LIST — Defines the list function.

NCOM — No commands. This is an optional parameter that specifies that CPROC commands are not to be listed. (Only the CPROC names will be listed, along with the other module names.)

TAO — Tape address old. Indicates that the old data set is on a 3420 Tape Drive.

TAU — Tape address update. Indicates that the update data set is on a 3420 Tape Drive.

UAO-cuu — Unit address old. Replace cuu with the address of the device that contains the old data set.

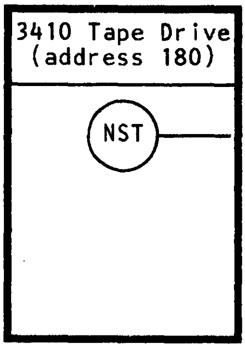
UAU-cuu — Unit address update. Replace cuu with the address of the device that contains the update data set.

Example

To list the old NST data set, including CPROC commands, mounted on a 3410 Tape Drive at address 180, enter:

```
MOD-UPDATE00,UAO-180,DTO-T3410,LIST
```

Old data set:



Listing:

```
START OF LISTING UA-180 DT-3410 DATASET NAME: NST 2.1
CESD *$$AAAR $$XXXX $$YYYY $$ZZZZ
*****************************************************
$$AAAA LEN=001DFC V/M=0001 DATE=083578296 CARDS=00072
$$ZHST                              (hhmmyrday)
#SXLNK
CESD CPRSC1 025 CP370A
*CP STA,ID-EPAGE1
*CP STA,ID-PGENXX,HALT
*CP MOD-PGENXX00,BUFSZ-4000
*CP TERM
 END
DFBA00
D34200
PATDAT
PGENXX
UDASDI
UPDATE
VCHCH0
V10170
V38950
XBGENC
XXYNUC
```

Copy to Tape

This UPDATE function copies the NST data set to tape from another tape or from disk. (To copy the NST data set to disk, see "Copy to Disk.")

Preparation

1. Ensure that the device that contains the NST data set that is to be copied is ready.
2. Ensure that a scratch tape is mounted (file-protect ring installed) on another tape drive and that the drive is ready.
3. Enter: `START,ID-UPDATE`

Command Format

To copy the NST data set to tape, enter:

```
                     ┌TAO      ┐         ┌TAN      ┐
MOD-UPDATE00,UAO-cuu,┤DAO      ├,UAN-cuu,┤         ├[,MSET-z3][,NMSG]
                     └DTO-xyyyy┘         └DTN-xyyyy┘
```

(For an example, see the next page.)

DAO Disk address old. Indicates that the old data set is on a 3330 Disk Drive.

DTN-xyyyy Device type new. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

DTO-xyyyy Device type old. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

MSET-z3 Mode set. This is an optional parameter that sets tape density for the new data set. When MSET is not specified, tape density is set to the default of 1600 bpi (MSET-C3). To set 6250 bpi, specify MSET-D3. When the new data set is to be written on an 8809 Tape Drive, specify MSET-83.

NMSG No messages listed or displayed. This is an optional parameter that prevents listing the NST data set as it is being copied.

TAN Tapė address new. Indicates that the new data set is on a 3420 Tape Drive.

TAO Tape address old. Indicates that the old data set is on a 3420 Tape Drive.

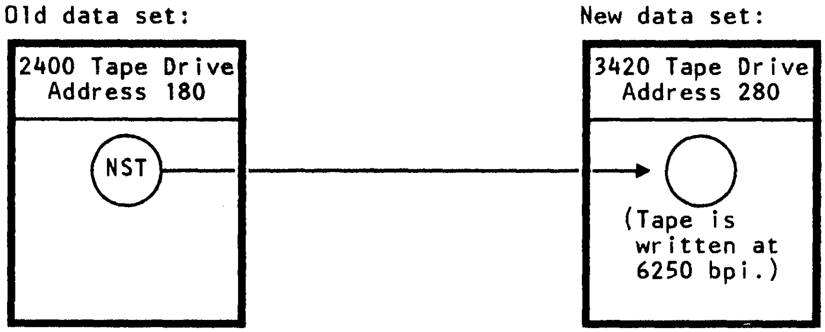
UAN-cuu Unit address new. Replace cuu with the address of the tape drive on which the scratch tape is mounted.

UAO-cuu Unit address old. Replace cuu with the address of the device that contains the old data set.

Example

To copy the NST data set from a 2400 Tape Drive at address 180 to a 3420 Tape Drive at address 280 with a tape density of 6250 bpi, enter:

```
MOD-UPDATE00,UAO-180,DTO-T2400,UAN-280,TAN,MSET-D3
```



Copy to Disk

This UPDATE function copies the NST data set to a customer disk pack from tape or from another disk pack. (To copy the NST data set to tape, see "Copy to Tape.")

Caution: Use a scratch disk pack. *Do not use a CE pack - CE cylinders will be destroyed.* After the NST data set has been copied, the remainder of the disk pack is available for customer use, but it may not be used for SYSRES.

Preparation

1. Format the disk pack that the NST data set is to be copied to. (See Note 1.)
2. Ensure that the device that contains the old data set is ready.
3. Ensure that the drive that contains the formatted disk is ready.
4. Enter: `START,ID-UPDATE`

Command Format

To copy the NST data set to a formatted disk, enter:

```
                         (TAO        )          (DAN        )
MOD-UPDATE00,UAO-cuu,{ DAO          },UAN-cuu,{             },CYL-zz[,NMSG]
                         (DTO-xyyyy  )          (DTN-xyyyy  )
```

For an example of the copy to disk command, see the next page. If a message indicating that the parameters were not entered correctly is displayed, see Note 2.

`CYL-zz` Replace `zz` with the number of cylinders required for the new data set:
2314 = 20; 3330 = 12; 3340 = 32; 3350 = 10

`DAN` Disk address new. Indicates that the new data set is on a 3330 Disk Drive.

`DAO` Disk address old. Indicates that the old data set is on a 3330 Disk Drive.

`DTN-xyyyy` Device type new. Replace `xyyyy` with the device class and type as shown under "Device Types for MODIFY Command."

`DTO-xyyyy` Device type old. Replace `xyyyy` with the device class and type as shown under "Device Types for MODIFY Command."

`NMSG` No messages listed or displayed. This is an optional parameter that prevents listing the NST data set as it is copied to the disk.

`TAO` Tape address old. Indicates that the old data set is on a 3420 Tape Drive.

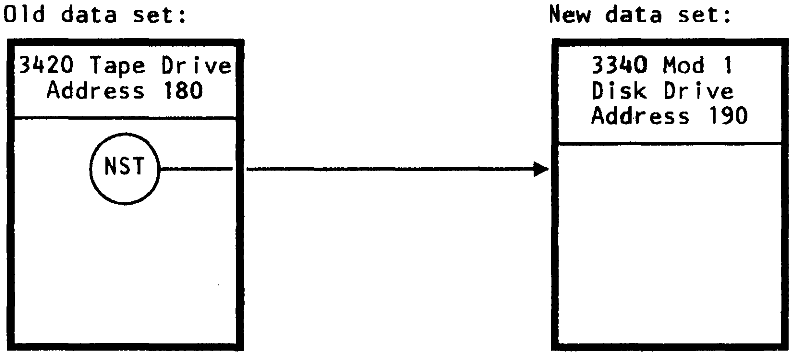
`UAN-cuu` Unit address new. Replace `cuu` with the address of the disk drive on which the new data set is to be copied.

`UAO-cuu` Unit address old. Replace `cuu` with the address of the device that contains the old data set.

Example

To copy an NST data set from a 3420 Tape Drive at address 180 to a 3340 Mod 1 Disk Drive at address 290 with no messages listed or displayed, enter:

```
MOD-UPDATE00,UAO-180,TAO,UAN-290,DTN-D334001,CYL-32,NMSG
```



Notes:

1. *The disk formatting programs contained in the NST data set (UDASDI and EMUFMT) can be used when alternate track assignments are not required. For additional information, see the UDASDI and EMUFMT program descriptions. Use a standard IBM formatting utility program when alternate track assignments are required. For reference, see "Standard IBM Utility Programs."*
2. *If this message is displayed:*

   ERROR - NMDSN PARM OR UAN PARM OR UIP PARM OR CYL PARM INCORRECT

   a. *Verify that all parameters are entered correctly.*

   b. *If the problem remains, format the disk with UDASDI again and then start UPDATE again to copy the NST data set to disk.*

Change (Modify) CPROC

Note: *Change CPROCs under the direction of engineering or the Field Support Center (FSC) only.*

Two UPDATE functions are required to change a CPROC:

- Punch the CPROC to cards. This allows making the changes to the cards.
- Add the changed CPROC to the NST data set.
  - When the old data set is on disk, the changed CPROC is added to the same data set using the update-in-place (UIP) parameter.
  - When the old data set is on tape, a new data set is created on another tape from the old data set and the update data set (changed CPROC).

Punching the CPROC

1. Load (IPL) NST. For details, see steps 1 through 3 of the "Run Procedures Summary" chart in the *NST Quick Reference Guide* (D99-NSTQR).
2. Ready a card punch with blank cards.
3. Enter: `START,ID-UPDATE`
   When the UPDATE program is loaded, this message is displayed:

   ```
   ENTER UPDATE PARAMATERS, OR QUIT
   ```
4. Enter:

```
                          (DAO        )           (PU         )
MOD-UPDATE00,UAO-cuu,    < TAO         >,UAN-cuu,<             >,CN
                          (DTO-xyyyy  )           (DTN-Pyyyy  )
```

   This command defines the old data set and card punch devices and specifies that additional console input will follow. (For parameter definitions, see the next page.)

5. Enter (when requested): `MOD-UPDATE00,UP-'ƀADDƀCPcpid'`

   ƀ = blank (space). Replace `cpid` with the name of the CPROC that is to be punched. For example, to punch CP370A, replace `cpid` with 370A.

6. Enter (when requested): `MOD-UPDATE00,UP-'ƀEOJ'`

   ƀ = blank (space).

Changing the CPROC Data

Make the desired changes to the cards containing the CPROC. (Obtain this information from engineering or the FSC.) Format the data as it is punched in the cards and keep the cards in the original sequence.

Proceed to "Adding the CPROC to Disk" *or* "Adding the CPROC to Tape."

Adding the CPROC to Disk

This procedure adds the changed CPROC directly to the old data set using the update in place (UIP) parameter.

After punching the CPROC, the message "ENTER UPDATE PARAMETERS, OR QUIT" is displayed. Proceed as follows:

1. Ready the update device: a card reader with the changed CPROC cards.
2. Enter:

```
                                  (CD        )
MOD-UPDATE00,UIP-cuu,UAU-cuu,    <            >
                                  (DTU-xyyyy )
```

Adding the CPROC to Tape

This procedure creates a new data set (on another tape) from the old data set and the update data set (changed CPROC).

After punching the CPROC, the message "ENTER UPDATE PARAMETERS, OR QUIT" is displayed. Proceed as follows:

1. Ready the update device: a card reader with the changed CPROC cards.
2. Ready the new data set device: a tape drive with a scratch tape mounted.
3. Enter: MOD-UPDATE00,

$$\text{UAO-cuu,}\left\{\begin{matrix}\text{TAO}\\ \text{DTO-xyyyy}\end{matrix}\right\}\text{,UAN-cuu,}\left\{\begin{matrix}\text{TAN}\\ \text{DTN-xyyyy}\end{matrix}\right\}\text{,UAU-cuu,}\left\{\begin{matrix}\text{CD}\\ \text{DTU-xyyyy}\end{matrix}\right\}\text{[,UMSG]}$$

Parameter Definitions

CD Indicates that the update device is a 2540 Card Reader.

CN Console. Allows adding additional input via the console.

DAO Disk address old. Indicates that the old data set is on a 3330 Disk Drive.

DTN-xyyyy Device type new. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

DTO-xyyyy Device type old. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

DTU-xyyyy Device type update. Replace xyyyy with the device class and type as shown under "Device Types for MODIFY Command."

PU Card punch. Indicates that the card punch is a 2540.

TAN Tape address new. Indicates that the new data set is on a 3420 Tape Drive.

TAO Tape address old. Indicates that the old data set is on a 3420 Tape Drive.

UAN-cuu Unit address new. Replace cuu with the address of the card punch when punching the CPROC and with the address of the scratch tape when adding the changed CPROC to a new data set.

UAO-cuu Unit address old. Replace cuu with the address of the device that contains the old NST data set.

UAU-cuu Unit address update. Replace cuu with the address of the device that contains the update data set.

UIP-cuu Update in place. Allows adding the CPROC directly on the disk that contains the old data set. Replace cuu with the disk address.

UMSG Update messages. This is an optional parameter that allows only update messages to be printed. When not specified, all messages print.

Replace Data

Note: *Replace data in the NST data set under the direction of engineering or the Field Support Center (FSC) only.*

Replace data is an UPDATE program function that allows changing data in NST modules. Three steps are required:

- Identify the device(s) that the data set(s) are mounted on.
  - When on disk, data is changed on the old data set using the update-in-place (UIP) parameter.
  - When on tape, a new data set is created (on another tape) from the old data set and the new data.
- Identify the module that is to be modified.
- Specify the new data and data address.

Preparation

1. Ensure that NST is loaded. For details, see steps 1 through 3 of the "Run Procedures Summary" chart in the *NST Quick Reference Guide* (D99-NSTQR).
2. Enter: `START,ID-UPDATE`
   When the UPDATE program is loaded, this message is displayed:

   ```
   ENTER UPDATE PARAMETERS, OR QUIT
   ```

3. Proceed to "Identifying the Old Data Set on Disk" *or* "Identifying the Old and New Data Sets on Tape."

Identifying the Old Data Set on Disk

To identify the old data set on disk and indicate that additional console input is required, enter:

```
MOD-UPDATE00,UIP-cuu,CN
```

For parameter definitions, see the next page. Proceed to "Identifying the Module that is to be Modified."

Identifying the Old and New Data Sets on Tape

Two tapes are required: the tape containing the old data set, and a scratch tape that the new data set will be written on. Ready a tape drive with a scratch tape mounted and enter:

```
                      ⎧TAO       ⎫         ⎧TAN       ⎫
MOD-UPDATE00,UAO-cuu, ⎨          ⎬,UAN-cuu,⎨          ⎬,CN[,UMSG]
                      ⎩DTO-xyyyy ⎭         ⎩DTN-xyyyy ⎭
```

For parameter definitions, see the next page. Proceed to "Identifying the Module that is to be Modified."

Identifying the Module that is to be Modified

After identifying the data set(s), this message is displayed: "ENTER UPDATE PARAMETERS, OR QUIT." Enter:
`MOD-UPDATE00,UP-'RREPb̸xxxxxxxx'`

b̸ = blank (space).
xxxxxxxx = the name of the module to be modified. Eight characters are required; pad with blanks to the right as needed.

Proceed to "Specifying the New Data and Data Address."

Specifying the New Data and Data Address

After identifying the module that is to be modified, this message is displayed: "ENTER UPDATE PARAMETERS , OR QUIT."

1. Enter:

```
MOD-UPDATE00,UP-'ƀREPƀƀyyyyyyƀƀƀƀzzzz'
```

ƀ Blank (space). Enter as shown.

yyyyyy Six-character data address.

zzzz Data to be entered at the specified address. If more than two bytes, enter in two-byte groups separated by a comma (,).

2. Enter (when requested):

```
              ⎧UP-'ƀEOF'⎫
MOD-UPDATE00, ⎨         ⎬
              ⎩UP-'ƀEOJ'⎭
```

UP-'ƀEOF' Enter this parameter when changing a module on tape.

UP-'ƀEOJ' Enter this parameter when changing a module on disk.

Parameter Definitions

CN Console. Allows entering instructions and data from the console keyboard.

DTN-xyyyy Device type new. Replace xyyyy with the device class and type codes of the new data set as shown under "Device Types for MODIFY Command."

DTO-xyyyy Device type old. Replace xyyyy with the device class and type of the old data set as shown under "Device Types for MODIFY Command."

TAN Tape address new. Indicates that the new data set is to be created on a 3420 Tape Drive.

TAO Tape address old. Indicates that the old data set is on a 3420 Tape Drive.

UAN-cuu Unit address new. Replace cuu with the address of the device that contains the new data set.

UAO-cuu Unit address old. Replace cuu with the address of the device that contains the old data set.

UIP-cuu Update in place. Allows replacing the module directly on the disk that contains the old data set. Replace cuu with the disk address.

UMSG Update messages. This is an optional parameter that allows only update messages to be printed. When not specified, all messages print.

UDASDI - Description and Function

UDASDI is a disk-formatting utility program in the NST data. All disks *must* be formatted before the NST data set can be copied to it, including disks that already contain the NST data. (Run EMUFMT on a 3310 disk before running UDASDI - see Note 1).

Caution: Use a scratch disk pack. *Do not use a CE pack - CE cylinders will be destroyed.* After the NST data set has been copied, the remainder of the disk pack is available for customer use, but it may *not* be used for SYSRES.

Formatting under UDASDI validates all home address and record 0 values, and writes a volume label and VTOC on the disk. Defective tracks are flagged, but alternate tracks are *not* assigned.

Use a standard IBM formatting utility program when alternate track assignments are required. For a list of standard IBM formatting utility programs, see Appendix B.

To format a disk using UDASDI:

1. Load the UDASDI program:
   Enter: `START,ID-UDASDI`
2. Ensure that the drive containing the disk pack that is to be formatted is ready.
3. Enter:

```
MOD-UDASDI00,UA-cuu,DT-yyyy,VOLID-volid[,CLIP]
```

`UA-cuu` Unit address. Replace `cuu` with the address of the disk to be formatted.

`DT-yyyy` Disk type. Replace `yyyy` with the disk type as shown under "Device Types for MODIFY Command." (Example: `DT-333011`)

`VOLID-volid` Volume ID. (Example: `VOLID-NSTWRK`) See Note 2.

`CLIP` This is an optional parameter that allows formatting a disk with no surface analysis (saves time when disk is known to have no defects).

UDASDI terminates automatically when formatting is completed. To copy the NST data set to the formatted disk, see "Copy to Disk."

Notes:

1. *NST can reside on a fixed-block architecture (FBA) disk only when the disk has been formatted to emulate a count-key-data (CKD) disk. Therefore, NST can reside on a 3310 disk only when it has been formatted for 2314 emulation. To format a 3310 disk for 2314 emulation, run EMUFMT before running UDASDI.*
2. *When the specified volume ID (*`volid`*) is not the same as the volume ID that has previously been written on the disk, UDASDI requests* "`ENTER RESUME TO CONTINUE.`" *This allows manually verifying that the desired disk is mounted. To continue, enter:* `RES,ID-UDASDI00`
   To cancel UDASDI, enter: `CAN,ID-UDASDI00`

EMUFMT - Description and Function

EMUFMT is an NST utility program that formats 3310 disks (attached to a 4331 Processor) for 2314 emulation. 3310 disks *must* run under 2314 emulation because NST can not reside on a fixed block architecture (FBA) device. After running EMUFMT, run UDASDI to create the VTOC and then perform a copy-to-disk function with the UPDATE program.

To format a 3310 disk for 2314 emulation:

1. Display the program load screen and select these emulation entries:
   a. NUMBER OF EMU BUFFERS = 1
   b. EMULATED DEVICE = S (2314)
   c. FIRST HOST ADDRESS = Address of 3310 to be formatted. (See note 1.)
   d. EMU ONLY MODE = N (no)
2. Select the remaining required entries from the program load screen and press ENTER to load NST.
3. Complete the load and select the desired NST options as described in steps 1 through 3 of the "Run Procedures Summary" chart in the *NST Quick Reference Guide* (D99-NSTQR).
4. Enter: `START,ID-EMUFMT`
5. Ensure that the 3310 Disk Drive that contains the disk pack to be formatted is ready.
6. Enter:

   ```
   MOD-EMUFMT00,UA-cuu,DT-3310,EDT-2314,E#-1
   ```

   `UA-cuu` = Address of the 3310 disk to be formatted.
   Enter the other parameters exactly as shown.

7. Cancel the EMUFMT program when formatting has completed. Enter:

   ```
   CAN,ID-EMUFMT00
   ```

   (See note 2.)

8. Run UDASDI to create the VTOC before attempting to copy the NST data set to the disk. Specify 2314 as the device type and use the emulated address as described in the following notes.

Notes:

1. *After running EMUFMT, the 3310 at the specified address appears to the system to be a 2314 at an emulated address. The emulated 2314 address is x90 (x is the channel number) when the FIRST HOST ADDRESS specified on the program load screen is the address of the 3310 that was formatted for emulation (as directed by step 1c). For example, if the 3310 that was formatted for 2314 emulation is at address 242 (and 242 was specified as the FIRST HOST ADDRESS on the program load screen), then the 3310 at address 242 appears to the system to be a 2314 at address 290 after running EMUFMT.*

2. *For additional information on emulated 2314 addresses, see the S-DIAG section of the 3310 Maintenance Information Manual (MIM).*

Appendix A. APAR Information for NST

For NST program maintenance, submit an Authorized Program Analysis Report (APAR). For problem correction, submit the APAR, with its documentation and dump, to the appropriate change team. Refer to the list of component IDs when submitting an APAR.

Before submitting an APAR, obtain a prescreening number from the Diagnostic Support Center. Note this number on the APAR in the appropriate field (USA only).

If an APAR is to be submitted for NST, please include the following: All printer and/or console output. Include a memory dump on a mini-reel, if possible. Otherwise a stand-alone memory dump for the first meg of storage.

To get a memory dump of NST on a tape, do the following:

1. Stop the system.
2. Record the current PSW (required information for the APAR).
3. Alter main storage at 806 to the 2 byte device address of a tape drive that has a scratch tape mounted with a ring. A mini-reel (300 ft.) can be used.
4. Alter the PSW to 000C 0000 0000 0800.
5. Start the system.

This will copy memory to the tape, which can then create the exact conditions that were present when the tape was created.

Mail APARs for NST to:

IBM Corporation
NST Coordinator
Department B44
P.O. Box 390
Poughkeepsie, N.Y. 12602

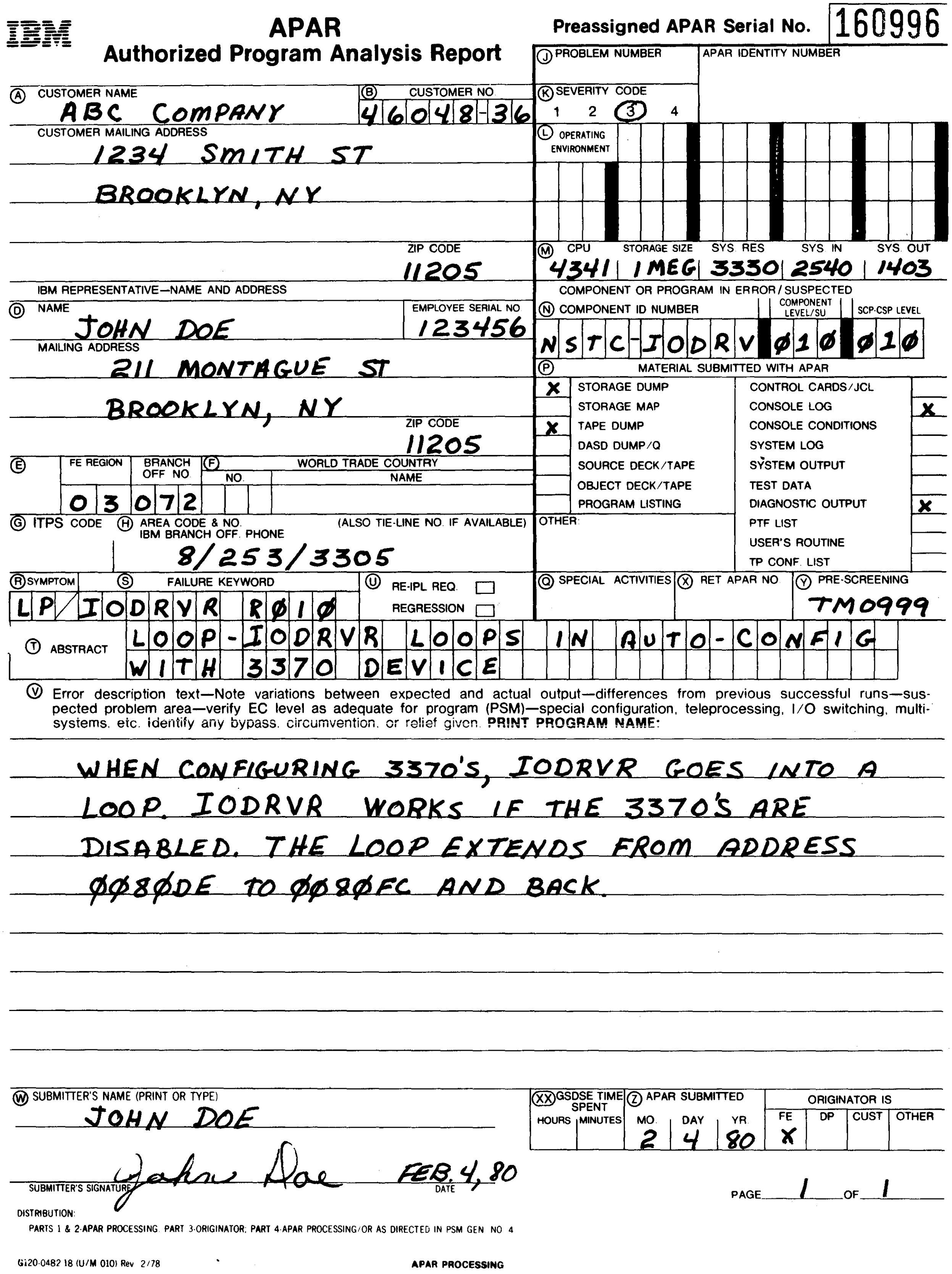
Refer to the *Diagnostic Reference Summary,* Order No. SY25-0512, for the details of submitting an APAR.

Component IDs

The following lists the component IDs to be used in submitting an APAR. Be sure to enter the proper ID when submitting an APAR.

NSTCNSTCP	NSTCP	Control Program
NSTCIODRV	IODRVR	I/O Driver
NSTCPGENX	PGENXX	Processor Test
NSTCPATDT	PATDAT	Dynamic Address Translation Test
NSTCIOCMD	I/O Command Blocks	All "V" Modules
NSTCUPDTE	UPDATE	Utility Program
NSTCEPAGE	EPAGE1	4300 VSE Mode Processor Test
NSTCCPROC	CPROC	Command Procedures
NSTCEMFMT	EMUFMT	2314 Emulation Utility
NSTCUDASD	UDASDI	Initialize Disk Utility

The following is an example of an APAR, correctly filled-in before submitting:



IBM

APAR
Authorized Program Analysis Report

Preassigned APAR Serial No. 160996

(J) PROBLEM NUMBER

APAR IDENTITY NUMBER

(A) CUSTOMER NAME
ABC COMPANY

(B) CUSTOMER NO
46048-36

(K) SEVERITY CODE
1 2 [✓ 3] 4

CUSTOMER MAILING ADDRESS
1234 SMITH ST
BROOKLYN, NY

ZIP CODE
11205

(L) OPERATING ENVIRONMENT

(M) CPU	STORAGE SIZE	SYS RES	SYS IN	SYS OUT
4341	1MEG	3330	2540	1403

IBM REPRESENTATIVE—NAME AND ADDRESS

(D) NAME
JOHN DOE

EMPLOYEE SERIAL NO
123456

MAILING ADDRESS
211 MONTAGUE ST
BROOKLYN, NY

ZIP CODE
11205

COMPONENT OR PROGRAM IN ERROR/SUSPECTED

(N) COMPONENT ID NUMBER	COMPONENT LEVEL/SU	SCP-CSP LEVEL
NSTC-IODRV	Ø1Ø	Ø1Ø

(E) FE REGION	BRANCH OFF NO	(F) WORLD TRADE COUNTRY NO.	NAME	
03	072			

(P) MATERIAL SUBMITTED WITH APAR

X	STORAGE DUMP	CONTROL CARDS/JCL	
	STORAGE MAP	CONSOLE LOG	X
X	TAPE DUMP	CONSOLE CONDITIONS	
	DASD DUMP/Q	SYSTEM LOG	
	SOURCE DECK/TAPE	SYSTEM OUTPUT	
	OBJECT DECK/TAPE	TEST DATA	
	PROGRAM LISTING	DIAGNOSTIC OUTPUT	X
	OTHER:	PTF LIST	
		USER'S ROUTINE	
		TP CONF. LIST	

(G) ITPS CODE (H) AREA CODE & NO. IBM BRANCH OFF. PHONE (ALSO TIE-LINE NO. IF AVAILABLE)
8/253/3305

(R) SYMPTOM
LP/

(S) FAILURE KEYWORD
IODRVR RØ1Ø

(U) RE-IPL REQ. □
REGRESSION □

(Q) SPECIAL ACTIVITIES

(X) RET APAR NO

(Y) PRE-SCREENING
TM0999

(T) ABSTRACT
LOOP-IODRVR LOOPS IN AUTO-CONFIG
WITH 3370 DEVICE

(V) Error description text—Note variations between expected and actual output—differences from previous successful runs—suspected problem area—verify EC level as adequate for program (PSM)—special configuration, teleprocessing, I/O switching, multi-systems, etc. identify any bypass, circumvention, or relief given. **PRINT PROGRAM NAME:**

WHEN CONFIGURING 3370'S, IODRVR GOES INTO A LOOP. IODRVR WORKS IF THE 3370'S ARE DISABLED. THE LOOP EXTENDS FROM ADDRESS ØØ8ØDE TO ØØ8ØFC AND BACK.

(W) SUBMITTER'S NAME (PRINT OR TYPE)
JOHN DOE

(XX) GSDSE TIME SPENT		(Z) APAR SUBMITTED			ORIGINATOR IS			
HOURS	MINUTES	MO.	DAY	YR.	FE	DP	CUST	OTHER
		2	4	80	X			

SUBMITTER'S SIGNATURE John Doe
DATE FEB. 4, 80

PAGE 1 OF 1

DISTRIBUTION:
PARTS 1 & 2-APAR PROCESSING PART 3-ORIGINATOR; PART 4-APAR PROCESSING/OR AS DIRECTED IN PSM GEN NO 4

GI20-0482 18 (U/M 010) Rev 2/78

APAR PROCESSING

Appendix B. Installing NST

The NST tape received from PID is IPLable and contains all the programs required to run NST on the processor to which it was sent. It may be desirable, however, to copy the NST data set to another tape (for back-up), or to a disk.

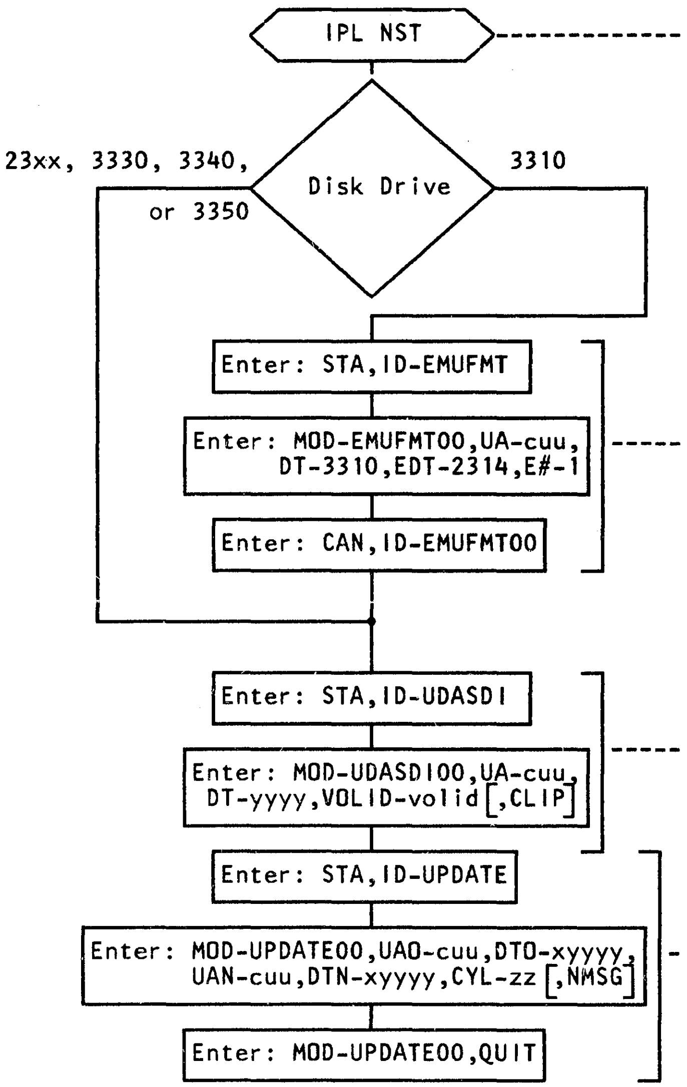
To copy the NST data set to another tape, load the UPDATE program and enter the MODIFY command with the correct parameters. For details, see "Copy to Tape."

To copy the NST data set to a disk pack, the disk pack must first be formatted. The "Installation Flowchart" that follows shows how to format a disk pack and copy the NST data set to the formatted pack.

Installation Flowchart

The following flowchart shows how to format a disk pack and copy the NST data set to the formatted pack.

Caution: Use a scratch disk pack. *Do not use a CE pack - CE cylinders will be destroyed.* After the NST data set has been copied, the remainder of the disk pack is available for customer use, but it may not be used for SYSRES.



```
Enter: STA,ID-EMUFMT

Enter: MOD-EMUFMT00,UA-cuu,
       DT-3310,EDT-2314,E#-1

Enter: CAN,ID-EMUFMT00

Enter: STA,ID-UDASDI

Enter: MOD-UDASDI00,UA-cuu,
 DT-yyyy,VOLID-volid[,CLIP]

Enter: STA,ID-UPDATE

Enter: MOD-UPDATE00,UAO-cuu,DTO-xyyyy,
       UAN-cuu,DTN-xyyyy,CYL-zz[,NMSG]

Enter: MOD-UPDATE00,QUIT
```

See steps 1 through 3 of the "Run Procedures Summary" chart in the NST QUICK REFERENCE GUIDE.

NOTE: IPL a 4331 in emulation mode when preparing to copy the NST data set to a 3310 Disk Drive. For details, see "EMUFMT - Description and Function."

These steps prepare a 3310 for 2314 emulation. For details, see "EMUFMT - Description and Function."

These steps format the disk and write the VTOC. For additional information, see "UDASDI - Description and Function."

These steps copy the NST data set to the formatted disk. Replace zz as follows:
2314 = 20; 3330 = 12; 3340 = 32; 3350 = 10
For details, see "Copy to Disk."
NOTE: 3310 must be emulating 2314.

Standard IBM Utility Programs

Formatting a disk with a standard IBM utility program allows for assigning alternate tracks when defective tracks exist. For information on formatting a disk using the standard IBM utility programs, see the documents indicated in the following chart.

Operating System	Reference
DOS/VSE	DOS/VSE SYSTEM UTILITIES (GC33-5381). See chapters: "Assign Alternate Track" and "Initialize Disk."
OS/VS	OS/VS UTILITIES (GC35-0005). The utilities used are IBCDASDI and IEHDASDR.
VM/370	IBM VIRTUAL MACHINE FACILITY/370; CMS USER'S GUIDE (GC20-1819). See "IBCDASDI."

Appendix C. Glossary

BUFSZ. The buffer size that defines, in hex, the number of bytes of storage to be used for program workspace.

clip. A parameter specified when using UDASDI to format a disk pack. When specified, surface analysis is bypassed to save time when formatting a disk pack known to be good.

CN. Console input parameter that selectively adds or deletes programs.

CPROC. A command procedure. Use of a CPROC permits NST to be invoked with a minimum of operator intervention. Intervention is minimized because default tests and options are invoked when the CPROC is selected.

DAN (disk address new). A parameter specified when using UPDATE. It specifies to the program a device class = disk, and a device type = 3330. Used when the device is the new (output) device.

DAO (disk address old). A parameter specified when using UPDATE. It specifies to the program a device class = disk, and a device type = 3330. Used when the device is the old (input) device.

delimiter. A character that groups or separates words or values in a line of input.

DTN (device type new). A keyword that is specified when using UPDATE, and specifies to the program a new master class and device type.

DTO (device type old). A keyword that is specified when using UPDATE, and specifies to the program an old master class and device type.

DTU (device type update). A keyword that is specified when using UPDATE, and specifies to the program the update unit class and device type.

EMUFMT. A utility provided on the NST media that permits the 3310 DASD to emulate a 2314 disk pack.

EPAGE1. The paging test, run under the NST control program, for processors running in VSE mode.

format. A function that validates all home address and record 0 values on a disk pack, and writes a volume label and VTOC.

IODRVR. An I/O driver program that can be invoked by an NST user. Tests (similar to ST370) are executed that exercise all configured and ready I/O.

keyword. Symbol(s) that identify a parameter; part of a command operand that consists of a specific character string.

PATDAT. A dynamic address translation (DAT) test that runs under the NST control program and tests the DAT feature of processors running in 370 (not VSE) mode.

PGENXX. A processor test that runs under the NST control program. All instructions are tested in a random series, an interrupt is generated, and the actual results are compared to expected (predicted) results.

seed. A value associated with a particular series of random instructions. This value can be used to re-execute the series of instructions, and is an aid in isolating a problem. Because the series of instructions is in reality psuedo-random, re-execution is only possible while no change has been made to the configuration being tested. Executing NST with the same seed value on a "similar" system results in a different series of instructions being executed.

TAN (tape address new). A parameter specified when using UPDATE, and specifies to the program a device class = tape, and a device type = 3420. Used when the device is the new (output) device.

TAO (tape address old). A parameter specified when using UPDATE, and specifies to the program a device class = tape, and a device type = 3420. Used when the device is the old (input) device.

TODC. Time-of-day clock; set during IPL.

UAN (unit address new). A parameter specified when using UPDATE, and specifies the new master unit address.

UAO (unit address old). A parameter specified when using UPDATE, and specifies the old master unit address.

UAT (unit availability table). A list of configured devices and their testing levels, provided when running IODRVR.

UAU (unit address update). A parameter specified when using UPDATE, and specifies the update unit address.

UDASDI. A utility, contained on the NST media, that formats a disk pack before installing or GENerating. It does not assign alternate tracks.

UIP (update in place). A parameter specified when using UPDATE, and may be used in place of UAO, DAO, UAN and DAN parameters.

UMSG. A parameter that specifies only UPDATE messages are to be listed or displayed.

UPDATE. A utility contained on the NST media that is used for various functions, such as: copying, listing, GEN, and replace.

Index