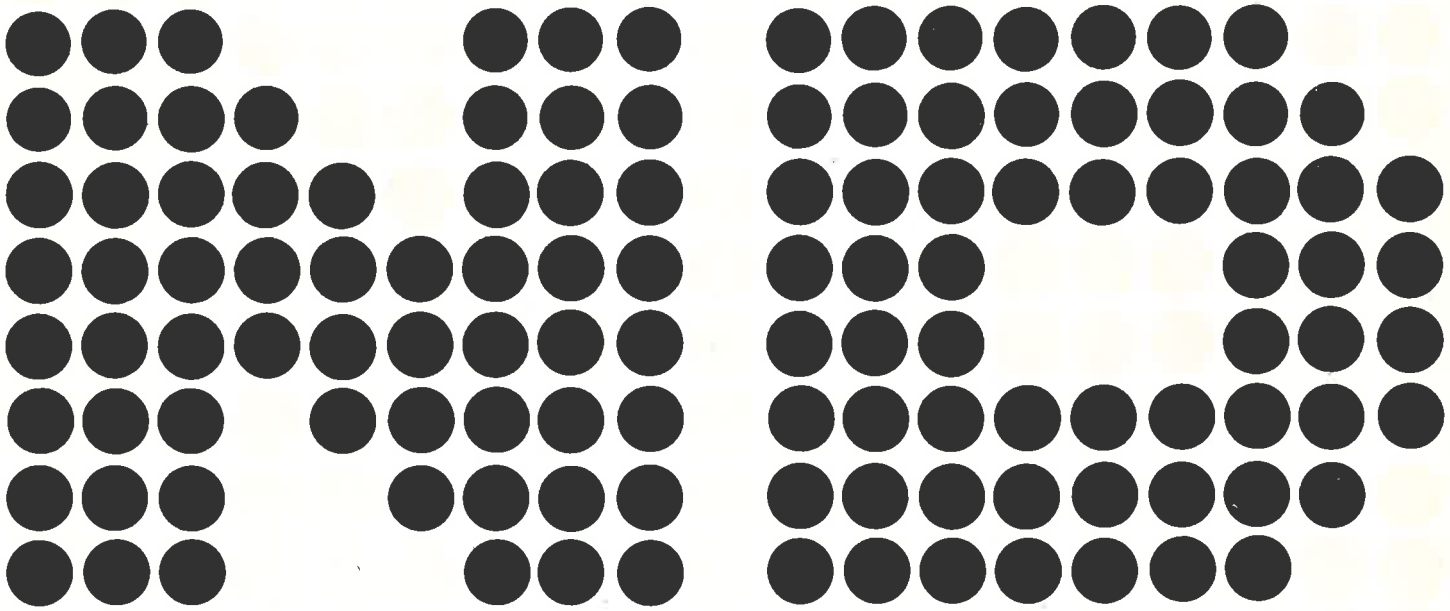
DCHT

Data Channel Test Program

NORSK DATA A.S



DCHT

Data Channel Test Program

User's Manual for

Nord-10 Data Channel Test Program

(DCHT)

REVISION RECORD

Revision	Notes
5/74	Original Printing

Publ. No.
ND-62.005.01
May 1974

A/S NORSK DATA ELEKTRONIKK
Lørenvn. 57, Oslo 5, Tlf. 21 73 71

PURPOSE

This manual describes a test program for the NORD-10 Data Channel. The test program can run many DMA (Direct Memory Access) transfers simultaneously. Several modes may be selected for different load characteristics.

The information in this manual will be of interest to NORD-10 maintenance personnel and persons maintaining the program.

--oo0oo--

TABLE OF CONTENTS

--ooOoo--

1 INTRODUCTION

The program is designed to test the data channel when one or more devices access the NORD-10 memory together with the CPU. Any type of DMA device may be included, but the present version only includes test drivers for disks, drums and plotters.

Before this program is run, all devices should be able to operate alone without errors. Special programs such as TECODR are available for testing of devices.

The program may run in three different modes, each placing different load characteristics on the channel.

The devices may be run in a normal mode, or the interfaces may run in test mode.

The basic version of the program has tables for:

- 4 disks
- 4 drums
- 4 magnetic tapes
- 2 Versatec plotters
- 1 CAMAC interface

These devices can be connected to any number of controllers, but the program only handles one active unit per controller (drum controllers take only one unit).

A given device will normally run through 'write and read' cycles. The core buffer is initialized with some predetermined pattern, normally a 'address in address' type. The buffer is then written onto the device and then read back. When read back, the data is checked. A plotter must be visually examined to detect data errors. As this is a write-only device, the program can not check the data by reading it.

When the interfaces run in test mode, they will place a known pattern in core and this can be checked. This data pattern is also sent to the device during the write cycle and checked in hardware. (Compare function.)

A mode exists so that when an error occurs, the failing device will freeze the normal write/read cycle and only repeat the failing operation. If this mode is not set, the device will be cleared and a write operation initiated each time an error occurs.

The test program takes full use of the CPU. All internal interrupts are enabled and if errors such as memory-out-of-range or memory parity occurs, they will be reported.

For related information refer to the manuals for each device in question and to the NORD-10 Reference Manual.

2 THEORY OF OPERATION

The program normally performs a write operation followed by a read operation on all active devices. This may be performed with the real device or with the interface in test mode.

A small command monitor accepts commands that control the program and the devices. These commands are treated in Chapter 3. This chapter devotes itself to the operation of the program.

The program may be run in three operational modes, each giving a different load characteristic on the data channel. (For this to be true, at least two devices must be active.) These modes are explained in Section 2.2.

Extensive use has been made of the NORD-10 interrupt system. It may prove useful to explain the tasks allocated to each program level in use.

2.1 Program Level Allocation

The following will briefly describe how the test program uses the different program levels. In all, twelve levels are used. These will be treated with descending (program) priority.

Level 14, Internal Interrupts

This level traps the internal interrupts. If an interrupt appears, (in this program this means that some error has occurred) it is identified and the following data is put in the 6TASK buffer:

- IIC - code
- PVL - code
- P - register on offending level
- T - register on level 14 (this contains the MON number if it was a monitor call)

Level 14 then schedules level 6 by setting PID (6). If the 6TASK buffer overflows, level 14 will suspend itself by clearing PIE (14).

Level 13, Clock Interrupts

This is the clock level. Basic clock period is 2 ms. Each 100 ms level 13 schedules level 9 by setting PID (9).

Level 12, TTY Input Interrupts

This level processes TTY input. Characters read from the TTY are put in the INB buffer and the monitor level is scheduled by setting PID (4). If the 'ESC' character (ASCII 33) is encountered, a program restart is initiated.

Level 11, Mass Storage Interrupts

This level traps interrupts from the DMA devices. If the program runs in interrupt mode (MODE = 1 or 2), the IDENT code is put in the 3TASK buffer and level 1 is scheduled by setting PID (1). If the program is in scanning mode (MODE = 0), then the interrupts to level 11 are ignored.

Level 10, TTY Output Interrupts

This level types TTY output from the OUTB buffer.

Level 9, Clock Slave

Level 9 is started from level 13 every 100 ms. All active devices are scanned and their time-out counters are decremented. If a device times out (its counter hits 0) and the program is in interrupt mode, then the IDENT code of that device is put in the 3TASK buffer and level 1 is scheduled by setting PID (1).

If the program is in scanning mode, level 9 only stops decrementing the counter when a time-out occurs. Normal time-out is 4 seconds.

Level 6, Internal Interrupt Processing

This level is scheduled from level 14 when an internal interrupt has occurred. Level 6 reads from the 6TASK buffer and prints an appropriate error report. When the 6TASK buffer is exhausted, level 6 gives up priority and reenables level 14 in case 6TASK had overflowed.

Level 4, Monitor Level

Reads and processes commands from the TTY. Level 4 is scheduled from level 12.

Level 3, Check Level

Level 3 tests and sets up transfers. When a new transfer is set up, the IDENT code of that device is put in the 2TASK buffer and level 2 is scheduled by setting PID (2).

Level 2, Transfer Level

This level activates the devices, thus starting the transfers set up by level 3. Level 2 reads the IDENT code of the devices to activate from the 2TASK buffer.

Level 1, Dummy Level

Level 1 starts level 3 if there is anything in the 3TASK buffer. This is the only way level 3 can be started. In this manner, the testing of transfers will run with higher priority than the transfers themselves, but are started with lower priority.

Level 0, Background.

This level PIN's the TTY input if it becomes unpinned by MOPC during CPU STOP mode. (This was convenient during program debugging.) The level also serves a purpose as it runs a fair amount of IOX during transfers (in interrupt mode only).

2.2 Program Modes

The program may be run in three modes which offer different load characteristics on the data channel. The mode is set by the MODE command (refer to Section 3.9).

2.2.1 MODE = 0, Scanning Mode

In this mode, level 3 will continously scan all active devices for completion or time-out. As devices become ready, the transfers are checked. Assuming no errors, the next operation (write or read) is decided and the new (next) transfer is set up. However, the device is not activated. The IDENT code of the device is put in the 2TASK buffer and level 2 is scheduled by setting PID (2). Level 3 does not give up priority until all active devices have been scheduled for a new transfer.

If some kind of permanent error exists on a device so that a new transfer is not possible, then the device will be timed out and subsequently suspended from operation.

When level 3 scans a device and finds it busy, it will reschedule that device by putting its IDENT code in the 3TASK buffer. Eventually, as all active devices have been scheduled for a new transfer, level 3 gives up priority and level 2 will be started.

Level 2 reads the IDENT code of devices to be activated from the 2TASK buffer. A scrambling algorithm ensures that the order of devices will change with time. As the transfer has been started, the time-out period is set and the IDENT code of the device is put in the 3TASK buffer. The testing is scheduled by setting PID (1). When all transfers have been started, level 2 gives up priority and level 1 is entered.

Level 1 merely startes level 3 if there is anything in the 3TASK buffer, i.e. if level 3 has anything to do. Level 3 will then scan for completion or time-out as previously explained.

It will be seen that this mode of operation puts the highest possible peak-load on the data channel. All active devices will start their transfers within a short period of time (while level 2 is running). Level 2 needs in the order of 100 μs to identify and start a transfer. The transfers will be checked as they come to completion but new transfers will not be started until all previous ones are finished or timed out.

The time-out counters are maintained by level 9.

2.2.2 MODE = 1, Interrupt Mode

When a DMA transfer comes to completion or an error is detected, the associated device will interrupt to level 11. The IDENT code of that device will then be put in the 3TASK buffer and level 1 scheduled by setting PID (1).

As for MODE = 0, level 1 merely starts level 3 and level 3 checks transfers and device status. When a new transfer has been set up (but not yet started), the IDENT code is put in the 2TASK buffer and level 3 now immediately gives up priority so that level 2 is started and activates the device. Level 2 then gives up priority and level 1 is started. If there is more for level 3 to do, it is now restarted from level 1.

This mode puts a higher average load on the data channel, but there will not be high traffic bursts as for MODE = 0.

2.2.3 MODE = 2, Interrupt Mode

This mode is a mixture of MODE = 0 and MODE = 1. The difference from MODE = 1 is that level 3 will not give up priority until it has exhausted 3TASK. This does not, however, imply that all active devices will be tested before new transfers are started (as in MODE = 0).

The data channel load tends to be more random than for the other two modes.

2.3 Device Description

The Device Description refers to the table maintained by the program and which tabulates device characteristics and status. The following is a summary of the most important variables. In this section capital letters denotes names that are known to the EX routine (refer to Section 3.10).

For device names, refer to Appendix B.

DSTS - Device Status

bit 0: 1 = read, 0 = write
bit 1: 1 = test mode
bit 2: 1 = device in operation
bit 3: 1 = inhibit error printout/device has failed
bit 4: 1 = freeze failing operation
(bit 5: 1 = suspend device after error)

DIOX - Basic IOX instruction

DFLG - Bit which flies in error status when device fails

DUNT - Unit number when several units are possible

DLCA - Device core buffer address

DLBA - Device block address (see DBLC and DBIN)

DLWC - Device core buffer size (number of words to transfer)

DBCW - Basic Control Word

This bit pattern will be or'ed into all control words Typically contains PIN bits and unit number.

DIDN - Device Ident code

DRTO - Reasonable time-out period (negative count of .1 sec. intervals)

DPTN - Data Pattern

Data in 1. word of transfer

DINC - Data Increment

Increment in data between following words in transfer

DTP0 - Test Pattern in Even Addresses

DTP1 - Test Pattern in Odd Addresses

DREB - Relevant Error bits in Test mode

(Some errors, such as parity, are irrelevant in test mode)

DBLC - Basic Block Address

DBIN - Block Address increment. This number is randomly added to DBLC to form actual Block Address DLBA

DMDE - Modus word

3.8 DFRZE < d-n >

The FRZE command is reset. Note that the STOP command also resets FRZE.

3.9 MODE < 0/1/2 >

Selects MODE as explained in Section 2.2. It should be noted that is not recommended to change mode while devices are running, as some devices may stop.

3.10 EX

This is a debug package primarily included for program debugging, but some of its features may be useful in normal program use.

The command syntax is similar to the MAC assembler with few exceptions. Symbolic reference may be used to access vital system variables. The following characters convey special meaning:

↶	-	return to command monitor
+	-	plus
-	-	minus
/	-	set address/examine
CR	-	deposit/examine next
*	-	value of current location
↑	-	content of current location
RUB	-	reset
␣	-	space carries same meaning as +
:	-	prints value of expression

Example of use:

```
DSK1 + DUNT/000000↵                    % examine unit number for disk 1
000000 RUB

123/ 123456  0↵                        % zero location 123
234561  *: 124   ↑: 234561  RUB

DSK1B/ 0↵                              % examine the core buffer for disk 1
1↵
2↵
3↵  *-1/ 2
```

3.11 ESUSP <d-n>

This command will cause the device to be suspended after an error. The memory buffer may then be examined in detail by use of the EX command.

This mode is reset by the STOP command and ESCape.

3.12 SAVE <d-n>

This command will save DCHT on the specified disk or drum in such a way that it can be reloaded by the normal NORD-10 mass storage loader.

Note that the program is saved from mass storage address 0.

If DCHT uses this area for testing, a warning is given. In this case the specified device should be reinitialized with a new block address and the program saved again.

3.13 ESCape

The ESCape key is treated as a break character and causes the program to a partial restart. During the partial restart all devices will be stopped and MODE is set to 0, but the devices need not be initialized again.

3.14 HELP

Prints some useful information and a command summary.

4 ERROR MESSAGES

Error reports are self-explanatory. If data errors occur, only the first encountered error will be reported. Status errors will be reported as an octal status word and a list of names of error bits.

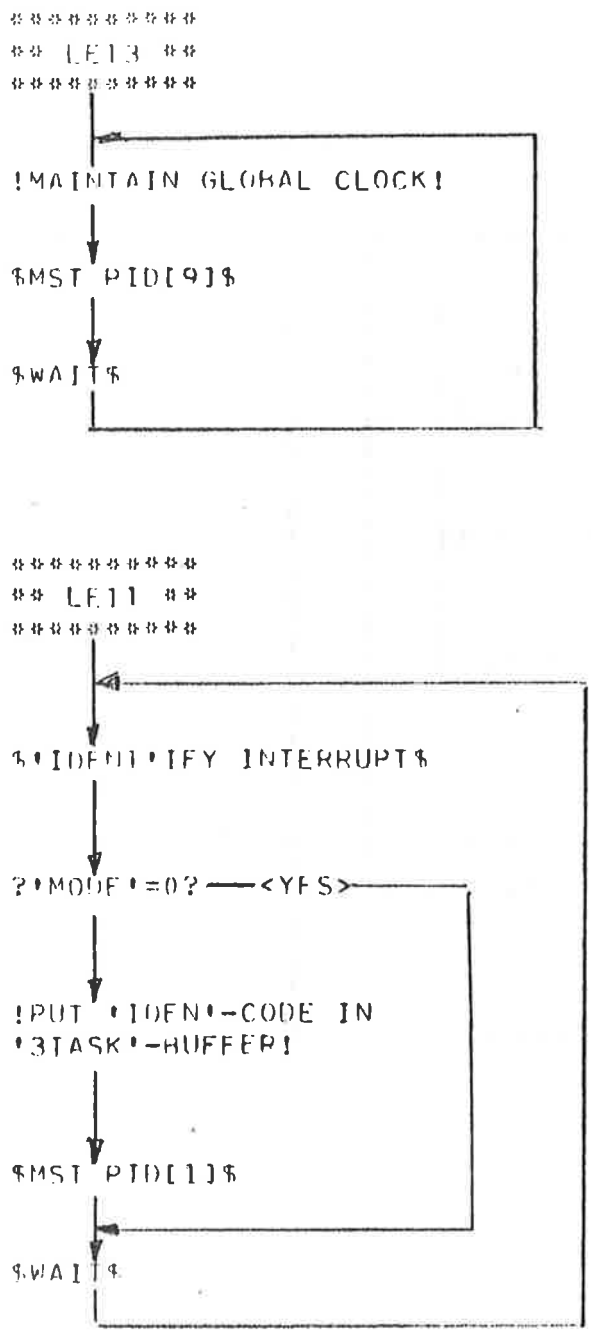
When an error is detected, further error reporting is suppressed for that device. However, a flag-word is put in the A register on level 15 where each device has an associated bit. This bit will flick each time an error is detected for that device. The bit allocation is as follows:

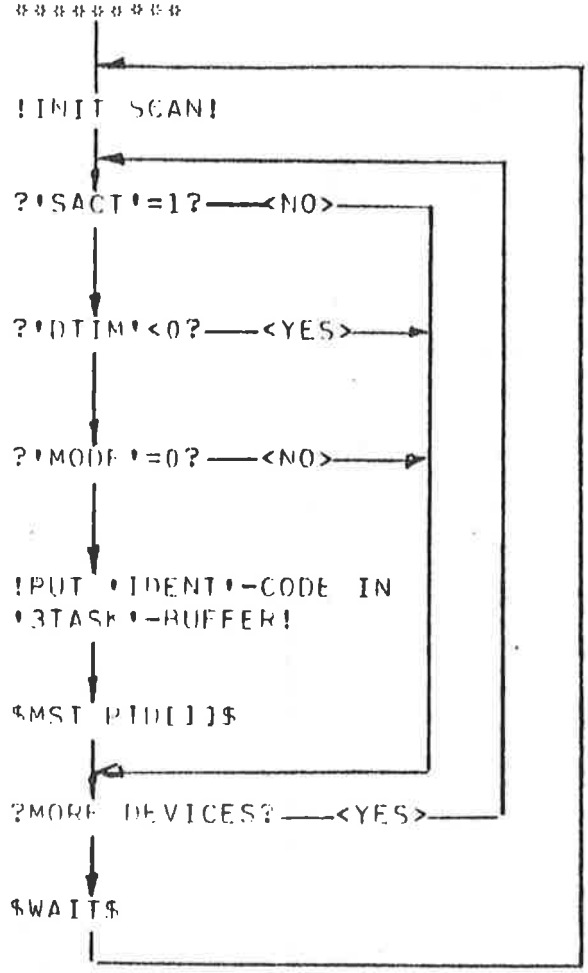
DSK1	-	DSK4	-	bit 0 - 3
DRM1	-	DRM4	-	bit 4 - 7
MT1	-	MT4	-	bit 8 - 11
VT1	-	VT2	-	bit 12 - 13
CAMAC			-	bit 15

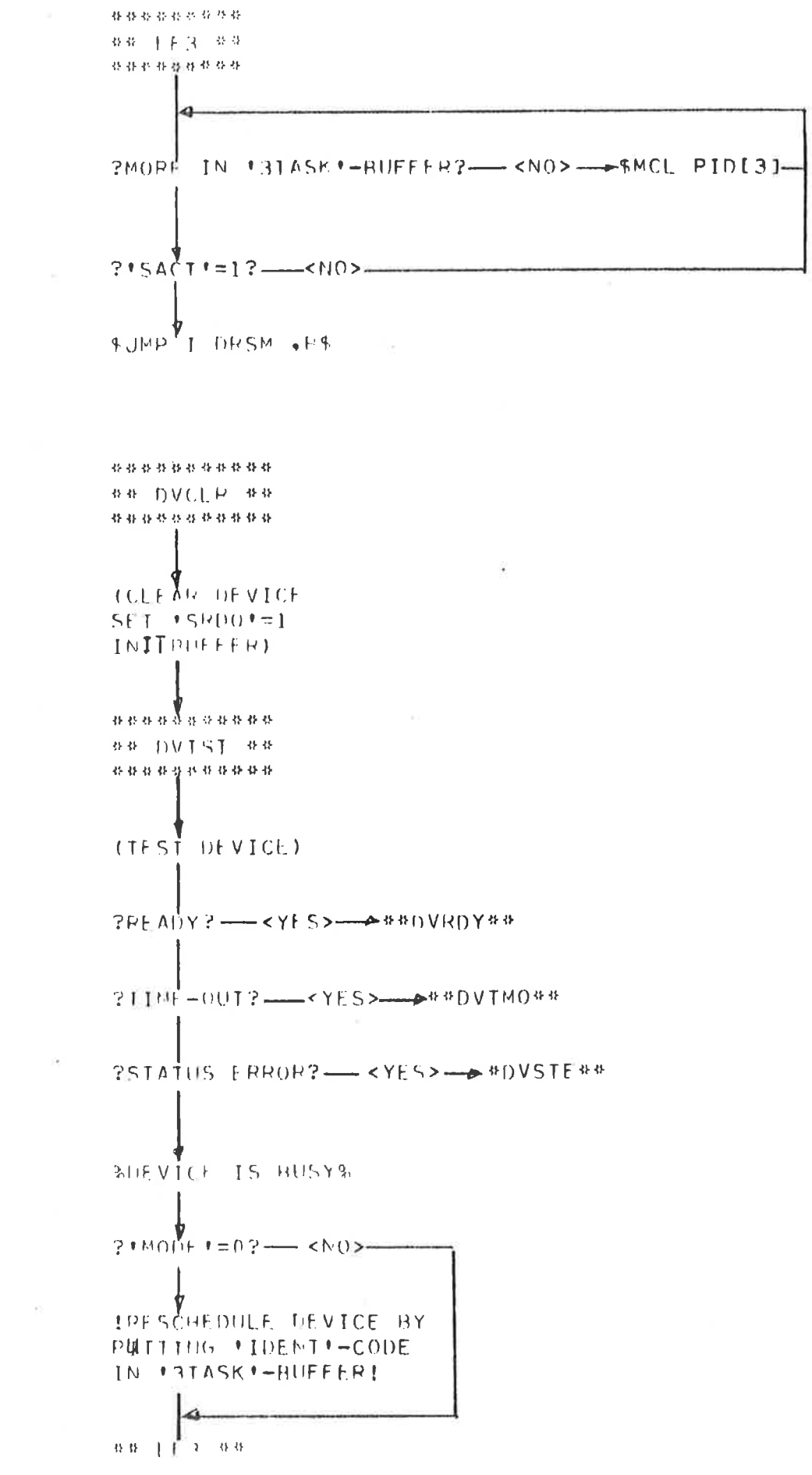
It will be useful to know that a cleared core buffer contains 007700.

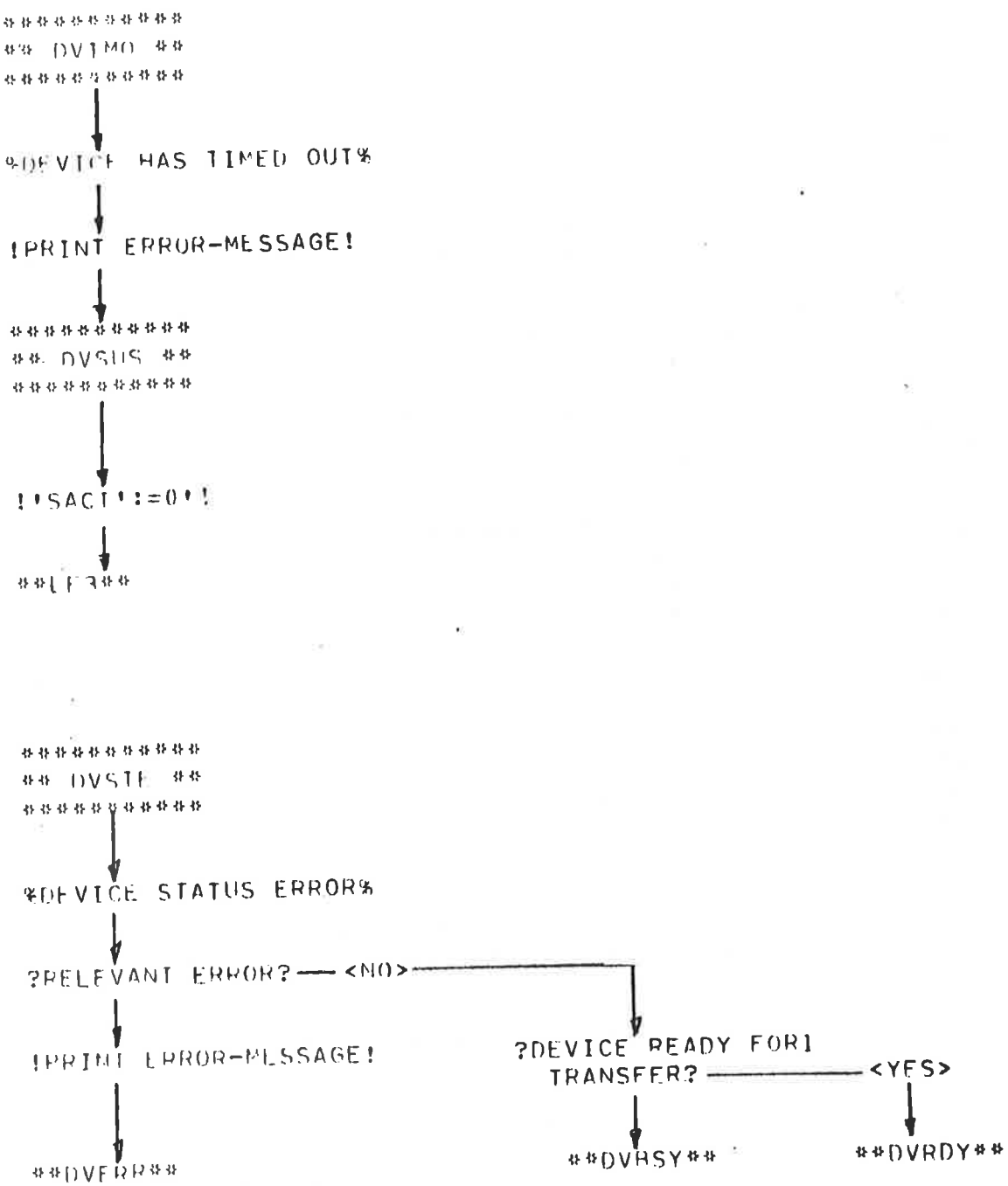
Error interrupts will be reported in full, stating interrupt type, offending level, address and instruction. In case of memory errors the PEA and PES registers will also be displayed.

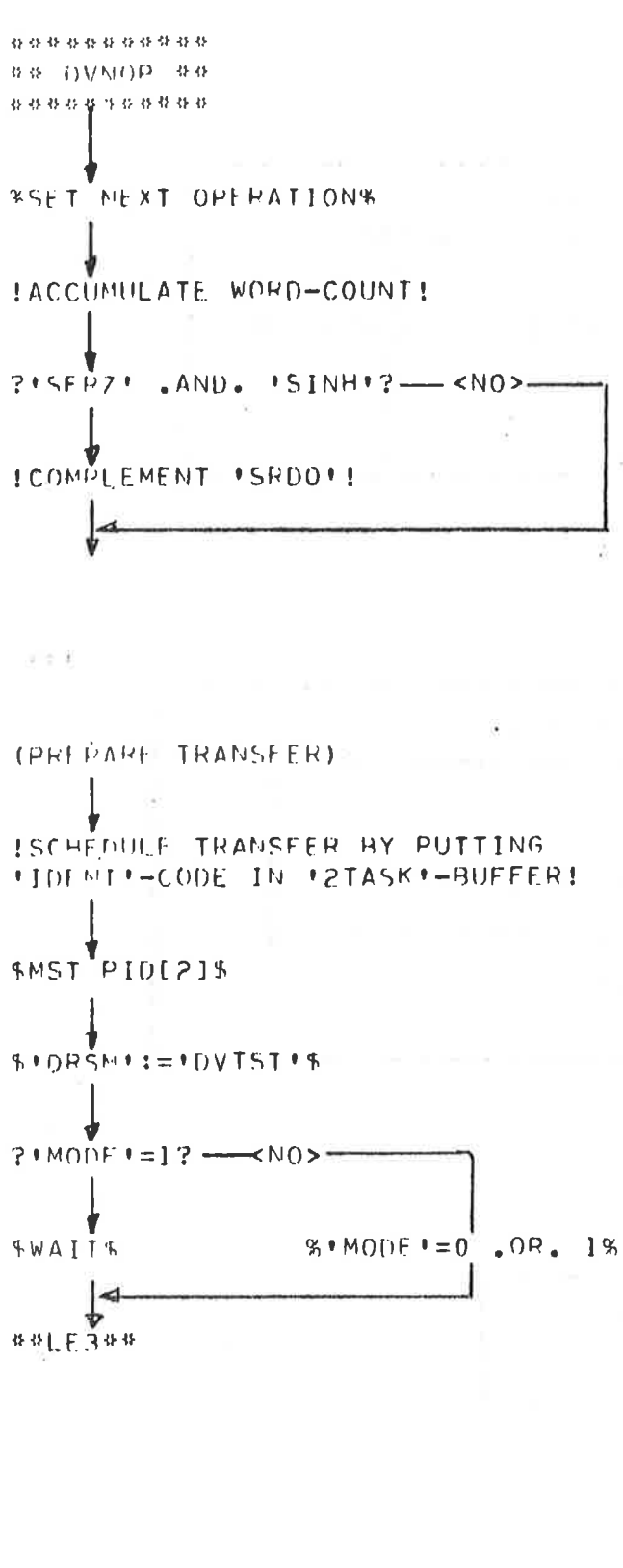
Unexpected IDENT-codes will be reported. Note, however, that this kind of error may be reported when a device is being stopped. This should be ignored, as it only means that a transfer was going on when the STOP command was executed.


****
** LE13 **
****
!MAINTAIN GLOBAL CLOCK!
$MST PID[9]$
$WAIT$
****
** LE11 **
****
$'IDENT'IFY INTERRUPT$
?'MODE'=0? <YES>
!PUT 'IDEN'-CODE IN
'3TASK'-BUFFER!
$MST PID[1]$
$WAIT$


****
** LIO **
****
!INIT SCAN!
?'SACT'=1?
<NO>
?'DTIM'<0?
<YES>
?'MODE'=0?
<NO>
!PUT 'IDENT'-CODE IN
'3TASK'-BUFFER!
$MST PID[ ]$
?MORE DEVICES?
<YES>
$WAIT$


****
** IE3 **
****
?MORE IN *3TASK*-BUFFER?
<NO>
$MCL PID[3]
?*SACT*=1?
<NO>
$JMP I DRSM ,P$
****
** DVCLR **
****
(CLEAR DEVICE
SET *SRDO*=1
INITBUFFER)
****
** DVTST **
****
(TEST DEVICE)
?READY?
<YES>
DVRDY
?TIME-OUT?
<YES>
DVTMO
?STATUS ERROR?
<YES>
*DVSTE**
%DEVICE IS BUSY%
?*MODE*=0?
<NO>
!RESCHEDULE DEVICE BY
PUTTING *IDENT*-CODE
IN *3TASK*-BUFFER!
** IE3 **


****
** DVTMO **
****
%DEVICE HAS TIMED OUT%
!PRINT ERROR-MESSAGE!
****
** DVSTS **
****
!'SACT':=0'!
LF3
****
** DVSTF **
****
%DEVICE STATUS ERROR%
?RELEVANT ERROR?
<NO>
!PRINT ERROR-MESSAGE!
DVERR
?DEVICE READY FOR TRANSFER?
<YES>
DVBSY
DVRDY


****
** DVNOP **
****
%SET NEXT OPERATION%
!ACCUMULATE WORD-COUNT!
?'SFB7' .AND. 'SINH'?
<NO>
!COMPLEMENT 'SRDO'!
(PREPARE TRANSFER)
!SCHEDULE TRANSFER BY PUTTING
'IDENT'-CODE IN '2TASK'-BUFFER!
$MST PID[2]$
$'ORSM':='DVTST'$
?'MODE'=1?
<NO>
$WAIT$
%'MODE'=0 .OR. 1%
LE3