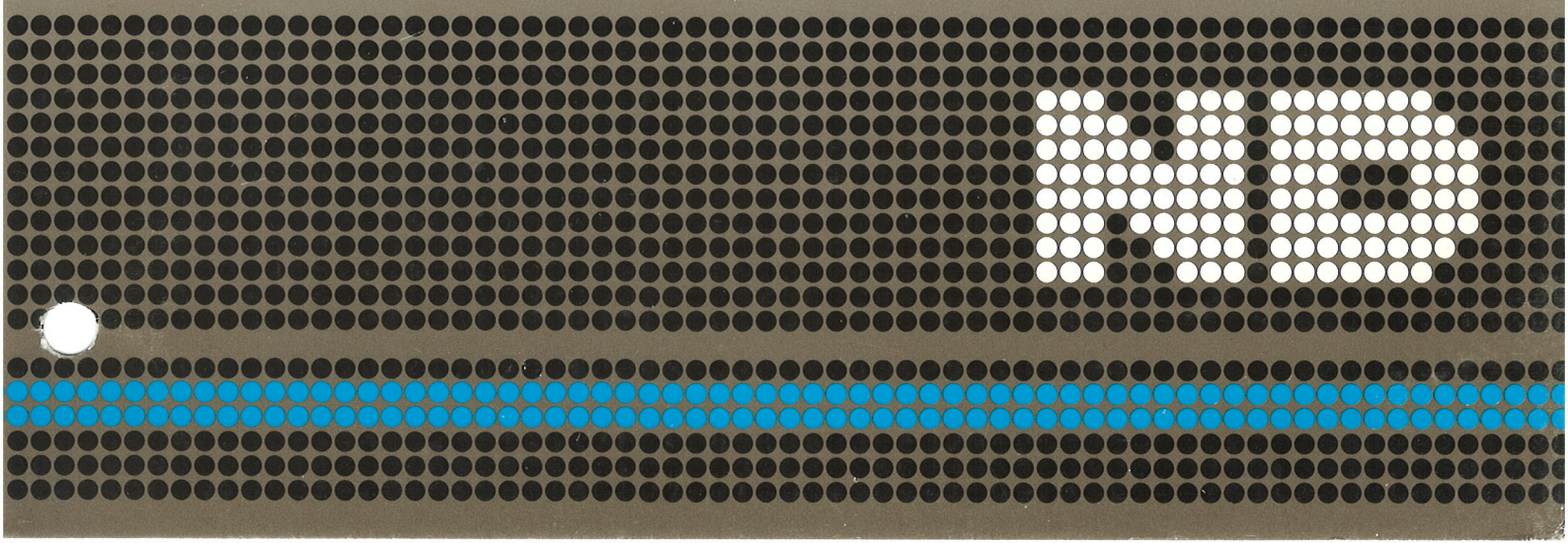
SINTRAN III
Release Information
K version

ND-60.230.2 EN



SINTRAN III Release Information
K-version

Norsk Data ND-60.230.2 EN

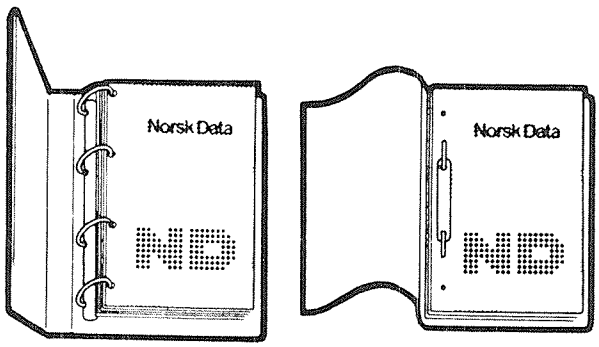
PRINTING RECORD	
PRINTING	NOTES
01/85	Version 01
06/86	Version 02 EN

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.



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:

Norsk Data A.S
Graphic Center
P.O.Box 25 BOGERUD
N-0621 OSLO 6 - Norway

I would like to order

..... Ring Binders, 40 mm, at NOK 45.- per binder

..... Plastic Covers, at NOK 10.- per cover

Name: .....

Company: .....

Address: .....

Preface:

VALID FOR:	
SINTRAN III/VSE	- ALL ND-100
SINTRAN III/VSX	- ONLY ND-100/CX
SINTRAN III/VSX-500	- ONLY ND-500 + ND-100/CX

VERSION K

Keywords for SINTRAN III K-version:

- Increase the configuration size on ND-500 systems considerably. Configurations supported should no longer be limited by operating system.
  In the VSX-version, this includes the utilization of 16 Page Index Tables and the possibility to use many more segments than in the J-version (up to 2048).

- Same functionality as in the J-version.

- At least same performance as the J-version.

- Few new options and features.

- Define standard systems on the VSX-version thus reduce the need to generate SINTRAN III for each system individually.

- Possible to run on all ND-100 CPUs (VSE still supported).

- Remove ND-Net.

TABLE OF CONTENTS

Section Page

Section Page

Section Page

Section Page

1. INSTALLATION

1.1 HARDWARE REQUIREMENTS

SINTRAN III/VSE may run on any ND-100 CPU.
SINTRAN III/VSX requires:
- ND-100/CX CPU with ECO 100-522 (48-bit floating representation)
  or ECO 100-523 (32-bit floating representation)
- Memory management II (16 PITs) with ECO 100-534 (level N)
- Memory management II (16 PITs) with ECO 100-534 (level N)

or - ND-110/CX CPU (CPU and memory management on one card) (level L)
- if SMD disk controller (10 MHz) is used, ECO level BD is required
- if Dual Disk Channel Switch is present, ECO level H is required.

1.2 CHANGES IN HARDWARE SUPPORTED

The new 28-Megabyte and 74-Megabyte disk drives introduced in ND-110 Satellites and ND-110 Compacts are supported.
The old 10-Megabyte, 33-Megabyte and 66-Megabyte disk drives, and the Hewlett-Packard magnetic tape drive are no longer supported in SINTRAN III/VSX version K.

1.3 CONFIGURATION

The SINTRAN III/VSE version K is generated individually for each installation in the same way as SINTRAN III/VSE version J. Standard configurations will exist for the ND-110 Satellite and ND-110 Compact systems.

Normally, SINTRAN III/VSX version K will be delivered as a standard system able to support a great variety of configurations. It will thus be little need to generate a system for a specific installation. SINTRAN III/VSX version K will adjust to the hardware configuration of the computer it is running on. A program to make specific reconfigurations is supplied, a description of it may be found on pages 67-73.

An example of options included in the SINTRAN III/VSX version K standard system is given below:

Disks: 4 ECC controllers with a maximum of 4 units each
and:
2 Winchester controllers with a maximum of 2 units each

Floppy/streamer: 2 controllers with a maximum of 3 units each.
(both types of floppy drives supported)

Magnetic tape:	2 controllers with a maximum of 4 units each (Cipher, Pertec, STC)
Terminals:	maximum 132 terminals
Communication:	12 HDLC interfaces 2 X.21 interfaces 2 synchronous modem interfaces 1 HASP DMA interface 4 PIOCs 2 Ethernet interfaces
Line printers:	2 line printer interfaces (parallel or DMA) 1 Versatec printer/plotter DMA interface
Miscellaneous:	6 universal DMA MPM IV option
Software options:	150 terminal/TAD background tasks 50 terminal access devices (TADs) for COSMOS 10 batch background tasks 20 spooling processes 128 RT-descriptions for user applications 200 ND-500 processes 12 SIBAS processes 10 connect data fields 50 semaphores 20 internal devices (byte-oriented) 2 internal devices (block-oriented) 734 segments XMSG COSMOS spooling 32 Symbolic Debugger tasks 50 Remote file access segments RT and I/O accounting TPS LAMU All monitor calls All RJE queues All logging facilities

1.4 CHANGES IN INSTALLATION PROCEDURE

SINTRAN III/VSE version K will be delivered on either 4 single-sided 8-inch diskettes, or one double-sided 5 1/4-inch or 8-inch diskette, just as for the J-version.

SINTRAN III/VSX version K will be delivered on either 6 single-sided 8-inch diskettes, or 2 double-sided 5 1/4-inch or 8-inch diskettes.

Note that on SINTRAN III/VSX standard systems, you will be asked to define the disk type of your system disk before you use the 10,0$ command to start copying from the first diskette.

The ND-500 system monitor will be installed automatically as part of SINTRAN; it is thus removed from the diskettes containing the ND-500 Monitor (background program, ND-210333) and the ND-500 Swapper (ND-211034). This means that explicit load of the ND-500 system monitor should no longer be done at cold start.

The same applies to RT Accounting; it is now installed as part of SINTRAN III when ordered.

Note also that the COSMOS file transfer server (XFTRA) no longer is part of XMSG (ND-210373), but has been moved to COSMOS Basic Module (ND-210374).

Furthermore, note the change in parameters for the @ENTER-DIRECTORY and related commands (often used in start-up files). The <DEVICE UNIT> parameter is now always required (see "Modified commands" on pages 7-11).

If you have a SINTRAN III/VSX standard system, the S3-CONFIG program (described on pages 67-73) should be used to adjust SINTRAN to your configuration. You should at least run it once after installing SINTRAN III version K for the first time, and adjust these parameters:

- number of background processes
- number of spooling programs
- number of ND-500 processes
- spooling device numbers
- line printer parameters (if you have line printers)

Also, note that you should run S3-CONFIG again each time you install SINTRAN from diskettes (i.e., each time you install a new patch file), but in this case you only have to use the command GENERATE to S3-CONFIG. You can then run S3-CONFIG in a very simple way, the command @S3-CONFIG GENERATE is enough.

1.5 CHANGES TO THE HENT-MODE AND LOAD-MODE FILES

The following changes must be made to the mode file to be run after a cold start (usually called HENT-MODE:MODE):

- Remove any commands used to load the ND-500 System Monitor explicitly.

- Remove any commands used to load RT Accounting explicitly.

- Ensure that all @ENTER-DIRECTORY commands on disks conform to the new parameter syntax - note that UNIT is now always required.

- Include @DEFINE-MASS-STORAGE-UNIT commands to take care of disk units without a fixed mounted pack, as well as magnetic-tape drives and floppy-disk drives.

- Remove any DEFINE-MEMORY-CONFIGURATION commands to the ND-500 Monitor.

The following changes must be made to the batch file to be run after a warm start (usually called LOAD-MODE:BATC):

- Ensure that all @ENTER-DIRECTORY commands on disks conform to the new parameter syntax - note that UNIT is now always required.

- Include @DEFINE-MASS-STORAGE-UNIT commands to take care of disk units without a fixed mounted pack, as well as magnetic-tape drives and floppy-disk drives.

- Include the DEFINE-MEMORY-CONFIGURATION command to the ND-500 Monitor if necessary (refer to page 112 for further details).

- Include the necessary *REMOVE-SPOOLING-HEADER commands (previously used only each time SINTRAN was loaded from diskette) (VSX only).

1.6 CHANGES TO THE NEW-SYSTEM PROGRAM

On most systems (VSX standard systems, ND-110 Satellite and ND-110 Compact), you will get a couple of new questions to answer:

Give CPU number (in decimal): Enter the unique CPU number of your system as shown on the confirmed order of your system, on the SINTRAN III order form, in the LIST-TITLE command, etc.

Give CPU type (in decimal): Enter the CPU type of your system. The CPU type is listed on the confirmed order of your system.

1.7 PATCHING WITH MACM

This section applies to the VSX-version only.
If the symbol DEBUG is undefined (you declare DEBUG as undefined by giving the command DEBUG^), the load procedure will stop after defining the patch macroes.
You can then use the patch macroes to patch SINTRAN or, you can start the system. The patch macroes available are listed when defined by MACM.

1.8 SPACE REQUIREMENTS

On VSX systems, a standard system will require approximately 435_8 pages of memory. This means that you should have at least 1 Megabyte of memory available for the ND-100.

The disk requirements for a VSX-system is calculated as follows:

```
  1090 pages
+ 128 x no. of background programs
+ (system segment size, default 5) x no. of background programs
+ (if MON ADP, 63) x no. of background programs
+ 5 x no. of ND-500 processes
+ 8 x no. of simultaneous Symbolic Debugger segments on the ND-100
+ 4 x no. of simultaneous Remote File Access segments
+ (size of spooling queue segm: 2-12, default 4) x no. of spooling pr.
```

For a standard VSX-system without any reconfigurations, this adds up to a total of approximately 7000 pages on disk.

2. SINTRAN III COMMANDS

The SINTRAN-Service-Program command *SET-COMMAND-PROTECTION may now be used to change the command protection of all commands (including file system commands). Command protection on file system commands may not be changed to a lower protection level (for example, a SYSTEM command may not be made public).

It is also possible to set command protection on reentrant subsystems and ND-500 standard domains (i.e. on entries in the reentrant subsystem table).

Furthermore, the SINTRAN-Service-Program command *REMOVE-SINTRAN-COMMAND now only disables the use of a command; the command is restored by *REINSERT-SINTRAN-COMMAND.

2.1 COMMANDS REMOVED

All commands which have been removed, were related to ND-Net.

2.1.1 @COMMUNICATIONS-LINE-STATUS

2.1.2 @COMMUNICATIONS-STATUS

2.1.3 @LOCAL

2.1.4 @REMOTE

2.1.5 @REMOTE-LOAD

2.1.6 @REMOTE-PASSWORD

2.1.7 @START-COMMUNICATION

2.1.8 @STOP-COMMUNICATION

2.2 MODIFIED COMMANDS

2.2.1 @APPEND-SPOOLING-FILE

The first and second parameters may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <PERIPHERAL FILE NAME>
<FILE NAME>
<NUMBER OF COPIES>
<TEXT (TERMINATED WITH ')>

Refer to chapter 7 ("Spooling") on page 79 for further details.

2.2.2 @CHANGE-DIRECTORY-ENTRY

The second parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]
(<SUBCOMMANDS>)

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.3 @COPY

The two parameters may now specify files on remote systems, the file specifications may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <DESTINATION FILE>
<SOURCE FILE>

2.2.4 @COPY-DEVICE

The two parameters DEVICE UNIT are now always required parameters.

The parameter sequence is thus: <DESTINATION DEVICE>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]
<SOURCE DEVICE>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.5 @COPY-FILE

The two parameters may now specify files on remote systems, the file specifications may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <DESTINATION FILE>
<SOURCE FILE>

2.2.6 @CREATE-DIRECTORY

The third parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DIRECTORY NAME>
<DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]
<BIT FILE ADDRESS>

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.7 @DELETE-SPOOLING-FILE

The first and second parameters may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <PERIPHERAL FILE NAME>
<FILE NAME>

Refer to chapter 7 ("Spooling") on page 79 for further details.

2.2.8 @DUMP-DIRECTORY-ENTRY

The second parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]
<OUTPUT FILE>

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.9 @ENTER-DIRECTORY

The third parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DIRECTORY NAME>
<DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.10 @FILE-STATISTICS

The first parameter may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <FILE NAME>
<OUTPUT FILE>

2.2.11 @LIST-DEVICE-FUNCTIONS

A new parameter is introduced as the first parameter.

The parameter sequence is thus: <COMMAND>
<OUTPUT FILE>

2.2.12 @LIST-EXECUTION-QUEUE

When the list of RT-programs in the time queue is greater than 16, the list is sorted columnwise from left to right.

2.2.13 @LIST-FILES

The first parameter may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <FILE NAME>
<OUTPUT FILE>

2.2.14 @LIST-TIME-QUEUE

When the list of RT-programs in the time queue is greater than 16, the list is sorted columnwise from left to right.

2.2.15 @LOOK-AT

In the VSX-version of SINTRAN III version K, there are some changes to the first parameter (space reference):

RESIDENT	is no longer available
IMAGE	is no longer available
COMMON-CODE	is a new area which can be investigated by the LOOK-AT command.

A segment name may also be used as space reference (without specifying SEGMENT).

2.2.16 @MOVE-SPOOLING-QUEUE-ENTRY

The first, second and last parameters may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <PERIPHERAL FILE NAME>
<FILE NAME>
<INSERT OR APPEND>
<FILE NAME>

Refer to chapter 7 ("Spooling") on page 79 for further details.

2.2.17 @RELEASE-DEVICE-UNIT

The second parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.18 @REMOVE-FROM-SPOOLING-QUEUE

The first and second parameters may now specify files on remote systems, the file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

The parameter sequence is unchanged: <PERIPHERAL FILE NAME>
<FILE NAME>

Refer to chapter 7 ("Spooling") on page 79 for further details.

2.2.19 @RENAME-DIRECTORY

The fourth parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <OLD DIRECTORY NAME>
<NEW DIRECTORY NAME>
<DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.2.20 @RENAME-FILE

Files which are open may no longer be renamed.

2.2.21 @RESERVE-DEVICE-UNIT

The second parameter, DEVICE UNIT, is now always a required parameter.

The parameter sequence is thus: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<SUBUNIT>]

Note that if you give an erroneous disk type as device type (this may happen, for example, if your system has different disk types as different units and you specify the wrong unit), the disk will enter fault state and the system will hang.

2.3 NEW COMMANDS

2.3.1 @DEFINE-MASS-STORAGE-UNIT

Define a mass storage device in the directory table. This is used to reserve a directory index for a device even if the device is not to be entered yet. On standard versions of SINTRAN III/VSX version K, devices are not allocated a directory index at generation time, but are placed in the directory table when the device is defined. Also some devices which cannot contain directories, for example magnetic tape drives, must be defined.
Defining a device is done either by entering it (@ENTER-DIRECTORY), or by this command.

Parameters: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<DEVICE SUB-UNIT>]

A definition of a device in the directory table does not survive a warm start.

This command is restricted to user SYSTEM only.

2.3.2 @DELETE-MASS-STORAGE-UNIT

Delete a definition of a mass storage device from the directory table.

Parameters: <DEVICE NAME>
<DEVICE UNIT>
[<FIXED(F) OR REMOVABLE(R)>]
[<DEVICE SUB-UNIT>]

This command is restricted to user SYSTEM only.

2.3.3 @GIVE-OBJECT-BLOCKS

This command will make it possible to create more than 256 files for a single user. To allow this, the user must be given more than the single object block given initially. Each object block contains object entries for 256 files. The maximum number of object blocks per user is 16 (which means a maximum of 4096 files). The number of files allowed for a user is reported by the command @USER-STATISTICS.

Parameters: <DIRECTORY NAME:USER NAME>
<NUMBER OF OBJECT BLOCKS>

If the directory resides on a diskette, this command is allowed for all users; on other devices it is restricted to user SYSTEM only.

2.3.4 @LIST-MASS-STORAGE-UNITS

List all mass storage units and corresponding directory index. The default value of the parameter <output file> is the terminal.

Parameter: <OUTPUT FILE>

This command is allowed for all users.

2.3.5 @SET-INITIAL-FILE-ACCESS

Set default file access to be used for all users subsequently created on this system. The default file access may be changed for each user individually by the command @SET-DEFAULT-FILE-ACCESS.

Parameters: <PUBLIC ACCESS (R,W,A,C,D IN COMBINATIONS OR N)>
<FRIEND ACCESS (R,W,A,C,D IN COMBINATIONS OR N)>
<OWN ACCESS (R,W,A,C,D IN COMBINATIONS OR N)>

This command is restricted to user SYSTEM only.

2.3.6 @SET-INITIAL-FRIEND-ACCESS

Set default friend access to be used for all users subsequently created on this system. The friend access may be changed for each user individually by the command @SET-FRIEND-ACCESS.

Parameters: <ACCESS (R,W,A,C,D IN COMBINATIONS OR N)>

This command is restricted to user SYSTEM only.

2.3.7 @TAKE-OBJECT-BLOCKS

This command is used to restrict the number of files for a single user. The object blocks to be "taken" must be free; thus if a user is allowed to have a maximum of 512 files, with only files number 0, 1, 2 and 300 used, both object blocks for this user are used and the command may not be given. The number of files allowed for a user is reported by the command @USER-STATISTICS, and the command @LIST-FILES will show which file numbers are used.

Parameters: <DIRECTORY NAME:USER NAME>
<NUMBER OF OBJECT BLOCKS>

If the directory resides on a diskette, this command is allowed for all users; on other devices it is restricted to user SYSTEM only.

3. MONITOR CALLS (ND-100)

3.1 MODIFIED MONITOR CALLS

3.1.1 WCI MON 20

On error return, the A-register previously contained a random value. It will now contain error code 44_8 ("Too long parameter").
MON 20 is for internal use by ND only.

3.1.2 TUSED MON 114

MON TUSED (MON 114) may now also be called from RT-programs.

3.1.3 ABSTR MON 131

The restriction that parameters had to reside on PIT 0 is removed for the VSX-version, thus:
Parameters may now reside on any PIT (VSX only).
Parameters must reside on PIT 0 (VSE only).

EXABS (MON 335) is recommended if you write code to be independent of whether it is run on VSE or VSX.

In function 43 (read format table) and 44 (write format table), a 32-bit INTEGER is now used as the third parameter (disk address).

3.1.4 ENTSG MON 157

MON ENTSG (MON 157) has been changed to "remember" which segments are entered (maximum 24 segments may be remembered), and reenter these after a power fail restart.
The segments may be removed from the selected PIT by using function 4 (REMSG) of MON SGMTY (MON 341).
Unfixing the segment (using MON 116 - UNFIX) also removes the segment from the Page Index Table.

There are some changes to the parameters (new range of PITs and a new set of output values).

Monitor call format:

```
        LDA  (PARLI             % A = address of parameter list
        MON  157                % ENTSG

PARLI,  (ISEG                   % segment
        (IPIT                   % PIT (for VSX, the range is 0-17₈)
        (INTLV                  % interrupt level
        (ISTRT                  % start address (entry point)
```

Input parameters:
<ISEG> segment
<IPIT> PIT (for VSX, the range is 0-17$_8$)
<INTLV> interrupt level (must be a free level, one of: 6,7,10$_8$,11$_8$)
<ISTRT> start address (entry point)

Output parameters:
A-register = 0 : ok
-1 : attempt to enter too many segments
-2 : illegal segment number
-3 : illegal Page Index Table
-4 : segment is not fixed
-5 : illegal interrupt level
-6 : PIT already in use

3.1.5 DEBUG MON 205

Functions for multi-segment debugger has been added to this monitor call. MON 205 is for internal use by ND only.

3.1.6 APSPF MON 240

The parameters containing the file name of the file to be appended to the spooling queue (pointed to by the X-register), and the peripheral file (pointed to by the A-register), may now contain a remote file specification. The complete remote file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

Refer to chapter 7 ("Spooling") on page 79 for further details.

3.1.7 SUSCN MON 241

If originally logged in as user RT, it is no longer possible to set user context to user SYSTEM.

3.1.8 DEABF MON 256

The parameters containing an abbreviated file name (pointed to by the X-register), and full a file name (pointed to by the A-register), may now contain a remote file specification. The complete remote file specification may contain the following parameters:

system(user(password:project)).(directory:user)file:type;version

If the abbreviated file name contains a remote specification, the name of the remote system may not be abbreviated.

3.1.9 CPUST MON 262

The ND-110/CX CPU is now supported and some new values are returned. The following words of the returned array are affected:

DISP	NAME	DESCRIPTION
1	HWINFI(0)	Hardware information Left byte = CPU type 0 - 3 unchanged 4 = ND-110/CX 48-bit floating 5 = ND-110/CX 32-bit floating 6 - 255 Not used Right byte = Instruction set 0 - 1 unchanged 2 = ND-100/CX with 4 PITs 3 = ND-100/CX, ND-110/CX with 16 PITs 4 - 255 Not used
2	HWINFO(1)	ND-110/CX microprogram version
3	HWINFO(2)	System type (100, 500, 502, 5561, ...) The system type is either supplied when the system is generated or you will be asked for it when you install SINTRAN from diskettes.

3.1.10 MLAMU MON 315

A new function, function 11_8 is introduced.

The monitor call format is unchanged:

```
        LDA   (PARLI     % A = address of parameter list
        MON   315        % MLAMU
        JMP   ERROR      % error return
        ..........       % normal return

PARLI, (FUNC
       (LAMID
       (SIZE
       (PHYSA
```

Input parameters:

FUNC	= 11	: Create system LAMU
LAMU id	= 0	: The system will return the selected <LAMU id>
	≠ 0	: The LAMU will be identified by the given <LAMU id> if the <LAMU id> is unused and inside legal range. Legal range for system LAMUs is -2:-1023.
size		: Number of pages in the LAMU. Legal range $1\text{-}200_8$
phys.addr	= 0	: The system will select a free memory area large enough for the LAMU, and reserve it. The first physical page number of this memory area will be returned in <phys.addr>.
	≠ 0	: Specifies the first physical page for the LAMU.

Output parameters:

LAMU id : The LAMU identifier
phys.addr : The first physical page number of the system LAMU.

Rules:

1. Only legal from users SYSTEM or RT, or from RT-programs running on protection ring 1.

3.1.11 FSMTY MON 327

Three new functions introduced: T-register = 2 : return block size
= 3 : get file name
= 4 : get file/device information

The monitor call format is thus:

```
        LDT   FUNC      % T = function
        LDA   FILNO     % A = open file number
        LDX   (BUFFR    % X = address of buffer to receive file name
        MON   327
        JMP   ERROR     % error return
        ..........      % normal return

FUNC,   2
FILNO,  101
BUFFR, 0; *+26/
```

Function no. 1:

Function:
Write back the open-file-table index-block for an open file to disk.

Input parameters:
T-register : function = 1
A-register : open file number

Output parameters:
Return: Error - A-register contains error code
Skip return: OK, normal return

Function no. 2:

Function:
Return block size of an open file.

Input parameters:
T-register : function = 2
A-register : open file number

Output parameters:
Return: Error - A-register contains error code
Skip return: OK, normal return, A-register = block size in words.

Function no. 3:

Function:
Get full file name of an open file.
File name may be a remote file and contain the following parameters:

system.(directory:user)file:type

Input parameters:
T-register : function = 3
A-register : open file number
X-register : address of buffer to receive file name

Output parameters:
Return: Error - A-register contains error code
Skip return: OK, normal return, buffer pointed at by X contains name.

Function no. 4:

Function:
Get information of an open file identified by open file number or device number.

Input parameters:
T-register : function = 4
A-register : open file number or device number

Output parameters:

Return: Error - A-register contains error code
Skip return: OK, normal return, the following information is returned:
A = open file number or device number (the value returned in the A register is the opposite of A as input, i.e., if open file number is input, device number is returned and the other way around).
X = TYPRING bits (TYPRING word from data field of device)
D = status : bit 0 = 1 if file is open for write.
bit 1 = 1 if spooling file or terminal/TAD.

3.1.12 UDMA MON 333

Function 62 (Wait on interrupt/DMA finish) now has two new subfunctions:

DPARI = 2 : Enable RT on interrupt (set repeated execution)
DPARI = 3 : Disable RT on interrupt (clear repeated execution)

3.1.13 IOMTY MON 336

Note that the functions and parameters used in the J-version of MON 336 (IOMTY) are now revised completely.

The I/O multifunction (IOMTY) monitor call is used to change some of the attributes of terminal and terminal access device (TAD) input and output. This monitor call needs a varying number of input and output parameters depending upon function; all parameters are therefore placed in an array.

Monitor call format:

```
        LDA   (PARLI     % A = address of parameter list
        MON   336        % IOMTY
        JMP   ERROR      % error return
        ..........       % normal return

PARLI,  FUNC             % address of function
        SIZE             % address of the length of the parameter array
        ARRAY            % address of the parameter array

ARRAY,  0        % function parameter 1  (word 1)
        0        % function parameter 2  (word 2)
        0        % function parameter 3  (word 3)
        !             !
        !             !
        0        % function parameter n  (word n)
```

Input parameters:

FUNC = function code
SIZE = length of function parameter array (must be greater than or equal number of input/output parameters specified for function).
ARRAY = function parameter array.

The function code (FUNC) is divided into 2 ranges, 0 to 77_8 and 100_8 to 177_8. Range 0-77 may only update SINTRAN memory area, and range 100 to 177 may update SINTRAN memory, image and save area.

The meaning of AREA when used in the input parameter array is:

AREA	Update resident?	Update image?	Update save?
1	Yes	No	No
2	No	Yes	No
3	Yes	Yes	No
4	No	No	Yes
5	Yes	No	Yes
6	No	Yes	Yes
7	Yes	Yes	Yes

Return : Error, A-reg = error code.
X-reg = area which failure occured:
1 = memory, 2 = image, 4 = save.
X-reg is only set for functions 100-177.
Skip return: OK.

The following functions are defined:

Function	Brief explanation
0	Set terminal to become own terminal.
1	Reset to original own terminal.
2	Set character conversion mode for device no. 0.
3	Get character conversion mode for device no. 0.
4	Set echo strategy.
5	Get echo strategy.
6	Set break strategy.
7	Get break strategy.
10_8	Set terminal mode.
11_8	Get terminal mode.
12_8	Set/reset 8-bit unmodified input and output.
13_8	List terminals and TAD's in system.
14_8	Display functions.
15_8	Change signals on a V.24 connection.
16_8	Set/reset terminal in/from test mode.
100_8	Return function parameters set by functions 101 - 177.
101_8	Set terminal type.
102_8	Set escape or local character.
103_8	Start and stop Xon/Xoff protocol, input control.
104_8	Start and stop Xon/Xoff protocol, output control.
105_8	Set Xon/Xoff only or dual function Xon/Xoff.
106_8	Set character length.
107_8	Set baud rate for terminal.
110_8	Set number of stop bits.
111_8	Set terminal to printer or reset printer to terminal.
112_8	Set half or full duplex on terminal.
113_8	Set/reset variable speed on terminal.
114_8	Set/reset terminal connected to printer.

Notes:

1. Device number 1 is console terminal.
2. If background program, then logical device number 0 means own terminal.
3. User SYSTEM does not have to reserve a device before changing attributes (except functions 0 and 111).

Rules:

Common rules are given in the following table. Note that there may be additional rules specified under description on each function.

Function number	Minimum function parameter array size	Update SINTRAN image/save area	Update own terminal's SINTRAN memory area	Update other terminal's memory area	Reservation of device needed	Allowed on TADs	Callable from
0	1	-	P	P	I,O	No	B
1	1	-	P	P	None	Yes	B
2	2	-	P	P	I	Yes	B
3	2	-	P	P	None	Yes	B
4	3	-	P	P	I	Yes	F,B
5	3	-	P	P	None	Yes	F,B
6	4	-	P	P	I	Yes	F,B
7	4	-	P	P	None	Yes	F,B
10	3	-	P	P	I,O	Yes	F,B
11	1	-	P	P	None	Yes	F,B
12	2	-	P	P	I	No	F,B
13	1	-	-	-	-	Yes	F,B
14	4	-	S	S	-	Yes	F,B
15	3	-	P	P	I	Yes	F,B
16	2	-	S,R	S,R	None	Yes	F,B
100	*)	-	**)	**)	None	Yes	F,B
101	3	S	P	S,R	I	Yes	F,B
102	4	S	P	S,R	I	Yes	F,B
103	5	S	P	P	I	No	F,B
104	5	S	P	P	O	No	F,B
105	3	S	P	P	I	No	F,B
106	4	S	P	S,R	I	No	F,B
107	4	S	P	S,R	I	No	F,B
110	3	S	P	S,R	I	No	F,B
111	4	S	S	S	I,O	No	F,B
112	3	S	P	S,R	I	No	F,B
113	3	S	S	S	I	No	F,B
114	3	S	S	S	I	No	F,B

Abbreviations used:

B: Background program
F: Foreground program (RT-program)
I: Input part of device
O: Output part of device
P: Public
R: User RT
S: User SYSTEM

*) Depends on which function to return parameter(s) from.

**) Only allowed to return function parameters for functions user is allowed to set.

Error returns from IOMTY, MON 336:

X-reg: Letter "N" means: X-reg is not set by IOMTY.
Letter "Y" means: X-reg = area which failure occured:
1 = memory, 2 = image, 4 = save.
A-reg: Octal.

A-reg	X-reg	Brief explanation
5	N	Device not reserved.
25	N	You are not authorized to do this
33	N	No such logical unit.
153	N	Address outside segments bounds.
155	N	File already opened by another user. (SINTRAN III/VSE-version only, will be returned if the file SINTRAN:DATA is opened by another user).
174	N	Illegal parameter.
201	N	Illegal function code.
240	N	Illegal device type.
346	N	Illegal baud rate specified.
347	N	Illegal character length.
350	N	Illegal parity.
351	N	Only legal for background.
352	N	A device already defined as own terminal.
353	N	Illegal break/echo strategy.
354	N	Function parameter array to small.
355	N	Illegal AREA specified.
356	Y	Not 8-bit character length.
357	N	Terminal already in a display table.
360	N	Terminal not a master terminal.
361	N	Terminal not connected to specified master terminal.
362	N	Display table contain a device which is not a terminal
363	Y	Error occurred during read/write in SINTRAN memory/image/save areas.
364	Y	Baud rate is not set by software.
365	Y	Illegal baud rate found in SINTRAN memory/image/save data field (TSPEED).
366	N	Terminal is active.

Function code 0:

See also function code 1.

Function description:
Set logical device number to become own terminal.
The terminal identified by the logical device number specified in word 1 will act as own terminal.

Input parameters:
Word 1 = Logical device number to become own terminal.

Rules:
1. Both the input and output part of a device must be reserved (even if user SYSTEM).

Function code 1:

See also function code 0.

Function description:
Reset to original own terminal.

Input parameters:
None

Output parameters:
Word 1 = logical device number for terminal, set by function code 0.
0 is returned if not set.

Function code 2:

See also function code 3.

Function description:
Set character conversion mode for input when reading from device number 0, see MON INBT.

Input parameters:
Word 1 = logical device number.
Word 2 = character conversion mode:
0 = no conversion.
1 = all characters uppercase.

Notes:
1. Only input in user mode is affected.

Function code 3:

See also function code 2.

Function description:
Get character conversion mode.

Input parameters:
Word 1 = logical device number.

Output parameters:
Word 1 = unchanged
Word 2 = character conversion mode as in function 2.

Function code 4:

See also function code 5 and MON ECHOM (MON 3).

Function description:
Set echo strategy.

Input parameters:
Word 1 = logical device number.
Word 2 = echo strategy, as A-reg in MON ECHOM (MON 3).
Word 3 = memory address to an 8-word bit map if user defined echo strategy, otherwise 0.

Function code 5:

See also function code 4.

Function description:
Get echo strategy.

Input parameters:
Word 1 = logical device number.
Word 2 = memory address to an 8-word bit map.

Output parameters:
Word 1 = unchanged
Word 2 = unchanged
Word 3 = echo strategy, if returned with value 7 (user defined echo strategy) then 8-word bit map is returned to memory address specified in input word 2.

Function code 6:

See also function code 7 and MON BRKM (MON 4).

Function description:
Set break strategy.

Input parameters:
Word 1 = logical device number.
Word 2 = break strategy, as A-reg in MON BRKM (MON 4).
Word 3 = memory address to an 8-word bit map if user defined break strategy, else 0.
Word 4 = maximum number of characters before break if break strategy $\geq$ 3 else 0

Function code 7:

See also function code 6.

Function description:
Get break strategy.

Input parameters:
Word 1 = logical device number.
Word 2 = memory address to an 8-word bit map.

Output parameters:
Word 1 = unchanged
Word 2 = unchanged
Word 3 = break strategy, if is returned with value 7 (user defined break strategy) then an 8-word bit map is returned to memory address specified in input word 2.
Word 4 = maximum number of characters before break if break strategy ≥ 3 else 0

Function code 10_8:

See also function code 11 and TERMO (MON 52).

Function description:
Set communication mode for a terminal.

Input parameters:
Word 1 = logical device number.
Word 2 = MODE, as in MON TERMO (MON 52).
Word 3 = communication mode:
0 = communication mode set as in MON TERMO (MON 52)
1 = set only functions marked "Y" in MODE, see MODE in TERMO
2 = reset functions marked "Y" in MODE, see MODE in TERMO

Function code 11_8:

See also function code 10.

Function description:
Get communication mode for a terminal.

Input parameters:
Word 1 = logical device number.

Output parameters:
Word 1 = unchanged.
Word 2 = MODE, as in TERMO (MON 52).

Function code 12_8:

Function description:
Set and reset 8-bit unmodified input/output.
Unmodified means no parity on the most significant bit in byte.

Input parameters:
Word 1 = logical device number.
Word 2 = 8-bit<status:
Word 2 = 0 = set 8-bit unmodified input/output.
1 = reset to 7-bit input/output, parity on most significant bit in byte.

Rules:
1. Only legal for terminals with 8-bit character length (see also function 106)

Notes:
1. 8-bit unmodified input/output only valid in user mode.

Function code 13_8:

Function description:
List all logical device numbers for terminals and TAD's defined in the system.

Input parameters:
Word 1 = logical device number.
0 if first time, otherwise last device number returned, see output parameter word 1.

Output parameters:
Word 1 = last device returned.
-1 if no more logical device numbers to be returned.
Word 2 = number of logical devices returned.
Word 3 = logical device number.
Word 4 = device type: 0= TAD, 1= terminal.
Word 5 = logical device number.
!
!
Word n

Rules:
1. Permitted for users SYSTEM and RT only.

Function code 14_8:

Function description:
Display functions.
This function consists of several subfunctions specified in input parameter word 1. It is possible to define several display tables. One display table consists of one master terminal and several slave terminals connected to the master terminal. All what is written on the master terminal will also be written on slave terminals.

Rules:
1. Only allowed for user SYSTEM.
2. A terminal may only be in one display table.

Subfunction code 1:

Subfunction description:
Define a master terminal.

Input parameters:
Word 1 = 1
Word 2 = logical device number to become a master terminal.

Subfunction code 2:

Subfunction description:
Undefine a master terminal, master terminal specified and slave terminals connected become "ordinary" terminals again.

Input parameters:
Word 1 = 2
Word 2 = logical device number for master terminal.

Subfunction code 3:

Subfunction description:
Insert a slave terminal in display table.

Input parameter:
Word 1 = 3
Word 2 = logical device number for master terminal.
Word 3 = logical device number for terminal to be inserted.

Subfunction code 4:

Subfunction description:
Remove a slave terminal from display table.

Input parameters:
Word 1 = 4
Word 2 = logical device number for master terminal.
Word 3 = logical device number for slave terminal to be removed.

Subfunction code 5:

Subfunction description:
List master terminals defined in system.

Input parameters:
Word 1 = 5
Word 2 = last device returned.
0 if first time, otherwise last device number returned, see output word 2.

Output parameters:
Word 1 = unchanged
Word 2 = last device returned.
-1 if no more logical device numbers to be returned.
Word 3 = number of devices returned
Word 4 = device number for master terminal
Word 5 = device number for master terminal
!
!
Word n = ...

Subfunction code 6:

Subfunction description:
List display table for a master terminal.

Input parameters:
Word 1 = 6
Word 2 = logical device number for master terminal.
Word 3 = last device returned.
0 if first time, otherwise last device number returned, see output word 3.

Output parameters:
Word 1 = unchanged
Word 2 = unchanged
Word 3 = last device returned.
-1 if no more logical device numbers to be returned.
Word 4 = number of devices returned.
Word 5 = device number for slave terminal connected
Word 6 = device number for slave terminal connected
Word 7 = device ...
!
!
Word n

Subfunction code 7:

See also subfunction code 10.

Subfunction description:
Enable/disable display for a master terminal.

Input parameters:
Word 1 = 7
Word 2 = logical device number for master terminal.
Word 3 = display status:
0 = enable display.
1 = disable display.

Subfunction code 10_8:

See also subfunction 7.

Subfunction description:
Get display status for a master terminal.

Input parameters:
Word 1 = 10
Word 2 = logical device number for master terminal.

Output parameters:
Word 1 = unchanged
Word 2 = unchanged
Word 3 = display status:
0 = enabled for display
1 = disabled for display.

Function code 15_8:

Function description:
Set signals on a RS-232 connection.
This function consist of several subfunctions specified in input parameter word 2.

Rules:
1. Subfunction 1 is only allowed if 8-terminal buffer interface with FIFO (ND-102730 and ND-102740).

Subfunction code 0:

Subfunction description:
Turn off the V.24 signal DTR. It will automatically be turned back on again after about 5 seconds.

Input parameters:
Word 1 = logical device number
Word 2 = 0

Output parameters:
None

Subfunction code 1:

Subfunction description:
Set level of V.24 control line available on pin 19 in the plug panel connector.

Input parameters:
Word 1 = logical device number.
Word 2 = 1
Word 3 = signal level:
0 = set the signal to -12V (V.24 signal OFF).
1 = set the signal to +12V (V.24 signal ON).

Output parameters:
None

Note:
1. If not buffer terminal I/O controller then error code 240 is returned (ILLEGAL DEVICE TYPE).

Function code 16$_8$:

Function description:
Set terminal interface in test mode.
Test mode will loop the transmitted data back to receive data.
Data will also be transmitted to the terminal line.
If the test mode is selected for one of the four interfaces, all four will be set in test mode.

Input parameters:
Word 1 = logical device number.
Word 2 = terminal mode:
0 = set terminal in test mode.
1 = reset terminal from test mode.

Output parameters:
None

Function code 100$_8$:

Function description:
Return function parameters as set by one of the functions 101-177.
Values from SINTRAN memory, image and save areas are returned.

The functions 101 to 177 have the same call format:

Word 1 = logical device number.
Word 2 = area to be updated.
Word 3 and upwards are parameters for specific functions.

When returning function parameters, word 3 is returned with number of the parameters returned from each area (value 1,2 or 3):

Value 1 means: Value of word 3 as set by specific function is returned from SINTRAN memory, image and save area (for instance function 101).

Value 2 means: Value of word 3 and 4 as set by specific function is returned from SINTRAN memory, image and save area (for instance function 102).

Value 3 means: Value of word 3, 4 and 5 as set by specific function is returned from SINTRAN memory, image and save area (for instance function 103).

Input parameters:
Word 1 = logical device number.
Word 2 = function code to return function parameters from.

Output parameters:
Case 1:
Word 1 = unchanged.
Word 2 = unchanged.
Word 3 = 1
Word 4 = memory area (value set from input word 3 to other functions).
Word 5 = image area (value set from input word 3 to other functions).
Word 6 = save area (value set from input word 3 to other functions).

Case 2:
Word 1 = unchanged.
Word 2 = unchanged.
Word 3 = 2
Word 4 = memory area (value set from input word 3 to other functions).
Word 5 = memory area (value set from input word 4 to other functions).
Word 6 = image area (value set from input word 3 to other functions).
Word 7 = image area (value set from input word 4 to other functions).
Word 8 = save area (value set from input word 3 to other functions).
Word 9 = save area (value set from input word 4 to other functions).

Case 3:
Word 1 = unchanged.
Word 2 = unchanged.
Word 3 = 3
Word 4 = memory area (value set from input word 3 to other functions).
Word 5 = memory area (value set from input word 4 to other functions).
Word 6 = memory area (value set from input word 5 to other functions).
Word 7 = image area (value set from input word 3 to other functions).
Word 8 = image area (value set from input word 4 to other functions).
Word 9 = image area (value set from input word 5 to other functions).
Word 10= save area (value set from input word 3 to other functions).
Word 11= save area (value set from input word 4 to other functions).
Word 12= save area (value set from input word 5 to other functions).

Note:
Values from SINTRAN image and save areas are only returned for user SYSTEM.

Function code 101₈:

See also MSTTY (MON 17).

Function description:
Set terminal type.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = terminal type, as A-reg in MSTTY (MON 17).

Rules:
1. If logical device number is a TAD, the SINTRAN image and save areas may not be updated.

Function code 102₈:

See also MSDAE (MON 227).

Function description:
Set escape and local character.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = escape character, -1 = no changes.
Word 4 = local character, -1 = no changes.

Rules:
1. If logical device number is a TAD, the SINTRAN image and save areas may not be updated.

Function code 103₈:

Function description:
Start and stop Xon/Xoff protocol input control.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = start/stop Xon/Xoff protocol:
0 = start Xon/Xoff protocol
1 = stop Xon/Xoff protocol.
Word 4 = Xon character, -1 = no changes.
Word 5 = Xoff character, -1 = no changes.

Rules:
1. New Xon/Xoff characters may be defined only when start Xon/Xoff protocol is selected (word 3 = 0).

Function code 104₈:

Function description:
Start and stop Xon/Xoff protocol, output control.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = start/stop Xon/Xoff protocol:
0 = start Xon/Xoff protocol
1 = stop Xon/Xoff protocol.
Word 4 = Xon character, -1 = no changes.
Word 5 = Xoff character, -1 = no changes.

Rules:
1. New Xon/Xoff characters may be defined only when start Xon/Xoff protocol is selected (word 3 = 0).

Notes:
1. Whenever output control functions are used in memory area, output driver is restarted if currently in stop (Xoff).

Function code 105₈:

Function description:
Set Xon/Xoff only or dual function Xon/Xoff control.
- Xon/Xoff only means ordinary Xon/Xoff protocol.
- Dual function Xon/Xoff is an alternative to normal Xon/Xoff checking and requires Xon/Xoff characters to be in sequence, any other combination will be treated as normal data characters.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = type of Xon/Xoff control:
0 = Xon/Xoff only.
1 = Dual function Xon/Xoff.

Function code 106$_8$:

Function description:
Set character length.
If memory area is updated, then the function immediately set to terminal interface (IOXT) is executed.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = character length:
0 = 8-bit character length
1 = 7-bit character length
2 = 6-bit character length
3 = 5-bit character length.
Word 4 = parity:
0 = no parity
1 = even parity.

Function code 107$_8$:

Function description:
Set baud rate on terminal.
If memory area is updated, then the function immediately set to terminal interface (IOXT) is executed.
When returning function parameters for this function, words 3 and 4 will be returned with value 0 if no baud rate is specified in software (data field).

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = Baud rate input, 0= no changes.
Word 4 = Baud rate output, 0= no changes.

Rules:
1. Not legal for device number 1 (console).
2. Baud rate will not be changed in SINTRAN memory area if no baud rate is specified in software (data field).
3. If no baud rate is specified in SINTRAN image/save areas, then both input and output baud rate must be specified.
4. 19200 baud only legal if buffer terminal interface with FIFO.

The following baud rates may be specified (decimal):
50, 75, 110, 134 (134.5), 150, 200, 300,
600, 1200, 1800, 2400, 4800, 9600, 19200.

Notes:
1. Only the interface ND-102740 is currently supporting 19200 baud. If this function is used on the interface ND-102730, the baud rate will be set to 100 baud (refer to the manual "8-Terminal Buffer Interface with FIFO", ND-11.022.1 EN for further details).

Function code 110$_8$:

Function description:
Set number of stop bits.

If memory area is updated, then the function immediately set to terminal interface (IOXT) is executed.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = number of stop bits:
0 = 1.5 stop bit for 5 bits character,
2 stop bits else.
1 = 1 stop bit.

Function code 111$_8$:

See also function code 12.

Function description:
Set terminal to printer or reset printer to terminal.
If set terminal to printer, then logical device number specified in word 1 will no longer act as a terminal.
If reset from printer to terminal, then logical device will act as ordinary terminal again.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = set/reset terminal to printer:
0 = set terminal to printer (remove terminal data field from background table).
1 = reset from printer to terminal (reinsert terminal data field into background table).
Word 4 = character length.
0 = 7–bit character length.
1 = 8–bit unmodified input and output.

Rules:
1. Only allowed for user SYSTEM.
2. Both input and output part of logical device must be reserved (even if user SYSTEM).
3. Only legal for terminals with 8–bit character length to set 8–bit unmodified input/output.
4. Only legal to set 8–bit unmodified input/output when function is set terminal to printer (word 3 = 0).

Function code 112$_8$:

Function description:
Set half or full duplex on terminal.
Half duplex means that no echo is given to terminal when typing on the keyboard.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = duplex function:
0 = half duplex
1 = full duplex

Function code 113$_8$:

Function description:
Enable/disable login with variable terminal speed.
When login with variable speed is enabled, the terminal is enabled to log in if speed setting is one of: 9600, 4800, 2400, 1800, 1200, 600, 300, 200, 150, 134.5, 110, 75 or 50 baud. The user should press <ESCAPE> in intervals of 2-3 seconds until SINTRAN has typed "ENTER:". If a user has not logged into SINTRAN within 30 seconds after "ENTER:" has been typed, the terminal will be aborted.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = 0 = enable login with variable speed.
1 = disable login with variable speed.

Rules:
1. Not legal for device number 1 (console).
2. Baud rate must be set by software (TSPEED in data field ≠ -1).
3. Escape character must have ASCII value 33B.

Function code 114$_8$:

Function description:
Set/reset printer connected to terminal.

Input parameters:
Word 1 = logical device number.
Word 2 = AREA (see table on page 19).
Word 3 = 0 = set terminal connected to printer.
1 = reset terminal connected to printer.

3.2 NEW MONITOR CALLS

3.2.1 RSREC MON 340

Purpose: Read system record.

Monitor call format:

```
        LDA RECTP       % A = record type
        LDT RTADR       % T = RT-description / segment number
        LDX (BUFFR      % X = address of buffer
        MON 340         % RSREC
        JMP ERROR       % Error return
        .........       % Normal return

RECTP,  1
RTADR,  54214
BUFFR, 0; *+26/         % Reserve 26₈ words
```

Input parameters:

A-register: Record type: 1 = RT-description
2 = Segment entry
T-register: RT-description address or
Segment number
X-register: Address of buffer to receive system record.
This buffer must have a minimum length of 22_{10} words
(function = 1) or 12_{10} words (function = 2)

Output parameters:

Return: Error, A-register contains error code (one of 153, 174 or 201)
Skip return: OK, the system record is read into the specified buffer.

3.2.2 SGMTY MON 341

Purpose: SGMTY is used to change the active segments of a program and/or the page index tables:

Monitor call format:

```
        LDT (PARLI      % T = address of parameter list
        MON 341         % SGMTY

PARLI,  (FUNC           % Function code
        (ADDR           % Start/return address
        (SEG1           % New segment 1
        (SEG2           % New segment 2
        (PITS           % New PITs
```

```
Input parameters:   FUNC = 0 - MCALL
                          1 - MEXIT
                          2 - MCALL w/PIT change
                          3 - MEXIT w/PIT change
                          4 - REMSG (remove segment entered by ENTSG)
```

For function 4 (REMSG), parameters 2 (ADDR), 4 (SEG2) and 5 (PITS) must be zero.

```
Output parameters:
Functions 0 and 2 (MCALL):  T = Old segment 1
                            D = Old segment 2
                            L = Return address
                            X = Old PITs if PITs changed (function 2)
Functions 1 and 3 (MEXIT):  No register change
Function  4       (REMSG):  A = 0 : ok
                               -1 : segment not entered
                               -2 : illegal segment number
                               -3 : no segments entered
                               -4 : illegal parameter
```

3.2.3 ADP MON 342

Purpose: MON ADP is used to handle a system LAMU from a program or a set of programs. It is intended for use by ADP software, containing commonly used routines.
MON ADP is reserved for internal use by ND.

Monitor call format:

```
        LDT FUNCT       % T = function code
        LDA ..          % A = function dependent parameter
        LDX ..          % X = function dependent parameter
        MON 342         % ADP
        .........       % return
```

The following functions are available:

Function	Brief explanation
1	Go to program LAMU
2	Go to subsystem
3	Connect to mailbox LAMU (create it if necessary)
4	Disconnect from mailbox LAMU
5	Delete mailbox LAMU
6	Create program LAMU (a new system LAMU)
7	Delete program LAMU (a system LAMU)
10_8	Connect to program LAMU (a system LAMU)
11_8	Disconnect from program LAMU (a system LAMU)
12_8	Set write protection on program LAMU
13_8	Clear write protection on program LAMU
14_8	Disconnect from both mailbox LAMU and program LAMU

MON 342 is only allowed from background programs.

The following error codes may be returned:

A-reg.	Explanation
25	You are not authorized to do this
201	Illegal function code
373	Illegal program LAMU identifier
374	Program LAMU already exists
375	No such program LAMU
376	Illegal program LAMU size
377	Program LAMU not connected
3200	Not allowed now

Function code 1:

Function description:
Go to program LAMU.

Input parameters:
T-register = 1
A-register = program LAMU number (bits 15_8-12_8)
routine number (bits 11_8-0)
X-register = address of parameters in mailbox LAMU

Output parameters:
T-register = unchanged
A-register = unchanged
X-register = unchanged
L-register = return address after monitor call

Rules:
1. Permitted from all users.
2. If an error occurs in function 1, the program will be aborted.

Function code 2:

Function description:
Go to subsystem.

Input parameters:
T-register = 2
A-register = return address
X-register = address of parameters in mailbox LAMU

Output parameters:
T-register = unchanged
A-register = unchanged
X-register = unchanged

Rules:
1. Permitted from all users.
2. If an error occurs in function 2, the program will be aborted.

Function code 3:

Function description:
Connect mailbox LAMU (create it if necessary).

Input parameters:
T-register = 3
A-register = not used

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from all users.

Function code 4:

Function description:
Disconnect mailbox LAMU.

Input parameters:
T-register = 4
A-register = not used

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from all users.

Function code 5:

Function description:
Delete mailbox LAMU.

Input parameters:
T-register = 5
A-register = not used

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Only permitted internally from SINTRAN III.

Function code 6:

Function description:
Create program LAMU.

Input parameters:
T-register = 6
A-register = program LAMU number
X-register = program LAMU size in pages

Output parameters:
T-register = unchanged
A-register = status (0 = ok)
X-register = unchanged

Rules:
1. Permitted from user SYSTEM only.

Function code 7:

Function description:
Delete program LAMU.

Input parameters:
T-register = 7
A-register = program LAMU number

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from user SYSTEM only.

Function code 10_8:

Function description:
Connect program LAMU as data bank.

Input parameters:
T-register = 10
A-register = program LAMU number

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from user SYSTEM only.

Function code 11₈:

Function description:
Disconnect program LAMU.

Input parameters:
T-register = 11
A-register = program LAMU number

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from user SYSTEM only.

Function code 12₈:

Function description:
Write protect program LAMU.

Input parameters:
T-register = 12
A-register = program LAMU number

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from user SYSTEM only.
2. must be done before a program LAMU can be used

Function code 13₈:
Write permit program LAMU.

Function description:

Input parameters:
T-register = 13
A-register = program LAMU number

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Permitted from user SYSTEM only.

Function code 14₈:

Function description:
Disconnect both mailbox LAMU and current program LAMU.

Input parameters:
T-register = 14
A-register = not used

Output parameters:
T-register = unchanged
A-register = status (0 = ok)

Rules:
1. Only permitted internally from SINTRAN III.

3.2.4 CONFG MON 343

Purpose: CONFG is used to read and/or change configuration parameters for SINTRAN III/VSX.

Monitor call format:

```
        LDA (PARLI      % A = address of parameter list
        MON 343         % CONFG
        JMP ERROR       % Error handling
        .........       % normal return

PARLI,  (FUNC           % Function code
        (INDEX          % Configuration parameter number
        (SUBIN          % Subindex (only used for some values of INDEX)
        (VALUE          % Input and/or output value (integer or string)
```

The following values of the function code are used, but not all functions apply to all configuration parameters:

```
FUNC = 1 : (Save)       Read value from SINTRAN III save area
                        (next value to be used after a cold start)
       2 : (Read)       Read current active value
       3 : (Write)      Write value to SINTRAN III save area
                        (next value to be used after a cold start)
       4 : (Generated)  Read generated value
       5 : (Free)       Read currently unused units
       6 : (Special)    Parameter dependant
```

MON 343 is restricted to user SYSTEM only, and is allowed from RT-programs.

Note that MON 343 is generally intended to be used by the reconfiguration program (S3-CONFIG).

Also note that only some functions are available in the VSE-version.

Furthermore, note that a cold start is necessary to make changes come into effect (all changes are made as "write new value to save area").

The following configuration parameters may be specified (note that only a few functions are available on the VSE-version):

Number	Parameter name	Possible functions: Save/Read/Write/Gen/Free/sPecial VSX	VSE
0	Standard system	RG	G
1	Number of device buffers	SRWG	G
2	First legal physical page for device buffers	SRW	
3	Number of TADs	SRWG	G
4	Number of batch processors	SRWG	G
5	Number of spooling programs	SRWG	G
6	Number of background programs	SRWG	G
7	Number of background programs in use	R	
10_8	Number of symbolic debugger segments	SRWG	G
11_8	Number of ND-500 processes	SRWG	G
12_8	Number of remote file access segments	SRWG	G
13_8	Spooling queue size in pages	SRW	G
14_8	System segment size	SRW	G
15_8	Mon ADP (MON 341)	SRWG	G
16_8	Background allocation	G	G
17_8	COSMOS spooling	G	G
20_8	Number of user RT-programs	GF	G
21_8	Number of user segments	GF	G
22_8	Number of terminals	RG	G
23_8	Number of telefix devices	G	G
24_8	Number of semaphores	G	G
25_8	Number of internal devices (total)	G	G
26_8	Number of internal devices (block)	G	G
27_8	Number of SIBAS processes	G	G
30_8	Number of open file entries	R	R
31_8	Number of allocated areas	SRW	G
32_8	Mon MLAMU (MON 315)	G	G
33_8	Maximum number of LAMUs	SRW	
34_8	Maximum number of LAMUs per program	SRW	
35_8	Maximum number of system LAMUs	SRW	
36_8	ND-500 software versions	R	
37_8	ND-500 microcode version (per CPU)	R	
40_8	ND-110 microprogram version	R	
41_8	Memory configuration	R	
42_8	Define HDLC-interfaces as HDLC or modem	SRWP	
43_8	Number of HDLC connections	G	
44_8	Number of synchronous modems on HDLC	G	
45_8	Number of X.21 connections	SRWG	
46_8	Define spooling device number	SRW	
47_8	Define printer type	SRWP	
50_8	Number of Fast UDMA RT-programs	SRWG	

The following error codes may be returned:

A-reg.	Explanation
25	You are not authorized to do this
33	No such logical unit
174	Illegal parameter
201	Illegal function code (which means "illegal function code for this index")
3201	Illegal index (which means "illegal index or subindex")

Other interpretations of error codes 201 and 3201 are given when applicable.

Configuration parameter: 0

Parameter name:
Standard system

Input parameters:
FUNC = Function, see below.
INDEX = 0
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = work mode version (if function = Read current active value)
standard system (if function = Read generated value):
1 = standard system - all disk types are included
0 = system is generated to meet specific configuration requirements

Functions allowed for this parameter:
Read current active value
Read generated value

Configuration parameter: 1

Parameter name:
Number of device buffers

Input parameters:
FUNC = Function, see below.
INDEX = 1
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of device buffers

Output parameters:
VALUE = Number of device buffers

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 2

Parameter name:
First legal physical page for device buffers

Input parameters:
FUNC = Function, see below.
INDEX = 2
SUBIN = Subindex not used for this configuration parameter
VALUE = First legal physical page for device buffers

Output parameters:
VALUE = First legal physical page for device buffers

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 3

Parameter name:
Number of TADs

Input parameters:
FUNC = Function, see below.
INDEX = 3
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of TADs

Output parameters:
VALUE = Number of TADs

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 4

Parameter name:
Number of batch processors

Input parameters:
FUNC = Function, see below.
INDEX = 4
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of batch processors

Output parameters:
VALUE = Number of batch processors

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 5

Parameter name:
Number of spooling programs

Input parameters:
FUNC = Function, see below.
INDEX = 5
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of spooling programs

Output parameters:
VALUE = Number of spooling programs

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 6

Parameter name:
Number of background programs.

Input parameters:
FUNC = Function, see below.
INDEX = 6
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of background programs

Output parameters:
VALUE = Number of background programs

Functions allowed for this parameter:
Read save area (next value) (*)
Read current active value (*)
Write to save area (new next value) (*)
Read generated value

The three functions marked (*) are only available when Background Allocation is present (see configuration parameter 16_8).

Configuration parameter: 7

Parameter name:
Number of background programs in use

Input parameters:
FUNC = Function, see below.
INDEX = 7
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of background programs in use

Functions allowed for this parameter:
Read current active value

Configuration parameter: 10_8:

Parameter name:
Number of symbolic debugger segments

Input parameters:
FUNC = Function, see below.
INDEX = 10
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of symbolic debugger segments

Output parameters:
VALUE = Number of symbolic debugger segments

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 11₈:

Parameter name:
Number of ND-500 processes

Input parameters:
FUNC = Function, see below.
INDEX = 11
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of ND-500 processes

Output parameters:
VALUE = Number of ND-500 processes

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 12₈:

Parameter name:
Number of remote file access segments

Input parameters:
FUNC = Function, see below.
INDEX = 12
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of remote file access segments

Output parameters:
VALUE = Number of remote file access segments

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 13₈:

Parameter name:
Spooling queue size in pages for each spooling program.
A queue size of 2 pages can contain 10 queue entries and each additional page will increase the queue length by approximately 7 new entries.

Input parameters:
FUNC = Function, see below.
INDEX = 13
SUBIN = Subindex not used for this configuration parameter
VALUE = Spooling queue size in pages

Output parameters:
VALUE = Spooling queue size in pages

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 14_8:

Parameter name:
System segment size.
This size will affect the number of open files for each user as follows: default system segment size (5) provides 48 file buffers, each additional page (up to total max. of 8) provides 16 buffers.
Each open file uses 2 buffers if sequential access, 1 if random access. Maximum number of open files is 64.

Input parameters:
FUNC = Function, see below.
INDEX = 14
SUBIN = Subindex not used for this configuration parameter
VALUE = System segment size

Output parameters:
VALUE = System segment size

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 15_8:

Parameter name:
Mon ADP (MON 341)

Input parameters:
FUNC = Function, see below.
INDEX = 15
SUBIN = Subindex not used for this configuration parameter
VALUE ≠ 0 if MON ADP (MON 341) is to be available
(only possible if this SINTRAN is generated with MON ADP)
0 if MON ADP (MON 341) is to be unavailable

Output parameters:
VALUE = 1 if MON ADP (MON 341) is available
0 if MON ADP (MON 341) is unavailable

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 16₈:

Parameter name:
Background allocation

Input parameters:
FUNC = Function, see below.
INDEX = 16
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = 1 if this SINTRAN is generated with Background Allocation
0 if this SINTRAN is not generated with Background Allocation

Functions allowed for this parameter:
Read generated value

Configuration parameter: 17₈:

Parameter name:
Cosmos spooling

Input parameters:
FUNC = Function, see below.
INDEX = 17
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = 1 if COSMOS spooling is available
0 if COSMOS spooling is unavailable

Functions allowed for this parameter:
Read generated value

Configuration parameter: 20₈:

Parameter name:
Number of user RT-programs

Input parameters:
FUNC = Function, see below.
INDEX = 20
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of user RT-programs

Functions allowed for this parameter:
Read generated value
Report currently free units (in memory)

Configuration parameter: 21$_8$:

Parameter name:
Number of user segments

Input parameters:
FUNC = Function, see below.
INDEX = 21
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of user segments

Functions allowed for this parameter:
Read generated value
Report currently free units (in memory)

Configuration parameter: 22$_8$:

Parameter name:
Number of terminals

Input parameters:
FUNC = Function, see below.
INDEX = 22
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of interfaces (if read current active value)
Maximum number of terminals possible to use (if read generated value)

Functions allowed for this parameter:
Read current active value
Read generated value

Configuration parameter: 23$_8$:

Parameter name:
Number of telefix devices

Input parameters:
FUNC = Function, see below.
INDEX = 23
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of telefix devices

Functions allowed for this parameter:
Read generated value

Configuration parameter: 24₈:

Parameter name:
Number of semaphores

Input parameters:
FUNC = Function, see below.
INDEX = 24
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of semaphores

Functions allowed for this parameter:
Read generated value

Configuration parameter: 25₈:

Parameter name:
Number of internal devices (total)

Input parameters:
FUNC = Function, see below.
INDEX = 25
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of internal devices (total)

Functions allowed for this parameter:
Read generated value

Configuration parameter: 26₈:

Parameter name:
Number of internal devices (block)

Input parameters:
FUNC = Function, see below.
INDEX = 26
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of internal devices (block)

Output parameters:
VALUE = Input value not used for this configuration parameter

Functions allowed for this parameter:
Read generated value

Configuration parameter: 27₈:

Parameter name:
Number of SIBAS processes

Input parameters:
FUNC = Function, see below.
INDEX = 27
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of SIBAS processes

Functions allowed for this parameter:
Read generated value

Configuration parameter: 30₈:

Parameter name:
Number of open file entries

Input parameters:
FUNC = Function, see below.
INDEX = 30
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of open file entries

Functions allowed for this parameter:
Read current active value

Configuration parameter: 31₈:

Parameter name:
Number of allocated areas

Input parameters:
FUNC = Function, see below.
INDEX = 31
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of areas allocated for MON FIXC5 (MON 61)

Output parameters:
VALUE = Number of areas allocated for MON FIXC5 (MON 61)

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 32₈:

Parameter name:
Mon MLAMU (MON 315)

Input parameters:
FUNC = Function, see below.
INDEX = 32
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = 1 if MON MLAMU (MON 315) is available
0 if MON MLAMU (MON 315) is unavailable

Functions allowed for this parameter:
Read generated value

Configuration parameter: 33₈:

Parameter name:
Maximum number of LAMUs

Input parameters:
FUNC = Function, see below.
INDEX = 33
SUBIN = Subindex not used for this configuration parameter
VALUE = Maximum number of LAMUs

Output parameters:
VALUE = Maximum number of LAMUs

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 34₈:

Parameter name:
Maximum number of LAMUs per program

Input parameters:
FUNC = Function, see below.
INDEX = 34
SUBIN = Subindex not used for this configuration parameter
VALUE = Maximum number of LAMUs per program

Output parameters:
VALUE = Maximum number of LAMUs per program

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 35₈:

Parameter name:
Maximum number of system LAMUs

Input parameters:
FUNC = Function, see below.
INDEX = 35
SUBIN = Subindex not used for this configuration parameter
VALUE = Maximum number of system LAMUs

Output parameters:
VALUE = Maximum number of system LAMUs

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 36₈:

Parameter name:
ND-500 software versions

Input parameters:
FUNC = Function, see below.
INDEX = 36
SUBIN = 1 for ND-500 swapper, 2 for ND-500 system monitor
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = ND-500 software version (swapper or system monitor)
VALUE must be an array; minimum length is 4 words if SUBIN=1, otherwise 6.

Functions allowed for this parameter:
Read current active value

Special error messages:
Error 174 (Illegal parameter) may also mean that the ND-500 Swapper is not loaded or that the ND-500 System Monitor is not started.
Error 3201 (Illegal index) may also mean that there is no ND-500 in this system.

Configuration parameter: 37₈:

Parameter name:
ND-500 microcode version (per CPU)

Input parameters:
FUNC = Function, see below.
INDEX = 37
SUBIN = ND-500 CPU number (0-7)
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = ND-500 microcode version
0 = microcode not loaded

Functions allowed for this parameter:
Read current active value

Special error messages:
Error 174 (Illegal parameter) may also mean that the specified ND-500 CPU does not exist.
Error 3201 (Illegal index) may also mean that there is no ND-500 in this system.

Configuration parameter: 40₈:

Parameter name:
ND-110 microprogram version

Input parameters:
FUNC = Function, see below.
INDEX = 40
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = ND-110/CX microprogram version
0 = not ND-110/CX CPU

Functions allowed for this parameter:
Read current active value

Configuration parameter: 41$_8$:

Parameter name:
Memory configuration

Input parameters:
FUNC = Function, see below.
INDEX = 41
SUBIN = Sub-parameter, see table below.
VALUE = Input value not used for this configuration parameter

Sub-parameter:

0	Memory configuration (total)
1	ND-100 local (including PIOC)
2	PIOC (per PIOC)
3	MPM-3
4	MPM-4
5	MPM-5
6	For swapping
7	For SINTRAN
10$_8$	RT-common
11$_8$	Reserved by ND 500

Output parameters:
VALUE = Memory configuration (in pages)

Functions allowed for this parameter:
Read current active value

Configuration parameter: 42$_8$:

Parameter name:
Define HDLC-interfaces as HDLC or modem

Input parameters:
FUNC = Function, see below.
INDEX = 42
SUBIN = HDLC interface number (1-32 limited by configuration)
VALUE = Define HDLC-interfaces as HDLC or modem:
0 = this HDLC is not to be used
1 = HDLC
2 = Synchronous modem

Output parameters:
VALUE = Type of HDLC-interface (if function ≠ 6):
0 = this HDLC is not used
1 = HDLC
2 = Synchronous modem
= Logical device number (if function = 6)

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Special function: read logical device number of HDLC interface

Configuration parameter: 43$_8$:

Parameter name:
Number of HDLC connections

Input parameters:
FUNC = Function, see below.
INDEX = 43
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of HDLC connections

Functions allowed for this parameter:
Read generated value

Note:
The number of HDLC devices defined by configuration parameters 43 and 44 may add up to a total exceeding the number of HDLC devices generated. The reason for this is that some devices may be used for either HDLC or synchronous modems or both (but not at the same time).

Configuration parameter: 44$_8$:

Parameter name:
Number of synchronous modems on HDLC

Input parameters:
FUNC = Function, see below.
INDEX = 44
SUBIN = Subindex not used for this configuration parameter
VALUE = Input value not used for this configuration parameter

Output parameters:
VALUE = Number of synchronous modems on HDLC

Functions allowed for this parameter:
Read generated value

Note:
See note on configuration parameter 43.

Configuration parameter: 45$_8$:

Parameter name:
Number of X.21 connections

Input parameters:
FUNC = Function, see below.
INDEX = 45
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of X.21 connections

Output parameters:
VALUE = Number of X.21 connections

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

Configuration parameter: 46$_8$:

Parameter name:
Define spooling device number

Input parameters:
FUNC = Function, see below.
INDEX = 46
SUBIN = Spooling index (1-60, limited by configuration)
VALUE = Define spooling device number

Output parameters:
VALUE = Define spooling device number

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)

Configuration parameter: 47₈:

Parameter name:
Define printer type

Input parameters:
FUNC = Function, see below.
INDEX = 47
SUBIN = Printer number (1-4, limited by configuration)
VALUE = Define printer as line-printer/Fujitsu:
0 = Printer is not to be used
1 = DMA-interface (Fujitsu)
2 = Parallel interface (CDC / Dataproducts)
3 = Serial interface

Output parameters:
VALUE = Type of printer and interface (if function ≠ 6):
0 = Printer is not to be used
1 = DMA-interface (Fujitsu)
2 = Parallel interface (CDC / Dataproducts)
3 = Serial interface
= Logical device number (if function = 6)

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Special function: read logical device number of printer

Configuration parameter: 50₈:

Parameter name:
Number of Fast UDMA RT-programs

Input parameters:
FUNC = Function, see below.
INDEX = 50
SUBIN = Subindex not used for this configuration parameter
VALUE = Number of programs which may use the fast UDMA option simultaneously.

Output parameters:
VALUE = Number of Fast UDMA RT-programs

Functions allowed for this parameter:
Read save area (next value)
Read current active value
Write to save area (new next value)
Read generated value

4. SINTRAN-SERVICE-PROGRAM

4.1 COMMANDS REMOVED

4.1.1 *DEFINE-USER-MONITOR-CALL

The command *DEFINE-USER-MONITOR-CALL is removed in the VSX-version.

Any user-defined monitor calls must therefore be defined manually:

Find a suitable area for the code of the monitor call and insert it (just as in the previous versions, and in the VSE-version).

Then the monitor call must be defined in the monitor call tables:
MCTAB (address of monitor call code - 1 word per monitor call entry)
MPPTAB (PIT where the code is located - 1 byte per monitor call entry)
TMCTAB (type of monitor call - 1 byte per monitor call entry)
GOTAB (level 14 table - type of monitor call - 1 word per entry.
The entry for a user monitor call should contain the address of the common monitor call handling routine MFELL.)

4.1.2 *SET-CHANNEL-PRIORITY

4.1.3 *LIST-ADDRESSES

4.2 MODIFIED COMMANDS

4.2.1 *CHANGE-DATAFIELD

The following symbolic displacements are added in the K-version of SINTRAN III/VSE:

FBSIZ	(-13$_8$)	Size of XMSG buffers to use
NOBUF	(-14$_8$)	Number of XMSG buffers to use
TADTYP	(-21$_8$)	TAD type

The following symbolic displacements are added in the K-version of SINTRAN III/VSX:

FBSIZ	(-13$_8$)	Size of XMSG buffers to use
NOBUF	(-14$_8$)	Number of XMSG buffers to use
SNMIQ	(64$_8$)	Disk sorting: max. number of elements in queue
STIMC	(61$_8$)	Disk sorting: no. of seek not completed
STREN	(70$_8$)	Disk sorting: sorting enabled (1) or disabled (0)
SUNGL	(62$_8$)	Disk sorting: no. of seek time out
SUNIH	(65$_8$)	Qisk sorting: inhibit flag for seek
TADTYP	(-21$_8$)	TAD type
TDFLGADDR	(-3$_8$)	Logical address within page of data field
TDFPHPAGE	(-4$_8$)	Physical page of data field

4.2.2 *CHANGE-VARIABLE

The following symbolic variable names are added in the K-version of SINTRAN III/VSX:

```
DVBFPAGE     (first physical page of memory legal for device buffers)
MINSWPAGES   (minimum number of pages of memory for swapping)
MXDVBUF      (maximum number of device buffers in this system)
```

The following symbolic variable names are no longer supported in SINTRAN III/VSE:

```
BGLPAGE
BGFPAGE
CACHLIM
SAFILNO
```

The following symbolic variable names are no longer supported in SINTRAN III/VSX:

```
BGLPAGE
BGFPAGE
CACHLIM
CCFPAGE
CCLPAGE
CNVRT
ENDCOR
EXTDS
IDNTS
RTFPAGE
RTLPAGE
SAFILNO
TABLES
TMCTAB
```

4.2.3 *DEFINE-USER-RESTART-PROGRAM

This command now defines the first RT-program to be started after a power failure. Other RT-programs to be started are defined by the command *NEXT-USER-RESTART-PROGRAM.

There is no change to parameters: <RT NAME>
<MEMORY?>
<IMAGE?>
<SAVE AREA?>

4.2.4 *HELP

The order of the parameters are now swapped, and the parameters are no longer optional.

The parameter sequence is thus:

<COMMAND>	(default: all)
<OUTPUT FILE>	(default: TERMINAL)

4.2.5 *INSERT-SPOOLING-HEADER

On the VSX-version, this command may only be used to insert the spooling header in the memory area.
A warm start is thus necessary to put the command into effect.
There is no change to the parameters for the VSE-version.

The parameter sequence is thus: <SPOOLING INDEX>
for the VSX-version.

4.2.6 *LIST-SERVICE-COMMANDS

The order of the parameters are now swapped, and the parameters are no longer optional.

The parameter sequence is thus: <COMMAND> (default: all)
<OUTPUT FILE> (default: TERMINAL)

4.2.7 *OCTAL-DUMP

On the VSX-version, the first parameter (area) is changed.

The parameter sequence is thus: <SEGMENT OR ALT-SEG>
<SEGMENT (NAME OR NUMBER (OCT))>
<LOWER ADDRESS (OCT)>
<UPPER ADDRESS (OCT)>
<OUTPUT FILE>
for the VSX-version.

4.2.8 *READ-BINARY

On the VSX-version, the first parameter (area) is changed.

The parameter sequence is thus: <SEGMENT OR ALT-SEG>
<SEGMENT (NAME OR NUMBER (OCT))>
<INPUT FILE>
for the VSX-version.

4.2.9 *REMOVE-SINTRAN-COMMAND

The commands are no longer removed, only marked as unavailable. Commands may be restored as available by the command *REINSERT-SINTRAN-COMMAND.

There is no change to parameters: <COMMAND>
<MEMORY?>
<SAVE AREA?>

4.2.10 *REMOVE-SPOOLING-HEADER

On the VSX-version, this command may only be used to remove the spooling header from the image area.
There is no change to the parameters for the VSE-version.

The parameter sequence is thus: <SPOOLING INDEX>
<FORM FEED BEFORE?>
<FORM FEED AFTER?>
for the VSX-version.

Note that this change implies that removal of spooling header for a specific spooling device will not survive a warm start (on the VSX-version). The *REMOVE-SPOOLING-HEADER command(s) should therefore be included in the LOAD-MODE batch-file to be run after a warm start.

4.2.11 *SET-COMMAND-PROTECTION

This command now also works on file system commands with the restriction that file system commands can only be made more restricted (a public command may be made SYSTEM only, but not the other way around).

Furthermore, this command may now be used to set command protection on reentrant subsystems and ND-500 standard domains in just the same way as commands. ND-500 standard domains can only be protected when used directly from SINTRAN III; they may be started by the ND-500 Monitor.

4.3 NEW COMMANDS

4.3.1 *CREATE-SYSTEM-LAMU

Parameters: <LAMU ID>
<SIZE>
<PHYSICAL START PAGE>

Create a system-LAMU. A system-LAMU is a special kind of LAMU created in memory (taken from the swapping area). The memory reserved for a system-LAMU is released to be used for swapping when the LAMU is deleted.

4.3.2 *INSERT-PROGRAM-IN-TIME-SLICE

Parameters: <RT NAME>
<MEMORY?>
<IMAGE?>
<SAVE AREA?>
<TIMESLICE CLASS>

The specified RT-program is to be time sliced. It will thus no longer run on a fixed priority.
Default value for the parameter <timeslice class> is 0.

4.3.3 *LIST-USER-RESTART-PROGRAMS

List all RT-programs to be started after a power failure.

Parameter: <OUTPUT FILE>

The default value for the parameter output file is terminal.

4.3.4 *NEXT-USER-RESTART-PROGRAM

Define further RT-programs to be started after a power failure. The first RT-programs to be started after a power fail are defined by the command *DEFINE-USER-RESTART-PROGRAM. A maximum number of 10 RT-programs can be started automatically after a power failure.

Parameters: <RT NAME>
<MEMORY?>
<IMAGE?>
<SAVE AREA?>

4.3.5 *REINSERT-SINTRAN-COMMAND

Restore a SINTRAN III command to be available again after being made unavailable with the command *REMOVE-SINTRAN-COMMAND.

Parameters: <COMMAND>
<MEMORY?>
<SAVE AREA?>

4.3.6 *REMOVE-PROGRAM-FROM-TIME-SLICE

Parameters: <RT NAME>
<MEMORY?>
<IMAGE?>
<SAVE AREA?>

The specified RT-program is no longer time sliced. It will thus run on a fixed priority, but will continue to run on the priority it had the moment it was removed from time slicing. The priority may be changed by the command @PRIOR in the usual way.

5. CONFIGURATION PROGRAM

The SINTRAN III Configuration Program (ND-211024) is available to make changes in a generated system of SINTRAN III/VSX version K. The program is installed together with SINTRAN and is started by the @RECOVER command: @S3-CONFIG in just the same way as other programs. Use of the program is restricted to user SYSTEM only. The configuration program is a screen-oriented program, but may also be run on a hardcopy terminal.

The program has 9 commands: 4 commands for selecting a menu of configuration parameters which may be changed:

BACKGROUND **IO-COMM** **LAMU** **VARIOUS**

a command to display the value of parameters which may not be changed:

DISPLAY

and 4 utility commands to print a report of the current configuration, saving the changed configuration, getting help and exit:

PRINT **GENERATE** **HELP** **EXIT**

You use the arrow keys, (←) or (→), or the first letter of a command name to navigate between commands.
If you select a command which enables you to change parameters, use the return key (↵) to enter the menu of parameters, and use the arrow keys, (↑) or (↓), to navigate. When finished, use the EXIT key, or the "Home" key (\) to exit from the menu.
The HELP key may be used at all times to get information about the current configuration parameter, etc.

5.1 THE UTILITY COMMANDS

The **PRINT** command will print the current configuration to a file. You will be asked for output file and this parameter has no default value.

The **GENERATE** command will save the changes you have made to the current configuration of SINTRAN III in a configuration file. It will also update the save-area of SINTRAN. You will be asked to confirm that you want to save the changes. Note that you must do a cold start to activate the changes because only the save-area is changed.
This command may be given as part of the @RECOVER command to run the configuration program: @S3-CONFIG GENERATE. This feature may be used to reconfigure SINTRAN in exactly the same way as before, after loading from diskettes, for example when installing a new patch file.

The **EXIT** command is used to exit from the configuration program. If you have made any changes to the configuration, and the changes have not been saved (by the GENERATE command), you will be asked if you want to save the changes.

The **HELP** command will give a brief explanation of the 4 commands used to select menus of configuration parameters which may be changed.

5.2 THE SELECTION COMMANDS

The **BACKGROUND** command will display a menu of configuration parameters related to background processes. The menu may look like this:

```
.....................................Current..Next....Max...(Input)
Number of TADs                           6       6      50       6
Number of batch processors               3       3      10       3
Number of spooling programs              4       4      20       4
Number of background programs           32      32     151      32
Background allocation                Present
Number of symbolic debugger segments     8       8      32       8
Number of ND-500 processes              21      21     201      21
Number of remote file access segments   16      16      50      16
System segment size                      5       5               5
Mon ADP                                  0       0       1       0


<Exit>,<Home> or . : return          <Help> or ? : field information
```

The configuration parameters listed in this menu are:

Number of TADs : Number of terminals on remote systems to use this system simultaneously.

Number of batch processors : Number of batch jobs to be run at a time.

Number of spooling programs: Number of printers used simultaneously.

Number of background programs : Number of terminals to be used simultaneously, if background allocation is present. Should be set to number of TADs + number of terminals + number of telefix devices.

Background allocation : Indicates if the background allocation system is in use. This parameter can not be changed.

Number of symbolic debugger segments : Number of simultaneous users of the Symbolic Debugger.

Number of ND-500 processes : Number of simultaneous users of the ND-500 part. Should usually be equal to "number of background programs" + 1.

Number of remote file access segments : Number of simultaneous users of remote file access.

System segment size : Size (in pages) of the system segment. This size will affect the number of open files for each user as follows: default size (5) provides 48 file buffers, each additional page (up to total max. of 8) provides 16 buffers. Each open file uses 2 buffers if sequential access, 1 if random access. Maximum number of open files are 64.

Mon ADP : Indicates if MON ADP (MON 342) is to be available.

Note: if the Background Allocation System is not present, the **Next** and **Input** fields of *Number of background programs* will not be used.

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start and the **Max** value is the maximum generated for this version of SINTRAN. All parameters may be changed, you use the return key (↵) to enter the menu, and the arrow keys, (↑) and (↓), to navigate. When finished, use the EXIT key or the HOME key (\) to exit.

The **IO-COMM** command will display a menu of some configuration parameters. The menu may look like this:

```
.....................................Current..Next....Max...(Input)
Number of HDLC connections                                  12
Number of synchronous modems on HDLC                         6
Number of X.21 connections                    0        0     2      0

Define spooling device numbers                                      x
Define HDLC interface as HDLC or modem                              x
Define printer type                                                 x

                                                      Type <> or E to
                                                     edit these tables

<Exit>,<Home> or .  : return          <Help> or ?  : field information
```

The configuration parameters listed in this menu are:

Number of HDLC connections : Number of HDLC connection generated for this system.

Number of synchronous modems on HDLC : Number of HDLC connections which may be used for synchronous modems.

Number of X.21 connections : Number of X.21 connections.

Define spooling device numbers

Define HDLC interface as HDLC or modem

Define printer type

Use the (<>) key or E to enter sub-menus for these parameters. See below.

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start and the **Max** value is the maximum generated for this version of SINTRAN. All parameters may be changed, you use the return key (↵) to enter the menu, and the arrow keys, (↑) and (↓), to navigate. When finished, use the EXIT key or the HOME key (\) to exit.

The 3 last configuration parameters in the menu **IO-COMM**, contains tables of values, and when you select one of these, you will get a sub-menu on your terminal.

The sub-menu for the parameter **Define spooling device numbers** may look like this:

```
Define spooling device numbers.  Enter a logical device number.

Spooling..Current..Next..(Input).....Spooling..Current..Next..(Input)
    1        5        5        5          2        59       59       59
    3        0        0        0          4        0        0        0
    5        0        0        0          6        0        0        0
    7        0        0        0          8        0        0        0
    9        0        0        0         10        0        0        0
   11        0        0        0         12        0        0        0
   13        0        0        0         14        0        0        0
   15        0        0        0         16        0        0        0
   17        0        0        0         18        0        0        0
   19        0        0        0         20        0        0        0

<Exit>/<Home>/.  :return  <Help>/?  :field information  N  :next
```

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start.

If you need to set spooling device numbers for spooling processes 21-40 or 41-60, similar sub-menus will appear for these. Either navigate "past" 20 (40), or select next menu (N) key.

The sub-menu for the parameter **Define HDLC interface as HDLC or modem** may look like this:

```
Define HDLC interface as HDLC or modem.
                                                0=Do not use this interface.
                                                1=HDLC.
                                                2=Synchronous modem.
..HDLC...Current...Next...(Input).....HDLC...Current...Next...(Input)
    1        1        1        1          2        1        1        1
    3        1        1        1          4        1        1        1
    5        1        1        1          6        1        1        1
    7        0        0        0          8        0        0        0
    9        0        0        0         10        0        0        0
   11        0        0        0         12        0        0        0
   13        0        0        0         14        0        0        0
   15        0        0        0         16        0        0        0
   17        0        0        0         18        0        0        0
   19        0        0        0         20        0        0        0

<Exit>/<Home>/.  :return  <Help>/?  :field information  N  :next
```

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start.

If you need to set status on HDLC connections 21-32, a similar sub-menu will appear for these. Either navigate "past" 20, or select next menu (N) key.

The sub-menu for the parameter **Define printer type** may look like this:

```
  Define printer type.

..Printer...Current...Next...(Input).....................................
    1          1         1         1
    2          0         0         0
    3          0         0         0
    4          0         0         0

    Types:  0 - Do not use this printer.
            1 - DMA          (Used for Fujitsu)
            2 - Parallel     (Used for CDC/DP)
            3 - Serial


<Exit>/<Home>/.  :return  <Help>/?  :field information
```

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start.

All parameters may be changed, you use the arrow keys (↑) or (↓) to navigate. When finished, use the EXIT key or the HOME key (↖) to exit.

The **LAMU** command will display a menu of some configuration parameters. The menu may look like this:

```
.....................................Current..Next....Max...(Input)
Mon MLAMU                             Present
Max number of LAMUs                     32       32              32
Max number of LAMUs per program          2        2               2
Max number of system LAMUs              16       16              16


<Exit>,<Home> or .  : return          <Help> or ? : field information
```

Only the first line of the menu will be shown if Mon MLAMU is not present.

The configuration parameters listed in this menu are:

Mon MLAMU : Indicates if MON MLAMU (MON 315) is to be present or not.

Max number of LAMUs : Maximum number of LAMUs (a LAMU is a reserved part of memory).

Max number of LAMUs per program : Maximum number of LAMUs accessible from a single program.

Max number of system LAMUs : Maximum number of system LAMUs (further information on system LAMUs is given on pages 16, 38-43 and 65.

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start.

All parameters may be changed, you use the return key (↵) to enter the menu, and the arrow keys, (↑) and (↓), to navigate. When finished, use the EXIT key or the HOME key (\) to exit.

The **VARIOUS** command will display a menu of some configuration parameters. The menu may look like this:

```
.....................................Current..Next....Max...(Input)
Number of device buffers                64      64      64      64
First legal phys. page for device buffer 0       0               0
Spooling queue size in pages            4       4               4
Number of allocated areas               40      40              40
Number of fast UDMA programs            0       0       0       0

<Exit>,<Home> or . : return          <Help> or ? : field information
```

The configuration parameters listed in this menu are:

Number of device buffers : Number of device buffers.

First legal phys. page for device buffer : First physical page in memory for device buffers.

Spooling queue size in pages : Size of each spooling queue - a queue size of 2 pages can contain 10 queue entries and each additional page will increase queue length by approximately 7 new entries.

Number of allocated areas : Number of areas in memory reserved by the monitor call MON FIXC5 (MON 61). Should be larger than number of system LAMUs.

Number of fast UDMA programs : Number of RT-programs to use fast UDMA.

The **Current** value is the value currently used by SINTRAN, the **Next** value is the value which will apply after the next cold start and the **Max** value is the maximum generated for this version of SINTRAN.

All parameters may be changed, you use the return key (↵) to enter the menu, and the arrow keys, (↑) and (↓), to navigate. When finished, use the EXIT key or the HOME key (\) to exit.

5.3 THE DISPLAY COMMAND

The **DISPLAY** command will show the current values of some configuration parameters which are not changeable.

The screen picture may look like this:

```
.....................................................Current.....Max.....
Number of user RT programs                             128
Number of user segments                                734
Number of terminals                                     25         128

Number of semaphores                                    50
Number of internal devices (total)                      30
Number of internal devices (block)                       2
Number of SIBAS processes                               12
Number of open file entries                             48
COSMOS spooling                                        Yes
Number of telefix devices                                1

Work mode version                                   000100B
Standard system                                        Yes
```

The configuration parameters listed in this command are:

Current value means:

Number of user RT programs : Number of free RT-descriptions.

Number of user segments : Number of free segments.

Number of terminals : Number of terminals used.

Number of semaphores : Number of semaphores generated.

Number of internal devices (total) : Total number of internal devices (generated value).

Number of internal devices (block) : Number of block oriented internal devices (generated value)

Number of SIBAS processes : Number of SIBAS processes generated.

Number of open file entries : Number of files open simultaneously (generated value).

COSMOS spooling : Indicates if COSMOS spooling is present.

Number of telefix devices : Number of telefix devices (generated val).

Work mode version : Version of work mode used when generating this SINTRAN system (for internal use by ND).

Standard system : Indicates if this SINTRAN is a standard system.

The **Max** value appearing for the *Number of terminals* parameter means maximum number of terminals supported by this system (generated value).

6. FILE SYSTEM

6.1 INTRODUCTION

One of the major drawbacks of the SINTRAN III file system is the limited number of files (objects) allowed under each user area. In order to overcome this restriction, the internal structure of the file system has been changed to support more than 256 files per user. The following pages will describe the new directory structure, how the object file is expanded, and changes in the data structures.

6.2 THE NEW DIRECTORY STRUCTURES

The main part of the directory structure is unchanged, i.e., the directory entry and the user file are identical to the previous versions. The object file may be extended when a user creates more than 256 files and it will then be subindexed even if there is no user with user index exceeding 63. The file system will automatically establish a subindexed structure when user 64 is created, or when the first object with index exceeding 255 is created by any user. This subindexed structure is illustrated in the figure on the next page.

All files belonging to a user are divided into blocks of 256 objects. Whenever creating file number 257, 513 and so on, a new index block is allocated for this user. The maximum number of index blocks a user may have is 16, which allows a maximum of 4096 files. The reason for this limit is the number of vacant bits in the object entry to specify the object block number of the current, next and previous versions of the file.

All users may have 4096 files. However, the maximum number of files can be set separately for each user area. A 4-bit word (MXOBL) in the user entry is used to specify how many extra object blocks the user is allowed to have. If MXOBL is zero, the user can only have 256 files, and if it is zero for all users, the file system will work as in previous versions of SINTRAN.

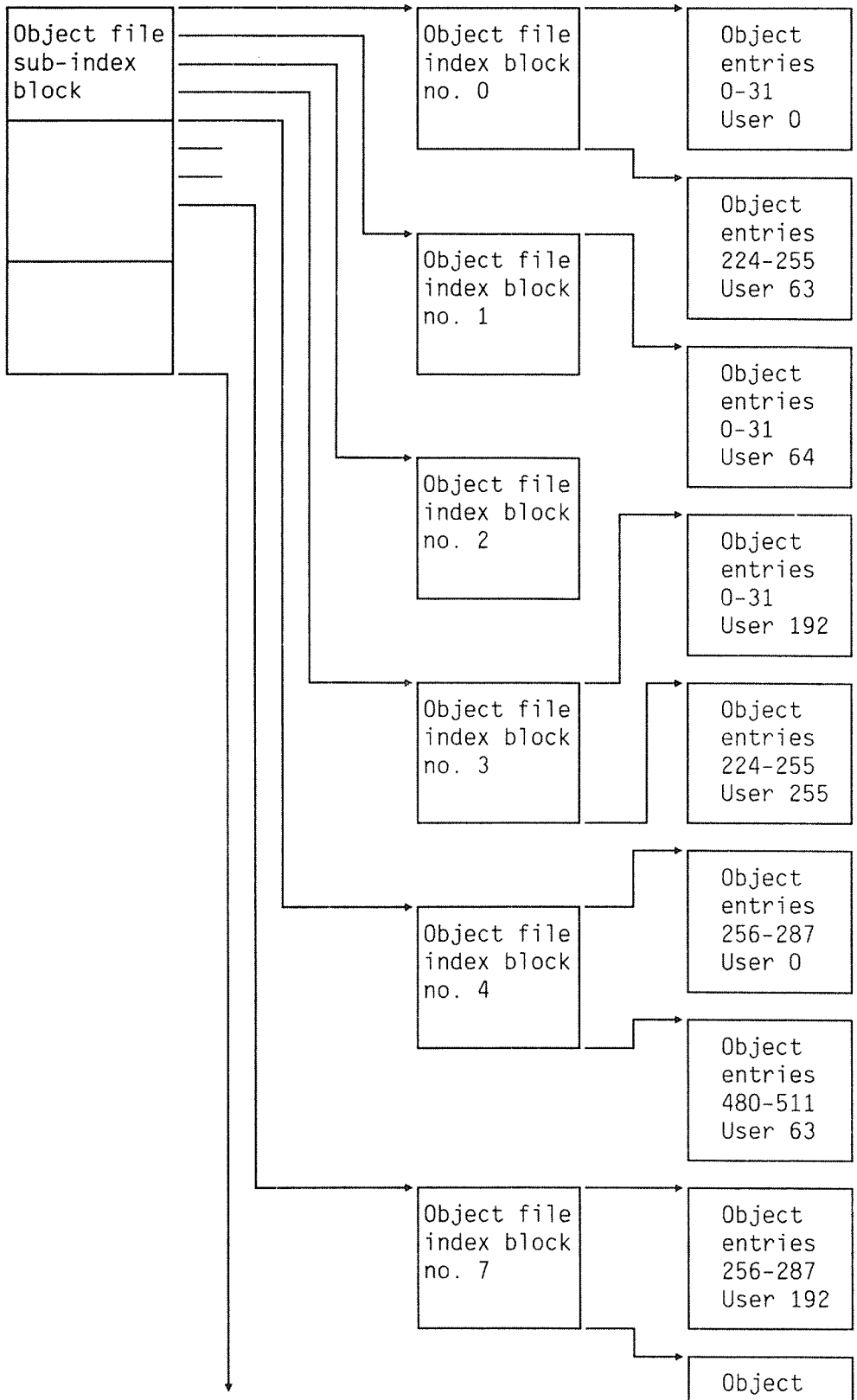
To avoid searching through all possible 4096 entries to get a specific object, another 4-bit word (ACOBL) in the user entry holds the actual number of extra object blocks in use. If ACOBL is zero, it is only necessary to search through the first 256 objects, if it equals 1, the user has less than 513 files and so on.

6.3 RESTRICTIONS - COMPATIBILITY

All versions of a file must have object entries in the same object block. The reason for this is to keep the directory structure compatible with earlier versions.

Refer to "Warning if moving to version J", on page 78, for things to consider if you move a directory containing files on object index > 255 from version K to version J.

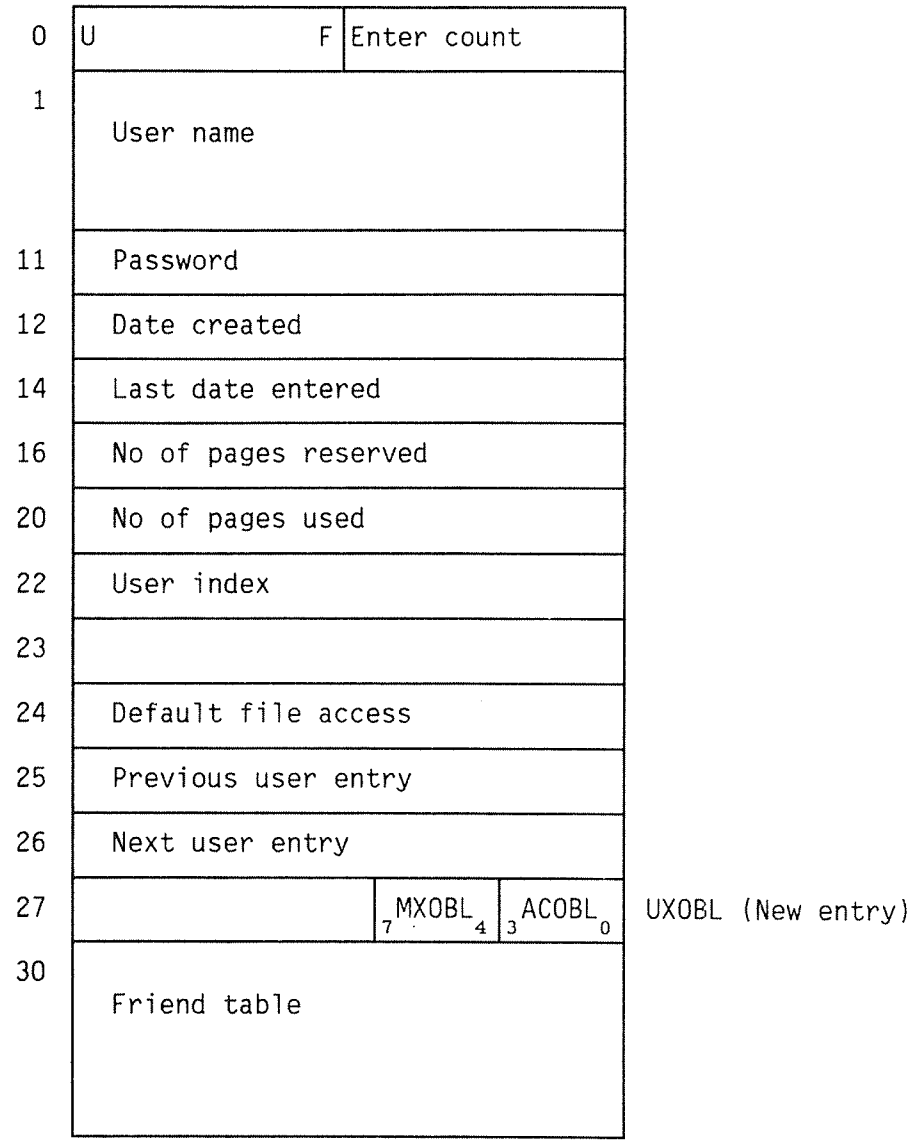
6.4 OBJECT FILE WITH SUBINDEX BLOCK



The last is object file index block no. 63, pointing to objects 4064-4095 for users 192-255

6.5 USER FILE ENTRY

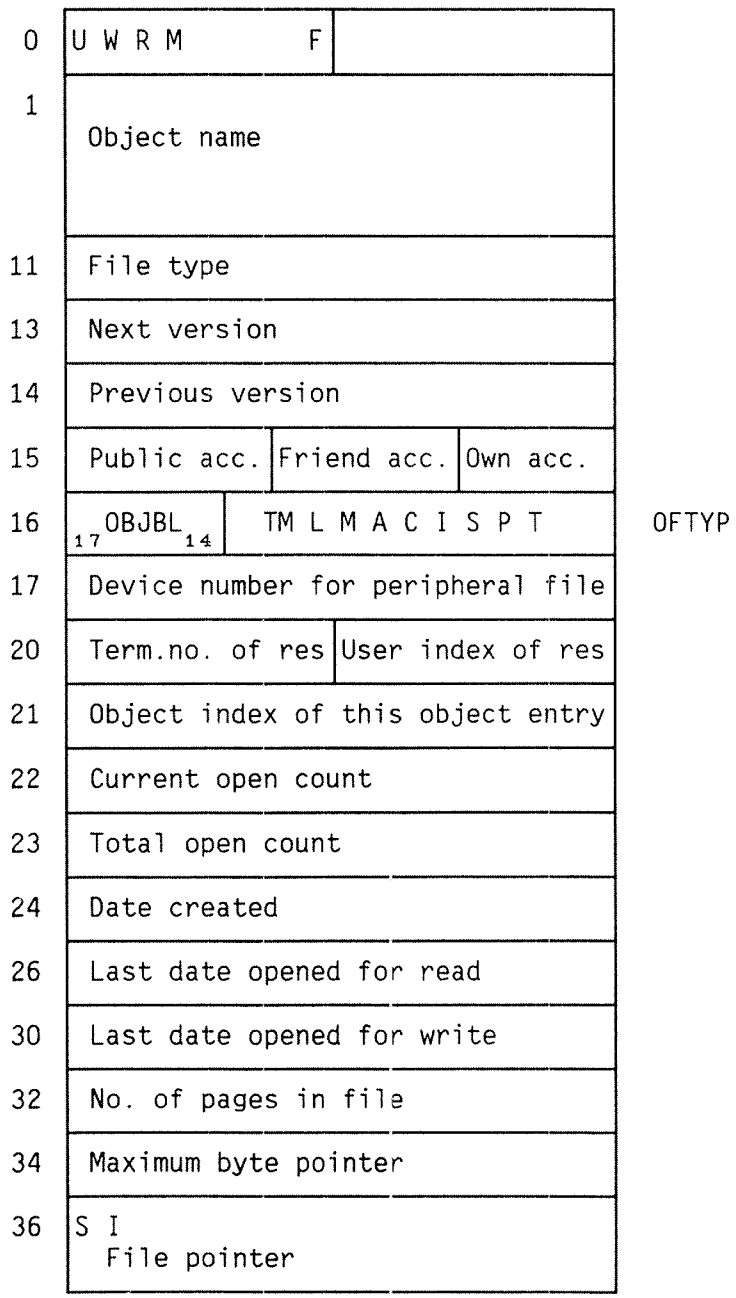
This figure shows the layout of the user entry. The free location in displacement 27_8 is used for MXOBL and ACOBL.



MXOBL is the maximum number of extra object blocks. ACOBL is the actual number of extra object blocks in use. If MXOBL (and ACOBL) is zero, object file extension is not allowed, and the file system will then work exactly as in previous versions of SINTRAN.

6.6 OBJECT FILE ENTRY

The object file entry must comprise the object block numbers of the current, next and previous versions of the file. Some vacant bits in OFTYP are used for the object block number of the file. The same object block number is used for both the current, next and previous versions.



Object file entry

6.7 OBJECT FILE BUFFER

The object file buffer header is increased by one word, holding the block number of the object in the buffer.

0	Lock number of object buffer	
1	Directory index	
2	Current object block no	(New entry)
3	Current object index (8 bit)	
4	First index in index buffer	
6	Index buffer	

6.8 THE OPEN FILE TABLE ENTRY

The 4 most significant bits in OFFTP (displacement 6) were free, and are now used to hold the object block number of the file.

6.9 WARNING IF MOVING TO VERSION J

If you move a directory (for example a removable disk, or a diskette) used from SINTRAN III version K to an installation still running version J, you must note the following if this directory contains files with object index > 255:

- All files with object index > 255 will be invisible (but they will reappear when you move back to version K).
- If you use the commands @REGENERATE-DIRECTORY or @TEST-DIRECTORY in the J-version on such a directory, the directory will end in an indeterminable state and the files on object index > 255 will be lost.
- The FILE-SYSTEM-INVESTIGATOR (for version J) used on such a directory will report errors when detecting files on object index > 255.

7. SPOOLING

It is now possible to do remote spooling without using the COSMOS remote spooling facility.

This is made possible because the commands @APPEND-SPOOLING-FILE, @DELETE-SPOOLING-FILE, @MOVE-SPOOLING-QUEUE-ENTRY and @REMOVE-FROM-SPOOLING-QUEUE now accept remote file specifications in both the ‹peripheral file› and ‹file name› parameters.

Note one restriction which is applicable if you have a network with some systems running SINTRAN III version J:

Remote spooling is not handled (except COSMOS remote spooling) in version J, thus you may not do remote spooling to a spooling device on a system running version J.

Also note of the following:

- if a spooling process is unable to access a remote file to be output due to network failure, the spooling request is moved to the back of that spooling queue. This process is repeated until the file can be accessed.

- if a network connection is broken during printing, thus causing the rest of a file to be inaccessible, the rest of that printout is lost.

- if you use the commands @DELETE-SPOOLING-FILE, @MOVE-SPOOLING-QUEUE-ENTRY or @REMOVE-FROM-SPOOLING-QUEUE to delete or move a spooling request, and this request contains a remote file, you may use unique abbreviations if it is possible to access the file, thus verifying the abbreviated names. If there is no connection to the remote system, however, you must specify a file to give an exact match with the file name given when the request was first entered.

Further descriptions of the changes in the spooling system, are given on pages 7-11 ("Modified commands") and page 15 ("Modified Monitor Calls").

Note that when using the @APPEND-SPOOLING-FILE command on a remote computer, the file is not copied to the remote computer. When the file is to be printed, the contents of the file is transferred page by page by COSMOS remote file access.

Also note that the @LIST-SPOOLING-QUEUE command does not support remote files.

Furthermore, note that if you try to define spooling files for COSMOS Spooling with the same name as spooling files for a local spooling program, the local spooling files will be lost.

8. TIME SLICING

Any RT-program may now be time sliced. Two new commands are introduced in the SINTRAN-Service-Program to make this possible:

```
*INSERT-PROGRAM-IN-TIME-SLICE
*REMOVE-PROGRAM-FROM-TIME-SLICE
```

An RT-program will enter the time slice at the highest priority in the selected time slice class when started (MON RT or @RT).

Note that background programs still must be removed/inserted by the commands:

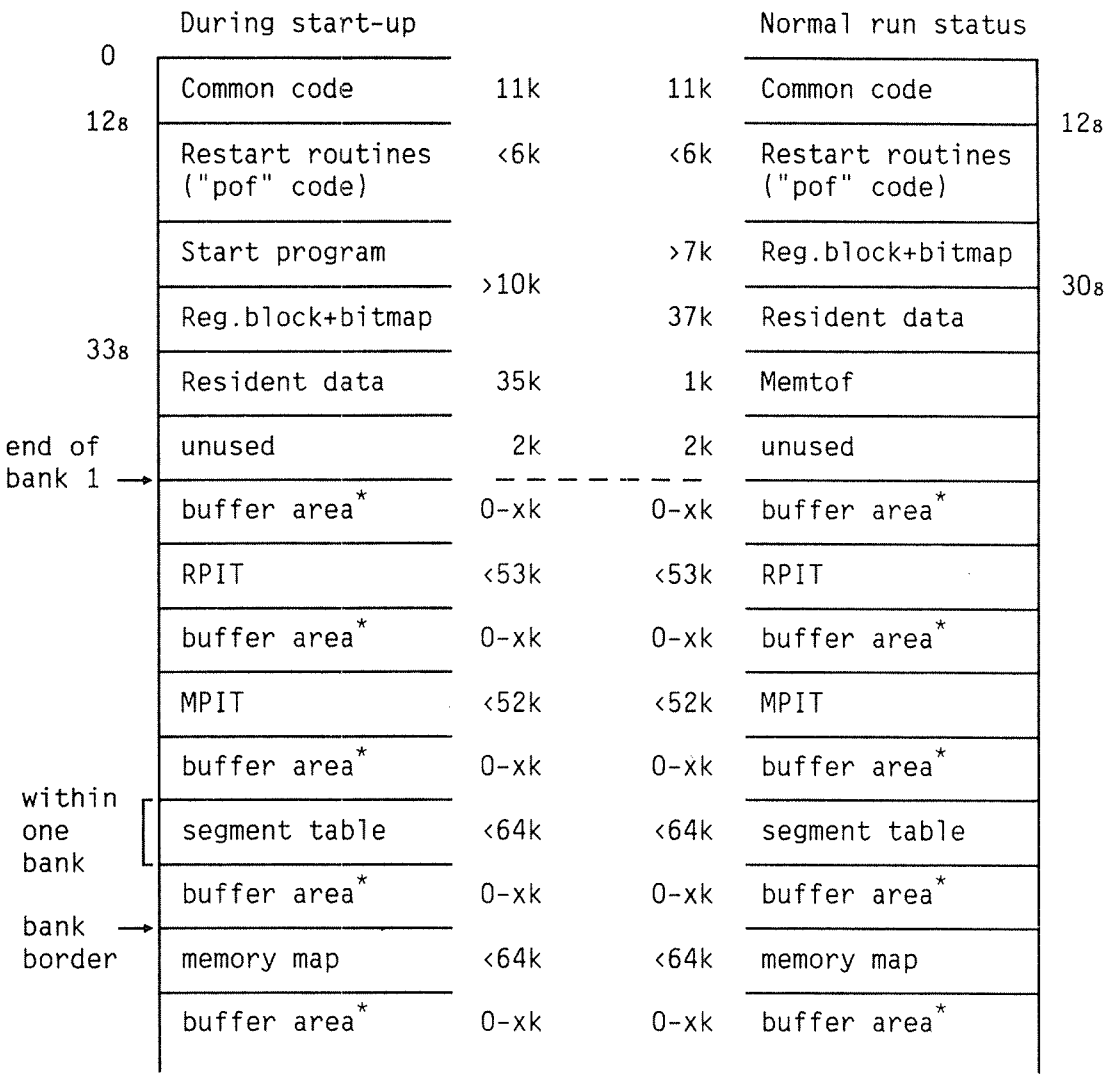
```
*INSERT-IN-TIME-SLICE
*REMOVE-FROM-TIME-SLICE
```

On the VSX-version, 16 time slice classes are available. By default, the 6 classes 0-5 are defined, the rest are free to use.
Also by default, the time slice elements $0\text{-}27_8$ are used to define the standard time slice parameters; elements $30\text{-}77_8$ are free to use.

Otherwise, there are no changes to the time slicing compared to the J-version.

9. SINTRAN III K-VERSION, SYSTEM LAYOUT (VSX)

9.1 PHYSICAL MEMORY LAYOUT



*) buffer areas are used for big terminal data fields and other non-PIT data.

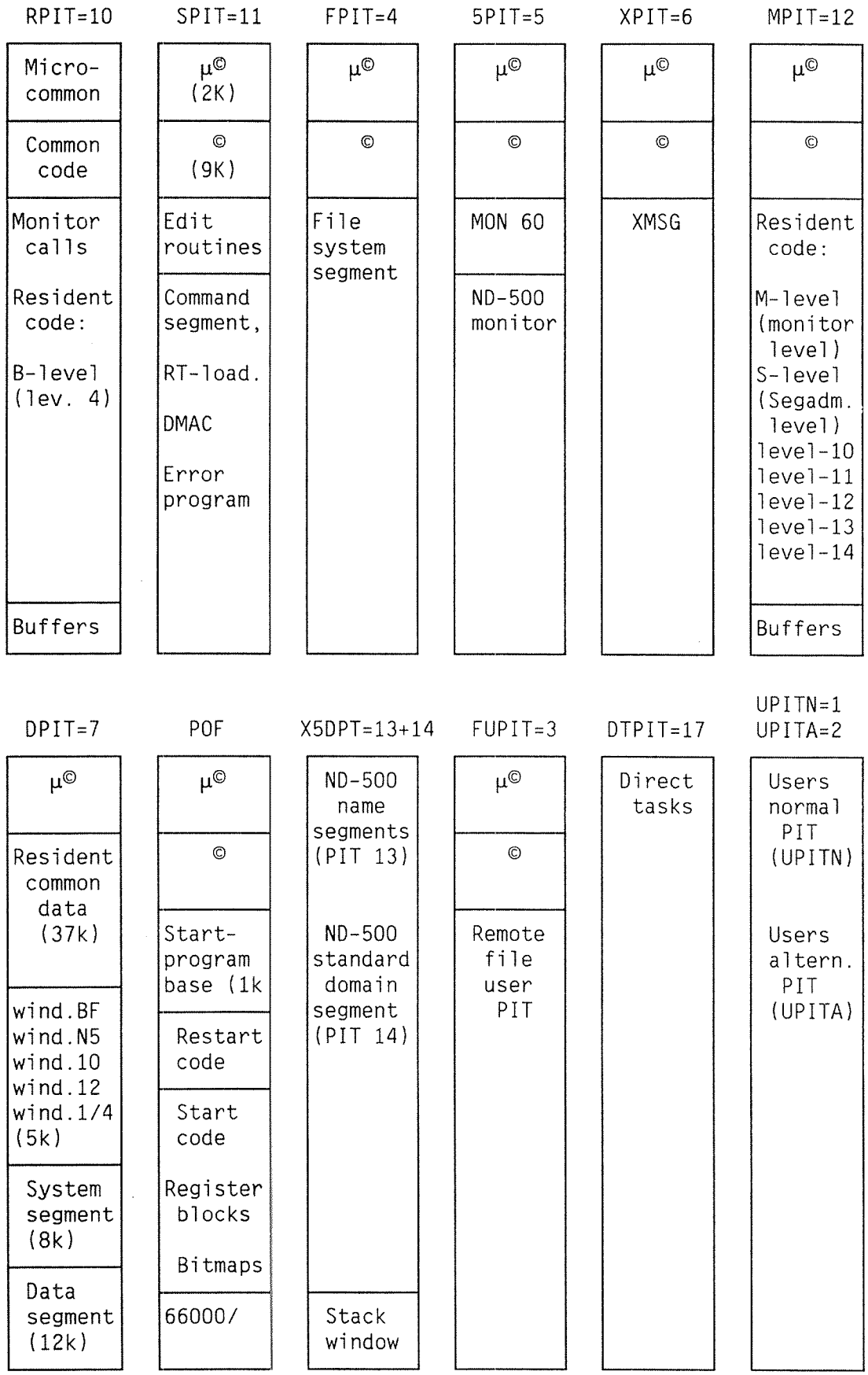
Note that common code always starts at physical address 0 and that resident data (DPIT) logical address 4000 starts at physical address 60000. All resident code is mapped as segments and is accessible through the segment table.

Logical device table is found in bank no. LOGDBANK at addresses found in the CNVRT array in DPIT.

9.2 SYSTEM LAYOUT ON DISK

File	Contents	Start address	Size	Disk addr.	Macro displ.	Patch macro
SINT RAN: DATA	Common Code			1		
	Start Restart					
	Resident Data			)GJEM area)HENT		
MACM-AREA: DATA	Error Program	30 000	12k	100	- 13	PERRP
	End Resid. data	112 000	2k	112		P2RDA
	System segment	130 000	3k	114	- 54	PSYSG
	Memtof	172 000	1k	117		
	RT-Loader	30 000	41k	137	- 14	PRTLO
SEG FILO: DATA	Common Code	0	13k	200	0	PCCST
	Start Restart	26 000	20k	213		
	Resident Data	4 000	43k	233	- 2	PRDAT
	End resid. Data	112 000	2k	277		
	System Segment	130 000	3k	301		
	Spooling Dataf.	150 000	1k	304	- 64	PSPDF
	RPIT	26 000	65k	305	- 13	PRPIT
	MPIT	26 000	65k	372	- 13	PMPIT
	Segment Table	0	20k	457	0	PSGTB
	File System	26 000	65k	477	- 13	PFILS
	Command-Segment	26 000	65k	564	- 13	POPCO
	5PIT	26 000	5k	651	- 13	P5PIT
	ND-500 monitor	40 000	60k	656	- 20	
				736		

9.3 PAGE INDEX TABLE LAYOUT



Note that (almost) all code must run in two-bank mode. Some code must even switch between one-bank and two-bank mode in order to access all its data (or use physical memory load and store intructions).
All system code will use DPIT as alternative page table.

Common code (©).

The common code part contains the routines that may be called from more than one PIT.

The common code should not exceed 11 K of code (0-25777$_8$).

μ© (micro common).

This part of common is also present in the data PIT (DPIT). It is mainly used for parameter fetching and other operations on the user's data area.

Resident code (RPIT).

This part contains code for most SINTRAN monitor calls except a few, which are placed on SPIT. File-system monitor calls are processed in the file system PIT. Other resident code that today is found in part 2 of resident should also be in this PIT, e.g., TAD resident code, resident RT-programs, configuration dependent code and "PIT3" code.

OUTBT/INBT level code is here.
Buffers accessed with RBGET/RBPUT are at the top of this PIT (they are also in MPIT).

Monitor PIT (MPIT).

Here is all code for:
monitor level
internal interrupts (level 14)
drivers for levels 10 to 13
and segadm level

Note that the part of this PIT that contains segadm is on ring 3. This makes it possible to run nearly always with paging on.

Buffers accessed with RBGET/RBPUT are at the top of this PIT (they are also in RPIT).

SINTRAN PIT (SPIT).

In this page index table we find the command, RT-Loader and DMAC segments. A segment will be removed from this PIT only when another segment must be entered. Note that the first page of the segment area (page 13$_8$) always contains the Edit routine with its related routines.

File system and file user, ND500, XMSG PITs (FPIT, FUPIT, 5PIT, XPIT).

These PITs each (currently) contain a single segment only, and a special strategy is applied to the setting and clearing of these page index tables to minimize context switch overhead.

ND-500 name- and standard domain segment PITs (X5DPT).

These PITs are used for the ND-500 name segment and standard domain segments. The last page of these PITs are used as a window to the ND-500 monitor stack page on the ND-500 user's data segment.

Data PIT (DPIT).

The data PIT contains the resident common data, as RT-descriptions, data fields and system global variables. The background system segments are placed in this PIT, as well as the ND-500 data segments and various file system segments. All windows are in this PIT. μ© is also included here.

User page index tables (UPITN, UPITA, DTPIT).

Three page index tables are reserved for the users. Two for background and RT-programs (normal and alternative PIT) and one for direct tasks.

Non-PIT data.

The following data is not in any PIT:
Segment table
memory map
RT-programs' register block and bit map
"big" terminal (TAD) data fields
ND-500 mail boxes
logical device number tables
ND-500 communication buffers (for MON 60)

9.4 DATA STRUCTURES

Segment data structures and operations:

The memory map always starts on a bank boundary. Physical page multiplied by four is the displacement into memory map, i.e., a memory map pointer can always be divided by four. If the two lower bits of the PREVIOUS pointer are nonzero, this entry is the first element in a page list and the upper 14 bits of the pointer contains the segment number this page list belongs to. The end of the page list is marked by a zero in the PAGLINK pointer.

Memory map element:

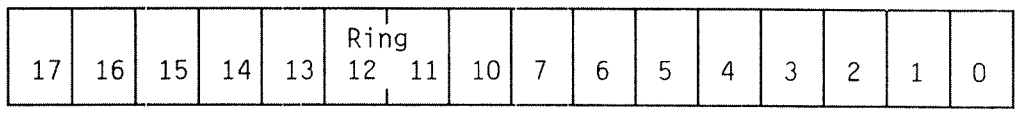
0	PAGLINK	Next page in page link.
1	PREVIOUS	Previous page in link.
2	PROTECT	Protect and status word (see below)
3	LOGPAGE	Logical page number.

Segment table entry:

0	SEGLINK	Link through active segments
1	PRESEG	Previous segment in link
2	LOGADR	First logical page of the segment
3	SEGLENGTH	Length of the segment in pages
4	MADR	Address of segment within the segment file
5	FLAG	Flag word (see below)
6	SGSTATUS	Segment status and protect word (see below)
7	BPAGLINK	Pointer to the page list of this segment

SGSTATUS and PROTECT bits:

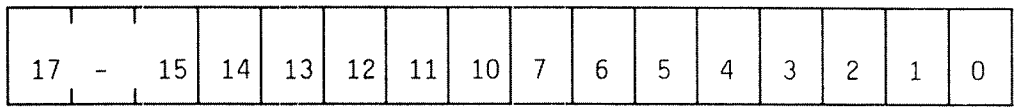
5NCLSEG: Never clear PIT for this kind of segment.
Pit entries are set up and cleared whenever
a page is given to/taken from the segment.
5FIX: Segment is fixed
5CMINH: This memory map element must not be used
5SPTFIXED: Segment is fixed in page table
5SMSYS: This memory map element is used
for system area.
5MRES: Memory map element is reserved
(with FIXC)
5CMIDUM: This memory map element is
in the free list.



Protection ring
5PGU: Page used.
5WIP: Written in page.
5FPM: Fetch permit
5RPM: Read permit
5WPM: Write permit

FLAG bits:

5OK: Segment is ready
5INHB: Segment not built
5SYSEGM: System segment
5SPROT: Protected segment
5REEP: Reentrant subsystem segment
5FIXC: Segment is fixed contiguously
5DEMAND: Demand segment



Segment file number for this segment

The following values are held in the CPU (on ND-110/CX only) and loaded at system start:

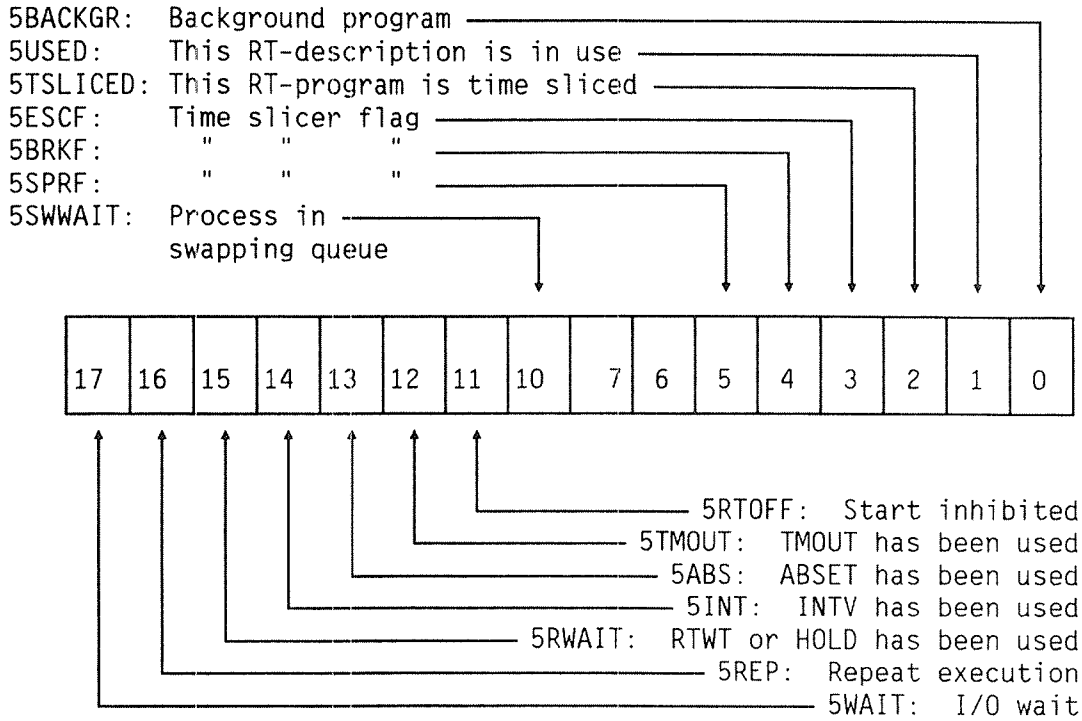
```
INTEGER CORMBANK  % Bank number of memory map
INTEGER SEGTBANK  % Bank number of segment table
INTEGER SEGSTART  % Displacement of segment table within bank
```

The RT-description looks like this:

0	TLINK	Time queue
1	STATUS	Status bits (see below)
2	INPRITY	Initial program priority
3	PRITY	Program priority
4	DTIM1	Start time
5	DTIM2	
6	DTIN1	Interval
7	DTIN2	
10	STADR	Start address
11	SEGM1	Initial segments
12	SEGM2	
13	WLINK	Waiting queue,execution queue
14	ACT1SEG	Actual segments
15	ACT2SEG	
16	INIPRI	Initial page tables and ring
17	ACTPRI	Actual page tables and ring (only)
20	BRESLINK	Beginning of reservation link
21	RSEGM	Reentrant segment
22	BUFWINDOW	Buffer window
23	TRMWINDOW	Terminal window, RT working field window
24	N5WINDOW	ND-500 mailbox window
25	RTDLGADDR	Logical address of register block

The register block and bit map are unchanged from the J-version.

Format of STATUS:



Format of INIPRI and ACTPRI:

0	Normal PIT				Alternative PIT				Level always = 1					1	Ring	
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	

9.5 INTERRUPT LEVEL USAGE (VSX)

15	Extremely fast user interrupts
14	Internal interrupts
13	Real Time Clock, HDLC drivers
12	Terminal Input & ND-100 - ND-500 Communication
11	Mass storage Input/Output
10	Terminal output
9	Direct tasks
8	
7	
6	
5	XMSG
4	I/O Monitor calls
3	Segment administration
2	SINTRAN III Monitor
1	Real time programs and Background programs
0	Idle loop

Note the changed use of levels 2 and 3.

9.6 SYSTEM INCLUDED SEGMENTS (VSX)

SEGMENT NO.	NAME	ADDRESS RANGE	PIT	DESCRIPTION
2	S3IMAGE	0 - 65777	1	Memory image of COMMON code, Start/restart program
3	S3COM	30000 - 177777	11	Command segment
4	S3RTL	30000 - 123777	11	RT-Loader segment
5	S3ERRS	130000 - 131777	7	System segment for error program
6	S3FS	26000 - 201777	4	File system segment
7	S3DMAC	64000 - 153777	11	DMAC segment
10	S3RTFIL	0 - 177777	2	RTFIL segment
11	S3ERRL	0 - 17777	1	Error log segment
12	S3FSSV	26000 - 177777	1	Initial file system segment
13	S3OPCSV	26000 - 177777	1	Initial command segment
14	S3ERRP	30000 - 53777	11	Error program segment
15	S3BFLY	26000 - 26000		Reserved for system extension
16	S3SRPIT	26000 - 177777	1	Initial RPIT (save area)
17	S3SMPIT	26000 - 177777	1	Initial MPIT (save area)
20	S3SDT5	0 - 175777	14	ND-500 standard domains segment
21	S3NMS5	0 - 175777	13	ND-500 name tables segment
22	S3RFAC	26000 - 155777	3	Remote File Access segment
23	S3DPIT	4000 - 111777	7	DPIT segment (global data)
24	S3SSGT	0 - 37777	1	Initial segment table
25	S3IRPIT	26000 - 177777	1	Image of RPIT
26	S3IMPIT	26000 - 177777	1	Image of MPIT
27	S3ISGT	0 - 37777	1	Image of segment table
30	S3SM5	40000 - 177777	5	ND-500 system monitor segment
31	S3SSPD	150000 - 151777	7	Initial spooling data fields
32	S3RTACC			Reserved, but not used
33	S3XMSGP	120000 - 177777	2	XMSG POF segment
34	S3XMSGD	0 - 117777	2	XMSG demand segment (XROUT)
35	S3MPIT	26000 - 121777	12	MPIT segment
36	S3TAD	110000 - 133777	11	TADADM segment
37	S3RTD	0 - 177777	1	RT-Loader data segment
40	S3FUDRT	150000 - 157777	7	File User data segment for RT-prog
41	S3IMED	26000 - 27777	1	Image of EDIT routines
42	S3ED	26000 - 27777	11	EDIT routines
43	S3PATCH	174000 - 177777	2	Reserved for internal use by ND
44	S3IDPIT	4000 - 111777	1	Memory image of system data (DPIT)
45	S3ISYS	130000 - 135777	1	Memory image of system segment
46	S3S5PIT	26000 - 37777	1	Save of 5PIT segment
47	S3RPIT	26000 - 121777	10	RPIT segment
50	S3I5PIT	26000 - 37777	1	Image of 5PIT segment
51	S35PIT	26000 - 37777	5	5PIT segment
52	S3SAVE	0 - 65777	1	Save of common code and start prog
53	S3SDPIT	4000 - 111777	1	Save of DPIT
54	S3SSYS	130000 - 135777	1	Save of system segment
55	S3SERRP	30000 - 53777	1	Save of error program
56	S3SRTC	30000 - 67777	1	Save of RT-Loader's code segment
57	S3SRTD	0 - 25777	1	Save of RT-Loader's data segment
60	S3SERD	112000 - 115777	1	Save of DPIT last two pages
61	S3IERD	112000 - 115777	1	Image of DPIT last two pages
62	S3SSM5	40000 - 177777	1	Save of ND-500 Monitor
63	S3MEMT	172000 - 173777	1	Memtof
64	S3ERD	112000 - 115777	7	DPIT last two pages

Note: Segments 2-64 will be given standard segment names the first time the RT-Loader is entered.

All system included segments are placed on segment file number 0 (SEGFILO:DATA), except segments 52-60 and 63 which are placed on the files SINTRAN:DATA and MACM-AREA:DATA.

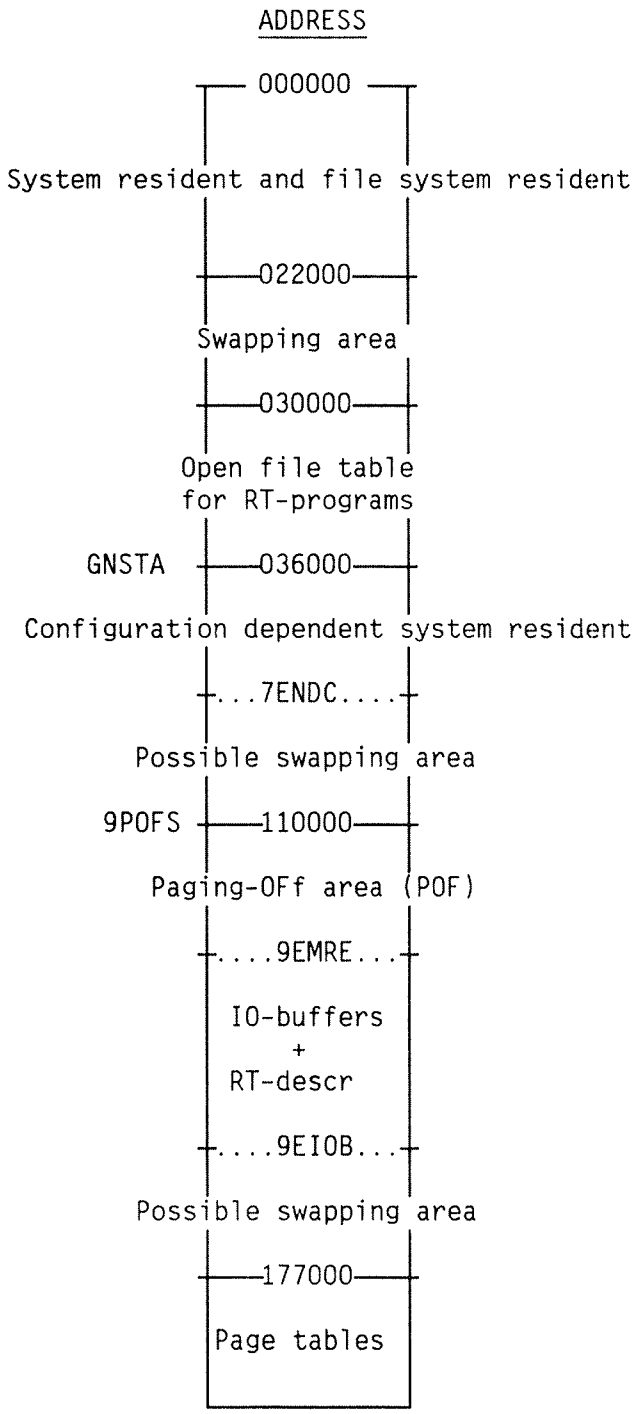
9.7 SYSTEM INCLUDED RT-PROGRAMS (VSX and VSE)

PROGRAM	PURPOSE
1SWAP	Queueing program requests for swapping
5SWAP	Performs ABSTR in ND-100 for the 500-swapper
ACCRT	RT accounting
BAKnn	Background process for terminal (BAK01-BAK99)
BKnnn	- " - (BK100-BK128)
BCHnn	Batch process
BPTMP	Timeout program for background allocation system
COSPO	COSMOS-spooling server
DUMM2	Dummy program used by the spooling system
DUMMY	Dummy program to prevent empty execution queue
FDRT1	Transfer data between interface buffer and memory. Floppy formatting. (FLOPPY-1)
FDRT2	Transfer data between interface buffer and memory. Floppy formatting. (FLOPPY-2)
FIXRT	Monitor call/command FIXC execution
RTDIL	Buffer transfer program for DISC-ACCESS-LOG
RTERR	Output error messages
RTRFA	Does remote file access for RT-programs (COSMOS - remote file access)
RTSLI	Time slicer. Changes priority on all time sliced processes.
RWRT1	Block data transfer. Activated from RFILE/WFILE/RPAGE/WPAGE for RT-programs
RWRT2	Open file from RT-programs
RWRT3	Block transfer on MAGTAPE-1 (MAGTP)
RWRT5	VERSATEC-1 DMA
RWRT6	CDC-DMA LINK
RWRT7	MAGTAPE-2
RWRT8	VERSATEC-2 DMA
RWRT9	FLOPPY-DISC 1
RWRT10	FLOPPY-DISC 2
RWRT11	LINE-PRINTER/VERSATEC -1 I/O
RWRT12	LINE-PRINTER/VERSATEC -2 I/O
RWRT13	Block oriented internal device 1 INPUT
RWRT20	Block oriented internal device 1 OUTPUT
RWRT14	Block oriented internal device 2 INPUT
RWRT21	Block oriented internal device 2 OUTPUT
RWRT15	Block oriented internal device 3 INPUT
RWRT22	Block oriented internal device 3 OUTPUT
RWRT16	Block oriented internal device 4 INPUT
RWRT23	Block oriented internal device 4 OUTPUT
RWRT17	Block oriented internal device 5 INPUT
RWRT24	Block oriented internal device 5 OUTPUT

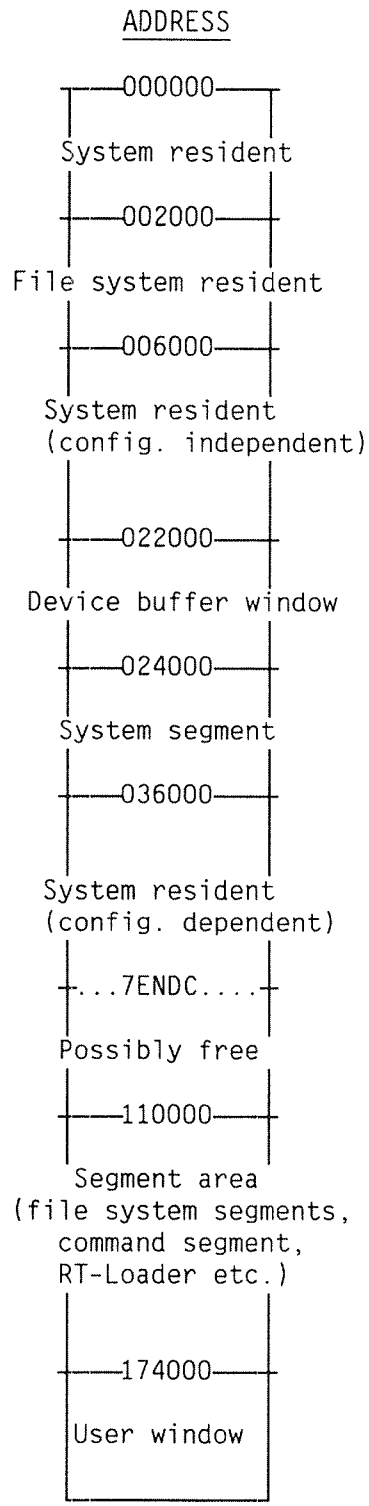
RWRT25	HASP DMA 1 INPUT
RWRT26	HASP DMA 1 OUTPUT
RWRT27	HASP DMA 2 INPUT
RWRT28	HASP DMA 2 OUTPUT
RWRT29	HASP DMA 3 INPUT
RWRT30	HASP DMA 3 OUTPUT
RWRT31	HASP DMA 4 INPUT
RWRT32	HASP DMA 4 OUTPUT
RWRT33	HASP DMA 5 INPUT
RWRT34	HASP DMA 5 OUTPUT
RWRT35	HASP DMA 6 INPUT
RWRT36	HASP DMA 6 OUTPUT
SPRTn	Spooling programs (1-9)
SPRnn	Spooling programs (10-30)
STSIN	Initialize SINTRAN III and start systems RT-programs
TADnn	Background process for Terminal Access Device
TADAD	Administrates connections to TADs from requesting users.
TERMP	Starts the user defined "clean-up" RT-program when RT-programs are aborted (if enabled)
TIMRT	Timer RT-program. Start timeout-routine for all devices in timer-table.
UDRnn	Performs Fast Universal DMA for user processes.

10. SINTRAN III K-VERSION, SYSTEM LAYOUT (VSE)

10.1 PHYSICAL MEMORY

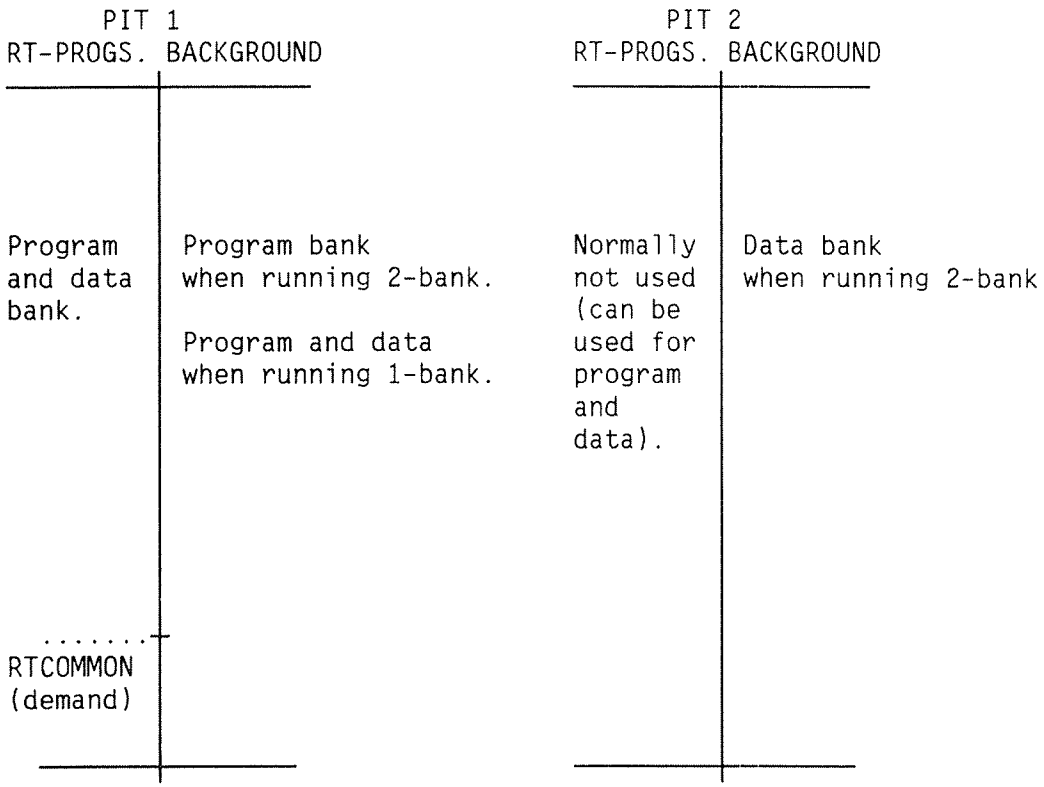


10.2 PAGE INDEX TABLE 0

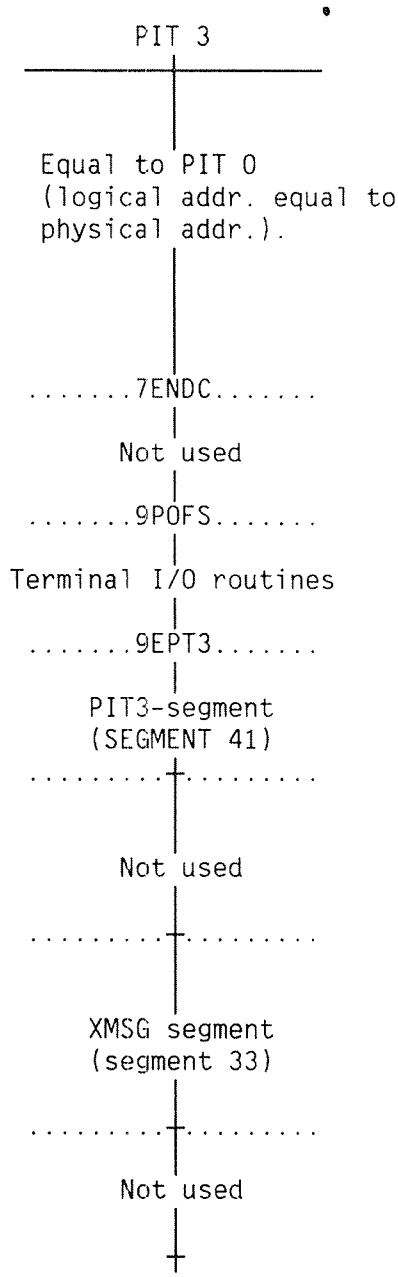


All resident pages are mapped physical page equal to logical page.

10.3 PAGE INDEX TABLES 1 AND 2



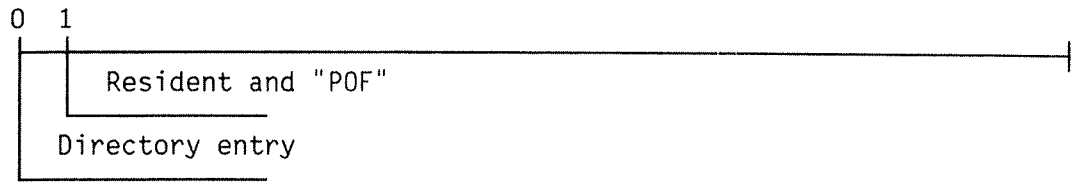
10.4 PAGE INDEX TABLE 3



10.5 SYSTEM LAYOUT ON DISK

10.5.1 SINTRAN:DATA

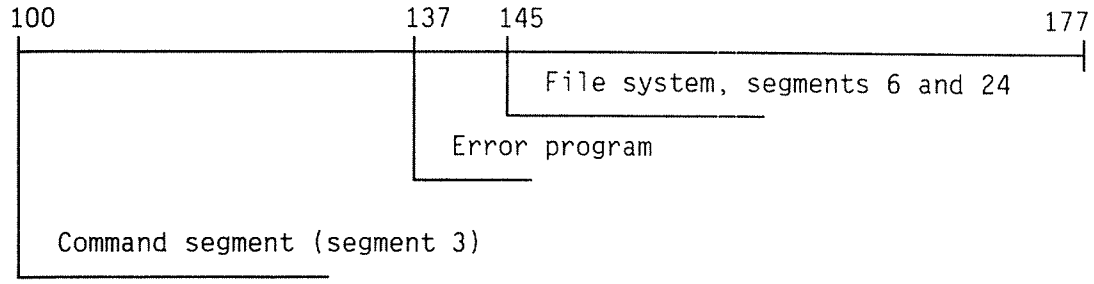
Disk address in pages



10.5.2 MACM-AREA:DATA

10.5.2.1 LAYOUT

Disk address in pages (octal), relative to the start of the directory



10.5.2.2 DISPLACEMENTS WHEN PATCHING

```
Command segment:          -110000
File system seg. 6 + 24:     2000
```

10.5.3 SEGFILO:DATA

Disk address in pages (octal), relative to the start of the directory.

Other segment files may reside in any directory at any disk address. The maximum size of a segment file is 16383 pages. Due to limitations in the RT-Loader, the sum of the used segment files cannot be greater than 32768 pages.

10.5.4 INTERRUPT LEVEL USAGE (VSE)

15	Extremely fast user interrupts
14	Internal interrupts
13	Real Time Clock, HDLC drivers
12	Terminal Input
11	Mass storage Input/Output
10	Terminal output
9	Direct tasks
8	
7	
6	
5	XMSG
4	I/O Monitor calls
3	SINTRAN III Monitor
2	Direct tasks
1	Real time programs and Background programs
0	Idle loop

This is the same as in the J-version.

10.5.5 SYSTEM INCLUDED SEGMENTS (VSE)

Note: Segments 2-43 will be given standard segment names the first time the RT-Loader is entered.

SEGMENT NO.	NAME	ADDRESS RANGE	PIT	DESCRIPTION
2	S3IMAGE	0 - 175777	1	Memory, image and POF
3	S3COM	110000 - 173777	0	Command segment
4	S3RTL	110000 - 147777	0	RT-Loader
5	S3ERRS	26000 - 31777	0	Error program "system segment"
6	S3FSCOM	110000 - 137777	0	File system common segment
7	S3DMAC	110000 - 153777	0	DMAC segment
10	S3RTFIL	0 - 177777	2	RTFIL segment
11	S3ERRL	0 - 17777	1	Error log segment
12	S3FS2SV	140000 - 173777	0	Initial reentrant file system seg.2
13	S3RTLSV	56000 - 147777	1	Initial RT-Loader segment
14	S3ERRP	110000 - 123777	0	Error program segment
15	S3SMSV	110000 - 173777	0	Initial service program and mail
16	S3IOMT	150000 - 167777	0	Initial IOMTY segment
17	S3PT3SV	0 - 27777	1	Initial PIT3 segment
20	S3SDT5			Reserved, but not used
21	S3NMS5			Reserved, but not used
22	S3RFUS1	110000 - 163777	0	Reentrant file user segment no. 1
23	S3SMSEG	110000 - 173777	0	Service program and mail
24	S3FSRS1	140000 - 173777	0	File system reentrant segment no. 1
25	S3FSRS2	140000 - 173777	0	File system reentrant segment no. 2
26	S3RFUS2	110000 - 163777	0	Reentrant file user segment no. 2
27	S3IOMTY	150000 - 167777	0	Segment for IOMTY mon. call
30	S3SM51			Reserved, but not used
31	S3SM52			Reserved, but not used
32	S3RTACC	110000 - 127777	0	RT accounting segment
33	S3XMSGP	140000 - 167777	2	XMSG POF segment
34	S3XMSGD	0 - 137777	2	XMSG demand segment
35	S3XMSGR	0 - 37777	2	Reserved for XMSG
36	S3TAD	110000 - 133777	0	TADADM segment
37	S3RTD	0 - 177777	1	RT-Loader segment
40	S3FUDRT	164000 - 173777	0	File user data segment for RT-prog.
41	S3PT3	116000 - 137777	1	Pit 3 segment
42	S3SPLSV	110000 - 137777	0	Initial spooling program segment
43	S3SPL	110000 - 137777	0	Spooling program segment

11. TERMINAL INPUT/OUTPUT

11.1 CHANGED DATA FIELDS - TERMINALS

11.1.1 TERMINAL INPUT AND OUTPUT DATA FIELD - VSX

DATA FIELD LAYOUT IN SINTRAN MEMORY AREA, DPIT PART

This layout applies both to the input and output data fields.

-4	TDFPHPAGE	Physical page of data field
-3	TDFLGADDR	Address within a page of data fields
-2	STDRIV	Driver start address
-1	DRIVER	Driver interrupt restart address
0	RESLINK	Reservation link
1	RTRES	Reserving RT-program
2	BWLINK	Beginning of waiting queue
3	TYPRING	Device type bits and ring
4	ISTATE	0 = active, 1 = I/O-wait, 2 = buffer wait, -1 & -2 = nowait
5	MLINK	Monitor queue link
6	MFUNC	Monitor level function address

INPUT DATA FIELD LAYOUT IN SINTRAN MEMORY AREA, PART OUTSIDE DPIT:

-44	PECH7	Echo table
-34	PBRK7	Break table
-24	IN5MSG	Address of ND-500 message when doing quick instring
-23	RSISTE	Echo pointer
-22	BRECHOFL	Break & echo flag
-21	ROUSPEC	Address of special subroutine
-20	NCBRK	Number of characters after last break
-17	CTTYP	Terminal type
-16	CESCP	Disconnect and escape characters
-15	BRKMAX	Maximum BHOLD before break
-14	TSPEED	Terminal speed
-13	CNTREG	Control register

To be continued

-12	DFLAG	Device flag bits
-11	ECHOTAB	Pointer to echo table
-10	BRKTAB	Pointer to break table
-7	LAST	Last typed character
-6	TMSUB	Time out subroutine
-5	TMR	Time out counter
-4	TTMR	Start value of TMR
-3	HDEV	Hardware device number
-2	STDRIV	Driver start address
-1	DRIVER	Driver interrupt restart address
0	TDRADDR	Address of data field in resident
1	XDFOPP	Address of DFOPP in resident
2	XOPPDF	Address of opposite data field (outside resident)
3	TYPRING	Device type bits and ring
4	XONCR	XON character, input control.
5	XOFCR	XOFF character, input control.
6		Not used
7	IOTRANS	Called from INBT/OUTBT to transfer
10	STDEV	Start device routine
11	SETDV	IOSET routine
12	DFOPP	Pointer to output channel data field
13	DERROR	Error code
14	BUFST	Start of ring buffer
15	MAX	Buffer capacity
16	BHOLD	Number of characters in buffer
17	HENTE	Fetch pointer
20	CFREE	Free positions
21	FYLLE	Store pointer
22	BSTATE	Background program state
23	TSTATE	Timeslice state
24	DBPROG	Background RT-program
25	DBADR	Saved P-reg on escape and file system monitor calls
26	RIFIL	For mode input file number
27	BCHISTS	For mode input status
30	DERO	Error information
30	BREGBLOCK	Register save at escape
32	DER2	Error information
40	DBPREG	P-register on page fault on IOBT level
41	DBACTPRI	ACTPRI on page fault on IOBT level
42	FLAGB	Background flags
43	EUSADD	Address for user-escape handling
44	LUSADD	Address for local-function handling

OUTPUT DATA FIELD LAYOUT IN SINTRAN MEMORY AREA, PART OUTSIDE DPIT

-10	SCREEN	Counter for stop on full page
-7	EMPTFLAG	Buffer empty flag
-6	TMSUB	Time out subroutine
-5	TMR	Time out counter
-4	TTMR	Start value of TMR
-3	HDEV	Hardware device number
-2	STDRIV	Driver start address
-1	DRIVER	Driver interrupt restart address
0	TDRADDR	Address of data field in resident
1	XDFOPP	Address of DFOPP in resident
2	XOPPDF	Address of opposite data field (outside resident)
3	TYPRING	Device type bits and ring
4	XONCR	Xon character, input control
5	XOFCR	Xoff character, input control
6	PDISPLAY	Pointer to next terminal in display table
7	IOTRANS	Called from INBT/OUTBT to transfer
10	STDEV	Start device
11	SETDV	IOSET routine
12	DFOPP	Pointer to output channel data field
13	DERROR	Error code
14	BUFST	Start of ring buffer
15	MAX	Buffer capacity
16	BHOLD	Number of characters in buffer
17	HENTE	Fetch pointer
20	CFREE	Free positions
21	FYLLE	Store pointer
22	MINBHOLD	Lower limit for break
23	ROFIL	For "mode" (output file number)
24	BCHOST	For "mode" (output status)
25	ON5MSG	Address for ND-500 message
26	CBUADR	Current user buffer address (outstring)
27	NOCHAR	Number of bytes in outstring monitor call
30	CNOCHAR	Number of words left to transfer in outstring
31	XNOCHAR	Working location for outstring
32	ZOPRG	P, X, T-registers in outstring
35	ZOARG	A, D and L-registers in outstring.
40	ZOSRG	S, B-registers + old page in outstring
43	SBHOLD	Saved BHOLD in outstring

INPUT DATA FIELD LAYOUT IN SINTRAN IMAGE AND SAVE AREA:

-4	ZDBPROG	Background RT-program
-3	HDEV	Hardware device number
-2	ZXONOFCR	XON and XOFF chars. (XOFF in most significant byte)
-1	ZDFLAG	Device flag bits
0	ZROUSPEC	Address of special subroutine
1	ZCTTYP	Terminal type
2	ZCESCP	Disconnect and escape characters.
3	TYPRING	Device type bits and ring
4	ZTSPEED	Terminal speed
5	ZCNTREG	Control register
6	MFUNC	Monitor level function address

OUTPUT DATA FIELD LAYOUT IN SINTRAN IMAGE AND SAVE AREA:

-4	ZTINFO	Various information bits for terminal
-3	HDEV	Hardware device number
-2	ZXONOFCR	XON and XOFF chars. (XOFF in most significant byte)
-1		Not used
0		Not used
1		Not used
2		Not used
3	TYPRING	Device type bits and ring
4	ZDFTYP	Data field type, 0 = terminal, 1 = telefix R, 2 = telefix B
5		Not used
6	MFUNC	Monitor level function address

11.1.2 TERMINAL INPUT AND OUTPUT DATA FIELD - VSE

INPUT DATA FIELD LAYOUT IN SINTRAN MEMORY, IMAGE AND SAVE AREA:

-47	IXONCR	XON character, input control.
-46	IXOFCR	XOFF character, input control.
-45	TINFO	Various information bits for terminal
-44	PECH7	Echo table
-34	PBRK7	Break table
-24	IN5MSG	Address of ND-500 message when doing quick instring
-23	RSISTE	Echo pointer
-22	BRECHOFL	Break & echo flag
-21	ROUSPEC	Address of special subroutine
-20	NCBRK	Number of characters after last break
-17	CTTYP	Terminal type
-16	CESCP	Disconnect and escape characters.
-15	BRKMAX	Maximum BHOLD before break
-14	TSPEED	Terminal speed
-13	CNTREG	Control register
-12	DFLAG	Device flag bits
-11	ECHOTAB	Pointer to echo table
-10	BRKTAB	Pointer to break table
-7	LAST	Last typed character
-6	TMSUB	Time out subroutine
-5	TMR	Time out counter
-4	TTMR	Start value of TMR
-3	HDEV	Hardware device number
-2	STDRIV	Driver start address
-1	DRIVER	Driver interrupt restart address
0	RESLINK	Reservation link
1	RTRES	Reserving RT-program
2	BWLINK	Beginning of waiting queue
3	TYPRING	Device type bits and ring
4	ISTATE	0=active, 1 = I/O-wait, 2 = buffer wait, -1 & -2 = nowait
5	MLINK	Monitor queue link
6	MFUNC	Monitor level function address
7	IOTRANS	Input IOTRANS routine

To be continued

10	STDEV	Start device routine
11	SETDV	IOSET routine
12	DFOPP	Pointer to output channel data field
13	DERROR	Error code
14	BUFST	Start of ring buffer
15	MAX	Buffer capacity
16	BHOLD	Number of characters in buffer
17	HENTE	Fetch pointer
20	CFREE	Free positions
21	FYLLE	Store pointer
22	BSTATE	Background program state
23	TSTATE	Time slice state
24	DBPROG	Background RT-program
25	DBADR	Saved P-reg on escape and file system monitor calls
26	RIFIL	For mode input file number
27	BCHISTS	For mode input status
30	DERO	Error information
30	BREGBLOCK	Register save at escape
32	DER2	Error information
40	DBPREG	P-register on page fault on IOBT level
41	DBACTPRI	Actpri on page fault on IOBT level
42	FLAGB	Background flags
43	EUSADD	Address for user-escape handling
44	LUSADD	Address for local-function handling

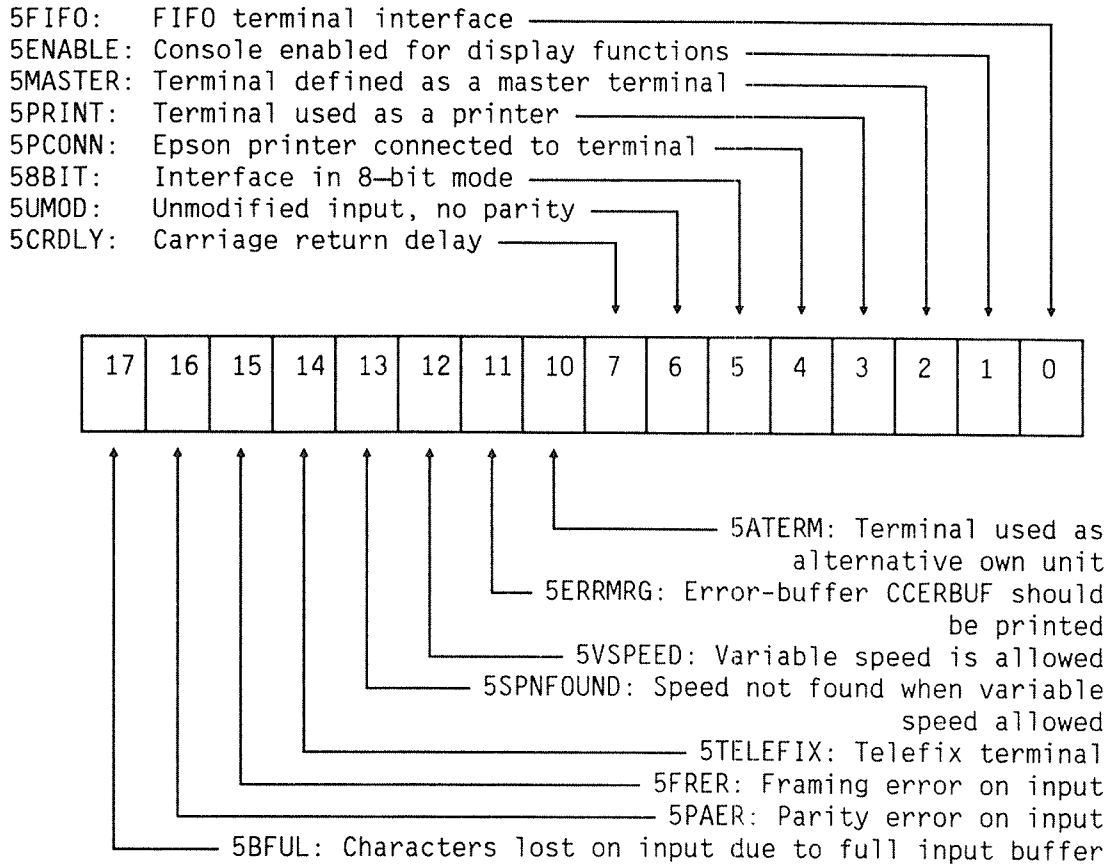
OUTPUT DATA FIELD LAYOUT IN SINTRAN MEMORY, IMAGE AND SAVE AREA:

-13	PDISPLAY	Pointer to next terminal in display table
-12	UXONCR	XON character, output control
-11	UXOFCR	XOFF character, output control
-10	SCREEN	Counter for stop on full page
-7	EMPTFLAG	Buffer empty flag
-6	TMSUB	Time out subroutine
-5	TMR	Time out counter
-4	TTMR	Start value of TMR
-3	HDEV	Hardware device number
-2	STDRIV	Driver start address
-1	DRIVER	Driver interrupt restart address

To be continued

0	RESLINK	Reservation link
1	RTRES	Reserving RT-program
2	BWLINK	Beginning of waiting queue
3	TYPRING	Device type bits and ring
4	ISTATE	0 = active, 1 = I/O-wait, 2 = buffer wait, -1 & -2 = nowait
5	MLINK	Monitor queue link
6	MFUNC	Monitor level function address
7	IOTRANS	Input iotrans routine
10	STDEV	Start device
11	SETDV	IOSET routine
12	DFOPP	Pointer to output channel data field
13	DERROR	Error code
14	BUFST	Start of ring buffer
15	MAX	Buffer capacity
16	BHOLD	Number of characters in buffer
17	HENTE	Fetch pointer
20	CFREE	Free positions
21	FYLLE	Store pointer
22	MINBHOLD	Lower limit for break
23	ROFIL	For "mode" (output file number)
24	BCHOST	For "mode" (output status)
25	ON5MSG	Address for ND-500 message
26	CBUADR	Current user buffer address (outstring)
27	NOCHAR	Number of bytes in outstring monitor call
30	CNOCHAR	Number of words left to transfer in outstring
31	XNOCHAR	Working location for outstring
32	ZOPRG	P, X, T-registers in outstring
35	ZOARG	A, D and L-registers in outstring.
40	ZOSRG	S, B registers + old page in outstring
43	SBHOLD	Saved BHOLD in outstring

TINFO bits:


5FIFO: FIFO terminal interface
5ENABLE: Console enabled for display functions
5MASTER: Terminal defined as a master terminal
5PRINT: Terminal used as a printer
5PCONN: Epson printer connected to terminal
58BIT: Interface in 8-bit mode
5UMOD: Unmodified input, no parity
5CRDLY: Carriage return delay
17 16 15 14 13 12 11 10 7 6 5 4 3 2 1 0
5ATERM: Terminal used as alternative own unit
5ERRMRG: Error-buffer CCERBUF should be printed
5VSPEED: Variable speed is allowed
5SPNFOUND: Speed not found when variable speed allowed
5TELEFIX: Telefix terminal
5FRER: Framing error on input
5PAER: Parity error on input
5BFUL: Characters lost on input due to full input buffer

12. SECURITY PRIMITIVES

The security primitives introduced in the J-version of SINTRAN III are extended slightly in the K-version. The variable named EXSECURITY now contains the following security flags:

Bit #0: No listing of command lines in the @TERMINAL-STATUS command except for own user. If the command is performed by user SYSTEM, the command lines for all background programs logged in will be listed. The command lines will also be listed for the background programs running under the same user as the one executing the @TERMINAL-STATUS command.

Bit #1: The background segment, both program and data bank, will be set to zero when logging out. This feature will delay the logout sequence considerably (seconds). If the background program was terminated abnormally, this zeroing will take place when you log in the first time after the abnormal termination.

Bit #2: The scratch file pages written to in the last session, will be set to zero when logging out. This will also slow down the logout sequence.

Bit #3: Zeroing of pages released from a file, normally in the @DELETE-FILE command.

Bit #4: Not allowed to log in if the user has no password. Only one login without a password is allowed after @CREATE-USER.

Bit #5: The commands @HELP and @LIST-REENTRANT will only list commands and reentrant subsystems/ND-500 standard domains available to the user giving the command. An unprivileged user will thus not "see" commands available only to users SYSTEM or RT.

The default value of the variable EXSECURITY is 7 (bits 0, 1 and 2 are set) but this may be changed by the @SINTRAN-SERVICE-PROGRAM command *CHANGE-VARIABLE.

13. MEMTOF

This chapter applies to the VSX-version of SINTRAN III version K only.

MEMTOF (MEMory TO Floppy dump) for the VSX-version is now a part of SINTRAN III (installed as part of SINTRAN), and may be run by this simple procedure:

- stop the system (if it isn't already stopped)

- dump the register block (use the OPCOM command 0<17RD)

- dump the internal registers (use the OPCOM command IRD)

- press the MCL (master clear) button on the panel

- type 15! (just 15 and an exclamation mark - without a return)

MEMTOF will then start, and ask you to insert formatted diskettes (one after another) in floppy unit 0 of floppy controller 1.

When the dump is finished, remove the diskettes, label them, and enclose the printout of the register contents when you send it to ND service.

14. RT-LOADER

14.1 REMOVED COMMANDS

The command IMAGE-LOAD is removed from the VSX-version, but is still present in the VSE-version.

14.2 CHANGES IN DATA STRUCTURE

The link table element and the RTFIL element in the VSX-version are both changed to use two words for containing segment numbers:

Link Table Element SINTRAN III/VSX version K;

	Layout	Displacements
0	Link	LLINK
1 2 3	Packed symbol name	LPNA1 (LFNAM) LPNA2 LPNA3
4	segment one	LSEG1 (LSEGD)
5	segment two	LSEG2
6	Priority/Flag	LDESC
7	Symbol value	LSUB1 (LSUBD)
10	Stadr/Sublist	LSUB2

Values of bits 0-2 in Flag (no change from J-version):

0	Defined symbol (entry point)
1	Referenced symbol
2	Defined common label
3	Declared common label
4	Global defined common label
5	Segment name
6	Declared RT-program
7	Defined RT-program

RTFIL Element SINTRAN III/VSX version K:

	Layout	Displacements
0 1 2	Packed symbol name	RFNA1 (RFNAM) RFNA2 RFNA3
3	segment one	RFSG1 (RFSGD)
4	segment two	RFSG2
5	Priority/Flag	RFDSC
6	Symbol value	RFSB1 (RFSBD)
7	Stadr/Sublist	RFSB2

The link table element and RTFIL element in the VSE-version are unchanged from previous versions.

15. ND-500 MONITOR

Note that the ND-500 System Monitor is now part of SINTRAN III/VSX-500 and is installed together with SINTRAN itself. Also note that you must change both the ND-500 Background Monitor (the subsystem part) and the ND-500 Swapper to run under the K-version of SINTRAN III.

15.1 CHANGED INSTALLATION PROCEDURE

The installation procedure for the ND-500 Background Monitor and Swapper is unchanged.

The ND-500 System Monitor will now be delivered on the SINTRAN III diskettes, and will be installed as part of SINTRAN III. This means that explicit installation of the ND-500 System Monitor done at cold start must be removed. The commands to be **REMOVED** will usually be found in the file HENT-MODE:MODE and should look like this:

```
@RT-LOADER
*READ-BINARY (BPUN-FILES)ND-500-SEG30-F:BPUN,30
*YES
*READ-BINARY (BPUN-FILES)ND-500-SEG31-F:BPUN,31
*YES
*EXIT
```

15.2 CONFIGURATION LIMITATIONS

The ND-500 systems may now run up to 254 processes. By default, the number of processes are set to 50, and may be changed by the S3-CONFIG program.

Furthermore, the number of physical segments on the ND-500 may now range up to 5000_8. Default value is ten times the number of processes plus 250. This value may be changed by the SET-SYSTEM-PARAMETERS command in ND-500 Monitor.

15.3 MODIFIED COMMANDS TO THE ND-500 BACKGROUND MONITOR

15.3.1 DEFINE-MEMORY-CONFIGURATION

The memory configuration is now defined automatically each time the system is started. This means that the DEFINE-MEMORY-CONFIGURATION command is no longer necessary unless you want to define the memory configuration differently from the default configuration. An explicit definition of memory configuration will, however, not survive a restart of the system, so you will have to do this after each restart. In practice, the change means that the DEFINE-MEMORY-CONFIGURATION command should be deleted from the HENT-MODE:MODE file run after each cold start.
On systems where the memory configuration needs to be defined differently from the default, a DEFINE-MEMORY-CONFIGURATION command must be included in the LOAD-MODE:MODE file to be run after each warm start.

15.3.2 DEFINE-STANDARD-DOMAIN

All types of domains may now be defined as standard domains.

Segment limits defined by the SET-SEGMENT-LIMITS command in the LINKAGE-LOADER when loading the domain are ignored for standard domains.

15.3.3 LIST-TABLE

The LIST-TABLE command is modified to cover the following new tables and functions:

HW-SEGM-TAB

Lists the physical segment table. One entry for each physical segment; the segment number is the index in the table. The use of the commands follows conventions given for the other tables. A single line layout is used for the table when the TO-THE-END command is used.

PROC-TAB

Lists process table entries. There is one entry for each process and the process number is the index in table. The use of the commands follows conventions given for the other tables.

LAST-N500-MSG

If the swapper version used has "message-log" implemented, the last couple (256 in current version) of messages sent from ND-100 to ND-500 may be recalled. Some special conventions are valid only for this table type. The message sent last is listed first and has the highest index number (377_8). Use of commands PREVIOUS/NEXT will go backwards/forward respectively in "time". Index zero gives the oldest message. PREVIOUS is the default command on CR when entering the mode.

NOTE 1: If using the TO-THE-END command, a convenient shorthand layout is used for the table in order to condense the most relevant information. Special care is taken with parameters for the PAGE-FAULT message. Here the page-fault address is given in 32-bit format, followed by the last three parameters. For all other messages, the six first parameters are listed.

Note 2: While listing the message-log on a "living system", the current last message is not updated if more messages arrive while in the mode. New messages will overwrite the oldest messages (index zero and up).

MEMORY-MAP

The new version has different layout, and contains some more information. A single line layout is used for the table when the TO-THE-END command is used.

SW-SEGM-TAB

The new version is extended to include all information contained in this table, also some indirect information retrieved through pointers from the table. Subcommands are made to obtain more information about swapper-files. Bit maps for REFER, SYSFIX and FIX are listed in terms of process numbers concerned. Bit map for MODIFIED lists page numbers modified.

The following additional commands are available when listing the swap-segments-table:

REFER
SYSFIX
FIX

Lists bit maps (process numbers) as represented in the table but now in full version. The bit maps may be larger than what is possible to display in the standard layout of the table. These commands will list the whole bit maps over several lines if necessary.

MODIFIED

As above, but represents the bit map (logical process number) for modified pages in swap-file.

PAGELINK
IX-PAGELINK

Starts memory-map listing in follow-link mode, by using page/indexpage-link from the current swap-entry as first page.

ORIG-FILE
SWAP-FILE

These are commands to list file-buffers for original-files and swapper-files using a given segment, (the one currently displayed). The pointer used is IXBP for the file concerned.

Both commands enter a sub command-processor, and give the prompt "FCOM:".

The initial screen shows the first file-buffer. Each buffer contains up to 64 page numbers. More buffers exist as appropriate, and describe the rest of the file's pages.

FCOM SUBCOMMANDS

NEXT

The command NEXT (or simply pressing CR), gives the next buffer in the same file. The command follows the pointer NX-IXB. When no more buffers exist, the error message "END OF INDEX-BUFFER LIST" appears.

FIRST

Responds by giving the first buffer.

THIS
Lists the current buffer.

EXIT
Returns to LIST-TABLE, (LTB:).

SUC
PREDEC
Lists buffers from the global list for the swapper. This is a circular linked list which contains all the file-buffers in the system. SUC and PREDEC pointers point in different directions around the list. The commands use the pointer with the same name as the command, and displays the next buffer in the list.

LIST-GLOBAL-LIST <full or partial>
Lists the whole global list by following the SUC-pointers, finishing with the current buffer. Giving FULL as parameter lists each buffer fully. PARTIAL, which is the default, gives only the current buffer address, SUC-pointer, PREDEC-pointer, and the flag NEW INDICES, on one line for each buffer.

NOTE: listing may be stopped by pressing any key.

OWNER
Finds and lists the swap-segment entry which "owns" the current file buffer. This command will also return control to the LIST-TABLE command-processor (LTB:).

HELP
Lists available commands.

@
SINTRAN commands may be used here with the same limitations as for the rest of the monitor.

FIELD-MATCH
COMPLIMENT-MATCH-KEYS
RESET-FIELD-MATCH

These are tools for searching in the tables. It is possible to set several conditions (up to 21 in the current version) within the table entry.

FIELD-MATCH <field name> <mask> <match value>

Sets one search condition for the current table type.

<field name>:
Name the field that is going to contain a specific value. The name may be abbreviated (follows ND conventions). If "HELP" is typed here, the available field names for the current table type are listed.

<mask>:
A mask that the field value is ANDed with before comparison. This is useful for separating specific bits or digits within a field. Default here is all 1s (ones), which has the effect of no masking.
(Alternatively a "mask" for the HELP option may be typed here.)

<match value>:
The value the result shall match up to. Default value is the value in the same field in the current table (last listed). NOTE: Enumerated values in the tables may not be entered by name, but must be entered by value. It is advisable to use default possibilities in such cases.

The FIELD-MATCH command may be repeated several times to set more than one condition, for different fields or even for the same field.

When conditions have been set, the commands NEXT/PREVIOUS will list the next/previous table entry that matches all conditions.

A search may be stopped by pressing any key. The conditions which have been set are not lost before conditions for another table type are set unless the following command is given:

COMPLIMENT-MATCH-KEYS

Default match mode is INCLUSIVE. In other words table entries are listed only if all the conditions are true, (as described above).

EXCLUSIVE mode may be toggled on/off by the COMPLIMENT-MATCH-KEY command. This is: list only records where all the conditions do not match.

The entries listed in EXCLUSIVE mode are therefore the complimentary set of those listed in INCLUSIVE mode.

RESET-FIELD-MATCH

This command will reset all match conditions which have been set, and the commands NEXT and PREVIOUS will have normal functions.

LIST-ADDRESS <table start address>
OR
<address>:

Lists the table of current type having the given start address.

MATCH-ADDRESS <address>
OR
<address>;

Lists a table of the current type where the given address is within the table.

TRACK-SWAPPER-MESSAGES <process>

Lists messages to the swapper from a given process as they arrive, default is all processes. Listing may be aborted by pressing any key.

Note: This command is intended for logging single events. It may not be able to track messages if they arrive too frequently over a period of time, as incoming messages may catch up with the output in the message ring buffer.

HELP

Lists available commands.

@

SINTRAN commands may be used here with the same limitations as for the rest of the monitor.

OTHER MODIFICATIONS

For the tables SW-SEGM-TAB, HW-SEGM-TAB, MEMORY-MAP and PROC-TAB, the user now may select the entry index directly as an optional parameter.

DEFAULTS: When going between the above mentioned table types, without stating the entry index, several defaults will be assumed. These will depend on each case to suit the most relevant situation.
The defaults are:

Going from HW-SEGM-TAB to:
- SW-SEGM-TAB, uses the same segment entry
- MEMORY-MAP, uses PAGENO

Going from SW-SEGM-TAB to:
- HW-SEGM-TAB, uses the same segment entry
- MEMORY-MAP, uses PAGELINK (as "PAGELINK" command)
- PROC-TAB, uses FIRSTPROC

Going from PROC-TAB to:
- HW-SEGM-TAB, uses PROCESS-SEGNO
- SW-SEGM-TAB, uses PROCESS-SEGNO

Going from MEMORY-MAP to:
- HW-SEGM-TAB, uses OWNER
- SW-SEGM-TAB, uses OWNER

Otherwise the default is zero.

Note: When going to MEMORY-MAP with a non-zero default entry index, follow-link mode is selected. Otherwise follow-table mode is used.

15.3.4 LOOK-AT

The LOOK-AT commands:

LOOK-AT-DATA
LOOK-AT-STACK
LOOK-AT-RELATIVE
LOOK-AT-REGISTER

are modified to work on dumpfile in inspect-dump mode.

LIST-TABLE <address>
MATCH-TABLE <address>

Intended to be used when looking at the swapper's data segment. This is a way of going directly into LIST-TABLE from LOOK-AT. Useful when a table pointer is found at a location, etc. Uses current location value as default pointer if no address given. Otherwise the commands have the same effect as commands with same name in LIST-TABLE. The commands will leave the user in LIST-TABLE mode. EXIT returns to LOOK-AT.

SET-SEARCH-OBJECT
SET-SEARCH-ADDRESSMODE
SEARCH

These commands are used to find specific data sequences, and are implemented for the following look-at modes:

LOOK-AT-DATA
LOOK-AT-PROGRAM
LOOK-AT-FILE
LOOK-AT-RESIDENT

SET-SEARCH-OBJECT <BYTE/HALF/WORD> <MASK> <EQ/NE/GT/LT/IN> <VALUE>

Defines what to search for.

<BYTE/HALF/WORD>:
Defines the size(s) of the data element(s); if more than one, each is separated by a semicolon. Default value is the current look-at size. Size qualifier may be abbreviated.

<MASK>:
Defines mask(s) through which data is ANDed with before comparison; also separated with semicolons if more than one. Default are all 1s (ones), i.e., no masking.

<EQ/NE/GT/LT/IN>:
Sets the search mode for each element. These may be:

- EQ, equal to value
- NE, not equal to value
- GT, greater than value
- LT, less than value
- IN, in a defined set (range)

Separated by semicolons if more than one. Default value is EQ.

<VALUE>:

Defines the final value sequence to search for, separated by semicolons if more than one. Sets or ranges are defined by two numbers in ascending order only separated by a colon.
Complimentary sets are defined by changing the two values (largest number first).

23:46 is the set from 23 to 46 inclusively.

46:23 is the complimentary set of 23:46. In other words values less than 23 or greater than 46.

ASCII SEARCH

A special case is available when ASCII-string search is wanted. A string may be entered directly as value a enclosed in single quotes ('). All above parameters are then ignored. Default search address mode will be in BYTE-mode (see the command SET-SEARCH-ADDRESSMODE).

SEARCH <from address> <to address>

Starts searching for the chosen value or sequence. Parameters here are optional, and give a possibility to limit the search area. Default start address is the current address, not inclusive. Default stop address is maximum address within segment. Search may be broken at any place by pressing any key.

NOTE: Search will terminate on illegal addresses, at "holes" in the file and at End-Of-File. This will cause the look-at mode to terminate as well, so the user is advised to set the top limit while searching, as the search data is lost. Using the break facility is another possibility.

SET-SEARCH-ADDRESSMODE <BYTE/HALF/WORD>

Defines address type for the start of the sequence. this is a way of eliminating "imaginary" sequences that may occur "out of phase".

BYTE: Sequence may start at any byte-address (no limitations).

HALF: Sequence must start on half word limit (address divisible by 2).

WORD: Sequence must start on a word limit (address divisible by 4).

Default address mode is the one currently used in LOOK-AT.

For special cases, a number may be entered to give other modes. This is useful to speed up the search when the sequence is known to be found at specific address-limits.

15.3.5 NEW INTEGER INPUT FORMAT

In both LOOK-AT and LIST-TABLE it is now possible to write addresses with shorthand notation for the page number.

The string 1'23"456 is interpreted as address 456 on page 23 in segment 1, in other words the address 1000114456, (all octal). This may be written as an address or as a location expression. This is useful when accessing page tables etc.

15.3.6 USE OF NEW SEARCH COMMANDS

FIELD-MATCH COMMAND IN LIST-TABLE

EXAMPLE 1.

Suppose you want to list all swap-segment entries with the flags IN-SYSTEM-DOMAIN and SHARED-SEGMENT set:

```
N500: LIST-TABLE SWAP-SEGM-TAB

(table entry zero will be listed).
.
.

LTB: FIELD-MATCH
Field name: IN-SYSTEM-DOMAIN
Mask:
Match value: 1

or..

LTB: FIELD-MATCH IN-SYSTEM-DOMAIN,,1

The monitor prompts...

Number of match keys :   1

Further...

LTB: FIELD-MATCH SHARED-SEGMENT,,1

Number of match keys :   2

LTB:
```

The NEXT/PREVIOUS commands now give the next higher/lower record with both the flags set. The command TO-THE-END will list all the records concerned.

The command RESET-FIELD-MATCH clears the mode.

FIELD NAMES FOR USE WITH FIELD-MATCH COMMAND

SW-SEGM-TAB

BIMLOGSEG	ODSEGNO	PP-PAGE	SHARED-SEGMENT
BIMPRNO	OFILE-PTR	PROCESS-SEGMENT	SINTRAN-III-SHARED
EXPANDABLE	OFILE-TYPE	REFP	SIXBP
F-LOGPAGE	OFILESTART	S-ERRCODE	SL1OPIND
FIRSTFIX	OIXBP	SDDOMNO	SMAX-MEMPAG
FIRSTPROC	OL1OPIND	SDEVICE	SMAXVERS
FIRSTSYSFIX	OLDSIBAS	SDIR-BASE	SMIN-MEMPAG
FREE-SP	OMAXVERS	SDSEGNO	SN-MEMPAG
IN-SYSTEM-DOMAIN	ONBLK-P-PAGE	SEGERROR	SN-PF
IX-PAGELINK	OP-FILNO	SEGM-SIZE	SNBLK-P-PAGE
MODIFIED	ORESLOCK	SEGMEMTYPE	SRESLOCK
N-PREF	ORIGINAL-DUMMY	SEGMTYPE	SUNIT
O-FILDISP	ORIGINAL-SEGMENT	SEQUENTIAL-ACCESS	SW-CAND
ODDOMNO	OUNIT	SFILE-PTR	TAKE-WIP
ODEVICE	PAGELINK	SFILE-TYPE	THIS-SEGNO
ODIR-BASE	PLPNO	SFILESTART	WRITE-PERMITTED

PROC-TAB

CL-TIME	P	ST1	CTE2
MAX-FIXPAGES	L	ST2	MTE1
MEM-TIME	B	PS	MTE2
N-FIXPAGES	R	TOS	TEMM1
NEXT-CLC	I1	LL	TEMM2
NEXT-OUT	I2	HL	
P-PFTIME	I3	THA	
PMAX-MEMPAGES	I4	CED	
PMIN-MEMPAGES	A1	CAD	
PN-PF	A2	CES	
PRIO	A3	CAS	
PROCESS-SEGNO	A4	MIC1	
BIMRP	E1	MIC2	
STATE	E2	OTE1	
STATE-LINK	E3	OTE2	
THIS-PRNO	E4	CTE1	

MEMORY-MAP

FIXSTATE
MLPNO
NEXT-PAGE
OWNER
PGU-CLEAR
PMEMPT
USAGE
WRITTEN-IN-PAGE

LAST N500-MSG

MSGTYPE	P6
SENDER	P7
P1	P8
P2	P9
P3	P10
P4	P11
P5	P12

HW-SEGM-TAB

IX-TYPE
PAGENO

SEARCH COMMANDS IN LOOK-AT

EXAMPLE 2.

The user wants to search in a data segment for a sequence of three elements:

First a word equal to 1460003_8. Then a half-word (16 bits) greater than 4000 decimal followed by a byte from 'A' to 'Z'.

```
N500: LOOK-AT-DATA 1'

D1          OB:         OB SET-SEARCH-OBJECT
BYTE/HALF/WORD: W;H;B
Mask:
EQ/NE/GT/LT/IN: EQ;GT;IN
Value: 146003;4000D;101:132
```

Suppose the sequence is to be found between the start of segment and the address 1'200000...

```
D1          OB:         OB SEARCH,,1'200000
```

The monitor will now search through the segment and terminate if the sequence is found, or the end of the search area is reached. If it is found, the address of the start of the sequence is displayed and control returns to the user:

```
D1      123411B:    146003B _     (etc.)
```

EXAMPLE 3.

Search for an ASCII string 'NORSK-DATA' on a file DAT:DATA ...

```
N500: LOOK-AT-FILE 0 DAT:DATA

<file>0       OB:           OB SET-SEARCH-OBJECT,,,,'NORSK-DATA'
<file>0       OB:           OB SEARCH
.
.
.
```

etc.

15.4 NEW COMMANDS TO THE ND-500 BACKGROUND MONITOR

15.4.1 DEBUG-SWAPPER

Parameter: <ON/OFF>

This command is installed for debugging purposes. It is intended for internal use and is restricted to user SYSTEM only.

15.4.2 DUMP-PHYSICAL-SEGMENT

Parameters: <FILE NAME>
<PHYSICAL SEGMENT NUMBER>

This command is installed for debugging purposes. It is restricted to user SYSTEM only.

15.4.3 DUMP-SWAPPER

Parameter: <FILE NAME>

This command is installed for debugging purposes. It is intended for internal use and is restricted to user SYSTEM only. If this command is to be used after a fatal error from swapper, an error flag must be reset before the command is given. The error flag is reset by giving the two commands: DEBUG-SWAPPER ON followed by DEBUG-SWAPPER OFF.

15.4.4 INSPECT-DUMP

Parameter: <FILE-NAME>

This command sets the monitor in inspect-mode. The file is supposed to be a mapped copy of the swapper's data segment. Default file type is ":DUMP". In this mode the LOOK-AT and LIST-TABLE commands will work on the dump file rather than on segments directly. The monitor prompt will be preceeded by the text, "<dump>", while inspect-dump mode is active. It is illegal to enter inspect-dump mode when segments are placed. The following commands are relevant in inspect-dump mode:

LOOK-AT-DATA
LOOK-AT-STACK
LOOK-AT-RELATIVE
LOOK-AT-REGISTER

Note: LOOK-AT-PROGRAM, LOOK-AT-RESIDENT and LOOK-AT-PHYSICAL-SEGMENT are illegal in inspect-dump mode.
LOOK-AT-CONTROL-STORE, LOOK-AT-HARDWARE and LOOK-AT-FILE will work normally.

LIST-TABLE SW-SEGM-TAB
HW-SEGM-TAB
PROC-TAB
MEMORY-MAP
LAST-N500-MSG

NOTE: The table type N500-MSG still gives the current table as in normal mode, and is not changed by the dump.

15.4.5 RESET-INSPECT-DUMP

Resets inspect-dump mode and sets monitor back in its normal state. An automatic reset will occur if segments are placed while inspect-dump mode is on.

15.5 MONITOR CALLS REMOVED (ND-500)

15.5.1 ABSTR MON 131

MON ABSTR is no longer available from ND-500.

15.6 MODIFIED MONITOR CALLS (ND-500)

15.6.1 N500M MON 60

Function 72_8 GSGTE (Get physical Segment Table Entry) is changed. The location OWNINDEX will now contain object block number. Function 72_8 returns an array of 66_8 words in the format listed on page 132.
Function 71_8 SSGTE (Search for physical Segment Table Entry) also returns this 66_8-word array.
Functions 67_8 SPRTE (Search for Process Entry) and 70_8 GPRTE (Get Process Entry) both returns an array of 123_8 words in the format shown on page 130.
Function 60_8 LIMEM (LIst MEMory configuration) now returns the ND-100 page address of the ND-500 register block when called from background (no longer the address of "page-used/written-in-page" table for the ND-500). For call from foreground, there is no change.
MON 60 is for internal use by ND only.

15.6.2 MAGTP MON 144

Functions 50, 51 and 52 are now available from ND-500. For direct transfer on STC magnetic tape, functions 50 and 51 will be converted internally to MON ABSTR (MON 131) functions 50 and 51 respectively.

Function 50 : Read multiple records, i.e., read a number of records from magnetic tape into a contiguous area of memory.

Function 51 : Write multiple records, i.e., write a contiguous area of memory to a number of records on magnetic tape.

Call format:
MAGTP,<number of parameters>,<function>,<buffer>,<logical device no>,
<number of records>,<record size>

Input parameters:
<function>
<buffer>
<logical device number>
<number of records to read/write in one call>
<record size in bytes>

Output parameters:
<number of records read/written>
<record size in bytes of record read>

On error return, the <number of records read/written>, will be the number of records read or written before the error occurred.

```
Example:
FUNCTION : W  BLOCK 1                % function
BUFFER   : BY BLOCK 4000B            % buffer, 2048 bytes
UNIT     : W  BLOCK 1                % logical device number
NOREC    : W  BLOCK 1                % number of records
RECSIZE  : W  BLOCK 1                % record size
MAGTP    : EQU 37000000144B          % MAGTP = MON 144

CALLG MAGTP,5,FUNCTION,BUFFER,UNIT,NOREC,RECSIZ % MON MAGTP with 5 par
IF K GO ERROR                        % on error return, W1 = error code
```

15.6.3 FSMTY MON 327

Three new functions are added: function = 2 : return block size
= 3 : get file name
= 4 : get file/device info.

Parameters: <function = 2>
<open file number>
<block size in bytes>

or: <function = 3>
<open file number>
<buffer to receive file name>

or: <function = 4> (same as MON GTYPR (MON 45))
<open file number or device number>
<returned TYPRING>
<returned status>
<returned SINTRAN III open file number or device no.>

For further details on MON 327, refer to page 17.

```
Examples:
FUNCTION  : W DATA 2                 % function code = 2
OPFILNO   : W BLOCK 1                % open file number
BLOCKSIZE : W BLOCK 1                % block size
FSMTY     : EQU 37000000327B         % FSMTY = MON 327

CALLG FSMTY,3,FUNCTION,OPFILNO,BLOCKSIZE % MON FSMTY with 3 parameters
IF K GO ERROR                        % on error return, W1 = error code

--------------------

FUNCTION : W DATA 3                  % function code = 3
OPFILNO  : W BLOCK 1                 % open file number
FILENAME : STRING 64                 % file name (string descriptor)
FSMTY    : EQU 37000000327B          % FSMTY = MON 327

CALLG FSMTY,3,FUNCTION,OPFILNO,FILENAME  % MON FSMTY with 3 parameters
IF K GO ERROR                        % on error return, W1 = error code
```

```
FUNCTION : W DATA 4                    % function code = 4
OPFILNO  : W BLOCK 1                   % open file number
TYPRING  : W BLOCK 1                   % returned TYPRING
STATUS   : W BLOCK 1                   % returned status
S3OPFILN : W BLOCK 1                   % returned SINTRAN III open
FSMTY    : EQU 37000000327B            % FSMTY = MON 327

CALLG FSMTY,5,FUNCTION,OPFILNO,TYPRING,STATUS,S3OPFILN  % MON FSMTY
IF K GO ERROR                          % on error return, W1 = error code
```

15.6.4 IOMTY MON 336

The I/O multifunction (IOMTY) monitor call is used to change some of the attributes of terminal and terminal access device (TAD) input and output. This monitor call need a varying number of input and output parameters depending upon function, all parameters are therefore placed in an array.

Parameters: <function>
<number of parameters (32-bit integers)>
<array of parameters>
<status-2>

The following rules apply:

1. The I1-register will contain status information corresponding to what is usually returned in the A-register on the ND-100. The parameter <status-2> contains information which is returned in the X-register on ND-100.
2. All parameters must be on 32-bit integer format, except for the bit maps referred to in functions 4 - 7. The echo and break strategy bit maps should be arrays of 16-bit integers (ie., INTEGER*2).

For further details on MON 336, refer to pages 19-36.

Example:

```
FUNCTION  : W  BLOCK 1                 % function
SIZE      : W  BLOCK 1                 % number of parameters
PARAM     :                            % start of array of parameters
PARAM1    : W  BLOCK 1                 % parameter no. 1 in array
PARAM2    : W  BLOCK 1                 % parameter no. 2 in array
    :                                          :
PARAMN    : W  BLOCK 1                 % parameter no. N in array
RETSTATUS : W  BLOCK 1                 % returned status
IOMTY     : EQU 37000000336B           % IOMTY = MON 336

CALLG IOMTY,4,FUNCTION,SIZE,PARAM,RETSTATUS % MON IOMTY with 4 params.
IF K GO ERROR                          % on error return, W1 = error code

%  Note: on error return, RETSTATUS contains extra information for
%                         function codes ≥ 100₈
```

15.7 NEW MONITOR CALLS (ND-500)

15.7.1 RSREC MON 340

Purpose: Read system record - RT-DESCRIPTION
or - ND-100 segment table entry

Parameters: <record type>
<RT-description address / segment number>
<buffer>
<format>

The parameter <record type> may take the values:

1: RT-description
2: ND-100 segment table entry

The parameter <format> (not found on the ND-100) may take the values:

0: Return information on 16-bit integer format
1: Return information on 32-bit integer format

For further details on MON 340, refer to page 37.

Example:

```
RECTYPE  : W  BLOCK 1            % RT-description/segment entry
ADDNO    : W  BLOCK 1            % address of RT-descr/ND-100 segment
ENTRY    : H  BLOCK 40           % buffer, 40 words
FORMAT   : W  BLOCK 1            % format (16/32-bit integer)
RSREC    : EQU 37000000340B      % RSREC = MON 340

CALLG RSREC,4,RECTYPE,ADDNO,ENTRY,FORMAT % MON RSREC with 4 parameters
IF K GO ERROR                    % on error return, W1 = error code
```

15.7.2 CONFG MON 343

Purpose: CONFG is used to read and/or change configuration parameters for a "standard system" SINTRAN III.

Parameters: <function>
<configuration parameter index>
<configuration parameter sub-index>
<array>

For further details on MON 343, refer to pages 43-61.

Example:

```
FUNCTION : W  BLOCK 1              % function code
INDEX    : W  BLOCK 1              % configuration parameter index
SUBINDEX : W  BLOCK 1              % config. parameter sub-index
BUFFER   : W  BLOCK 10             % array
CONFG    : EQU 37000000343B        % CONFG = MON 343

CALLG CONFG,4,FUNCTION,INDEX,SUBINDEX,BUFFER % MON CONFG with 4 params
IF K GO ERROR                      % on error return, W1 = error code
```

15.8 NEW MONITOR CALL - ONLY AVAILABLE ON ND-500

15.8.1 AttachSegment MON 440

This Attach Segment monitor call (440_8) is used to map a logical ND-500 data segment onto shared ND-100/ND-500 physical memory. The specified physical memory area must be defined in the "Not initialize page" table by use of the *CHANGE-TABLE command in the SINTRAN-Service-Program.

Example:

```
@SINTRAN-SERVICE-PROGRAM
*CHANGE-TABLE
TABLE: MEMORY-AREA-INVISIBLE-FOR-THIS-SYSTEM
FUNCTION: INSERT-ELEMENT
IMAGE OR SAVE AREA (DEFAULT IS IMAGE): IMAGE
FIRST PAGE (OCT): 10000
LAST PAGE (OCT): 13777
FUNCTION: EXIT
*EXIT
```

You must then do a warm start to put the change into effect.

The monitor call has 3 functions:

```
Function = 0 : detach (forget) a previously attached segment.
Function = 1 : attach segment. If physical segment does not exist,
               create and map segment onto physical ND-100 address
               area.
Function = 2 : map existing segment onto physical ND-100 address area.
```

Function 0

Function description:
Detach (forget) a previously attached segment.

Parameters:
<function = 0>
<segment number in the range $0:31_{10}$>

Example:

```
FUNCTION : W DATA 0                    % function = 0
SEGNO    : W DATA 5                    % segment no. 5

CALLG 37000000440B,2,FUNCTION,SEGNO % Mon 440 with 2 parameters
IF K GO ERROR                          % on error return, W1 = error code
```

Function 1

Function description:
Attach segment. If physical segment does not exist, create and map segment onto physical ND-100 address area.

Parameters:
<function = 1>
<segment number in the range $0:31_{10}$> If segment no. = 0, the first free segment will be used.

<length of segment in bytes>
<physical ND-100 word address>
<segment name> Maximum 36_{10} characters in segment name, including optional user name. Must be terminated by an apostrophe. The parameter is a string descriptor.
<read/write area> 0 = read only access, 1 = read/write access
<returned logical segment number>

Example:
```
FUNCTION : W DATA 1                  % function = 1
SEGNO    : W DATA 0                  % use first free segment
LENGTH   : W DATA 4000B              % segment length = 2056(10) bytes
N1ADDR   : W DATA 20000000B          % mapped at address 200000008
SEGNAME  : STRINGDATA 'ATTSEG1'      % segment name = 'ATTSEG1'
ACCESS   : W DATA 1                  % read/write access
RETSEGNO : W BLOCK 1                 % returned segment number

CALLG 37000000440B,7,FUNCTION,SEGNO,LENGTH,N1ADDR,SEGNAME,ACCESS,&
      RETSEGNO                       % Mon 440 with 7 parameters
IF K GO ERROR                        % on error return, W1 = error code
```

Function 2

Function description:
Map existing segment onto physical ND-100 address area.

Parameters:
<function = 2>
<first address> ND-500 logical data address where segment starts. Must be the lower address of the existing segment.
<length of segment in bytes> Must always cover the whole segment.
<physical ND-100 word address>

If the segment is shared, the segment cannot be used by other processes at the time of the call. If the segment is used, the call will give an error return.

Example:
```
FUNCTION : W DATA 2                  % function = 2
STADDR   : W DATA 2000000000B        % segment no. 2, lower bound = 0
LENGTH   : W DATA 4000B              % segment length = 2048(10) bytes
N1ADDR   : W DATA 20000000B          % mapped at address 200000008

CALLG 37000000440B,4,FUNCTION,STADDR,LENGTH,N1ADDR   % Mon 440
IF K GO ERROR                        % on error return, W1 = error code
```

15.9 SOME NOTES ON FILES USED FROM ND-500

- When a file is connected as segment, the file access specified will also reflect the segment access.

- Files may be opened for Common Write from ND-500 and simultaneously opened from ND-100 programs. This may cause problems, however, since the ND-100 and the ND-500 in this case access the physical file differently. You should use MON WSEGN (MON 416) frequently to avoid problems in this case.

- If a file open for use from ND-500 is set permanently open, it will still be closed by the command @CLOSE-FILE -1 or MON CLOSE (-1) executed from the ND-500.

- Using the command CLOSE-FILE in the Monitor calls MON COMND (MON 70) or MON UECOM (MON 317) will not close files open from ND-500 programs as seen from the ND-500. The files will be closed on the ND-100.

15.10 CHANGED DATA STRUCTURES (ND-500)

PROCESS TABLE ENTRY

0	Process table entry status (0=free/1=used)
1	Process physical segment number
2	Process name (22_8 words)
24	Program capability table (40_8 words)
64	Data capability table (40_8 words)

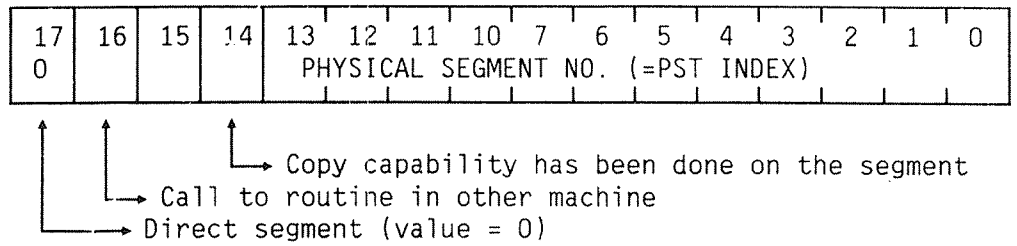
Information returned by MON N500M functions 67_8 and 70_8. See page 124.

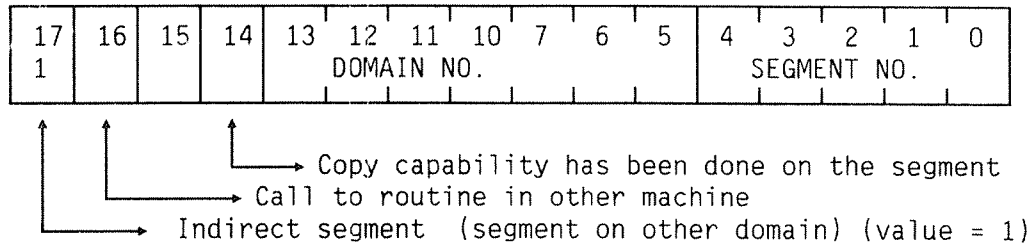
The capability tables for program and data segments are kept in the 'stack' page on the data segment belonging to the process.

PROGRAM CAPABILITY TABLE

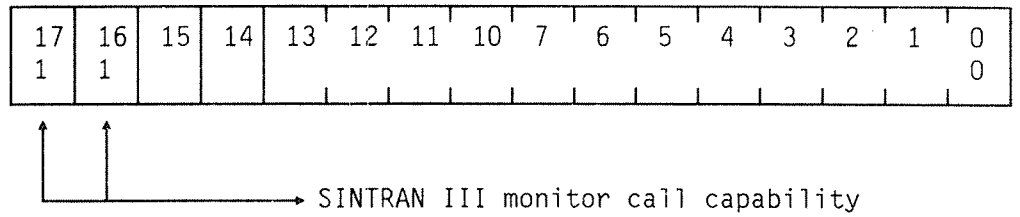
The size is one word for each of the 40_8 program segments.

Program capability:





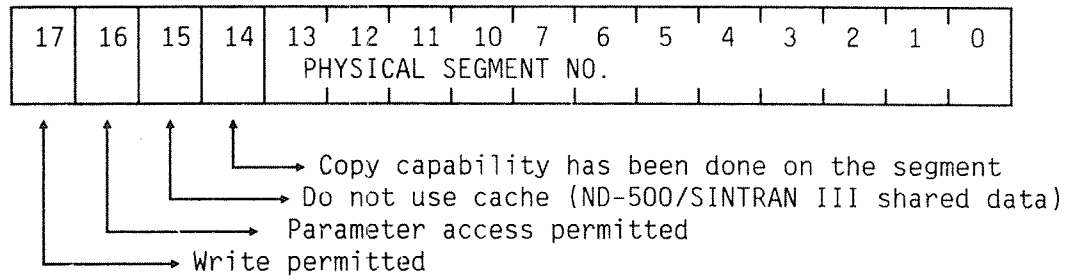
Logical program segment no. 37_8:



DATA CAPABILITY TABLE

The size is one word for each of the 40_8 data segments.

Data capability:



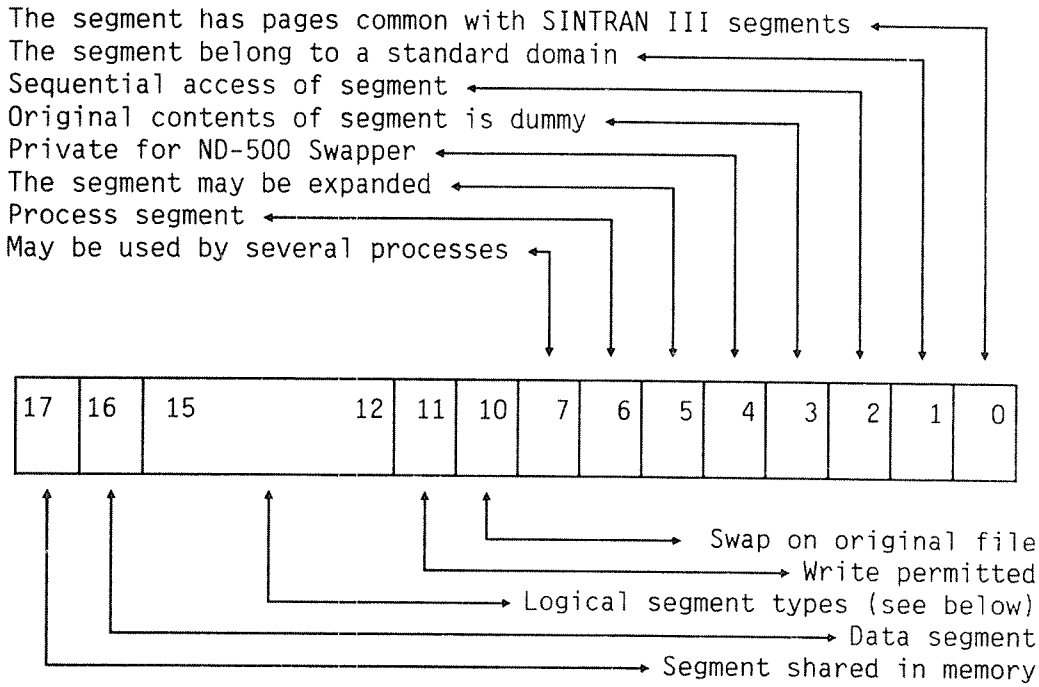
ND-500 MONITOR'S SEGMENT TABLE ENTRY

0	Link to next physical segment in list	
1	Link to previous physical segment in list	
2	Physical segment number	
3	Physical segment type (see next page)	
4	Process number of the 1st process using the segm.	
5	Logical segment no. of the 1st process using this segment	
6	Open file number if "swap on original"	
7	Directory index of file	User index of file
10	Object block no.	Object index of file
11	Number of processes using the segment	
12	Address of swap-file-part if swapped on swapfile Address of segment name if attached segment	
13	Segment size in pages	
14		
15	Displacement of segment start in file (in pages)	
16		

17	Logical device number of segment directory
20	Logical unit number of segment directory
21	Number of sectors per page in segment directory
22	Page address of the segment's index page
23	
24	Segment directory base address in sectors
25	
26	Physical segment name (40_8 words)

Information passed to swapper when segment is placed
This is not part of the monitor's segment entry.

Physical segment type bits:



Logical segment types (bits 15-12):

0 - Reentrant program
1 - Debug program
2 - Other machine
3 - Indirect segment
4 - Read-only segment
5 - Copy read/write shared
6 - Copy read-only shared
7 - Copy exclusively
10 - Original read/write shared
11 - Original shared
12 - Original exclusively
13 - Scratched shared
14 - Scratch exclusively
15 - Attached shared

16. XMSG VERSION K

16.1 REQUIREMENTS

XMSG version K can only run under SINTRAN III version K. It is also the only version of XMSG that may be run under SINTRAN III version K.

XMSG version K is tailored to SINTRAN at installation time, the product contains code for two different versions of XMSG:

- one to be run under SINTRAN III/VSE - this version will be entered on PIT 3 in much the same way as previous versions.

- the other to be run under SINTRAN III/VSX - this version will be entered on PIT 6 - XPIT.

Also note that the COSMOS file transfer server (XFTRA) is no longer part of XMSG (ND-210373), but has been moved to COSMOS Basic Module (ND-210374).

16.2 COMMANDS MODIFIED - XMSG-COMMAND PROGRAM

16.2.1 DEBUG-MODE

To use this command to debug a dumped XMSG system, both the system to be debugged, and the running system must be version K.

16.2.2 DEFINE-NETWORK-CONNECTION

This command now accepts a string of 90 characters for the parameter <Remote DTE address>. The name given as the parameter <Port or System name?>, may not start with a digit (0-9).

16.2.3 DEFINE-NETWORK-REMOTE-ENDPOINT

Two of the parameter prompts have been changed:
'Dial-up cost?' is changed to 'Connect charge?' and
'Data cost/ksegment' is changed to 'Connect time?'.

16.2.4 LIST-NETWORK-REMOTE-ENDPOINTS

When accessing a COSMOS X.25 server which is able to handle this command and the response from the X.25 server indicates the speed to be 0 (zero), the field 'Speed' will be filled with spaces.

16.2.5 LIST-SERVERS

This command replaces the old command LIST-SERVICE-PORTS. The command asks an XROUT to dump out, from its name table, all named ports.

16.2.6 LIST-SERVICE-PORTS

This command has been removed. It is replaced by the command LIST-SERVERS.

16.2.7 LIST-VERSION

The information listed by this command has been extended to include system name, version/revision, patch level and product number/name. This information is also listed for all network servers used by XMSG.

16.2.8 SET-MAXIMUM-HOP-COUNT

This command is now only available in advanced mode.

16.3 MODIFIED FUNCTIONS

16.3.1 CREATE DRIVER WITH CONTEXT (XFCRD)

If run under SINTRAN III/VSX, this function now requires that the XFPON option is set - because a driver must always run with paging on.

16.3.2 RECEIVE AND READ MESSAGE (XFRRE)

A new option bit (XFRMR) has been implemented in XFRRE which allows a task to receive a message, read the data, and, if the last byte in the message was read, release the message.

16.3.3 SENDING MESSAGE (XFSND)

A new option bit (XFTCM) in XFSND provides a way of sending the 'task current' message (without having to set the message as the 'port current' message using the Change Current Message (XFSCM) function).

16.4 NEW FUNCTIONS

16.4.1 FREE ALLOCATED MESSAGE BUFFERS (XFFRM)

The XFFRM function is used to free message buffers previously allocated by the XFALM (allocate message buffers) function.

16.4.2 WRITE AND RETURN MESSAGE (XFWRT)

As an alternative to the 'Write Message' (XFWRI) and 'Send Message' (XFSND) functions, the task can execute the XFWRT function which will transfer the user data into a message and then send the message back to the port from which it came.

16.5 MODIFIED XROUT SERVICES

16.5.1 GET/CHECK ATTRIBUTE (XSGAT)

This service is divided into two sub-services:
XSGXV - get XMSG version.
XSGLO - get local system number and name.

16.5.2 GET NETWORK SERVER INFORMATION (XSNSI)

It is now possible to call this privileged service from an unprivileged caller.

16.6 NEW XROUT SERVICES

16.6.1 GET INFORMATION ABOUT A LINK (XSLIN)

This privileged service will return information about an active link (i.e., it will return information for an active link which has been started using the Start-Link and/or the Start-Network-Server command of the background command program).

16.6.2 GET INFORMATION ABOUT NAMED PORTS (XSPIN)

This service allows any user to check and obtain information about named ports by sending a message containing a name as input parameter.

17. AFFECTED SUBSYSTEMS

ND-500 MONITOR	only version H of ND-500 Background Monitor (ND-210333) may be used.
ND-500 SWAPPER	only version H of ND-500 Swapper (ND-211034) may be used.
XMSG	only version K of XMSG (ND-210373) may be used.
COSMOS BASIC MODULE	version D of Cosmos Basic Module (ND-210374) is required when running XMSG version K.
BACKUP-SYSTEM	version H of the Backup system (ND-210337) is required to handle files with file index > 255 (more than 256 files per user).
BACON	version dated 860424 or later of Bacon is required.
FILE-MANAGER	versions A or B of the File Manager (part of ND-210518) will not handle files with file index > 255 (more than 256 files per user).
FILE-SYSTEM-INVESTIGATOR	version O of the File System Investigator (part of ND-210022) is required to handle files with file index > 255 (more than 256 files per user).
LINKAGE-LOADER	version H of the Linkage Loader (ND-210319) is required to handle communication with RT-programs due to the changed RTFIL format.
SYMBOLIC-DEBUGGER	version F of the Symbolic Debugger (ND-210336) may be used when running SINTRAN III version K to debug RT-programs.
TELEFIX-LOCAL	version C01 of Telefix-Local (ND-210775) is required.
USER-ENVIRONMENT	version B of User Environment (ND-210518) must be changed slightly to run under the VSX-version of SINTRAN III version K. Version C of User Environment offers a highly improved performance when used under the K-version of SINTRAN.

********** 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: SINTRAN III Release Information
K version

Manual number: ND-60.230.2 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 _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

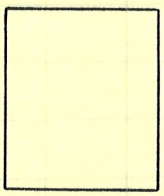
_____

_____

_____

_____

Answered by _____ Date _____



Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo6, Norway