

Ten X 99™

Reference Manual



Box
10/10/10

COPYRIGHT 1983, 1984, 1985
TEN X TECHNOLOGY, INC

All rights reserved.

The information and/or drawings set forth in this document and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques, or apparatus described herein are the exclusive property of Ten X Technology, Inc.

No disclosure of information or drawings shall be made to any other person or organization without the prior consent of Ten X Technology, Inc.

Release 3.1.1

Ten X 99 is a trademark of Ten X Technology, Inc.

TILINE is a trademark of Texas Instruments Incorporated.

FEDERAL COMMUNICATIONS COMMISSION
RADIO FREQUENCY INTERFERENCE STATEMENT

This equipment complies with the requirements for a CLASS A Computing Device in FCC Rules Part 15 Subpart J. Operation of this device in a residential area may cause harmful interference requiring the user to take whatever steps may be necessary to correct the interference.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment on and off, the user is encouraged to try to correct the interference by one or more of the following measures:

- * Reorient the receiving antenna.
- * Relocate the equipment with respect to the receiver.
- * Move the equipment away from the receiver.
- * Plug the equipment into a different outlet so that equipment and receiver are on different branch circuits.

If necessary, consult your dealer service representative or an experienced radio/television technician for additional suggestions. Additional information is available from the Federal Communications Commission entitled:

How to Identify and Resolve Radio-TV
Interference Problems

This booklet is available from the U.S. Government Printing Office: Washington DC 20402; Stock No. 004-000-00345-4.

The manufacturer is not responsible for any radio or television interference caused by unauthorized modifications to this equipment. It is the responsibility of the user to correct such interference.

PREFACE

You will find your Ten X 99 COBOL Accelerator easy to install in your TI 990 minicomputer. You can use it with COBOL Release 3.3, 3.4, or 3.5 executing under DX10 3.3 through 3.7 or under DNOS 1.1, 1.2, or 1.3.

This manual describes installation and operation of the Accelerator and accompanying software. It provides programming tips to help you utilize your Ten X 99 to its greatest advantage.

The first two sections are directed to the person who installs your Ten X 99. SECTION 1 is for the DX10 user. SECTION 2 is for the DNOS user.

An Installation Checklist has been provided to ensure complete and proper installation of your Ten X 99, see Appendix G.

Each section includes step-by-step procedures for installation of both the hardware and software.

SECTION 3 is for the COBOL programmer. Your existing COBOL programs DO NOT NEED ANY CHANGES to run with the Accelerator. By following the tips in this section you can optimize your new programs and improve some of your existing ones.

SECTION 4 describes the Ten X command. This command is used by system operators to display the TILINE addresses and interrupts defined for the Ten X 99 Accelerators and to turn Ten X on and off. It is used by system analysts to gather performance data.

Should a problem arise, you may use the troubleshooting information in SECTION 5. It lists the most common problems and their causes and tells you how to recover from them. If a problem persists, please contact the dealer from whom you purchased the Accelerator. Your dealer has current problem solving information from Ten X. Once this support has been exhausted call the Ten X Hotline at (512) 346-0477 or use the out of Texas watts line 800-922-9050.

The appendices contain summaries of the Ten X commands, a list of error messages, COBOL POP command descriptions, and TILINE information for the Accelerator.

This manual does not apply directly to Ten X 99 COBOL Accelerator releases prior to Release 3.0. If you have an earlier release contact your local distributor for an update.

We want you to get the most from your Ten X 99 COBOL Accelerator and we hope the information you need is in this manual. If you should have any questions please feel free to contact us at any time. Thank you for choosing the Ten X 99. We sincerely hope that it meets your performance expectations.

Ten X Technology, Inc.
3933 Steck Ave. Suite B114
Austin, Texas 78759
(512) 346-8360

TABLE OF CONTENTS

SECTION 1 INSTALLATION FOR DX10

1.1	Copying Your Software.	1
1.1.1	Diskette.	1
1.1.2	Magnetic Tape.	2
1.2	Installing Ten X SCI Commands.	3
1.3	Choosing A Slot.	3
1.4	Pre-SYSGEN -- Defining Ten X System Interface.	4
1.4.1	Pre-SYSGEN for DX10 versions 3.3 - 3.7	4
1.5	SYSGEN -- Defining The System.	7
1.5.1	DX10 Release 3.3	7
1.5.2	DX10 Release 3.4	8
1.5.3	DX10 Release 3.5	9
1.5.4	DX10 Release 3.6	11
1.5.5	DX10 Release 3.6.1	13
1.5.5	DX10 Release 3.7	13
1.5.6	DX10 Release 3.6.1 AND 3.7 for Multiple Boards	14
1.6	Completing System Generation.	15
1.7	Installing The Hardware.	16
1.8	Patching COBOL Program Files.	18
1.9	Initializing The Accelerator.	19
1.10	Testing The Accelerator.	20

SECTION 2 INSTALLATION FOR DNOS

2.1	Copying Your Software.	21
2.1.1	Diskette.	21
2.1.2	Magnetic Tape.	22
2.2	Installing Ten X SCI Commands.	23
2.3	Choosing A Slot.	23
2.4	Pre-SYSGEN -- Defining Ten X System Interface.	24
2.5	SYSGEN -- Defining The System.	26
2.5.1	DNOS Release 1.1.	27
2.5.2	DNOS Release 1.2.	28
2.5.2	DNOS Release 1.3.	28
2.6	Completing System Generation.	29
2.7	Installing The Hardware.	30

TABLE OF CONTENTS

2.8	Patching COBOL Program Files.	32
2.9	Initializing The Accelerator.	33
2.10	Testing The Accelerator.	34
SECTION 3 PROGRAMMING TIPS		
3.1	Program Performance.	35
3.2	Maximizing Performance.	35
SECTION 4 THE TENX COMMAND		
4.1	TENX Command Overview.	37
4.2	Ten X Command Operation.	37
4.3	Ten X Statistics.	40
4.4	Examples of Statistics.	42
	4.4.1 An I/O Intense Environment.	42
	4.4.2 A Compute Intense Program.	43
SECTION 5 TROUBLESHOOTING		45

TABLE OF CONTENTS

APPENDICES

A	Error Messages.	49
B	Ten X 99 Commands.	59
C	POP Command Descriptions.	67
D	POP Command Generation.	69
E	Ten X 99 TILINE Interface.	73
F	TILINE Access Granted and Interrupt Location.	76
G	Installation Checklist.	78
H	Power Calculation Checklist.	79

LIST OF FIGURES

1-1	Modifying The Ejector Lever.	17
2-1	Modifying The Ejector Lever.	31

LIST OF TABLES

1-1	TILINE Address Selection.	17
2-1	TILINE Address Selection.	32
4-1	POP Command Ranges.	41
4-2	Additional Statistics.	41
5-1	System Generation Problems.	46
A-1	Diagnostic Errors.	56

INSTALLATION FOR DX10

SECTION 1

INSTALLATION FOR DX10

Please use the supplied Installation Checklist, Appendix G, to ensure complete and proper installation of your Ten X 99.

Calculate the power load of your system per Appendix H.

1.1 COPYING YOUR SOFTWARE

Ten X provides software to its customers on a double-sided double-density diskette (FD 1000). This software is also available upon request on magnetic tape. It is necessary to copy this software to a directory on your system disk.

*** NOTE ***

This section includes several examples of Ten X commands. Refer to APPENDIX B for more information about these commands. The appendix shows the GENERIC responses to the prompts.

1.1.1 DISKETTE

To copy the software from a diskette, perform the following steps:

1. Insert the diskette in the appropriate disk drive.
2. Perform an Install Volume (IV) command.

EXAMPLE

```
[ ] IV
INSTALL VOLUME
UNIT NAME: DSOx
VOLUME NAME: TENXDX10
```

3. If you have a .TENX directory on your system disk delete it with the Delete Directory (DD) command.

EXAMPLE

```
[ ] DD
DELETE DIRECTORY
DIRECTORY NAME: .TENX
LISTING ACCESS NAME: <return>
ARE YOU SURE? (Y,N): Y
```

INSTALLATION FOR DX10

*** CAUTION ***

IF YOU ALREADY HAVE A .TENX DIRECTORY ON
YOUR SYSTEM DISK DO NOT SKIP STEP 3.

4. Copy the contents of the diskette to the directory .TENX (created by the CD command) on the system disk using a Copy Directory (CD) command.

EXAMPLE

```
[ ] CD
COPY DIRECTORY
  INPUT PATHNAME: TENXDX10
  OUTPUT PATHNAME: .TENX
  CONTROL ACCESS NAME: <return>
  LISTING ACCESS NAME: <return>
  OPTIONS: ADD
  EXECUTION MODE(F/B): FOREGROUND
```

Examine the display listing to verify that the copy completed without any warnings or errors.

5. Unload the diskette using the Unload Volume (UV) command.

EXAMPLE

```
[ ] UV
UNLOAD VOLUME
  VOLUME NAME = TENXDX10
```

1.1.2 MAGNETIC TAPE

To copy the software from magnetic tape perform the following steps:

1. Mount the magnetic tape reel that contains Ten X 99 software on your tape drive and make the drive ready.
2. If you have a .TENX directory on your system disk delete it with the Delete Directory (DD) command.

EXAMPLE

```
[ ] DD
DELETE DIRECTORY
  DIRECTORY NAME: .TENX
  LISTING ACCESS NAME: <return>
  ARE YOU SURE? (Y,N): Y
```

*** CAUTION ***

IF YOU ALREADY HAVE A .TENX DIRECTORY ON
YOUR SYSTEM DISK DO NOT SKIP STEP 2.

INSTALLATION FOR DX10

3. Enter the Restore Directory (RD) command to copy this software to the .TENX directory (created by the RD command) on the system disk.

EXAMPLE

```
[ ] RD
RESTORE DIRECTORY
      DEVICE NAME: MTOx
      OUTPUT NAME: .TENX
CONTROL ACCESS NAME: <return>
LISTING ACCESS NAME: <return>
      OPTIONS: ADD
EXECUTION MODE(F/B): FOREGROUND
```

1.2 INSTALLING TEN X SCI COMMANDS

Next, copy the Ten X command directory to your command directory and perform the following steps:

1. Access the installation command file with the .USE primitive.

EXAMPLE

```
.USE .TENX..S$PROC
```

2. Copy the Ten X 99 commands to either the system command directory or your alternate command directory with the ITXP command.

EXAMPLE

```
[ ] ITXP
INSTALL TEN X PROCS
      DX10 OR DNOS: DX10
      PROC DIRECTORY: .S$PROC
LISTING ACCESS NAME: <return>
```

3. Examine the listing to verify that the commands were correctly installed.

1.3 CHOOSING A SLOT

Select a slot for each Ten X 99 COBOL Accelerator. Choose a slot in the MAIN CHASSIS; the Ten X 99 does NOT work in an expansion chassis. The Accelerator uses the interrupt connected to the right (P2) side of the slot. CHOOSE A SLOT THAT DOES NOT SHARE THE P2 INTERRUPT WITH ANY OTHER SLOT.

The Ten X board uses 5 amps of current from the +5 volt power supply. If you are already using 11 slots of the 13-slot chassis or 16 slots of the 17-slot chassis, an overload on the +5 volt power supply may occur. Verify the capacity of the power supply in your chassis. Add the amp rating of all boards in your

INSTALLATION FOR DX10

chassis (including the Ten X 99). If this value is above the rating of the chassis power supply, you may loose power immediately, or randomly during normal operation. Should this situation arise, there are two options (refer to Appendix H for power chart):

1. Move one or more devices to an expansion chassis so that you can install the Ten X 99 in the main chassis without overloading the +5 volt power supply.
2. Remove one or more devices from the system.

You MUST assign a TILINE address for each Ten X 99 Accelerator. Select any unused TILINE address. Set the switches for the TILINE address when you install the board, as described in a subsequent paragraph. This TILINE address CANNOT conflict with any other previously defined TILINE address. Previously defined TILINE addresses can be identified by performing a List Device Configuration (LDC) command.

1.4 PRE-SYSGEN -- DEFINING TEN X SYSTEM INTERFACE

The remainder of this section is dependent upon your particular version of DX10. You may ignore sections that do not pertain to your system. Please use the list below to enable you to proceed directly to the proper section.

DX10 VERSION	PARAGRAPH	PAGE
-----	-----	----
3.3	1.4.1	4
3.4	1.4.1	4
3.5	1.4.1	4
3.6.0	1.4.1	4
3.6.1	1.4.1	6
3.7	1.4.1	6

1.4.1 PRE-SYSGEN -- DX10 VERSIONS 3.3 THROUGH 3.7

In order for the Ten X 99 software to work properly in your system it must be tailored for your system. The ITX command must be executed to provide information to the Ten X 99 software. The information the software needs is:

1. The maximum number of COBOL tasks to await service by the Accelerator.
2. The number of Accelerators in the system.
3. The TILINE addresses and interrupt levels of those Accelerators.

Your responses to the ITX command supply this information. The following is an explanation of the required information.

INSTALLATION FOR DX10

QUEUE LENGTH:

The COBOL interpreter translates each COBOL statement into a set of POP commands. Whenever a COBOL task is running in your system it calls the Ten X 99 software to access the Accelerator to process each POP command. The software may put the task in a queue until the Accelerator finishes processing POP commands for another task. You can specify the length of this queue in a range of from 1 to 8 tasks. The queue length should relate to the capacity of your system. A queue length of 8 can require that 8 tasks be resident in memory at the same time. Each task could use 64K bytes of memory, or 8 X 64K total. You could need 512K bytes of memory for 8 COBOL tasks in addition to the memory required by the OS. Shared procedure code and programs less than 42k may reduce this request somewhat. Make the queue as long as possible without exceeding memory capacity. Four is typical.

TILINE ADDRESS

TILINE addresses are determined by the setting of switches on the Ten X 99 Accelerator. These are described in a subsequent paragraph. As shown in the example, enter a TILINE address opposite the prompt for the interrupt that corresponds to the Ten X 99. Enter addresses only for interrupts being used for Ten X 99s. The value for any interrupt that is not used by a Ten X 99 is zero.

The following is an example of an ITX command for one Ten X 99 installed at interrupt 7.

EXAMPLE

```
[ ]ITX
Ten X 99 Operating System Definition <RELEASE 3.1>
DX10 VERSION:( 33 - 37)
```

For DX10 versions 3.3 through 3.5 the following information will be displayed:

```
Ten X 99 Operating System Definition <RELEASE 3.1>
  QUEUE LENGTH(1-8): 4
    INTERRUPT 3: >0
    INTERRUPT 4: >0
    INTERRUPT 5: >0
    INTERRUPT 6: >0
    INTERRUPT 7: >F840
    INTERRUPT 8: >0
    INTERRUPT 9: >0
    INTERRUPT 10: >0
    INTERRUPT 11: >0
    INTERRUPT 12: >0
    INTERRUPT 13: >0
    INTERRUPT 14: >0
    INTERRUPT 15: >0
```

INSTALLATION FOR DX10

For DX10 versions 3.6.0 and 3.6.1 you must answer the DX10 VERSION prompt with either 360 or 361. These answers will produce the following results:

EXAMPLE OF 3.6.0

[]ITX

Ten X 99 Operating System Definition <RELEASE 3.1>
DX10 VERSION: 360

Ten X 99 Operating System Definition <RELEASE 3.1>

QUEUE LENGTH(1-8): 4

INTERRUPT 3: >0

INTERRUPT 4: >0

INTERRUPT 5: >0

INTERRUPT 6: >0

INTERRUPT 7: >F840

INTERRUPT 8: >0

INTERRUPT 9: >0

INTERRUPT 10: >0

INTERRUPT 11: >0

INTERRUPT 12: >0

INTERRUPT 13: >0

INTERRUPT 14: >0

INTERRUPT 15: >0

EXAMPLE OF 3.6.1

[]ITX

Ten X 99 Operating System Definition <RELEASE 3.1>
DX10 VERSION: 361 (Or 37 for DX10 Release 3.7)

Ten X 99 OS Definition - 3.6.1/3.7 <RELEASE 3.1>
QUEUE LENGTH(1-8): 4

When all the prompts for ITX have been answered the following message is displayed:

- WAIT FOR BACKGROUND TO COMPLETE

Enter the WAIT command and press RETURN. The ITX command should complete in approximately five minutes. When the command completes satisfactorily it displays this message:

*** TENX - NORMAL TERMINATION ***

If an error occurs, the command displays an error termination message. Refer to APPENDIX A for error listing and SECTION 5 for Troubleshooting. Correct the error and re-enter the Install Ten X (ITX) command.

To verify the completion of the Install Ten X command (ITX) under DX10 Versions 3.3 through 3.6.0 perform a Show File (SF) on file .TENX.DX10.LST.BLDTENX.

To verify the completion of the Install Ten X command (ITX) under DX10 Versions 3.6.1 through 3.7 perform a Show File (SF) on file .TENX.DX10.LST.BTENX361.

INSTALLATION FOR DX10

1.5 SYSGEN -- DEFINING THE SYSTEM

In the example that applies to your system enter the responses as they are shown. The ONLY response that you supply is the input name; that is the system name of your working system. You can find this name by entering a Show Volume Status (SVS) command for disk drive DS01. Use the name shown as the PRIMARY SYSTEM IMAGE.

All user synonyms should be deleted before beginning to generate the new system. The system generation tasks, especially those that complete the system generation, require almost the entire synonym table. Use the Modify Synonym (MS) command to delete YOUR synonyms and the Q\$SYN command to clear SYSTEM synonyms.

Verify that the sysgen directories, S\$SYSGEN and PATCH, are resident on the system disk. Ensure that the .S\$SYSGEN directory contains a directory with the same name as the PRIMARY SYSTEM IMAGE. This can be verified by performing a List Directory command (LD) and scrolling the list upward until located.

When the ITX has completed satisfactorily execute the XGEN proc to define a new operating system. Generating the new system adds the Ten X XOP to your system. An example of adding the Ten X XOP to your working system is shown for each of the six versions of DX10.

For detailed information about system generation errors, see VOLUME V of the TEXAS INSTRUMENTS DX10 OPERATING SYSTEMS manual set. See VOLUME VI for error codes and messages. The remainder of this section is dependant upon your particular version of DX10. You may ignore sections that do not pertain to your system. Please use the list below to enable you to proceed directly to the proper section.

DX10 VERSION	PARAGRAPH	PAGE
-----	-----	----
3.3	1.5.1	7
3.4	1.5.2	8
3.5	1.5.3	9
3.6.0	1.5.4	11
3.6.1	1.5.5	13
3.7	1.5.5	13

1.5.1 System Definition for DX10 Release 3.3

Enter the XGEN command.

EXAMPLE

[] XGEN

XGEN Prompts

Responses

INSTALLATION FOR DX10

DATA DISK:	enter	DS01
TARGET DISK:	enter	DS01
INPUT:	enter	<input name>
OUTPUT:	enter	TENX
DEVICE:	press	RETURN
XOP:	enter	2
FILE:	enter	.TENX.DX10.OBJ.TENXOP
PC:	enter	TENXPC
WP:	enter	TENXWP
XOP:	press	RETURN
DEVICE:	press	RETURN
SVC:	press	RETURN

Configuration file complete

Do you want to save it? (YES) press RETURN

XGEN begins to execute and displays these messages as it completes each phase of the processing:

*****	CONFIGURATION FILES SAVED	*****
*****	D\$DATA FILE NOW BEING BUILT	*****
*****	THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT	*****
*****	BATCH FILE FOR SYSGEN IS NOW BEING BUILT	*****
*****	GEN990 TERMINATED *****	

1.5.2 System Definition for DX10 Release 3.4

Enter the XGEN command.

EXAMPLE

```
[ ] XGEN
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DS01 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: TENX
```

XGEN Prompts	Responses
-----	-----
DEVICE TYPE?	enter XOP
XOP LEVEL?	enter 2
FILE?	enter .TENX.DX10.OBJ.TENXOP
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
XOP LEVEL?	press RETURN
NEXT?	enter B
BUILD? (YES)	enter Y

XGEN begins to execute and displays these messages as it completes each phase of the processing:

*****	CONFIGURATION FILES SAVED	*****
*****	D\$DATA FILE NOW BEING BUILT	*****
*****	THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT	*****
*****	BATCH FILE FOR SYSGEN IS NOW BEING BUILT	*****
*****	GEN990 TERMINATED *****	

INSTALLATION FOR DX10

1.5.3 System Definition for DX10 Release 3.5

NOTE: If your system includes communications or 940 terminals, the effect of the installation upon the system table area may cause an error during the ALGS portion of the system generation. To determine the amount of change that needs to be made in the TABLE parameter of the XGEN, use the following procedure (calculations can be done using the SV command):

NOTE: The symbol ">" represents hexadecimal value.

1. Do a show file (SF) on file .S\$SYSGEN.<system name>.LINKMAP
2. For systems with 940 terminals, find the value of X, where $X = ((\text{origin of DSR940}) + (\text{length of DSR940}) + >12C0 + >AFO)$
3. For other communications, find the longest PHASE 2 OVERLAY, where $X = ((\text{origin of OVERLAY}) + (\text{length of OVERLAY}) + >AFO)$
4. If "X" is greater than >F800, reduce the XGEN TABLE parameter by performing the following calculation, where "T" is in DECIMAL: $T = (X - >F800) / 2$

Enter the XGEN command.

EXAMPLE

```
[ ] XGEN
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DS01 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: TENX
```

ADDING TEN X

The XGEN prompts and responses for adding the Ten X 99 COBOL Accelerator to a working system are different when the working system includes a previous version of Ten X software. Use these prompts and responses when your system does not include any Ten X software:

XGEN Prompts	Responses
-----	-----
DEVICE TYPE?	press CMD key
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter WIZ
WIZARD?	enter XXY
NEXT?	enter XOP
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXVP
FILE?	enter .TENX.DX10.OBJ.TENXOP
XOP LEVEL?	press RETURN
NEXT?	enter B
BUILD? (YES)	enter Y

XGEN begins to execute and upon completion displays the following:

INSTALLATION FOR DX10

```
***** CONFIGURATION FILES SAVED *****
***** D$DATA FILE NOW BEING BUILT *****
***** THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT *****
***** BATCH FILE FOR SYSGEN IS NOW BEING BUILT *****
***** GEN990 TERMINATED *****:
```

REPLACING TEN X

Use these prompts and responses when you are replacing Release 2.x.x version of Ten X software with the current Ten X software release:

NOTE: If the XOP at Level 2 already exists, it will need to be deleted before continuing to the next step. Perform a LIST to determine if the XOP exists.

XGEN Prompts -----	Responses -----
DEVICE TYPE?	press CMD key
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter XOP
XOP LEVEL TO BE CHANGED OR DELETED?	enter 2
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
FILE?	enter .TENX.DX10.OBJ.TENXOP
COMMAND?	enter STOP
BUILD? (YES)	enter Y

XGEN begins to execute and displays these messages as it completes each phase of the processing:

```
***** CONFIGURATION FILES SAVED *****
***** D$DATA FILE NOW BEING BUILT *****
***** THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT *****
***** BATCH FILE FOR SYSGEN IS NOW BEING BUILT *****
***** GEN990 TERMINATED *****:
```

INSTALLATION FOR DX10

1.5.4 System Definition for DX10 Release 3.6

NOTE: If your system includes communications or 931/940 terminals, the effect of the installation upon the system table area may cause an error during the ALGS portion of the system generation. To determine the amount of change that needs to be made in the TABLE parameter of the XGEN, use the following procedure (calculations can be done using the SV command):

NOTE: The symbol ">" represents hexadecimal value.

1. Do a show file (SF) on file .S\$SYSGEN.<system name>.LINKMAP
2. For systems with 931/940 terminals, find the origin and length of DSR93X where X = A, B or C, perform the following calculation: $X = ((\text{origin of DSR93X}) + (\text{length of DSR93X}) + >\text{AFO})$
3. For other communications, find the longest PHASE 2 OVERLAY, where $X = ((\text{origin of OVERLAY}) + (\text{length of OVERLAY}) + >\text{AFO})$
4. If "X" is greater than >F800, reduce the XGEN TABLE parameter by performing the following calculation, where "T" is in DECIMAL: $T = (X - >\text{F800}) / 2$

Enter the XGEN command.

EXAMPLE:

```
[ ] XGEN
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DSO1 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: TENX
```

ADDING TEN X

The XGEN prompts and responses for adding the Ten X 99 COBOL Accelerator to a working system are different when the working system includes a previous version of Ten X 99 software. Use these prompts and responses when your system DOES NOT include any Ten X software:

XGEN Prompts	Responses
-----	-----
DEVICE TYPE?	press CMD key
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter (
TI ANALYST MODE:	enter @AUSTIN
NEXT?	enter XOP
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
FILE?	enter .TENX.DX10.OBJ.TENXOP
XOP LEVEL?	press RETURN
NEXT?	enter B
BUILD? (YES)	enter Y

XGEN begins to execute and upon completion displays the following:

INSTALLATION FOR DX10

```
***** CONFIGURATION FILES SAVED *****
***** D$DATA FILE NOW BEING BUILT *****
***** THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT *****
***** BATCH FILE FOR SYSGEN IS NOW BEING BUILT *****
***** GEN990 TERMINATED *****:
```

REPLACING TEN X

Use these prompts and responses when you are replacing Release 2.x.x version of Ten X 99 software with the current release of Ten X software:

NOTE: If the XOP at Level 2 already exists, it will need to be deleted before continuing to the next step. Perform a LIST to determine if the XOP exists.

XGEN Prompts -----	Responses -----
DEVICE TYPE?	press CMD key
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter XOP
XOP LEVEL TO BE CHANGED OR DELETED?	enter 2
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
FILE?	enter .TENX.DX10.OBJ.TENXOP
COMMAND?	enter STOP
BUILD? (YES)	enter Y

XGEN begins to execute and displays these messages as it completes each phase of the processing:

```
***** CONFIGURATION FILES SAVED *****
***** D$DATA FILE NOW BEING BUILT *****
***** THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT *****
***** BATCH FILE FOR SYSGEN IS NOW BEING BUILT *****
***** GEN990 TERMINATED *****:
```

INSTALLATION FOR DX10

1.5.5 System Definition for DX10 Release 3.6.1 and 3.7

Enter the XGEN command.

EXAMPLE:

```
[ ]XGEN
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DSO1 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: TENX
```

ADDING TEN X

The XGEN prompts and responses for adding the Ten X 99 COBOL Accelerator to a working system are different when the working system includes a previous version of Ten X 99 software. Use these prompts and responses when your system DOES NOT include any Ten X software:

XGEN Prompts -----	Responses -----
DEVICE TYPE?	enter SD
INTERFACE TYPE?	enter NONE
SPECIAL DEVICE ADDRESS?	enter the TILINE address of Ten X
DEVICE NAME?	enter TE01
KSB ADDRESS? (NONE)	press RETURN key
DSR WORKSPACE?	enter TNXWP1
INTERRUPT ENTRY?	enter TNXIPC
PDT FILE?	enter .TENX.DX10.SRC.PDT1
DSR FILE?	enter .TENX.DX10.OBJ.DX10DSR
OVERRIDE (YES)?	enter YES
INTERRUPT?	enter interrupt of the Ten X
DEVICE TYPE?	press CMD key
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter (
(TI ANALYST MODE)	enter @AUSTIN
NEXT?	enter XOP
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
FILE?	enter .TENX.DX10.OBJ.TENXOP
XOP LEVEL?	press CMD key
COMMAND?	enter LIST (to verify entries)
COMMAND?	enter B
BUILD?	enter Y

XGEN begins to execute and upon completion displays the following:

```
***** CONFIGURATION FILES SAVED *****
***** DSDATA FILE NOW BEING BUILT *****
***** THE LINK EDIT COMMAND SOURCE FILE IS BEING BUILT *****
***** BATCH FILE FOR SYSGEN IS NOW BEING BUILT *****
***** GEN990 TERMINATED *****
```

INSTALLATION FOR DX10

1.5.6 System Definition for DX10 3.6.1 and 3.7. multiple boards.

A special device must be defined for each Ten X Accelerator included in your system. Each of these special devices must have a unique device name, DSR workspace, PDT file, interrupt and TILINE address. Thus, these sysgen prompts must be answered differently for each Accelerator defined. The interrupt and TILINE address prompts will be answered depending on your hardware configuration. Refer to the list below to see how to answer the device name, DSR workspace and PDT file prompts for each Accelerator.

	DEVICE NAME	DSR WORKSPACE	PDT FILE
Accelerator #1	TE01	TNXWP1	.TENX.DX10.SRC.PDT1
Accelerator #2	TE02	TNXWP2	.TENX.DX10.SRC.PDT2
Accelerator #3	TE03	TNXWP3	.TENX.DX10.SRC.PDT3
Accelerator #4	TE04	TNXWP4	.TENX.DX10.SRC.PDT4

The following example shows how the sysgen prompts should be answered for the second Accelerator.

EXAMPLE

DEVICE TYPE?	enter SD
INTERFACE TYPE?	enter NONE
SPECIAL DEVICE ADDRESS?	enter the TILINE address of Ten X
DEVICE NAME?	enter TE02
KSB ADDRESS?(NONE)	press return
DSR WORKSPACE?	enter TNXWP2
INTERRUPT ENTRY?	enter TNXIPC
PDT FILE?	enter .TENX.DX10.SRC.PDT2
DSR FILE?	enter .TENX.DX10.OBJ.DX10DSR
OVERRIDE (YES)?	enter YES
INTERRUPT?	enter interrupt of Ten X board

1.6 COMPLETING SYSTEM GENERATION

To complete the system generation execute the ALGS and PGS commands and install the new system for testing using the TGS command. These commands use most of the synonym table area assigned to a terminal. Ensure your synonyms are deleted before you continue. DO NOT delete system synonyms at this time. Perform the following steps:

1. Enter the ALGS command. NOTE: For DX10 version 3.3, a valid pathname must be entered for D\$DATA and BATCH LISTING prompts.

EXAMPLE

```
[ ] ALGS
ASSEMBLE AND LINK GENERATED SYSTEM
  DATA DISK: DSO1
  TARGET DISK: DSO1
  SYSTEM NAME: TENX
D$DATA LISTING: <return>
BATCH LISTING: <return>
```

2. ALGS takes several minutes to complete. Enter WAIT and press RETURN.
3. When ALGS completes normally the system displays this message:

*** ALGS - NORMAL TERMINATION ***:

4. If the system displays an error message enter a Show File command (SF) to look at .S\$SYSGEN.TENX.ALGSLIST, the batch listing file. Identify the error, correct it, and repeat steps 1 through 3.
5. Enter the PGS command.

EXAMPLE

```
[ ] PGS
PATCH GENERATED SYSTEM
  DATA DISK: DSO1
  TARGET DISK: DSO1
  SYSTEM NAME: TENX
BATCH LISTING: <return>
```

6. PGS takes fifteen minutes or more to complete. Enter WAIT and press RETURN.
7. When PGS completes normally the system displays this message:

SYSTEM PATCH STREAM ERROR COUNT = 0:

8. If this message is not displayed, or if the error count is not zero, enter a Show File (SF) command to look at .S\$SYSGEN.TENX.PGSLIST, the batch listing file. Identify the error, correct it, repeat steps 5-7.

INSTALLATION FOR DX10

9. Enter the TGS command.

EXAMPLE

```
[ ] TGS
TEST GENERATED SYSTEM
  TARGET DISK: DS01
  SYSTEM NAME: TENX
```

10. Verify from the display that the secondary system is TENX, the newly generated system. The display looks like this:

```
IGS/TGS
<Current System Name> SYSTEM IPL STATUS
PRIMARY SYSTEM = <Current System Name> SECONDARY SYSTEM = TENX
TEST SECONDARY SYSTEM
```

This completes the installation of the Ten X 99 software.

1.7 INSTALLING THE HARDWARE

BEFORE CONTINUING REMOVE CHASSIS POWER PLUG FROM THE WALL OUTLET.

Refer to Appendix F for jumper and interrupt location.

13 SLOT CHASSIS

If your computer is in a 13-SLOT CHASSIS you must remove the TILINE ACCESS GRANTED jumper for each slot in which you intend to install an Accelerator. Slide the chassis out of the cabinet. As you face the side of the chassis into which you insert the boards, the jumpers are located on your right. For slots 1 through 7 the jumpers are above the plugs. For slots 8 through 12 they are below the plugs. You can use slot 13 for your Accelerator board even though it has no jumper.

17 SLOT CHASSIS

If your computer is in a 17-SLOT CHASSIS you must set the TILINE ACCESS GRANTED switch for each Accelerator to the OFF position. Switches 1 through 8 in switch unit S2 correspond to slots 2 through 9, switches 1 through 7 in switch unit S1 correspond to slots 10 through 16. Switch 8 in unit S1 is not used. You can use slot 17 for your Ten X 99 even though it has no switch.

NOTE: If a TILINE Master device is not installed in a slot, the ACCESS GRANTED jumper must be installed (13 slot) or the ACCESS GRANTED switch must be ON (17 slot).

The Ten X 99 Accelerator has an ejector lever at each side of the front. Before you install the board in the 17-slot chassis you must cut each ejector lever as shown in Figure 1-1.

INSTALLATION FOR DX10

With wirecutter or similar tool, remove tabs from ejector ear as indicated by arrows, two places.

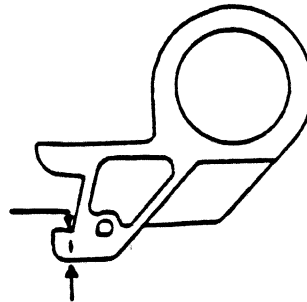


Figure 1-1. Modifying the Ejector Lever

Carefully remove the Ten X 99 from its protective envelope and set the TILINE address switches on the board to the address you used in the ITX command. The eight switches correspond to the bits represented by the two middle hexadecimal digits of the TILINE address. The most significant digit is always F, the least significant digit is always zero. Table 1-1 shows the settings of the switches for several TILINE addresses.

The Ten X 99 TILINE address switch is located at the bottom left hand corner of the board, coordinates BB05.

Table 1-1. TILINE Address Selection

TILINE ADDRESS	SWITCH							
	1	2	3	4	5	6	7	8
F900	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
F810	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
F820	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF
F830	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
F840	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF
F850	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON
F860	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF
F870	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
F880	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF
F900	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
FA00	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF

Slide the board into the proper slot. Press firmly on the front edge near the ejector levers to seat it properly.

Restore power to the system and load (IPL) the newly generated system. Verify that all terminals can log on and all devices operate properly.

NOTE: If a second consecutive system load (IPL) is performed, the system will revert to the original running system. You may, at this time, re-enter the TGS command and perform the IPL again to test the new system.

INSTALLATION FOR DX10

1.8 PATCHING COBOL PROGRAM FILES

Before you use your Ten X 99 Accelerator you must use the Patch COBOL Object (PCOBJ) command to modify your application program files. It is necessary to patch all COBOL runtime procedures linked with your COBOL tasks. To identify which procedures should be patched use the Find COBOL Runtime (FCR) command to examine all application program files.

EXAMPLE

```
[ ] FCR
Find COBOL Runtime Procedures <RELEASE 3.1>
    DIRECTORY: (directory or volume name you wish to search)
    LISTING: (filename where the results will be posted)
```

When all the prompts for FCR have been answered the following message is displayed:

-WAIT FOR BACKGROUND TO COMPLETE

When the background task has completed, examine the listing file for the results. An example of the information in the listing file follows.

PROGRAM FILE NAME	PROC NAME	PROC ID	COBOL VERSION	PATCHED
FIXED.S\$PROGA	RCOBOL	10	DX10 34	Y
FIXED.AP.PROG	RCOBOL	01	DX10 34	N

ONLY COBOL RUNTIME PROCEDURES ARE LISTED.

To patch the COBOL runtime procedure in .S\$PROGA for COBOL release 3.4 perform the following steps:

1. Enter PCOBJ as follows.

EXAMPLE

```
[ ] PCOBJ
Patch COBOL Runtime
    PROGRAM FILE NAME: .S$PROGA
    PROCEDURE NAME OR ID: RCOBOL
    COBOL RELEASE: 34
```

2. When PCOBJ successfully completes patching one program file it displays the following message:

FILE=.S\$PROGA PROG=RCOBOL ** PATCHED **

3. When PCOBJ displays an error message or completion code refer to the error messages in APPENDIX A for more information.

The Ten X command set also includes the UPCOBJ command to remove the Ten X patches from your program files.

INSTALLATION FOR DX10

1.9 INITIALIZING THE ACCELERATOR

The Ten X 99 COBOL Accelerator contains a processor that requires a program. The program, called the microcode, must be loaded into the memory of the Ten X 99 each time you load (IPL) your system. Loading the microcode is one of the functions of the TXL command.

The TXL command performs these functions:

- * Verifies installation of the Accelerator at the TILINE address and interrupt level assigned when the ITX was performed.
- * Verifies the operation of the Accelerator by running diagnostic tests.
- * Loads and verifies the microcode.
- * Enables the Ten X 99 COBOL Accelerator.

Your Ten X software includes two files containing microcode that you can load into the Accelerator when you initialize it:

- * DX1033 -- for users of COBOL Release 3.3.
- * DX1034 -- for users of COBOL Release 3.4.
- * DX1035 -- for users of COBOL Release 3.5.

Enter the TXL command with one of these file names as the COBOL type.

The following is an example of the TXL command using COBOL release 3.3. Only one version of COBOL can be used per Accelerator system.

EXAMPLE

```
[ ] TXL
Ten X Load
MICROCODE FILE NAME: .TENX.MICRO.DX1033
```

When TXL completes satisfactorily it displays a message similar to this one for an Accelerator using interrupt 7.

```
ASSIGNED LUNO = >AC
TEN X 99 INSTALLED AT INTERRUPT 7
```

If TXL detects a failure it displays an error message. Refer to Appendix A for explanations of error messages.

To initialize the Accelerator as part of the Initial Program Load, use the Text Editor (XE command) to add the TXL command to the Initialize System command in the file .S\$PROC.IS.

INSTALLATION FOR DX10

1.10 TESTING THE ACCELERATOR

Your Ten X software includes several COBOL test programs. One of these programs, TIMEMARK, executes in 3 seconds on an idle system when Ten X hardware and software are properly installed. To execute TIMEMARK enter the following command.

EXAMPLE

```
[ ] XCPF
EXECUTE COBOL PROGRAM FOREGROUND
  OBJECT ACCESS NAME: .TENX.TIMEMARK
    DEBUG MODE: NO
  MESSAGE ACCESS NAME: <return>
    SWITCHES: 00000000
  FUNCTION KEYS: NO
```

When TIMEMARK executes correctly you can be reasonably certain that your system and Accelerator are operating properly. Enter the Install Generated System (IGS) command to install the new system as the primary system.

EXAMPLE

```
[ ] IGS
INSTALL GENERATED SYSTEM
  TARGET DISK: DS01
  SYSTEM NAME: TENX
```

The newly generated system is now the primary system and it will be loaded whenever you load (IPL) the system.

SECTION 2

INSTALLATION FOR DNOS

Please use the supplied Installation Checklist, Appendix G, to ensure complete and proper installation of your Ten X 99.

Calculate the power load of your system per Appendix H.

2.1 COPYING YOUR SOFTWARE

The software for your Ten X 99 may come to you on a double-sided double-density diskette (FD1000). This software is also available upon request on magnetic tape. It is necessary to copy this software to a directory on your system disk.

*** NOTE ***

This section includes several examples of Ten X commands. Refer to APPENDIX B for more information about these commands. The appendix shows the GENERIC responses to the prompts.

2.1.1 DISKETTE

To copy the software from a diskette perform the following steps:

1. Insert the diskette in a disk drive.
2. Perform an Install Volume (IV).

EXAMPLE

```
[ ] IV
INSTALL VOLUME
  DEVICE NAME: DSOx
  VOLUME NAME: TENXD NOS
```

3. If you have a .TENX directory on your system disk delete it with the Delete Directory (DD) command.

EXAMPLE

```
[ ] DD
DELETE DIRECTORY
  DIRECTORY NAME: .TENX
  LISTING ACCESS NAME:<return>
  ARE YOU SURE? (Y,N): Y
```

*** NOTE ***

IF YOU ALREADY HAVE A .TENX DIRECTORY ON YOUR SYSTEM DISK DO NOT SKIP STEP 3.

INSTALLATION FOR DNOS

4. Copy the directory on the diskette to directory .TENX (created by the CD command) on the system disk using a Copy Directory (CD) command.

EXAMPLE

```
[ ] CD
COPY DIRECTORY
      INPUT NAME: TENXDNOS
      OUTPUT NAME: .TENX
CONTROL ACCESS NAME: <return>
LISTING ACCESS NAME: <return>
      OPTIONS: ADD
EXECUTION MODE(F/B): FOREGROUND
```

5. Unload the diskette using the Unload Volume (UV) command.

EXAMPLE

```
[ ] UV
UNLOAD VOLUME
      VOLUME NAME = TENXDNOS
```

2.1.2 MAGNETIC TAPE

To copy the software from magnetic tape perform the following steps:

1. Mount the magnetic tape reel that contains Ten X 99 software on your tape drive and make the drive ready.
2. If you have a .TENX directory on your system disk delete it with the Delete Directory (DD) command.

EXAMPLE

```
[ ] DD
DELETE DIRECTORY
      DIRECTORY NAME: .TENX
LISTING ACCESS NAME: <return>
ARE YOU SURE? (Y,N): Y
```

*** NOTE ***

IF YOU ALREADY HAVE A .TENX DIRECTORY ON
YOUR SYSTEM DISK DO NOT SKIP STEP 2.

INSTALLATION FOR DNOS

3. Enter the Restore Directory (RD) command to copy the software to the .TENX directory (created by the CD command) on the system disk.

EXAMPLE

```
[ ] RD
RESTORE DIRECTORY
      DEVICE NAME: MTOx
      OUTPUT NAME: .TENX
CONTROL ACCESS NAME: <return>
LISTING ACCESS NAME: <return>
      OPTIONS: ADD
EXECUTION MODE(F/B): FOREGROUND
```

2.2 INSTALLING TEN X SCI COMMANDS

Next copy the Ten X command directory to your command directory. Perform the following steps:

1. Access the installation command file with the .USE primitive.

```
.USE .TENX..S$CMDS
```

2. Copy the Ten X 99 commands to either the system command directory or your alternate command directory with the ITPX command.

EXAMPLE

```
[ ] ITPX
INSTALL TEN X PROCS
      DX10 OR DNOS: DNOS
      PROC DIRECTORY: <.S$CMDS or alt. directory>
LISTING ACCESS NAME: <return>
```

3. Examine the listing to verify that the commands were correctly installed.

2.3 CHOOSING A SLOT

Select a slot for each Ten X 99 COBOL Accelerator. Choose a slot in the main chassis. The Ten X 99 does NOT work in an expansion chassis. The Accelerator uses the interrupt connected to the right (P2) side of the slot. CHOOSE A SLOT THAT DOES NOT SHARE THE P2 INTERRUPT WITH ANY OTHER SLOT.

The Ten X board uses 5 amps of current from the +5 volt power supply. If you are already using 11 slots of the 13-slot chassis, or 16 slots of the 17-slot chassis, an overload the +5 volt power supply may occur. Verify the capacity of the power supply in your chassis.

INSTALLATION FOR DNOS

Add the amp rating of all boards in your chassis (including the Ten X 99). If this value is above the rating of the chassis power supply, you may lose power immediately or randomly during normal operation. Should this situation arise, there are two options (refer to Appendix H for power chart):

1. Move one or more devices to an expansion chassis so you can install the Ten X 99 in the main chassis without overloading the +5 volt power supply.
2. Remove one or more devices from the system.

You MUST assign a TILINE address for each Ten X Accelerator. Select any unused TILINE address. Set the switches for the TILINE address when you install the board as described in a subsequent paragraph.

2.4 PRE-SYSGEN -- DEFINING TEN X SYSTEM INTERFACE

In order for the Ten X 99 software to work properly in your system it must be tailored for your system. The ITX command must be executed to provide information to the Ten X software. The information the software needs is:

1. The DNOS version.
2. The maximum number of COBOL tasks to await service by the Accelerator.
3. The number of terminals on your system.
4. The number of Accelerators in the system.
5. The TILINE addresses and interrupt levels of those Accelerators.

Your responses to the ITX command supply this information. The following is an explanation of the required information. See example following explanations.

DNOS VERSION

Respond with 11 for version 1.1, 12 for version 1.2 or 13 for version 1.3.

QUEUE LENGTH

The COBOL interpreter translates each COBOL statement into a set of POP commands. Whenever a COBOL task is running in your system it calls the TEN X 99 software to access the Accelerator to process each POP command. The software may put the task in a queue until the Accelerator finishes processing POP commands for another task. You can specify the length of this command in a range of from 1 to 8 tasks. The queue length should relate to the capacity of your

INSTALLATION FOR DNOS

system. A queue length of 8 can require that 8 tasks be resident in memory at the same time. Each task could use 64K bytes of memory, or 8 X 64K total. You could need 512K bytes of memory for 8 COBOL tasks in addition to the memory required by the OS. Make the queue as long as possible without exceeding memory capacity. Four is typical.

NUMBER OF TERMINALS

Enter the number of terminals that are being used on your system.

The number of terminals on your system will be a good estimate of the maximum number of COBOL tasks that could be on the DNOS active queue at any one time.

TILINE ADDRESS

TILINE addresses are determined by the setting of switches on the Ten X 99 Accelerator. These are described in a subsequent paragraph. As shown in the following example, enter a TILINE address opposite the prompt for the interrupt that corresponds to the Ten X 99. Note that multiple board installation requires additional address entries at the corresponding interrupt prompts. Enter addresses only for interrupts being used for Ten X 99s. A value of zero is entered for the remaining unused interrupts.

The following is an example of an ITX command for one Ten X 99 installed at interrupt 7.

EXAMPLE

```
AOITX
Ten X 99 Operating System Definition
  DNOS VERSION (11, 12 or 13):11
    QUEUE LENGTH(1-8):4
      NUMBER OF TERMINALS: 25
        INTERRUPT 3: >0
        INTERRUPT 4: >0
        INTERRUPT 5: >0
        INTERRUPT 6: >0
        INTERRUPT 7: >F840
        INTERRUPT 8: >0
        INTERRUPT 9: >0
        INTERRUPT 10: >0
        INTERRUPT 11: >0
        INTERRUPT 12: >0
        INTERRUPT 13: >0
        INTERRUPT 14: >0
        INTERRUPT 15: >0
```

When all the prompts for ITX have been answered the following message is displayed:

-WAIT FOR BACKGROUND TO COMPLETE

INSTALLATION FOR DNOS

Enter the WAIT command and press RETURN. The ITX command should complete in approximately five minutes. Upon normal termination, the following messages should be returned:

BACKGROUND EXECUTION COMPLETE

*** TENX - NORMAL TERMINATION ***

If an error occurs the command displays an error termination message. Refer to APPENDIX A. Correct the error and re-enter the Install Ten X (ITX) command. To verify the completion of the Install Ten X (ITX) command, perform a Show File (SF) on file .TENX.DNOS.LST.BLDTENX.

2.5 SYSGEN -- DEFINING THE SYSTEM

When the ITX command has completed satisfactorily execute the Execute System Generation Utility (XSGU) to define a new operating system. Generating the new system adds the Ten X XOP to your system.

This addition to your system will consume, from the system table area, 550 bytes PLUS 22 bytes times the number of terminals.

Also, before beginning the sysgen, verify that sysgen directories S\$SGU\$ and S\$OSLINK exist before you start. This can be checked by doing a List Directory (LD) on your system disk.

In the example that applies to your system enter the responses as they are shown. One response that you supply is the input name; that is the system name of your running system. You can get this name by entering a Show Volume Status (SVS) command for disk drive DS01. Use the name shown as the PRIMARY SYSTEM IMAGE. This name must also be in the sysgen directory. The other response that you supply is the TILINE address you assigned for the Accelerator board. Use the LDC command to find an unused address.

NOTE:

DURING THE SYSGEN, ONLY ONE SPECIAL DEVICE (SD) IS REQUIRED AS ENTRY (TE) FOR THE TEN X 99 BOARD(S). MORE THAN ONE WILL CAUSE FAILURE IN THE ALGS PORTION OF THE SYSGEN. Multiple board installation is accomplished during ITX. Doing a "LIST" prior to performing the "BUILD" at the end of the XSGU confirms that only one TE entry exists.

For detailed information about system generation errors refer to your Texas Instruments DNOS System Generation Manual. See the TI DNOS Messages and Codes manual for the error codes.

Determine your version of DNOS and refer to the list below to allow you to go directly to the correct page.

DNOS VERSION	PARAGRAPH	PAGE
1.1	2.5.1	28
1.2	2.5.2	29
1.3	2.5.2	29

INSTALLATION FOR DNOS

2.5.1 System Definition for DNOS Release 1.1

Enter the XSGU command:

EXAMPLE

```
[ ] XSGU
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DS01 (or appropriate volume name)
  TARGET DISK/VOLUME: DS01 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: S$TENX (this must be a unique name)
  ASSEMBLE AND LINK?: NO
```

XSGU Prompts -----	Response -----
ENTITY?	press CMD
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter WIZ
WIZARD?	enter ZZYX
ENTITY?	enter XOP
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
PATHNAME?	enter .TENX.DNOS.OBJ.TENXOP
XOP LEVEL?	press RETURN
ENTITY?	enter DVC
DEVICE TYPE?	enter SD
TILINE DEVICE?	enter Y
TILINE ADDRESS (>F800)	enter TILINE address
VALIDATE OPENS? (NO)	enter NO
TIME OUT? (30)	enter 0
DEVICE NAME?	enter TE
IS THE PDT THE INTERRUPT WORKSPACE? (YES)	enter NO
KEYBOARD? (YES)	enter NO
INTERRUPT? (15)	enter NONE
DEVICE TYPE?	press CMD
COMMAND?	enter LIST (Verify entries)
COMMAND?	enter STOP
BUILD? (NO)	enter Y

XSGU begins to execute and displays these messages as it completes each phase of the processing:

```
***** CONFIG IS NOW BEING BUILT *****
***** D$SOURCE IS NOW BEING BUILT *****
***** LINKSTREAMS ARE NOW BEING BUILT *****
***** ALGSSTRM IS NOW BEING BUILT *****
XSGU COMPLETE:
```

INSTALLATION FOR DNOS

2.5.2 System Definition for DNOS Release 1.2 and 1.3

Enter the XSGU command.

EXAMPLE

```
[ ] XSGU
EXECUTE SYSTEM GENERATION UTILITY
  DATA DISK/VOLUME: DS01 (or appropriate volume name)
  TARGET DISK/VOLUME: DS01 (or appropriate volume name)
  INPUT CONFIGURATION: <input name> (a running configuration)
  OUTPUT CONFIGURATION: S$TENX (this must be a unique name)
  ASSEMBLE AND LINK?: NO
```

XSGU Prompts -----	Response -----
ENTITY?	press CMD
COMMAND?	enter CHANGE
PARAMETER TO BE CHANGED?	enter WIZ
WIZARD?	enter ZZYX
ENTITY?	enter XOP
XOP LEVEL?	enter 2
PC LABEL?	enter TENXPC
WP LABEL?	enter TENXWP
PATHNAME?	enter .TENX.DNOS.OBJ.TENXOP
XOP LEVEL?	press RETURN
ENTITY?	enter DVC
DEVICE TYPE?	enter SD
VALIDATE OPENS? (NO)	enter NO
TIME OUT? (60)	enter 0
INTERFACE TYPE?	enter NONE
TILINE DEVICE?	enter Y
TILINE ADDRESS (>F800)	enter TILINE address
DEVICE NAME?	enter TE
IS THE PDT THE INTERRUPT WORKSPACE? (YES)	enter NO
KEYBOARD? (YES)	enter NO
INTERRUPT? (15)	enter NONE
DEVICE TYPE?	press CMD
COMMAND?	enter LIST (Verify entries)
COMMAND?	enter STOP
BUILD? (NO)	enter Y

XSGU begins to execute and displays these messages as it completes each phase of the processing:

```
***** CONFIG IS NOW BEING BUILT *****
***** D$SOURCE IS NOW BEING BUILT *****
***** LINKSTREAMS ARE NOW BEING BUILT *****
***** ALGSSTRM IS NOW BEING BUILT *****
XSGU COMPLETE:
```

INSTALLATION FOR DNOS

2.6 COMPLETING SYSTEM GENERATION

To complete the system generation execute the ALGS and PGS commands and install the new system for testing using the TGS command. Perform the following steps:

1. Enter the ALGS command:

EXAMPLE

```
[ ] ALGS
ASSEMBLE AND LINK GENERATED SYSTEM
  DATA DISK: DS01
  TARGET DISK: DS01
  SYSTEM NAME: S$TENX
```

2. ALGS takes several minutes to complete. Enter WAIT and press RETURN.
3. When ALGS completes normally the system displays this message:

0 ERRORS ENCOUNTERED IN ASSEMBLING AND LINKING

4. If the error count is not zero enter a Show File (SF) command to look at .S\$SGU\$.S\$TENX.ALGSLIST, the batch listing file. Identify the error, correct it, and repeat steps 1 through 3.
5. Enter the PGS command.

EXAMPLE

```
[ ] PGS
PATCH GENERATED SYSTEM
  DATA DISK: DS01
  TARGET DISK: DS01
  SYSTEM NAME: S$TENX
  PATCH KERNAL?: YES
  PATCH UTILITY?: NO
  PATCH COMM?: NONE
```

Note: The PATCH KERNAL, PATCH UTILITY, and PATCH COMM will be displayed under DNOS release 1.2 only.

6. PGS takes fifteen minutes or more to complete. Enter WAIT and press RETURN.
7. When PGS completes normally the system displays this message:

0 ERRORS ON DNOS KERNEL PATCH STREAM

INSTALLATION FOR DNOS

8. If the error count is not zero enter a Show File (SF) command to look at .S\$SGU\$.S\$TENX.PGSLIST, the batch listing file. Identify the error, correct it, and repeat steps 5 through 7.
9. Enter the TGS command:

EXAMPLE

```
[ ] TGS
TEST GENERATED SYSTEM
  TARGET DISK: DS01
  SYSTEM NAME: S$TENX
```

10. Verify from the display that the secondary system is S\$TENX, the newly generated system. The display looks like this:

```
IPL STATUS OF DS01:
TEST SECONDARY SYSTEM NAME =
      S$TENX
```

This completes installation of the Ten X 99 software.

2.7 INSTALLING THE HARDWARE

BEFORE CONTINUING REMOVE POWER FROM THE SYSTEM.

Refer to APPENDIX F for jumper and interrupt location.

13 SLOT CHASSIS

If your computer is in a 13-slot chassis you must remove the TILINE ACCESS GRANTED jumper for each slot in which you intend to install an Accelerator. Slide the chassis out of the cabinet. As you face the side of the chassis into which you insert the boards, the jumpers are located on your right. For slots 1 through 7 the jumpers are above the plugs. For slots 8 through 12 they are below the plugs. You can use slot 13 for your Accelerator board even though it has no jumper.

17 SLOT CHASSIS

If your computer is in a 17-slot chassis you must set the TILINE ACCESS GRANTED switch for each Accelerator to the OFF position. Switches 1 through 8 in switch unit S2 correspond to slots 2 through 9, switches 1 through 7 in switch unit S1 correspond to slots 10 through 16. Switch 8 in unit S1 is not used. You can use slot 17 for your Ten X 99 even though it has no switch. Refer to the DS990 Systems Field Engineering Handbook for further information.

The Ten X 99 Accelerator has an ejector lever at each side of the front. Before you install the board in the 17-slot chassis you must cut each ejector lever as shown in Figure 2-1.

INSTALLATION FOR DNOS

With wirecutter or similar tool, remove tabs from ejector ear as indicated by arrows, two places.

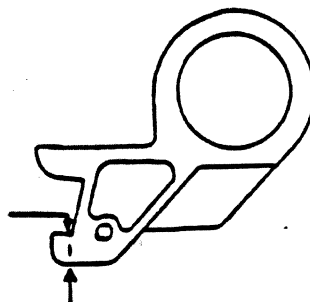


Figure 2-1. Modifying the Ejector Lever

Carefully remove the Ten X 99 from its protective envelope and set the TILINE address switches on the board to the address you used in the ITX command. The eight switches correspond to the bits represented by the two middle hexadecimal digits of the TILINE address. The most significant digit is always F, the least significant digit is always zero. Table 2-1 shows the settings of the switches for several TILINE addresses.

The Ten X 99 TILINE address switch is located at the bottom left corner of the board, coordinates BB05.

Table 2-1. TILINE Address Selection

TILINE ADDRESS	SWITCH							
	1	2	3	4	5	6	7	8
F800	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
F810	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
F820	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF
F830	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
F840	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF
F850	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON
F860	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF
F870	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
F880	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF
F900	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
FA00	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF

Slide the board into the proper slot. Press firmly on the front edge near the ejector levers to seat it properly.

Restore power to the system and load the newly generated system. Verify that all terminals can log on and all devices operate properly.

INSTALLATION FOR DNOS

2.8 PATCHING COBOL PROGRAM FILES

Before you use your Ten X 99 Accelerator you must use the Patch COBOL Object (PCOBJ) command to modify your application program files. It is necessary to patch all COBOL runtime procedures linked with your COBOL tasks. To identify which procedures should be patched use the Find COBOL Runtime (FCR) command to examine all application program files.

EXAMPLE

```
[ ]FCR
Find COBOL Runtime Procedures <RELEASE 3.1>
    DIRECTORY: (directory or volume name you wish to search)
    LISTING: (filename where the results will be posted)
```

When all prompts for FCR have been answered the following message is displayed:

-WAIT FOR BACKGROUND TO COMPLETE

When the background task has completed examine the listing file for the results. An example of the information in the listing file follows.

PROGRAM FILE NAME	PROC NAME	PROC ID	COBOL VERSION	PATCHED
FIXED.S\$SHARED	RCOBOL	10	DNOS 34	Y
FIXED.AP.PROG	RCOBOL	01	DNOS 34	N

ONLY COBOL RUNTIME PROCEDURES ARE LISTED.

To patch the COBOL runtime procedure in .S\$SHARED for COBOL RELEASE 3.4 perform the following steps.

1. Enter PCOBJ as follows:

EXAMPLE

```
[ ] PCOBJ
Patch COBOL Runtime
    PROGRAM FILE NAME: .S$SHARED
    PROCEDURE NAME OR ID: RCOBOL
    COBOL RELEASE: 34
```

2. When PCOBJ successfully completes patching the program file it displays the following message:

FILE=.S\$SHARED PROG=RCOBOL ** PATCHED **

3. When PCOBJ displays an error message or completion code refer to the error messages in APPENDIX A for more information.

The Ten X command set also includes the UPCOBJ command to remove the Ten X patches from your program files.

2.9 INITIALIZING THE ACCELERATOR

The Ten X 99 COBOL Accelerator contains a processor that requires a program. The program, called the microcode, must be loaded into the memory on the Ten X 99 each time you load your system.

Loading the microcode is one of the functions of the TXL command.

The TXL command performs these functions:

- * Verifies installation of the Accelerator at the TILINE address and interrupt level you assigned when the ITX was performed.
- * Verifies the operation of the Accelerator by running diagnostic tests.
- * Loads and verifies the microcode.
- * Enables the Ten X 99 COBOL Accelerator.

Your Ten X software includes files containing microcode that you can load into the Accelerator when you initialize it.

- * DNOS33 -- for users of COBOL Release 3.3.
- * DNOS34 -- for users of COBOL Release 3.4.
- * DNOS33P -- for users of COBOL PLUS Release 3.3.
- * DNOS34P -- for users of COBOL PLUS Release 3.4.
- * DNOS35 -- for users of COBOL Release 3.5.
- * DNOS35P -- for users of COBOL PLUS Release 3.5

Enter the TXL command with one of these file names as the COBOL type. For example:

EXAMPLE

```
[ ] TXL
Ten X Load <RELEASE 3.1>
    MICROCODE FILE NAME: .TENX.MICRO.DNOS33
```

INSTALLATION FOR DNOS

When TXL completes satisfactorily it displays a message similar to this one for an Accelerator using interrupt 7.

```
ASSIGNED LUND = >AC
TEN X 99 INSTALLED AT INTERRUPT 7
```

If TXL detects a failure it displays an error message. Refer to Appendix A for explanations of error messages.

To initialize the Accelerator as part of the Initial Program Load use the Text Editor (XE command) to add the TXL command to the batch stream at .S\$ISBTCH.

2.10 TESTING THE ACCELERATOR

Your Ten X software includes several COBOL test programs. One of these programs, TIMEMARK, executes in 3 seconds on an idle system when Ten X hardware and software are properly installed. To execute TIMEMARK enter the following command:

EXAMPLE

```
[ ] XCPF
EXECUTE COBOL PROGRAM FOREGROUND
  OBJECT ACCESS NAME: .TENX.TIMEMARK
    DEBUG MODE: NO
MESSAGE ACCESS NAME: <return>
  SWITCHES: 00000000
  FUNCTION KEYS: NO
```

When TIMEMARK executes correctly you can be reasonably certain that your system and Accelerator are operating properly. Perform the Install Generated System (IGS) command to install the new system as the primary system.

EXAMPLE

```
[ ] IGS
INSTALL GENERATED SYSTEM
  TARGET DISK: DS01
  SYSTEM NAME: S$TENX
```

The newly generated system is now the primary system and it will be loaded whenever you load (IPL) the system.

SECTION 3

PROGRAMMING TIPS

3.1 PROGRAM PERFORMANCE

As you become familiar with your Ten X 99 COBOL Accelerator you will notice that some programs show a much greater improvement in performance than others. This section provides tips that will help you write your programs to achieve optimum performance from your COBOL Accelerator. You may use these tips as you write new programs or as you "clean up" programs you already have. Sometimes it is better to completely rewrite old programs to fully utilize the capabilities of the Ten X 99.

3.2 MAXIMIZING PERFORMANCE

Here are several ways to program for increased performance with your Accelerator. Since the Ten X 99 executes the non-I/O parts of a COBOL program about 30 times faster than a 990/10 or 10 times faster than a 990/12 with Cache Memory, that part of a program is not very critical. Input and output, on the other hand, are limited by the speed of the particular I/O device being addressed and the 990 CPU time required by the Operating System and the Device Service Routine. Disk references are especially costly, being limited in speed by the mechanical characteristics of the particular disk. The most beneficial strategy to increase performance is to try to reduce the number of disk references made by a program.

First, DESIGN YOUR FILES so that one disk reference gets all the pertinent information in a record. Make your records as long as possible. For example, consider packing several discrete data items into a GROUP which can be accessed using indexing or table access instead of writing separate records for each data item.

Second, TURN OFF KEY LOGGING FOR TEMPORARY FILES. Key logging significantly increases the number of disk references for all file operations. Partial logging can be used for normal files to decrease the number of disk references required to insert a record. CAUTION: Read about the MKL command in the T.I. SCI manual.

WRITE AS MUCH OF YOUR APPLICATION AS POSSIBLE IN COBOL. Calls to assembly language routines slow down your programs and load down your 990 computer. Assembly language routines are generally slower than the equivalent COBOL on the Ten X 99. Using pure COBOL has the further benefit of unloading the 990 so that it may more efficiently handle I/O and scheduling. Also, the portability of your application is enhanced if assembly language routines are avoided.

PROGRAM YOUR APPLICATIONS TO RUN COBOL CONCURRENTLY rather than serially. While the 990 is doing I/O for one program the Ten X 99 can be running another. This overlap cannot happen if

PROGRAMMING TIPS

only one program is run at a time. The Accelerator can service several COBOL tasks during one Operating System time slice.

Use ACCEPT and DISPLAY for I/O to the screen and keyboard. Although the Accelerator does not perform the actual screen I/O, it performs ACCEPT/DISPLAY statements more efficiently than the 990.

AVOID UNNECESSARY CALL STATEMENTS. Do not use CALL when you can use PERFORM. The Accelerator must switch back to the 990 to get CALL executed, since the CALLED program may not be COBOL. All frequently used COBOL subroutines should be executed with PERFORM.

Use COMP-1, COMP-3, and COMP-4 data for computations instead of COMP or DISPLAY data. These types require less memory space and fewer memory cycles to manipulate.

Use COMP-1 for subscripts. If decimal data types are programmed for subscripts, they will be converted to binary each time the subscript is used. Conversions between decimal and binary waste time.

AVOID MIXING BINARY AND DECIMAL DATA TYPES. If binary and decimal data types are mixed in the same COBOL statement, unnecessary time is wasted in processing type conversions.

THE TENX COMMAND

SECTION 4

THE TENX COMMAND

4.1 TENX COMMAND OVERVIEW

The Ten X 99 does its work automatically when called by the COBOL Interpreter. The TENX command performs the following functions:

- * Displays TILINE addresses and interrupts for Accelerators.
- * Provides commands that:
 - Turn all Ten X 99 activity on or off.
 - Turn an individual Accelerator on or off.
 - Start and stop accumulation of user program statistics.
 - Display a report of accumulated statistics.

You can turn off all Ten X activity if you need to or you can turn off an individual Accelerator when there are more than one in the system.

The TENX Utility monitors the performance of the Ten X system by collecting and displaying statistics in a report. Because these statistics reflect all COBOL activity in the computer they provide useful information during heavy multiprogramming loads as well as when only one task is executing. You can turn the accumulation of statistics on and off with the TENX command.

This section tells you how to use the TENX command and the statistics it accumulates. This command can be used when you are running on a system that is generated for the Ten X Accelerator.

4.2 TENX COMMAND OPERATION

Enter the TENX command.

EXAMPLE

```
[ ]TENX
Ten X 99 Utility
LISTING ACCESS NAME: <file pathname>
```

NOTE: The File Pathname entered above will be automatically created if it does not exist and will be appended each time the statics are produced. Existing files will be appended when the statics are produced.

The command displays a screen similar to the following example. The example screen shows a Ten X installation using four Accelerators.

THE TENX COMMAND

```
+-----+
|=====|
|          T E N   X   9 9   U T I L I T Y          |
|=====|
|Ten X Status:  Ten X - On  Microcode-DX1034  Statistics Status - Off|
|
|Board Interrupt:  6          8          11          15
|Board Tiline:    >F840    >F860    >F880    >F8A0
|Board Status:    Enable   Enable   Disable Enable
|
|Utility Functions:
|
|1.  Turn Ten X On              5.  Begin Statistics Collection
|2.  Turn Ten X Off            6.  Suspend Statistics Collection
|3.  Enable a Board             7.  Continue Statistics Collection
|4.  Disable a Board           8.  Produce Statistics Report
|                              9.  Display Statistics Report
|
|Select one of the Utility Functions (CMD to exit):
|=====|
+-----+
```

The top half of the screen displays the status of Ten X.

- * Ten X Status -- shows the current values for the two system-wide variables.
- * Ten X On/Off -- the state of the entire set of Ten X Accelerators. This can be changed by function 1 or 2. A value of "On" means that the boards are available at the XOP interface to receive work. A value of "Off" means that no boards are being allocated work by the XOP processor and all COBOL POP commands are being executed by the Interpreter.
- * Statistics Status - On/Off/Wait -- the state of the statistics data collection process. "Off" means that no statistics are being collected. "On" means that statistics are being collected to count POP commands and Ten X activity. The "Wait" status indicates that statistics collection has been suspended and no report has been produced.
- * Microcode - DX1033/DX1034/DNOS33/DNOS34/DNOS33P/DNOS34P/NONE/TESTxx -- indicates currently loaded microcode version. NONE means no microcode is loaded. TESTxx means a test version is loaded. (xx indicates version number)
- * Board Interrupt -- the interrupt level of each Ten X board defined for the currently executing system.

THE TENX COMMAND

- * Board TILINE -- the TILINE address of each board defined for the currently executing system.
- * Board Status -- either "Enable" or "Disable". "Enable" means that the board is available to receive work from the XOP processor. "Disable" means that the board cannot receive work. A board may be disabled by function 2 or 4.

The bottom half of the screen is a list of the functions of the TENX utility. You can select one by typing the number of the function and pressing RETURN. The functions are:

1. Turn Ten X On -- set the Ten X Status to "On". Makes all enabled boards available to the system.
2. Turn Ten X Off -- set the Ten X Status to "Off". Effectively disables all boards.
3. Enable a Board -- sets a board status to "Enable". The board is identified by its interrupt level. Note that an enabled board with Ten X Status "Off" shows a board status of disabled since it cannot receive work. Changing the Ten X Status to "On" changes the board status to "Enable".
4. Disable a Board -- sets a board status to "Disable". The board is identified by its interrupt level.
5. Begin Statistics Collection -- begin accumulating totals for the statistics report. This function also clears all totals as it begins.
6. Suspend Statistics Collection -- stop collecting statistics. Totals collected are maintained and the statistics collection status goes to "Wait".
7. Continue Statistics Collection -- begin collecting statistics again after a "suspend" command. Totals collected thus far are maintained (just like a "begin" function except that totals are not cleared).
8. Produce Statistics Report -- reads collection totals, clears the totals, and produces a report, without changing Statistics Status. The new report is appended to any existing report.
9. Display Statistics Report -- displays the latest report file on the screen (the last report is at the END of the file). The file may be examined from beginning to end, restarted at the beginning, and deleted. You must perform function 8 before you perform function 9.

Press the CMD key to exit the TENX utility.

THE TENX COMMAND

4.3 TEN X STATISTICS

When starting the collection of statistics the TENX Utility begins counting the number of executions of each POP command by the COBOL Interpreter. A POP command is a high level command generated by the COBOL Compiler and performed by the COBOL Interpreter. The Accelerator executes POP commands as its machine or native language. Appendix C lists the 81 POP commands. Appendix D lists typical COBOL statements and the POP commands that result when the Compiler processes the statements. Counts of POP commands tell you what processing the COBOL Interpreter is doing. When the Accelerator is off these counts tell you what processing the COBOL executing tasks require. Counts taken while the Accelerator is on tell you how much of the total processing the Interpreter is doing. The Ten X 99 Accelerator normally executes all POP commands that do not require I/O. Appendix C identifies the POP commands that the Accelerator can execute. Non-zero counts for POP commands that the Accelerator could have done mean either that a numeric field contained non-numeric data or that a string was over 256 bytes long. Thus the executing program is not optimized for the Ten X 99.

The TENX Utility also counts statistics for the Accelerator. To understand these counts consider how the Accelerator works. When the Accelerator services a COBOL task it begins processing POP commands for that task. When the task calls for a POP command that the Accelerator does not execute (I/O, for example), that task is returned to the Operating System for I/O processing and the Accelerator begins to service another task. Table 4-1 lists ten ranges of the number of POP commands executed by the Accelerator before it switches to another task. The TENX Utility maintains a count for each range. Each time the Accelerator switches to another task the count for the applicable range of POP command executions is increased by one.

These counts are all zero on a report of statistics taken while the Accelerator is off. On a report taken while the Accelerator is on counts for the ranges show how efficiently the Accelerator is running. Programs are well matched to the Ten X 99 when the statistics show few counts in the ranges below 64 POP commands. Whatever you can do to increase the counts for the high ranges increases the throughput of your COBOL tasks that use the Accelerator.

THE TENX COMMAND

Table 4-1. POP Command Ranges

Report Label	POP Commands Executed
001 POP	Zero or 1
002 POPS	2 or 3
004 POPS	4 thru 7
008 POPS	8 thru 15
016 POPS	16 thru 31
032 POPS	32 thru 63
064 POPS	64 thru 127
128 POPS	128 thru 255
256 POPS	256 thru 511
512 POPS	512 or more

The TENX Utility also maintains the counts listed in Table 4-2. These counts include a count of Accelerator accesses, the total number of POP commands executed by the Accelerator, and the total number of POP commands executed by the Software Interpreter. For statistics taken while the Accelerator is off the first two counts are zero. For statistics taken while the Accelerator is on all counts are significant. Comparing the counts for the ranges with the SWITCH count shows what proportion of the total accesses is in each range. Comparing the TENX POP count with the SOFT POP count shows what proportion of the work the Accelerator is doing.

Table 4-2. Additional Statistics

Report Label	Statistic accumulated
SWITCH	Number of times the Ten X 99 was brought into execution.
TENX POP	Total number of POP commands executed by the Ten X 99.
SOFT POP	Total number of POP commands executed by the software interpreter.

THE TENX COMMAND

4.4 EXAMPLES OF STATISTICS

4.4.1 and 4.4.2 show two examples of statistics for programs. Two sets of statistics are shown for each benchmark. One with the Accelerator disabled and one with the Accelerator enabled.

4.4.1 AN I/O INTENSE ENVIRONMENT

These statistics were gathered from an I/O intense multiprogramming environment. Both interactive and batch jobs were executing.

The first report shows five minutes of statistics taken with the Accelerator disabled.

RTE.....0	ACC.....52	ADD.....13048	AVS.....12796
CAL.....1	CLS.....0	COS.....12729	CRT.....683
CSE.....0	DEF.....0	DID.....0	DIS.....682
EXP.....17867	INS.....0	JAF.....25596	JAR.....16508
JAT.....23010	JEX.....0	JOD.....168	JUN.....29418
LOB.....14465	LOD.....57083	LOW.....1641	LTW.....168
MAV.....12796	MUD.....0	OPN.....0	PUP.....0
REF.....680	REN.....0	RFL.....0	RNL.....0
ROD.....0	RWF.....0	SAC.....1	SAE.....2321
SDE.....2043	SIZ.....1	SLI.....683	SNR.....1361
SPO.....683	SSR.....26889	STD.....22636	STF.....0
STP.....0	STW.....16676	SUD.....0	SYS.....1641
TAC.....0	TDE.....18929	TDG.....12796	TDL.....679
TNC.....0	TSE.....15521	TSG.....0	TSL.....0
TSW.....0	UNL.....0	WRF.....1641	SRK.....0
ADN.....0	DIN.....0	MUN.....0	SUN.....0
SCERR.....0	SCTD.....0	SEXIT.....0	SAADV.....1613
SBADV.....0	SAADV.....28	SBADV.....0	SDEFADV.....0
SCRT.....0	SDATE.....0	SDAY.....0	STIME.....0
SFRK.....0	SNEG.....0	SNO-OV.....0	SOVAN.....0
SCRTO.....0	001 POP.....0	002 POPS.....0	004 POPS.....0
008 POPS.....0	016 POPS.....0	032 POPS.....0	064 POPS.....0
128 POPS.....0	256 POPS.....0	512 POPS.....0	SWITCH.....0
TENX POP.....0	SOFT POP.363892		
DATE...04/19/84	START..10:35.23	STOP...10:40.32	TIME...00:05.09

THE TENX COMMAND

The second report shows five minutes of statistics with the Accelerator enabled.

RTE.....0	ACC.....87	ADD.....0	AWS.....0
CAL.....1	CLS.....0	COS.....0	CRT.....0
CSE.....0	DEF.....0	DID.....0	DIS.....3387
EXP.....0	INS.....0	JAF.....0	JAR.....0
JAT.....0	JEX.....0	JOD.....0	JUN.....0
LOB.....0	LOD.....0	LOW.....0	LTW.....0
MAW.....0	MUD.....0	OPN.....0	PUP.....0
REF.....3386	REN.....0	RFL.....0	RNL.....0
ROD.....0	RWF.....0	SAC.....0	SAE.....0
SDE.....0	SIZ.....1	SLI.....0	SNR.....0
SPO.....0	SSR.....0	STD.....0	STF.....0
STP.....0	STW.....0	SUD.....0	SYS.....0
TAC.....0	TDE.....0	TDG.....0	TDL.....0
TNC.....0	TSE.....0	TSG.....0	TSL.....0
TSW.....0	UNL.....0	WRF.....8182	SRK.....0
ADN.....0	DIN.....0	MUN.....0	SUN.....0
SCERR.....0	SCTD.....0	SEXIT.....0	SAADV.....0
SBADV.....0	SAADV.....0	SBADV.....0	SDEFADV.....0
SCRT.....0	SDATE.....0	SDAY.....0	STIME.....0
SFRK.....0	SNEG.....0	SNO-OV.....0	SOVAN.....0
SCRT0.....0	001 POP.....19	002 POPS.....0	004 POPS...3388
008 POPS....564	016 POPS...4701	032 POPS.....62	064 POPS...3401
128 POPS....457	256 POPS...2365	512 POPS....549	SWITCH....15507
TENX POP..1819083	SOFT POP..15044		

DATE...04/19/84 START..11:01.34 STOP...11:06.43 TIME...00:05.11

Notice that the total number of POP commands executed in five minutes using the Accelerator is over four times the number executed in the same length of time without the Accelerator. Thus for this multiprogramming environment the Ten X provided over 4 times the throughput.

4.4.2 A COMPUTE INTENSE PROGRAM

These statistics were gathered from a program with relatively little I/O. The program reads data from a file, computes a financial summary, formats a report, and writes the report to another file. The program ran in 9 minutes and 45 seconds on a 990/10 and in 32 seconds on the same 990/10 with a Ten X 99 installed. It ran 18 times as fast with the Accelerator.

THE TENX COMMAND

Statistics for the full run time with the Accelerator disabled.

RTE.....0	ACC.....1	ADD.....30169	AWS.....75965
CAL.....1	CLS.....3	COS.....7368	CRT.....2
CSE.....0	DEF.....0	DID.....2082	DIS.....1
EXP.....8309	INS.....0	JAF.....23376	JAR.....7785
JAT.....38480	JEX.....0	JOD.....130	JUN.....18902
LOB.....130502	LOD.....102430	LOW.....741	LTW.....130
MAV.....129748	MUD.....2128	OPN.....3	PUP.....0
REF.....172	REN.....0	RFL.....0	RNL.....0
ROD.....4156	RWF.....0	SAC.....7	SAE.....2743
SDE.....6448	SIZ.....1	SLI.....2	SNR.....3
SPO.....2	SSR.....9007	STD.....36009	STF.....0
STP.....1	STW.....7915	SUD.....950	SYS.....750
TAC.....0	TDE.....41865	TDG.....13386	TDL.....4767
TNC.....0	TSE.....1666	TSG.....0	TSL.....0
TSW.....0	UNL.....0	WRF.....741	SRK.....0
ADN.....0	DIN.....0	MUN.....0	SUN.....0
SCERR.....0	SCTD.....0	SEXIT.....0	SAADV.....728
SBADV.....0	SAADV.....13	SBADV.....0	SDEFADV.....0
SCRT.....8	SDATE.....1	SDAY.....0	STIME.....0
SFRK.....0	SNEG.....0	SNO-OV.....0	SOVAN.....0
SCRT0.....0	001 POP.....0	002 POPS.....0	004 POPS.....0
008 POPS.....0	016 POPS.....0	032 POPS.....0	064 POPS.....0
128 POPS.....0	256 POPS.....0	512 POPS.....0	SWITCH.....0
TENX POP.....0	SOFT POP.708839		

DATE...04/19/84 START..13:45.12 STOP...13:54.57 TIME...00:09.45

Statistics for the full run time with the Ten X enabled.

RTE.....0	ACC.....1	ADD.....0	AWS.....0
CAL.....1	CLS.....3	COS.....71	CRT.....0
CSE.....0	DEF.....0	DID.....0	DIS.....1
EXP.....0	INS.....0	JAF.....0	JAR.....0
JAT.....0	JEX.....0	JOD.....0	JUN.....0
LOB.....0	LOD.....0	LOW.....0	LTW.....0
MAV.....0	MUD.....0	OPN.....3	PUP.....0
REF.....172	REN.....0	RFL.....0	RNL.....0
ROD.....0	RWF.....0	SAC.....0	SAE.....0
SDE.....0	SIZ.....1	SLI.....0	SNR.....0
SPO.....0	SSR.....0	STD.....0	STF.....0
STP.....1	STW.....0	SUD.....0	SYS.....1
TAC.....0	TDE.....0	TDG.....0	TDL.....0
TNC.....0	TSE.....0	TSG.....0	TSL.....0
TSW.....0	UNL.....0	WRF.....741	SRK.....0
ADN.....0	DIN.....0	MUN.....0	SUN.....0
SCERR.....0	SCTD.....0	SEXIT.....0	SAADV.....0
SBADV.....0	SAADV.....0	SBADV.....0	SDEFADV.....0
SCRT.....0	SDATE.....1	SDAY.....0	STIME.....0
SFRK.....0	SNEG.....0	SNO-OV.....0	SOVAN.....0
SCRT0.....0	001 POP.....54	002 POPS.....5	004 POPS.....16
008 POPS....174	016 POPS.....53	032 POPS.....38	064 POPS.....25
128 POPS....557	256 POPS.....73	512 POPS...1127	SWITCH.....2122
TENX POP.707843	SOFT POP....996		

DATE...04/19/84 START..14:33:03 STOP...14:33.35 TIME...00:00.32

SECTION 5

TROUBLESHOOTING

This section can help you to identify and correct problems. It contains tables of common symptoms, probable causes, and solutions. Table 5-1 lists problems that can occur during the generation of the system. Table 5-2 lists installation problems that can occur after the Ten X COBOL Accelerator is installed.

TROUBLESHOOTING -- WHAT, WHERE, WHEN, WHO, AND WHY.

The key to troubleshooting is to have a concise problem description. A good problem description helps you determine why the problem is occurring and once you know that it is easy to fix the problem. Below are questions you can ask to formulate a concise problem description.

WHEN does the problem occur?

- when I run ANY COBOL program?
- when I run a CERTAIN COBOL program?
- when I use my 931 terminal?
- when I execute TXL and after the green light flashes?

WHAT is the release of the operating system, COBOL, and Ten X?

- often problems are fixed in a later release.
- Ten X microcode release is determined by the date in the microcode file. Do a Show File (SF) on .TENX.MICRO.XXXX and scroll to the second page. The date in the middle of the second page indicates the version.

WHO wrote the program or performed the installation?

- This person will have more information.

WHERE do the error messages come from?

- This is very, very important. Certainly, SVC ERROR 15, COBOL ERROR 15, and TIFORM ERROR 15 have very different meanings.

TROUBLESHOOTING

Table 5-1. System Generation Problems

Symptom	Probable Cause	Solution
XGEN -- Error at the start.	Wrong file name or system files missing.	Check file pathnames. Check directory for system files.
ALGS -- Error at the start.	Wrong file name or system file missing.	Check file pathnames. Check directory for system files.
ALGS fails with assembly error.	Symbols undefined.	Consult a Systems Analyst.
ALGS fails with linking errors.	Undefined symbols.	TEN X XOP defined incorrectly. Verify XOP parameters for XGEN (Section 1) or XGSU (Section 2). Verify linking of TEN X XOP.
PGS finishes sooner than usual.	Synonym table overflow.	Delete user synonyms and restart PGS.
Sysgen fails with SYSTEM ROOT error.	System table area too large.	Generate system with reduced table area size.
LONGEST OVERLAY PATH EXCEEDS >F800	System table area too large.	* See solution listed below.

* Solution for calculating table size reduction.

1. Do a show file (SF) on the system link map .S\$SYSGEN.TENX.LINKMAP
2. For DX10 3.6 with 931 or 940: Find the origin and length of DSR93x where x = A,B OR C. Perform the following calculation: T will be the amount of reduction of the sysgen TABLE parameter. Use the SCI show value (SV) command to perform the following calculations:

$$T = (\text{<origin of DSR93x>} + \text{<length of DSR93x>} - \text{>F800}) / 2$$

3. For DX10 3.5 with 940: Find the origin and length of DSR940.

$$T = (\text{<origin of DSR940>} + \text{<length of DSR940>} + \text{>7D0} - \text{>F800}) / 2$$

4. All other cases: Find the longest PHASE 2 OVERLAY.

$$T = (\text{<origin of OVERLAY>} + \text{<length of OVERLAY>} - \text{>F800}) / 2$$

PROBLEM - SYSTEM WILL NOT BOOT.

1. Check for broken TLAG chain, shared TILINE address or interrupt.
2. Check chassis power load.
3. Are you sure that the genned system matches the hardware config? Use the TENX command to verify the TILINE address

TROUBLESHOOTING

and interrupt level defined for the Accelerator.
Use the LDC command to verify the addresses and interrupts for all other devices.

4. Are you sure that the TILINE address switches on the Ten X board match the TILINE address in your gen?
5. Possible bad board. See if the system will boot with the Ten X board removed (ensure no other master controllers are located below it).

PROBLEM - PCOBJ ERROR.

1. Check COBOL version. COBOL 3.2 not supported.
2. Sometime the origin of the COBOL runtime is not 0. This usually happens when DBMS is installed. Find the origin of the runtime and retro-fit the patch for this situation. See Appendix A.

PROBLEM - SYSTEM HANGS WHEN TXL IS PERFORMED.

IF ---- The green light never came on.
IF ---- System also hangs when TENX command is performed.
IF ---- DNOS or DX10 3.6.1 or later
CAUSE - TEO1 special device is not defined or misspelled.
 TEO1 is the very 1st device defined in the system
 (this is a bug and is fixed in Ten X release 3.1.0).
OTHERWISE - broken TILINE access granted chain or shared
 interrupt or shared TILINE device or bad unit.
 BE SURE the generated system matches the hardware
 configuration.

PROBLEM - SYSTEM CRASHES WHEN TXL IS PERFORMED.

CRASH CODE	CAUSE
-----	-----
13 thru 1F	The SYSGEN does not match the hardware configuration.
>20(DX10)	Broken TLAG chain, shared TILINE address or interrupt.
>61(DNOS)	Memory parity error.
>62(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.
>63(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.
>65(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.

PROBLEM - TXL ERRORS. TEN X 99 FAILED DIAGNOSTICS. BOARD nn. CODE =>XXXX

CODE	CAUSE
----	-----
02	Shared TILINE address.
03	Shared TILINE address or wrong TILINE address.
04	The SYSGEN you are booted under does not include Ten X.
12	Faulty Ten X 99 or broken TILINE access granted chain.
31	Interrupt not received by 990. The SYSGEN does not match the interrupt configuration. No interrupt is

TROUBLESHOOTING

- hooked up to the slot in which Ten X is installed.
- FF Faulty Ten X unit or broken TILINE access granted chain.
- XX Any other error. replace unit. If 2nd unit fails check for broken TILINE access granted chain, shared TILINE address, and shared interrupt.

PROBLEM - SYSTEM HANGS AFTER A PERIOD OF RUNNING.

- o First check for broken TLAG chain, shared TILINE address or interrupt.
- o Check power.
- o Press halt, then ST under display - write down the lights.
- o Force a crash.
- o May be bad board.

PROBLEM - SYSTEM CRASHES AFTER A PERIOD OF RUNNING.

CRASH CODE	CAUSE
-----	-----
>20(DX10 3.6.1)	Ten X release 3.0.0 installed.
>20(DX10)	Broken TLAG chain, shared TILINE address or interrupt.
>20(DX10)	Memory parity error.
>61(DNOS)	Memory parity error.
>62(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.
>63(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.
>65(DNOS)	Broken TLAG chain, shared TILINE address or interrupt.
OTHER	Problem not related to Ten X.

PROBLEM - TEN X NOT WORKING. GREEN LITE DOESN'T COME ON.

1. Didn't do TXL.
2. Program files not patched.
3. Program files are patched but patch is not in memory.
4. Somebody turned Ten X off using TENX command.
5. Release 3.0.x .TENX directory installed on system with release 2.x.x SYSGEN. Microcode loaded = Test. Can't turn TEN X on from TENX command. TXL leaves TEN X off.

PROBLEM - FEW OR ONLY ONE COBOL PROGRAMS RECEIVE RUNTIME ERRORS OR GET INCORRECT RESULTS.

IF - same problem occurs with Ten X off then it is a programmer error.

OTHERWISE - microcode problem

PROBLEM - ALL OR MOST COBOL PROGRAMS RECEIVE RUNTIME ERRORS.

IF --- system log shows task error = >4 then you are booted off a sysgen that does not include TEN X.

OR --- the microcode version downloaded does not match your your COBOL release.

ELSE - broken TLAG chain, shared TILINE address or bad board

PROBLEM - COBOL DEBUGGER DOESN'T WORK WITH TEN X ON.

CAUSE - It's not designed to. Link the COBOL program you wish to debug with it's own unpatched copy of the runtime.

APPENDIX A

ERROR MESSAGES

A.1 NORMAL COMPLETION MESSAGES

The Ten X commands return messages when they complete satisfactorily. These are the completion messages:

ITX Command:

*** TENX - NORMAL TERMINATION ***

This message indicates that the Ten X software modules have been assembled and linked.

WAIT FOR BACKGROUND TO COMPLETE

This message indicates that the batch stream for ITX is being executed. This is an information message; not an error message.

PCOBJ Command:

FILE=<filename> PROC=<name or ID> ** PATCHED **

This message indicates that the named procedure on the specified program file has been patched successfully.

TXL Command:

TEN X 99 INSTALLED AT INTERRUPT YY.

This message indicates that a Ten X 99 has been successfully installed in the system at interrupt YY. The Accelerator has successfully passed all validation tests.

UPCOBJ Command:

FILE=<filename> PROC=<name or ID> ** UNPATCHED **

This message indicates that the previous contents of the named procedure on the specified program file have been successfully restored.

A.2 ERROR MESSAGES

The Ten X commands display information messages as appropriate and error messages when an error is detected. This section lists the error messages and their meanings.

ERROR MESSAGES

ITXP Command:

ONLY DX10 OR DNOS

This message indicates that an invalid response was entered for the operating system prompt. Enter the command again specifying either DX10 or DNOS.

ITX Command:

QUEUE LENGTH MUST BE < 9

This message indicates that the response to the queue length prompt was greater than eight. Enter the command again using a queue length of 1 through 8.

QUEUE LENGTH MUST BE > 0

This message indicates that the response to the queue length prompt was less than zero. Enter the command again using a queue length of 1 through 8.

TILINE Address must be >F800

This message indicates that either no TILINE address was entered or an invalid TILINE address was entered. Enter the command again using a TILINE address greater than >F800 for each Accelerator.

*** TENX - DNOSDEF ASSEMBLY ERROR = nnnn ***
*** TENX - DNOSDEF ASSEMBLY ERROR ***

These messages indicate that the assembly of the source module built by the ITX command failed. Consult the messages and codes manual for your operating system to determine the correct definition of error code nnnn. Examine the listing .TENX.DNOS.LST.DNOSDEF. Consult the Model 990 Assembly Language Reference Manual. Correct the error and enter the ITX command again.

*** TENX - DX10DEF ASSEMBLY ERROR = nnnn ***
*** TENX - DX10DEF ASSEMBLY ERROR ***

These messages indicate that the assembly of the source module built by the ITX command failed. Consult the messages and codes manual for your operating system to determine the correct definition of error code nnnn. Examine the listing .TENX.DX10.LST.DX10DEF. Consult the Model 990 Assembly Language Reference Manual. Correct the error and enter the ITX command again.

ERROR MESSAGES

*** TENX - TENXOP LINK ERROR = nnnn ***
*** TENX - TENXOP LINK ERROR ***

These messages indicate that the linking of the Ten X XOP failed. Consult the messages and codes manual for your operating system for the definition of error code nnnn. Examine .TENX.DNOS.LST.DNOSXOP or .TENX.DX10.DX10XOP. Consult the link editor reference manual. Correct the error and enter the ITX command again.

*** TENX - DSRTENX LINK ERROR = nnnn ***
*** TENX - DSRTENX LINK ERROR ***

These messages indicate that the linking of the Ten X DSR failed. Consult the messages and codes manual for your operating system for the definition of error code nnnn. Examine .TENX.DNOS.LST.DSRTENX. Consult the link editor reference manual. Correct the error and enter the ITX command again.

PCOBJ Command:

FILE=<filename> ** INVALID PATHNAME **

This message indicates that the file pathname for the program file is not available. Verify the pathname and re-enter the command.

PROC=<name or ID> ** INVALID PROCEDURE NAME OR ID **

This message indicates that the program file does not contain the procedure name or ID shown. Verify the procedure name or ID and re-enter the command.

FILE=<filename> PROC=<name or ID> ** VERIFICATION ERROR **

This message indicates that the address specified in the attempted patch did not contain the value expected. The patch was not made. Verify COBOL release and re-enter the command if incorrect.

Sometimes the origin of COBOL is not 0. This usually happens when DBMS is installed. Use Map Program File(MPF) SCI command to find the origin of the COBOL runtime. If the origin of the COBOL runtime = X, then the following show retro-fitted patches for all supported versions of COBOL.

DX10 COBOL RELEASE 3.3
MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>148+X
V=>04C9
D=>2C81
R=N

ERROR MESSAGES

DX10 COBOL RELEASE 3.4

MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>18C+X
V=>04C9
D=>2C81
R=N

DX10 COBOL RELEASE 3.5

MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>18C+X
V=>04C9
D=>2C81
R=N

DNOS COBOL RELEASE 3.3

MPI PF=<program file name>
MT=PROCEDURE,
MN=<runtime COBOL procedure name>
A=>18C+X
V=>04C9.
D=>2C81.
R=N

DNOS COBOL PLUS RELEASE 3.3

MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>1BD0+X
V=0.0.0.0.0.0
D=>2C81.>2F60.>1AE4+X.1.>460.>182+X
MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>17C+X
V=>2F60.>1AE4+X
D=>460.>1BD0+X

DNOS COBOL RELEASE 3.4

MPI PF=<program file name>
MT=PROCEDURE
MN=<runtime COBOL procedure name>
A=>18C+X
V=>04C9
D=>2C81
R=N

ERROR MESSAGES

DNOS COBOL PLUS RELEASE 3.4

```

MPI    PF=<program file name>
        MT=PROCEDURE
        MN=<runtime COBOL procedure name>
        A=>3DB8+X
        V=>DEAD.>DEAD.>DEAD.>DEAD.>DEAD.>DEAD
        D=>2C81.>2F60.>1B06+X.1.>460.>182+X
MPI    PF=<program file name>
        MT=PROCEDURE
        MN=<runtime COBOL procedure name>
        A=>17C+X
        V=>2F60.>1B06+X
        D=>460.>3DB8+X

```

DNOS COBOL RELEASE 3.5

```

MPI    PF=<program file name>
        MT=PROCEDURE
        MN=<runtime COBOL procedure name>
        A=>18C+X
        V=>04C9
        D=>2C81
        R=N

```

DNOS COBOL PLUS RELEASE 3.5

```

MPI    PF=<program file name>
        MT=PROCEDURE
        MN=<runtime COBOL procedure name>
        A=>3DB8+X
        V=>DEAD.>DEAD.>DEAD.>DEAD.>DEAD.>DEAD
        D=>2C81.>2F60.>1B06+X.1.>460.>182+X
MPI    PF=<program file name>
        MT=PROCEDURE
        MN=<runtime COBOL procedure name>
        A=>17C+X
        V=>2F60.>1B06+X - 1B26
        D=>460.>3DB8+X

```

FILE=<filename> ** WRITE PROTECTED **

This message indicates that the named program file is write protected. Enter an MFP command to remove write protection and re-enter command.

FILE=<filename> PROC=<name or ID> CC=nnnn

This message indicates that the patch command returned completion code nnnn. Consult the messages and codes manual for your operating system for the definition of the completion code.

TENX Command:

invalid selection

The number entered is not one of the nine

ERROR MESSAGES

functions of the TENX Utility. Enter a number in the range of 1 through 9.

Report file is empty

The listing access file is empty. Enter commands in this sequence when producing a statistics report:

- 5 Start Statistics Collection
- 6 Interrupt Statistics Collection
- 8 Produce Statistics Report
- 9 Display Statistics Report

After entering the Interrupt Statistics Collection command, you can enter a Restart Statistics Collection command (before producing a report) without losing the totals you have. You must enter another Interrupt Statistics Collection command before producing the report.

...microcode not loaded

An enable board function has been attempted while no microcode has been loaded.

TXL Command:

ERROR ON FILE OPEN/ASSIGN. CODE=>XX.

An error was detected while attempting to open the microcode file. CODE XX is the hexadecimal error code returned by the operating system except for one case in which a code of >FF is generated by TXL. This is when the microcode file record size is not 1026 as expected for a valid file. Consult the messages and codes manual for your operating system for the definitions of other values of XX. The most probable cause is an invalid pathname supplied to the TXL command. Other causes may be related to disk errors.

ERROR READING MICROCODE. CODE=>XX.

An error was detected by the operating system while attempting to read a record of the microcode file into memory. CODE XX is the error code value returned by the operating system. This error is probably caused by a disk malfunction, a media compatibility problem, or some file copy process failure. The error can occur while reading either the diagnostic microcode or the COBOL microcode.

ERROR MESSAGES

CHECKSUM ERROR READING MICROCODE.

A checksum error was detected by the loader while reading microcode into memory. Restore the microcode file from backup. The error indicates that data in a microcode record has been changed.

MICROCODE COMPARE FAILED. BOARD nn.

Microcode is transferred first from the input file to memory and then from memory to the Ten X 99 processor memory. To validate the transfer from memory to Ten X the microcode is read back into 990 memory and compared with what was sent to the Accelerator. This error message indicates that some differences occurred. BOARD nn indicates the interrupt level of the board that failed. That board cannot be installed in the system. In multiple board systems the other boards are not affected. This failure can indicate either a hardware failure on a Ten X 99 or improper installation in the 990 chassis. Be sure to check that the TILINE access granted switches/jumpers are properly installed. All slots in the chassis must have either a jumper or a TILINE device. Do not put a TILINE device in a slot with the switch on/jumper in place. Do not allow a slot with the TILINE access granted switch off/jumper missing to be empty or contain a board for a non-TILINE device.

TEN X 99 FAILED DIAGNOSTICS. BOARD nn. CODE=>XXX.

Each Ten X 99 in the system must pass a diagnostic test before the COBOL microcode is installed. This message indicates that the Ten X 99 at interrupt nn did not pass the diagnostic test. The failed Accelerator will have the red FAULT indicator on. The fault condition cannot be cleared without a system IPL. In a multiple Ten X environment the remaining boards are not affected. The failure detected by the diagnostic is indicated by code XXX. Contact your supplier or Ten X Technology Inc. for assistance. The codes returned by the diagnostic are listed in Table A-1. These error codes pertain ONLY to the Ten X 99. They DO NOT affect the operation of the 990 processor or the TI operating system. Error code 031. Interrupt not received by 990, may indicate that the 990 interrupt jumper is not wired correctly or that a Ten X 99 is not installed in the proper slot in the chassis.

TEST VERSION LOADED. CODE = >XXXX

A test version of the microcode has been loaded. The test version number is specified by XXXX.

ERROR MESSAGES

Table A-1. Diagnostic Errors

<p>Note: The first number of the error code represents the type of error. The X indicates an undefined error or an error location.</p>	
CODE	TYPE OF ERROR
004	The GEN does not include Ten X
012	Memory test - check TLAG jumper
013	Read/modify/write
014	Pack
019	INDEX
020	IMMEDIATE FIELD
031	INTERRUPT NOT RECEIVED BY 990
0FF	Wild branch - check TLAG jumper
1XX	SYSTEM TABLE READ
2XX	NIBBLE ROTATE
3XX	JUMP ERRORS
301	JEQ
302	JGT
303	JH
304	JL
306	JLE
307	JLT
308	JNC
309	JOC
310	JNO
311	JOD
400	BRANCH AND LINK
5XX	SIMPLE ALU ERRORS
50X	COMPARE
51X	ADD
52X	SUBTRACT
53X	INCREMENT
530	BY 1
535	BY 2
54X	DECREMENT
540	BY 1
545	BY 2
55X	SUBTRACT-ADD
56X	MULTIPLY
560	UNSIGNED
565	TWO'S COMPLEMENT
57X	DIVIDE
58X	SIMPLE LOGICAL
6XX	BCD ALU ERRORS
610	ADD
620	SUBTRACT
630	SUBTRACT-ADD
670	DIVIDE BY 2
6AX	CONVERSION ERRORS
6A0	B>BCD
6A5	BCD>B

ERROR MESSAGES

INVALID TILINE ADDRESS. BOARD nn. CODE=>XX.

Indicates that the DX10DEF or DNOSDEF source module used to generate the operating system is incorrect. The affected Accelerator at interrupt nn is not available to the system. In a multiple Ten X system the remaining Accelerators are not affected. The code XX is the operating system's task termination error code. Verify that:

- * The TILINE TPCS address switches are set properly.
- * The Ten X 99 is installed at the appropriate interrupt level.

LEVEL 2 XOP NOT INSTALLED. CODE=>XX.

The loader program encountered an error when it issued the level 2 XOP. Code XX is the task termination error code returned by the operating system. Verify that the system you are using has been generated with the Ten X XOP processor (look at the link map for the module TENXOP). Carefully follow the instructions for preparing the XOP. Make sure that you have installed the new system using either TGS or IGS.

TASK TERMINATED IN ERROR. CODE=>XX.

The loader task terminated due to an error detected by the operating system. The code XX is the system task error termination code. See the messages and codes manual for your operating system for the exact definition.

UPCOBJ Command:

FILE=<filename> ** INVALID PATHNAME **

This message indicates that the file pathname for the program file is not available. Verify the pathname and re-enter the command.

PROC=<name or ID> ** INVALID PROCEDURE NAME OR ID **

This message indicates that the program file does not contain the procedure identified. Verify the procedure name or ID and re-enter the command.

FILE=<filename> PROC=<name or ID> ** VERIFICATION ERROR **

This message indicates that the address specified in the attempted unpatch did not contain the value expected. The unpatch was not made. Verify that

ERROR MESSAGES

the named procedure was previously patched. Also verify the COBOL release and re-enter the command if incorrect.

FILE=<filename> ** WRITE PROTECTED **

This message indicates that the named program file is write protected. Enter an MFP command to remove write protection and re-enter command.

FILE=<filename> PROC=<name or ID> CC=nnnn

This message indicates that the unpatch command returned completion code nnnn. Consult the messages and codes manual for your operating system for the definition of the completion code.

APPENDIX B

TEN X 99 COMMANDS

B.1 INTRODUCTION

This appendix lists the commands supplied with your Ten X 99 Accelerator. The commands are listed in alphabetical order. They differ from the examples shown in Sections 1, 2, and 4 in that they show the generic names for responses rather than typical responses. Since there are differences in the commands for DX10 and DNOS the appendix includes a section for each operating system.

The DX10 commands and the paragraphs in which they are described are:

Command -----	Paragraph -----
Find COBOL Runtime (FCR)	B.2.1
Ten X 99 Operating System Definition (ITX)	B.2.2
Install Ten X Procs (ITXP)	B.2.3
Patch COBOL Object (PCOBJ)	B.2.4
Ten X 99 Utility (TENX)	B.2.5
Ten X Load (TXL)	B.2.6
Unpatch COBOL Object (UPCOBJ)	B.2.7

The DNOS commands and the paragraphs in which they are described are:

Command -----	Paragraph -----
Find COBOL Runtime (FCR)	B.3.1
Ten X 99 Operating System Definition (ITX)	B.3.2
Install Ten X Procs (ITXP)	B.3.3
Patch COBOL Object (PCOBJ)	B.3.4
Ten X 99 Utility (TENX)	B.3.5
Ten X Load (TXL)	B.3.6
Unpatch COBOL Object (UPCOBJ)	B.3.6

B.2 DX10 COMMANDS

The DX10 commands are described in the following paragraphs.

B.2.1 FCR Command

[]FCR

Find COBOL Runtime Procedures <RELEASE 3.1>

DIRECTORY: (directory or volume name to search)

LISTING: (filename where results are posted)

The responses are:

TEN X 99 COMMANDS

DIRECTORY:

Enter the directory name or volume name you wish to search for program files which contain COBOL procedures.

LISTING:

Enter the filename where the results of this search will be placed.

Paragraphs 1.8 and 2.8 show examples of this command.

B.2.2 ITX Command

```
[ ] ITX
Ten X 99 Operating System Definition
  QUEUE LENGTH(1-8): 4
    INTERRUPT 3: >0
    INTERRUPT 4: >0
    INTERRUPT 5: >0
    INTERRUPT 6: >0
    INTERRUPT 7: >0
    INTERRUPT 8: >0
    INTERRUPT 9: >0
    INTERRUPT 10: >0
    INTERRUPT 11: >0
    INTERRUPT 12: >0
    INTERRUPT 13: >0
    INTERRUPT 14: >0
    INTERRUPT 15: >0
```

The responses are:

QUEUE LENGTH:

Enter a digit, 1 through 8, for the length of the queue of tasks for the Ten X 99. A typical value is 4.

TILINE ADDRESS:

Enter a TILINE address, >F800 through >FBF0, for each Ten X 99. Enter the address after the prompt for the interrupt level of the slot in which each Accelerator is installed.

Paragraphs 1.4 and 2.4 show examples of this command.

B.2.3 ITXP Command

```
[ ] ITXP
INSTALL TEN X PROCS
  DX10 OR DNOS: DX10
  PROC DIRECTORY: <.S$PROC or alt. directory>
  LISTING ACCESS NAME: <return>
```

The responses are:

PROC DIRECTORY:

Enter the name of the procedure directory on which to install the Ten X procedures. Either .S\$PROC or an alternate procedure directory. When you do not supply a pathname the command installs the procedures on .S\$PROC.

LISTING ACCESS NAME:

Enter the file pathname that is to receive the messages and listing of the command. If you leave this response blank the messages and listing are displayed on the screen.

Paragraphs 1.2 and 2.2 show examples of this command.

B.2.4 PCOBJ Command

```
[ ] PCOBJ
Patch COBOL Runtime
PROGRAM FILE NAME: <filename of program file>
PROCEDURE NAME OR ID: <procedure name or ID>
COBOL RELEASE: 33, 34 or 35
```

The responses are:

PROGRAM FILE NAME:

Enter .S\$PROGA or the filename of a program file that contains a copy of the COBOL runtime procedure.

PROCEDURE NAME OR ID:

Enter RCOBOL or other name for the COBOL runtime procedure. Or enter the ID of the COBOL runtime procedure.

COBOL RELEASE:

Enter 33 for Release 3.3 or 34 for Release 3.4. or 35 for Release 3.5.

Paragraphs 1.8 and 2.8 show examples of this command.

B.2.5 TENX Command

```
[ ] TENX
Ten X 99 Utility
LISTING ACCESS NAME: <file pathname>
```

TEN X 99 COMMANDS

The response is:

LISTING ACCESS NAME:

Enter a file pathname (not a device name) to receive
the statistics reports of TENX.

Paragraph 4.2 shows an example of this command.

B.2.6 TXL Command

```
[ ] TXL
Ten X Load
MICROCODE FILE NAME: .TENX.MICRO.<COBOL Type>
```

The response is:

MICROCODE FILE NAME:

The pathname of a microcode file: .TENX.MICRO.<COBOL
type>. COBOL type is one of the following:

- * DX1033 -- for COBOL Release 3.3
- * DX1034 -- for COBOL Release 3.4
- * DX1035 -- for COBOL Release 3.5

Paragraphs 1.9 and 2.9 show examples of this command.

B.2.7 UPCOBJ Command

```
[ ] UPCOBJ
Unpatch COBOL Runtime
PROGRAM FILE NAME: <filename of program file>
PROCEDURE NAME OR ID: <procedure name or ID>
COBOL RELEASE: 33, 34 or 35
```

The responses are:

PROGRAM FILE NAME:

Enter .S\$PROGA or the filename of a program file that
contains a copy of the COBOL runtime procedure.

PROCEDURE NAME OR ID:

Enter RCOBOL or other name for the COBOL runtime procedure.
Or enter the ID of the COBOL runtime procedure.

COBOL RELEASE:

Enter 33 for Release 3.3, 34 for Release 3.4 or 35 for
Release 3.5.

This command is mentioned in paragraphs 1.8 and 2.8.

B.3 DNOS COMMANDS

The DNOS commands are described in the following paragraphs.

B.3.1 FCR Command

```
[ ]FCR
Find COBOL Runtime Procedure <RELEASE 3.1>
    DIRECTORY: (directory or volume name to search)
    LISTING: (filename where results are posted)
```

The responses are:

DIRECTORY:

Enter the directory name or volume name you wish to search for program files which contain COBOL procedures.

LISTING:

Enter the filename where the results of this search will be placed.

B.3.2 ITX Command

```
[ ]ITX
Ten X 99 Operating System Definition
    DNOS VERSION (11, 12 or 13):
        QUEUE LENGTH(1-8): 4
        NUMBER OF TERMINALS:
            INTERRUPT 3: >0
            INTERRUPT 4: >0
            INTERRUPT 5: >0
            INTERRUPT 6: >0
            INTERRUPT 7: >0
            INTERRUPT 8: >0
            INTERRUPT 9: >0
            INTERRUPT 10: >0
            INTERRUPT 11: >0
            INTERRUPT 12: >0
            INTERRUPT 13: >0
            INTERRUPT 14: >0
            INTERRUPT 15: >0
```

The responses are:

DNOS VERSION:

Enter 11 for version 1.1, 12 for version 1.2 or 13 for version 1.3.

QUEUE LENGTH:

Enter a digit, 1 through 8, for the length of the queue of tasks for the Ten X 99. A typical value is 4.

TEN X 99 COMMANDS

NUMBER OF TERMINALS:

Enter the number of terminals on your system.

TILINE ADDRESS:

Enter a TILINE address. >F800 through >FBFO. for each Ten X 99. Enter the address after the prompt for the interrupt level of the slot in which each Accelerator is installed.

Paragraphs 1.4 and 2.4 show examples of this command.

B.3.3 ITPX Command

```
[ ] ITPX
INSTALL TEN X PROCS
    DX10 OR DNOS: DNOS
    PROC DIRECTORY: <.S$CMDS or alt. directory>
    LISTING ACCESS NAME:
```

The responses are:

PROC DIRECTORY:

Enter the name of the procedure directory in which to install the Ten X procedures. Either .S\$CMDS or an alternate procedure directory. When you do not supply a pathname the command installs the procedures in .S\$CMDS.

LISTING ACCESS NAME:

Enter a file pathname to receive the messages and listing of the command. If you leave this response blank the messages and listing are displayed on the screen.

Paragraphs 1.2 and 2.2 show examples of this command.

B.3.4 PCOBJ Command

```
[ ] PCOBJ
Patch COBOL Runtime
    PROGRAM FILE NAME: <filename of program file>
    PROCEDURE NAME OR ID: <procedure name or ID>
    COBOL RELEASE: 33, 34, 35, 33P, 34P or 35P
```

The responses are:

PROGRAM FILE NAME:

Enter .S\$SHARED or the filename of a program file that contains a copy of the COBOL runtime procedure.

TEN X 99 COMMANDS

PROCEDURE NAME OR ID:

Enter RCOBOL or other name for the COBOL runtime procedure.
Or enter the ID of the COBOL runtime procedure.

COBOL RELEASE:

Enter 33 for Release 3.3, 34 for Release 3.4, 35 for
Release 3.5, 33P for COBOL + Release 3.3, 34P for COBOL +
Release 3.4 or 35P for COBOL + Release 3.5.

Paragraphs 1.8 and 2.8 show examples of this command.

B.3.5 TENX Command:

```
[ ] TENX
Ten X 99 Utility
LISTING ACCESS NAME: <file pathname>
```

The response is:

LISTING ACCESS NAME:

Enter a file pathname (not a device name) to receive
the statistics reports of TENX.

Paragraph 4.2 shows an example of this command.

B.3.6 TXL Command

```
[ ] TXL
Ten X Load
MICROCODE FILE NAME: .TENX.MICRO.<COBOL Type>
```

The response is:

MICROCODE FILE NAME:

The pathname of a microcode file: .TENX.MICRO.<COBOL
type>. COBOL type is one of the following:

- * DNOS33 -- for COBOL Release 3.3
- * DNOS34 -- for COBOL Release 3.4
- * DNOS35 -- for COBOL Release 3.5
- * DNOS33P -- for COBOL + Release 3.3
- * DNOS34P -- for COBOL + Release 3.4
- * DNOS35P -- for COBOL + Release 3.5

Paragraphs 1.9 and 2.9 show examples of this command.

TEN X 99 COMMANDS

B.3.7 UPCOBJ Command

```
[ ] UPCOBJ
Unpatch COBOL Runtime
  PROGRAM FILE NAME: <filename of program file>
  PROCEDURE NAME OR ID: <procedure name or ID>
  COBOL RELEASE: 33, 34, 35, 33P, 34P or 35P
```

The responses are:

PROGRAM FILE NAME:

Enter .S\$SHARED or the filename of a program file
that contains a copy of the COBOL runtime procedure.

PROCEDURE NAME OR ID:

Enter RCOBOL or other name for the COBOL runtime procedure.
Or enter the ID of the COBOL runtime procedure.

COBOL RELEASE:

Enter 33 for Release 3.3, 34 for Release 3.4, 35 for
Releases 3.5, 33P for Release 3.3+, 34P for Release 3.4+
or 35P for Release 3.52+.

This command is mentioned in paragraphs 1.8 and 2.8.

POP COMMAND DESCRIPTIONS

APPENDIX C

POP COMMAND DESCRIPTIONS

Mnemonics correspond to the report label.

Mnemonic	Ten X Executable	Description
RTE	(not a POP)	COBOL RunTime Error detected
ACC	No	Accept string from keyboard
ADD	Yes	Add Decimal
AVS	Yes	Add word to subscript
CAL	No	Call subprogram
CLS	No	Close file
COS	Yes/No	Copy string (not executed if > 256 bytes)
CRT	Yes	CRT access setup
CSE	Yes/No	Copy string edited (not executed if > 256 bytes)
DEF	No	Delete file
DID	Yes	Divide Decimal
DIS	No	Display string to CRT
EXP	Yes	Exit paragraph
INS	Yes	Inspect string
JAF	Yes/No	Jump if answer is false..
JAR	Yes/No	Jump and save return.....: not executed if
JAT	Yes/No	Jump if answer is true...: the jump is to
JEX	Yes/No	Jump to exit.....: a non-memory
JOD	Yes/No	Jump or decrement.....: resident overlay
JUN	Yes/No	Jump unconditionally.....:
LOB	Yes	Load binary word
LOD	Yes	Load decimal
LOW	Yes	Load word
LTW	Yes	Load temporary word
MAW	Yes	Multiply and add word
MUD	Yes	Multiply decimal
OPN	No	Open file
PUP	No	Perform use procedure
REF	No	Read file
REN	No	Read next
RFL	No	Read file no lock
RNL	No	Read next no lock
ROD	Yes	Round Decimal
RWF	No	Write file
SAC	Yes	Set action code
SAE	Yes	Set alternate exit
SDE	Yes	Store decimal edited
SIZ	Yes	Set size
SLI	Yes	Set line
SNR	Yes	Set numeric reference
SPO	Yes	Set position

POP COMMAND DESCRIPTIONS

POP Command Descriptions (Continued)

Mnemonic	Ten X Executable	Description
SSR	Yes	Set string reference
STD	Yes	Store decimal
STF	No	Start file
STP	No	Stop program
STW	Yes	Store temporary word
SUD	Yes	Subtract decimal
SYS	Yes/No	*System instruction class (expanded below)
TAC	Yes	Test alpha class
TDE	Yes	Test decimal equal
TDG	Yes	Test decimal greater
TDL	Yes	Test decimal less
TNC	Yes	Test numeric class
TSE	Yes	Test string equal
TSG	Yes	Test string greater
TSL	Yes	Test string less
TSW	Yes	Test switch
UNL	No	Unlock file
WRF	No	Write file
SRK	Yes	Set relative key
ADN	Yes	Add normalized
DIN	Yes	Divide normalized
MUN	Yes	Multiply normalized
SUN	Yes	Subtract normalized
*SCERR	No	System compilation error
*SCTD	Yes	System convert to decimal
*SEXIT	No	System exit program
*SAADV	Yes	System after advancing
*SBADV	Yes	System before advancing
*SAADV P	Yes	System after advancing page
*SBADV P	Yes	System before advancing page
*SDEFADV	Yes	System default advancing
*SCRT	Yes	System CRT series
*SDATE	No	System date
*SDAY	No	System day
*STIME	No	System time
*SFRK	No	System fetch relative key
*SNEG	Yes	System negate accumulator
*SNO-OV	Yes	System set no overflow
*SOVAN	Yes	System set overflow answer
*SCRTO	No	System CRT option

* Further breakout of the SYS POP commands.

POP COMMAND GENERATION

APPENDIX D

POP COMMAND GENERATION

D.1 INTRODUCTION

This appendix consists of examples of the POP commands which are generated for some general COBOL statements. These are the approximate resulting POP commands.

D.2 ARITHMETIC

Statement	Generated POP Commands
-----	-----
ADD...TO...	LOD ADD ...repeat ADD for each ...item STD
DIVIDE...BY...GIVING...	LOD SCTD...if binary DID STD
COMPUTE...	uses varying combinations of LOD STD ADN DIN MUN SUN SNEG
...ROUNDED...ON SIZE ERROR	ROD SAE STD JUN*

D.3 CONTROL

Statement	Generated POP Command
-----	-----
GO TO...	JUN*
GO TO...DEPENDING ON...	LOB JOD*

* If jump is to a non-resident overlay it is not executed by Ten X.

POP COMMAND GENERATION

	JOD*
	.
	.
	JOD*
PERFORM...THROUGH...	LOB
	STW
	LTW
	JOD*
	STW
	JAR*
	JUN*
...VARYING...FROM...BY...UNTIL...	LOD
	STD
	.
	...evaluate condition
	.
	JAT*
	JAR*
	LOD
	ADD
	STD
	JUN*
IF...	.
	...evaluate condition
	.
	JAT* or JAF*

D.4 DATA MANIPULATION

Statement	Generated POP Command
-----	-----
INSPECT...REPLACING...	
.LEADING...BY...AFTER INITIAL...	SR
	COS
	SSR
	COS
	SSR
	COS
	SSR
	COS
	SSR
	INS
MOVE...TO...	LOD
	STD
	...repeat for each pair
	if CORRESPONDING

D.5 LOW-VOLUME I/O

Statement	Generated POP Commands
-----	-----
ACCEPT...LINE...POSITION...SIZE...	CRT

POP COMMAND GENERATION

SLI
SPO
SIZ
SAC
SCRTO
ACC*
STD

DISPLAY...LINE...POSITION...SIZE... CRT
SLI
SPO
SIZ
SAC
SCRTO
DIS*

D.6 FILE I/O

Statement -----	Generated POP Command -----
OPEN	SAE SAC OPN*
READ...NEXT...INTO...	SAE SAC LOD SRK RFL* SFRK* STD JAT** LOD STD JUN**
REWRITE...FROM...	LOD STD SAE LOW LOD SRK RVF* JAT** SAE PUP*
START...KEY...	SAE SAC LOD SRK

- * POP command not executed by Ten X.
- ** If jump is to a non-resident overlay it is not executed by Ten X.

POP COMMAND GENERATION

	LOW
	STF*
	JAT**
	SAE
	PUP*
UNLOCK...	SAE
	UNL*
WRITE..FROM... (non-print device)	LOD
	STD
	SAE
	LOW
	STD
	SRK
	WRF*
	SFRK*
	STD
	JAT**
	SAE
	PUP*
WRITE...FROM... (print device)	LOD
	STD
	SAE
	LOB
	SAADVP or other control
	LOW
	WRF*
additional for all subscripts	LOB
	MAW ...subscript 1
	.
	...repeat for more dimen.
	.
	LOB
	MAW ...subscript N
	AWS

* POP command not executed by Ten X.

** If jump is to a non-resident overlay it is not executed by Ten X.

TEN X 99 TILINE INTERFACE

APPENDIX E

TEN X 99 TILINE INTERFACE

E.1 INTRODUCTION

This appendix describes the interface of the Ten X 99 Processor with Texas Instruments 990 series computers. This interface is based on the use of the TILINE Peripheral Control Space (TPCS) beginning at the address set in the switches on the Ten X 99. The Ten X 99 Processor functions as both a master and a slave device on the TILINE bus.

If your computer is equipped with a Programmer's Panel you can set the TILINE address in the address register and read and write to the Ten X 99 Accelerator.

E.2 TPCS STATUS REGISTER

The TPCS of the Ten X 99 Processor status register contains various status and control bits. The Ten X 99 status register is located at the TILINE address plus 8. The following figure shows the assignments of the bits in the status register. Subsequent paragraphs describe each bit.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+-----+-----+-----+-----+-----+-----+															
IDL		CMP		ONL		FLT		IEN		RST		RWC		0	
								NEX		MER		TO		0	
												STP		0	
														0	
														0	
+-----+-----+-----+-----+-----+-----+															

* These bits generate an interrupt to the 990 if IDL=1 and IEN=1.

E.2.1 IDLE

You can either read or write the Idle bit (IDL). When IDL is set to 1 the Ten X 99 Processor is not running. When IDL is set to 0 only the Stop bit (STP) is altered by writing and all other status bits are undefined. Setting IDL to 0 causes the Ten X 99 processor to begin execution.

E.2.2 COMPLETE

You can either read or write the Complete bit (CMP). When CMP is set to 1 the Ten X 99 processor has completed execution of a task and has restored the System Table without error. Setting CMP to 1 generates a TILINE Interrupt if the Interrupt Enable Bit (IEN) and the Idle Bit (IDL) are set to 1. The interrupt must be cleared by the host by setting CMP or IEN to 0.

TEN X 99 TILINE INTERFACE

E.2.3 ONLINE

You can only read the Online bit (ONL). When ONL is set to 1 the Ten X 99 Processor has completed the internal self-test successfully and is ready to execute a task.

E.2.4 FAULT

You can only read the Fault bit (FLT). When FLT is set to 1 the Ten X 99 Processor has detected a failure of the loaded self-tests and cannot execute a task.

E.2.5 INTERRUPT ENABLE

You can either read or write the Interrupt Enable bit (IEN), which controls the TILINE interrupt. After IEN is set to 1 a TILINE interrupt is issued for any interrupt condition in the Status Register. Ten X 99 processor interrupt is disabled by setting IEN to 0.

E.2.6 RESET

You can only read the Reset bit (RST). When RST is set to 1 the Ten X 99 Processor has been issued either a Power-On reset or an I/O reset. When RST is set to 1 no other bits in the Status Register, except Idle, are valid. Writing to the control store RST to 0.

E.2.7 READ/WRITE CONTROL STORE

You can either read or write the Read/Write Control Store bit (RWC). Setting RWC to 1 enables reading and writing the control store in the Ten X 99 Processor. The data from the control store is read from the Data Register (part of the TILINE interface) and each subsequent access of the Data Register increments the control store address. The data to be written to the control store is placed in the Data Register and each subsequent access increments the control store address. The Read/Write Control Store bit must be set to zero (RWC=0) before starting the Ten X 99 processor.

E.2.8 NON-EXECUTABLE

You can either read or write the Non-Executable Bit (NEX). When NEX is set to 1 the Ten X 99 Processor has encountered an opcode that cannot be executed. The Ten X 99 Processor restores the System Table and sets IDL to 1.

TEN X 99 TILINE INTERFACE

E.2.9 MEMORY ERROR

You can either read or write the Memory Error bit (MER). MER is set to 1 when a memory error occurs during a memory cycle. The Ten X 99 Processor considers this a fatal error and does not restore the System Table. When MER is set to 1 the Ten X 99 processor sets IDL and halts execution of the task.

E.2.10 TIME OUT

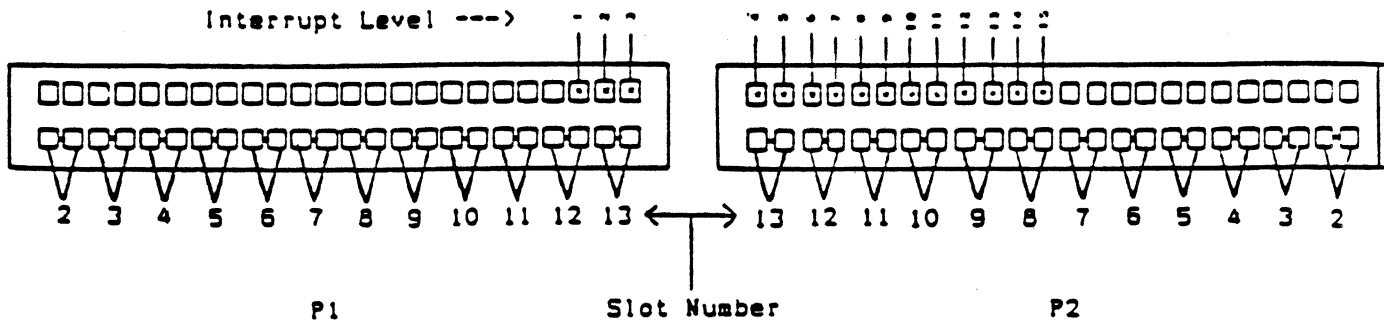
You can either read or write the Time-out bit (TO). TO is set to 1 when a reference is made to non-existent memory. The Ten X 99 Processor considers this a fatal error and does not restore the System Table. When TO is set to 1 the Ten X 99 processor sets IDL and halts execution of the task.

E.2.11 STOP

You can either read or write the Stop bit (STP). The host processor set STP to 1 to halt execution of the Ten X 99 Processor at the end of the current instruction. The Ten X 99 Processor restores the System Table updates the Location Counter and sets IDL to 1. STP is the only bit that may be written when IDL is set to 0.

INTERRUPT JUMPER PLUG FOR
13-SLOT CHASSIS
(LATEST CONFIGURATION)

APPENDIX F



MODEL 2 INTERRUPT PLUG
2261946-0001

P1(LEFT)		SLOT	P2(RIGHT)	
CRU	INT		INT	CRU
2E0	3	2	4	2C0
2A0	N/A	3	N/A	2B0
260	15	4	15	240
220	15	5	15	200
1E0	10	6	10	1C0
1A0	13	7	13	180
160	12	8	12	140
120	10	9	10	100
0E0	11	10	11	0C0
0A0	7	11	7	080
060	9	12	9	040
020	15	13	15	000

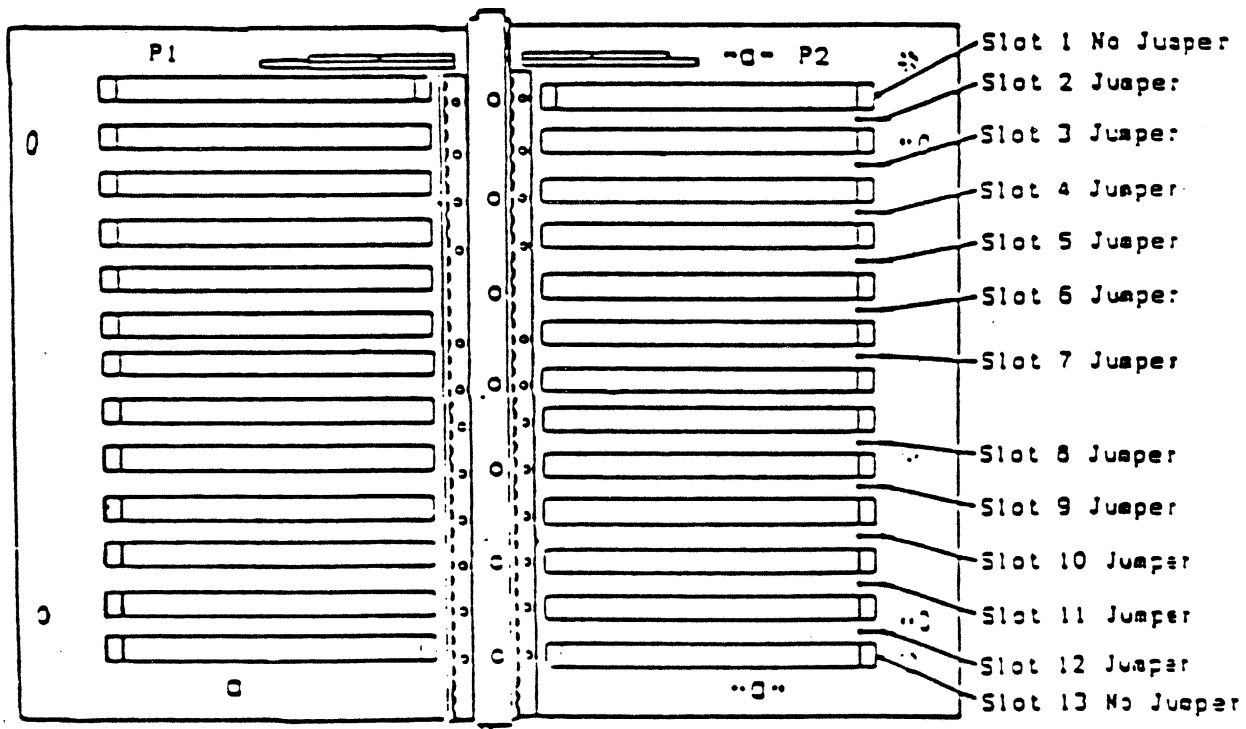
MODELS 3, 4, 4PP, 6, 7, 8, 9 MAIN/EXPANSION
and 8PP, 20, 29, 30 EXPANSION CHASSIS

INTERRUPT PLUG
0949970-0003*

P1(LEFT)		SLOT	P2(RIGHT)	
CRU	INT		INT	CRU
1A0	13**	7	13	180
160	9	8	9	140
120	8	9	10	100
0E0	12	10	11	0C0
0A0	3	11	7	080
060	14	12	4	040
020	15	13	6	000

TIME ACCESS GRANTED JUMPER
LOCATIONS FOR 13-SLOT CHASSIS

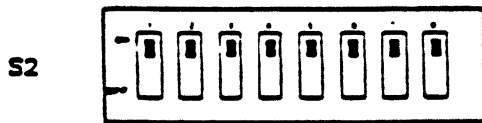
* INTERRUPT PLUG IS 0949970-0005
FOR MODEL 3
** INTERRUPT LEVEL IS 9 FOR MODEL 3



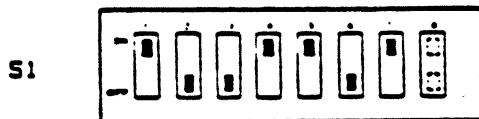
NOTE: Jumper may either be a removable jumper plug or a wire that must be cut. Only right half of chassis contains jumpers.

17-SLOT CHASSIS TILINE ACCESS GRANTED (TLAG) JUMPER SWITCHES

ON = Means no TILINE "Master Device" installed.
OFF = Means TILINE "Master Device" MUST be installed.



Chassis Slot: 2 3 4 5 6 7 8 9

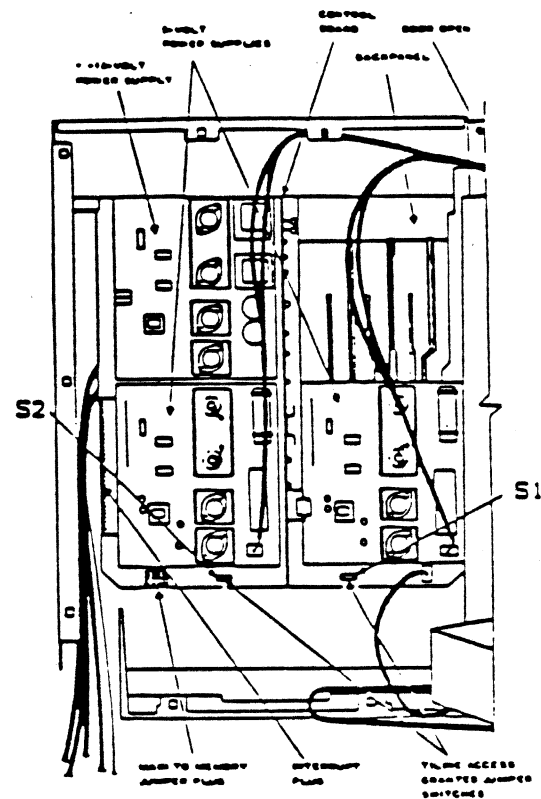


Chassis Slot: 10 11 12 13 14 15 16 N/C

NOTES: 1. Switch positions shown are set for:
a. System disk controller - Slot 11
b. Mag tape controller - Slot 12
c. TILINE flexible disk controller - Slot 15

2. All switches must be "ON" for those slots not containing a TILINE controller.

17-SLOT CHASSIS VIEWED FROM REAR (REAR COVER REMOVED)



17-SLOT MAIN CHASSIS STANDARD INTERRUPT CONFIGURATION

* SLOTS 1-5 TO BE USED FOR TILINE DEVICES ONLY.

P1(TOP)		SLOT	P2(BOTTOM)	
CRU	INT		INT	CRU
*	11	5	11	*
2E0	10	6	10	2C0
2A0	15	7	15	280
260	12	8	12	240
220	8	9	8	200
1E0	3	10	3	1C0
1A0	13	11	13	180
160	9	12	9	140
120	10	13	10	100
0E0	11	14	11	0C0
0A0	7	15	7	080
060	14	16	4	040
020	6	17	6	000

APPENDIX G
INSTALLATION CHECKLIST

NOTE: This checklist has been provided to ensure complete and proper installation of your Ten X 99 COBOL Accelerator. Please check each item after completion.

- () Compute the power load for your chassis to determine if the power supply will support the addition of another device. See Appendix H for power calculation chart.
- () Delete TENX directory if it currently exists.
- () Copy the new TENX directory to your system.
- () Install TENX Procs.
- () Define an unused chassis slot, TILINE address, and interrupt for installation of the Ten X 99.
- () Perform ITX, install Ten X.
- () Delete ALL user synonyms and nonessential system synonyms.
- () Perform a SYSGEN to add the Ten X 99 to your system.
- () Perform the ALGS to assemble and link the new system.
- () Perform the PGS to patch the new system.
- () Perform the TGS for testing the new system.
- () Power down the system for the Ten X 99 board installation. Unplug power cord for 17 slot chassis to prevent electrical shock.
- () Remove TILINE access granted from the slot in which the Ten X 99 board is being installed.
- () Ensure that the correct, non-shared interrupt is available for the slot the Ten X 99 board is being installed.
- () Set the Ten X 99 board to the pre-determined TILINE address.
- () Install the Ten X 99 board insuring proper and complete insertion into the chassis. Trim ejector ears for 17 slot chassis.
- () Restore power to the chassis and reboot (IPL) the newly generated system.
- () Patch COBOL Runtime/Program Files.
- () Load the Ten X 99 Microcode (TXL). This must be done after each re-boot (IPL).
- () Verify that all devices on the new system are functioning.
- () Test the Ten X 99 using the Ten X programs or system programs.
- () Perform an IGS. (install generated system) after system operation has been verified.

APPENDIX H

POWER CALCULATION CHART (Approximate value in AMPS)

CHASSIS RATINGS: 13 Slot = 40 13A = 60 17 Slot = 40 to 120

DESCRIPTION	UNIT VALUE	TOTAL
TEN X 99 COBOL ACCELERATOR	5.00	()
PROGRAMMER'S FRONT PANEL	0.54	()
PROCESSORS		
990/10 SMI/AU	10.00	()
990/10A W/512 KB Memory	6.00	()
990/12 SMI/AU	21.00	()
MEMORY CONTROLLERS		
64 KB Standard controller	2.55	()
96 KB Standard controller	2.56	()
64 KB Cache controller	5.00	()
MEMORY EXPANSION (ARRAYS)		
64 KB Expansion	0.69	()
128 KB Expansion	0.70	()
192 KB Expansion	0.71	()
256 KB Expansion	0.72	()
TILINE COUPLER	0.12	()
DISK CONTROLLERS (TILINE)		
T25/T50/T200/T300 (DS200/DS300)	8.00	()
CD1400 64/96 MB (Phoenix)	6.00	()
DC1400 Controller	8.00	()
WD500/800 (Winchester)	3.00	()
DS10 10MB (CDC)	5.50	()
DS31/44 Diablo	4.50	()
FD1000 DD Floppy	4.50	()
DISK CONTROLLERS (CRU)		
FD800 Single Sided Floppy	3.00	()
TAPE CONTROLLERS		
979 Controller, 800 bpi	5.00	()
979 Controller, 1600 bpi	7.00	()
REMOTE TERMINAL CONTROLLER		
Any Product (Half-size board)	0.60	()
COMMUNICATIONS INTERFACE		
Four Channel Comm Cont. (FCCC)	3.50	()
990 Comm I/F Module	1.50	()
CI 402	0.50	()
CI 403/404	3.00	()
TTY/EIA	0.40	()
INTERNAL MODEMS and INTERFACE		
Synchronous/Asynchronous modems	0.20	()
Internal/external auto-call	1.00	()
VIDEO DISPLAY UNIT CONTROLLERS		
911 VDU, Single	2.62	()
911 VDU, Dual	4.20	()
Televideo MUX (Multiplexer)	0.55	()
LINE PRINTER INTERFACE		
743/745 KSR	0.40	()
810/820/LQ45 Printer	0.40	()
LP300/600 Printer	0.53	()
Other printers (TTY/EIA I/F)	0.40	()
(16 I/O Data Mod)	0.53	()
CRU EXPANSION board	0.90	()
CRU BUFFER board	0.65	()