# Test Program Description for ND-100/ND-110

ND-30.005.3 EN

# Test Program Description for ND-100/ND-110

ND-30.005.3 EN

## Preface

**THE MANUAL:**

This manual describes the test programs for ND-100/ND-110. Chapter 1 gives a detailed description of the TPE-monitor. Chapter 2-20 describes the test programs. Chapter 21 describes the utility programs:

- FILESYSTEM INVESTIGATOR
- DIR-BACKUP
- MCOPY-TANB

**THE READER:**

This manual is intended for the Norsk Data field-service engineers and personell directly involved in maintaining the ND-100/ND-110 computer systems.

**PREREQUSITE KNOWLEDGE:**

It is necessary to have a basic knowledge of the hardware in the ND-100/ND-110 computer system. This can be obtained either by attending the Norsk Data courses introducing the ND-100/ND-110, or by carefully studying the following manuals:

- ND-100 Functional Description (ND-06.015)
- ND-100 Reference Manual (ND-06.014)

# Chapter 1 TPE MONITOR

# CHAPTER 1 TPE MONITOR

The TPE monitor is used for running test programs, either online (using the operating system) or stand-alone. It controls test programs written in PLANC and assembler languages. Other languages may also be used. Functions to interface directly with TELEFIX are included (See chapter 1.6).

## 1.1 GENERAL

The TPE MONITOR makes the different test programs behave in a uniform way. It is based on a command structure, i.e. the user can specify the desired action by writing a "command" to the program.

Commands are handled by the monitor. When the monitor is ready to accept a new command it types the sign 'TPE>'.

**Terminating a test:**

In order to return to the monitor, you can terminate the commands in different ways, but it is always possible to use ESCAPE.

The monitor has its own set of commands (monitor commands). The test program running under control of the monitor has another set of commands (program commands) specific to this test program.

The TPE MONITOR uses two devices:

1. The CONSOLE terminal.

2. The PRINTER device, a terminal or a line printer where the test program prints its outputs.

At the original start-up, the PRINTER device is set equal to the CONSOLE terminal.

The TPE MONITOR also includes functions for communicating directly with the TELEFIX system.

## 1.2 DIFFERENT EXECUTION MODES

The ND-100/ND-110 TPE MONITOR runs on any type of ND-100/ ND-110 computer system. It may be used in two modes:

a. NORMAL MODE. This is a monitor program without any test program loaded, but it has a LOAD command allowing any test program file to be placed into the memory area.

b. PROGRAM-INCLUDED MODE. The monitor and the test program are enclosed in a single BPUN file loadable from the FLOPPY MONITOR (stand-alone), or loadable with @LOAD-BINARY (SINTRAN), or executable as an RT program (SINTRAN).

For each of these two modes, the TPE monitor can run in four environments:

1. Stand alone.
2. Under SINTRAN, as an interactive job.
3. Under SINTRAN, as a MODE or BATCH job.
4. Under SINTRAN, as a RT program.

**NOTE:** When running stand-alone, the Real Time Clock is active and gives interrupts used for several internal functions. If the RTC does not work correctly, the TPE MONITOR cannot run.

## 1.2.1 Normal Mode

At the start-up, the monitor is alone in user memory. It is able to load a test program from a diskette (stand-alone) or from the SINTRAN file system. The test program file includes only the test program, and not a copy of the monitor as for the other mode. The test program file type is :TEST.

### Commands available when running stand-alone

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
DUMP-PRINTER-BUFFER (<from> <to>)
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-FILES <file name>
LIST-MACROS <macro name>
LOAD-PROGRAM <file name>
MONITOR-HELP <command>
OPCOM
PRINT-NOTE <note number>
PROGRAM-HELP <command>
PROGRAM-STATUS
SET-CONSOLE-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-MODE <mode> (<buffer size>)
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
UPDAT <minute> <hour> <day> <month> <year>
```

### COMMANDS AVAILABLE WHEN RUNNING UNDER SINTRAN BUT NOT RT

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-FILES <file name>
LIST-MACROS <macro name>
LOAD-PROGRAM <file name>
MONITOR-HELP <command>
PRINT-NOTE <note number>
PROGRAM-HELP <command>
PROGRAM-STATUS
SET-PRINTER-FILE <file name>
SET-PRINTER-MODE <mode> (<buffer size>)
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
```

### COMMANDS AVAILABLE WHEN RUNNING UNDER SINTRAN AS A RT PROGRAM

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-FILES <file name>
LIST-MACROS <macro name>
LOAD-PROGRAM <file name>
MONITOR-HELP <command>
PRINT-NOTE <note number>
PROGRAM-HELP <command>
PROGRAM-STATUS
SET-CONSOLE-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-FILE <file name>
SET-PRINTER-MODE <mode> (<buffer size>)
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
```

## 1.2.2 PROGRAM INCLUDED MODE

The TPE MONITOR and the test program are one single BPUN format file. This is the same as when the TEST PROGRAM MONITOR runs a test program.

## Commands available when running stand-alone

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
DUMP-PRINTER-BUFFER (<from> <to>)
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-MACROS <macro name>
MONITOR-HELP <command>
OPCOM
PRINT-NOTE <note number>
PROGRAM-HELP <command>
PROGRAM-STATUS
SET-CONSOLE-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-MODE <mode> (<buffer size>)
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
UPDAT <minute> <hour> <day> <month> <year>
```

## Commands available when running under SINTRAN but not RT

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-MACROS <macro name>
MONITOR-HELP <command>
PRINT-NOTE <note number>
PROGRAM-HELP <command>
PROGRAM-STATUS
SET-PRINTER-FILE <file name>
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
```

## Commands available when running under SINTRAN as a RT program

```
MODE <input file name> (<output>)
DATCL
DEFINE-MACRO <macro name> <macro body>
DELETE-MACROS
EXIT
EXPLAIN-COMMAND <command>
HELP <command>
LIST-MACROS <macro name>
MONITOR-HELP <command>
PRINT-NOTE <note number>
PROGRAM-HELP <command>
```

```
PROGRAM-STATUS
SET-CONSOLE-DEVICE-NUMBER <logical or hardware number>
SET-PRINTER-FILE <file name>
SET-TERMINAL-TYPE <type>
TERMINAL-MODE <full page stop>
```

## 1.3 The Command Processor

The command processor interfaces the user to the monitor and the test program through the CONSOLE terminal. The original CONSOLE terminal is:

1. The system console, logical unit 1, when running stand-alone or as a SINTRAN RT program.
2. The terminal from where the program is recovered, when running as a SINTRAN background program.
3. The command input file when running as a SINTRAN MODE or BATCH job.

## Operator inputs

The method is the same in all modes. It is possible to type characters ahead, 60 characters maximum.

The command processor is terminal-type oriented. This means that it handles different editing keys and display modes depending on the CONSOLE terminal type, paper terminal or VDU. The original terminal type is 2 (paper terminal) when running stand-alone, or the SINTRAN defined type.

An input may always be cancelled by pressing ESCAPE.

## Editing keys

CR is always used to terminate the input. Before CR is given, ESCAPE may be used to cancel the input.

<u>When using a paper terminal (as for SINTRAN commands):</u>

| | | |
|---|---|---|
| DEL or CTRL+A | : | Delete one character. |
| CTRL+W | : | Delete one word. |
| CTRL+K | : | Delete the whole line. |
| CTRL+V | : | Enter one control code, if allowed. |

When using a VDU terminal (like in NOTIS):

```
DEL or CTRL+A : Delete one character.
CTRL+K        : Delete the whole line.
```

| | |
|---|---|
| CTRL+D X | : Delete up to and including character X.<br>Char X=CTRL+D :<br>Delete the whole line.<br>Char X=CR :<br>Delete the rest of the line.<br>Char X=CTRL+R :<br>Delete from the beginning of the line. |
| Cursor LEFT | : Move to previous character. |
| Cursor RIGHT | : Move to next character. |
| CTRL+F CTRL+F | : Move forward beyond last character. |
| CTRL+R CTRL+R | : Move to beginning of the line. |
| CTRL+E | : Set/reset expand mode. |
| CTRL+V | : Enter one control code, if allowed. |

## IMMEDIATE ACTION INPUT

When giving input, or at any time during a command execution, some keys on the CONSOLE terminal have an immediate effect:

**In all modes:** ESCAPE: Breaks the command execution and returns to the command processor.

**Stand alone only:** CTRL+O: Activates OPCOM.

In addition, when running stand-alone, XON/XOFF sequences are handled.

## SYNTAX OF COMMANDS AND PARAMETERS

The command processor is ready to accept a command when the prompt sign 'TPE>' is displayed on the CONSOLE terminal. A command consists of the command name and, depending on the command type, one or more parameters.

You can give the parameters on the same line as the command name. They must be separated by one or more spaces, or by a comma.

When typing in command and parameter names, observe the following:

1. A default value may be available. If you press CR immediately in response to the prompt, the default value is displayed and ready to be edited. If you press CR once more, the default value is validated and executed. When the default value has been displayed it may be modified before being executed.

   If no default value is available, the terminal bell is sounded when you press CR to indicate that you must provide input. It is not possible to disregard this request for input except by pressing ESCAPE.

2. A help text may be available. If '?' is typed, the text is displayed and the input is restarted at the point it was interrupted by the help request. If no help text is available, the '?' character will become a part of the input as all other normal characters.

3. After a command has been executed, the default value for the next command input is usually the previous command together with its parameters. When at the command processor level, you can display this default value by giving CTRL+H. You can type CTRL+D if you wish to repeat the line (equivalent to CTRL+H and CR).

## PARAMETER INPUT

There are different types of parameters:

- ASCII characters string
- Numeric
- Boolean

The ASCII characters string: The ASCII characters string input type is used for command names as well as for parameters. In special cases, it may include control codes which are part of the string.

**The numeric input**

The numeric input may be given in different ways:

When the prompt is displayed, the default radix and the legal range for this numeric input are usually also displayed.

If a legal range is specified, the prompt is repeated as long as the input does not fit. It is also repeated as long as illegal characters are typed.

The default radix may always be overruled by adding a standard trailing character:

- B to force an octal input.
- D to force a decimal input.
- X to force a binary input.
- H to force an hexadecimal input.

Any numeric input may be prefixed with a plus or minus sign.

**The boolean input**

The boolean input has only two possibilities, either a true or a false answer.

## 1.4 START AND RESTART

**Stand-alone:**

- Press the MCL-button.
- Give the OPCOM command, 1560&.

This loads the FLOPPY MONITOR which must also be present on the diskette. The FLOPPY MONITOR then automatically loads the TPE MONITOR, as this is the only BPUN file on the diskette.

If the ND-100 display panel is present and working, the hardware calendar is read and the software clock is updated.

**Selecting the Console terminal:**

If the T register on level 15 contains an IOX number for a valid and existing terminal interface, the message "IF HERE TYPE ANY CHARACTER" is sent both to this device and to the system console terminal. The first of these two terminals where a key is activated is now selected as the CONSOLE terminal and the PRINTER device.

| FUNCTION | REG.LEV.15 | FLOPPY MONITOR | TPE |
|---|---|---|---|
| Alt. Device<br>Autoload | T<br>X | Legal<br>HW dev.no. | Legal<br>HW dev.no. |

Remember that the FLOPPY MONITOR used to load the TPE MONITOR also does this when starting.

If the X register on level 15 is different from 1, the autoload function is activated. If there is only one TEST type file on the diskette, it is automatically loaded through the LOAD-PROGRAM command.

Remember that the FLOPPY MONITOR also executes the autoload function to load the TPE MONITOR, but this will only occur if the X register on level 15 is different from 177777B. So, a complete autoload for a single TEST file is done if the X register on level 15 is different from 177777B and 1.

The TPE autoload function searches the diskette to see if it contains only one testprogram. If that is the case, it will load this testprogram for you.

**Running under SINTRAN:**

When running under SINTRAN, the autoload function is also available. You must give the test program file name on the same line as the TPE-MONITOR program name:

```
@TPE-MON <file name>
```

To start under SINTRAN as an RT program, you give the command:

```
@RT TPEMON
```

In this case, the program starts on the system console, device number 1. To start on another terminal, the background program TPE-RT-MON must be activated from user SYSTEM or preferably from user RT on that terminal:

```
@TPE-RT-MON;
```

Be aware that user RT must have at least two pages, one of which must be free. If user SYSTEM is entered, it must have directory and write access to user RT.

## 1.5 Monitor Commands

## HELP

Format: HELP <command>

Lists the monitor commands, the test program commands and the defined macro commands, at the CONSOLE terminal. These three different groups are separated with headings. Depending on how the TPE is running, the commands which may be executed (legal) are prefixed with an arrow '>'.

Parameters:

<command> : Matching name for the commands to be listed. DEFAULT:All existing command names.

## PROGRAM-HELP

Format: PROGRAM-HELP <command>

Same as the HELP command, but only the test program commands are listed.

## MONITOR-HELP

Format: MONITOR-HELP <command>

Same as the HELP command, but only the monitor commands are listed.

## EXPLAIN-COMMAND

Format: EXPLAIN-COMMAND <command>

An explanatory text corresponding to the command <command> is printed at the CONSOLE terminal.

Parameters:

<command>: The name of the command to be explained.
DEFAULT:The test program purpose, if loaded.

## DEFINE-MACRO

Format: MACRO <macro name> <macro body>

This command allows the constructing of new commands (macro commands), made of existing commands and already defined macros.

Parameters:

<macro name>: Is any name which is not already used for a command or another defined macro.
DEFAULT: No default value.

<macro body>: Is the list of commands or macros, separated by a semicolon (;).
DEFAULT: No default value.

You can activate the new macro by giving its name when the command processor prompts for a new command. If the commands in the macro body have missing parameters, they are automatically prompted for by the command processor so they can be manually entered.

The new macro appears in the list from the HELP and LIST-MACROS commands.

The macros are stored in a limited size buffer, and an error message may appear if the buffer is full and an attempt is made to construct a new macro. In such a case the command DELETE-MACROS may be used to empty this buffer.

## DELETE-MACROS

Format: DELETE-MACROS

All macros built by the DEFINE-MACRO command are deleted. It then becomes possible to redefine new ones.

## LIST-FILES

Format: LIST-FILES <file name>

Lists the available files at the CONSOLE terminal.

Parameters:

<file name>: Matching name for the listed files. The files are located on the diskette inserted in the floppy disc controller 1, unit 0, when running stand-alone. Under SINTRAN, you can specify any directory/user name.
DEFAULT:All files from the diskette or under the logged-in user.

## LOAD-PROGRAM

Format: LOAD-PROGRAM <file name>

Places one test program, taken from the diskette or a SINTRAN file, into memory so it can be run. The file type for the test program is TEST.

Parameters:

<file name>: Matching name for the test program to be loaded.

When running stand-alone, the programs are located on the diskette inserted in the floppy disc controller 1, unit 0. Under SINTRAN, you can specify any directory/user name.
DEFAULT: No default value.

If <file name> is not identical to any command name and no program is already loaded, you can omit the command name LOAD-PROGRAM

## SET-CONSOLE-DEVICE-NUMBER

Format: SET-CONSOLE-DEVICE-NUMBER <logical or hardware number>

Selects the device used as the CONSOLE terminal.

Parameters:

| | |
|---|---|
| <log/hw No.>: | Octal value which may be specified either as a logical unit number (software number) or as an IOX number (hardware number), corresponding to any existing terminal.<br>DEFAULT:All existing terminals. A message is sent to all terminals, and the first one where any key is activated is selected as the console. |

## SET-PRINTER-DEVICE-NUMBER OR SET-PRINTER-FILE

Format: SET-PRINTER-DEVICE-NUMBER <logical or hardware number>

Format: SET-PRINTER-FILE <file name>

Selects the device or the SINTRAN file used as the PRINTER device.

Parameters:

| | |
|---|---|
| <log/hw No.>: | Only in stand-alone mode. Octal value which may be specified either as a logical unit number (software number) or as an IOX number (hardware number) corresponding to any existing terminal or line printer.<br>DEFAULT: No default value. |
| <file name>: | Only under SINTRAN. File name with SYMB as default file type.<br>DEFAULT:LINE-PRINTER. |

## SET-PRINTER-MODE

Format: SET-PRINTER-MODE <mode> (<buffer size>)

Parameters:

| | |
|---|---|
| <mode>: | NORMAL:<br>This is the default mode. Output to printer device will be routed to the device selected as printer device.<br>DUPLICATED:<br>The output to PRINTER device will ALSO go to the CONSOLE device. |

BUFFERED:
Only available in stand-alone. The printer device becomes an internal memory buffer. The specified number of pages (1 to 64), or 16 pages (32 Kbytes) if the <buffer size> parameter is omitted, are allocated.

Some words concerning the BUFFERED mode:

- When the buffered printer mode is reset to NORMAL, the printer device is the one you last selected with the command

  SET-PRINTER-DEVICE-NUMBER

- In case of a system crash, it is possible to look at the physical memory to fetch the program outputs. If running via TELEFIX, the CONVERT-OCTAL-TO-ASCII-MANUAL-MODE command is particularly appropriate. Two 32 bits integers (INTEGER4) may be inspected to get the lower and upper limit of the buffer area:

  Physical addr.10-11 = Lower physical addr. of buffer.
  Physical addr.12-13 = Upper physical addr. of buffer.

## DUMP-PRINTER-BUFFER

Format: DUMP-PRINTER-BUFFER (<from> <to>)

The printer buffer defined to replace the PRINTER device by the command SET-PRINTER-MODE,BUFFERED is printed to the previously defined PRINTER device.

Parameters:

(<from> <to>): The optional parameters (<from> <to>) will allow you to print a part of the buffer. The buffer is just an area in memory, and you can specify the area by giving the addresses <FROM> and <TO>.
0 means the beginning of the buffer, 177777B means the end (byte displacements from zero). NOTE that the end address depends on the buffer size specified by the command:

SET-PRINTER-MODE

| Buffer size | Buffer area |
|---|---|
| 16 pages | 0- 77777B |
| 32 pages | 0-177777B |
| 64 pages | 0-377777B |

## PRINT-NOTE

Format: PRINT-NOTE <note number>

Prints the test program note(s), on the device selected as the PRINTER device.

Parameters:

<note number>: The note number to be printed.
DEFAULT:All existing notes.

## PROGRAM-STATUS

Format: PROGRAM-STATUS

Prints some information at the CONSOLE terminal:

- TPE MONITOR version.
- Day, hour, minute.
- CONSOLE terminal logical unit number.
- PRINTER device logical unit number or open file number.
- Printer mode.
- Stop on full page.
- Name of the loaded test program.

The test program may add its own information.

## TERMINAL-MODE

Format: TERMINAL-MODE <full page stop>

Sets or resets the stop on full page function for the CONSOLE terminal. Any output to the CONSOLE is stopped after 22 lines and bell is activated once. The output restarts when any key is pressed.

Parameters:

<page stop> : You must answer with YES or NO.
DEFAULT:The opposite of the present state.

## SET-TERMINAL-TYPE

Format: SET-TERMINAL-TYPE <type>

Selects the type for the CONSOLE terminal. Used to handle editing functions when giving inputs from the CONSOLE terminal keyboard. Only a few terminal types can be specified here.

2= Paper printer terminal.
3= Tandberg, any type. (Includes also 36,
   53, 83, 93)
4= Infoton, any type. (Includes also 5, 12)

The original value is 2, or the SINTRAN defined type.

Parameters:

| | |
|---|---|
| <type> : | A decimal value corresponding to one of the valid types.<br>DEFAULT: No default value. |

## OPCOM

Format: OPCOM

This command starts OPCOM on the system console (device 1). If you run TPE from the system console, OPCOM will take over until you terminate it.

OPCOM is terminated by pressing the ESCAPE-key on the system console. If you were running TPE from the system console, you will now have TPE back.

## DATCL

Format: DATCL

Prints current time and date at the CONSOLE terminal. If the display panel is present, the monitor clock was updated from the hardware calendar at the start-up time (it is also updated in case of a restart or a power failure restart).

## UPDAT

Format: UPDAT <minute> <hour> <day> <month> <year>

Sets the correct program time and date, and writes the hardware calendar if the display panel is present.

Parameters:

<time> : Numeric decimal values with legal ranges.
DEFAULT: No default value.

## MODE

Format: MODE <input name> <output file>

The input for the command processor is taken from the specified file name with default type :SYMB. The output is switched to the PRINTER device presently defined.

Parameters:

<input name>: This is the name of a file containing commands to TPE and/or commands to the test program.
Default file type is :SYMB.

TPE will now take its command from the output file just as a SINTRAN mode job.

- A new MODE command may appear in the MODE file, but there is no return mechanism to the previous one.
- Some monitor commands are not allowed in MODE:

OPCOM
SET-CONSOLE-DEVICE-NUMBER
SET-PRINTER-DEVICE-NUMBER / SET-PRINTER-FILE
SET-TERMINAL-TYPE
TERMINAL-MODE

<output file>:
Sends the output to the specified file. The output file is in fact the TPE printer device. Specifying the output file will redefine the printer device.

## EXIT

Format: EXIT

Stops the computer, or returns to SINTRAN. If a subset of commands is entered, EXIT may return to a previous command level instead. The command EXPLAIN-COMMAND EXIT always indicates what the EXIT command will do.

## 1.6 Using the TPE Monitor via TELEFIX

The TPE MONITOR includes functions for direct communication with TELEFIX.

## The TELEFIX command LOAD-BINARY

TELEFIX command format: LOAD-BINARY <BPUN file name>

This command can be used from the TELEFIX system to load the TPE MONITOR in the remote system memory through the communication line. This is the same procedure as for loading any binary program.

In case of communication failure or hang up, the remote system may be forced to enter STOP mode by using the MANUAL-MODE to send 3 consecutive ESCAPEs.

## The TELEFIX command LOAD-PROGRAM

TELEFIX command format: LOAD-PROGRAM <TEST file name>

The TPE MONITOR is assumed to be already running in the remote system and waiting for a command. At the TELEFIX system side, the LOAD-PROGRAM command may be used to send a TEST file to the TPE MONITOR through the communication line. After the transfer is terminated, the MANUAL-MODE is to be activated from the TELEFIX terminal in order to be in connection with the remote system in the same situation as after the execution of a LOAD-PROGRAM command directly given at the TPE MONITOR.

In case of communication failure or hang up, the remote system may be forced to enter STOP mode by using the MANUAL-MODE to send 3 consecutive ESCAPEs.

## THE TELEFIX COMMAND CONVERT-OCTAL-TO-ASCII-MANUAL-MODE

This command is used to run the MANUAL-MODE:

Whenever a string of characters that has the appearance of a 6 digits octal number is received from the remote system, it is replaced and displayed as two ASCII character codes corresponding to this octal number. Control codes which are not CR or LF are displayed as a dot.

This command is particularly appropriate if the TPE MONITOR crashes in the remote system during a test program run, and the printer device is in buffered mode (the SET-PRINTER-MODE BUFFERED command has been issued). It is then possible to fetch the program output messages from the buffer in main memory.

Four memory locations must first be inspected in the remote system by using the normal MANUAL-MODE:

- Content of locations 10-11 = 32 bit physical memory address for the printer buffer.

- Content of locations 12-13 = 32 bit physical memory address for the upper bound of the text present in the printer buffer.

Then, the CONVERT-OCTAL-TO-ASCII-MANUAL-MODE command can be used to dump the remote main memory between these two limits, by sending the OPCOM command:

```
xxxxxxxx<yyyyyyyy.
```

(xxxxxxxx = 32 bit content of memory locations 10-11, yyyyyyyy = 32 bit content of memory locations 12-13).

## THE TELEFIX SCRIPT CODE PROTOCOL

The script code PROTOCOL ON is intended to activate a communication protocol for messages exchanged between TELEFIX and the TPE MONITOR during execution of a TELEFIX automatic mode. When the protocol is in use, communication errors on the line are automatically handled. The protocol is turned off by the script code PROTOCOL OFF.

In case of communication hangup, the TPE MONITOR stops the remote system after a timeout. If the MANUAL-MODE is entered while the TPE MONITOR is waiting for a command and the communication protocol is still on, it may be turned off by sending 3 consecutive CTRL+N. It is not possible to communicate directly through the MANUAL-MODE as long as the protocol is on in the TPE MONITOR.

## 1.7 Error Messages From the Monitor

Error messages displayed by the TPE MONITOR are of 3 kinds:

1. Errors concerning a monitor command decoding and execution, including the errors returned by the file system.
2. System errors.
3. PLANC library runtime errors.

### Errors from monitor commands

An error during execution of a monitor command does not stop the monitor, but usually aborts the command and returns to the command processor.

Errors may be detected when giving a command and parameters.

**Example:**

```
*** Ambiguous command ***
```

Errors may be detected during the command execution.

**Example:**

```
*** Printer not ready ***
```

Errors may come from the file system, either stand-alone or SINTRAN.

**Examples:**

```
NO SUCH FILE NAME
DEVICE NOT READY
```

## SYSTEM ERRORS

Errors may be detected by the internal interrupt system.

**Example:**

```
==TPE42=> Memory parity error. PEA, PES:  xxxxxx xxxxxx
========> Level (decimal), instruction, address: x  xxxxxx  xxxxxx
========> Fatal error, STOP forced.
```

Other hardware errors may also be detected.

**Example:**

```
==TPE42=> The clock does not work
========> Fatal error, STOP forced.
```

Errors may be detected by the monitor because the test-program command does not handle the error situations by itself. In principle, this should not happen.

**Example:**

```
==TPE43=> Unexpected program error. ERRCODE = xxxxxxB
========> Fatal error, STOP forced.
```

## PLANC LIBRARY RUNTIME ERRORS

The monitor is written in PLANC, and the PLANC library includes some routines which may stop the monitor in case of internal error detection in the PLANC logic. In principle, this should not happen.

**Example:**

```
 - STACK OVERFLOW AT xxxxxxB
```

# CHAPTER 2 CACHE-100

# CHAPTER 2 CACHE-100

This chapter describes the CACHE-100 test program.

## 2.1 GENERAL

The CACHE-100 is designed to test the cache memory on the ND-100 computer systems. It runs on all ND-100 types in stand-alone mode only.

CACHE-100 is an improvement of the old CACHE-D.

It is mandatory that the main memory system works correctly, otherwise wrong diagnoses or program crash may occur.

There is no maintenance or test feature in the cache logic, and the cache status register gives only poor information. The only way to test the cache is to manipulate the inhibit limit logic. This makes it possible to prevent writing in cache for a defined memory-address range, but still allows reading from the cache if the corresponding data is present.

All tests are based on this mechanism:

- Enable Cache.
- Write data to memory with copy in cache (1).
- Disable Cache.
- Write another data pattern to the same memory location (2).

You will now have this situation in memory and cache:

|  | Addr | Cache | Memory |
|---|---|---|---|
| (1) | n | 177777B | 177777B |
| (2) | n | 177777B | 0B |

The memory and cache contents are now different and, since cache is not cleared, you will get a hit in cache when reading from the same location. In this way you can test the cache.

## 2.2 Starting and Initial Action

The Cache-100 is started from the TPE-MONITOR by the command:

```
TPE>Load CACHE-100 <CR>
```

At start-up the program finds out if the cache option is really present, and if the manual disable switch is ON or OFF. It is not possible to distinguish between a manual disable and no cache present on the ND-100.

Anytime the RUN command is activated, the program does the following:

1. Check if the manual disable switch has not been turned ON.

2. Find the memory size.

3. Clear all existing pages to zero.

## 2.3 Program Commands

## SET-PARAMETERS

This command allows you to define how the program should run the tests.

Format:

```
SET-PARAMETERS <loop mode> (<loops>) <abort mode> (<errors>)
               <suppress mode>
```

<loop mode> : You must answer YES or NO.
YES means the test or test sequence is repeated.
NO means the test runs only once.
DEFAULT: No.

<loops> : If <loop mode> was YES, you must specify the number of times the test or test sequence shall repeat.
DEFAULT: Infinite.

<abort mode> : You must answer YES or NO.
YES means the test(s) are aborted when the maximum number of errors is reached.
NO means the test(s) will never abort.
DEFAULT: Yes.

<errors> : If <abort mode> was YES, you must specify the maximum nuber of errors allowed. When this limit is reached, the test(s) will abort.
DEFAULT: 10.

<suppress mode>: You must answer YES or NO.
YES means error messages are supressed.
NO means error messages are printed.
DEFAULT: No.

When starting CACHE-100 you will get the default setting on all parameters.

## RUN

This command allows the execution of one specific test, or a subset of tests in sequence.

Format:
RUN <Test number(s)>

Test numbers are in the range 1 to 6. Here you can choose one of four ways to use RUN:

- ALL TESTS: Give default value
- SEQUENCE : Test numbers separated by comma or space
- RANGE : First and last test number separated by :
- SINGLE : One single test number

**Examples**

```
RUN,,:                    % Run All
RUN 3,5,2:6,1             % Run 3-5-2-3-4-5-6-1
```

The RUN command will print a start header containing the loop mode, Abort mode and the current time. Each test will identify itself and give a 'end of test' indication. The test sequence will repeat if the loop mode is selected.

## 2.4 Test Description

It is normally advisable to run the test sections in the order they are presented. If, however, a fatal error occurs in any section of the tests, the other sections must be run separately. All parts of the cache system are interdependent. Some parts of the system are therefore needed to test other parts. In the event of a heavy failure, it would not therefore be certain that a correct diagnosis could be obtained. If the whole test runs correctly to the end, it is likely that the cache system is working correctly. When a test is carried out successfully, the tested cache memory can only be considered to be working correctly in memory configurations up to the size of the machine tested.

The time needed to run the program is dependent on the size of the main memory. This can differ considerably from one system to another.

Some abbreviations used later in the text are explained here:

CSR Cache Status Register, ND-100 Internal Register number 10B.

CUP Cache Updated. CSR bit 0. Set if next memory reference causes a write to cache.

CON Cache On indicator. CSR bit 1. Set if cache is present. CON is 0 during cache clear and when the cache is manually disabled.

## TEST 1 - BASIC FUNCTIONS

Execution time: 0:00:01 (1Mbyte) - 0:00:01 (3Mbyte)

- CSR bit 1 - CON is tested during execution of a CACHE CLEAR. It verifies that CON is set to one after a while.
- CSR bit 0 - CUP is tested for set and reset conditions.
- Source conditions for data output, either from main memory or from cache, are verified.

## TEST 2 - INHIBIT LIMITS

Execution time: 0:00:30 (1Mbyte) - 0:01:30 (3Mbyte)

- Looking at the CUP bit while executing the TRA 10 instruction in all possible pages, for many possible configurations of lower and upper limits in the inhibit registers.

The paging system is used to place the "TRA 10; EXIT" sequence in every page, and to jump to this sequence. The program may crash without any diagnosis if the paging system fails.

## Test 3 - Cache "Used bit memory" test

Execution time: 0:01:20 (1Mbyte) - 0:01:25 (3Mbyte)

- The USED BIT is tested for 1, for 0, and for correct addressing.

This single bit, which is part of the cache memory, is present in the 1024 memory locations to flag if it contains valid information.

## Test 4 - Cache "Data memory" test

Execution time: 0:00:50 (1Mbyte) - 0:00:50 (3Mbyte)

- Static and dynamic test of the memory page used in the test.
- Source test for data output, either memory or cache.
- Static and dynamic memory test is carried out with all data addresses covered by the cache. This means the whole 1Kword data area is in the cache.

## Test 5 - Cache "Page number memory" test

Execution time: 0:00:40 (1Mbyte) - 0:01:30 (3Mbyte)

This test verifies that the page number bits in the cache memory are correct.

- Static, address and dynamic tests are carried out by identifying the page number where the data coming from the cache belongs.

  When mismatch is discovered, the program makes an attempt to find out which page number is in the faulty cache location. This is done by trying to read from all possible memory pages until the data comes from the cache. As it is not possible to use the pages where the program is running, the search may give no result.

Only the "PAGE NUMBER" bits corresponding to the maximum physical memory size range are tested.

## Test 6 - Cache efficiency

Execution time: 0:00:07 (1Mbyte) - 0:00:07 (3Mbyte)

- Loop executed outside the cache.
- Loop executed inside the cache.
- Calculating the speed gain.

The calculated speed gain must be equal to or greater than:

| Cpu type | Speed gain with cache |
|---|---|
| Fast cycle | 28 % |
| Slow cycle | 20 % |

NB: The values concern only this particular loop.

## 2.5 Error Messages

Since an error is very likely to be followed by many others of the same type, error messages are printed in the form of tables. To speed up printout, this type of message will start by printing an error-message header followed by lines of Error Data. The header will be repeated at regular intervals. The very first error message in an error sequence will be preceded by the text

*** ERROR *** In test xxx. Time: yyyy.mm.dd hh:mm:ss

You can suppress the error message printout by selecting the <Suppress mode> with the

SET-PARAMETERS command.

The header text is then only printed the first time an error occurs within a test section.

# Chapter 3 CACHE-110

# CHAPTER 3 CACHE-110

This chapter describes the CACHE-110 test program.

## 3.1 GENERAL

The CACHE-110 is designed to test the cache memory on the ND-110(/CX) CPU. It runs in stand-alone mode only.

The main memory system must be working correctly. Otherwise wrong diagnoses or program crash may occur.

There is no maintenance or test feature in the cache logic, and the cache status register gives only poor information. The only way to test the cache is to manipulate the inhibit limit logic. This makes it possible to prevent writing in cache for a defined memory-address range, but still allows reading from the cache if the corresponding data is present.

All tests are based on this mechanism:

- Enable Cache.
- Write data to memory with copy in cache (1).
- Disable Cache.
- Write another data pattern to the same memory location (2).

You will now have this situation in memory and cache:

|  | Addr | Cache | Memory |
|---|---|---|---|
| (1) | n | 177777B | 177777B |
| (2) | n | 177777B | 0B |

The memory and cache contents are now different. And since cache is not cleared, you will get a hit in cache when reading from the same location. In this way you can test the cache.

The ND-110/CX cache is more difficult to test than the ND-100, as part of it is reserved for instructions and cannot be accessed to read its content. The instruction caches are tested as described above, but instead of writing the data, you must first write instructions into memory and then run them in order to have them cached.

## 3.2 Starting and Initial Action

The Cache-110 is started from the TPE-MONITOR by the command:

```
TPE>Load CACHE-110 <CR>
```

At start-up the program finds out if the cache option is really present, and if the manual disable switch is ON or OFF. It is not possible to distinguish between a manual disable and no cache present on the ND-110.

Whenever the RUN command is activated, the program does the following:

1. Check if the manual disable switch has not been turned ON.
2. Find the memory size.
3. Clear all existing pages to zero.

## 3.3 Program Commands

## SET-PARAMETERS

This command allows you to define how the program should run the tests.

Format:

```
SET-PARAMETERS <loop mode> (<loops>) <abort mode> (<errors>)
               <suppress mode>
```

<loop mode> : You must answer YES or NO.
YES means the test or test sequence is repeated
NO means the test runs only once.
DEFAULT: No.

<loops> : If <loop mode> was YES, you must specify the number of times the test or test sequence shall repeat.
DEFAULT: Infinite.

<abort mode> : You must answer YES or NO.
YES means the test(s) are aborted when the maximum number of errors is reached.
NO means the test(s) will never abort.
DEFAULT: Yes.

<errors> : If <abort mode> was YES, you must specify the maximum nuber of errors allowed. When this limit is reached, the test(s) will abort.
DEFAULT: 10.

<suppress mode>: You must answer YES or NO.
YES means error messages are supressed.
NO means error messages are printed.
DEFAULT: No.

When starting CACHE-110 you will get the default setting on all parameters.

## RUN

This command allows the execution of one specific test, or a subset of tests in sequence.

Format:

RUN <Test number(s)>

<Test number(s)> :
Test numbers are in the range 1 to 8. Here you can choose one of four ways to use RUN:

ALL TESTS: Give default value
SEQUENCE : Test numbers separated by comma
or space
RANGE : First and last test number separated by :
SINGLE : One single test number

**Examples**

```
RUN,,:                          % Run All
RUN 3,5,2:6,1                   % Run 3-5-2-3-4-5-6-1
```

The RUN command will print a start header containing the loop mode, Abort mode and the current time. Each test will identify itself and give a 'end of test' indication. The test sequence will repeat if the loop mode is selected.

## LOOK-AT-CONTROL-STORE

This command enables you to inspect and change the control store. A LOOK-AT-MEMORY syntax is shown below:

```
    .  = Exit
  <CR> = Move to next location
 nnnn/ = Move to location nnnn
```

a b c d<CR> = Change current CS location. A, b, c and d are numbers put into the groups of the current location

When entering a location, its contents will be displayed as four 16-bit numbers. Each number represents one group.

## 3.4 Test Description

It is normally advisable to run the test sections in the order they are presented. If, however, a fatal error occurs in any section of the tests, the other sections must be run separately. All parts of the cache system are interdependent. Some parts of the system are therefore needed to test other parts.

In the event of a heavy failure, it would not therefore be certain that a correct diagnosis could be obtained. If the whole test runs correctly to the end, it is likely that the cache system is working correctly. When a test is carried out successfully, the tested cache memory can only be considered to be working correctly in memory configurations up to the size of the machine tested.

The time needed to run the program is dependent on the size of the main memory. This can differ considerably from one system to another.

Some abbreviations used later in the text are explained here:

CSR Cache Status Register, ND-110 Internal Register number 10B.

CUP Cache Updated. CSR bit 0. Set if next memory reference causes a write to cache.

CON Cache On indicator. CSR bit 1. Set if cache is present. CON is 0 during cache clear and when the cache is manually disabled.

CIP Clear In Progress. CSR bit 3. Set when clearing cache.

The ND-110/CX has 4 caches. Two data and two instruction.

The ND-110 has only one instruction cache. When testing a ND-110, the CACHE-110 must skip all testloops concerning data cache and test only one of the two instruction caches. This makes CACHE-110 execute faster on a ND-110 than on a ND-110/CX.

## Test 1 - Control Store verification (Upper 2k)

Execution time ND-110:    0:01:34 (1Mb) - 0:01:34 (3Mb)
Execution time ND-110/CX: 0:01:29 (1Mb) - 0:01:29 (3Mb)

The upper 2K of control store are verified as a memory.

- Static test with all bit set to zero and all bit set to one.
- Address test.
- Dynamic test.

The upper 2K of control store is used as instruction cache.

## Test 2 - Basic functions

Execution time ND-110   : 0:00:02 (1Mb) - 0:00:02 (3Mb)
Execution time ND-110/CX: 0:00:18 (1Mb) - 0:00:18 (3Mb)

This section includes tests for the cache status bits.

Bit 1 (CON) must still be one during execution of a first cache clear, while bit 3 (CIP) must be zero. Bit 3 must come back to one when the other set is cleared. The same test is carried out with a double cache clear.

The cache status bit 0 (CUP) is tested for set and reset conditions.

The source conditions for data output, either from main memory or from cache, are verified:
   Either memory or cache 1, in paging OFF
   Either memory or cache 2, in paging ON with APT
   Either memory or cache 3, on all levels (not 13), but not 1
   Either memory or cache 4, on level 1

The source conditions are also tested between the several caches:
   Cache 1 or 2, when using all possible addressing modes
   Cache 3 or 4, when using all possible levels
   Cache 1, 2, 3, 4, by overwriting one cache at a time

The cache conflict logic is tested by checking that a cache location is cleared when another one is written at the same cache address:
   Cache 1 location cleared by a store in cache 2
   Cache 2 location cleared by a store in cache 1
   Cache 3 location cleared by a store in cache 1 or 2
   Cache 4 location cleared by a store in cache 1 or 2

## Test 3 - Inhibit limits

Execution time ND-110 : 0:00:11 (1Mb) - 0:00:12 (3Mb)
Execution time ND-110/CX: 0:00:18 (1Mb) - 0:00:19 (3Mb)

This test is carried out by looking at the CUP bit in the cache status register while executing the TRA 10 instruction in all possible pages, for many possible configurations of lower and upper limits in the inhibit registers.

The paging system is used to place the "TRA 10; EXIT" sequence in every page, and to jump to this sequence. The program may crash without any diagnosis if the paging system fails.

## Test 4 - Enable/Inhibit pages

Execution time ND-110 : 0:00:35 (1Mb) - 0:05:37 (3Mb)
Execution time ND-110/CX: 0:00:32 (1Mb) - 0:05:16 (3Mb)

This is the same test as test number 3 but, instead of using the inhibit registers, the bit map is directly written with the TRR CLIP instruction. Only one page is enabled at a time while all possible pages are used.

## Test 5 - Cache "Used bit memory" test

Execution time ND-110 : 0:06:50 (1Mb) - 0:06:50 (3Mb)
Execution time ND-110/CX: 0:09:35 (1Mb) - 0:09:35 (3Mb)

This single bit, which is part of the cache memory, is present in the 1024 memory locations of each of the 4 caches to flag if it contains valid information. As there are two sets of "used bit" memory, a test is carried out at least twice, with a clear cache in between to switch to the other set. Unfortunately, it is not possible to know which set is in use at any given time.

Each of the 4 caches is tested for 1, for 0, and for correct addressing. Owing to the complexity of this test, and to the many combinations to be tested, you will need to wait a few minutes while this section is executed.

## Test 6 - Cache "Data memory" test

Execution time ND-110 : 0:00:18 (1Mb) - 0:00:18 (3Mb)
Execution time ND-110/CX: 0:00:42 (1Mb) - 0:00:42 (3Mb)

The memory page used for this test is first tested with static and dynamic patterns. After that, static and dynamic memory test is carried out with all data references covered by the cache. It means the whole 1kw data area is in the cache. This is done for the two data caches, caches 1 and 2.

The instruction caches are tested in another way: a "SAT" instruction is placed in the cache location and executed. By using several arguments within the SAT, it is possible to know the status of the 8 lower data bits by checking the T-register content after execution. Static test with fixed arguments, and address test are carried out in the two instruction caches, caches 3 and 4.

## TEST 7 - CACHE "PAGE NUMBER MEMORY" TEST

Execution time ND-110 : 0:00:23 (1Mb) - 0:00:53 (3Mb)
Execution time ND-110/CX: 0:01:24 (1Mb) - 0:03:46 (3Mb)

This test verifies that the page number bits in the cache memory are correct. Static, address and dynamic tests are carried out, by identifying the page number of the data coming from the cache.

When mismatch is discovered, the program makes an attempt to find out which page number is in the faulty cache location. It does this by trying to read from all possible memory pages until the data comes from the cache.

As it is not possible to use the pages where the program runs, the search may give no result. For the instruction caches, a search of this kind is not possible.

Only the "PAGE NUMBER" bits corresponding to the physical maximum memory size range are tested.

## TEST 8 - CACHE EFFICIENCY

Execution time ND-110 : 0:00:06 (1Mb) - 0:00:06 (3Mb)
Execution time ND-110/CX: 0:00:06 (1Mb) - 0:00:06 (3Mb)

A loop is executed with and without cache, and the speed gain is calculated. This must be equal to or greater than:

| Cpu type | Speed gain with cache |
|---|---|
| ND-110 | 40 % |
| ND-110/CX | 65 % |

NB: The values concern only this particular loop.

It is NOT an ND-110 spec.

## 3.5 Error Messages

Since an error is very likely to be followed by many others of the same type, error messages are printed in the form of tables. To speed up printout, this type of message will start by printing an error-message header followed by lines of Error Data. The header will be repeated at regular intervals. The very first error message in an error sequence will be preceded by the text

***** ERROR ***  In test xxx. Time: yyyy.mm.dd hh:mm:ss**

You can suppress the error message printout by selecting the <Suppress mode> with the

SET-PARAMETERS command.

The header text is then only printed the first time an error occurs within a test section.

# Chapter 4 CONFIGURATION INVESTIGATOR

# CHAPTER 4 CONFIGURATION INVESTIGATOR

## 4.1 GENERAL

The basic test program for testing the I/O system is the Configuration Investigator. This program is more a utility program than a hardware-test program. It checks the hardware configuration of an ND-computer installation and gives a lot of information.

## 4.2 STARTING AND INITIAL ACTION

The Configuration test program is started from the TPE-MONITOR by the command:

```
TPE>Load CONFI <CR>
```

## 4.3 PROGRAM COMMANDS

The following list shows the commands used in this test program:

```
RUN (<quick-check>)
PRINT-MEMORY-MAP
LIST-ALL-DEVICES <sorted alphabetically>
LIST-OVERLAPPING-DEVICES
SET-PARAMETERS <memory map included>,<list-...All devices>
```

## RUN (<QUICK-CHECK>)

This command will print out information about the hardware configuration of the computer, as seen from the ND-1x0 CPU. The information given is:

| | |
|---|---|
| 1. CPU type | (ND-1x0/ normal, CE or CX) |
| 2. Floating point format | (32 or 48 bits) |
| 3. Memory management type | (MMS 1, MMS 2) |
| 4. Cache | (yes or no) |
| 5. ALD register | (xxxxxB) |
| 6. Memory size | (xx.yyy Mbytes) |

7. A list of all devices present, containing: Device name, device number(s), ident code(s), logical device number.

8. A list of all present devices able to generate interrupts, printed in order of interrupt priory. The list contains: Interrupt level, ident code, device name.

9. A map of all memory, printed with a resolution of 64 Kw (1 bank). The possible memory types are:

   ```
   Local - local ND-1x0 memory
   Pioc  - pioc memory
   Ether - ethernet memory
   Mpm 3 - multiport 3 memory (big mpm)
   Mpm 4 - multiport 4 memory
   Mpm 5 - multiport 5 memory
   ```

   Memory is checked for each 16 Kw block. If holes are found, indicating partially filled banks, the memory map is printed with a resolution of 16 Kw.

This command can be executed in a special mode, where ONLY errors are printed. This is done by specifying the optional parameter <QUICK-CHECK>. Legal values are: "YES" or "NO". (Default value is "NO")

**Example:**

```
TPE>RUN <CR>


                  H A R D W A R E    C O N F I G U R A T I O N
                  ===========================================

CPU type...............: ND-110 or ND-110 CX
Floating point format..: 32 bits
Memory management type.: MMS 2 included on CPU card
Cache..................: Yes
ALD register...........: 1560B
Memory size............: 1 Mbyte


                            FIRST   LAST       I D E N T  C O D E S      LOG.
HARDWARE DEVICE NAME        DEVNO  DEVNO    LEV10  LEV11  LEV12  LEV13  DEVNO
----------------------------------------------------------------------------
Real-time clock          1     10     13                             1
Terminal interface       1    300    307    == identcode not checked ==      1
Terminal interface       5    340    347       44            44             44
Terminal interface       6    350    357       45            45             45
Terminal interface       7    360    367       46            46             46
Terminal interface       8    370    377       47            47             47
ST506 5" Disc Contr.     1    500    507                1                 1224
Floppy disc PIO          1   1560   1567               21                 1145
ECCR                       100115 100115
Universal DMA            1 140050 140057           140010                 2100
Terminal interface      65 140400 140407   140040        140040           2000
Terminal interface      66 140410 140417   140041        140041           2001
Terminal interface      67 140420 140427   140042        140042           2002
```

```
Terminal interface      68 140430 140437   140043          140043          2003
Terminal interface      69 140440 140447   140044          140044          2004
Terminal interface      70 140450 140457   140045          140045          2005
Terminal interface      71 140460 140467   140046          140046          2006
Terminal interface      72 140470 140477   140047          140047          2007
Terminal interface      73 140500 140507   140050          140050          2010
Terminal interface      74 140510 140517   140051          140051          2011
Terminal interface      75 140520 140527   140052          140052          2012
Terminal interface      76 140530 140537   140053          140053          2013
Terminal interface      77 140540 140547   140054          140054          2014
Terminal interface      78 140550 140557   140055          140055          2015
Terminal interface      79 140560 140567   140056          140056          2016
Terminal interface      80 140570 140577   140057          140057          2017



  I N T E R R U P T   P R I O R I T Y
  ====================================

LEVEL     IDENT     HARDWARE DEVICE NAME

--------------------------------------------------
   13         1     Real time clock              1
   12        44     Terminal interface           5
   12        45     Terminal interface           6
   12        46     Terminal interface           7
   12        47     Terminal interface           8
   12    140044     Terminal interface          69
   12    140045     Terminal interface          70
   12    140046     Terminal interface          71
   12    140047     Terminal interface          72
   12    140040     Terminal interface          65
   12    140041     Terminal interface          66
   12    140042     Terminal interface          67
   12    140043     Terminal interface          68
   12    140050     Terminal interface          73
   12    140051     Terminal interface          74
   12    140052     Terminal interface          75
   12    140053     Terminal interface          76
   12    140054     Terminal interface          77
   12    140055     Terminal interface          78
   12    140056     Terminal interface          79
   12    140057     Terminal interface          80
   11         1     ST506 5" Disc Contr.         1
   11        21     Floppy disc PIO              1
   11    140010     Universal DMA                1
   10        44     Terminal interface           5
   10        45     Terminal interface           6
   10        46     Terminal interface           7
   10        47     Terminal interface           8
   10    140044     Terminal interface          69
   10    140045     Terminal interface          70
   10    140046     Terminal interface          71
   10    140047     Terminal interface          72
   10    140040     Terminal interface          65
   10    140041     Terminal interface          66
   10    140042     Terminal interface          67
```

```
    10     140043     Terminal interface          68
    10     140050     Terminal interface          73
    10     140051     Terminal interface          74
    10     140052     Terminal interface          75
    10     140053     Terminal interface          76
    10     140054     Terminal interface          77
    10     140055     Terminal interface          78
    10     140056     Terminal interface          79
    10     140057     Terminal interface          80


=== END OF INVESTIGATION ===
===  NO ERRORS DETECTED  ===



                         M E M O R Y   M A P
                        ======================

!=========!=======!=======!=======!=======!=======!=======!=======!=======!
! Bank no !   0   !   1   !   2   !   3   !   4   !   5   !   6   !   7   !
!=========!=======!=======!=======!=======!=======!=======!=======!=======!
!  000B   ! Local ! Local ! Local ! Local !       !       !       !       !
!---------!-------!-------!-------!-------!-------!-------!-------!-------!
Empty until bank 024B                                                     !
!---------!-------!-------!-------!-------!-------!-------!-------!-------!
!  020B   !       !       !       !       ! Local ! Local ! Local ! Local !
!---------!-------!-------!-------!-------!-------!-------!-------!-------!

Total memory size.: 1 Mbyte

TPE>
```

## PRINT-MEMORY-MAP

This command prints a map of all memory, with a resolution of 64 Kw (1 bank).
The possible memory types are:

```
Local - local ND-1x0 memory
Pioc  - pioc memory
Ether - ethernet memory
Mpm 3 - multiport 3 memory (big mpm)
Mpm 4 - multiport 4 memory
Mpm 5 - multiport 5 memory
```

Memory is checked for each 16 Kw block. If holes are found, indicating partially filled banks, the memory map is printed with a resolution of 16 Kw.

See the example of a memory map described under the RUN-command.

## LIST-ALL-DEVICES <SORTED ALPHABETICALLY>

This command will print a list of all hardware devices defined by Norsk Data, sorted on device name or device number. The list contains all ND-100 devices and most Nord-10 devices used by the ND-100.

A complete list of hardware devices may be obtained through the command:

```
SET-PARAMETERS
```

by answering "YES" to the question:

```
"Include ALL devices in LIST-... commands (YES/NO):"
```

## LIST-OVERLAPPING-DEVICES

This command will print a list of the devices with conflicting device numbers.

**Example:**

```
TPE>List-overlapping-devices <CR>

                              FIRST   LAST   IDENT
HARDWARE DEVICE NAME          DEVNO  DEVNO   -CODE

EXTERNAL INTERRUPT      2       20      23       6   Overlaps with
REAL TIME CLOCK         3       20      23       6

SYNCHRONOUS MODEM       7      160     167      40   Overlaps with
LINE PRINTER            3      160     163      35
LINE PRINTER            4      164     167      36

SYNCHRONOUS MODEM       8      170     177      10   Overlaps with
LINE PRINTER            5      170     173      33
LINE PRINTER            6      174     177      32
.
.

==ESCAPE==
```

## SET-PARAMETERS

This command is used to alter the mode of the RUN and LIST commands. The parameters are:

- Include memory map in the RUN command (Yes/NO)
  (Default is YES)

- Include ALL devices in the LIST-... commands (YES/NO)
  (Default is NO)

# Chapter 5 DISC-TEMA

# Chapter 5 DISC-TEMA

This chapter describes the DISC-TEMA test program.

## 5.1 General

The program DISC-TEMA is intended to be a service program used for testing disk or disk-controllers. The testing is mainly aimed at the functionality of the drive and the disk-system.

**The Disk Types:**

These are the disk types available in the I11 version of Disc-Tema:

1) DISC-14MB-1
2) DISC-14MB-2
3) DISC-16MB-1
4) DISC-16MB-2
5) DISC-21MB-1
6) DISC-21MB-2
7) DISC-23MB-1
8) DISC-23MB-2
9) DISC-28MB-1
10) DISC-28MB-2
11) DISC-30MB-1
12) DISC-30MB-2
13) DISC-38MB-1
14) DISC-38MB-2
15) DISC-45MB-1
16) DISC-45MB-2
17) DISC-60MB-1
18) DISC-60MB-2
19) DISC-70MB-1
20) DISC-70MB-2
21) DISC-74MB-1
22) DISC-74MB-2
23) *DISC-75MB-1
24) DISC-75MB-2
25) DISC-75MB-3
26) DISC-75MB-4
27) DISC-90MB-1
28) DISC-90MB-2
29) DISC-2-70MB-1-F
30) DISC-2-70MB-2-F
31) DISC-2-70MB-3-F
32) DISC-2-70MB-4-F
33) DISC-140MB-1-F
34) DISC-140MB-2-F
35) DISC-140MB-3-F
36) DISC-140MB-4-F
37) DISC-2-75MB-1
38) DISC-2-75MB-2
39) DISC-3-75MB-1
40) DISC-3-75MB-2
41) DISC-3-75MB-3
42) DISC-3-75MB-4
43) DISC-225MB-1-R
44) DISC-225MB-2-R
45) DISC-225MB-3-R
46) DISC-225MB-4-R
47) DISC-4-70MB-1-R
48) DISC-4-70MB-2-R
49) DISC-4-70MB-3-R
50) DISC-4-70MB-4-R
51) DISC-288MB-1-R
52) DISC-288MB-2-R
53) DISC-288MB-3-R
54) DISC-288MB-4-R
55) DISC-4-70MB-1-F
56) DISC-4-70MB-2-F
57) DISC-4-70MB-3-F
58) DISC-4-70MB-4-F
59) DISC-288MB-1-F
60) DISC-288MB-2-F
61) DISC-288MB-3-F
62) DISC-288MB-4-F
63) DISC-4-70MB-1-E
64) DISC-4-70MB-2-E
65) DISC-4-70MB-3-E
66) DISC-4-70MB-4-E
67) DISC-288MB-1-E
68) DISC-288MB-2-E
69) DISC-288MB-3-E
70) DISC-288MB-4-E
71) DISC-6-70MB-1-F
72) DISC-6-70MB-2-F
73) DISC-6-70MB-3-F
74) DISC-6-70MB-4-F
75) DISC-2-225MB-1-F
76) DISC-2-225MB-2-F
77) DISC-2-225MB-3-F
78) DISC-2-225MB-4-F
79) DISC-450MB-1-N
80) DISC-450MB-2-N
81) DISC-450MB-3-N
82) DISC-450MB-4-N
83) DISC-6-70MB-1-N
84) DISC-6-70MB-2-N
85) DISC-6-70MB-3-N
86) DISC-6-70MB-4-N
87) DISC-2-225MB-1-N
88) DISC-2-225MB-2-N
89) DISC-2-225MB-3-N
90 DISC-2-225MB-4-N
91) DISC-450MB-1-N
92) DISC-450MB-2-N
93) DISC-450MB-3-N
94) DISC-450MB-4-N

* - This is the default disc type, chosen when answering CR. This will change to the type selected, and in command SET-DISC-TYPE, the last selected will be default.

## 5.2 Starting and Initial Action

The DISC-TEMA is started from the TPE-MONITOR by the command:

```
TPE>Load DISC-TEMA <CR>
```

The program starts by asking for a DISC TYPE. Most installations have only one DISC TYPE, therefore the program asks only once for the DISC TYPE. The type you specify will then be both the "TO" and the "FROM" disc.

You may also use the command SET-DISC-TYPE to specify the different disc types.

The default DISC TYPE is DISC-75MB-1.

## 5.3 The Program Commands

The following program commands are available in Disc-Tema:
CHANGE
CLEAR-DEVICE
PRIORITY-SELECT
COMPARE
MATCH
COPY
DUMP-DISC-CONTENT
FORMAT
PARITY-CHECK
SET-DISC-CONTENT
SEEK
TRANSLATE
VERIFY
ALIGN
SET-DISC-TYPE
DUMP-FLAW-TABLE
SET-PARAMETERS
CLEAR-COUNTERS
FUNCTION
SCOPE-LOOP
RANDOM
REFRESH
ALLOCATE-BUFFERS

### CHANGE

This command is used to inspect and change single words on the disk.

# SET-PARAMETERS

You can run some of the program-commands in different modes. These modes can be changed by the SET-PARAMETER command. The different parameters are described below.

```
Directory-mode on (Yes/No):
Marginal recovery allowed (Yes/No):
ECC correction allowed (Yes/No):
Single surface format (Yes/No):
Override obligatory reallocating while formatting (Yes/No):
Number of retries on READ (0-64 Dec.):
Number of retries on WRITE (0-64 Dec.):
Number of test patterns to use during formatting (1-3 Dec.):
No. of bits to accept before reallocating (1-11 Dec.):
Skip any test in FUNCTION (Yes/No):
Number of times for FUNCTION to loop (1-65535.):
Do you want to release units after operations (Yes/No):
```

**Explanation**

Default and initial values are in brackets, <DEFAULT>.

**Directory mode on (YES or NO): <YES>**
This command defines how to calculate start addresses and amounts.

Answer = YES:
When in directory mode, start and stop addresses for the from or to disk are defined when unit- (subunit-) number is given. The amount to transfer is also defined, and if different directory sizes, the least of the two is transfered.

Answer = NO:
When in "non directory mode", the operator must give the starting point (sector address). The amount (number of sectors) to transfer must also be given. If FROM and TO disk is selected, then two start points must be given.

**Marginal recovery allowed (YES or NO): <YES>**
Only relevant for ECC disks. ( 10 or 15 Megahertz )

If allowed, a read/write operation giving error will lead to a retry with the heads slightly adjusted off track. First in one direction, then the other. If any of the "off track" operations give a good status, the operation is terminated as OK.

For a thorough test, marginal recovery should not be allowed.

**ECC correction allowed (YES or NO): <YES>**

Only relevant for ECC disks. ( 10 or 15 Megahertz )

In effect only when error during a read-operation from the disk. The data has been read into memory, the DMA operation is terminated. The status on the controler after read is indicating error, but data is correctable. The driver (program in ND100 CPU) will then perform a correction of the data read from disk by using the ECC. The max. number of bits to be corrected is 11.

For a thorough test, ECC operation should not be allowed.

**Single surface format (YES or NO): <NO>**
Should not be used. Only for debugging HW. Documentation may be obtained from DOCUS on special request.

**Override obligatory reallocating**
When formatting any disk, you may skip reallocating of failing tracks. This is not recomended.

**No of retries on READ (0 to 64 dec): <27>**
If a read from the disk gives a bad status, you may repeat read-operation. The number of repeats, before terminating with error exit, is decided by the number of read retries.

For a thorough test, READ retries should be 0.

**No of retries on WRITE (0 to 64 dec): <3>**
If a write from the disk gives a bad status, you may repeat the write-operation. The number of repeats, before terminating with error exit, is decided by the number of write retries.

For a thorough test, WRITE retries should be 0.

**Number of testpatterns to use during formatting (0 to 3 dec): <3>**
When a track is formatted, a write- and read-operation is performed. You may select how many times to do this by answering 0,1,2 or 3 test write/read to be done. The more testing you specify, the better assurance for an error to be detected, and bad tracks reallocated.

**Number of bits to accept before reallocating (1 to 11 dec): <1>**
Only relevant for ECC disks. ( 10 or 15 Megahertz )

In effect only when error occures during a read-operation from the disk. The data has been read into memory, the DMA operation is terminated. The status on the controler after read is indicating an error. The number of bits to be corrected is given in this command. The driver (program in ND-100 CPU) will then perform a correction on the data read from disk by using the ECC.

For a thorough test, 1 bit should be selected. If ECC is not allowed, this command has no effect.

**Skip any test in function (YES or NO): <NO>**
Only relevant for ECC disks. ( 10 or 15 Megahertz )

If a particular test in function is to be skipped, you may do so by answering YES. Then D-T will promt each test in function, and ask if the test is to be included or not.

**Number of times for function to loop (1 to 65535 dec): <4>**
You may specify how many times for function test repeat itself. Escape will also terminate function test.

**Do you want to release units after operations (YES or NO): <NO>**
Only relevant when you have a disk switch and ECC type disk. D-T may issue a release command to the switch after certain commands if enabled (YES).

## CLEAR-DEVICE

This command moves the Read/Write-head back to track 0 (return-to-zero seek). The "FROM" disk (see the command SET-DISC-TYPE) is used.

**Example:**

```
TPE>clear-device <CR>

DISC-75MB-1
Unit (0-1 Oct.):0 <CR>
TPE>
```

## PRIORITY-SELECT

This command is used for disks with dual port option.

When PRIORITY-SELECT is used for selecting a disc unit, the other channel cannot be used.

## COMPARE

This command is used to compare data from two sources. The mechanism is implemented partially in hardware. By using a bitcompare on the interface, data fetched from memory is compared to data read from disc.

If you have an ECC error, you will get an errormessage. Your data may still be OK.

**Example:**

```
TPE>compare <CR>

FROM DISC-75MB-1
UNIT NUMBER (0-3 OCT.):0 <CR>
CYLINDER NUMBER (0-1466 OCT.):0 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):0 <CR>
SECTOR NUMBER (0-21 OCT.):0 <CR>

FROM DISC-75MB-1
UNIT NUMBER (0-3 OCT.):0 <CR>
CYLINDER NUMBER (0-1466 OCT.):0 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):0 <CR>
SECTOR NUMBER (0-21 OCT.):0 <CR>
AMOUNT (NO. OF SECTORS) (1-220526 OCT.):220526 <CR>
DISC AREAS OVERLAP EACH OTHER. OK ? (YES/NO):YES <CR>
BLOCK SIZE (1-132 OCT.):132 <CR>
TPE>
```

## VERIFY

This verifies the contents of two specified disk areas. The contents are read and compared, word by word.

**Example:**

```
TPE>verify <CR>

FROM DISC-75MB-1
UNIT NUMBER (0-3 OCT.): 0 <CR>
CYLINDER NUMBER (0-1466 OCT.):14 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):11 <CR>
SECTOR NUMBER (0-21 OCT.):21 <CR>

FROM DISC-75MB-1
UNIT NUMBER (0-3 OCT.): 2 <CR>
CYLINDER NUMBER (0-1466 OCT.):14 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):1 <CR>
SECTOR NUMBER (0-21 OCT.):21 <CR>
AMMOUNT (NO. OF SECTORS) (1-216374 OCT.):1 <CR>
TPE>
```

## MATCH

This command is a combination of the two commands COMPARE and VERIFY. COMPARE is executed until a compare error is detected. Then it works like the command VERIFY on the spot where the compare error was detected.

The command cannot be run under SINTRAN.

**Example:**

```
TPE>match <CR>

FROM DISC-75MB-1
Unit (0-1 Oct.):0 <CR>

TO DISC-75MB-1
Unit (0-1 Oct.):0 <CR>
TPE>
```

## COPY

**NOTE:** This command will "destroy" the data on the destination disk.

The command copies data from one disk area to another disk area. The command cannot be run under SINTRAN.

**Example:**

```
TPE>Copy <CR>

FROM DISC-75MB-1
Unit (0-1 Oct.):1 <CR>

TO DISC-75MB-1
Unit (0-1 Oct.):0 <CR>
TPE>
```

## PARITY-CHECK

You can use this command to test whether or not the program is able to read addresses and data without getting ECC-error. The contents of one disk area is read without being stored in memory. The disk defined as the "FROM" disk is used.

**Example:**

```
TPE>parity-check <CR>

FROM DISC-75MB-1
Unit (0-1 Oct.):1 <CR>
TPE>
```

## TRANSLATE

This command can translate as follows:

| | | |
|---|---|---|
| Physical address | ←→ | Logical address |
| Physical address | ←→ | Page address |
| Logical address | ←→ | Page address |
| Disc status | → | Decoded text |

**Example:**

```
TPE>Translate <CR>

ADDRESS TYPE:help <CR>
1) *PHYSICAL-ADDRESS
2)  LOGICAL-ADDRESS
3)  PAGE-ADDRESS
4)  EXIT
5)  STATUS
ADDRESS TYPE:physical-address <CR>

DISC-75MB-1
Cylinder number (0-1466 Oct.):5 <CR>
Surface  number (Data head) (0-4 Oct.):0 <CR>
Sector   number (0-21 Oct.):21 <CR>

------------------------------------------------
*      * CYLI * SUR  * SEC * LOGICAL * PAGE NO.*
*      * NDER * FACE * TOR * ADDRESS *         *
------------------------------------------------
* OCT. *    5 *    0 *  21 *     723 *     351 *
------------------------------------------------
* DEC. *    5 *    0 *  21 *     467 *     233 *
------------------------------------------------
```

## ALIGN

You can only run this command on drives with removable disk packs. You use the command when aligning the read/write-heads. You also need a special ALIGN disk-pack.

**Example**

```
TPE>Align <CR>
DISC-75MB-1
Unit (0-3 Oct): 0 <CR>
CYLINDER (0-1466 Oct): 365 <CR>
HEAD (0-4 Oct) 0 <CR>

Read is activated on head  0

Press:
C=> D-T asks for cylinder number.
E=> Exit from align.
```

```
F=> First head (0) is chosen, activate read is performed.
H=> D-T asks for head number.
L=> Lists head and cylinder currently selected.
N=> Next head is selected, activate read is performed.
P=> Previous head is selected, activate read is performed.
R=> Return to zero seek will be performed.
```

## SET-DISC-TYPE

You can use this command to define the "FROM" and "TO" disk. They are both set to the same at the start of the test program, so this command is only useful if you have different disk types.

**Example:**

```
TPE>set-disc-type <CR>


SET NEW "FROM" DISC NAME:<CR>DISC-75MB-1<CR>
                                                (Default
SET NEW "TO" DISC NAME:<CR>DISC-75MB-1<CR>       -value)
TPE>
```

## DUMP-FLAW-TABLE

This command prints the contents of the FLAW-table, containing information about BAD TRACKS/ALTERNATIVE TRACKS. This information is found on the last track of the disk.

**Example:**

```
TPE>dump-flaw-table <CR>

DISC-140-1-F
Unit  (0 to 3 oct): 0 <CR>

This disk is formatted with DISC-TEMA-I.11

No.of reallocated tracks: 2
*******************************************
*      BAD TRACKS      * ALTERNATIVE TRACKS *
* CYLINDER *  HEAD    * CYLINDER *  HEAD    *
*******************************************
*      37         3         801          0   *
*      41         3         801          0   *
*******************************************
All numbers are decimal.

TPE>
```

# FUNCTION

**NOTE:**

This command can "destroy" data on the disk.

Function is mainly a test of the disk interface, and the disk drive and the disk pack is only partially tested.

**Example:**

```
TPE>function <CR>

DISC-75MB-1

On this disk type FUNCTION will "destroy" data in the last
cylinder in the spare track buffer pool.
Most new disks have an area reserved for function test. The
data pool and the flaw-table are not affected by the
function test.

DISC-TEMA will give a warning ONLY when data may be lost.



Do you still want to continue (Yes/No):Y <CR>

Specify from 1 to 4 unit numbers, terminated by CR. Separate
the numbers with a space or a comma (,). The units specified
must be turned ON and ready. The units NOT specified must be
turned OFF, or nonexistent.
UNIT(S) (0-3 Dec.): 0 1 2 3 <CR>
TPE>
```

# FORMAT

**NOTE:** This command will destroy the existing data on the disk.

The FORMAT command writes the address part of the sectors

**Example:** Formatting "SMD, FUJITSU, MMD, RSD AND FSD 340":

```
TPE>format <CR>

DISC-140MB-1-F
Unit (0-3 Oct.): 0 <CR>

The previous content will be destroyed on
Disc system : 1 Unit : 0

Do you still want to continue (Yes/No): Y <CR>

This disk is formatted with DISC-TEMA-I.11   (**** PRE-
RELEASE ****)

These tracks will be reallocated:

************************
*  CYLINDER  *   HEAD  *
*  Dec.  Oct.*  (Dec.) *
************************
*   14    16 *     3   *
*   19    23 *     8   *
*   20    24 *     8   *
*  346   532 *     2   *
*  366   556 *     2   *
*  370   562 *     3   *
*  400   620 *     4   *
*  456   710 *     5   *
************************
Do you want to edit the table (yes/no): Y <CR>

*: help <CR>
 1) APPEND
 2) DELETE
 3) LIST-TABLE
 4) REMOVE-TABLE
 5) START-FORMATTING
*:remove-table <CR>
REMOVE-TABLE

Delete the whole table (Yes/No): Yes <CR>
*:list-table <CR>

Table of reallocated tracks is empty !
*:append <CR>
APPEND

Input by cylinder or surface: Cylinder <CR>
CYLINDER
```

```
Cylinder      (0-822 Dec.):40
Head (Surface) (0-9 Dec.): 0

Cylinder      (0-822 Dec.):42
Head (Surface) (0-9 Dec.): 7
   .      .     .      .
   .      .     .      .
   .      .     .      .
   .      .     .      .
   .      .     .      .
Cylinder      (0-822 Dec.):766
Head (Surface) (0-9 Dec.): 9

Cylinder      (0-822 Dec.):786
Head (Surface) (0-9 Dec.): 8

Cylinder      (0-822 Dec.): <CR> Finished

*:list-table <cr>
LIST-TABLE

***********************
*  CYLINDER  *   HEAD  *
*  Dec.  Oct.*  (Dec.) *
***********************
*   40    50 *     0   *
*   42    52 *     7   *
*   93   135 *     6   *
*  144   220 *     4   *
    .     .        .
    .     .        .
    .     .        .
*  766  1376 *     9   *
*  786  1422 *     8   *
***********************
*:start <CR>
START-FORMATTING

*---------------------------*
* BAD TRACKS  * SUBST TRACK *
*---------------------------*
* CYLI-* SUR- * CYLI-* SUR- *
* NDER * FACE * NDER * FACE *
*---------------------------*
*   50 *    0 * 1441 *    0 *
*   52 *    7 * 1441 *    1 *
*  135 *    6 * 1441 *    2 *
*  220 *    4 * 1441 *    3 *
    .       .      .       .
    .       .      .       .
    .       .      .       .
* 1376 *   11 * 1441 *   11 *
*---------------------------*
No. of reallocated tracks (dec.) 20
TPE>
```

## DUMP-DISC-CONTENT

With this command you can dump the contents of a disk area on the device specified by the command

SET-PRINTER-DEVICE-NUMBER.

The contents of the specified area will the be listed on that device, line by line.

If the line to be listed is identical to the previous line, the word "SAME" will be listed (see the example below).

The "FROM" disk is used.

**Example:**

```
TPE>dump-disc-content <CR>


DISC-75MB-1
UNIT NUMBER (0-3 OCT.): 0 <CR>
CYLINDER NUMBER (0-1466 OCT.):300 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):2 <CR>
SECTOR NUMBER (0-21 OCT.):7 <CR>
AMOUNT (NO. OF SECTORS) (1-22 OCT.):1 <CR>

DISC DUMP OF
DISC SYSTEM : 1 UNIT : 0
CYLIN : 300 SURFACE : 2 SECTOR : 7

  0 123456 123457 123460 123461 123462 123463 123464 123465
 10 123466 123467 123500 123501 123502 123503 123504 123505
 20 123506 123507 123510 123511 123512 123513 123514 123515
 30 123516 123517 000000 000000 000000 000000 000000 000000
 40 000000 000000 000000 000000 000000 000000 000000 000000
SAME
750 000000 000000 000000 000000 442200 125100 177777 044403
760 063204 014560 025400 014230 422340 177400 166427 024533
770 025604 032132 021620 032126 764731 154100 623207 054127
TPE>
```

## SET-DISC-CONTENT

When using this command, you can write one or more sectors on the disk. You must specify the pattern, and this is done by answering the questions:

```
FIRST WORD (0-177777 OCT.):
and MODIFIER (0-177777 OCT.):
```

The pattern specified as FIRST WORD is written into the FIRST word on the first sector specified.

The first word plus the MODIFIER is written into the SECOND word.

The second word plus the MODIFIER into the THIRD word and so on.

**Example:**

```
TPE>set-disc-content <CR>


DISC-75MB-1
UNIT NUMBER (0-3 OCT.): 0 <CR>
CYLINDER NUMBER (0-1466 OCT.):700 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):0 <CR>
SECTOR NUMBER (0-21 OCT.):4 <CR>
AMMOUNT (NO. OF SECTORS) (1-101722 OCT.):10 <CR>
BLOCK SIZE (1-132 OCT.):11 <CR>
FIRST WORD (0-177777 OCT.):0 <CR>
MODIFIER (0-177777 OCT.):1 <CR>
TPE>
```

## SEEK

This command starts consecutive seeks between two specified disk addresses. You can stop the seeking by pressing the ESCAPE button on the keyboard.

The "FROM" disk is used.

**Example:**

```
TPE>seek <CR>

FROM DISC-75MB-1
UNIT NUMBER (0-3 OCT.): 0 <CR>
CYLINDER NUMBER (0-1466 OCT.):0 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):0 <CR>
SECTOR NUMBER (0-21 OCT.):0 <CR>


TO DISC-75MB-1
CYLINDER NUMBER (0-1466 OCT.):1466 <CR>
SURFACE NUMBER (DATA HEAD) (0-4 OCT.):2 <CR>
SECTOR NUMBER (0-21 OCT.):11 <CR>
```

**Comment:**

NOTE that you must press ESCAPE to terminate the SEEK-command.

## CLEAR-COUNTERS

With this command you can clear the counters found in the last line of the program status. You may use the command PROGRAM-STATUS to check these counters.

**Example:**

```
TPE>clear-counters <CR>

CLEAR ECC COUNTER (YES/NO):no <CR>
CLEAR WRITE RETRY COUNTER (YES/NO):no <CR>
CLEAR READ  RETRY COUNTER (YES/NO):no <CR>
CLEAR MARGINAL RECOVERY COUNTER (YES/NO):no <CR>
CLEAR TRANSFER COUNTER (YES/NO):no <CR>
TPE>
```

## RANDOM-TEST

The test is intended to be a media-checking and a servo-stressing test.

Typing

```
TPE>RANDOM <CR>
RND:
```

will enter a sub-menu allowing different test parameters to be defined:

```
RUN-TEST
ADDRESS-MODE
TEST-TYPE
LIMIT-ADDRESSES
AMOUNT-SET
WRITE-PROTECT
LIST-PARAMETERS
SET-PARAMETERS
EXIT
```

If the RUN command is given, without altering initial test parameters, the operation carried out will be:

READ on first sector,
READ on last sector,
COMPARE on first sector,
COMPARE on last sector and
the same operations on decreasing/increasing sector address.

The DATA on the disk media will NOT normally be ruined by D-T random test, unless WRITE is specified. To enable write, you must bypass write protect.

**RND:RUN-TEST<CR>**
The testing starts on the basis of the selected parameters. A message is written on the console, listing the operations to be carried out on the disk and the commands that can be given during the run without interrupting the testing.

**RND:ADDRESS-MODE<CR>**

Two ways of addressing may be selected:

1: INCREMENT-DECREMENT is similar to the way SUPER-RANDOM or WINCHESTER-RANDOM generated the addresses.

2: RANDOM generates addresses "randomized". A new start (RUN) of testing will generate the same pattern unless size of test area is altered.

**RND:TEST-TYPE<CR>**

You may specify what type of operations are to be carried out on the disk media.

1 read parity: D-T checks the parity at the address calculated
by D-T. No data is transferred to memory.

2 read only: D-T reads at the address calculated by D-T. Data is transferred to memory.

3 read-compare: D-T reads data from disk into memory, the interface then compares memory buffer and disk content.

4 write-compare: D-T writes data onto disk. The interface then compares memory buffer with disk content.

5 seek-only: A seek to calculated address (track) is generated. When "ON CYLINDER", the operation is finished.

(Default & initially read and compare is selected)

**RND:LIMIT-ADDRESSES<CR>**

Testing is restricted to the directory area, and can be restricted even further. A minimum of two sectors must be selected as disk-test area. If the area is small, you should not select "random addressing".

The reason for this is that testing will be slowed down by the extra time used by the random generator to find an address within the selected area. (The whole directory is default and initial)

**RND:AMOUNT-SET<CR>**

If the address generator is "random", you are allowed to specify an amount (number of sectors) to be transferred.

The amount can be from one sector up to the maximum buffersize of the memory, or one cylinder of the disk media. If "random" amount is selected, this will be one or more sectors, but less than the full buffer or one cylinder.

When address generating is of the "old" type (increment-decrement), you may only select one sector as amount.

When operation is SEEK, the amount has no influence. (Initially, one sector is selected as amount).

**RND:WRITE-PROTECT<CR>**
This command is only a threshold that can be set to prevent accidental selection of write operation. The setting of this protect flag makes the program ask if the operator wants to continue when write is specified as an operation.

(Initially, this flag will be set to ON.)

**RND:LIST-PARAMETERS<CR>**
This command starts a printout of the selected parameters on the console.

**RND:MODE<CR>**
The "strength" of the test can be set by this command. When the disk is of type ECC (10 or 15 Megahertz), you may specify whether or not marginal recovery or/and ECC correction is allowed.

For ECC type and ST506 type, you also may set number of read and write retries.

Initially ECC correction and marginal recovery are not allowed. Initially no read or write retries are allowed.

**RND:EXIT<CR>**
You are leaving the random test, and returning to disc-tema. All parameters are reset.

## REFRESH

The refresh facility provides a means of formatting and restoring one track on the disk media. This is the only way to "fix" an address-mismatch type error without formatting the whole media. The refresh facility naturally also solves ECC or CRC type errors. Most disks can be refreshed, but Phoenix disks cannot be refreshed in Disc-Tema-I11.

Experience shows that a ruined data area on a disk will appear where the heads most frequently are positioned. This is the case, at least, for the errors caused by an abnormal situation in the system, i.e. power surge, noise or drop-out.

**NOTE:**

Using REFRESH to "repair" a track ruined this way, does not guarantee that your data is valid.

You should find a way to confirm your software after you have made a backup of it.

**The refresh strategy:**

The selected "FROM" disk is used.

The operator must specify a unit number. The program then reads the alternative track table if there is one. A memory buffer big enough to hold one track is "zeroed".

Then you must specify the cylinder and surface. Cyl. and surf. may be found by the command TRANSLATE if the page or logical address is known.

The program checks the alternative track table to ascertain whether or not this track has previously been reallocated. When attempting to refresh a reallocated track you will receive a WARNING, and you should not continue.

**Example:**

```
TPE>refresh <CR>
Unit  (0 to 3 oct): 0 <CR>

Alternative track table is read from disk

Cylinder  (0 to 1440 oct): 37 <CR>
Surface  (0 to 11 oct): 3 <CR>

Read OK on sector:   0
Read OK on sector:   1
Read OK on sector:   2
Read OK on sector:   3
Read OK on sector:   4
Read OK on sector:   5
Read OK on sector:   6
Read OK on sector:   7
Read OK on sector:   10
Read OK on sector:   11
Read OK on sector:   12
Read OK on sector:   13
Read OK on sector:   14
Read OK on sector:   15
Read OK on sector:   16
Read OK on sector:   17
Read OK on sector:   20
Read OK on sector:   21
All sectors on the track are read, do you
want to reformat and write them back  (YES or NO): yes <CR>

Done
TPE>
```

# ALLOCATE-BUFFERS

This command is for inspecting and altering memory buffers for disk DMA transfer.

You use this command mainly when you have a disk system number two, separate I/O crate (MPM4) and MPM5.

When Disc-Tema is loaded, the buffers are calculated on the assumption that the disk controller(s) have uniform work buffers and that DMA transfer from bank 0 and up is possible.

If there is a non-uniform configuration of memory channels in MPM5 systems, it is necessary to move the buffer area for disk DMA to an address area where the disks have a common buffer with no address offset.

The command may also be convenient if you for some reason want to run DMA traffic to a given area in memory.

After entering the command, a sub menu and a promt are displayed.

```
TPE>allocate-buffers <CR>

Buffalo: exit <CR>
EXIT
TPE>
```

The commands are:

1) *SHOW-BUFFERS
2) MANUALLY-ALLOCATE-BUFFERS
3) AUTOMATIC-ALLOCATE-BUFFERS
4) EXIT

1) SHOW-BUFFERS
This command shows the selected DMA buffer's size and start address at bank number.

2) MANUALLY-ALLOCATE-BUFFERS
You may force DMA traffic into whatever part of the memory you want, except bank 0. You specify the startpoint of the buffer area and size. If you are giving more memory than D-T has found present, you will receive a warning. Ignoring this will eventually lead to a memory-out-of-range interrupt, and termination of the program.

3) AUTOMATIC-ALLOCATE-BUFFERS
This command starts a routine that investigates DMA ways on its own. The disk-controller(s) send a pattern (in testmode) by the DMA ways into memory.(1) The pattern is investigated by D-T, and a map of the memory is made. The map is the basis for creating DMA-buffers. For ST506-type interfaces (Micropolis or Finch) this command is not possible.

## 5.4 Error Messages

Error messages consist of:

TEXT, see the alphabetical list later in this section.

PRINT NOTES, appearing as "2)", "3)" etc. These are explained later in this section.

STATUS-CODE, found in a 16 bit register on the controller. You can find this status by using the command TRANSLATE.

## Print Note

These print notes: 1), 2), 3), etc. appear with some of the error messages.

The explanation of these print notes:

1) The controller did not respond to the IOX-instruction from the CPU.

   Possible errors:
   - Error in the controller
   - Wrong device number (switch-setting)
   - Controller is missing

2) An interface has given an interrupt on level 11, and this is not legal.

3) Error on the controller.

4) Same as 3).

5) Error on the interrupt-system.

6) Timeout without HW-status error from the disk.

7) If the CPU and the controller have different DMA-channels (ports), you cannot provoke overrun (as in f.ex MPM-5 systems).

8) The test found a bad track that was not in the flaw table.

## Error Messages, alphabetical list

**Abnormal completion**
Status bit 12 is decoded. Transfer operations on the disk are destroyed. Caused by HW-error on the disk (could also be on the controller).

**Address mismatch**
Status bit 8 is decoded. Probably a SW-error. Any error in the address-field on the specified track (or sector) leads to this error condition. Format or refresh the disk and see if the error disappears.

**Block address reg. 1 error 3)**
Error on the data-way between the controller and the CPU.

**Compare error**
Status bit 10 is decoded. Data from memory and data from the disk are compared on the controller. If there is a mismatch, this error message appears.

**Controller not active after activate**
Status bit 2 is decoded. This error message is probably caused by a serious error on the disk drive.

**Core address register not as specified**
This means that a disk transfer did not finish, probably because of error on the data-way between the controller and the CPU.

**CRC error**
Status bit 9 is decoded. SW-error in the data-field on a track (or a sector). You can correct this error by formatting or refreshing the disk.

**Data error**
Error in the datafield on a track (or sector) on ECC-type disks.

**Data way error**
Error on the dataway between the controller and memory.

**Disc unit not ready**
Status bit 13 is decoded. The disk has not started, or is not able to start (ECC disks only).

**DMA channel error (FIFO over/underrun or ND-100 Bus error)**
Error between the controller and memory.

**Error in test-mode**
Error on the DMA-way between the controller and memory, or error on the controller.

**Error. After COMPARE**
Error after timeout. This general HW-error could be caused by almost anything in the system.

**Error. After READ**
Error after timeout. This general HW-error could be caused by almost anything in the system.

**Error. Not on cylinder**
HW-error on the disk drive.

**Error. Status bit <octal word>**

| Status bit no: | Explanation: |
|---|---|
| 0 | Controller not active interrupt enabled |
| 1 | Error interrupt enabled |
| 2 | Controller active |
| 3 | Controller finished with a device operation |
| 4 | Inclusive OR of errors (bits 5-13) |
| 5 | Illegal load, i.e. load while the unit is not on cylinder |
| 6 | Timeout |
| 7 | Hardware error (disk fault + missing read clocks + missing servo clocks + ECC parity error) |
| 8 | Address mismatch |
| 9 | CRC error |
| 10 | Compare error |
| 11 | DMA channel error |
| 12 | 0 |
| 13 | Disc unit not ready |
| 14 | On cylinder |
| 15 | 0 used to distinguish from 10 Mb |

**ERROR. The core address reg. stopped before 0170 words were written (Parity error expected).**
Possible errors:

1) Controller error
2) Error on disk pack
3) Error in the address field on the test-track

In order to test the ECC-CRC, Disc-Tema writes parity errors in the data area. If it is impossible to correct these errors, this error message will be written.

**Hardware error (Disc fault + Missing servo clocks +**
**Missing servo clocks + ECC parity error)**
Status bit 7 is decoded. HW-error on the disk drive.

**Illegal load**
Disc-Tema has initialized the controller while the disk was active.

**Log. block address out of range.**
Disc-Tema tried to access an address outside the disk's area.
Possible reason: CPU/MMS malfunction.

**No align cylinder for this disk.**
You need a special ALIGN-pack to run ALIGN.

**No alternative track table found**
Possible errors:

1) HW-error on the disk drive.
2) The disk has not been formatted.
3) The table-track has been destroyed.

**No table on this disk type**
The disk has no table showing reallocated tracks.

**No tracks in the spare track pool are used.**
The disk has no bad spots.

**Not on cylinder.**
HW-error on the disk drive. Status bit 14 is decoded.

# Chapter 6 FLOPPY-STREAMER test program

# CHAPTER 6 FLOPPY-STREAMER TEST PROGRAM

This chapter describes the FLOPPY-STREAMER test program.

## 6.1 GENERAL

The program can test the following cards :

3027 : Floppy disc controller.
3106 : Floppy and streamer.
3112 : Floppy and streamer 8" & 5.25".

In addition the floppy drive/streamer itself is tested.

An ND-100 normally has only one floppy interface and one drive. This interface/drive is then used to load the test program.

If loading and starting up the program is successful, you already know that the interface and the drive can function quite well for read and bootstrap operations.

If you cannot load the test program, you may get a load error. In such a case you must use TELEFIX to place the program in ND-100.

You can also read the load-error code in the error code display on the interface. On the next page you will find a table showing the different error codes and their meanings.

If an error condition exists, and you do not suspect any specific part, you should start by running the command FUNCTION on both floppy and streamer. This could verify whether or not the error is on a drive or on an interface.

**NOTE:** It is important to remember that looping on functions that use the media for a long period may cause overheating problems and dirty drive heads.

ERROR CODE DISPLAY:

| Display: | Meaning: | |
|---|---|---|
| 000 | OK | |
| 001-004 | Not used | |
| 005 | CRC error | |
| 006 | Sector not found | |
| 007 | Track not found | |
| 010 | Format not found | |
| 011 | Diskette defect (impossible to format) | |
| 012 | Format mismatch | |
| 013 | Illegal format specified | |
| 014 | Single-sided diskette inserted | |
| 015 | Double-sided diskette inserted | |
| 016 | Write-protected diskette | |
| 017 | Deleted record | |
| 020 | Drive not ready | |
| 021 | Controller busy on start | |
| 022 | Lost data (over or underrun) | |
| 023 | Track zero not detected | |
| 024 | VCO (Voltage-Controlled Oscillator) frequence out of range | |
| 025 | Microprogram out of range | |
| 026 | Timeout | |
| 027 | Undefined error | |
| 030 | Track out of range | |
| 031 | Not used | |
| 032 | Compare error (during compare of data) | |
| 033 | Internal DMA errors | |
| 034-037 | Not used | |
| 040 | ND-Bus-error command fetch | |
| 041 | ND-100 Bus-error status transfer | |
| 042 | ND-100 Bus-error data transfer | |
| 043 | Illegal command | |
| 044 | Word count not zero | |
| 045 | Illegal completion (count. transfer) | |
| 046 | Addr-reg. error | |
| 047 | Not used | |
| 050 | No bootstrap found on diskette | Error during |
| 051 | Wrong bootstrap (too old flo-mon version) | autoload |
| 052-053 | Not used | |
| 054 | End of recorded data | |
| 055 | Division error | |
| 056 | Filemark detected | |
| 057 | False interrupt | |
| 060 | Streamer-handshake error | |
| 061 | Streamer-status transfer error | |
| 062 | Bad cartridge | |
| 063 | No cartridge installed | |
| 064 | End of tape, cartridge full | |
| 065 | Streamer-drive error | |
| 066 | Unidentified exception | |
| 067 | Illegal command to streamer | |
| 070 | PROM-checksum error | |
| 071 | RAM error | |
| 072 | CTC error | |
| 073 | DMACTRL error | |
| 074 | VCO error | |
| 075 | Floppy-controller error | |
| 076 | Streamer-Data register error | |
| 077 | ND-100 register error | |

## 6.2 Starting and Initial Action

The FLOPPY-STREAMER is started from the TPE-MONITOR by the command:

```
TPE>Load FLOPPY-STREA <CR>
```

**Example:**

```
TPE>load Floppy-Strea <CR>

    FLOPPY-STREAM  - Version : x - yyyy-mm-dd

Found device number : 1560, 3112 Floppy streamer DMA 8" & 5.25"

TPE>select-device <CR>

Device name......: ?
Correct answers are:
FLOPPY-DISC-1
STREAMER-TAPE-1

Device name......: Floppy-disc-1 <CR>
Unit  (0 to3 Dec): 01 <CR>

Controller....: 3112 Floppy streamer DMA 8" & 5.25"
Micro program : F


TPE>
```

## 6.3 Error Messages

All errors are printed with the format :

```
 ***** ERROR < additional text >
       < additional text >
```

If the error is a status error from the interface, the following message will be printed:

```
***** ERROR XX < error text > ***** AT :  < time >
      Operation was        : < What function was carried out >
      Controller number    :         X
      Selected unit        :         X
      Format used          :         X
      Device start address :         X (Dec.)       X (Oct.)
      Transfer size        :         X (Dec.)       X (Oct.)
```

All available status can be printed out if specified by the SET-PARAMETERS command :

```
 Short error message (Yes/No): No <CR>
```

This status printout looks as the one you can get printed by the INFO command :

READ-EXTENTED-STATUS.

## 6.4 Program Commands

For Monitor commands, see chapter 1, The TPE Monitor.

```
PROGRAM-COMMANDS:                          Executable on :
*****************
SELECT-DEVICE                         <--  Floppy and Streamer
FUNCTION                              <--  Floppy and Streamer
SCOPE-LOOP                            <--  Floppy and Streamer
REWIND                                <--  Streamer
RETENTION                             <--  Streamer
TEST-CARTRIDGE-CAPASITY               <--  Streamer
FORMAT                                <--  Floppy
COPY-FLOPPY                           <--  Floppy
CHECK-FLOPPY/                         <--  Floppy
SET-PARAMETERS                        <--  Floppy and Streamer
INFO                                  <--  Floppy and Streamer
```

## FUNCTION

This command brings you into a new command list for functional tests. You can either run the functional tests one by one, or you can run them all in sequence by using the command

"RUN".

You may also skip some of the tests executed by the RUN command by the command SKIP-TESTS-IN-RUN.

**SKIP-TESTS-IN-RUN**
This command disables and enables tests from execution in the FUNCTION command. Single tests in PROVOKE ERRORS can also be enabled or disabled. Only the tests valid for the selected device are asked for.

**RUN**
This command executes all applicable tests for the selected

device. These tests may also be run separate by using the following commands:

**ACTIVATE-SELFTEST**
This command clears the device and waits for the self-test to finish.

**IOX DATAWAY TEST**
Verifies the controller's IOX data register. The test writes data to and from the controller's data register by using test macro number 2.

**DMA DATAWAY TEST**
Verifies the controller's DMA data transport. The test pattern is set up in ND-100 and test macro no 9 is used to write the data into the controller's Z80 memory on the card. With test macro number 10 the data is read from the Z80 memory to ND-100 and tested.

**Z80 RAM TEST**
Verifies the controller's local memory buffer. Different test patterns are written into the controller's memory by using test macro no 13. With test macro number 14 they are read back and tested.

**DISPLAY TEST**
This is a test to verify the controller's error display. The test will make the error display count from 0-9 in all digits by using test macro number 12.

**TEST FORMATS**
This test is only executed if FLOPPY is selected.

This test verifies whether or not the controller is able to format the floppy in all possible formats. The test track (can be changed in the SET-PARAMETERS command) is used for formatting. Only formats available on single-sided floppies are used on single-sided floppies, only formats available on double-sided floppies are used on double-sided floppies.

The test track is first formatted by using the function FORMAT TRACK. The track is then filled with a test pattern (WRITE function) and tested for errors with the CHECK FLOPPY function. At the end of the command the test track is reformatted to the original format.

**WEAR TEST**
This test is only executed if FLOPPY is selected.

The test will verify whether or not the signal from the floppy is steady and not declining. The test will write one sector at the test track with the test pattern and then read it back as often as is specified (SET-PARAMETERS command).

### OVER WRITE

This test is only executed if FLOPPY is selected.

This test will first write a test pattern into one sector in the test track and then read it. This is repeated a specified number of times. (SET-PARAMETERS command)

### ARM TEST

This test is only executed if FLOPPY is selected.

This is a random address write/read test on floppy to verify arm movement as under normal use and a general test for normal use of the controller and drive in read and write operations. When this test is run as a command, the number of writes/reads are specified.

The default number of loops is infinite when used as a command and can only be stopped by pressing ESC. When this test is run in the FUNCTION command the parameters are as set up in the SET-PARAMETERS command.

The sequence of a loop is as follows :

- Fill the buffer with test pattern.
- Generate two random addresses on the floppy.
- Put the address generated as the first word in the buffer and write one sector for both generated addresses.
- Read back the sector at the first generated address.
- Test that the first word in buffer contains the address of the sector.
- Read and test the last generated address.

This sequence is set up so that the arm must move to read and test the written sector.

**Example:**

```
FUNC>arm-test <CR>

Number of addresses tested (1-616 Dec.): Looping

== ESCAPE ==
```

### DATA TEST

This test is only executed if STREAMER is selected.

This test uses the tape in the same way as a normal write/read operation. It will test that the buffer-handling in the interface and drive does not mix blocks and that critical data is written on tape successfully. After each write or read sequence, the number of rewrites/rereads (status byte 2 and 3) is reported so that a media test can be carried out using this information as statistics.

When reading the tape, every block is tested to see if it has the correct value according to the expected block.

The command uses combined test patterns. Every block has its logical block address as data (address in address) and a write and/or read critical data pattern.

**Example:**

```
FUNC>data <CR>
Read/ Write mode..... : Write-and-read <CR>
Number of blocks (dec): 100 <CR>

Finish writing. 100 blocks written. 0 blocks rewritten.
Finish reading. 100 blocks read. 0 blocks reread.
```

### START-STOP-TAPE

The command will carry out write and read operations on tape with a delay between each transfer to provoke over/under run on the drive.

### FILEMARK

The command will write data and filemarks on the tape and then read it to check that correct blocks are found (address in address).

### PROVOKE-ERRORS

This test simulates some error conditions and checks whether or not the interface can detect them. If not, an error message will be printed. The test can be run as single tests all in sequence.

| | |
|---|---|
| E05 | CRC error |
| E06 | Sector not found |
| E07 | Track not found |
| E10 | Format not found |
| E12 | Format mismatch |
| E13 | Illegal format spesifed |
| E17 | Deleted record |
| E25 | Microprogram out of range |
| E26 | Timeout |
| E30 | Track out of range |
| E43 | Illegal command |
| E45 | Illegal completion |
| E54 | End of recorded data |
| E56 | File mark detected |
| E64 | End of tape |

## SCOOP-LOOP

This command has a set of sub-commands:

**THE EXIT COMMAND**
Exit from scope-loop sub-command set.
**MULTI-CCB**
This command has the possibility to execute several CCB blocks in a sequence. It has these sub-commands :

CLEAR-LIST
This command clears the current link of CCB blocks and status.

APPEND-LIST
This command appends a new CCB after the last appended. It asks for parameters like the command CHANGE-CCB-CONTENT.

EXECUTE-LIST
This command executes all CCB's in the link and saves the status for all of them.

PRINT-LIST
This command print out all the CCB blocks with status.

**THE WRITE COMMAND**
This command will write one sector/block on the media from the buffer. If loop mode is ON, then the device address (for floppy) is incremented according to the set-up in the SET-DEVICE-ADDRESS-ADDER command.

**THE READ COMMAND**
This command will read one sector/block from the media into the buffer. If loop mode is ON, then the device address (for floppy) is incremented according to the set-up in the SET-DEVICE-ADDRESS-ADDER command.

**THE DUMP-CCB-CONTENT COMMAND**
This command prints out the contents of the user-specified command block and status block for the interface (CCB = Command Control Block).

**THE CHANGE-CCB-CONTENT COMMAND**
This command gives the user the possibility to set up a CCB that can be activated with the command ACTIVATE-WITH-CCB-CONTENT.

**THE DUMP-BUFFER-CONTENT COMMAND**
This command prints out the buffer content in ASCII, octal

and hexadecimal values.

**THE CHANGE-BUFFER-CONTENT COMMAND**
With this command, the user can display and change single words in the buffer. Ths syntax is :

displacement in buffer/ old value new value<CR>

Ending the look-at by typing the @.

**THE EXECUTE-IOX COMMAND**
This command gives a subset of commands that executes single IOX instructions on the interface.

**THE Z80-OPCOM COMMAND**
This command is a look-at function in the Z80 memory on the interface card.

**THE ACTIVATE-WITH-CCB-CONTENT COMMAND**
This command will ask fora controller word:

**Example:**

```
SCOPE>Activate <CR>

Controller word (oct): 20402 <CR>

Type "space" to repeat, "S" to stoptest or "L" to loop.
```

This command will take the CCB content specified with the command CHANGE-CCB-CONTENT and activate the interface with it. If loop mode is ON, the execution will repeat. When errors are detected a warning will be printed with the error code.

After the interface has finished the operation, the console device is scanned for input (loop mode ON) or waited for input from. The letter "S" will stop the sub-command and return to normal SCOPE-LOOP. If loop mode is not off, the letter "L" will start looping and a space will repeat the command ONCE. This means that ESC is not needed to break a loop.

**THE CLEAR-DEVICE COMMAND**
This command executes a clear device IOX and waits until the device is ready.

**THE FILL-BUFFER COMMAND**
This command fills the buffer with a specified test pattern.

**THE SET-DEVICE-ADDRESS-ADDER COMMAND**
This command sets up limit and step for the READ and WRITE commands when the loop mode is ON.

**THE ALIGNMENT COMMAND**
This starts to read one sector from a specified address on a floppy and ignores all errors.

**THE JUMP-BETWEEN-ADDRESSES COMMAND**
This command starts reading between two specified addresses on a floppy.

**THE SET-PARAMETERS AND INFO COMMANDS**
See specification of these commands.

## REWIND

This command will rewind the cartridge installed to Beginning-Of-Tape (BOT). It is only executable if STREAMER if selected.

## RETENTION

This command will wind the tape to End-Of-Tape (EOT) and back to BOT If LOOP-MODE is ON, it will repeat until ESC is pressed. It is useful for polishing a new tape. It is only executable if STREAMER if selected.

## TEST-CARTRIDGE-CAPASITY

This commmand uses the function on the interface with the same name. It will write on the tape until End-Of-Media (EOM) is detected. The number of blocks written is reported back and printed out as number of blocks, pages and megabytes (1 Mb = 1000 * 1000). This is useful when you need to ascertain whether or not a tape will be able to store a specified amount of data. It is only executable if STREAMER if selected.

## FORMAT

This command will format a specified number of tracks. If address zero and all tracks are selected, the function to format the whole floppy is executed. It is only executable if FLOPPY if selected.

**Example:**

```
TPE>format <CR>

Format number... (0-17 Oct.): 0 <CR>

Physical address on floppy
  Track  (0-76 Dec.): 0 <CR>
Track amount  (1-77 Dec.): 77 <CR>
```

## COPY-FLOPPY

This command will copy two floppies on the same controller with the same format by using the function COPY FLOPPY on the interface. It is only executable if FLOPPY if selected.

**Example:**

```
TPE>copy <CR>

From unit (0-3 Dec.): 0 <CR>
To   unit (0-3 Dec.): 0 1 <CR>
```

## CHECK-FLOPPY

This is a function in the interface that reads data to the controller's memory and test for CRC errors. The command is only executable on a floppy. A specified number of sectors from a physical address is checked. If a CRC error is reported, the logical and physical address to this is printed out, and a search of the rest of the specified area is set up.

**Example:**

```
TPE>check-floppy <CR>


Physical address on floppy
  Track  (0-76 Dec.): 0 <CR>
  Sector  (1-8 Dec.): 1 <CR>
Sector amount (1-616 Dec.): 616 <CR>
```

## SET-PARAMETERS

This command allows the user to change parameters used in all commands and to change the execution parameters for the FUNCTION command. It is also possible to change the setting of bits in the controller-word written to the controller (compare/ step rate...etc.).

Loop mode allows the setting of a flag so that some commands will repeat execution endlessly if not aborted by the max-error count. The prompts given differ according to what controller part is used. See examples of different prompts.

**Example:**

```
TPE>set-parameters <CR>
Loop mode ......................(Yes/No): Yes <CR>
Short error message ........... (Yes/No): Yes <CR>
Maximum device errors before abort (dec): 8 <CR>
Test pattern............................: Write-and-read-critical <cr>
Compare mode ...................(Yes/No): Yes <CR>
Disable precomp................ (Yes/No): No <CR>
Door lock...................... (Yes/No): Yes <CR>
Slow step rate................. (Yes/No): Yes <CR>
Format number.............. (0-17 Oct.): 0 <CR>
Test Track................. (0-76 Dec.): 38 <CR>
Number of loops for media tests .. (dec): 50 <CR>

TPE>
```

## INFO

This command gives a subset of commands :

- LIST-FORMATS
- LIST-ERROR-CODES
- LIST-FUNCTIONS
- READ-FORMAT
- READ-EXTENDED-STATUS
- TRANSLATE-ADDRESSES

They enable the user to list the text associated with a format number, an error code or a function with the sub-commands LIST-***.

The sub-command READ-FORMAT will execute the function read-format on the interface and the format found on the floppy will be saved for later use (as defalut).

When an error occurs and only a short error message is printed out, the sub-command READ-EXTENTED-STATUS can be used to read status from interface and decode it fully.

The sub-command TRANSLATE-ADDRESSES will calculate between physical, logical and page addresses for the current format.

**Example:**

```
TPE>info <CR>

INFO> read-extended-status <CR>


Controller status
  Status 1                  : 000010
    Device Ready For Transfer
    Transfer OK
  Status 2                  : 000000
    Selected unit           : 0
  Last memory address       :    60650 (Oct.)
  Remaining words           :        0 (Oct.)
Z80 memory status
  PROM status
    Card number             : 3112
    Micro program version   : B
    Micro program sub ver.: 29 NB!! PRE-release
  RAM status
    Buffer start            : 2200H
    Buffer size             : 1000H
Z80 CPU status
  Z80 AF-reg                : 0000H
  Z80 BC-reg                : 0000H
  Z80 DE-reg                : 0000H
  Z80 HL-reg                : 0000H
  Z80 PC-reg                : 0000H
  Z80 SP-reg                : 0000H
  Z80 IX-reg                : 0000H
  Z80 IY-reg                : 0000H

INFO> translate-aaddress <CR>

ADDRESS-TYPE: ?
CORRECT ANSWERS ARE:
PHYSICAL
LOGICAL
PAGE
ADDRESS-TYPE: physical <CR>

Physical address on floppy
  Track  (0-76 Dec.): 0 <CR>
  Sector  (1-8 Dec.): 1 <CR>

-------------------------------------------------------
!      ! Side ! Track ! Sector ! Log.addr. ! Page     !
-------------------------------------------------------
! Dec. !    0 !     0 !      1 !         0 !        0 !
! Oct. !    0 !     0 !      1 !         0 !        0 !
-------------------------------------------------------
INFO> EXIT
```

# Chapter 7 GRAPHIC-TERM-BOO

# CHAPTER 7 GRAPHIC-TERM-B00

This chapter describes the GRAPHIC-TERM test program.

Version A00 of this program has been released before as a separate product.

## 7.1 GENERAL

This product is made to test and verify the TDV 2200S terminals equipped with the graphic option. Each command checks one or more of the graphic functions. When a command is typed, the program responds with a short command description and waits until the <CR> button is pressed. After the button is pressed, the specified test is carried out.

The program waits for a new <CR> before the screen is erased and a new command can be given. In the command "RUN-ALL-TESTS" there is no prompting for <CR>. Instead, the program will wait for 5 seconds (default) before the next test/description is displayed. It is possible to change this delay in SET-PARAMETERS, and other RUN facilities.

**Terminal types:** This version supports the following terminals:

ND-324 (Notis) and ND-325 (Net)
ND-246, ND-285, ND-320 and ND-322 w/graphic option

## 7.2 STARTING AND INITIAL ACTION

GRAPHIC-TERM is started from the TPE-MONITOR by the command:

```
TPE>Load GRAPH-TERM <CR>
```

## 7.3 Program Commands

### SELECT-DEVICE

Select the terminal to be tested by specifying the terminal's logical device number.

### SET-PARAMETERS

Set different modes and change program parameters such as delay between tests, loop mode, how many times, abort mode, abort after how many errors and full or normal error printout selection.

### RUN-ALL-TESTS

Execute all commands in sequence. There is no prompting for <CR>. Instead, a delay (default 5 seconds) is inserted between each description/test. If loop-mode is set, RUN-ALL-TESTS will be repeated until ESC is pressed or max-loops is reached.

### CLEAR-SCREEN

Clearing screen, both graphics and alphanumerics. The graphic memory is cleared, the terminal memory is cleared, the writing mode is set to dot on, the line type is set to solid, character size is set to 1x.

### CLEAR-GRAPHIC

Erases the graphic display without affecting the terminal in TDMODE.

### GRAPHIC-VIDEO

Disable/Enable graphic video plane.

## MOVE-CURSOR

Moves the cursor in different directions. Testing space right, space down, space left, space up and carriage return.

## RECTANGLE-FILL

Fill screen by rectangle filling in VMODE. Direct scaling is disabled (Tektronix 4010 scaling), and fill pattern set number zero (default) is defined. Two coordinates equal to the screen are defined in VMODE, before the rectangle fill function is carried out.

## CHESS-BOARD

Draw chess-board pattern on whole screen. Different coordinates are defined in VMODE to check rectangle filling. Rectangle fill is carried out in current rectangle-fill modes (see RECTANGLE-FILL).

## COORDINATES

Draw lines in horizontal direction. Direct scaling is disabled (Tektronix 4010 scaling). Lines are drawn from left to right to check coordinate adressability in VMODE.

## LINES

Draw lines of different styles to check the set-line-type function. First line is self-defined as 15 dots,5 space,1 dot, 5 space. The rest of the lines are standards, long dashed, short dashed, dot dashed, dotted and solid.

## FONTS

Write text with down-loaded fonts, fonts are read from a font file "GRAPHIC-FONTS:FONT". Direct scaling is enabled (Tandberg scaling). Font 7 is selected and down-load is initialized. Texts are written in normal and italic character types, in upper and lower case.

## FLASHING-CROSSHAIR

The crosshair cursor is positioned in the center of the screen and flashing crosshair is enabled.

## PLOT-POINTS-LINES

Uses PPMODE and IPMODE to plot points and lines in different directions.

## GRAPHIC-MEMORY

Graphic memory transfer test. A sequence of bits is transferred to graphic memory, and read back into the host. These bits are transferred as a compressed bitmap. The Block Transfer Load address set, Compressed bit map data transfer to/from graphic functions are tested.

## POLYGONS

Draw polygons with different fill patterns. Tektronix scaling is entered, and fill-pattern one, two and three, are selected for the polygons. The printout tests polygon definition mode.

## COPY-WINDOW

Copies data between two windows on the screen. Tektronix scaling is entered, and fill-pattern two is selected. A filled polygon is drawn on the screen and two windows are defined. Data is copied between these windows. Fill pattern selection, define window and copy function are tested.

## CIRCLES

Draws circles with different fill-patterns. Tektronix scaling is entered, and fill-patterns from 0-10 are selected. Circles are defined in draw-circle function.

## ARCS

Draws arcs with different fill patterns. Tektronix scaling is entered, and fill-patterns from 0-10 are selected. Clipped arcs are drawn at different angles

## INKING

Enables "INKING" in graphic input mode. Vectors connect each point where the cursor moves.

## GRAPHIC-HARDCOPY

This command will execute a graphic memory dump to an Epson MX100. It can drive an Epson MX80 / FX80 or RX80, but only with rotated plot.

# CHAPTER 8 HDLC TEST PROGRAM

# CHAPTER 8 HDLC TEST PROGRAM

## 8.1 GENERAL

This program is a verification and debugging program for (DMA) HDLC and Megalink interfaces. The program is a stand-alone version and contains three separate tests and a number of commands to specify parameters when changing default values.

The RUN command includes the basic test.

The LOOP-TEST is the main test, which may be run in different ways depending on the SPECIFY-TEST-CONFIGURATION.

## 8.2 STARTING AND INITIAL ACTION

HDLC TEST PROGRAM is started from the TPE-MONITOR by the command:

```
TPE>Load HDLC <CR>
```

When the program starts, it checks how much memory is available (up to 512K words), then it checks if there are any disk controllers present. The disk controllers and additional HDLCs and Megalinks are used to produce DMA-load during testing. The device number is initially set to the first HDLC found (normally 1640). If no HDLC/MEGALINK is found, the program will indicate this.

## 8.3 TEST PROCEDURES

## NORMAL PROCEDURE TO TEST HDLC

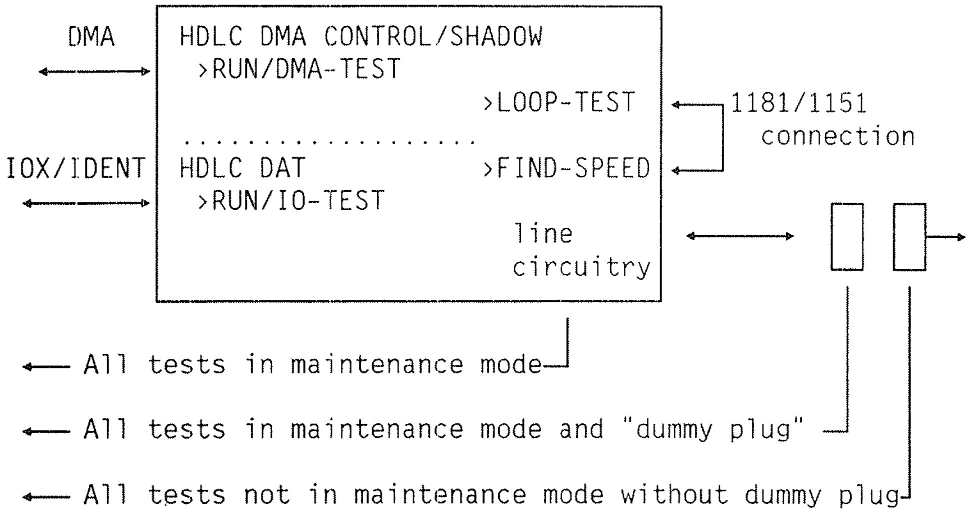
**Verification of an HDLC Installation:** The test listed below will in most cases be sufficient to verify the hardware in an HDLC/MEGALINK interface.

- TPE>SPECIFY-TEST-CONFIGURATION % How tests are to be run
- TPE>RUN % Basic tests
- TPE>FIND-SPEED % Finds the transmission

  speed
- TPE>LOOP-TEST % Run data in specified mode

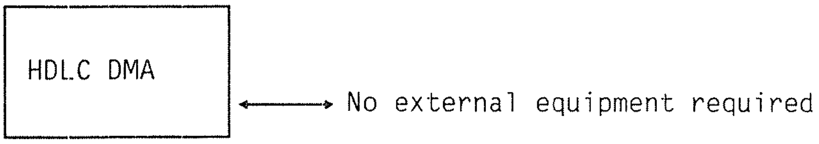
All tests should be run in maintenance mode first.

## HDLC-MEGALINK Test Area Illustration


DMA
HDLC DMA CONTROL/SHADOW
>RUN/DMA-TEST
>LOOP-TEST
1181/1151 connection
IOX/IDENT
HDLC DAT
>FIND-SPEED
>RUN/IO-TEST
line circuitry
All tests in maintenance mode
All tests in maintenance mode and "dummy plug"
All tests not in maintenance mode without dummy plug

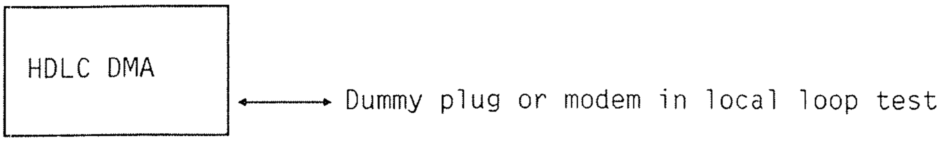
Data and clocks must be returned from either
- a modem in local loop test or
- a link (modem or computer link) connected to a remote CPU with HDLC echoing the signals.

## HDLC-MEGALINK Test Setups



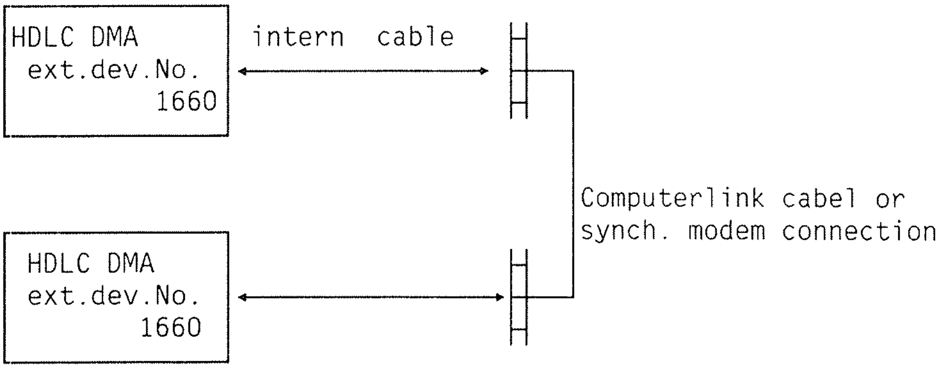
```
TPE>SPECIFY-TEST-CONFIGURATION<CR>
HDLC DEVICE NO. (OCT.): 1640<CR>
```

```
MAINTENANCE MODE (YES/NO): YES<CR>

TPE>LOOP-TEST<CR>
```



```
TPE>SPECIFY-TEST-CONFIGURATION<CR>
HDLC DEVICE NO. (OCT.): 1640<CR>
MAINTENANCE MODE (YES/NO): NO<CR>
ECHO FROM HDLC IN THE SAME MACHINE (YES/NO): NO<CR>

>LOOP-TEST<CR>
```

## Two HDLC-MEGALINK in the Same Machine



```
TPE>SPECIFY-TEST-CONFIGURATION<CR>
HDLC DEVICE NO. (OCT.): 1640<CR>
MAINTENANCE MODE (YES/NO): NO<CR>
ECHO FROM HDLC IN THE SAME MAHINE (YES/NO): YES<CR>
ECHO DEVICE NO. (OCT).: 1660<CR>
```

## 8.4 Program Commands

### CHANGE-TEST-MODE

This command controls the test sequence and error printout mode. The example below should be self-explanatory

Some of the questions will be omitted depending on the answer given in the previous question. For example, if the answer to SUPPRESS ERROR PRINT was NO, the question RING

BELL ON ERROR is not asked.

**Example:**

```
TPE>CHANGE-TEST-MODE<CR>
LOOP IN TEST (YES/NO): YES<CR>
HOW MANY TIMES (DEC.): 0<CR>  (0=forever)
SUPPRESS ERROR PRINTOUT (YES/NO): NO <CR>
SHORT ERROR PRINT (YES/NO): YES <CR>
RING BELL ON ERROR (YES/NO):YES <CR>

TPE>
```

## DEFINE-HDLC-PARAMETERS

This command makes it possible to change parameters used in other tests:
<FRAME SIZE>
<DATA-PATTERN>
<PCR-REGISTER>
<SYNC/ADDRESS-REGISTER>
<CHARACTER LENGTH-REGISTER>
<DMA BLOCK SIZE>
<DISPLACEMENT 1>
<DISPLACEMENT 2>

**<FRAME SIZE>**
You can change the size of the frames to be transmitted during the LOOP-TEST by this parameter. Frame number 1 is generated with the first frame-size specified, frame number 2 with the second and so on. If only one fixed frame size is required, only one is specified.

It is possible to specify up to 17 different frame sizes from 2 to 1024 bytes. You must give the frame sizes on the same line, separated by commas or spaces. If you only press CR (Carriage Return), the frame size will be unchanged.
Initial values: 70,2,2,300,5,13,177,101,3,101,122,2,3

**<DATA-PATTERN>**
Frames may have different data contents specified by this parameter. It is possible to specify 7 patterns. Either repeating patterns (maximum 7 times) or 7 different patterns.

You must specify the parameters on the same line. If you only press CR (Carriage Return), the patterns remain unchanged. The different patterns are:

- ONLY-ZEROS (1)
- BYTE-377b (2)
- BYTE-252b (3)

- INCREASING (4)
- DECREASING (5)
- BYTE-177b (6)
- BYTE-176b (7)
- *UNCHANGED (8)

*Initial patterns are: 1-7

**<DMA-BLOCK-SIZE>**
This parameter specifies the DMA block size (1-2000 oct.), and the displacement (0-block size) used by the LOOP-TEST.

Initial block size is: 200b.

**<DISPLACEMENT 1>**
This is the number of free bytes reserved at the beginning of blocks containing "START OF MESSAGE" frame.
Initial value: 0.

**<DISPLACEMENT 2>**
This is the number of free bytes in the beginning of blocks NOT containing "START OF MESSAGE" frame.
Initial value: 0.

## FIND-SPEED

The command finds the rate for the transmission clock. The test transmits bytes for one second and computes the number of bytes transmitted.

If the interface is operating in maintenance mode (internal looping), the data rate is equal to the internal clock rate. If the interface is not in maintenance mode, the data rate is equal to the modem clock rate or to the internal clock rate if a CPU-CPU cable is connected to the interface.

**Example:**

```
TPE>FIND-SPEED<CR>

CLOCK-RATE : 76.81 KBITS/S

TPE>
```

## INCREASE-DMA-LOAD

Using this command makes it possible to increase the load on the DMA. The program searches for other HDLC's and for one disk controller, and it asks whether or not they should used to produce DMA load. The devices to be used will run in maintenance mode.

This is not executable under SINTRAN.

**Example:**

```
TPE>INCREASE-DMA<CR>

RUN DISC DEV.NO 1540 IN MAINTENANCE MODE (YES/NO): YES<CR>

TPE>
```

## LOOP-TEST

During the loop test, frames with different sizes and different data patterns are transmitted. Received frames (if any) are checked for errors. How the frames are looped depends on the setting of the SPECIFY-TEST-CONFIGURATION command.

If maintenance mode is not used, looping must be provided by:

- cable looping
- modem looping
- external echo device (HDLC-interface in another computer)

All frames transmitted have the following format:

FLAG, ADDRS-BYTE, SEQUENCE-BYTE, DATA BYTES, FCS1, FCS2, FLAG

where

```
ADDRS-BYTE    = Constant value, initially zero, changed by
                SET-PROTOCOL command.
SEQUENCE-BYTE = Frame sequence counter from 0-255.
DATA          = N data bytes (0 <= N <= 1022).
FCS1 FCS2     = Framecheck sequence (CRC).
```

The LOOP-TEST will continue infinitely if nothing else is specified. It can be stopped by pressing the ESCAPE key on the keyboard. Every time a key is depressed on the terminal, a status report is written. Afterwards the test continues.

Example of a status report:

```
             FRAMES O/U-RUN CRC-ERR DATA-ERR ABORT ASTAT LSTAT
HDLC OUT:     1890      0       0        0     0  7404  5404
HDLC IN :     1890      0       0        0     0  3750  3750
```

The different columns signify:
FRAMES: = Number of frames transmitted (generator out) or received (generator in).
O/U-RUN = Number of over/under-runs detected.
CRC-ERR = Number of CRC-errors detected, i.e. CRC of received was correct, but the data was still corrupted.
ABORT = Number of aborts detected.
ASTAT = Accumulated status bits so far (receiver/ transmitter transfer status register).
LSTAT = Last status read from receiver/transmitter transfer-status register.

## REDUCE-DMA-LOAD

The command turns off the HDLC's and the disk running in maintenance mode, started by the command INCREASE-DMA-LOAD.

Not executable under SINTRAN.

**Example:**

```
TPE>REDUCE-DMA<CR>

STOP DISC DEV.NO: 1540 (YES/NO): YES<CR>

STOP HDLC DEV.NO: 1660 (YES/NO): YES<CR>

TPE>
```

## RUN

This command runs the basic tests, either each of them or both in sequence. The test are IO-TEST, DMA-TEST. DEFAULT is BOTH-IO-DMA.

**Example:**

```
TPE>RUN<CR>
I/O-DMA OR BOTH: <CR>
BOTH-IO-DMA

TESTS ARE LOOPING
```

**DMA-TEST**

This is a sub-test under the RUN command.
The test checks the DMA channel of the interface to the computer memory. During the first part of the test, the "INIT" function is used with all possible variations of the parameter values (PCR, SAR,CHLR, D1, D2, MAXBL). The result is read back with the "DUMP" function and tested for errors.

During the second part of the test, the interface memory (register 40-377 is loaded from the computer memory ("LOAD") and read back("DUMP"). The "LOAD" address starts at the lower end of physical memory and is moved 340 octal words upwards for each transfer. The "DUMP" address starts at the upper end of the memory and is moved downwards. When all of memory has been tested, the test repeats itself with a different data pattern.

The data patterns used are:

- All zeros
- "Address in address" (16 bit)
- 125252
- 077176
- 177777

If any difference is found between the data written ("LOAD") to the interface, and the data read back ("DUMP"), the data is read back once more to a fixed buffer in lower memory. Error printouts take the following form:

```
xx DMA ERROR FROM yyy/ddd to zzz/ppp (qqq)
```

where

```
xx  = word index in block (0 <= xx < 377)
yyy = address from which a word was
      transferred TO the interface
ddd = contents of address yyy
zzz = address to which data was transferred
      FROM the interface
ppp = contents of address zzz
qqq = contents of word number xx in fixed buffer
      after extra "DUMP"
```

**IO-TEST**

This is a sub-test under the RUN command.
It is an elementary test to check for IOX errors. It checks READY-FOR-SENDING and DATA-SET-READY. The test also checks ident codes and interrupt levels of the interface.

## SET-MEMORY-LIMITS

Initially the program uses all of the memory (except the part occupied by itself) for the DMA transfers. If for some reason only part of the memory is used, you should give this command.

The parameters are given as page numbers and the default values are the initial values.

**Example:**

```
TPE>SET-MEMORY-LIMITS<CR>
LOWEST PAGE (12-128 Dec.):20<CR>
HIGHEST PAGE (20-128 DEC.): 64<CR>

TPE>
```

## SET-TEST-CONFIGURATION

This command specifies how the tests are run

LOOP-TEST,
RUN or
FIND-SPEED

The command sets up the default device number and specifies whether tests should run in maintenance or not. If maintenance mode is not used, and if there are other HDLC's in the system, it will ask if one of them should run in echo mode.

The command is not executable under SINTRAN.

**Example:**

```
TPE>SPECIFY-TEST-CONFIGURATION<CR>
HDLC DEVICE NO. (OCT.): 1640<CR>
MAINTENANCE MODE (YES/NO): NO<CR>
ECHO FROM HDLC IN THE SAME MACHINE (YES/NO): YES<CR>
ECHO DEVICE NO. (OCT.): 1660<CR>

TPE>
```

# Chapter 9 INSTRUCTION-VERIFY-COO

# CHAPTER 9 INSTRUCTION-VERIFY-COO

This chapter describes the INSTRUCTION-VERIFY test program.

## 9.1 GENERAL

This is a verification program for the ND-100, ND-100/CE, ND-100/CX and the ND-110 instruction set. It runs as a stand-alone test program under control of the TPE-monitor. None of the tests are executable under SINTRAN.

## 9.2 STARTING AND INITIAL ACTION

INSTRUCTION-VERIFY is started from the TPE-MONITOR by the command:

```
TPE>Load INSTR <CR>
```

When the program is loaded , it prints information about the CPU.

**Example:**

```
TPE>LOAD INSTR <CR>

    INSTRUCTION - Prerelease: COO - 1986-08-27

CPU type..........: ND-100/CX upgraded for 16 PITS
Floating format...: 48 bits
Memory Management.: MMS-2
Cache.............: Yes
ALD register......: 1560B
Cpu cycle.........: Fast

TPE>
```

**NOTE:** A cache which is turned off, is reported as not present.

## 9.3 Program Commands

### RUN

The complete instruction set for the current CPU type is automatically selected and tested. The connection between CPU type and instruction groups selected is shown in the table below.

| | ND-100 | | | | | | | | ND-110/CX<br>ND-110 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | CE | CE | CX | CX | CX<br>no<br>MMS | CX<br>no<br>MMS | | |
| | 48 | 32 | 48 | 32 | 48 | 32 | 48 | 32 | 48 | 32 |
| ARGUMENT | X | X | X | X | X | X | X | X | X | X |
| MEMORY-REFERENCE | X | X | X | X | X | X | X | X | X | X |
| SEQUENCE | X | X | X | X | X | X | X | X | X | X |
| REGISTER-OPERATIONS | X | X | X | X | X | X | X | X | X | X |
| BIT-OPERATIONS | X | X | X | X | X | X | X | X | X | X |
| SHIFT-INSTRUCTIONS | X | X | X | X | X | X | X | X | X | X |
| 32-BITS-FLOATING | | X | | X | | X | | X | | X |
| 48-BITS-FLOATING | X | | X | | X | | X | | X | |
| PRIVILEGED | X | X | X | X | X | X | X | X | X | X |
| BYTE-STRING | X | X | X | X | X | X | X | X | X | X |
| ND100-24BIT | X | X | X | X | X | X | X | X | X | X |
| BCD | | | X | X | X | X | X | X | X | X |
| ND100-CX | | | | | X | X | | | X | X |
| STACK | | | | | X | X | X | X | X | X |
| SEGMENT | | | | | X | X | | | X | X |

NOTE: Tests for the new ND-110 instructions for the SINTRAN III K-version have not yet been implemented.

**Example:**

```
TPE>run CR>

=== RUNNING TESTS ON LEVEL 1  ===
=== ARGUMENT INSTRUCTIONS                  === END OF TEST ===
=== MEMORY REFERENCE INSTRUCTIONS          === END OF TEST ===
=== SEQUENCING INSTRUCTIONS                === END OF TEST ===
=== REGISTER INSTRUCTIONS                  === END OF TEST ===
=== BIT INSTRUCTIONS                       === END OF TEST ===
=== SHIFT INSTRUCTIONS                     === END OF TEST ===
=== 48 BITS FLOATING INSTRUCTIONS          === END OF TEST ===
=== PRIVILEGED INSTRUCTIONS                === END OF TEST ===
=== BYTE INSTRUCTIONS                      === END OF TEST ===
=== PHYSICAL MEMORY INSTRUCTIONS           === END OF TEST ===
=== BINARY CODED DECIMAL INSTRUCTIONS      === END OF TEST ===
=== CX INSTRUCTIONS                        === END OF TEST ===
=== STACK INSTRUCTIONS                     === END OF TEST ===
=== SEGMENT INSTRUCTIONS                   === END OF TEST ===
=== RUNNING TESTS ON LEVEL 2  ===
                 :
                 :
=== RUNNING TESTS ON LEVEL 9  ===
=== END OF TEST ===

TPE>
```

If you want a list of the instructions for each group, use the command

```
     EXPLAIN-COMMAND <instruction group>,
```

or see the ND-100 Reference Manual (ND-06.014).

**NOTE:** By giving the name of an instruction group, it is possible to run tests belonging to that instruction group only. The program will not check that the given instruction group is legal for the current CPU, so care must be taken.

## SET-PARAMETERS

Allows you to set different modes affecting the test execution as follows:

**LOOPMODE**
Tests will loop a given number of times.
Initial value: Infinite.

**ABORTMODE**
Tests are aborted after a given number of errors.
Initial value: Infinite.

**EXTENDED ERROR PRINTOUT**
Register and memory content relevant for the failing instruction will be printed.
Initially: Not extended.

**MICROPROGRAM STRESS**
It is possible to start some activities on the interrupt levels 10 - 14 to maximize the microprogram stress. These activities are:

Level 10: Dummy output (ASCII 0) to the console.
Level 11: Memory read/write.
Level 12: Forced IOX-error.
Level 14: Serving the forced IOX-error.

The RT-clock always runs on level 13.

Available stress modes are:

No : No stress activity started.

Increasing: Each time the whole test section is completed, one stress activity is started, until all four stress activities are running.

Full : All stress activities are started.

Initially no stress activities.

**LIST INSTRUCTIONS BEFORE TEST**
If the CPU hangs during a test section, it will be useful to set this flag. The last instruction listed, is the one which made the CPU hang.

The default values are equal to the initial setting.

**Example:**

```
TPE>set-parameters <CR>

Loop mode                              (YES or NO): Yes <CR>
How many times                               (dec): Infinite <CR>
Abort mode                             (YES or NO): No <CR>
Extended error printout                (YES or NO): Yes <CR>
Microprogram stress                               : Full <CR>
List each instruction before test (YES or NO): No <CR>

TPE>
```

## 9.4 Test Description

### Test of Instructions

The specified instructions are executed in sequence on level 1 - 9 (dec.). An instruction is tested as follows:

- The registers on the actual level and the memory cells involved are initialized.
- The actual level is activated by setting the corresponding bit in the PIE and the PID register. The instruction is followed by a wait instruction.
- Involved registers and memory locations are checked for correct values. A check is also made to ensure that registers not involved in the operation are unchanged.
- If there is any mismatch, error messages are given.

## Test of Internal Interrupts

These internal interrupts are provoked:

Monitor call
Protect violation
Page fault
Illegal instruction
Error indicator on
Privileged instruction
IOX error

A check is made to ensure that the conditions above gave interrupt and that the IIC (internal interrupt code) register contains the correct value.

## 9.5 Error Messages

Error messages during the instruction test have the following format:

```
*** ERROR ***  Operation was: XXX  Time: 1986.09.09 08:47:05
```

If "extended error printout" is specified by the SET-PARAMETERS command, the register content and the relevant memory content will also be printed.

Example with extended error printout:

```
*** ERROR ***  Operation was: NLZ  Time: 1986.09.09 08:47:39

          -----------------------------------------------------
          !                Register content                   !
          -----------------------------------------------------
          !   Reg       !  Before    !  After     !  Expected !
          -----------------------------------------------------
          !   P         !  116443    !  116445    !  116445   !
          !   X         !  163777    !  163777    !           !
          !   T         !       0    !   40001    !           !
          !   A         !       1    !  100000    !   40100   !
          !   D         !  177777    !       0    !       0   !
          !   L         !  177777    !  177777    !           !
          !   B         !  116340    !  116340    !           !
          -----------------------------------------------------
```

Example of error messages during the internal interrupt test:

```
*** ERROR ***  Operation was:
INTERNAL INTERRUPT TEST Time: 1986.09.09 12:53:23
Wrong IIC register
Expected IIC value:       5
Found IIC value:          4
Interrupt provoked by:    Illegal instruction (code 143700)
```

In some cases of errors in the instruction set, the program may not be able to give error messages. Example of such error conditions are:

- Memory is overwritten and the program is destroyed.
- The next instruction is never fetched.
- The CPU makes a random jump in the program execution.

In these cases the program will probably be "hanging".

# CHAPTER 10 LINE PRINTER TEST PROGRAM

# CHAPTER 10 LINE PRINTER TEST PROGRAM

This program has various tests for line-printers. Most of the tests are visual.

## 10.1 GENERAL

The program sends test patterns or control characters that must be checked by the user.

All commands use the device-number/file-name that is specified when entering the program. To change this device-number/file-name, you must give the monitor command

```
SET-PRINTER-DEVICE-NUMBER
```

You can use the following commands in this program:

```
COLUMN-TEST
FORMAT-TEST
INTERFACE-TEST
SET-PARAMETERS
RUN-TEST-PATTERNS
STRESS-BUSY-TEST
```

All tests are stopped by pressing the ESC button on the keyboard.

## 10.2 STARTING AND INITIAL ACTION

LINE PRINTER TEST PROGRAM is started from the TPE-MONITOR by giving the command:

```
TPE>Load LINE-PRINT <CR>
```

When the first test is started,the program asks for the printer's device number. The number is checked to see if it is a legal dev. number for a printer, and that it exists. If not it will ask again. When running under SINTRAN, the program asks for the periferal file name for the printer.

**NOTE:** Before running any test you must give the command:

```
@STOP-SPOOLING <CR>
```

Otherwise no output will occur before you exit from the test program.

## 10.3 Program Commands

This section gives an explanation of the commands used in this test program.

## COLUMN-TEST

**Parameters: <characters> <column from,(to)>**
This command prints the specified character in one or more specified columns.

The command first asks you for a character. The default character is "E", but you can specify any character.

Then the command asks for column (from,<to>). If you specify two numbers, separated by a comma, the program will print the specified characters between these two columns. The first number you specify has to be smaller than the last. If you specify only one number, the character will only be printed in that colomn. The default value is all columns.

**Example:**

```
TPE>Column-test <CR>

CHARACTER: X <CR>
COLUMN (FROM,<TO>) (1-132 DEC.): 40,50 <CR>
```

## FORMAT-TEST

**Parameter: <channel>**
On some printers it is possible to make the line printer skip to specified lines on the page by giving different control characters. These positions are referred to as channels. In addition, it is possible to carry out a form-feed (FF), carriage-return (CR) and line-feed (LF). This command tests the form and channel-feed on printers with such an option.

First a form-feed is done, and the sentence

"*****THIS IS TOP OF FORM"

is written. Then it skips to the specified channel and prints

"*****THIS IS CHANNEL NN"

The command will then ask for a new channel until you answer EXIT.

**Example:**

```
TPE>Format <CR>
FORMAT-TEST
CHANNEL: 5 <cr>
CHANNEL-5
CHANNEL: Exit <CR>
EXIT
TPE>
```

COMMENT

The printer does a form-feed and skips to channel 5.

# INTERFACE-TEST

**Parameter: <interface type>**
This command cannot be run under SINTRAN.

You can choose one of the following interface types:

1) CDC-LP-INTERF.-1047
2) TERMINAL-BUFFER-1095
3) PARALLEL-BYTE-1109
4) PARALLEL-1130-1140
5) DUAL-ASYNC-MODEM-1147
6) FLOPPY-4TERM-3010
7) EIGHT-TERMINAL-3013

This program sends all illegal characters to the specified interface and checks that the interface detects them.

Then it checks the data-way by writing all legal characters to the interface and reading them back again.

The command also checks the ident code for the device number.

**Example:**

```
TPE>Interface-test <CR>
INTERFACE TEST
INTERFACE: CDC <CR>
CDC-LP-INTERF.-1047

ILLEGAL CHARACTER TESTED.
INTERFACE TESTED WITH ALL CHAR.
TEST ARE LOOPING.
```

## SET-PARAMETERS

**Parameters: <line length> <small letters> <skip pattern> <loop mode>**

This command sets/resets different parameters used by the other commands. The MODE-command PROGRAM-STATUS will present these parameters.

- LINE LENGTH (1-136 DEC.):
  Sets the maximum line-length for the printer. Default is 132.

- LINES PR. PAGE (DEC):
  Sets the number of lines between each form-feed. Default is no check.

- SMALL LETTERS (YES/NO):
  Sets/resets if the patterns in the RUN-TEST-PATTERN should include small letters (96 characters). The default value is NO.

- SKIP Pattern (1-5 DEC.):
  It is possible to skip some (or all) patterns used by the command RUN-TEST-PATTERN. Answer with the numbers (1-5) of the patterns you want to skip on one line, separated by a comma. or a space. Terminate with CR. The default value is none skipped.

- LOOPMODE (YES/NO)
  If you answer YES, the program will ask

  ```
  HOW MANY TIMES (DEC.):
  ```

  Here you can specify how many times you want the tests to loop.

## RUN-TEST-PATTERNS

This command runs 5 different test-patterns on the Line-printer. Each pattern starts with a form-feed and the pattern number.

- TEST-PATTERN 1:
  All characters are printed, then shifted one position to the left, until all characters are printed in all positions.

- TEST-PATTERN 2:
  On the first line the maximum number of characters will be printed, and on the second line just the first????? character is printed. The patterns will then alternate between decreasing the maximum line length and increasing the minimum line length with one character for each line printed.

- TEST-PATTERN 3:
  All characters are printed, filling up one line each.

- TEST-PATTERN 4:
  The letter "M", separated by spaces, is printed in every ODD column (1,3,5 etc.) up to max line length.

- TEST-PATTERN 5:
  The letter "M", separated by spaces, is printed in every EVEN column (2,4,6 etc.) up to max line length.

## STRESS-BUSY-TEST

Missing characters on line printers could be due to the printer sending the busy signal too late, and thus not being able to accept characters already sent from the CPU. The command tries to make the printer busy again by having it print the numbers from 0-9, one on each line, and then carry out a form-feed.

**Example:**

```
TPE>Stress-busy-test <CR>

STRESS-BUSY-TEST
0
1
2
3
4
5
6
7
8
9
```

# PROGRAM-STATUS

This Monitor command presents variables from the Monitor, and also some variables set by the SET-PARAMETERS command

**Example:**

```
TPE>Program-status <CR>

TPE PROGRAM STATUS: 1986.12.04 09:56:15
TPE version..................: A02 - 1986-10-30
Console device...............: 39D / 47B
Printer device/mode..........: OD / OB / Normal
Stop on full page............: off
Test Program file name.......: LP-TEST-E00

Line length..................:    132
Character set................:     64 - Capitals only
Patterns used................:        1      2      3      4       5

TPE>
```

# Chapter 11 MAGTAPE

# CHAPTER 11 MAGTAPE

This program tests Pertec & Cipher Magtape controllers (ND-557), and also the different functions on the magtape drives.

## 11.1 GENERAL

The test program is a stand-alone program. Stand-alone means you must run it without SINTRAN operative. The program runs under the TPE monitor.

When running the program, the program will report the magtape controller (Cipher/Pertec), with ECO level if the controller is an ordinary Pertec.

**NOTE:** The ECO level is just given for your information. It is NOT a detected ERROR.

If the controller is a Pertec/Cipher controller, two tables may be written out instead:

**The Extended Error Drive Status**

The first table is called Extended Error Drive Status, and is only written out if status error occurs. The table contains information about cache, retries, track in error, density etc.

**The Drive Status**

The second table is called Drive Status and is written out at the end of the test. It contains information about read/write retries, error counts in every track, reel size, configuration etc.

The first program-part tests that the interface part of the magtape controller is working properly against ND-100. The second part tests functions on the magtape drive (via the formatter).

**NOTE:** If you must make hardware changes in the magtape system (switch setting, cables etc), you must restart the program by pressing the MCL button on the operator panel and type 20! on the keyboard.

If you must change memory or memory switch setting, you must reload the program.

## 11.2 Starting and Initial Action

The MAGTAPE test program is started from the TPE-MONITOR by the command:

```
TPE>LOAD MAGTAPE <CR>

MAGTAPE TEST PROGRAM - Version : x - yyyy-mm-dd


Magtape controller 1 is present

TPE>
```

The test program monitor writes out its prompt and is ready to accept commands. If two magtape controllers are present, you are asked to select which one to test.

**NOTE:** Before you continue with the MAGTAPE test program, you must do as follows:

- Install a scratch tape into the magtape drive. The tape must have the WRITE ENABLE ring present.
- Switch the magtape drive ON-line.

## 11.3 Program Commands

The available commands are:

- SET-PARAMETERS
- RUN

**Example:**

```
TPE>Set-parameters <CR>
Loop mode .....  (Yes or No): No <CR>
Density 3200 BPI (YES or NO): Yes <CR>

TPE>RUN <CR

==> Memory-address register
==> Control register
==> Modus register
==> Wordcounter
==> Reading data on DMA in testmode
==> Unit ready test
```

```
   - Cipher tape unit 0 ready -

*** WARNING : ECO level on the ND-100 controller is "N" or lower.
              Remote density select and extended error status will be
              ignored.
==> File-protect ring test
==> Load point test
==> Overflow in read test
==> Write consecutive test
==> Write, read, data  and status test
==> Write consec., read, read consec. and read treshold test
==> Backspace test
==> EOF test
==> Interrupt test

TPE>EXIT <CR>
```

## Test Description

This section gives a short description of the different tests.

Errors detected in Status and Mode registers are reported by a short error message.

**Interface tests:**

**Memory address register test for interface only**
Load Memory Address register (MAR), Read MAR, and compare this with data loaded.

**Control register test for interface only**
Test that status register is cleared after Device Clear. Write 1,2,0 and 100000 (octal) into Control Word register (CWR) and test that the corresponding bits in Status register are set.

**Mode register test for interface only**
Load Mode register with special patterns, read the Mode register, and compare this with data loaded.

**Word-counter test for interface only**
Simulate write-transfer to magtape with controller in Test Mode (bit 3 in CWR set). Test that MAR and Word Counter (WC) are correct after "transfer". Test that Status bit Device Active is reset within a given time limit. If not a timeout error message is printed.

**Reading data on DMA in test for interface only**
Simulated read-transfer from magtape. The data simulated is compared with data expected.

**Device function tests:**

**Unit ready test**
Checks that only one tape unit is online. If more than one tape unit is online, an error message is displayed. If no tape unit is online, the test sequence is aborted. It also reports what kind of magtape drive is online (Cipher or Pertec).

The ECO level check is carried out immediately after the Unit Ready Test.

**File-protect ring test**
Test that the Write-Enable Ring is present on the tape. This ring must be present to enable WRITING on a magtape.

**Load point test**
Rewind Magtape and check that it is standing on the load point (beginning of tape - BOT).

Write erase gap and test that the load-point status bit is reset, then rewind and test that load point is reached again.

**Overflow in read test**
Remote density is selected for this function. The word counter is set to one less than the number of words in a record being read. This makes it possible to detect overflow. Then the word counter is set to one more than the number of words in the record being read, which causes underflow in the read.

**Write consecutive test**
Test that the formatter is busy after consecutive write operations. Then a normal write is carried out and the formatter is reset.

**Write, read, data and status test**
Write a number of records, read them back and compare the data read with the data expected. All status errors and mode errors will be written out.

**Write consec., read, read consec. and read threshold test**
Write number of records consecutive, read normal and compare data read with data expected. Rewind, read consecutive and compare data read with data expected. Rewind, read with extra high signal amplification (THR2 bit 10 in mode register set). Compare data read with data expected.

### Backspace test

Test read of record and space reverse. Test writing of a record and space reverse. Test forward space and reverse read.

### EOF test

Search reverse for FM (File Mark = End Of File). Test that correct FM is found. Search forward for FM, and check that this is detected. Check that FM is found when a record is read. Check that FM status bits and error bits are correctly set.

### Interrupt test

Test that interrupt is generated on level 11 (PID bit 11 set in CPU). Verify that corresponding ident code is correct and that the IDENT instruction resets interrupt on the controller.

# Chapter 12 MEMORY

# CHAPTER 12 MEMORY

This chapter describes the test program called MEMORY.

## 12.1 GENERAL

This test program will test memory modules on ND-100 computer systems. It runs with the test-program monitor and library. The program permits testing of systems with:

- non-correcting local memory
- error-correcting local memory
- PIOC and Ethernet memory
- big multiport memory (BMPM)
- Multiport 4 Memory (MPM 4)
- Multiport 5 Memory (MPM 5)

The program includes a simple "Service Program", which acts upon the multiport ERROR LOG in order to print the contents of the ERROR LOG, and to scan the "Port Status". One disk interface can be run in test mode, in parallel with the actual memory test.

Please note the following:

Extended addressing will always be set.

The first 64K of memory is mapped onto the page table 0.

Memory is tested in blocks of 4K words (10.000 oct. words).

The 64K of memory currently under test is mapped onto page table 1 and is accessed by enabling Alternative Page Table usage (setting bit 0 of the status word) and then turning on and off the paging system according to needs.

## 12.2 Starting and Initial Action

MEMORY is started from the TPE-MONITOR by the command:

```
TPE>Load MEMORY <CR>
```

At start-up, the program will do the following:

- Allocate program buffers. They are placed directly after the program code and will contain information about the memory configuration.

- Find the computer type (100 / 110).

- Initialize Paging Control Registers and Page Index Table 0.

- Find out the memory type, and size and build the memory maps. When the computer has BMPM memory, the detection of the memory type (which is done by forcing parity errors) will cause errors to be reported to the ERROR LOG. These errors will be printed when using the BMPM Service Program command ERROR-LOG-SCAN.

- Print the information found by steps 2 and 4.

- Initialize the first part of the program variables:

  Indicate no tests skipped.
  Set lower and upper limits to include the total memory.
  Indicate that Single Test Mode is not selected.
  Indicate that Loop Mode is selected with Infinite loop.
  Indicate that Abort Mode is selected with abort after 8 error messages.
  Indicate that error messages are not to be suppressed.

- Clear the error lamps on ECCR and BMPM memory.

A restart of the program (at address 20) will have no influence on the setting of the different program variables i.e. SET-PARAMETERS-setting and memory-limits definitions will be unchanged.

## 12.3 Program Commands

### RUN

This command will start the actual memory test sequence as defined by the **SET-PARAMETERS** command. The following tests exist:

**Read test on program part:**
The memory area where the program is situated is read, in order to try to detect parity errors in that area. This test runs with paging off.

**Addresses in addresses:**
Memory is written with its own address as data and then read back. If the data read back is different from the memory address, an error message is displayed.

**Write/read test (7 patterns):**
Test patterns are written to and read back from memory. If the data read back is different from the data written, an error message is given.

The used patterns are:

0 52525 125252 44444 146314 31463 and 177777

**Rapidly changing address bits:**
This test will compute a checksum by executing ADD ,X; ADD ,B.

X is incremented by 1 and B is decremented by 1 until you have done this for the part of memory currently under test. Then you do the same once more and compare the two results. If they are different, an error message is given.

**Parity error detection:**
You run this test only if the tested memory has error correction. The memory is written to while forcing parity errors, and then read back. For each location read back, you should be able to expect a parity error interrupt to level 14. If this is the case then all is in order. If you do not get the interrupt or if the contents of the PES and/or PEA register is not as expected, an error message is displayed.

When running with multiport memory, this test will cause errors to be logged in the BMPM ERROR LOG. You must take this into account when using the command **ERROR-LOG-SCAN** in the BMPM Service Program. It is therefore best to run the Parity-Error-Detection test separately first, then clear the ERROR LOG, remove the Parity-Error-Detection test from the test sequence (see the command **SKIP-TESTS**) and then run the remaining tests.

**Walk test (34 patterns)**:
The patterns are:

```
0 1 2 4 10 20 40 100 200 400 1000 2000 4000 10000 20000 40000
100000 177777 177776 177775 177773 177767 177757 177737 177677
177577 177377 176777 175777 173777 167777 157777 137777 77777
```

**Example:**

```
TPE>RUN <CR>

AREA TESTED: 0.5-0.15
READ TEST ON  PROGRAM  PART     === END OF TEST ===
ADDRESSES IN ADDRESSES          === END OF TEST ===
WRITE/READ TEST (7 PATTRENS)    === END OF TEST ===
RAPIDLY CHANGING ADDRESS BITS   === END OF TEST ===
PARITY ERROR DETECTION          === END OF TEST ===
WALK TEST (34 PATTERNS)         === END OF TEST ===

=== THE TESTS ARE NOW LOOPING ===
```

## START-DMA-TRANSFERS

With this command, you can activate a driver on level 11 that carries out continuous DMA transfers in test mode either on device 500 or on device 1540. The program will tell which DMA device is used.

One can use this command to load the bus while running the actual memory test. The data that is transferred is not tested. The only error that can be caused by the DMA transfers is therefore MOR (Memory Out of Range).

**Example:**

```
TPE>START-DMA <CR>

DMA DEVICE : 500

== TRANSFERS STARTED ==

TPE>
```

## STOP-DMA-TRANSFERS

Used to stop the DMA transfers, activated by the **START-DMA-TRANSFERS** command.

**Example:**

```
TPE>STOP-DMA <CR>

== TRANSFERS STOPPED ==

TPE>
```

## PRINT-MEMORY-MAP

This command prints a map of the existing memory.

X: Memory without error correction or MPM 5

E: Memory with error correction (local or MPM 4)

M: Multiport 3 memory (Big multiport)

P: Pioc or Ethernet memory

**Example:**

```
TPE>Print-Memory-Map <CR>

                === M E M O R Y   M A P ===
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  (4K
UNIT)
    04 08 12 16 20 24 28 32 36 40 44 48 52 56 60 64  (X 1K)
BANK-----------------------------------------------
   0 E  E  E  E  E  E  E  E  E  E  E  E  E  E  E  E
   1 E  E  E  E  E  E  E  E  E  E  E  E  E  E  E  E
   2 E  E  E  E  E  E  E  E  E  E  E  E  E  E  E  E

     TOTAL MEMORY SIZE : 192K WORDS

TPE>
```

## DEFINE-TEST-AREA <LOWER BANK> <LOWER 4K> <UPPER BANK> <UPPER 4K>

This command allows you to define the memory area to be tested. It asks for 4 parameters with which you can define the lower and upper limits of the memory to be tested. The resolution of these limits is 4k (10.000 octal).

Suppose, for example that Bank 2 and 3 both have 64k memory and you want to test the upper 32k in bank 2 and the lower 32k in bank 3. Define the following values (decimal):

Lower    bank: 2
Lower 4k unit: 8
Upper    bank: 3
Upper 4k unit: 7

**Example:**

```
TPE>Define-Test <CR>
LOWER    BANK   (0-2 DEC.): 1 <CR>
LOWER 4K UNIT  (0-15 DEC.): 0 <CR>
UPPER    BANK   (1-2 DEC.): 1 <CR>
UPPER 4K UNIT  (0-15 DEC.): 7 <CR>

TPE>
```

## DEFINE-DMA-AREA <LOWER BANK> <UPPER BANK>

To define the lower and upper bank of memory that will be used by the DMA transfers (see commands **START**-and **STOP-DMA-TRANSFERS**). The resolution is 1 bank (64k of memory).

**Example:**

```
TPE>DEF-DMA <CR>
LOWER BANK (0-2 DEC.): 0 <CR>
UPPER BANK (0-2 DEC.): 1 <CR>

TPE>
```

## SET-PARAMETERS <YES/NO> <VALUE> <YES/NO> <VALUE> <YES/NO>

You can use this command to define how the memory test should run and when to abort the test. The defaults for the different parameters are always equivalent to the current setting.

The following possibilities exist:

**Single test mode:**
If you select this mode, only one test is run. The test to be run will be asked for when giving the command **RUN**.
DEFAULT= No.

**Loop mode:**
This mode makes it possible to specify how long the test(s) will run. If you select the mode, you can define a maximum loop-count which specifies how many times the test(s) shall run. If you give 0 as maximum loop-count, the test will run until it is stopped. If you do not select this mode, the test(s) will run only once.
DEFAULT= Yes.

**Abort mode:**
By selecting this mode, you can specify the number of error messages to be given before the test is aborted. If abort mode is not selected, tests will never be aborted.
DEFAULT= Yes.

**Suppress errors:**
If you select this mode, no error messages are printed except

'*** ERROR ***'

at given intervals. You can use this mode to speed up the program while still making use of the error condition. If the mode is not selected, error messages will be printed.
DEFAULT= No.

**Example:**

```
TPE>SET-PARAMETERS <CR>
SINGLE TEST  (YES/NO): NO <CR>
LOOPMODE     (YES/NO): YES <CR>
HOW MANY TIMES (0=INFINITE) (DEC.): 10 <CR>
ABORTMODE    (YES/NO): Y <CR>
AFTER HOW MANY ERRORS (1-65535 DEC.): 4 <CR>
SUPP. ERRORS (YES/NO): N <CR>

TPE>
```

## REFRESH-TEST <NUMBER OF MINUTES DELAY>

This command allows you to test if the refresh on the memory modules functions correctly. Make sure the other tests run correctly before running this test. The default value for <delay> is 5 minutes.

**Example:**

```
TPE>REFRESH-TEST <CR>
DELAY BETWEEN WRITE AND READ IN MINUTES (DEC.): 1 <CR>
 0,00:03 == START OF TEST ==
 0,00:04 === END OF TEST ===

TPE>
```

## SKIP-TESTS <YES/NO> <YES/NO> <YES/NO> <YES/NO> <YES/NO>

With this command, you can define one or more tests that are to be skipped by the **RUN** command. The test(s) is(are) only skipped if the program is not in single-test mode (see the **SET-PARAMETERS** command). The initial values are:

No tests skipped.

The current setting will be the new default values.

**Example:**

```
TPE>SKIP <CR>
ANSWER "YES" FOR EACH TEST YOU WANT TO SKIP

READ TEST ON  PROGRAM  PART      (YES/NO): Y <CR>
ADDRESSES IN ADDRESSES           (YES/NO): N <CR>
WRITE/READ TEST (7 PATTRENS)     (YES/NO): N <CR>
RAPIDLY CHANGING ADDRESS BITS    (YES/NO): N <CR>
PARITY ERROR DETECTION           (YES/NO): Y <CR>
WALK TEST (34 PATTERNS)          (YES/NO): N <CR>

TPE>
```

# BIG-MPM-SERVICE-PROGRAM

This command allows you to access the BMPM ERROR LOG. Five commands are possible:

**CLEAR-ERROR-LOG:**
This will initialize the ERROR LOG module by reading it (the data read is not displayed). This is necessary after power-up.

**Example:**

```
TPE>BIG-MPM-SERV-PROG <CR>

SERVICE COMMAND: CLEAR-ERROR-LOG <CR>
CLEAR-ERROR-LOG

== LOG CLEARED ==

SERVICE COMMAND: EXIT <CR>
EXIT

TPE>
```

**PORT-STATUS-SCAN:**
This gives the BMPM configuration concerning crates, banks, ports, lower-limit and upper-limit. Legal values are:

Crate: 0-7; Bank: x or y; Port: a, b, c or d; Limits: 0-77

**Example:**

```
TPE>BIG-MPM-SERV-PROG <CR>

SERVICE COMMAND: PORT-STATUS-SCAN <CR>
PORT-STATUS-SCAN

SCAN REG. ERR. CRATE BANK PORT LOWER  UPPER (NOTE 3)
140000:    0     0    X    A   000000
100102     0     0    X    A          000002

** END OF SCAN **

SERVICE COMMAND: EXIT <CR>
EXIT

TPE>
```

**ERROR-LOG-SCAN:**
This command reads (and therefore resets) the ERROR LOG. It will tell either that no errors have been found or, if there were errors, in which crate, module and IC.

Please note that the errors forced by the parity-error detection test will be reported to the ERROR LOG. Also, when building up the memory maps during program start-up, parity errors are forced in order to detect the memory type. These errors are also reported the first time you give the command error-log-scan.

**Example:**

```
TPE>BIG-MPM-SERV-PROG <CR>

SERVICE COMMAND: ERROR-LOG-SCAN <CR>
ERROR-LOG-SCAN
CRATE MODULE IC-POSITION
  0     00         37
  0     00         37

SERVICE COMMAND: EXIT <CR>
EXIT

TPE>
```

**MODULE-TO-SLOT-CONVERSION:**
Gives the correspondence between "module" (as given by the ERROR LOG SCAN) and slot-position in the BMPM crate.

**Example:**

```
TPE>BIG-MPM-SRV-PROG <CR>

SERVICE COMMAND: MODULE-TO-SLOT-CONV <CR>
MODULE-TO-SLOT-CONVERSION
    MODULE: 0  1  2  3  4  5  6  7  10  11  12  13
CRATE SLOT: 1  2  3  4  5  6  7  8  32  31  30  29

SERVICE COMMAND: EXIT <CR>
EXIT

TPE>
```

EXIT:
Leave the service-program

## 12.4 Error Messages

Since an error is very likely to be followed by many others of the same type, error messages are printed in the form of tables. To speed up printout, this type of message will start by printing an error-message header followed by lines of Error Data. The header will be repeated at regular intervals. The very first error message in an error sequence will be preceded by the text

'*** ERROR ***    TEST NR: <test no>'.

You can suppress error message printouts by selecting the correct mode with the command **SET-PARAMETERS**. If Suppress-Error-Message mode is selected, then only the text

```
'*** ERROR    TEST NR: <test no>'
```

will be printed for every 200000 (oct.) error messages (within one test routine).

Each error message will cause a counter to be incremented and it is this counter that is used to find out if the running test should be aborted (if Abort Mode is selected). Parity errors will also increment this counter.

## PARITY ERRORS

Parity errors can occur anywhere during the run of the test program. They are reported by a routine that runs on level 14.

**Example:**

```
TPE>RUN <CR>

READ TEST ON PROGRAM PART     === END OF TEST ===
ADDRESSES IN ADDRESSES

*** MEMORY PARITY ERROR *** TIME: 0,00:05
***    ADDRESS (PEA): 000000  BANK  (DEC.): 1
***     STATUS (PES): 035001
*** PES-REG DECODED:
***    BIT 13 : FATAL ERROR. MULTIPLE ERROR OCCURRED
***    ERROR CORRECTION CODE (BITS 8-12 OF PES)(OCT.): 00
*** (PREVIOUS LEVEL (DEC.): 0   ADDRESS: 021525)
```

Decoding of the PES registers is done with respect to the computer you are running on since the format of the PES register is different for each computer type.

If the parity error occurs during an error message printout, the currently running test will be aborted by making a call to the the escape-handling routine.

## ERRORS DURING TEST 1: READ TEST ON PROGRAM PART

No error messages are printed by this test though it can happen that you get parity errors.

## ERRORS DURING TEST 2: ADRESSES IN ADRESSES

**Example:**

```
*** E R R O R ***  IN ADDRESSES IN ADDRESSES
  TIME    BANK  ADDRESS  EXPECTED  FOUND (TEST 2)
 0,00:06   1    170060   170060    000001
 0,00:06   1    170061   170061    000002
 0,00:06   1    170062   170062    000003
 0,00:06   1    170063   170063    000004 *** ROUTINE ABORTED
***

TPE>
```

## ERRORS DURING TEST 3: WRITE/READ TEST PATTERNS

The format of the error message given by this test is the same as that in test 2.

**Example:**

```
*** E R R O R ***  IN WRITE/READ TEST (7 PATTERNS)
  TIME    BANK  ADDRESS  EXPECTED  FOUND (TEST 3)
 0,00:07   1    040200   125252    125242
 0,00:07   1    040202   125252    125242
 0,00:07   1    040206   125252    125242
 0,00:07   1    040210   125252    125242 *** ROUTINE ABORTED
***

TPE>
```

## ERRORS DURING TEST 4: RAPIDLY CHANGING ADDRESS BITS

**Example:**

```
*** E R R O R ***  IN RAPIDLY CHANGING ADDRESS BITS

                 LOWER   UPPERR
  TIME   BANK   ADDRESS ADDRESS CHECKSUM: FIRST SECOND  (TEST 4) (NOTE 2)
 0,00:10   1    040000  047777            161140 161370
 0,00:10   1    050000  057777            160550 161010
 0,00:10   1    060000  067777            160560 161020
 0,00:10   1    070000  077777            160560 160570
*** ROUTINE ABORTED ***

TPE>
```

## Errors during test 5: Parity error detection

This test has three different error-message types. The first type is used when the forced parity did not cause interrupt on reading memory back.

**Example:**

```
*** E R R O R ***  IN PARITY  ERROR  DETECTION
DID NOT GET PARITY ERROR INTERRUPT (TEST 5)
  TIME    BANK  ADDRESS
 0,00,12   1    000000
 0,00,12   1    000001
 0,00,12   1    000002
 0,00,12   1    000003 *** ROUTINE ABORTED ***

TPE>
```

The second error message type is used when you did have parity error interrupt, but the data you read back was not as expected.

**Example:**

```
*** E R R O R ***  IN PARITY  ERROR  DETECTION
   TIME   BANK  ADDRESS  EXPECTED  FOUND (TEST 5)
 0,00,15   0    050000   000000    000001
 0,00,15   0    050001   000000    000001
 0,00,15   0    050002   000000    000001
 0,00,15   0    050003   000000    000001 *** ROUTINE ABORTED
***

TPE>
```

The third error message type is used when, after parity error interrupt, the contents of the PES and/or PEA register is not as expected. The format of this message is:

**Example:**

```
*** E R R O R ***  IN PARITY  ERROR  DETECTION
WRONG BANK/ADDRESS FOUND IN PES/PEA AFTER PAR.ERR.INTERRUPT (TEST 5)
  TIME   EXP. BANK/ADDRESS  FOUND BANK/ADDRESS  (PES-REG)
 0,00:24        1  050000          0     025500    037400
 0,00:24        1  050001          0     025500    037400
 0,00:24        1  050002          0     025500    037400
 0,00:24        1  050003          0     025500    037400
*** ROUTINE ABORTED ***

TPE>
```

## Errors during test 6: Walk test

Since the Walk Test is basically a write/read test, the error-message format is the same as that in tests 2 and 3.

**Example:**

```
*** E R R O R ***  IN WALK TEST (34 PATTERNS)
  TIME    BANK  ADDRESS  EXPECTED  FOUND (TEST 6)
 0,00:21   1    040201   000010    000000
 0,00:21   1    050200   000010    000000
 0,00:21   1    060200   000010    000000
 0,00:21   1    070200   000010    000000
*** ROUTINE ABORTED ***

TPE>
```

# Chapter 13 PAGING

# Chapter 13 PAGING

This chapter describes the PAGING-test program.

## 13.1 General

The program PAGING is designed to test the paging and memory-management system functions on the ND-100 / ND-110 computer systems. It runs on all ND-100 / ND-110 types and with both MMS 1 and 2 types in stand-alone mode only.

## 13.2 Starting and Initial Action

The PAGING is started from the TPE-MONITOR by the command:
TPE> Load PAGING <CR>

At start-up, the program does the following:

1. Find CPU and MMS configuration.
2. If MMS type 2 is found, the ND-100 microprogram is checked to find whether or not it is updated to read the extended PCR and PSG bits. If not, these bits will not be checked through this test.

   If the computer is an ND-100/CX, a message is output stating that an ECO essential to the running of SINTRAN III VSX/K is missing from the microprogram.

## 13.3 Program Commands

## SET-PARAMETERS

This command allows you to specify how the program shall run the tests.

Format:
SET-PARAMETERS <loop> <abort> <suppress> <REX> <SEX> <MMS mode>

<loop> : You must answer YES or NO.
YES means the test or test sequence is repeated.
NO means the test runs only once.
DEFAULT: No.

<loops> : If <loop> was YES, you must specify the number of times the test or test sequence shall repeat.
DEFAULT: Infinite.

<abort> : You must answer YES or NO.
YES means the test(s) are aborted when maximum number of errors are reached.
NO means the test(s) will never abort.
DEFAULT: Yes.

<errors> : If <abort> was YES, you must specify the maximum number of errors allowed. When this limit is reached, the test(s) will abort.
DEFAULT: 10.

<suppress> : You must answer YES or NO.
YES means error messages are supressed.
NO means error messages are printed.
DEFAULT: No.

<REX> : You must answer YES or NO.
YES means test in Normal (REX) mode.
NO means do not test in REX mode.
DEFAULT: No.

<SEX> : You must answer YES or NO.
YES means test in Extended (SEX) mode.
NO means do not test in SEX mode.
DEFAULT: Yes.

<MMS mode> : You must answer MMS-1 or MMS-2.
DEFAULT: MMS-2 if present, else MMS-1..

When starting PAGING you will get the default setting on all parameters.

## RUN

This command allows the execution of one specific test, or a subset of tests in sequence.

Format:
RUN <Test number(s)>

<Test number(s)> :
Test numbers are in the range 1 to 11.
Here you can choose one of four ways to use RUN:

- ALL TESTS: Give default value
- SEQUENCE : Test numbers separated by comma or space
- RANGE : First and last test number separated by :
- SINGLE : One single test number

**Examples**

```
RUN,,:                        % Run All
RUN 3,5,2:6,1                 % Run 3-5-2-3-4-5-6-1
```

The RUN command will print a start header containing the loop mode, Abort mode and the current time. Each test will identify itself and give a 'end of test' indication. The test sequence will repeat if the loop mode is selected.

## DECODE

This command is used to decode a 16 bits octal value following the bits assignement for a specific item.

Format:
DECODE <item> <value>

```
<item>  :
    PGS       = paging status register.
    PCR       = paging control register.
    PIT-ENTRY = page table entry, control bits.

<value> :
    The octal value for the specified item.
```

After giving the octal value, the decoded information is printed out.

## 13.4 Test Description

You can carry out the tests in all possible modes. You select this with the command SET-PARAMETERS. Default are: EXTENDED and MMS-2 if the CPU have this option.

NORMAL mode : 4 page table mode, 18 bits address (ND-10 compatible, REX).

EXTENDED mode: 4 page table mode, 24 bits address (SEX).

MMS-1 mode : 4 page table mode.

MMS-2 mode : 16 page table mode, 24 bits address.

The NORMAL and EXTENDED modes are used for both MMS-1 and MMS-2 types. The MMS-2 mode is used only for a MMS-2 type.

## TEST 1 - PAGING AREA AS MEMORY (ADDRESS AND DATA BITS)

This first section includes a test for the CPU status bits 0 (APT select) and 13 (Extended mode indicator). This is only a test for setting and resetting the bits, not for the functions associated with these bits.

The page tables - the shadow memory - are tested like a main memory area, from the corresponding lower address (the first address of the first table: 177000 for MMS-1, 174000 for MMS-2), to the upper address 177777.

A static test is carried out with the patterns:

000000, 177777, 125252, 052525

An address test is carried out by writing address in address and reading back.

A dynamic test is carried out by setting up known random patterns in a page table, moving this page table to another one (fast mixed read write operations), and reading and checking both source and destination page tables. This test is carried out for all page tables, and repeated 400 times.

## TEST 2 - PAGING CONTROL REGISTERS ON ALL LEVELS

The 16 PCR registers are tested. With a MMS-1 type, only bits 0,1,7,8,9,10 may be read back. With a MMS-2 type and an updated micro program, the extended bits 2,11,12,13,14 are also tested.

A static test is carried out with patterns using all accessible bits.

Patterns:

```
MMS 1 : 000000 003603 002402 001201 003000 000600
MMS 2 : 000000 077603 042000 021002 010401 040203 003600 074000
```

An address test is carried out by writing each PCR with its interrupt level number in bits 7-10, and reading back.

The whole test runs 1000 times.

## TEST 3 - PGU/WIP BITS FOR ALL PITS AND ENTRIES

The PCR registers and one page table are set up to run the program in paging ON on ring 2. All other page tables are set up to access the same memory page from all entries (the first free page following the program) and to be used as alternative page tables.

All entries in all alternative page tables are used to read the memory. The PGU bit (page used, bit 11) must be one, the WIP bit (written in page, bit 12) must be zero.

All entries in all alternative page tables are used to write the memory. The PGU bit must be one, the WIP bit must be one.

The test is carried out with each page table used to run the program.

As this section is the first one running in paging ON, the CPU status bit 14 (paging ON indicator) is tested. In the event of a heavy paging failure, the program may crash or hang up at this section.

## TEST 4 - ALTERNATIVE PIT USAGE ON ALL LEVELS

The PCR registers and one page table are set up to run the program in paging ON on ring 2. All other page tables are set up to access the same memory page from all entries and to be used as alternative page table. On interrupt level 1 to 13: the memory is read through all entries from all alternative page tables, using the addressing modes:

```
   I
,B
,B I
     ,X
,B   ,X
   I ,X
,B I ,X
```

The value read from memory is checked to ensure that access was made at the correct address. The test is carried out with each page table used to run the program.

## TEST 5 - RING VIOLATION INTERRUPT ON ALL LEVELS

Tests that the ring violation interrupt (IIC code 2) is generated when doing the following accesses onto a higher ring:

| | |
|---|---|
| On instruction fetch | (JMP ,X) |
| On data read | (LDA ,X) |
| On data write | (STA ,X) |
| Address fetch of indirection | (LDA I ,B) |

The PCR registers and one page table are set up to run the program level 0 on ring 3. For interrupt level 1 to 12: another page table is set up to run the interrupt program part on ring 0, and the other addressable parts on ring 1. Another one is set up to be used as an alternative page table on ring 1, for data access. The three ring violations are provoked and tested on each interrupt level.

The test is restarted by incrementing the ring numbers by one, up to the last possibility, and with each page table used for the interrupt program.

The whole test is carried out with each page table used to run the main program part on level 0.

## TEST 6 - PAGE FAULT INTERRUPT

Tests that the page-fault interrupt (IIC code 3) is generated when doing the following accesses:

| | |
|---|---|
| On instruction fetch | (JMP ,X) |
| On P relative read | (LDA *+2) |
| On P relative write | (STA *+2) |
| Address fetch of indirection | (LDA I ,B) |
| On indirect read | (LDA I *+2) |
| On indirect write | (STA I *+2) |

The PCR registers and one page table are set up to run the program part on ring 2, while the remaining entries point to a single memory page (for the "on instruction fetch" part of the test). The other page tables are set up to be used as alternative page tables on ring 2, for data access.

The specified access rights are: No fetch permit, no write permit, no read permit. All entries in all tables are used to force the several page-fault interrupts.

The test is carried out with each page table used to run the program.

## TEST 7 - READ PROTECT VIOLATION INTERRUPT

Tests that the read protect violation interrupt (IIC code 2) is generated when doing the following accesses:

| | |
|---|---|
| On P relative read | (LDA *+2) |
| Address fetch of indirection | (LDA I ,B) |
| On indirect read | (LDA I *+2) |

The PCR registers and one page table are set up to run the program part on ring 2. The other page tables are set up to be used as alternative page tables on ring 2, for data access.

The specified access rights are: Fetch permit, write permit, no read permit. All entries in all tables are used to force the several read-protect violation interrupts.

The test is carried out with each page table used to run the program.

## TEST 8 - WRITE PROTECT VIOLATION INTERRUPT

Tests that the write-protect violation interrupt (IIC code 2) is generated when doing the following accesses:

| | |
|---|---|
| On P relative write | (STA *+2) |
| On indirect write | (STA I *+2) |

The PCR registers and one page table are set up to run the program part on ring 2. The other page tables are set up to be used as alternative page tables on ring 2, for data access.

The specified access rights are: Fetch permit, no write permit, read permit. All entries in all tables are used to force the several write-protect violation interrupts.

The test is carried out with each page table used to run the program.

## TEST 9 - FETCH PROTECT VIOLATION INTERRUPT

Tests that the fetch protect violation interrupt (IIC code 2) is generated when doing the following access:

| | |
|---|---|
| On instruction fetch | (JMP ,X) |

The PCR registers and one page table are set up to run the program part on ring 2, while the remaining entries point to a single memory page. The specified access rights are: No fetch permit, write permit, read permit. All entries after the program area are used to force the fecth protect violation interrupt.

The test is carried out with each page table used to run the program.

## TEST 10 - PRIVILEGED INSTRUCTION INTERRUPT

Tests that the privileged instruction interrupt (IIC code 6) is generated when running on ring 0 or 1, and is not generated when running on ring 2 or 3. In addition, verifies that the monitor call (MON) instruction, does not generate the privileged instruction interrupt, but the monitor call interrupt (IIC code 1).

The PCR registers and one page table are set up to run the main program part on level 0 and ring 2. One page table is set up to run a program containing privileged instructions. This program is executed on level 1 to 12 and on all rings, through all entries of the page table.

The test is restarted for all page tables, and with different page tables used for the main program on level 0.

The privileged instructions used for test are:

MON 0
IOX 302
TRA OPR
TRR CCLR
MCL STS
MST STS
IRW 170 DA
IRR 170 DA
EXAM
PION

## TEST 11 - PHYSICAL ADDRESSES GENERATION

Tests that the physical memory address going out of the translation through the page tables is correct.

The main memory is addressed several times in all possible physical pages, from bank 255 down to bank 0. It is addressed both through the paging to activate the address translation logic, and by the means of physical addressing instructions (EXAM and DEPO) which does not use the same address generator.

When addressing a nonexistent memory area, the verification is carried out by reading the PES/PEA registers on memory out of range.

When addressing an existing memory area, the verification is carried out by writing and reading back several patterns.

## 13.5 Error Messages

Since an error is very likely to be followed by many others of the same type, error messages are printed in the form of tables. To speed up printout, this type of message will start by printing an error-message header followed by lines of Error Data.

The header will be repeated at regular intervals. The very first error message in an error sequence will be preceded by the text

***** ERROR *** In test xxx. Time: yyyy.mm.dd hh:mm:ss**

You can suppress the error message printout by selecting the <Suppress mode> with the SET-PARAMETERS command. The header text is then only printed the first time an error occurs within a test section.

# Chapter 14 PIOC-ETHER

# CHAPTER 14 PIOC-ETHER

The purpose of this test program, is to test PIOC and ETHERNET controllers, STAND ALONE.

It is advisable to run tests 1-11 before the other tests. This will ensure a proper initializing of the PIOC.

## 14.1 STARTING AND INITIAL ACTION

PIOC-ETHER is started from the TPE-MONITOR by the command:

```
TPE>Load PIOC-ETHER <CR>
```

## 14.2 PROGRAM COMMANDS

This section gives a short description of the different program commands.

### LIST-PRESENT-DEVICES

This command prints a list of all present PIOC and ETHERNET devices.

### SELECT-DEVICE <DEVICE NAME> <DEVICE NUMBER,1-16)>

This command is used to select current PIOC / ETHERNET device.

### LIST-ALL-DEVICES

This command prints a list of all defined PIOC and ETHERNET devices.

## RUN <TEST NUMBER(S)>

This command allows the execution of one specific test, or a subset of tests in sequence.

You can choose one of four ways to use RUN:

- ALL TESTS: Give default value
- SEQUENCE : Test numbers separated by comma or space
- RANGE : First and last test number separated by :
- SINGLE : One single test number

**Examples:**

```
TPE>RUN,,                  % Run All

TPE>RUN 3,5,2:6,1          % Run 3-5-2-3-4-5-6-1
```

The RUN command will print a start header. Each test will identify itself and give an 'end of test' indication. The test sequence will repeat if loop mode is selected.

## SET-PARAMETERS <V-LEVEL> <LOOP MODE> (<LOOPS>) <ABORT MODE> (<ERRORS>) <SUPPRESS MODE>

This command allows you to define how to run the tests.

**The parameters:**

**<V-level>**
You must answer YES or NO.
YES means LINE-TEST with V-level test-plug, and NO means X-level test-plug. (Only for test 40)
DEFAULT: Yes.

**<loop mode>**
You must answer YES or NO.
YES means the test or test sequence is repeated.
NO means the test runs only once.
DEFAULT: No.

**<loops>**
If <loop mode> was YES, you must specify the number of times the test or test sequence shall repeat.
DEFAULT: Infinite.

**NOTE:** If you want the loop-count on the screen, you must use the command:

```
TPE>Set-Terminal-type 53 <CR>
```

**<abort mode>**
You must answer YES or NO.
YES means the test(s) is aborted when the maximum number of errors is reached.
NO means the test(s) will never abort.
DEFAULT: Yes.

**<errors>**
If <abort mode> was YES, you must specify the maximum nuber of errors allowed. When this limit is reached, the test(s) will abort.
DEFAULT: 10.

**<suppress mode>**:
You must answer YES or NO.
YES means error messages are supressed.
NO means error messages are printed.
DEFAULT: No.

When starting PIOC-ETHER-A00 you will get the default setting on all parameters.

## 14.3 Test Description

These are the tests for p(ioc) and e(thernet):

p= test executed for Pioc.
e= test executed for Ethernet.

Test 1 to 11 checks basic funtionality of the selected PIOC / ETHERNET.

| | |
|---|---|
| p/e | 1. Check iox communication with ND-100. |
| p/e | 2. Quick check and initializing of memory, done by ND-100. |
| p/e | 3. Check that MC68000 enters run mode, and stops again. |
| p/e | 4. Check interrupt to ND-100, and read ident code. |
| p/e | 5. Check Interrupt controller, reset. (AM 9519) |
| p/e | 6. Check Interrupt controller, software interrupt. |
| p/e | 7. Check System timing controller, reset. (AM 9513) |
| p/e | 8. Check System timing controller, write/read of registers |
| p/- | 9. Check DMA controller, write/read of registers. (AM 9517) |
| p/- | 10. Check SIO controller, maintenace mode. (MK3884) |
| p/e | 11. Check OPCOM interrupt to PIOC, and RTC generator. (AM 9513) |

Test 20-27 tests the PIOC/ETHERNET memory as viewed from ND-100 and from MC68000.

|  |  |  |  |
|---|---|---|---|
| p/e | 20. Pattern-test | : | Write/read different patterns from ND-100. |
| p/e | 21. Address-test | : | Write/read address in address from ND-100. |
| p/e | 22. Pattern-test | : | Write/read different patterns from MC68000. |
| p/e | 23. Address-test | : | Write/read address in address from MC68000. |
| p/e | 24. ECC-test | : | Test the error correction from ND-100. |
| p/e | 25. ECC-test | : | Test the error correction from MC68000. |
| p/e | 26. Protect-test | : | Test the write protection from MC68000. |
| p/e | 27. refresh-test | : | Test the refresh of mem. from MC68000. |

Test 40 tests the lines with test-plugs.

p/- 40. Test line 0-3 either with v- or x-level test-plug.

**NOTE:** From version B00 of the test program, test 40 will be run without test plug (default). If you then want to run the test WITH test plug, this must be specified with the command SET-PARAMETERS.

# CHAPTER 15 POWER-FAIL

# CHAPTER 15 POWER-FAIL

This chapter describes the POWER-FAIL test program.

## 15.1 GENERAL

The program POWER-FAIL is designed to test the power failure and restart detection on ND-100/ND-500 computer systems, seen from ND-100. It also verifies that the main memory content is not corrupted during the time the power is OFF. It runs on all ND-100/ND-500 types, only in stand-alone mode. The program run under the TPE-MONITOR.

## 15.2 STARTING AND INITIAL ACTION

POWER-FAIL is started from the TPE-MONITOR by the command:

```
TPE>Load POWER-FAIL <CR>
```

## 15.3 PROGRAM COMMANDS

There is only one program command in this test program:

RUN.

Before running this command, you have to take some precautions:

- Write protect the disks
- Set the key in the locked position
- Set the ALD (upper thumbwheel switch on the CPU to 12

**Write protect**

This is done in order to secure the data on the disks. It can be done by the WRITE PROTECT button, or by turning the disks off.

**Key in locked position**

In order to make the computer restart after a power fail, you must put the panel key in the "lock" position.

**ALD set to 12**

The ALD thumbwheel on the CPU card must be set to 12, which is the normal setting. This means restart from address 20 if the standby power is working.

Now you are ready to give the RUN command.

## RUN

Format:

```
RUN
```

The program will now initialize

- Memory with patterns.
  You will see the initialization progress on the console.
- Bus Expanders and Bus Controllers.
  The initialized devices will be reported on the console.

The following message will be displayed:

```
-WAITING FOR POWER FAIL-I am now waiting for a power failure
                       -Will you please turn me off !
```

Turn the power OFF; Now you must turn the power OFF. Use the main AC switches, or just pull out the mains AC plug.

If you just want to test the power fail mechanism, you can turn the power ON again after 3 seconds. If you want to test the STANDBY power capasity you must wait up to the time guaranteed by ND before you switch the power ON again.

**Turn the power ON**

Now you can turn the power ON. If everything is working normal, TPE will respond with the following message:

```
==TPE42=> *** POWER FAIL ***
```

The POWER-FAIL test program will be restarted

**NOTE:**

If you do not get this message, but a new TPE from the floppy, it means that something is wrong with the standby power.

POWER-FAIL will tell you which device the power fail was detected by:

```
POWER FAIL OCCURED- <Source of the power fail>
```

The source could be: ND-100 CPU
A BUS Expander
A BUS Controller

The memory is checked:

```
-MEMORY VERIFICATION - Checking memory for valid contents.
                       This may take some time.
```

If errors occure, they will be reported like this:

```
*** ERROR ***         It seems that the display panel
                      calendar is not working properly.


*** ERROR ***         MEMORY CONTENT IS WRONG.
                       Address      Expected     Found
                       .......      ........     .....
```

# CHAPTER 16 THE PRINTER TEST PROGRAM

# CHAPTER 16 THE PRINTER TEST PROGRAM

The printer test program described here replaces the old LINE-PRINTER test program. It incorporates most of the functions of the older program but, since the range of interfaces have shrunk during the last few years, most of the interface test has been omitted.

The only interface that can still be tested is the Plotter-Printer-DMA-Interface, which is currently used for interfacing to the Fujitsu line-printer. Most other printers are connected via an ordinary terminal asynchronous interface (RS232 or current loop), and so do not need an interface test of their own.

This test program has been modified to reflect changes in printers in the last few years.

## 16.1 GENERAL INFORMATION

The test program is divided into different test routines, not all of which are appropriate for all printers.

The routines are designed to give an indication of the state of the mechanics of the printers, assuming that the controlling software is correct.

**NOTE:** Some tests - in particular diagonal-lines, circles and cross-hatch, can take some time to run.

The test routines cover such operations as:

- horizontal and vertical positioning of the print-head
- horizontal tabulation
- print attributes (bold, italics, underline etc.)
- fonts/character sizes
- selection of feeder/sorter bin
- printing of graphics

The graphics tests serve two purposes. First, they give an indication of whether the paper feed or print-head positioner needs adjustment; second, they test the individual needles of dot-matrix printers.

In addition to the 'general' routines, there are also a few that can only be run on specific printers, e.g. the big-letters test, which prints big letters with the block graphics characters on a Mannesmann-Tally MT6xx printer.

Since some of the tests use only one character font/size etc, it is sensible to allow the user to be able to change some of the default settings. This can be done with the set-parameter command.

There is also a 'run' routine which executes all the most important tests on the specified printer type. Test routines that tend to use lots of paper are omitted.

## 16.2 Starting and Initial Action

THE PRINTER TEST PROGRAM is started from the TPE-MONITOR by giving the command:

```
TPE>Load PRINT <CR>
```

## 16.3 The Commands

The following program commands can be used:

```
SELECT-DEVICE
SET-PARAMETER
SELECT-TEST-PATTERNS
PRINT-TEST-PATTERNS
COLUMN-TEST
PRINT-STYLES
ATTRIBUTES
TABULATOR-TEST
HORISONTAL-POSITIONING
VERTICAL-POSITIONING
NEEDLE-TEST
BIDIRECTIONAL-PRINTING-TEST
CIRCLES
CROSS-HATCH
FEEDER-TEST
SORTER-TEST
SEND-FORM-FEEDBIG-LETTERS
DIAGONAL-LINES
INTERFACE-TEST
RUN
```

## SELECT-DEVICE

This routine allows the operator to select which device to test, and to specify it with regard to type, paper size, graphics printing ability etc. Before this routine is run, no test routine can be executed.

## SET-PARAMETER

Allows the operator to change the settings of given default parameters. For a description of the different fonts/pitches etc, see the section :"Different Printer Information". The possible choices are:

**SET-PARAMETER PITCH**
Enables the operator to change the default character size. The operator is also asked to specify the size of the given pitch number, so that the program can calculate the maximum number of characters on a line.

**SET-PARAMETER FONT**
Enables the operator to change the default font. The font and pitch parameters should be modifiable, so that the operator can test different combinations of typeface, character size and attributes than the default set.

**SET-PARAMETER FEEDER**
Selects the feeder bin from which the printer takes the paper. Note that for the Elpho-20 printer, a parameter value of 0 means that the printer takes paper from the first non-empty bin found. 0 is the default value, and is set by the command

```
SET-PARAMETER RESET
```

**SET-PARAMETER SORTER**
Selects the sorter bin to which the printer sends the output. Note that for the Elpho-20 printer, a parameter value of 0 means that the printer sends the output to the first non-full bin found. 0 is the default value, and is set by the command

```
SET-PARAMETER RESET
```

**SET-PARAMETER LINE-LENGTH**
Sets the line length for printers with a fixed pitch/maximum line length. This parameter cannot be set. for printers with multiple character sizes.

**SET-PARAMETER RESET**
Resets the parameters to the values specified by the program.

## SELECT-TEST-PATTERNS

The program has 5 built-in test patterns, which can be individually selected. Running SELECT-TEST-PATTERNS before the pattern-printing routine (PRINT-TEST-PATTERNS) enables the operator to specify which of the test patterns to print.

## PRINT-TEST-PATTERNS

This routine prints a number of test patterns selected from a pool of 5 patterns.

**TEST PATTERN 1**
A full line of characters in ascending order is printed. Then the pattern is shifted one position to the left and printed until all characters have been printed in every position.

**TEST PATTERN 2**
On the first line an ascending sequence of characters will be printed up to the maximum line length, while on the second line only one character will be printed. The program will then alternately print these two lines, decreasing the length of the first line and increasing the length of the second line, until the second line has reached maximum line length.

**TEST PATTERN 3**
All characters are printed, filling up one line each.

**TEST PATTERN 4**
The letter 'M' is printed in every odd column (1, 3, 5...) up to the maximum line length.

**TEST PATTERN 5**
The letter 'M' is printed in every even column (2, 4, 6...) up to the maximum line length.

## COLUMN-TEST

Prints a specified character in all columns in a given range. If nothing else is specified, the program prints the character

E

in all columns up to the maximum line length.

## PRINT-STYLES

Prints a sample of all possible combinations of typeface and character size. Note that some combinations may be undefined for certain printers. For these combinations the printer will select a pitch or typeface that will yield a valid combination. For this test the line length is adjusted according to the biggest character size, so that samples of the smaller character sizes may be broken over several lines even though they could easily be fitted onto one line.

## ATTRIBUTES

Prints a sample of all the print attributes that the printer is capable of. The samples are printed with the default pitch and font (selectable). Note that, for some printers, not all fonts/pitches can be used in conjunction with all attributes, so the printer could state that it is printing with a given attribute when it is not.

## TABULATOR-TEST

Prints a line with several occurrences of the word 'TEST' prefixed by asterisks ('*'). At every asterisk the program sets a tabulator stop. On the next line the program sends a sequence of horizontal tab codes and asterisks, which should give a line of asterisks aligned with the asterisks on the line above.

## HORISONTAL-POSITIONING

The printer prints some lines, but instead of printing the characters in sequence, it prints first the leftmost character, then the rightmost, and continues in towards the middle of the line.

## VERTICAL-POSITIONING

The program sends a sequence of text and vertical movement commands to the printer. If the printer works correctly, the lines should end up in reading order. If the printer is capable of doing bit graphics, a line of graphics is printed along with each line of text. These lines should be evenly spaced, except for the spacing between the groups of text lines.

## NEEDLE-TEST

The program makes the printer do several graphics dumps, but the graphics data is so arranged that for each dump only a single needle is used. If a particular needle is stuck or broken, this routine enables the operator to identify it quickly. For obvious reasons, this routine should only be run on impact dot-matrix printers.

## BIDIRECTIONAL-PRINTING-TEST

This routine tests whether the head-positioning mechanism of the printer needs to be adjusted or tightened. It works by sending lines of vertical bars to the printer in bidirectional printing mode. If the head positioner needs alignment, the bars will not fit completely together.

## CIRCLES

This routine builds some circles in a graphics buffer and prints them out on the printer. If there is anything wrong with the paper movement or head positioning, the circles on the output are likely to be distorted.

## CROSS-HATCH

This routine prints a grid on the printer, using the graphics printing ability. Inconsistencies in paper movement and/or print-head positioning should be clearly visible on the output from this test.

## FEEDER-TEST

This routine takes paper from each of the feeders alternately. The operator is asked to input the number of sheets to be taken from each feeder. The output is sent to the default sorter bin.

## SORTER-TEST

This routine is similar to FEEDER-TEST, except that it tests the different sorting bins instead of the feeders. The operator is prompted for the number of sheets to send to each sorter bin. The paper is taken from the default feeder.

## SEND-FORM-FEED

This routine sends a number of form-feed commands to the specified printer. Nothing is printed on the ejected forms, since the point of this test designed is to ascertain that the paper-handling mechanism is capable of ejecting the forms at maximum speed without making a mess of the paper at the output. This test should only be used for high-throughput printers using continuous forms.

## BIG-LETTERS

This routine only works with Mannesmann-Tally printers, where it prints out letters in various heights and widths. The letters are constructed from block graphics elements.

## DIAGONAL-LINES

This routine sets up a graphics pattern of diagonal lines in computer memory and then sends this pattern to the printer. Since only the ELPHO-20 printer has enough memory to keep an entire page of bit-mapped graphics, the test can only be run on that type of printer.

## INTERFACE-TEST

This routine can only be run off-line, i.e. on a computer running without an operating system. Furthermore, it tests only one particular type of interface, the plotter-printer-DMA-interface, so such an interface must be present in the computer and a DMA printer selected. The routine works by sending single words to the DMA interface and then reading them back to compare with the original. Any differences are reported to the operator.

### RUN

This routine executes a subset of the routines appropriate for the type of printer that is being tested. Note that it does not run all possible tests for a given printer, so that running other routines in addition to the set executed by RUN may be desirable.

## 16.4 Different Printer Information

Printers must be prepared for the tests, as described below:

### Philips GPXXX

The only assumption for this printer type is that the initial default paper feed mechanism is sheet feeder 1, therefore the operator must insert paper into this feeder. Alternatively, the operator can change the default feeder (by the SET-PARAMETER command) prior to running any test, but after selecting the device.

### Genicom 3024

For this type of printer the operator must manually set the form length and select the US-ASCII character set, both by the printers front-panel switches.

### Mannesmann-Tally MT6XX

This type of printer does not require any direct operator action prior to starting the program.

### Elpho-20

This printer requires no special preparations.

## Fujitsu M3023D

For this printer type, the operator must ensure that the line spacing is set to 6 lpi (lines per inch). Operation at 8 lpi is possible, but in this case the operator must specify 3/4 of the actual form length when selecting the printer.

## 16.5 CHARACTERISTICS OF THE DIFFERENT PRINTERS

### PHILIPS GPXXX

**PRINT-STYLES**
The 3 highest font numbers specify down-loadable (soft) fonts. If these fonts have not been loaded, the printer will default to font 1 (draft/data) when these fonts are selected. Also, not all typefaces can be printed in all pitches. In cases where the required combination does not exist, the printer defaults to the largest defined character size for the selected font.

**ATTRIBUTES**
In this test, the printer may state that it is printing 'bold' letters while it is actually delivering underlined print. This is because the bold-printing capability is an upgrade to the original microprogram. Some (early) printers will not have this capability, and default to underlined printing instead.

### GENICOM 3024

**TABULATORS**
The Genicom actually has tabulator operations, but it requires that all tabulator stops are set at the same time. This does not quite suit the way the test-routine is written, so the test program defines the Genicom as having no tabulator operations.

**ROUTINES DOING GRAPHICS DUMPS**
When doing a graphics dump, the Genicom 'forgets' where the top of form is, which means that there is no relation between the side breaks in the output text and on the continuous forms. The printer also stops completely at times, which can only be remedied by pressing 'Online' twice.

**NEEDLE-TEST**
The Genicom printer uses only 7 out of 9 needles for graphics dumps, so the needle test does not test all the needles. For this reason, the needle test should be seen in conjunction with the attribute test, so that the operator can infer the state of the lowermost two needles from descenders and underline printing.

## MANNESMANN-TALLY MT6XX

### PRINT-STYLES

Only font 1 (data/draft) can be printed in 15 cpi. When the other fonts are selected, choosing 15 cpi selects font 1.

### NEEDLE-TEST

The MT6XX printer is of the impact dot-matrix type, but it has the print needles arranged horizontally. This means that the print needles cannot be tested in the same way as for printers with vertically arranged needles. Instead of printing one graphics line with each needle, the MT6XX prints an all-black rectangle, where bad needles can be identified by areas that are of a lighter shade than the rest.

## CANON LBP-8 A1

### PRINT-STYLES

In the same way as some of the other printers, the Canon cannot print all fonts in all character sizes. When an invalid (undefined) combination is selected, the printer tries to find something reasonably close to the given specification.

### ATTRIBUTES

The phenomenon mentioned above also stretches to the attribute test, since the printer has separate fonts for some of the attributes (bold and light printing) instead of generating the required characters from the 'standard' font. Because of this, selecting certain attributes could easily make the printer select some other font or character size, or even ignore the attribute selection altogether, if the printer's memory contains no character set with this attribute. This is the reason that a 'standard' Canon printer will state that it is printing with the 'light printing' attribute, when it clearly is not.

### FEEDER-TEST

The Canon printer has two feeding methods, automatic and manual. These are nominated feeder 1 and feeder 2, respectively, by the program. Automatic feed can be selected at any time, but hand feeding requires that the printer must be put offline first. For this reason, the operator must stand by the printer during this test, pressing 'Error Skip' followed by 'Online' every time the printer's display starts flashing '02' (for feeder 2 - manual feed). Then the operator must feed sheets into the printer until the 'Auxiliary feed' lamp goes out.

## Elpho-20

No eccentric behaviour has so far been found with this printer.

## Fujitsu M3023D

No eccentric behaviour has so far been found with this printer.

## 16.6 The fonts, pitches etc

## THE FONTS

| | | | |
|---|---|---|---|
| **Philips GPXxxx:** | 1. | First ROM font | (Draft) |
| | 2. | Second ROM font | (Gothic) |
| | 3. | Third ROM font | (Symbols) |
| | 4. | Fourth ROM font | (Micro) |
| | 5. | Fifth ROM font | (Orator Italic) |
| | 6. | Sixth ROM font | (Courier) |
| | 7. | First down-loaded font | |
| | 8. | Second down-loaded font | |
| | 9. | Third down-loaded font | |

| | | |
|---|---|---|
| **Genicom 3024:** | 1. | Draft |
| | 2. | Near letter quality |

| | | |
|---|---|---|
| **Mannesmann-Tally MT6XX:** | 1. | Draft |
| | 2. | Near letter quality |
| | 3. | OCR A |
| | 4. | OCR B |
| | 5. | Line drawing |

| | | |
|---|---|---|
| **Canon LBP-8 A1:** | 1. | Line printer |
| | 2. | Pica |
| | 3. | Elite |
| | 4. | Courier |
| | 5. | Garland |

| | | |
|---|---|---|
| **Elpho-20:** | 1. | Letter Gothic |
| | 2. | Pica 10 |
| | 3. | Orator |
| | 4. | Elite 12 |
| | 5. | Elite 15 |
| | 6. | Courier |
| | 7. | NOTIS 1 |
| | 8. | NOTIS 2 |
| | 9. | NOTIS 3 |
| **Fujitsu M3023D:** | 1. | OCR B |

# PITCHES

| | | |
|---|---|---|
| **Philips GPXXX:** | 1. | 10 CPI |
| | 2. | 12 CPI |
| | 3. | 15 CPI |
| | 4. | Proportional, fixed space pitch |
| | 5. | Proportional, space pitch determined by character generator |
| | 6. | 14.4 CPI |
| | 7. | 18 CPI |
| **Genicom 3024:** | 1. | 10 CPI |
| | 2. | 12 CPI |
| | 3. | 17 CPI |
| **Mannesmann-Tally MT6XX:** | 1. | 10 CPI |
| | 2. | 15 CPI |
| **Canon LBP-8 A1:** | 1. | 10 CPI |
| | 2. | 12 CPI |
| | 3. | 15 CPI |
| | 4. | 6 CPI |
| **Elpho-20:** | 1. | 10 CPI |
| | 2. | 12 CPI |
| | 3. | 15 CPI |
| | 4. | Proportional, fixed space pitch |
| | 5. | Proportional, space pitch determined by character generator |

| | | |
|---|---|---|
| **Fujitsu M3023D:** | 1. | 12 CPI |

# FEEDERS

| | | |
|---|---|---|
| **Philips GPXXX:** | 1.<br>2. | Sheet feeder 1<br>Sheet feeder 2 |
| **Genicom 3024:** | 1. | Tractor feed |
| **Mannesmann-Tally MT6XX:** | 1. | Tractor feed |
| **Canon LBP-8 A1:** | 1.<br>2. | Automatic feed<br>Manual feed (Auxiliary feed) |
| **Elpho-20:** | 1.<br>to<br>4. | Feeder bin 1<br><br>Feeder bin 4 |
| **Fujitsu M3023D:** | 1. | Tractor feed |

# SORTERS

| | | |
|---|---|---|
| **Philips GPXXX:** | 1. | Output bin |
| **Genicom 3024:** | 1. | Continuous forms output |
| **Mannesmann-Tally MT6XX:** | 1. | Continuous forms output |
| **Canon LBP-8 A1:** | 1. | Output tray |

| | | |
|---|---|---|
| **Elpho-20:** | 1. | Output bin 1 |
| | to | |
| | 10. | Output bin 10 |

# Chapter 17 SYNCHRONOUS MODEM TEST PROGRAM

# CHAPTER 17 SYNCHRONOUS MODEM TEST PROGRAM

This chapter describes the Synchronous Modem test program.

## 17.1 GENERAL

This program tests the communication line between two machines with synchronous modem buffer interface (1050 card) and runs some basic tests on the 1050 card in test mode. The communication test can operate both in half duplex or full duplex mode.

If you press "carriage return" during communication, the test will tell how many blocks have been received and the number of cycle redundancy code (CRC) errors detected.

There are three ways to run this test:

- Communication test between two interfaces/machines
- Communication test in maintenance mode
- Maintenance mode

**Communication test between two interf./machines**

The output part of the program transmits blocks with 200 patterns in each block (see sketch below). After the last pattern in each block, a cycle redundancy code (CRC code) formed by the 200 patterns, is transmitted.

A sketch of each block transmitted:

```
                        block
                         No.
 ------------------------------------------------------------------------
|26 | 26 | 26 | 1 | 200 | 2 0 0  T E S T  P A T T E R N S | CRC code |
 ------------------------------------------------------------------------
 3 synch char          number of                                    CRC
<-------------->      patterns(oct.)                          checksum
```

The input part of the program reads the incoming patterns and a CRC code is formed. This CRC code is checked against the received CRC code and, if they are unequal, an error message is given. This may indicate noise on the line or an error in modem/buffer card. If no error message is given, the communication line is in order.

When running communication test between two interfaces/machines you must load and run this test program on both machines.

**Communication test in maintenance mode**

The communication test can also run in maintenance mode by looping output data back via a modem. In this way the interface can be tested/maintained. Maintenance mode cannot be run in half duplex. Check that the push button on the modem is set in loop mode.

**Maintenance mode**

The maintenance mode program carries out the following tests on the interface:

- Provoke underrun.
  The output part of interface is activated but no character is transmitted to it. This is done several times and the interface then begins to transmit characters from the output-synch register. Output-status bit 5 (synch transmitted) shall be set and input shall receive output-synch register char (301). If this does not happen, error messages will be given.

- Provoke overrun.
  By not reading one character from the input buffer, the character is overwritten and the input status bit 7 (overrun) is set. An error message will be given if this does not happen.

  Test that some input-status bits are correctly set. The bits tested are 2,4,5 and 9. If they are not correctly set an error message will be given.

- Test that the receiver-reset function works correctly.
  The input-synch register is set to 26 & 62. If the input part of interface gives interrupt on characters different from these two, an error message will be given.

  NB. The maintenance mode should be executed before the communication test is started.

  The maintenance mode operates in test mode, i.e. the output data is loop-transmitted back on interface at a rate of 19200 bit/s.

## 17.2 Starting and Initial Action

The SYNCH-MODEM test program is started from the TPE-MONITOR by the command:

```
TPE>Load Synch-modem <CR>

    SYNCH-MODEM - Version : x - yyyy-mm-dd
```

The command SELECT-DEVICE must be run to select the device you want to test. If there is only one device present, this device will be choosen automatically.

The device you specify must be identical to the DEVICE NO. set by the select function in position 15E on the 1050 buffer card.

The select function in position 1E selects the IDENT CODE for the buffer card. This select function must correspond to the device number switch (see hardware device table). If no correspondence here, error messages will be displayed.

## 17.3 Program Commands

This program contains the commands:

```
 RUN
 SELECT-DEVICE
 SET-PARAMETERS
```

## SET-PARAMETERS

You can use this command to specify HOW to run the test, see the examples below:

```
TPE>SET-PARAMETERS <CR>
Maintenance mode.. (YES or NO): No <CR>
Half duplex .......(YES or NO): No <CR>
Abort on CRC error (YES or NO): No <CR>
```

```
TPE>SET-PARAMETERS <CR>
Maintenance mode.. (YES or NO): Yes <CR>
```

```
TPE>SET-PARAMETERS <CR>
Maintenance mode.. (YES or NO): No <CR>
Half duplex .......(YES or NO): Yes <CR>
```

```
Abort on CRC error (YES or NO): Yes <CR>
Max errors before abort. (dec): 10 <CR>
```

## RUN

This command starts the test-run, as specified in the SET-PARAMETERS
command.

**Example:**

```
TPE>SET-PARAMETERS <CR>
Maintenance mode.. (YES or NO): Yes <CR>


TPE>Run <CR>

Underrun test                         === End of test ===
Overrun test                          === End of test ===
Input status bit test                 === End of test ===
Receiver reset test with input
sync register = 26 & 62               === End of test ===
```

## SELECT-DEVICE

The command SELECT-DEVICE must be run to select the device you want to test. If there is only one device present, this device will be choosen automatically. The device you specify must be identical to the DEVICE NO. set by the select function in position 15E on the 1050 buffer card.

## 17.4 Error Messages

The error messages always begin with 4 asterixes (****) and include the word ERROR. Status errors can occur in the input channel or the output channel of the buffer card. These errors are respectively referred to as "input status error" and "output status error". A status error will be reported whenever it occurs.

**Example:**

```
**** INPUT STATUS ERROR ****
STATUS: 004200
CARRIER MISSING
OVERRUN
```

**Explanation of other error messages:**

**** ERROR - WRONG CRC CODE ****

This message means that the received CRC code has been changed during transmission.

**** ERROR - FALSE SYNC INTERRUPT ****

This message means that, even though the input part of the buffer card is in receiver-reset mode (synchronised on 26), it gives interrupt on char different from 26.

**** ERROR - 2. SYNC CHAR MISSING ****

The start of each block being transmitted consists of three synch characters (26). If the second character received is different from 26, this error message will be given.

**** ERROR - 3. SYNC CHAR MISSING ****

This message means that the third and last synch character in each block is different from 26.

**** NEW SEARCH FOR THREE SYNC CHARACTERS ****

The input part of the program will not begin to read the 200 test patterns before three synch characters (26) in succession are received. If one of the synch characters is missing, the input part of interface is set in receiver reset mode (synchronised on 26) and the message above will be given.

**NOTE:**

When running the communication test, you should remember that some or all of the 5 error messages mentioned above will sometimes be given before the test has stabilized (after approx. 10 sec). If this happens, the error messages should be ignored.

**** ERROR - NO CONNECTION WITH THE MODEM/OTHER MACHINE ****

If 500 or more blocks have been transmitted and no blocks have been received, this message will be given. It may indicate burst on the line, an error in the interface/modem or wrong button(s) pushed on the modem. But it is most likely to indicate that the test program in the other machine has not yet started. NB. This message will not be given in half duplex.

# Chapter 18 TERMINAL ASYNC

# CHAPTER 18 TERMINAL ASYNC

This chapter describes the TERMINAL-ASYNC test program. This program tests asynchronous interfaces and carries out some tests to check terminal devices both on input and output.

## 18.1 GENERAL

In all tests, the program will continuously check:

- the status word
- the parity
- the interrupt
- the ident codes.

This description covers the F00 version of TERMINAL-ASYNC.

When running stand-alone, all present devices are cleared up and the speed is programmed to 9600 b.p.s on all devices.

**NOTE:** After hardware changes (switches, cables etc.), you must press Master Clear on the operator panel, and restart the program by typing 20! on the console.

## 18.2 STARTING AND INITIAL ACTION

The Terminal-Asynch test program is started from the TPE-MONITOR by the command:

```
TPE> Load Term-Async <CR>
```

After the program has been loaded, different tables will be allocated and set to zero.

If you want to test several devices simultanously, you must define a device table (see the command DEFINE-DEVICE-TABLE).

The device table is used by the following commands:

KEYBOARD-TEST
DISPLAY-TEST
INTERFACE-TEST
FIND-SPEED
SET-SPEED
VERIFY-TERMINAL-SPEED
FILL-SCREEN

When a device table has been defined, these commands will start by asking "USE DEVICE TABLE (Yes/No): " (default answer is YES).

If YES, the commands will run through the test on all devices in the table.

If NO, the command will ask for a device number.

## 18.3 The Program Commands

This section explains the different commands you can use under this test program:

COMMUNICATION-TEST
DEFINE-DEVICE-TABLE
DISPLAY-TEST
FILL-SCREEN
FIND-SPEED
INTERFACE-TEST
KEYBOARD-TEST
LIST-TERMINAL-DEVICES
LOOP-TEST
SET-PARAMETERS
SET-SPEED
TRANSLATE
VERIFY-TERMINAL-SPEED
PROGRAM-STATUS

## COMMUNICATION-TEST

Parameter: <Device number>

This command cannot be used under SINTRAN.

This command checks the asynchronous communication between two systems. The program starts sending synch characters and waits for an answer.

As soon as this answer is received, one of the systems sends a frame (all characters from 0-177 octal). The other system returns the frame, and a check is carried out to ensure that the returned frame is identical to the original.

Then the second system starts sending the frame and the first system returns it.

If no communication is established within 10 seconds, the following message appears:

```
NO REPLY, STILL TRYING.
```

After the first frame has been transmitted the program indicates:

```
COMMUNICATION RUNNING
```

If the receiver has not received a character within approximately 1 second, the program gives a message:

```
COMMUNICATION DEAD
```

While the program is running, it is possible to get the number of frames transmitted so far. This is done by pressing the space-bar on the keyboard.

If the communication is current-loop, remember to use the "blue box" (ND-252).

**Example:**

```
TPE>communication-test <CR>
Terminal device number     (oct): 1300 <CR>

Communication running
```

## DEFINE-DEVICE-TABLE

Parameter:<Yes/no for each present device>

This command does no testing on the interfaces. It finds which devices are present and asks if you want to test the device. If YES, the device number is entered into a table which might be used in tests.

The default answer is NO.

The device table is presented by the command PROGRAM-STATUS.

When executed under SINTRAN, this command will find all available logical devices and ask if they should be tested. "TERMINAL RESERVED FOR TEST-PROGRAM" is written on each terminal in the table. When the program is left (EXIT or loading another test program), the text "TERMINAL RELEASED" will be written on the terminals.

**Example 1:**

```
TPE>define-device-table <CR>

Test all present devices (YES or NO): Yes <CR>

TPE>
```

Now all present devices are entered into the device table, EXCEPT DEVICE 300.

**Example 2:**

```
TPE>define-device-table <CR>
Test all present devices (YES or NO): no <CR>

Test following terminal devices ?
   300  (YES or NO): no <CR>
   340  (YES or NO): yes <CR>
   350  (YES or NO): yes <CR>
   360  (YES or NO): yes <CR>
   370  (YES or NO): yes <CR>
  1300  (YES or NO): no <CR>
  1310  (YES or NO): no <CR>
  1320  (YES or NO): no <CR>
  1330  (YES or NO): no <CR>

TPE>
```

## DISPLAY-TEST

Parameter: <device number>

If no device table is defined, default is the console device number.

This command runs 5 different test-patterns on the screen.

```
Pattern 1: All characters in all positions.
Pattern 2: Alternating max. and min. lines
           (decreasing/increasing).
Pattern 3: All characters, one line of each.
Pattern 4: Alternating "M" and spaces.
Pattern 5: Alternating "U" and "*".
```

Patterns 4 and 5 will be held on the screen for about 5 seconds.

**Example:**

```
TPE>display-test <CR>
Terminal device number     (oct): 360 <CR>

ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
```

# FILL-SCREEN

Parameters: <device number> <character>.

If no device table is defined, default is the console device number. The default character is "H".

This command sends one character continuously. Any character can be specified.

**Example:**

```
TPE>fill-screen <CR>
Use device-table  (YES or NO): No <CR>
Terminal device number     (oct): 360 <CR>
Character                       : H <CR>
HHHHHHHHHHHHHHHHHHHH
```

# FIND-SPEED

Parameter: <use device-table (yes/no)> <device number>.

This command has no default value for the device number. It is illegal to specify the console device number.

This command cannot be used under SINTRAN.

The command finds the speed on the interface of the specified device. This is done by writing characters in test mode for 5 seconds. It counts how many characters are written and calculates the speed.

**Example:**

```
TPE>find-speed <CR>
Use device-table  (YES or NO): yes <CR>

Dev.No.: 1300
Speed on interface is 9600 bps.

Dev.No.: 1310
Speed on interface is 1200 bps.

Dev.No.: 1320
Speed on interface is 2400 bps.

Dev.No.: 1330
Speed on interface is  110 bps.

TPE>
```

# INTERFACE-TEST

This command cannot be run under SINTRAN.

The command tests the following interfaces:

Term buffer 1095
Four asynch 1122
Dual asynch 1147
Four terminal 3010
Eight terminal 3013
Eight terminal interface with FIFO   ND-273

This test writes test patterns to the interface output part and reads them back from the input part. The test patterns are:

000,177,152,125,000,125,152,177,000

It is checked for correct status word, ident-code, and that data is correctly transmitted and received. The speed is checked before running the pattern.

Data can be looped from the output part to the input part by using an external connection between output and input on each device. With this connection, even the line drivers of the interface are tested. It is also possible to loop data by setting the interface in maintenance mode. Data will then loop internally on the card, and no external connection is necessary. In the latter case, the line drivers will not be tested.

You can test the interfaces at one or more speeds. The default value is ALL-SPEEDS. Several speeds may be specified on the same line, separated by a comma or a space.

**Examples:**

```
TPE>interface-test <CR>
Use device-table  (YES or NO): Yes <CR>
Maintenance mode  (YES or NO): Yes <CR>
Speed                        : All-speeds <CR>


Testing Device No.:   1300

Tests are looping.


TPE>interface-test <CR>
Use device-table  (YES or NO): Yes <CR>
Maintenance mode  (YES or NO): Yes <CR>
Speed                        : 300,600,1200,2400,9600 <CR>


Testing Device No.:   1300

Tests are looping.
```

If the device table is not used, the program will ask for the number of lines per card and device number. If 8 lines per card is specified, the program also asks for the first device number in the second group.

**Examples:**

```
TPE>interface-test <CR>
No. lines pr. card              : 8 <CR>
First device number        (oct): 1300 <CR>
Device number second group (oct): 1350 <CR>
Maintenance mode  (YES or NO): Yes <CR>

Speed                           : 9600 <CR>

Testing Device No.:   1300 1310 1320 1330 1340 1350 1360
1370


TPE>interface-test <CR>
No. lines pr. card              : 4 <CR>
First device number        (oct): 1300 <CR>
Maintenance mode  (YES or NO): Yes <CR>
Speed                           : 1200   2400   4800 <CR>


Testing Device No.:   1300
Tests are looping.

TPE>
```

## KEYBOARD-TEST

Parameters: <device number> <hard copy terminal (yes/no)>.

In this test you type all the keys on the keyboard and check that they work correctly. Note that special characters and signs vary from terminal to terminal. The character on the screen should of course correspond to the character typed on the keyboard.

The program distinguishes between HARDCOPY and SCREEN terminals.

On HARDCOPY terminals, the program only checks the characters up to Z.

The program starts by displaying capital letters and ask you to enter them. After this, the same string of letters is displayed on the screen again, and you are requested to enter the string as control characters.

(Press the CTRL -key + the specified letter key).

Next it displays a string of special characters. (NOTE that this string starts with a space)

Finally the test will ask if you have a lower case keyboard. If YES, the displayed lower case letters must be typed.

If the device table has been defined, the program prints:

```
IF HERE TYPE ANY CHARACTER
```

on each device and runs the test on the first keyboard used. It then waits for you to type a character on the next keyboard to be tested. The console-device will list the devices that are tested.

When run under SINTRAN, it is only possible to test the console-keyboard, i.e. the terminal on which the program is run.

The test is stopped by pressing the ESC-key on the keyboard.

**Example:**

```
TPE>Keyboard <CR>

DO YOU HAVE HARDCOPY TERMINAL (Y/N) ? N <CR>

TYPE THE FOLLOWING STRING, UPPER-CASE CHARACTERS.

ABCDEFGHIJKLMNOPQRSTU......... ABCDE......
```

## LIST-TERMINAL-DEVICES

This command finds all the terminal devices present and presents them in a table.

To verify whether or not a terminal is connected, the program checks for a framing error (bit 5 in the status word). This can be reset by pressing Master Clear and then restarting the program (20!).

**Example:**

```
TPE>list-terminal-devices <CR>

-----------------------------------------------------------
!                 Terminal Devices present                !
-----------------------------------------------------------
!  300  !   340  !   350- !   360- !   370- !  1300-  !
-----------------------------------------------------------
! 1310- !  1320  !  1330  !        !        !         !
-----------------------------------------------------------
  -  Means that there is probably no terminal connected,
  B  Means buffered type interface

TPE>
```

# LOOP-TEST

Parameter: <device no>

This command is similar to the COMMUNICATION-TEST. The main difference is that it sends frames between different devices in the same system instead of checking communication between two systems.

Input and output may be on the same device number, but there must be an external connection between the input and the output channel. This can be done by using a special looping plug.

If no communication is established within 10 seconds, the following message appears:

```
NO REPLY, STILL TRYING.
```

After the first frame has been transmitted the program indicates:

```
COMMUNICATION RUNNING
```

If the receiver has not received a character within approximately 1 second, the program gives a message:

```
COMMUNICATION DEAD
```

While the program is running, it is possible to get the number of frames transmitted so far. This is done by pressing the space-bar on the keyboard.

If the communication is current-loop, remember to use the "blue box" (ND-252).

**Example:**

```
TPE>LOOP-TEST <CR>

OUTPUT DEVICE NO.:340 <CR>
INPUT DEVICE NO.:350 <CR>

COMMUNICATION RUNNING
```

## SET-PARAMETERS

Parameters: <loopmode> <abortmode> <suppress errors>

This command sets different modes, used by the other commands:

LOOPMODE - Tests will loop as many times as you specify. DEFAULT=Yes.
ABORTMODE - Tests are aborted after specified number of errors. DEFAULT=No.
SUPPRESS ERRORS - Tests are run without giving error messages. DEFAULT=No.

**Example:**

```
TPE>set-parameters <CR>
Loopmode          (YES or NO): Yes <CR>
How many times          (dec): Infinite <CR>
Abortmode         (YES or NO): No <CR>
Suppress errors   (YES or NO): No <CR>

TPE>
```

## SET-SPEED

Parameters: <device number> <speed>
Default value of the speed: 9600 baud.

This command cannot be run under SINTRAN.

This command sets (by software) the speed on the interface of the specified device, and then does a FIND-SPEED to check it. This takes about 5 seconds.

**Example:**

```
TPE>set-speed <CR>
Terminal device number     (oct): 1300 <CR>

Speed                           : 600 <CR>

Takes 5 sec. to check speed.
Speed set to 600 bps.

TPE>
```

# TRANSLATE

Parameters: <input type> <number>

This command gives a cross-reference between:

- terminal number
- hardware device number
- ident-code
- logical device number
- background program number
- thumbwheel setting on the interface card.

Terminal number, hardware device number, ident-code and logical device number can be input.
Default is a list of all numbers. This command gives output to the printer device.

**Example:**

```
TPE>translate <CR>
Input type                    : Logical <CR>
Logical-dev-no           (dec): 39 <CR>

---------------------------------------------------------------------
!Ter.No.!  Device  !  IDENT-  ! Logical Dev.No.!Back-  !Thumbwheel- !
!  Dec. !    No.   !   Code   !  Oct.     Dec. !ground !setting     !
---------------------------------------------------------------------
!    8  !    370   !      47  !      47     39 ! BAK08 ! 0       1  !
---------------------------------------------------------------------

Input type                    : Exit <CR>

TPE>
```

# VERIFY-TERMINAL-SPEED

Parameter: <device number>

This command finds the speed of the terminal and sets the interface to the same speed. Note that it is impossible to set the interfaces to the following speeds:

```
7200 bps
3600 bps
 900 bps
```

To find the speed, you must press the ESC-button on the specified terminal a few times, and the speed will be written on the specified terminal and the console.

Note that it is not possible to find the speed of the console device.

**Example:**

```
TPE>verify <CR>
Terminal device number     (oct): 1300 <CR>
```

Press ESC. on terminal 1300 to find terminal speed. Speed on the terminal is 9600 bps.

## PROGRAM-STATUS

This monitor command presents variables from the monitor, and also variables set by different program commands (DEFINE-DEVICE-TABLE, SET-PARAMETERS).

Note that the terminal device numbers are those defined by the command DEFINE-DEVICE-TABLE, and not all devises present in the system.

**Example:**

```
          TPE>program-status <CR>

          TPE PROGRAM STATUS: 1986.09.15 18:45:04

          TPE version..................: A01 - 1986-09-04
          Console device...............: 39D / 47B
          Printer device/mode..........: 39D / 47B / Normal
          Stop on full page............: Off
          Test Program file name.......: TER-ASYNC-FOO
          Loopmode.....................: Infinite
-----------------------------------------------------------
!                      DEVICE TABLE                       !
-----------------------------------------------------------
!   34 !     36 !     37 !     38 !     39 !     50 !
-----------------------------------------------------------
!   45 !     57 !     58 !     59 !        !        !
-----------------------------------------------------------
```

## 18.4 Error Messages

Error messages always appear with this heading:

```
*** ERROR *** DEVICE NO.: XXXX     TIME:    dd,hh:mm
```

An error will only be reported once. If a new error occurs, or there is a change in status, a new error message is written.

After an error, the latest read status will be printed and decoded. Some of the error messages also contain a number: n). This number refers to a note (See the monitor command

PRINT-NOTE for further information).

**Example:**

```
*** ERROR *** DEVICE NO.:  350     TIME:    10,54:30
WRONG IDENT CODE. FOUND: 0. EXPECTED 44  SEE NOTE 3)   LEVEL
STATUS: 004041
   DEVICE NOT READY.
   FRAMING ERROR                 SEE NOTE 4)
   CARRIER MISSING
```

# CHAPTER 19   UNIVERSAL-DMA / VICOM TEST PROGRAM

# CHAPTER 19 UNIVERSAL-DMA / VICOM TEST PROGRAM

This chapter describes the UNIVERSAL-DMA test program.

## 19.1 GENERAL

This test program will test the Universal DMA interface and its memory channel.

The most important commands are:

- INITIALIZE: This command is used to specify some test parameters used by the other tests.
- DMA-CHANNEL-TEST: This command is used to test the memory channel.
- SINGLE-VERIFICATION: This command is used when testing just one interface and requires special test equipment:

  1995 AA Universal DMA Test Card (Number 324455).

- DOUBLE-VERIFICATION: This command is used for a complete test of two cards and their memory channels and requires special test equipment:

  Test cable 325429 02 Rev. B (test no 1)<br>
  Test cable 325433 02 Rev. B (test no 2)

The other commands are used for debugging or initializing.

## 19.2 STARTING AND INITIAL ACTION

The UNIVERSAL-DMA test program is started from the TPE-MONITOR by the command:

```
TPE>Load Universal-DMA <CR>

UNIVERSAL-DMA - Version : x - yyyy-mm-dd
```

## 19.3 Program Commands

This section describes the different commands used in this test program.

## INITIALIZE

If you want to run a double-verification, you must answer N to the prompt

SINGLE TEST (YES/NO):

The program will then ask for a second device number. It checks that the device is present and verifies IDENT code.

It is also possible to move the buffer arrays in memory. When using multiport memory, the address of that area is not always the same when viewed from the CPU as when viewed from Universal-DMA, and must be specified separately.

**Example:**

```
TPE>INITIALIZE<CR>
Single test (yes/no): Y<CR>

First device number: (oct.): 140050 <CR>

Do you want to move the buffer area (yes/no): Y

First physical buffer page (100 to 177777oct): 100 <CR>

TPE>
```

## DMA-CHANNEL-TEST

This command runs two tests, Test 1 and Test 2. During the test-run, 5 different commands are available:

B : Back to start of same test.
N : Next test (only in Test 1).
P : Previous test (only in Test 2).
R : Restart = Jump to Monitor.
C : Continue next pattern (in Test 2, only after data error).

When you have given the command DMA-CHANNEL-TEST, you must answer some questions from the program:

**Example:**

```
TPE>DMA-CHANNEL-TEST<CR>

Do you want test to loop (yes/no): Y<CR>
Do you want to loop on error (yes/no): Y<CR>

Addresses tested by CPU (yes/no): Y<CR>

32-bit data channel (16 bit=no) (yes/no): N<CR>

Set length of test array (oct.): 3777<CR>

Clear test array before run (Yes/no): Y<CR>

       End of test 1

       End of test 2

Tests are looping.

ESC

== escape ==
TPE>
```

# SINGLE-VERIFICATION

When you have given the command SINGLE-VERIFICATION, you must answer some questions:

DO YOU WANT TO RESET DATA ARRAY BEFORE RUN  (YES/NO):

Reset of data array allows operator to reset all counters in all options.

DO YOU WANT TEST TO LOOP   (YES/NO):

Loop mode means that test shall loop and not stop before a stop is forced.

DO YOU WANT MESSAGES WRITTEN ON TERMINAL   (YES/NO):

If answer is NO, the terminal will be silent. YES will cause messages to be written.

DO YOU WANT CONTINOUS OUTPUT MODE   (YES/NO):

Continuous output mode will make the test continue after the message is written. "N" will make test stop and wait for a new command .

OPTION TESTS:

You can answer ALL-TESTS, or you can choose between eight different options:

| | |
|---|---|
| Option No.: 1 | Test of Core Address Register |
| Option No.: 2 | Test of Control/Status Registers |
| Option No.: 3 | Test of Data Channel |
| Option No.: 4 | Test of DMA Output |
| Option No.: 5 | Test of DMA Input |
| Option No.: 6 | Test of Word Counter |
| Option No.: 7 | All options 1:6 |
| Option No.: 8 | Output of status of all tests after run |

These three commands are available:

| | |
|---|---|
| "ESCAPE" | : Test is aborted and you may choose new test |
| "CR" | : Test continues |
| "A" | : Test is aborted and status of test aborted is available |

Errors are reported in the following way...:

| MESSAGE | ERROR VALUE | RIGHT VALUE | OPTION NO |
|---|---|---|---|

**Example:**

```
TPE>SINGLE-VERIFICATION<CR>

Do you want to reset data array before run  (yes/no): Y <CR>

Do you want test to loop   (yes/no): Y <CR>

Do you want messages written on terminal   (yes/no): Y <CR>

Do you want continuous output mode   (yes/no): Y <CR>
Option tests: ALL-TESTS<CR>
   Option No.     1  Completed  000000 000001 Times

   Option No.     2  Completed  000000 000001 Times

   Option No.     3  Completed  000000 000001 Times

   Option No.     4  Completed  000000 000001 Times

   Option No.     5  Completed  000000 000001 Times

   Option No.     6  Completed  000000 000001 Times

Test(s) are now looping.
Nothing but error messages is written.
 ESC

TPE>
```

# DOUBLE-VERIFICATION

After you have given the command DOUBLE-VERIFICATION, you must answer some questions:

DO YOU WANT TO RESET DATA ARRAY BEFORE RUN (YES/NO):

Reset of data array allows operator to reset all counters in all options.

DO YOU WANT TEST TO LOOP (YES/NO):

Loop mode means that test shall loop and not stop before a stop is forced.

DO YOU WANT MESSAGES WRITTEN ON TERMINAL (YES/NO):

If answer is NO, the terminal will be silent. YES will cause messages to be written.

DO YOU WANT CONTINOUS OUTPUT MODE (YES/NO):

Continuous output mode will make the test continue after the message is written. "N" will make test stop and wait for a new command .

DO YOU WANT TO LOOK FOR ERRORS IN DMA DATA ARRAY (YES/NO)

You may also have the opportunity to look at errors occurring in both output and input array in connection with DMA.

OPTION TESTS:

You can answer ALL-TESTS, or you can choose from five different options:

| | |
|---|---|
| Option No.: 1 | Test of Control/Status. |
| Option No.: 2 | Test of DMA Input from Device 2.<br>Device 1 as Output from memory. |
| Option No.: 3 | Test of DMA Input from Device 1.<br>Device 2 as Output from memory. |
| Option No.: 4 | All options 1:3. |

Option No.: 5 Status Output of all options. These three commands are available:

```
"ESCAPE" : Command is aborted.
"CR"     : Test continues.
"A"      : Option currently running is aborted.
```

The error format is usually as follows :

**MESSAGE      ERROR VALUE   RIGHT VALUE   OPTION NO**

An additional error format is used in options 2 and 3.

**MESSAGE      ADDRESS   ERROR VALUE   RIGHT VALUE**

**Example:**

```
TPE>DOUBLE-VERIFICATION <CR>

Do you want to reset data array before run   (yes/no): Y<CR>

Do you want test to loop   (yes/no): N<CR>

Do you want messages written on terminal   (yes/no): Y<CR>

Do you want continuous output mode   (yes/no): Y<CR>

Do you want to look for errors in DMA data array  (yes/no): Y<CR>
Option tests: ALL-OPTIONS<CR>
     Option No 1 completed 000000 000001 times.
     Option No 2 completed 000000 000001 times.
     Option No 3 completed 000000 000001 times.
Tests are now looping.
Nothing but error messages are given on terminal.
 ESC
TPE>
```

# IORUN

IORUN is a general-purpose single address IOX exerciser. When ready to use it prints an asterisk (*). Then four possible commands are available:

```
    H   = Print explanation
  (ESC) = Go back to TPE-Mon
(SPACE) = Go and ask for mode with last used type of IOX.
DN+XX   = (Where X is a number between 0 and 77) this number
          is added to the device number written when TPE-Mon
          started. If this number is odd (i.e. output), it
          also asks for data to be used.
```

When it asks for mode(as Mode:), four different commands are available:

```
     B   = Return to asterisk-condition.
   (ESC) = Go back to TPE-Monitor.
     C   = Start executing specified IOX instruction.
ZXXXXX   = (Where XXXXX is an octal number between 0 and
           77777) Burst execution. It executes so many IOX-
           instructions. When finished, it prints a +-sign
           and waits for new mode. If the number is 1 and the
           IOX is an input, the read data is written on the
           terminal.
(SPACE)  = Will generate another burst of IOX.
```

If C is pressed, it can be stopped with an S, and started again as many times as desired.

**Example:**

```
*DN+4 <CR>
Mode:Z10 <CR>
+B <CR>

*DN+1 <CR>
Data: (oct):0 <CR>
Mode:Z10 <CR>
+B <CR>

*
```

# HANDEX

HANDEX will exercise the handshake logic in test and PIO mode. No DMA.

# DMA-EXERCISE

This program will exercise the DMA and handshake logic. It is intended for scope debugging of a Universal DMA channel. The results are not checked.

When the program has written an asterisk (*), it is waiting for one of the following commands:

```
   (ESCAPE)  :  Return to Monitor
     H       :  Write explanation
    (CR)     :  Start DMA after 3 questions has been
                answered
```

It can be stopped by typing B, and then it returns to the asterisk mode. If the test mode is not selected, you must use a special connection plug to connect REQUEST and COMPLETE.The test plug needed for SINGLE-VERIFICATION will do.

## DUAL-HANDEX

This program will exercise the handshake logic between two cards. Test mode is not used.

# Chapter 20 COLOUR-TERM-A00

# CHAPTER 20 COLOUR-TERM-A00

## 20.1 GENERAL

This program is based on the earlier TNCT program, which tested the NCT (Nord Colour Terminal). The tests are for the most equal to those in the old program, TNCT.

The Colour-term program is also able to test the newer CTI (Colour Terminal Interface), which is by and large compatible with the old NCT terminal.

## 20.2 STARTING AND INITIAL ACTION

- Under SINTRAN.

  The program will ask for file name for the colour terminal.

- Stand alone.

  If only one CTI device is present, the program will automaticly select this one. If several or no CTI interfaces are present, you will be asked for hardware device number instead. This is specified as in the command SELECT-DEVICE.

## 20.3 PROGRAM COMMANDS

### FUNCTION-TEST

This test checks the control-functions on the colour terminal, and is therefore an important initial test. The test is self-explanatory.

## BUFFER-TEST

Symbols will be displayed as magnified symbols, made of a collection of the symbol itself.

32 symbols are displayed in one picture, the next 32 will be displayed each time any key is pressed until all 256 are shown.

If loopmode, pictures are shown continously.

## COMMUNICATION-TEST

This is a very useful test to run if communication is suspected for errors. These 4 sections are executed sequently:

```
-TEST 1 : TEST IF UN-EXPECTED INPUT(S)
-TEST 2 : TEST IF RESPONSE ON ENQUERY (005)
-TEST 3 : TEST VALUE OF RESPONSE
-TEST 4 : TEST LINE/INTERFACE QUALITY
```

TEST 4 transmits blocks of 256 bytes to the colour terminal, and then the result of ENQUERY is tested.

Normally 40 blocks are transmitted before the result printout.

If loopmode, the test will go on until you press the <CR> key on the keyboard. Then the results will be printed. If you want a result printout without breaking the test, type any other key.

## TRIM-PATTERN

Writes a square-pattern for geometry adjustment.

## SCALE-VERIFICATION <COLOUR>

This test is useful for trimming/adjustments of the monitor and the colour generator output. It displays a scale that should indicate the linearity on the DACs of the colour generator.

## KEYBOARD-TEST

A picture of the NCT/CTI keyboard is drawn at the colour terminal. Touching an active key shall alter the corresponding mark, and the detected key code is shown.

It is possible by this test to perform a verification of all active keys on the NCT/CTI keyboard without knowledge about the expected value of the codes. An active key means a key giving a code directly.

**Example:**

| | |
|---|---|
| - Active keys: | A,B,C,ESC etc. |
| - Non-active keys: | SHIFT,CTRL etc. |

Note that some codes may come from two keys(double-marked)!

## COPY-CHARACTER-TO-CTI <FROM KEYBOARD>

This function simply generate codes to the colour-terminal.

The code may be given either from console keyboard or from colour-terminal keyboard

## COPY-VALUE-TO-CTI

This function will simply generate codes to the colour-terminal. The code must be given as octal values from the console keyboard.

If loopmode, the code will be repeated until any console key is typed.

## SET-COLOUR-TERMINAL-TYPE <TYPE>

Select the colour terminal type. Two types are available: NCT and CTI.

## SET-PARAMETERS <LOOPMODE>

This command will allow you to set / reset loopmode.

When loopmode is set, some of the tests will run until you stop them.

The tests affected by the mode-setting are:

COMMUNICATION-TEST
BUFFER-TEST
COPY-VALUE-TO-COLOUR-TERMINAL

For further information, take a look at the explanation of the actual test.

## SELECT-DEVICE <DEVICE NO / FILE NAME>

When running under SINTRAN a periferial file must be specified. When running stand alone the hardware device no. must be specified.

## LIST-PRESENT-CTI-DEVICES

The hardware CTI device numbers present in the machine will be listed.

# Chapter 21 Stand-alone Utility Programs

# CHAPTER 21 STAND-ALONE UTILITY PROGRAMS

This chapter describes these stand-alone utility programs:

- FILESYSTEM INVESTIGATOR
- DIR-BACKUP
- MCOPY-TANB

The programs are found on the floppy

"SINTRAN Utility Programs" - ND-210628.

## 21.1 THE FILE SYSTEM INVESTIGATOR

The File System Investigator can be used to check that there are no serious errors in the file system. The check takes only 5 minutes on a 75 Mbyte disk. The following Disk-types can be used:

DISC-38MB-1
DISC-38MB-2
DISC-75MB-1
DISC-75MB-2
DISC-288MB-1-R
DISC-288MB-2-R
DISC-30MB-1
DISC-60MB-1
DISC-90MB-1
DISC-30MB-2
DISC-60MB-2
DISC-90MB-2
DISC-10MB-1
DISC-10MB-2
DISC-33MB-1
DISC-33MB-2
DISC-66MB-1
DISC-66MB-2
DISC-3-75MB-1
DISC-3-75MB-2
DISC-2-75MB-1
DISC-2-75MB-2
DISC-21MB-1
DISC-21MB-2
DISC-14MB-1
DISC-14MB-2
FLOPPY-DISC-1
FLOPPY-DISC-1
DISC-45MB-1
DISC-45MB-2
DISC-23MB-1
DISC-23MB-2
DISC-4-70MB-1-R
DISC-4-70MB-2-R
DISC-70MB-1
DISC-70MB-2
DISC-2-70MB-1-F
DISC-2-70MB-2-F
DISC-140MB-1-F
DISC-140MB-2-F
DISC-4-70MB-1-F
DISC-4-70MB-2-F
DISC-288MB-1-F
DISC-288MB-2-F
DISC-6-70MB-1-F
DISC-6-70MB-2-F
DISC-2-225MB-1-F
DISC-2-225MB-2-F
DISC-450MB-1-F
DISC-450MB-2-F
DISC-225MB-1-R
DISC-225MB-2-R
DISC-16MB-1
DISC-16MB-2
DISC-74MB-1
DISC-74MB-2
DISC-28MB-1
DISC-28MB-2
DISC-288MB-1-E
DISC-288MB-2-E
DISC-4-70MB-1-E
DISC-4-70MB-2-E
DISC-6-70MB-1-N
DISC-6-70MB-2-N
DISC-2-225MB-1-N
DISC-2-225MB-2-N
DISC-450MB-1-N
DISC-450MB-2-N

**The Procedure:**

- Put the floppy disk with the stand-alone program in the floppy disk unit 0.
- Type 1560& on the console to load the floppy disk monitor, and an asterisk (*) appears on the screen.
- Start the FILE SYSTEM INVESTIGATOR as shown in the following example:

**Example:**

```
* Load Fiesys <CR>
. . .

DEVICE NAME: Disc-75MB-1 <CR>
UNIT: 0 <CR>
```

Information about the available commands can be obtained by typing:

```
>Help <CR>
```

Execute the three following commands for checking of the consistency of the directory:

```
>Dump-directory-entry <CR>
. .

>List-users <CR>
. .

>Page-list <CR>
. .

E
```

Errors will be listed, and the output will describe the type of error and where it is located.

This listing may be directed to a line printer by using this command:

```
>Set-printer-device-number <CR>


DEVICE NO.: 430 <CR>
>
```

**NOTE:** Information about different variables used in the program, such as the DEVICE NUMBER, can be obtained by using the command:

```
>Program-status <CR>
```

## Serious Error Messages

Serious error messages are:

- Page conflict (Refer System Supervisor, Chapt 10.3.2)
- Serious error in bit file (Refer System Supervisor, Chapt 10.3.2)

You can correct these errors by:

- @Test-directory <CR> - corrects bit-file errors
- @Regenerate-dir <CR> - corrects error in object/user/directory entry or bit-file

**WARNING:** Do not use the commands TEST and REGENERATE if you get the error messages

PAGE-CONFLICT
or OUTSIDE-DEVICE-LIMITS

Corrections of these errors must be done by using the command CHANGE in DISK-TEMA. This requires knowledge about the internal structure of the filesystem.

## 21.2 DIR-BACKUP

This program can be used for backup of all the Winchester disks. Each streamer cartridge can store 45 Mb of data, so you need more than one cartridge for the backup if the disk is large (e.g. 74 Mb).

After the backup, the first part of the tape will contain information about:

- the backup date
- which disk unit the backup was copied from
- the number of pages
- CPU-information and the version of the backup system.

The following disks can be used:

```
DISC-14MB-1
DISC-14MB-2
DISC-16MB-1
DISC-16MB-2
DISC-21MB-1
DISC-21MB-2
DISC-23MB-1
DISC-23MB-2
DISC-28MB-1
DISC-28MB-2
DISC-45MB-1
DISC-45MB-2
DISC-74MB-1
DISC-74MB-2
```

**COPYING FROM A DISK TO A STREAMER:**

Insert the floppy disk containing the backup program into FLOPPY-DISC-1, UNIT 0 and load the program:

```
##1560& <CR>
*Dir-backup <CR>

========================================================
=                Directory backup for                  =
=             Winchester-disc and Streamer             =
=                                                      =
=    Time and date:  day/Month-Year      hh:mm:ss      =
========================================================

   DIR-BACKUP - Version : C01 - OCTOBER 16, 1985

The command HELP gives you a list of the commands


Now you must insert the cartridge into the streamer drive.
```

**NOTE:** Remember that each backup requires one cartridge, except 74 Mb discs which need two cartridges. You are asked to change streamer Cartridge when needed.

If you wish to change the default values of the program, you should first execute the MODE command as shown below:

**Example (default values):**

```
>Mode <CR>


Running on a screen terminal.......(Yes/No) : Y <CR>
Restart SINTRAN after backup.......(Yes/No) : N <CR>
Compare after copy.................(Yes/No) : Y <CR>
Abort when compare error...........(Yes/No) : Y <CR>
Disable ESC function...............(Yes/No) : N <CR>
System name or number..............(Yes/No) : <CR><CR>
```

## THE BACKUP COMMAND

This command is for backup from Winchester disk (ST506) to streamer.

```
>Backup <CR>
TO
STREAMER system   (1-2 Oct.) 1 <CR>
Unit................(0-3 Oct.) 0 <CR>

FROM
DISC-NAME.......................: Disc-45MB-1 <CR>
```

COMMENT
If you want a list of the disk types,
you may just write Help <CR> here.

```
Unit..................(0-3 Oct.) 0 <CR>

==  hh:mm:ss    Initialize disc and streamer  ==
```

COMMENT
Takes 3-4 minutes.

```
==  hh:mm:ss    Copy from disc to streamer    ==
```

COMMENT
Takes 10-12 minutes.

```
Current Page: xxxxxx
```

COMMENT
Current page (Oct.) beeing copied.

The program will inform you when the copying is finished, and the compare procedure will start:

```
==  hh:mm:ss           End of volume          ==

==  hh:mm:ss    Compare disc and streamer     ==

==  hh:mm:ss           End of volume          ==

Ready with next streamer tape (Yes/No): Y <CR>

==  hh:mm:ss    Initialize disc and streamer  == >
```

If the COMPARE AFTER COPY function in the MODE command is selected, the backup of a 45 Mb disk takes about 20 minutes.

If the backup was not successful, an error message will be written on the terminal.

## The RECOVER command

This command is for restoring the backup from streamer to disk.

```
>Recover <CR>

RECOVER
TO
DISC-NAME...................................:Disc-45MB-1 <CR>
DISC-45MB-1

Unit............(0-3 Oct.) 0 <CR>

FROM
STREAMER system (1-2 Oct.): 1 <CR>

Unit............(0-3 Oct.) 0 <CR>

==  hh:mm:ss    Initialize disc and streamer  ==

==BAC08=> This backup is recorded day/month-year  hh:mm
========> From DISC-xxMB-x Unit o System TEST
========> This tape is number : 1, Starting at page number : 000000
========> Backup done with DIR-BACKUP-C01

==  hh:mm:ss  Copy TO DISC from streamer      ==

==  hh:mm:ss          End of volume           ==

==  hh:mm:ss  Compare disc and streamer       ==

==  hh:mm:ss          End of volume           ==
```

## 21.3 MCOPY

MCOPY is a program for copying between disks and magnetic tapes. This program copies directories on disk to and from these magtape drives:

- Cipher
- Tandberg
- Pertec
- STC

The following discs can be used:

DISC-75MB-1
DISC-75MB-2
DISC-288MB-1-R
DISC-288MB-2-R
DISC-38MB-1
DISC-38MB-2
DISC-30MB-1
DISC-60MB-1
DISC-90MB-1
DISC-30MB-2
DISC-60MB-2
DISC-90MB-2
DISC-2-75MB-1
DISC-2-75MB-2
DISC-3-75MB-1
DISC-3-75MB-2
DISC-10MB-1
DISC-10MB-2
DISC-33MB-1
DISC-33MB-2
DISC-66MB-1
DISC-66MB-2
DISC-14MB-1
DISC-14MB-2
DISC-21MB-1
DISC-21MB-2
DISC-45MB-1
DISC-45MB-2
DISC-23MB-1
DISC-23MB-2
DISC-4-70MB-1-R
DISC-4-70MB-1-R
DISC-70MB-1
DISC-70MB-2
DISC-2-70MB-1-F
DISC-2-70MB-2-F
DISC-140MB-1-F
DISC-140MB-2-F
DISC-4-70MB-1-F
DISC-4-70MB-2-F
DISC-288MB-1-F
DISC-288MB-2-F
DISC-6-70MB-1-F
DISC-6-70MB-2-F
DISC-2-225MB-1-F
DISC-2-225MB-2-F
DISC-450MB-1-F
DISC-450MB-2-F
DISC-225MB-1-R
DISC-225MB-2-R
DISC-74MB-1
DISC-74MB-2
DISC-28MB-1
DISC-28MB-2
DISC-288MB-1-E
DISC-288MB-2-E
DISC-4-70MB-1-E
DISC-4-70MB-2-E
DISC-6-70MB-1-N
DISC-6-70MB-2-N
DISC-2-225MB-1-N
DISC-2-225MB-2-N
DISC-450MB-1-N
DISC-450MB-2-N

**NOTE:**

A backup on mag-tape must be copied back to the same type of disk that it was copied from.

**Example of using MCOPY:**

```
#MACL <CR>
##1560&
*Load MCOPY <CR>
MAG TAPE - DISK COPY, HUT 1649POD. ISSUED: AUG 20, 1986
THE COMMAND HELP GIVES YOU A LIST OF THE COMMANDS

>LIST-SPECIAL-COMMANDS <CR>

COPY-TO-MAG-TAPE
COPY-FROM-MAG-TAPE
COMPARE-DISK-TAPE
SINTRAN-BLOCK-SIZE
SET-MAG-TAPE-DEVICE-NUMBER
1600-BPI
SYSTEM-COPY
AUTOMATIC-COMPARE
SET-DISK-TYPE
CHANGE-MEMORY-BUFFER-ADDRESS
```

Backup produced on mag-tape by BACKUP-SYSTEM with the command D EVICE-COPY, can be read back by MCOPY using the command COPY-FROM-MAG-TAPE.

## COPY/COMPARE

Three of these commands are used to copy or compare:

```
COPY-TO-MAG-TAPE
COPY-FROM-MAG-TAPE
COMPARE-DISK-TAPE
```

These commands has a set of parameters:

- The disk type, if not specified before

```
DISK NAME:  Disc-75MB-1 <CR>
```

- If you use one of the following disk types:

```
38/75/288/30/60/90 Mb
```

  you will be asked whether you want the old or new directory size:

```
DIRECTORIES CREATED BY SINTRAN VERSION E OR LATER
ARE A FEW PAGES SMALLER THAN THOSE CREATED BY
VERSIONS A, B, C, D. DO YOU WANT THE NEW SIZE
(VERSION E OR LATER) OR THE OLD SIZE (VERSION D OR
EARLIER)?
PLEASE ANSWER OLD OR NEW : new <CR>
```

- The unit number:

```
DISK UNIT /DECIMAL, 0-3) : 0 <CR>
```

- If your disk is one with multiple directories on the same unit, for example 10 Mb, 30/60/90 Mb, 2-75 Mb, 3-75 Mb, you will be asked one of the following questions, depending on whether the system copy is turned on or off. If the system copy is on, you get the prompt:

```
REMOVABLE OR FIXED : Fixed <CR>
SUBUNIT (DECIMAL, 0-2) :1 <CR>
```

  The first prompt refers to the directory that is to be copied. It can be either REMOVABLE or FIXED for 10 Mb and 30/60/90 Mb.

  The second prompt asks for the subunit and applies to disks with more than one subunit (for example 60/90 Mb, 2-75 Mb, 3-75 Mb).

- Next question is about the magnetic tape unit:

```
MAG-TAPE UNIT (DECIMAL, 0-3) : 2 <CR>
```

- If the system copy is turned off, and your disk type is one with multiple directories on the same unit, the program asks for the file number on the tape:

  ```
  MAG-TAPE FILE NUMBER (DECIMAL, 0-99): 2 <CR>
  ```

  If you have copied several directories to one tape by giving file number, it makes it easier to select one specific directory later if you want to copy it back to a disk.

  Remember that the first file number on the mag-tape is file number 0.

**Starting the COPY/COMPARE:**

When you have answered all the prompts, the program asks you:

```
OK?
```

If you answer YES, the COPY/COMPARE operation will start.

## Changing Disk Type and Mag-tape Device Number

The commands are:

- SET-MAG-TAPE-DEVICE-NUMBER
- SET-DISK-TYPE

When a COPY/COMPARE command is given, the program checks if the disk type has already been specified, for example in connection with an earlier COPY/COMPARE operation.

If this is the case, the program will continue to use this disk type until it is changed by using the command:

```
>Set-disk-type <CR>
DISK NAME : Disk-2-75MB-1 <CR>
```

When the program is started, it assumes that the mag-tape drive is connected to mag-tape controller 1, device number 520 octal. This may be changed by giving the command:

```
>Set-mag-tape-device-number <CR>
MAG-TAPE DEVICE NUMBER (OCTAL, 520-530) :530 <CR>
```

Mag-tape drives connected to mag-tape controller 2 have device number 530 octal.

## Changing Modes of the Program

The commands are:

- SYSTEM-COPY
- AUTOMATIC-COMPARE
- 1600-BPI
- SINTRAN-BLOCK-SIZE

These commands are used to turn ON/OFF (set/reset) special features of the program. The relevant feature (mode) is either turned off or on, depending on what state it was in previously. When such a command is given, the program will answer whether the relevant feature is turned on or off.

**Example:**

```
>System-copy <CR>
SYSTEM COPY TURNED ON
```

**AUTOMATIC COMPARE:** When the program is started, automatic compare is turned on. This means that after each copy operation, the program will automatically do a compare. When copying directories occupying multiple reels of tape (for example a 75 Mb directory on 1600 BPI), the program will compare each reel of tape after it has been copied. You thus avoid the trouble of having to mount each reel of tape twice.

**WARNING:** NOTE that the automatic compare can be turned OFF, but then do not know if the backup was successful:

```
>AUTOMATIC-COMPARE <CR>
 AUTOMATIC COMPARE TURNED OFF
```

**SYSTEM COPY** With disk types having multiple directories on the same unit, it is possible to copy/compare multiple directories by using the command SYSTEM-COPY.

This mode is initially turned off. It can be turned on by giving the command:

```
>System-copy <CR>
SYSTEM COPY TURNED ON
```

**SETTING THE BPI MODE:** The STC mag-tape drive may record data on magnetic tapes in either the 1600 BPI or the 6250 BPI mode. The mode is initially set in 6250 BPI, but this may be changed by giving the command:

```
>1600-BPI <CR>
1600 BPI TURNED ON
```

Recording the data in 1600 BPI mode makes it possible to read the tape on the Pertec mag-tape drive.

If you have a Cipher mag-tape drive, recording may be done in a 3200 BPI 1600 BPI mode. The mode however must be set manually by the user.

**SETTING SINTRAN BLOCK SIZE:**

The record size on tape is usually set at more than 1K (1024) 16 bit words. This enables a more efficient use of the tape. The command SINTRAN-BLOCK-SIZE can however be used to set the record size on tapes to exactly one page each. The purpose of this command is to enable the user to copy from tape to disk while SINTRAN is running. The SINTRAN command COPY-DEVICE is used for such copying.

SINTRAN block size is initially turned off.

**WARNING:**

It is not possible to recover directories that occupy more than one reel of tape when using the SINTRAN command COPY-DEVICE. Note also that setting record size to one page leads to inefficient utilization of the tape. In addition, it also makes the copy/compare operation slower. The command SINTRAN-BLOCK-SIZE should therefore only be used if necessary.

**Example:**

```
>Sintran-Block-Size <CR>
SINTRAN BLOCK SIZE TURNED ON
```

# Index

## UPDATING

Manuals can be updated in two ways, new versions and revisions. New versions consist of a completely new manual which replaces the old one, and incorporate all revisions since the previous version. Revisions consist of one or more single pages to be merged into the manual by the user, each revised page being listed on the new printing record sent out with the revision. The old printing record should be replaced by the new one.

New versions and revisions are announced in the ND Customer Support Information and can be ordered from the address below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and give an evaluation of the manual. Both detailed and general comments are welcome.

| PRINTING RECORD | |
|---|---|
| PRINTING | NOTES |
| 03/83 | Preliminary version |
| 04/83 | Version 2 EN |
| 03/87 | Version 3 EN |
| | |
| | |
| | |
| | |
| | |

Test Program Description for ND-100/ND-110

Publ.No. ND-30.005.3 EN

## RING BINDER OR PLASTIC COVER

The manual can be placed in a ring binder for greater protection and convenience of use. Ring binders may be ordered at a price of NKr. 45.- per binder.

The manual may also be placed in a plastic cover. This cover is more suitable for manuals of less than 100 pages than for larger manuals.

Please send your order, as well as all types of inquiries and requests for documentation to the local ND office, or (in Norway) to:

Graphic Center
Norsk Data A.S
P.O.Box 25 BOGERUD
N-0621 OSLO 6 - Norway

I would like to order

........ Ring Binders, B5, at NOK 35,- per binder

........ Ring Binders, A4, at NOK 45.- per binder

........ Plastic Covers, A4, at NOK 10.- per cover

Name: ..........................................................................

Company: ....................................................................

Address: ......................................................................

## *************** SEND US YOUR COMMENTS!!! ***************

Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card — and an answer to your comments.

Please let us know if you
- find errors
- cannot understand information
- cannot find information
- find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!

## ************ HELP YOURSELF BY HELPING US!! ************

Manual name: Test Program Description for ND-100/ND-110 Manual number: ND-30.005.3 EN

What problems do you have? (use extra pages if needed) ____________

____________

____________

____________

____________

Do you have suggestions for improving this manual ? ____________

____________

____________

____________

____________

____________

Your name: ____________ Date: ____________

Company: ____________ Position: ____________

Address: ____________

____________

What are you using this manual for ? ____________

____________

**NOTE!**
This form is primarily for documentation errors. Software and system errors should be reported on Customer System Reports.

**Send to:**
Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

Norsk Data's answer will be found on reverse side

Answer from Norsk Data ______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

Answered by ______________________________ Date ______________

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

**Norsk Data A.S**
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo6, Norway

ND