NORD-50 Hardware Maintenance Manual

ND-30.010.02

NORSK DATA A.S

NORD-50 Hardware
Maintenance Manual

ND-30.010.02

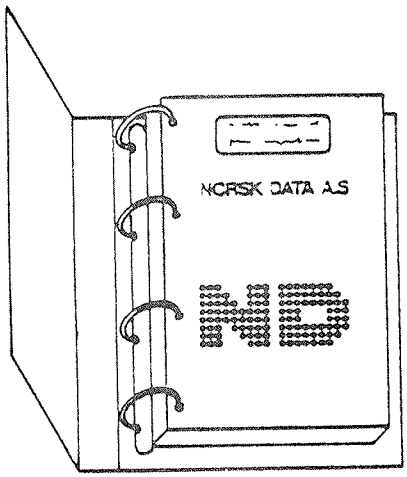
NOTICE

The information in this document is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this document. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S.

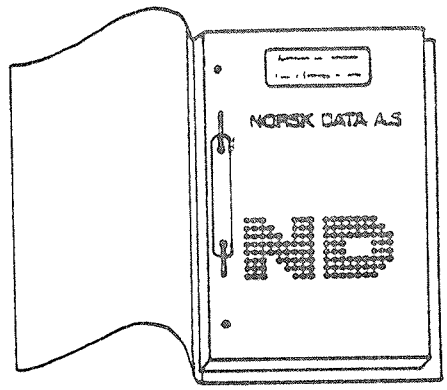
This manual is in loose leaf form for ease of updating. Old pages may be removed and new pages easily inserted if the manual is revised.

The loose leaf form also allows you to place the manual in a ring binder (A) for greater protection and convenience of use. Ring binders with 4 rings corresponding to the holes in the manual may be ordered in two widths, 30 mm and 40 mm. Use the order form below.

The manual may also be placed in a plastic cover (B). This cover is more suitable for manuals of less than 100 pages than for large manuals. Plastic covers may also be ordered below.



A Ring Binder



B Plastic Cover

Please send your order to the local ND office or (in Norway) to:

Documentation Department
Norsk Data A.S
P.O. Box 4, Lindeberg gård
Oslo 10

ORDER FORM

I would like to order

..... Ring Binders, 30 mm, at nkr 20.- per binder

..... Ring Binders, 40 mm, at nkr 25,- per binder

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

Name .....
Company .....
Address .....
.....
City .....

PRINTING RECORD

Printing	Notes
04/81	Version 01
01/82	Version 02

NORD-50 Hardware Maintenance Manual
Publ. No. ND-30.010.02
January 1982

NORSK DATA A.S
P.O. Box 4, Lindeberg gård
Oslo 10, Norway

Manuals can be updated in two ways, new versions and revisions. New versions consist of a complete new manual which replaces the old manual. New versions 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 Bulletin and can be ordered as described below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and to give an evaluation of the manual. Both detailed and general comments are welcome.

These forms, together with all types of inquiry and requests for documentation should be sent to the local ND office or (in Norway) to:

Documentation Department
Norsk Data A.S
P.O. Box 4, Lindeberg gård
Oslo 10

Preface:

THE PRODUCT

This manual covers the:

NORD-50 CPU

The NORD-50 CPU is used in all ND 1400 configurations.

THE READER

This manual has been prepared for Norsk Data Service Department field service engineers and technical personnel directly involved with maintaining the NORD-50 CPU.

PREREQUISITE KNOWLEDGE

A general knowledge of the NORD-10 S hardware including the Input Output and Memory System. A familiarity with the ND Operating System SINTRAN.

THE MANUAL

Maintenance information is provided in eight chapters in this manual. Chapter numbers and a brief description of their contents are listed below:

Chapter 1	-	System architecture and physical layout of the NORD-50 CPU.
Chapter 2	-	NORD-50 hardware summary. Description of the three racks in the NORD-50 CPU and the operators panel.
Chapter 3	-	The NORD-50 Monitor. A NORD-50 Monitor command summary containing the commands needed for hardware maintenance of the NORD-50 CPU.
Chapter 4	-	NORD-50 Error Messages. The error messages are listed with hardware maintenance information.
Chapter 5	-	NORD-50 Test Program Information. Test programs for the external arithmetic and the CPU are overviewed and described.
Chapter 6	-	NORD-50 Memory System. The multiport memory in a NORD-50 configuration is depicted. This section also contains Memory Test Program information.
Chapter 7	-	NORD-50 Memory Expansion. The hardware implementation and the software command SET-MEM is explained in detail.
Chapter 8	-	This section covers the power system.

RELATED MANUALS

Manuals relating to the general and functional description of the NORD-50 are as follows:

Norsk Data Number	Title
ND - 05.003	NORD-50 Reference Manual
ND - 30.007	NORD-50 Test Programs
ND - 05.007	NORD-50 Functional Description
ND - 06.005	NORD-10 NORD-50 Communication System
ND - 60.083	NORD-50 Loader Users Guide
ND - 60.076	NORD-50 MONITOR Users Guide and System Documentation
ND - 60.075	NORD-50 Assembler
ND - 60.095	NORD-50 FORTRAN Reference Manual
ND - 60.098	NORD-50 BRF Editor

The manual that deals with the special systems interfaced with or utilized with the NORD-50 is:

Norsk Data Number	Title
ND - 06.007.01	BIG MULTIPORT MEMORY

These manuals are described in Appendix H, Documentation Review. In addition special hardware and software machine books are prepared for each computer by Norsk Data. These are on site working notebooks containing information about that special computer hardware and software configuration.

The Norsk Data Service Department also utilizes a Hardware Maintenance Notebook that covers the procedures utilized by Norsk Data in Norway.

TABLE OF CONTENTS

Section Page

1. NORD-50 COMPUTER SYSTEM ARCHITECTURE AND PHYSICAL LAYOUT

This section starts with a short description of the Nord-50 computer system architecture followed by the highlights and the electrical and physical specifications for the Nord-50 cabinet.

The rest of the section deals with schematics of the physical layout inside a Nord-50 cabinet.

To gain access to the interior of a Nord-50 the front and rear doors have to be removed. This door removing procedure is depicted in Figure 1.

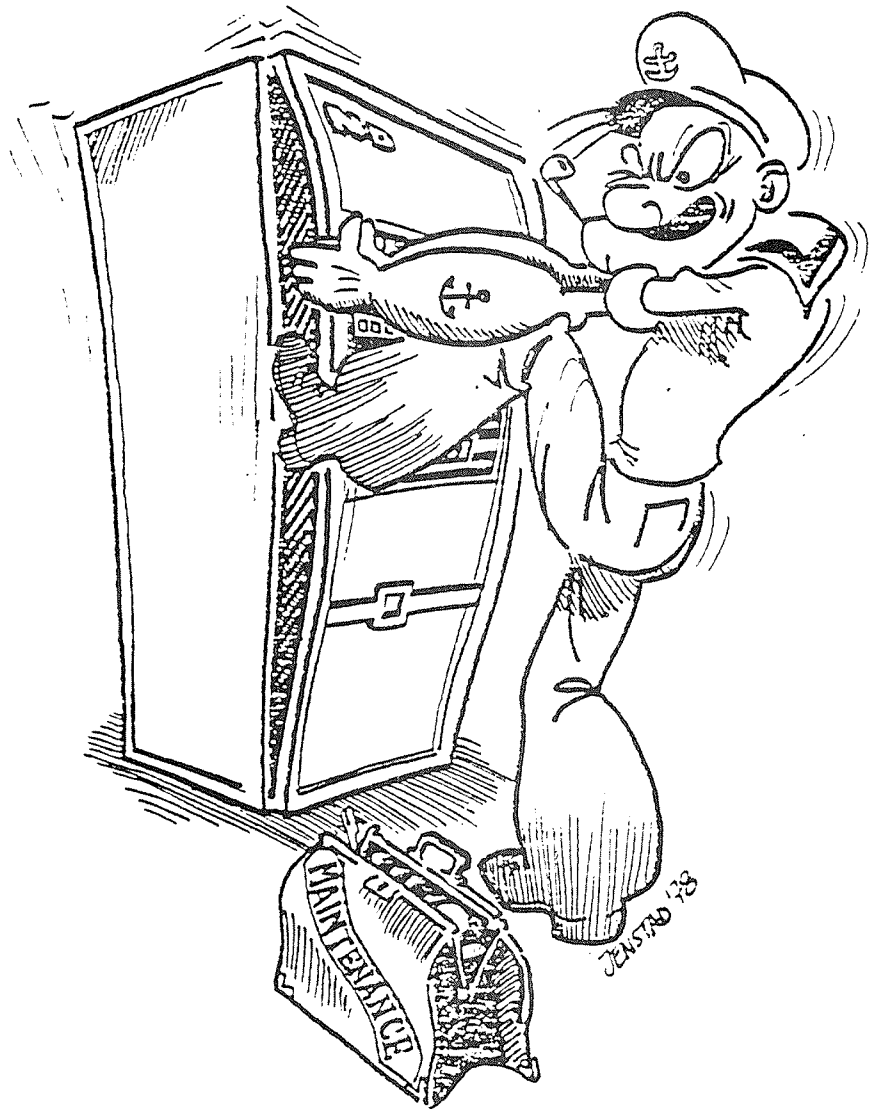


Fig. 1. Door removing procedure

In Figure 2 the physical layout of the main components is shown. The Nord-50 shown in Figure 2 has the multiport memory installed in the rear of the cabinet.

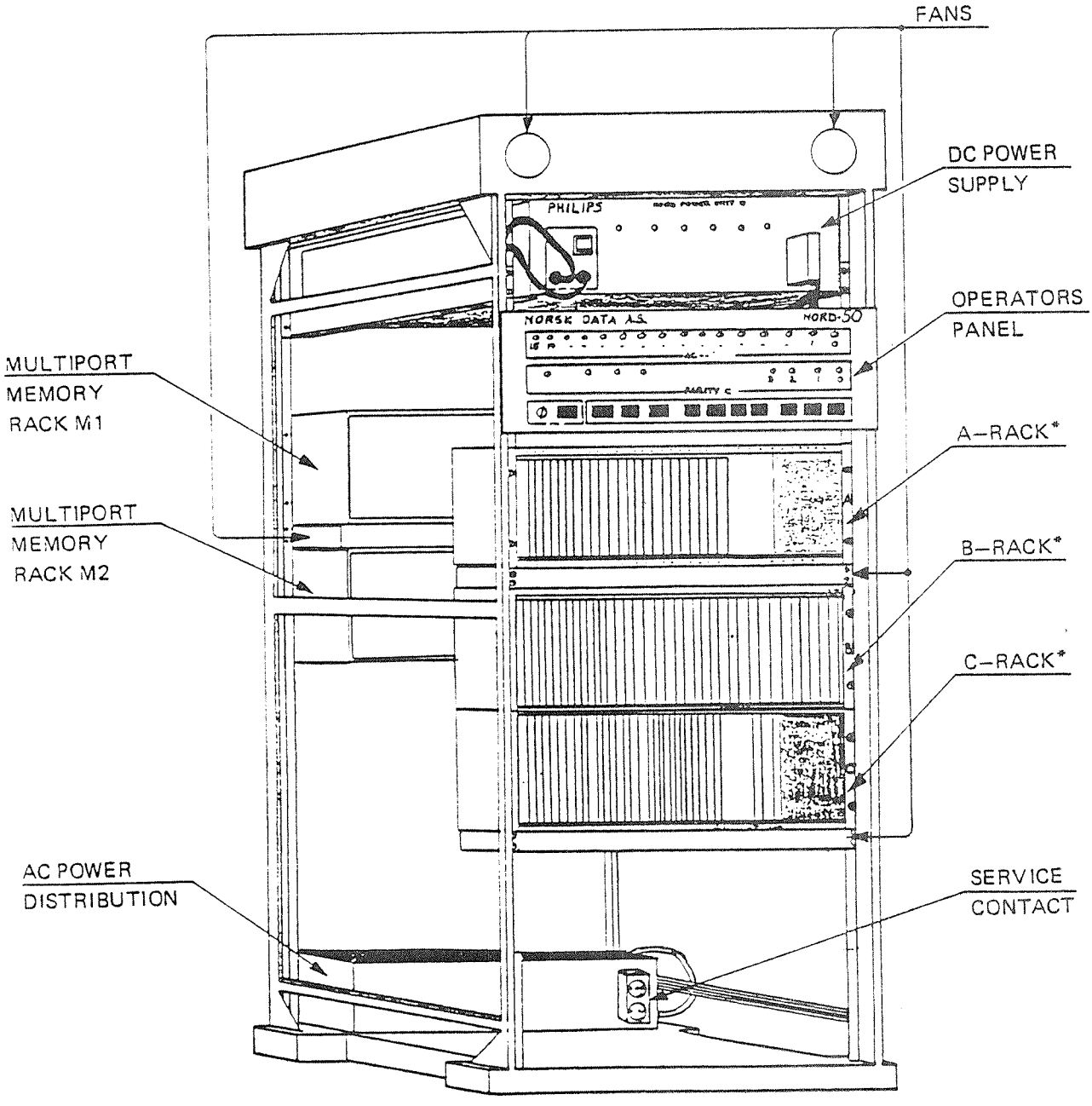


Fig. 2. NORD-50 main component physical layout

More specific information concerning the three racks of the Nord-50 is shown in Figure 3.

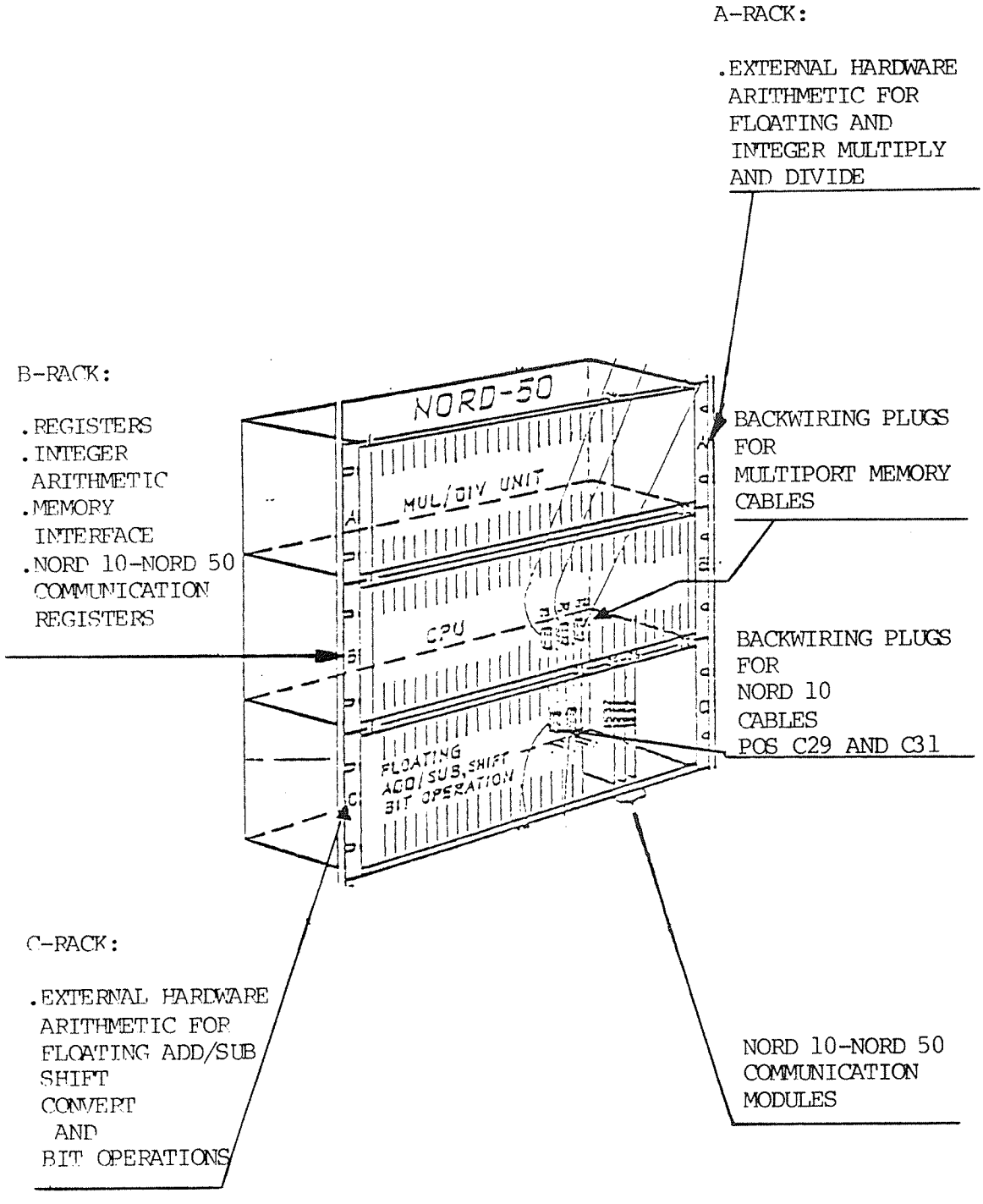


Fig. 3. NORD-50 A B AND C rack physical layout

1.1. Nord-50 Computer System Architecture

The Nord-50 Computer System consists of:

- Nord-10/S CPU
- Nord-10/S I/O System and Peripherals
- Nord-50 CPU
- Multiport Memory System

Figure 4 illustrates a Nord-10/S Computer System configuration in a Nord-50 Computer System. The Nord-10/S has the following functions:

- Acting as the I/O System for the Nord-50 by performing I/O transfer to/from the Nord-50 memory.
- Supervision and sychronization of the Nord-50. See Figure 5.
- Running the operating system, SINTRAN III.
- Generating Nord-50 executable machine code via Nord-50 Assembler or FORTRAN Complier.

To control the Nord-50, the Nord-10/S has a Nord-10/S - Nord-50 communication interface for starting and stopping the Nord-50. In addition, different control and status information is passed between the processors, such as specified break for overflow, underflow, divide by zero, memory protect error, etc. The communication interface also supports a memory examine/deposit function such that the Nord-10/S may reach all of Nord-50 memory even if this memory is larger than the normal maximum Nord-10/S addressable memory of 512K bytes.

In addition, both processors are connected to the Multiport Memory System and may both use the shared memory. Therefore, this area also provides means of communication between the processors. Each processor may also have its private memory, inaccessible to the other.

Note. With the DMA-ADDRESS-EXTENDER option installed, the disk may access the whole memory, including the Nord-50 private memory.

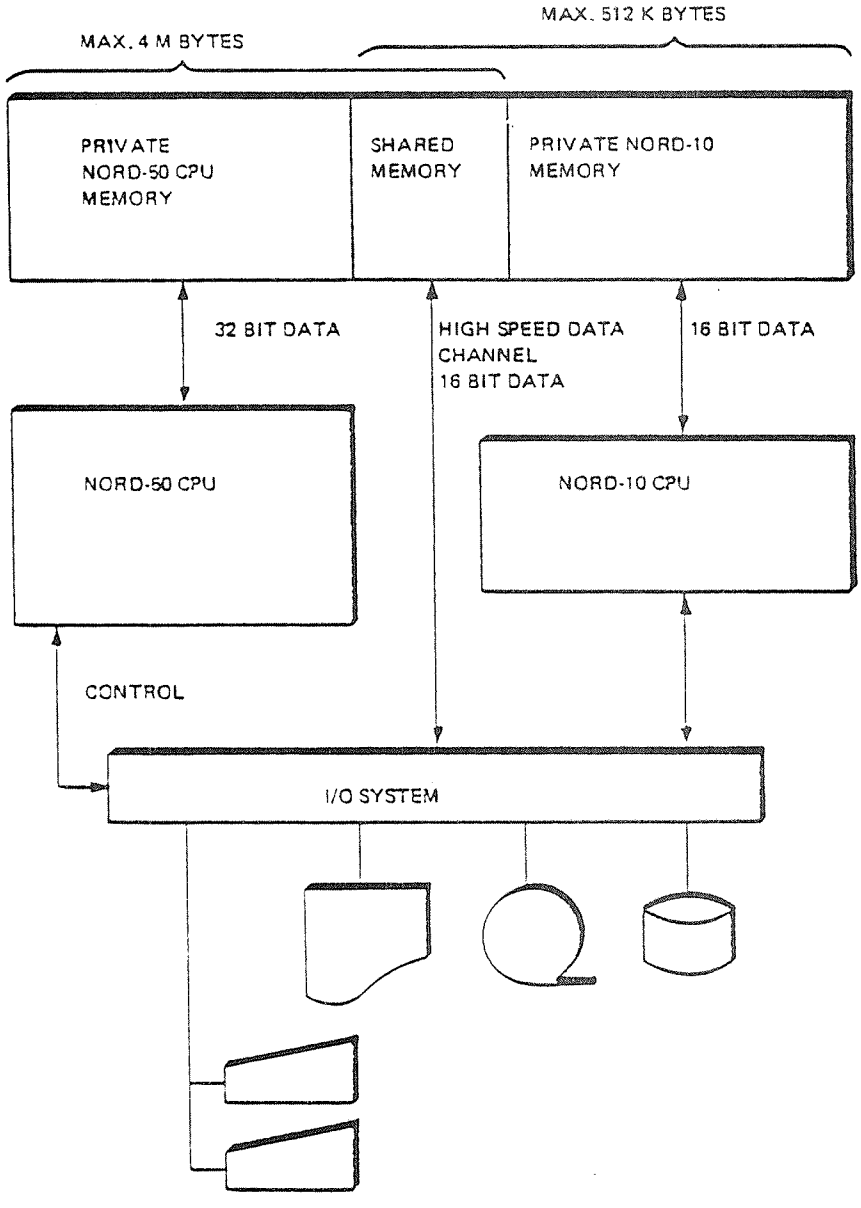
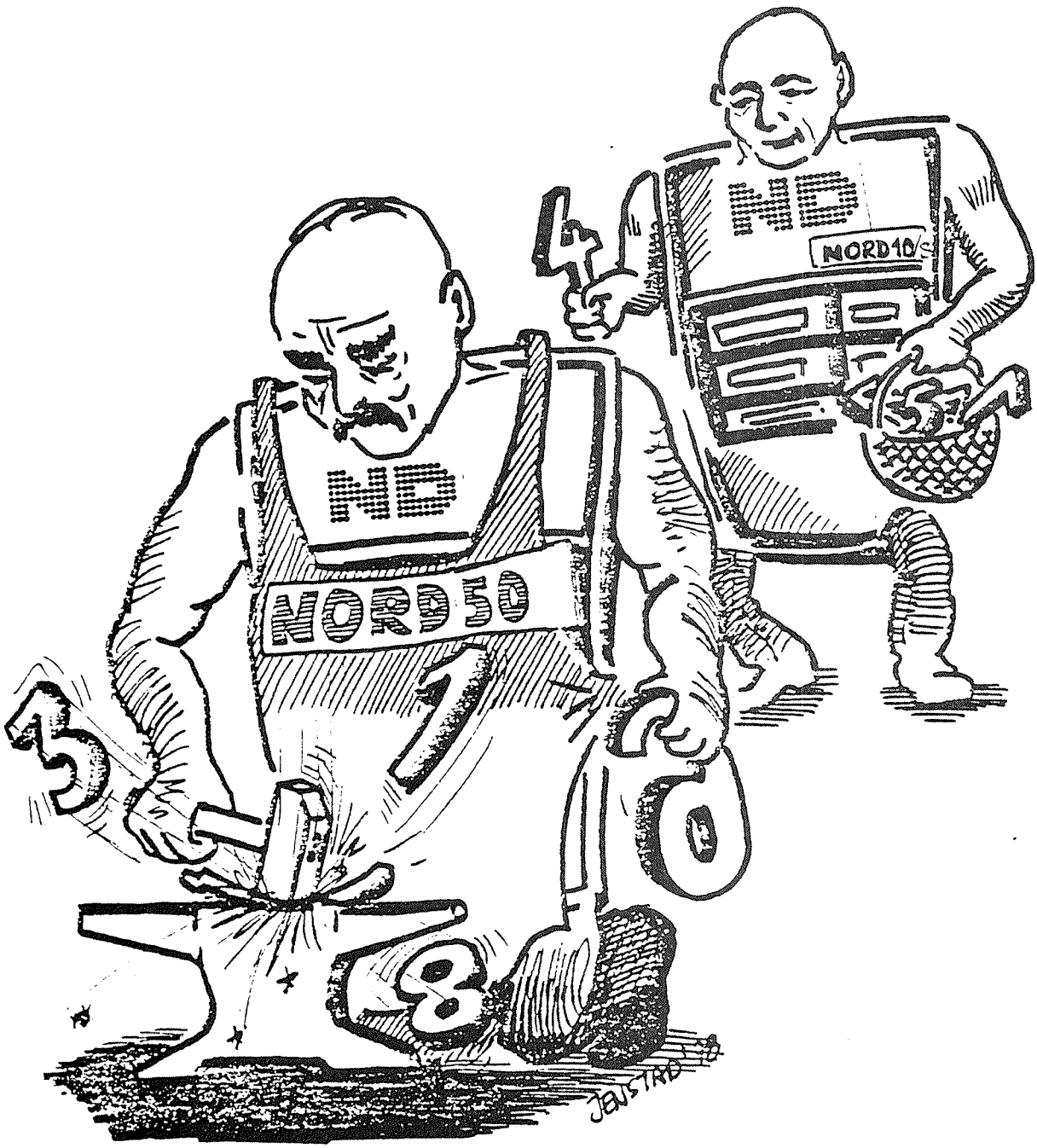


Fig. 4. NORD-50 computer system parts



THE NUMBER CRUNCHER

Fig. 5. Supervision and synchronization of the NORD-50

Some other important attributes of the Nord-50 CPU are as follows:

- 32 bit word length.
- Physical addressing range of 4Mbytes (20 address bits).
- High data precision - Floating point format of 23 or 55 bit mantissa and a 9 bit exponent plus a sign bit.
- Extensive instruction set with single and double precision floating point instructions.
- Multiported shared and private memory.
- Static high speed, private memory for frequently used code and data.
- Instruction look-ahead for increased performance.
- Specialized arithmetic unit for floating multiply/divide and integer multiply/subtract.
- Specialized arithmetic unit for shift, bit operations and floating add/subtract.
- 2 identical blocks of 64 registers, each 32 bits, to obtain increased speed in register operations.

1.2. Electrical and Physical Specifications (Nord-50 Cabinet)

Power consumption	1000W
Power requirements:	
ND 1400	220VAC+/- 10%, 50Hz+/-2Hz
ND 2400	110VAC+/- 10%, 60Hz+/-2Hz
Operating temperature	+ 10 degrees C to + 35 degrees C
Operating relative humidity	+ 10% to 90% noncondensing
Physical dimensions:	
Width	58 cm
Height	160 cm
Depth	68 cm
Weight	100 kg

2.NORD-50 HARDWARE SUMMARY

2.1.General

This section starts with some general comments about debugging the Nord-50. It then continues with an overview of the Nord-50 hardware contained in the three 19-inch racks (also referred to as Rack-A, Rack-B and Rack-C). For each of these three racks the main functions are listed followed by a data flow and a short module description.

Figure 6 shows a Nord-10 - Nord-50 Configuration.

Section:	Describes:
2.3	The B-Rack or the Nord-50 CPU rack.
2.4	The A-Rack* or the multiply/divide unit.
2.5	The C-Rack* or the floating add/ subtract and SHIFT UNIT. The C-Rack also contains the Nord-10/Nord-50 communication hardware.
2.6	The Nord-50 Operator's Panel.

*The A and C-Racks are also referred to as external arithmetic.

2.2.Nord-50 Hardware Maintenance

Of the 79 modules distributed on the A,B and C-Racks, only 27 modules are different. This makes the Nord-50 easy to troubleshoot. All modules with the same number are exchangeable.

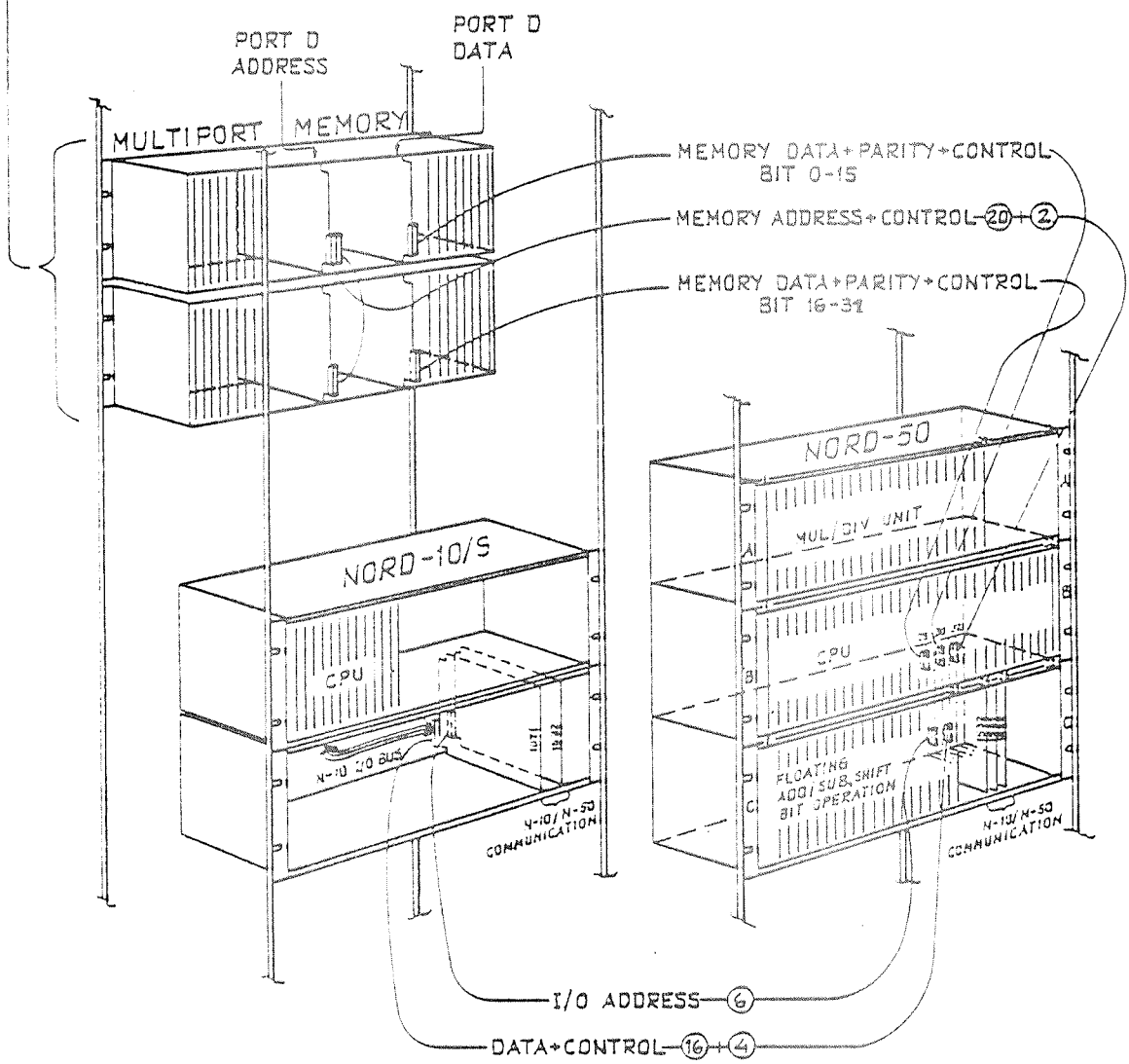
If the test programs indicate a hardware error, swap modules and run the test once more to see if the failing test pattern is changed. Because different parts of the module are used in the different postions, one module may work OK in one position but will fail in another position.

NOTE THESE:

+ THE 5V DC POWER SHOULD NOT BE TURNED OFF

DURING MODULE SUBSTITUTION.

Note.
The Multiport Memory can also be located in the NORD-50 Cabinet.



NORD-10 - NORD-50
CONFIGURATION

Fig. 6. NORD-10 - NORD-50 configuration

+ ERRORS ON INSTRUCTIONS BEING EXECUTED IN THE A-RACK (MUL/DIV) COULD BE CAUSED BY A MALFUNCTION IN THE B-RACK OR IN THE C-RACK*.

+ ERRORS ON INSTRUCTIONS BEING EXECUTED IN THE C-RACK COULD BE CAUSED BY A MALFUNCTION IN THE B-RACK OR IN THE A-RACK*.

* Sections 2.4 and 2.5 contain information about disconnecting the A-Rack and the C-Rack.

2.3.The Nord-50 CPU Rack - Rack-B

The Nord-50 CPU contains the registers, integer arithmetic, the communication registers Nord-10 - Nord-50, memory address and data line drivers/receivers and line drivers/receivers for the external arithmetic, and the C and A-Racks.

Figure 7 shows the data flow to and from external units in the Nord-50 CPU (The B-Rack).

These functions are organized on three different circuit boards, each handling four bits:

Address Arithmetic	1501
Register	1502
Arithmetic Buffer	1503

The 32 bit CPU uses eight of each boards, making a total of 24 boards.

The timing and control section of the CPU uses eight different boards:

Nord-50 I/O Control	1500
Nord-50 Controller	1504
Register Address	1505
Cycle Counter	1506
Arithmetic Control	1507
Chip Select	1508
Instrument Control	1510
Timing Control	1519

Figure 8 is the Nord-50 CPU Module Description.

Figure 9 shows a Nord-50 CPU DATA FLOW.

Figure 10 is a Nord-50 CPU Control Module Description.

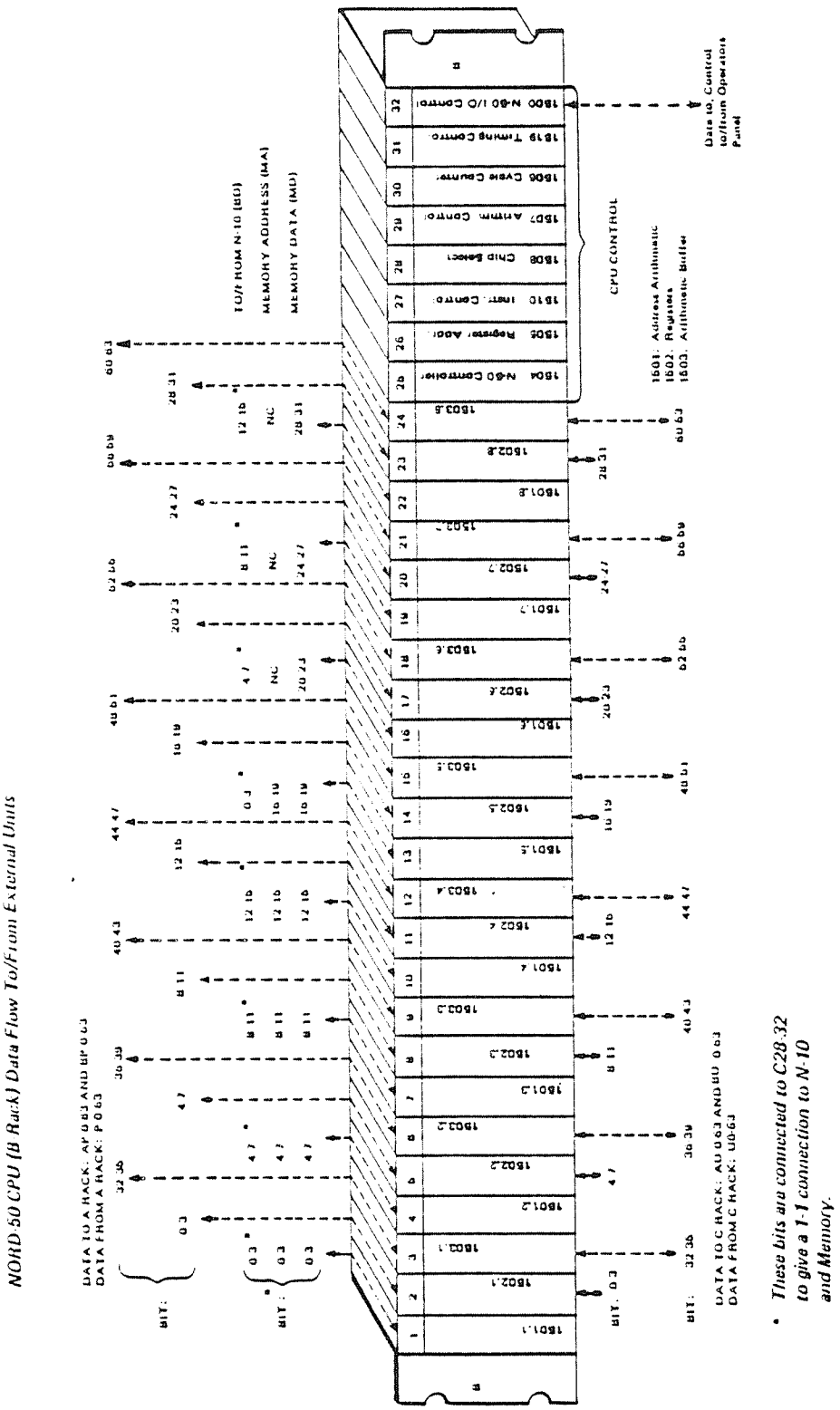
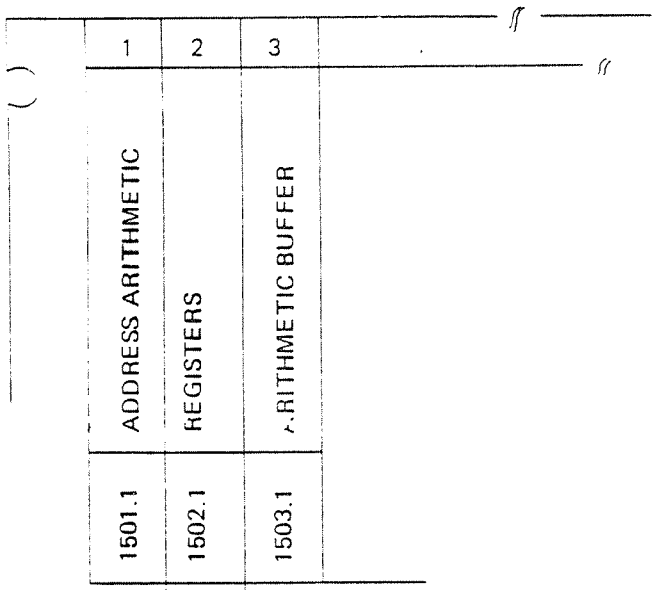


Fig. 7. NORD-50 CPU (B-rack) data flow tofrom external units



1503 = .INDEX REGISTERS (XO-15)
.BASE REGISTERS (BO-15)
.ADDRESS ARITHMETIC
.DRIVERS FOR OPERANDS TO THE EXTERNAL ARITHMETIC (Bit 63-32)
.EXTERNAL ARITHMETIC RESULT RECEIVERS

1502 = .REGISTER BLOCK A (64 REGISTERS)
.REGISTER BLOCK B (64 REGISTERS)
.INTEGER PRODUCT OVERFLOW REGISTER
.INTEGER DIVIDE REMAINDER REGISTER
.A AND B OPERAND SELECTOR/LATCH
.CPU INTEGER ALU
.ALU OUTPUT EQUAL ZERO DETECT LOGIC
.DRIVERS FOR OPERANDS TO THE EXTERNAL ARITHMETIC
.EXTERNAL ARITHMETIC RESULT RECEIVERS

1501 = .MEMORY ADDRESS DRIVERS
.MEMORY DATA DRIVERS/RECEIVERS
.DATA PARITY GENERATE/LOGIC
.PART OF ADDRESS ARITHMETIC
.NORD 10/NORD 50 COMMUNICATION REGISTERS (SA,SD,TA,TD,BP,BQ, and PC)
.ADDRESS VIOLATION COMPARE LOGIC (BP,BQ)
.INSTRUCTION REGISTER, NEXT INSTRUCTION REG AND DISPLACEMENT REGISTER

Fig. 8. NORD-50 CPU module description

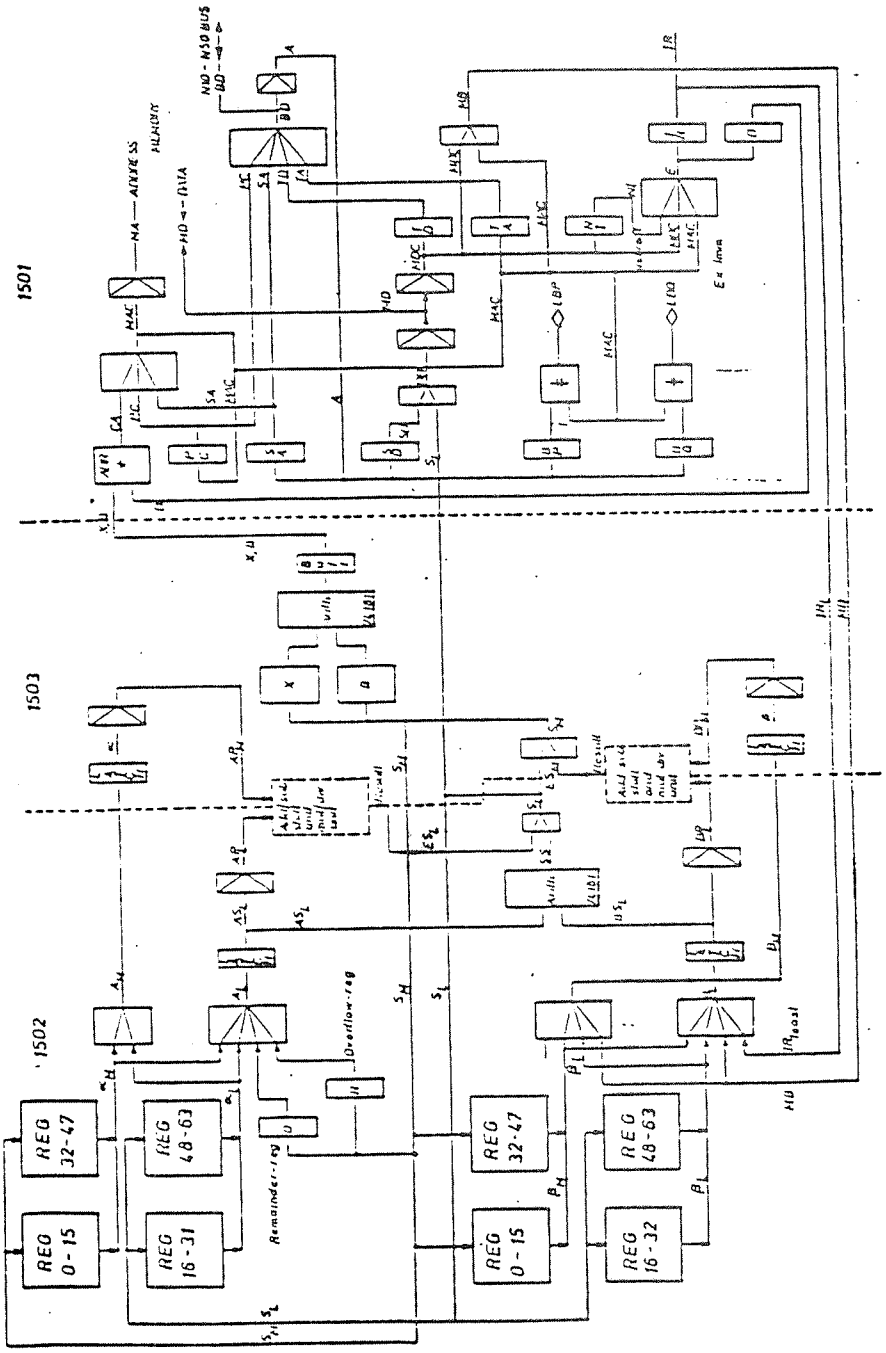
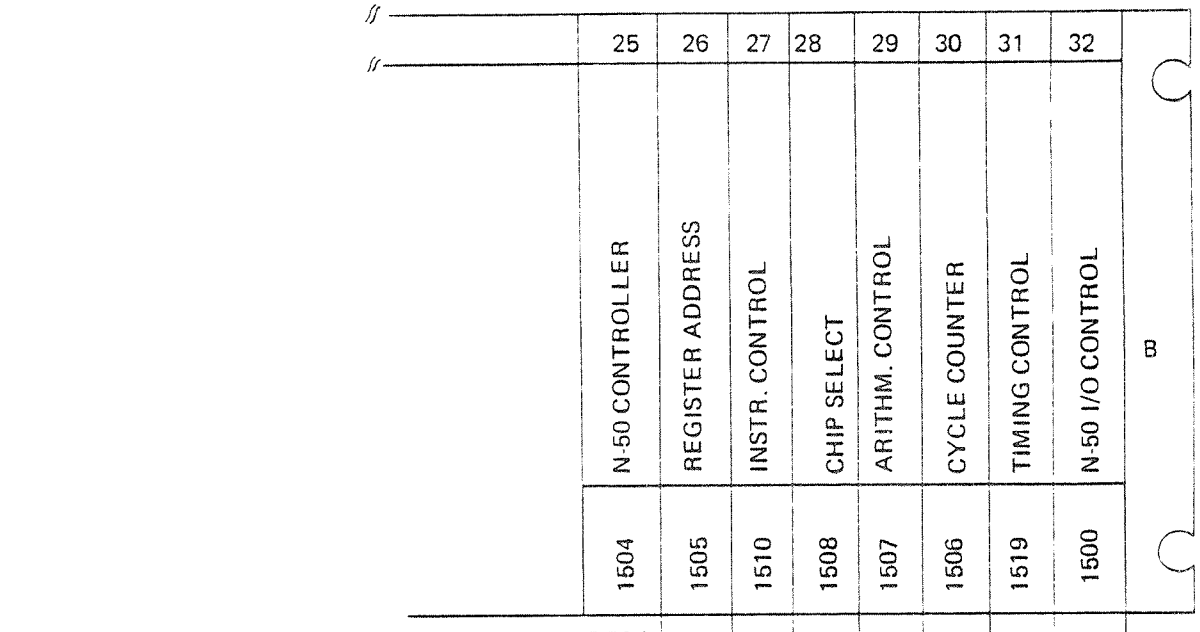


Fig. 9. NORD-50 CPU data flow



1504

.PARITY GENERATE/CHECK LOGIC .ADDR VIOLATION DETECT LOGIC .IOX INSTR. DECODE
.RECEIVERS FOR EXT. ARITH CONTROL SIGNALS

1505

.REGISTER SELECT SIGNALS (A AND B-BLOCK) INDEX AND BASE REGISTERS
.REGISTER ZERO DETECT LOGIC

1510

.INSTRUCTION DECODE LOGIC. .REGISTER WRITE SIGNALS .Next instruction and instruction reg control signals .Instruction hangup counter

1508

."EXTERNAL" INSTRUCTIONS decode logic .External data control signals . A and B-operand latch

1507

.CPU-ALU function select bit generate
.Carry flip-flop .Ext arith result bus module (63-32)

1506

.Main states flip-flops .Next instruction decision logic .Test condition decode logic
.Program counter control

1519

Cycle decode logic .Cycle read and write phase flip-flops .Request signal generate .Register write pulse
.NORD-10/S completion signal (INTERRUPT to level 12)

1500

.Operators panel control logic
.Receivers for address ready (AR) and data ready (DR) signals from memory

Fig. 10. NORD-50 CPU control module description

2.4. Nord-50 Multiply/Divide Unit (A-Rack)

2.4.1. General

-- Integer multiply/divide
and
-- Floating multiply/divide
(both single and double precision)

The A-rack can be looked upon a special processor being controlled by the Nord-50 CPU.

The processor is activated when:

-- the CPU fetches a floating integer, multiply or divide instruction for execution, and

-- the operands are latched in the CPU and presented on the A and B-operand lines to the A-rack.

For a floating double memory reference instruction, this implies that two memory references, each reading a 32 bits operand, have to take place before the processor is activated.

While the processor is active, the CPU waits until a READY signal is received indicating that the SUM-bus to CPU contains an achieved result.

The ready signal triggers a CPU - WRITE - cycle, writing the result back to the register, specified in the destination field of the Instruction Register.

The 64 bits result of a floating double instruction is written back to the register block in parallel.

An integer product of 32 bits is written back to the specified register and in parallel the 32 overflow bits is written into the overflow register.

A quotient of 32 bits, the result of an integer division, is written back to the specified register and in parallel the 32 bits remainder is written into the remainder register.

The A-Rack can easily be disconnected from the CPU (B-Rack) by pulling out all the 1522 modules (1522.1 - 1522.8). This will not influence the execution of instructions in the B-Rack and in the C-Rack.

2.4.2. Maintenance Hints for the A-Rack

- If one or two result bits in the A-Rack are failing, suspect the 1522 modules or the 1502 or 1503 modules in the CPU (B-Rack).

- The exponent arithmetic of floating instructions is handled on a special module, the 1524 module.

- Check the 5V DC power in the A-Rack.

  Note! Not less than 4.9 volts.

- Check that the fans underneath the A-Rack are rotating. If the fans are not rotating this could be the cause of sporadic errors.

Figure 11 is a diagram showing the A-Rack (Mul/Div Unit) data flow.

Table 2.1 gives module positional information for Rack-A: A Nord-50 Multiply/Divide Unit.

Figure 12 shows the Nord-50 Multiply/Divide Unit Modules.

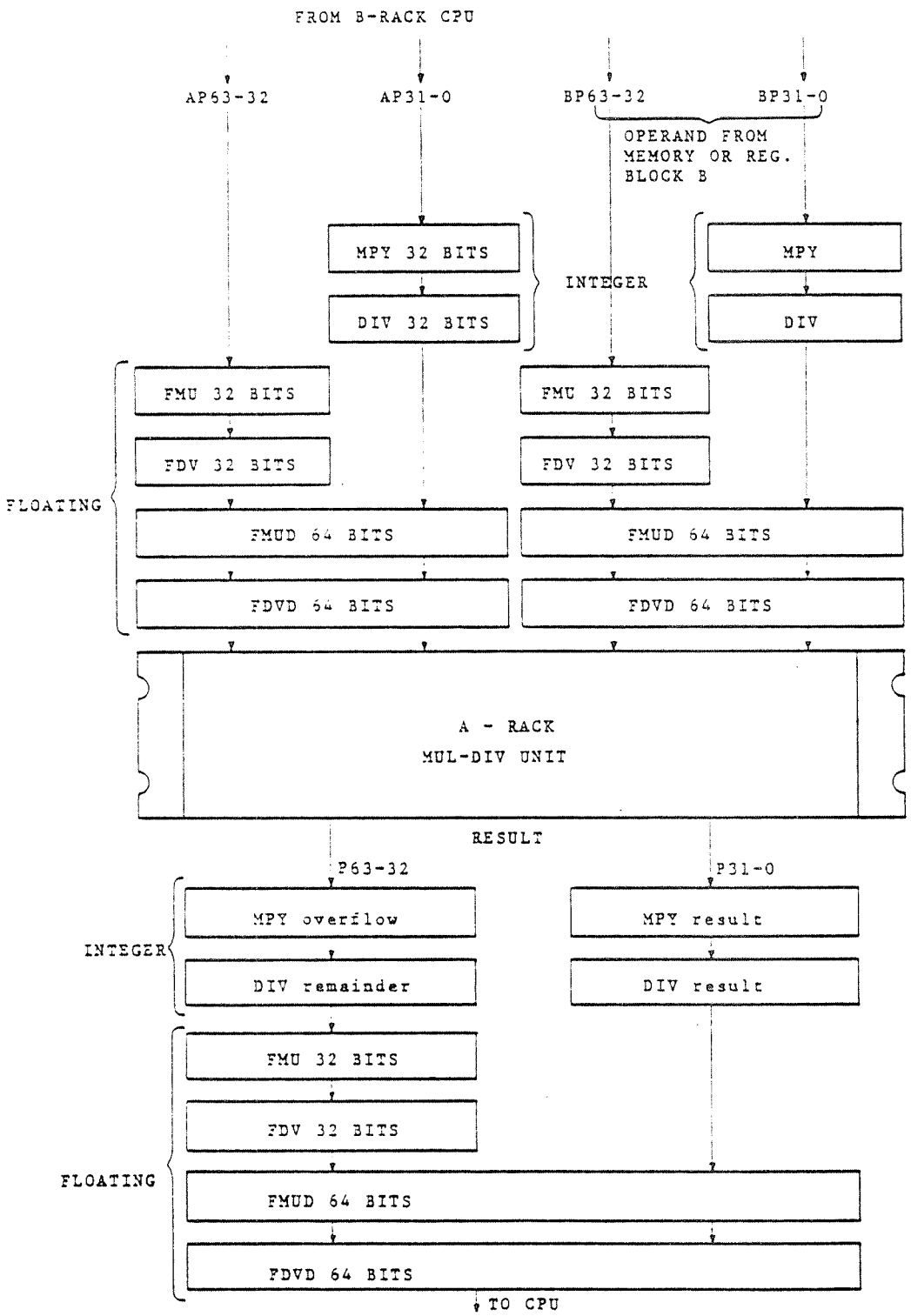


Fig. 11. A-RACK (MULDIV UNIT) data flow

Position:		
1	1521.1	MUL-DIV ARITHMETIC
2	1522.1	OUTPUT SELECTOR
3	1521.2	MUL-DIV ARITHMETIC
4	1522.2	OUTPUT SELECTOR
5	1523.4	B-INPUT
6	1521.3	MUL-DIV ARITHMETIC
7	1522.3	OUTPUT SELECTOR
8	1521.4	MUL-DIV ARITHMETIC
9	1522.4	OUTPUT SELECTOR
10	1523.2	B-INPUT
11	1511	SHIFT MATRIX A
12	1512	SHIFT MATRIX B
13	1525	MUL-DIV TIMING
14	1526	MUL-DIV CONTROL
15	1521.5	MUL-DIV ARITHMETIC
16	1522.5	OUTPUT SELECTOR
17	1523.3	B-INPUT
18	1521.6	MUL-DIV ARITHMETIC
19	1522.6	OUTPUT SELECTOR
20	1521.7	MUL-DIV ARITHMETIC
21	1522.7	OUTPUT SELECTOR
22	1523.1	B-INPUT
23	1524	EXPONENT ARITHMETIC
24	1522.8	OUTPUT SELECTOR
25		
26		
27		
28		
29		
30		
31		
32		

Note. Refer to manual Nord-50 General Description and Module Descriptions, ND-05.008.01, for module descriptions.

Table 2.1: RACK-A (MUL/DIV UNIT), POSITIONAL INFORMATION

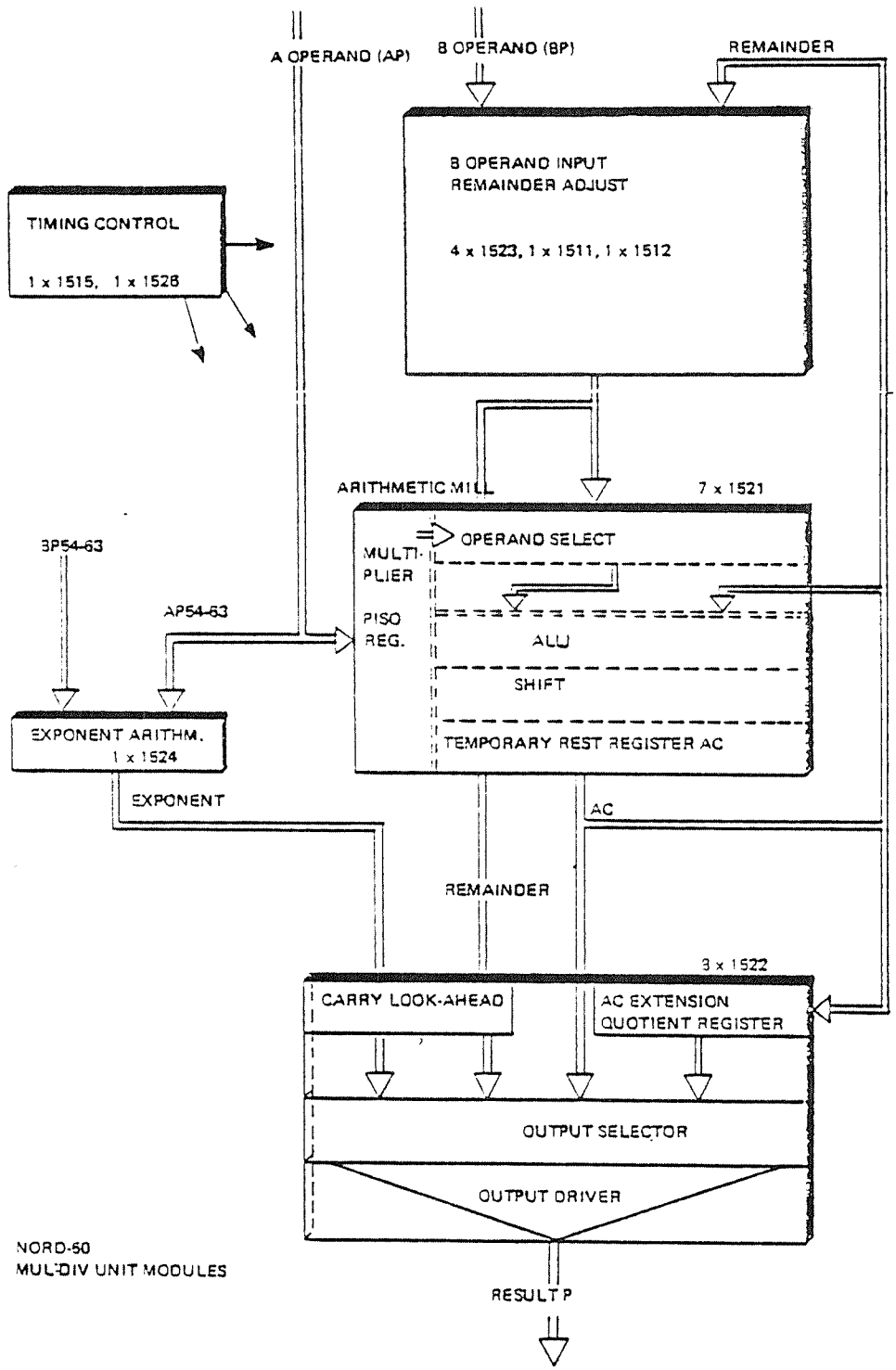


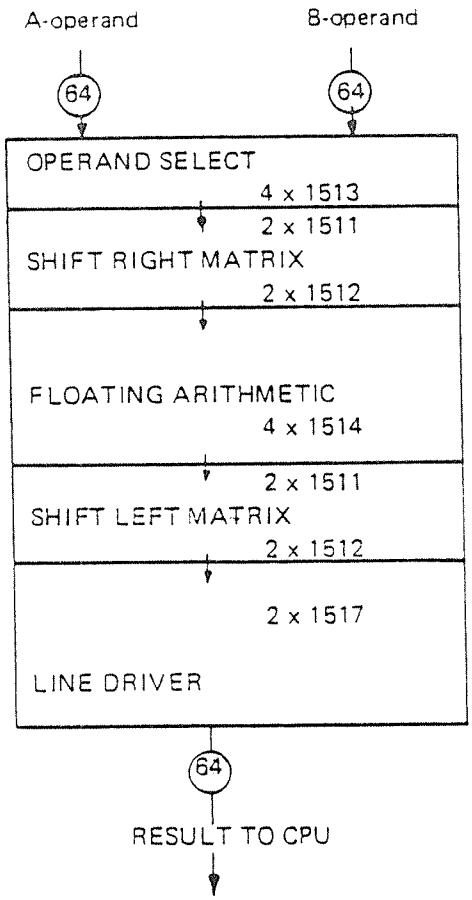
Fig. 12. NORD-50 multiply divide modules

2.5. The Nord-50 C-Rack

In the C-Rack, as part of the external arithmetic, the following instructions are executed:

Mnemonic:	Memory Ref. Inst.	Inter. Register Instr.	Means:	No.of operand bits:
FAD	X		Floating Add F+(EA)	32
FADD	X		Floating DP Add FD+ (Ea, Ea+1)	64
RAF		X	Floating Reg. Add	32
RAFD			Floating DP Reg.Add	64
FSB	X		Floating Subtract F-Ea	32
FSBD			Floating DP Subtract FD-(Ea, Ea+1)	64
FIX		X	Convert floating to Integer	32/32
FIXD		X	Convert DP floating to Integer	64/32
FIR		X	Convert floating to Rounded Integer	32/32
FIRD		X	Convert DP floating to Rounded Integer	64/32
FLO		X	Convert Integer to Floating	32/32
FLOD		X	Convert Integer to DP Floating	32/64
		X	Shift a 32 or 64 bit operand up to 63 places to the right or left with arithmetic, logical or rotational shift	32 or 64
BST		X	Bit set	32
BCL		X	Bit clear	32
BCM		X	Bit complement	32
BSZ		X	Bit skip on zero	32
BSO		X	Bit skip on one	32

As the table indicates, four of the instructions are memory reference instructions (one operand from memory and one from a register) and the rest are inter-register instructions (both operands, if two, are taken from the register block).



DATA FLOW

The C-rack consists of this main card set: OPERAND SELECT 4 x 1513.

Four 1513 cards select the operand to the internal bus.

SHIFT RIGHT MATRIX

Two 1511 cards shift the operand 0, 8, 16, 24, 32, 40, 48 or 56 places. The shift can be rotational or arithmetic.

Two cards shift the operand 0, 1, 2, 3, 4, 5, 6 or 7 places.

FLOATING ARITHMETIC 4 x 1514

The floating mantissas are added or subtracted on these cards (exponent arithmetic on card 1515.

Shift operands go directly through. For bit instructions, the specified bit should now be in 0 and is manipulated and checked on card 1516.

SHIFT LEFT MATRIX

As right shift matrix, but shifted left.

LEFT DRIVER

Drives result on tri-state lines to CPU.

Figure 13 shows the C-Rack data flow.

Table 2.2 gives module positional information for Rack-C: A Nord-50 Arithmetic Unit.

2.5.1. Control Modules

1515 SHIFT LEFT

- Right and left shift-matrix control (shift-count)
- Operand to shift-right matrix select
- Priority encoder (most significant "1" bit detect)
- Floating exponent arithmetic

1516 FLOATING CONTROL

- Instruction register bit from CPU decode logic (shift, bit and convert instructions)
- Drivers for control signals back to CPU
- DATA READY decision-logic (One Shot)

The C-Rack may easily be disconnected from the CPU by pulling out the two 1517 modules for DATA and the 1516 module for CONTROL SIGNALS. This will not influence the execution of instructions in the A-Rack and in the B-Rack.

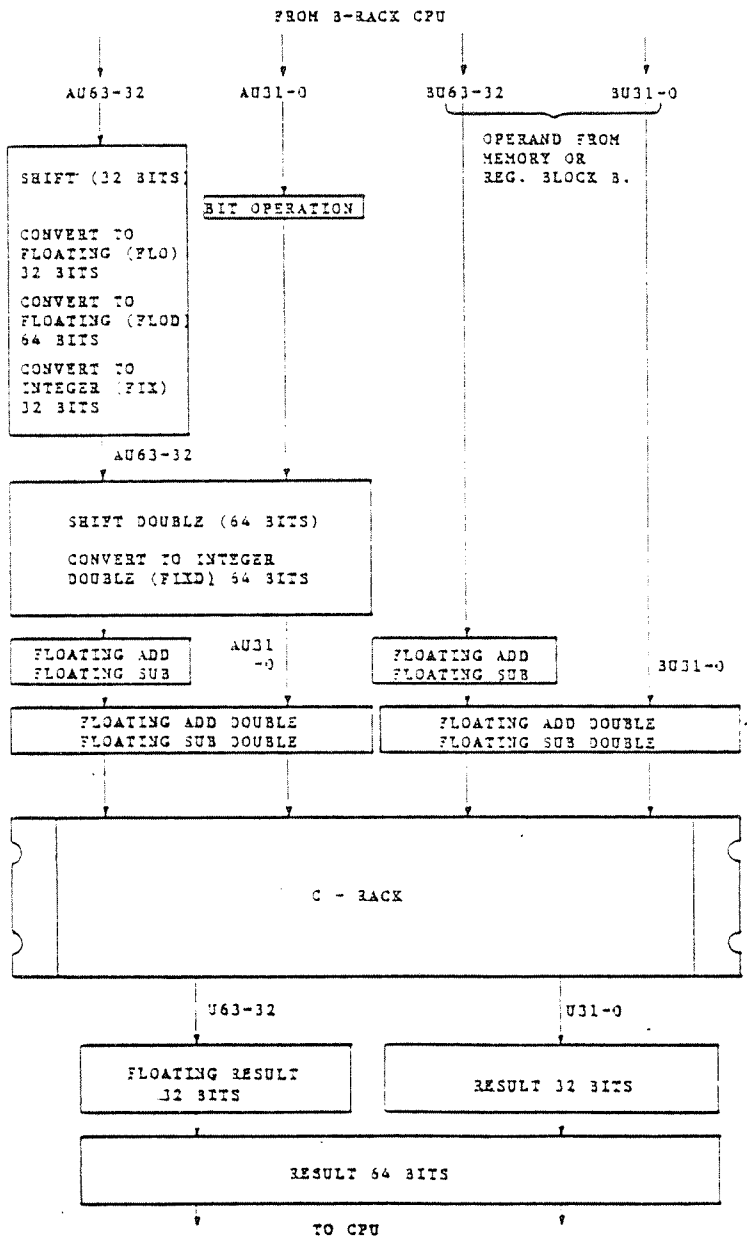


Fig. 13. C-rack data flow

1		
2	1513.1	DATA SELECTOR
3	1513.2	DATA SELECTOR
4	1513.3	DATA SELECTOR
5	1511.1	SHIFT MATRIX A
6	1511.2	SHIFT MATRIX A
7	1512.1	SHIFT MATRIX B
8	1512.2	SHIFT MATRIX B
9	1514.1	FLOATING ARITHMETIC
10	1514.2	FLOATING ARITHMETIC
11	1514.3	FLOATING ARITHMETIC
12	1514.4	FLOATING ARITHMETIC
13	1511.3	SHIFT MATRIX A
14	1511.4	SHIFT MATRIX A
15	1512.3	SHIFT MATRIX B
16	1512.4	SHIFT MATRIX B
17	1517.1**	DATA BUFFER OR 1520
18	1517.2	DATA BUFFER
19	1515	SHIFT CONTROL
20	1518*	FLOATING CONTROL
21	1513.4	DATA SELECTOR
22		
23		
24	1532	DATA & BUS SELECTOR
25	1531	DEVICE REGISTERS
26	1532	ADDRESS BUS SELECTOR
27		
28		
29		
30		
31		
32		

* According to ECO 50-50 this module should be a 1518 module and not a 1516 module.

** In the F-16 configuration, this module is replaced with a compatible 1520 module that also includes a clock.
Note. ECO 50-49 on the 1507 module in the CPU is also necessary.

Table 2.2: RACK-C (ARIT UNIT), POSITIONAL INFORMATION

2.5.2. Nord-10 - Nord-50 Communication Hardware

To control the Nord-50, the Nord-10/S has a Nord-10/S - Nord-50 communication system interface for starting and stopping the Nord-50. In addition, different control and status information is passed between the processors, such as specified break for overflow, underflow, divide by zero, memory protect error, etc. The communication interface also supports a memory examine/deposit function such that the Nord-10/S may reach all of Nord-50 memory even if this memory is larger than the normal maximum Nord-10/S addressable memory of 512K bytes.

Data and instructions may be transmitted to the Nord-50 from the Nord-10 via the communication interface simulating the Nord-50 main memory. 16 bits of data are exchanged at a time. 2 IOX instructions are then needed to transfer a 32 bit word.

The communication hardware consists of 5 modules. 3 are located in the Nord-50 (C-Rack) and 2 in the I/O Rack of the Nord-10/S. Note that 3 of the modules are identical and exchangeable (1532).

2.5.2.1. Modules in the Nord-50

1532 (POSITION C24)

- 16 Bit Differencial Line Drivers/Receivers

1531 (POSITION C25)

- Nord-50 Modus Register (M0-15)
- Nord-50 Status Register (SB0-10)

1532 (POSITION C26)

- Receivers for Nord-10/S IOX Addresses
- Receivers for Nord-10/S Control Signals (START, STOP and STROBE)

2.5.2.2. Modules in the Nord-10/S

1532

- IOX Address Bit Drivers

1071

- 16 Bit Differencial Line Drivers/Receivers
- IOX Instruction Decode
- Nord-50 Control Logic
- Nord-10 Interrupt Signal Generate

2.6. Nord-50 Operator's Panel

Figure 14 shows the Active Address portion of the Nord-50 Operator's Panel.

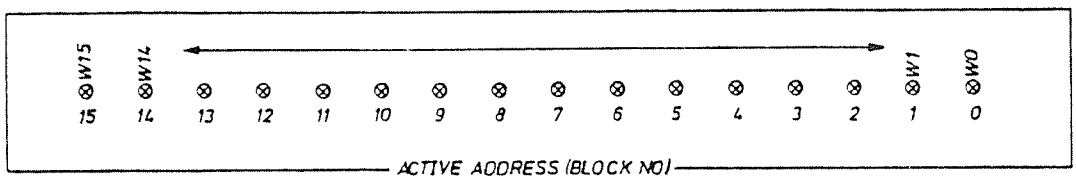


Fig. 14. Active address portion of the operators panel

The program address (PC) and data address are displayed when either DISPLAY PC (switch No. 10) or DISPLAY DATA REF (switch No. 9) is pushed.

The address is displayed in 4K blocks

Address from 0- 4K = Light in 0, etc. -- 60-64K = Light in 15

Address from 64-68K = Light in 0, etc. --

Figure 15 shows the Parity Check portion of the Nord-50 Operator's Panel.

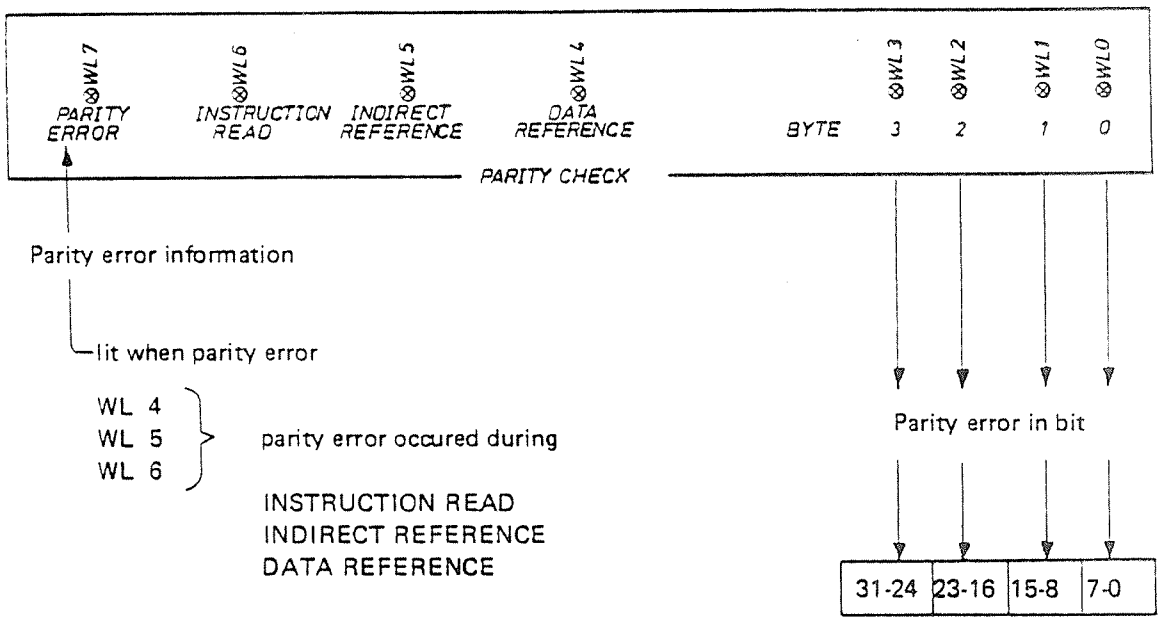


Fig. 15. Parity check portion of the operators panel

Figure 16 shows the Power Switch and the STATUS portions of the Nord-50 Operator's Panel.

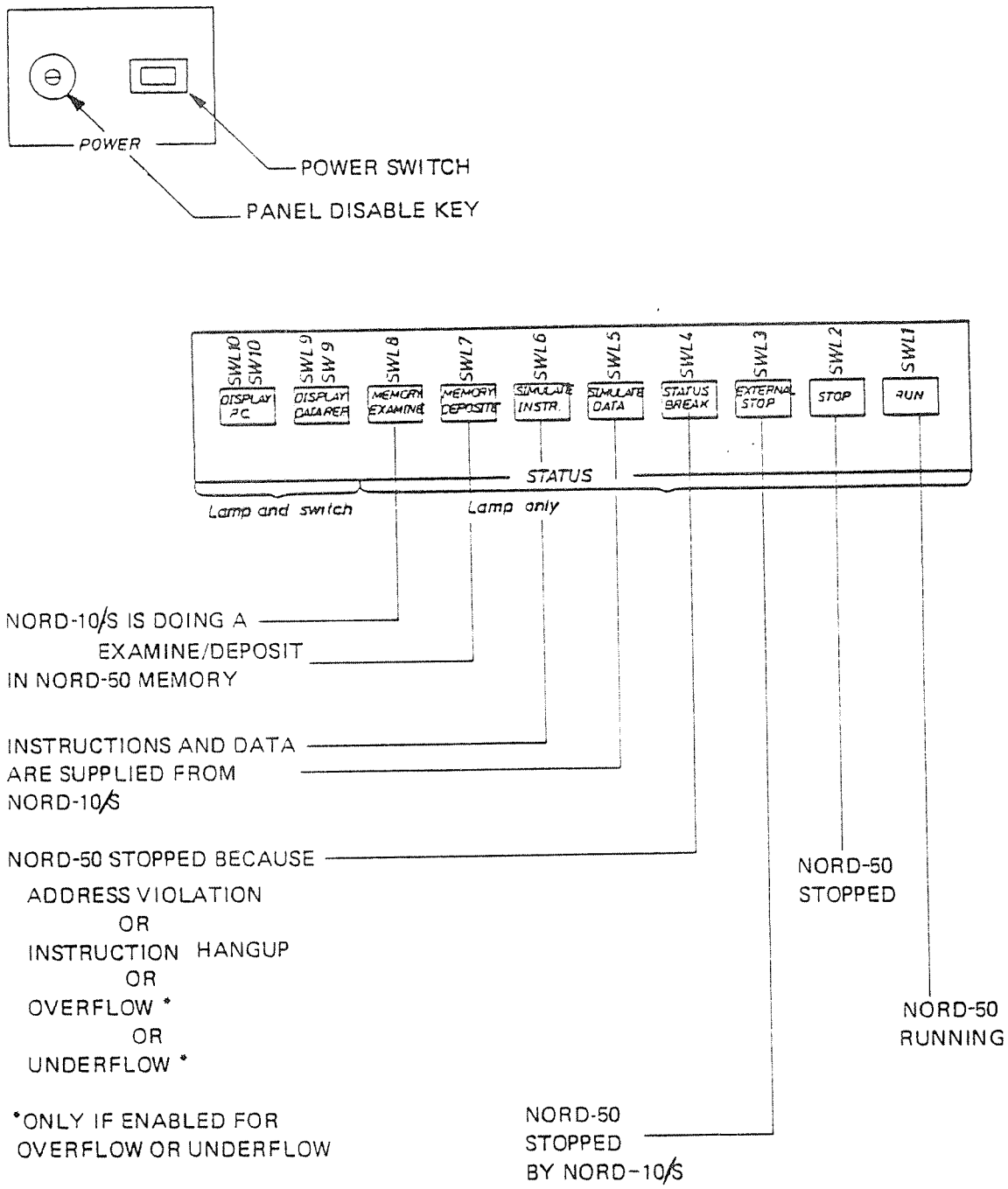


Fig. 16. Power switch and status portions of operators panel

For electrical and mechanical details, refer to the Nord-10/S technical/mechanical drawings:

10S - 48, 49 and 50.

3.NORD-50 MONITOR

3.1. Introduction

The Nord-50 MONITOR is a system program running in the Nord-10/S CPU as an extention of the SINTRAN III/VS Operating System, supervising the execution of programs in the high speed 32 bit Nord-50 CPU.

The functions of the Nord-50 MONITOR can be divided into six groups:

- Performing I/O operations and operating system functions requested from an active program.
- Loading programs into the Nord-50's memory, assigning files, starting execution and providing debugging facilities to the user.
- Reporting errors during program execution.
- Examining the Nord-50 registers/memory after the program has come to a (normal or abnormal) stop.
- System initialization and testing, assigning static, dynamic, sharable and/or local memory to be used by programs.
- The Nord-50 MONITOR also contains a simple assembler. This makes it possible to write small debugging loops in Nord-50 assembly code. Refer to Appendix E for the format.

3.2. Activation of the Nord-50 MONITOR

The Nord-50 MONITOR is activated from an interactive terminal or batch processor using the same commands:

```
@ N-50 'CR'    Note. 'CR' indicates Carriage Return.
```

In a multi Nord-50 configuration the command:

```
@ N-50 1 'CR'  will enter N-50.1
@ N-50 2 'CR'  will enter N-50.2 , etc.
```

If only the command N-50 is given, the first free Nord-50 is entered.

The commands summarized here are only those necessary for performing hardware maintenance on the Nord-50.

3.3. Nord-50 Monitor Commands

Commands for taking a program in executable format on a file and placing it in Nord-50 Memory and starting it.

PLACE<file name or file number>

Moves the executable Nord-50 program file prepared by the Nord-50 loader into the Nord-50 memory. Default file type: NOR5.

Nord-50 memory is allocated according to the break conditions set from the Nord-50 Loader when the core image file is prepared. See Nord-50 Loader and the command BREAK-CONDITIONS.

RUN

Starts execution of the Nord-50 program in the main program start address. Continues execution after a break. Continues execution after a program is stopped by "escape".

The RUN command will start with the save/unsave routines in location A. Refer to Figure 17.

GOTO<address>

Starts execution of the Nord-50 program in the specified address.

The GOTO command will start with the save/unsave routines in location A. Refer to Figure 17.

LOAD<file name or file number>

Equals the command PLACE and RUN performed in one operation.

Start in location A. Refer to Figure 17.

HARDWARE-GOTO<address>

Starts execution of a Nord-50 program in the specified address. The Nord-50 register block is not saved when this command is used and therefore the register contents cannot be examined. This command is usually used when debugging the Nord-50 hardware.

The HARDWARE-GOTO command will start in location B. Refer to Figure 17.

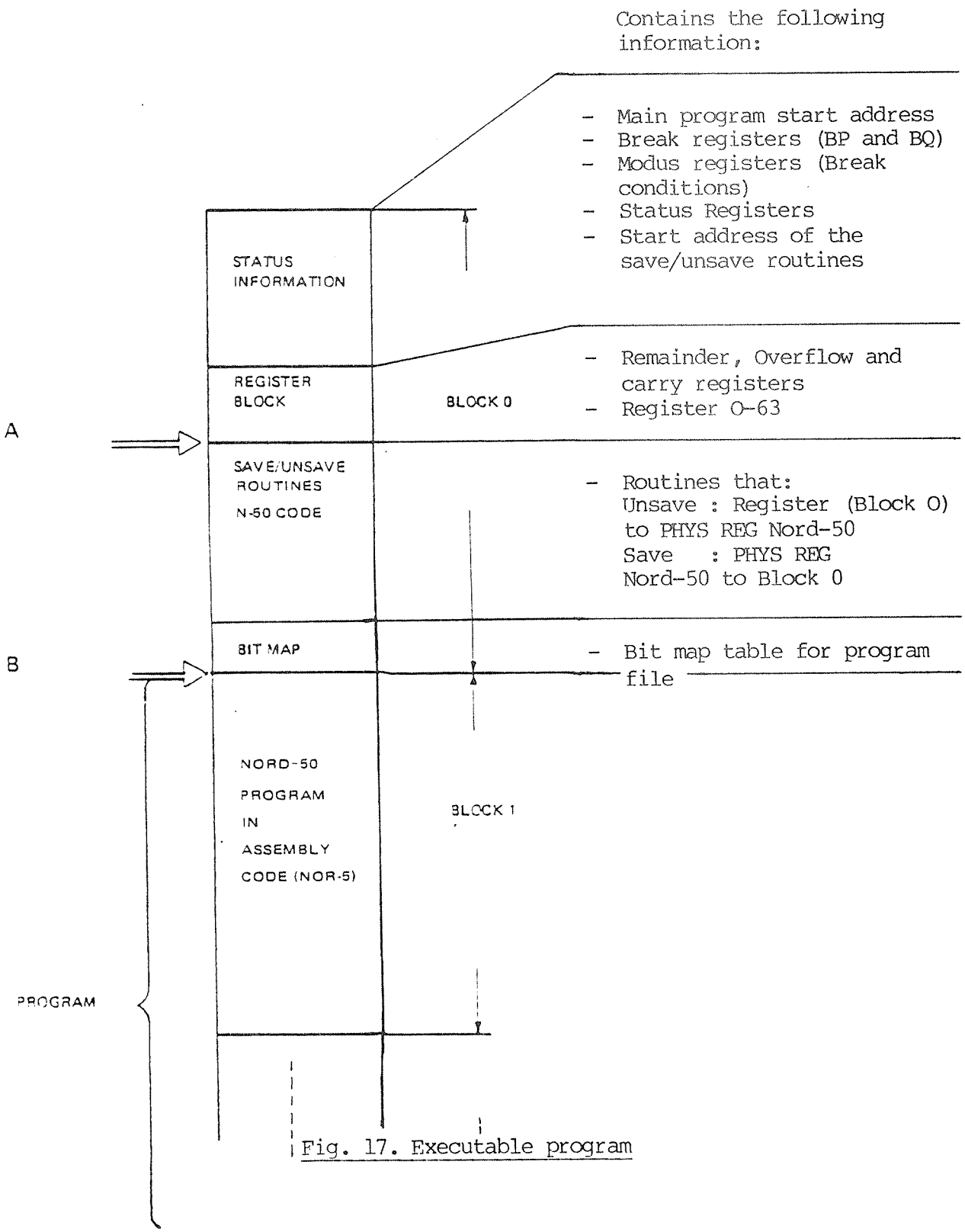


Fig. 17. Executable program

BREAK-CONDITIONS <BP address>, <BQ address>, <break conditions>

Set the Nord-50 breakpoint registers and break conditions. <Break conditions> may be A, D, F, S, O, U, P or any combination of these characters.

A stop on any reference in BP<=BQ
D stop on data reference in BP<=BQ
S stop on store reference in BP<=BQ
F stop on fetch reference in BP<=BQ
O stop on overflow
U stop on underflow
P stop on parity error in memory

The BREAK-CONDITIONS command also affects the memory allocation for Nord-50. If BP < BQ and the condition A or S (any or store reference) is on, some of the memory in address interval <BP,BQ> is free and protected from any Nord-50 write operation. Otherwise all of the Nord-50 memory is reserved for Nord-50.

MEMORY EXAMINE/DEPOSIT

The Nord-50 memory is displayed as:

<MEMORY ADDRESS>/ 'CR'	% 20/ will display the content of memory location 20 (octal).
DISPLAYED MEMORY ADDRESS	The output format may be changed with the FORMAT command. Default is: O = Octal and I = instruction disassemble.

By typing Carriage Return the next memory location is displayed.

NORD-50 MEMORY DEPOSIT:

<MEMORY ADDRESS>/ 'CR'

DISPLAYED MEMORY ADDRESS	xxxxx 'CR'	% the xxxxx is deposited into the memory address.
@ or .	% return to normal command mode.	

REGISTER EXAMINE/DEPOSIT:

R<register number>/ 'CR'	R 10/ 'CR' will display the content of register 10 (Octal).

DISPLAYED REGISTER

REGISTER DEPOSIT: The same as for memory.

Note. By typing R<register number>D 'CR' the decimal number is displayed.

FORMAT<formats>

Sets output formats to be used in the commands DUMP-ADDRESS and MEMORY or REGISTER EXAMINE DEPOSIT. Formats may be O, D, F, S, I, A, B, T, C or any combination of these characters.

O	octal
D	decimal
F	floating point (64 bits)
S	floating point (32 bits)
I	instructions (dissassembled)
A	ASCII (one word = 4 ASCII characters)
B	binary
T	two 16 bits octal numbers (Nord-50) compatible output)
C	four 8 bit octal numbers

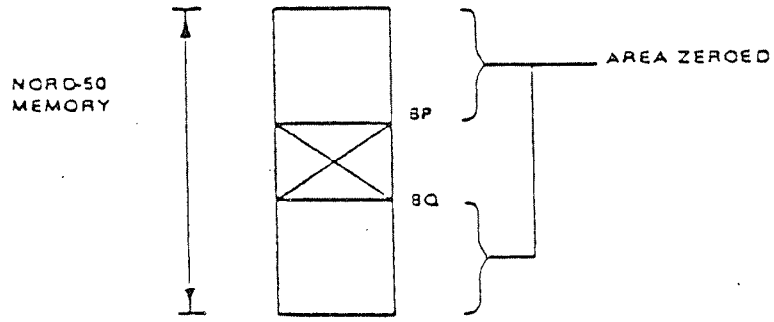
If the FORMAT command is not used formats are defaulted to OI.

STATUS

STATUS prints some information about the Nord-50 status. This command is useful when the Nord-50 stops because of any error condition. Refer to the example below.

ZERO-MEMORY

This command stores zero in the Nord-50 memory in the areas outside the BP, BQ area. If break-conditions O, U or P occur the whole memory is zeroed.



MASTER-CLEAR

Brings the Nord-50 out of any hang-up state.

HARDWARE-STATUS

Prints some information about the Nord-50 hardware status: STATUS, PC, SA, TA, TD (actually read from hardware).

```
% Status Register Format in Appendix D
% Program Counter
% Start Address Register
% Last Memory Address
% Last Memory Data
```

CARRY

Prints the "saved" Nord-50 CARRY-flip-flop.

REMAINDER (Integer Divide Remainder)

The contents of the "saved" Nord-50 remainder register is printed.

OVERFLOW (32 Most Significant Integer Product Bits)

The contents of the "saved" Nord-50 overflow register is printed.

TEST-MEMORY <address>,<number of blocks>

Tests the Nord-50 memory. The number of blocks is decimal default and block size i 1K words.

Refer to Section 6.3 for additional information.

LOOP-ON

Turn on the deposit/examine loop for the TEST-MEMORY command.

LOOP-OFF

Turns OFF the deposit/examine loop for the TEST-MEMORY command.

MEMORY-MAP <output file>

The command lists a map of the Nord-50 memory. The command is especially useful for multiple Nord-50 CPUs with shared memory. In this case a cross-reference of memory addresses as seen from all the CPUs is listed. All addresses are given in octal page numbers (1 page = 512 words). To get the Nord-50 address multiply the octal page number by 1000_8. The command can only be issued by user system.

4. NORD-50 ERROR MESSAGES

4.1. General

This section covers the error messages given by the Nord-50 MONITOR and also information about Nord-50 FORTRAN error messages. General Nord-50 MONITOR error messages are explained with appropriate illustrations.

Only error messages that could be caused by hardware failures are dwelt with.

> Note. The error messages covered are only those error messages from programs that have been tested and accepted as "error free".

Refer to the following manuals for detailed error message information for the Nord-50 LOADER and the Nord-50 FORTRAN COMPILER:

NORD-50 FORTRAN Reference Manual	ND-60.095.02	Appendix F
NORD-50 Loader Users Guide	ND-60.083.01	Appendix B

4.2. Nord-50 Monitor Error Messages

4.2.1. General

When one of the error messages listed below is printed, the Nord-50 STATUS-REGISTER should always be checked. The STATUS-REGISTER is displayed by printing STATUS 'CR'.

The STATUS command will provide the following information:

- The reason for STOP
- The program start address (SA)
- The last memory reference address (TA)
- The last data to or from memory (TD)
- The setting of the address violation break register (BP = Lower and BQ = Upper)
- The setting of the break conditions

A STATUS command and the resulting STATUS MESSAGE is illustrated in Section 3.3.

A special procedure is necessary to obtain correct information when a PARITY ERROR error message is given. This procedure is described in the PARITY ERROR error message description that follows.

4.2.2.Error Messages With "-NORD-50 ERROR: <Error Message>" Formats

The listed error message will be in the following format:

-NORD-50 ERROR: <Error Message>

The error message will be followed by an "Explanation:" and an "Action:" to be taken as appropriate.

REGISTER SAVE OR UNSAVE ROUTINE FAILED

Explanation:

The "save/unsave" routines, executed in the Nord-50 prior to the program start, are not executed in the correct sequence.

The "save/unsave" routine of 40 Nord-50 instructions is initiated when one of the following commands are given:

* LOAD <program name>
* RUN
* GOTO <address>

At program START:

The "unsave" routine is started when the initial values of all Nord-50 registers are loaded into the registers.

At program STOP:

The "save" routine is started in order to save the Nord-50 register contents in memory.

The "save/unsave" routines may be bypassed by this Nord-50 command:

* H-G <address> % Hardware GOTO the specified address.

Note. It is not possible to examine/deposit in the Nord-50 registers when the Nord-50 stops.

Action:

Either start a test program such as TINST, TREG or TMEM with the hardware GOTO command,
or

Stop the system and run the Nord-50 TEST SYSTEM. Refer to Section 5.2.1.

ILLEGAL PARAMETER POINTER IN MOINTER CALL

Explanation:

The location, after the STOP or MONITOR CALL instruction, will contain a parameter pointer or parameter address.

This error message is given if one bit from 20-31 is set (specifying an address >4 Mbytes).

A STOP instruction is probably executed as a MONITOR CALL instruction, which was not the intention. This could be caused by:

- Destroyed code
- Execute instructions in the data area

Action:

- Run CPU tests such as:
  TINST, TREG

- Run memory tests such as:
  TMEM, TEST-MEM

ILLEGAL MEMORY ADDRESS

Explanation:

Examine/deposit in the memory area that is protected by the BP and BQ registers is checked in software before being accessed and the ILLEGAL MEMORY ADDRESS error message is given.

Action:

Check the BP and BQ registers. Also check the break-conditions.

ILLEGAL PARAMETER IN MONITOR CALL

Explanation:

Wrong value of the MONITOR CALL instruction.

Action:

As for ILLEGAL PARAMETER POINTER IN MONITOR CALL error message.

ILLEGAL MONITOR CALL

Explanation:

The Monitor Call Number is >200 .

Action: As for ILLEGAL PARAMETER POINTER IN MONITOR CALL.

MEMORY OUT OF RANGE

Explanation:

No memory reply (memory time-out) when the MONITOR is performing a EXAMINE/DEPOSIT in the Nord-50 memory, or the Nord-50 is addressing nonexisting memory.

Note. Both memory banks have to respond for completing a memory access.

Action:

- Run memory test such as:
  TMEM, TEST-MEM

  Note. See Section 6.

4.2.3.Error Messages With "-<ERROR MESSAGE>-AT:<octal address>" Format

The listed error message will be in the following format:

-<ERROR MESSAGE>-AT:<octal address>

The error message will be followed by an "Explanation:" and an "Action:" to be taken as appropriate.

Error messages with this format will be given if a bit is set in the status-register and the corresponding modus bit is also set (enabled for).

ADDRESS VIOLATION

Explanation:

A Nord-50 memory access was stopped. The access did not match the break-register BP and BQ set up.

Action:

Check the STATUS information given by the STATUS-COMMAND in the MONITOR. Check the BQ and BP setting against the last memory address (TA).

If the address was illegal, the break-registers must be set once more with the B-C command.

If the address was correct, check the hardware. The address violation hardware is located on these modules:

(1501.1 - 1501.5 the 1504 and the 1519 modules)

INSTRUCTION HANG UP

Explanation:

The Nord-50 did more than 16 indirect access operations or executed more than 16 EXECUTE IMMEDIATE instructions.

Action:

- Run TINST instruction test program

  or

- Stop the system and run the Nord-50 TEST-SYSTEM

The "Hang-up" hardware is located on the 1510 and the 1519 modules.

ARITHMETIC OVERFLOW

ARITHMETIC UNDERFLOW

Explanation:

The instruction executed in the external arithmetic gave ARITHMETIC OVERFLOW or UNDERFLOW.

Note. This error message should not appear when the the corresponding bit in the modus register is not set *B-C ,,O,U Enable for Overflow and Underflow.

Action:

- Run test programs testing the external arithmetic as follows:

  HAMB MET = A and C-Racks
  FTEST = C-Rack
  DFTEST = C-Rack

  Note. The overflow/underflow signals from the external arithmetic are received on the 1504 module in the CPU.

PARITY ERROR

Explanation:

Memory parity error has been detected. With Big multiport memory system, only multiple errors will give this error message.

Action:

Check the address where the parity error was detected by giving the command:

```
*H-S 'CR'      % Hardware status.
```

The TA-register should contain the failing address.

Note. The registers are not saved during Parity Error.

The Operator's Panel should give information about the failing byte, but the panel information can be destroyed by the software.

If the parity error was in the Nord-50 local memory and this memory has been idle since last power ON/OFF sequence, do either of the following:

* A ZERO-MEMORY Command
  or
* A TEST-MEMORY Command

If not in the Nord-50 private memory, run a Memory test program. Refer to Section 6.3.

Check the memory system, the 1501 modules, or the following CPU control modules:

1504
1519

MEMORY OUT OF RANGE

Explanation:

No reply from one or both multiport memory racks.

Action:

Check the last memory address contained in the TA-Register by giving the STATUS-Command.

- Run memory tests such as:
  TMEM, TEST-MEM

and specify the lower and upper address around the non-responding address. If the error is in shared memory, also run a Nord-10/S memory test program, for example, MULTI.

EXTERNAL STOP

Explanation:

This error message is given when the Nord-50 is stopped by a Nord-10. It appears when ESCAPE is printed on the terminal.

Action:

If the Nord-10 did not intend to stop the Nord-50, then:

- Stop the system and run:
  TEST-SYSTEM

  Note. The first tests will test the Start/Stop sequence of the Nord-50.

- Check the Nord-10 - Nord-50 communication hardware, the 1504 or 1519 modules in the CPU.

4.2.4.Error Message With "*** Nord 50 IS RUNNING" Format

The error message listed below will have this format:

*** NORD 50 IS RUNNING

Explanation:

The Nord-50 was started while in CONTINUE. Bit 3 in the Nord-50 Interface Status Register was ZERO.

Action:

- Give the command:

  * M-C % Master Clear.
    or
  * STOP

If the Nord-50 is still running, do the following:

- Stop the system and run:
  TEST-SYSTEM

  Note. The first tests will test the Start/Stop sequence on the Nord-50.

4.2.5. Error Messages Caused by the SET-MEMORY Command

The following error messages are caused by the SET-MEMORY command in the Nord-50 MONITOR.

ERROR MESSAGE A:

MEMORY IS OUT OF RANGE ON THE UPPER ALLOCATED ADDRESS

ERROR MESSAGE B:

MAX MEMORY ADDRESS NOT IN ALLOCATED AREA.

Explanation:

This command will specify for the Nord-50 memory the segments that make up the various parts of the memory.

After this command is specified, the Nord-50 MONITOR checks the Nord-50 physical memory to see if the memory is according to the parameters in the SET-MEM command.

The last address, for example AA, is accessed. This address should reply. Then the last address +1, for example AB, is accessed. This address should only give the reply "memory out of range".

- If address AA gives no reply, ERROR MESSAGE A is given.
- If address AB replies, ERROR MESSAGE B is given.

Action:

- Give the command:
  *LIST-MEM

- Check to see if the size of the memory given by software matches the size of the physical memory.

  Note. The total sum of the size of the segments x 512 = the number of Nord-50 words in shared memory.

If the size specified in software looks equal to the physical memory and ERROR MESSAGE A is given, check the memory.

If ERROR MESSAGE B is given, check the CPU (1501, 1519, 1500 modules) and the memory.

The SET-MEM and LIST-MEM commands are specified in Section 7.

4.2.6. FORTRAN RUN-TIME Error Messages

Some examples of FORTRAN RUN-TIME error messages are as indicated below:

```
*INTEGER OVERFLOW ON INPUT
*BAD CHARACTER ON INPUT
*TOO LARGE ARG.IN
  -
  -
  - etc.
```

Explanation:

The Nord-50 is in a state such that it is not advisable to run FORTRAN programs.

Action:

- Run assembly programs such as:
  TINST, TREG, TMEM
  in order to test the Nord-50.

  or

- Stop the system and run:
  TEST-SYSTEM

5. NORD-50 TEST PROGRAM INFORMATION

This section contains the following test program information for testing the CPU, the external arithmetic, and the communication system:

- Test Program Groups
- Accessing Nord-50 test programs
- FORTRAN programs as test programs
- Test program overview
- The Nord-50 test system
- Nord-50 test program descriptions
- The verification programs

Appendix B contains a step-by-step procedure for the Nord-50 test sequence.

5.1. Nord-50 Test Programs

The general information about Nord-50 test programs includes information about the test program groups, the N50-TEST "user", FORTRAN programs, an overview of test programs and the Nord-50 test system.

5.1.1. Test Program Groups

The test programs for the Nord-50 are divided into two groups:

Group I - Test Programs Running in the Nord-10/S

Test programs in this category are the Nord-50 TEST-SYSTEM and MEMORY TEST PROGRAMS testing the shared memory (MULTI, MOVER, 32K MOS).

Group II - Test Programs Running in the Nord-50

All test programs running in the Nord-50 have to be loaded and started by the Nord-50 MONITOR. These programs are normally available on the disk in a Nord-50 executable format (:NOR5). The Nord-50 assembly programs have been assembled and the FORTRAN programs have been compiled and loaded on the disk by the Nord-50 LOADER during system integration at the factory.

5.1.2. N50-TEST N50-USER

A special "user" has been created and designated N50-TEST, N50-USER. This "user" is often referred to as just TEST.

All Nord-50 test programs running in the Nord-50 are generated under this "user". The "user" normally has no password.

5.1.3. FORTRAN Programs as Test Programs

The following test programs are FORTRAN test programs:

HAMB-MET, ITEST, FTEST, and DFTEST

When one of these programs is loaded with the Nord-50 MONITOR, 16K of Formatted Input/Output (FIO) routines are also loaded into the Nord-50 memory.

These FIO routines are started when the FORTRAN programs require input or output from Nord-10/S I/O devices.

The programs listed above communicate with the Console Terminal.

The FIO routines execute Nord-50 instructions for:

- Converting the ASCII code to an internal representation
- Checking the input format
- Checking the number of characters
- "STORE" the input data to the program area. The program will test against this location.

The Nord-50 hardware has to be in working condition for exchanging these FIO routines. If the hardware is not in working condition an error message from the FIO system will appear on the terminal and the program will not be started. In event the program will not start it is best to either run assembly test programs or to run the Nord-50 TEST-SYSTEM.

These FORTRAN programs serve the function of verification programs as opposed to test programs.

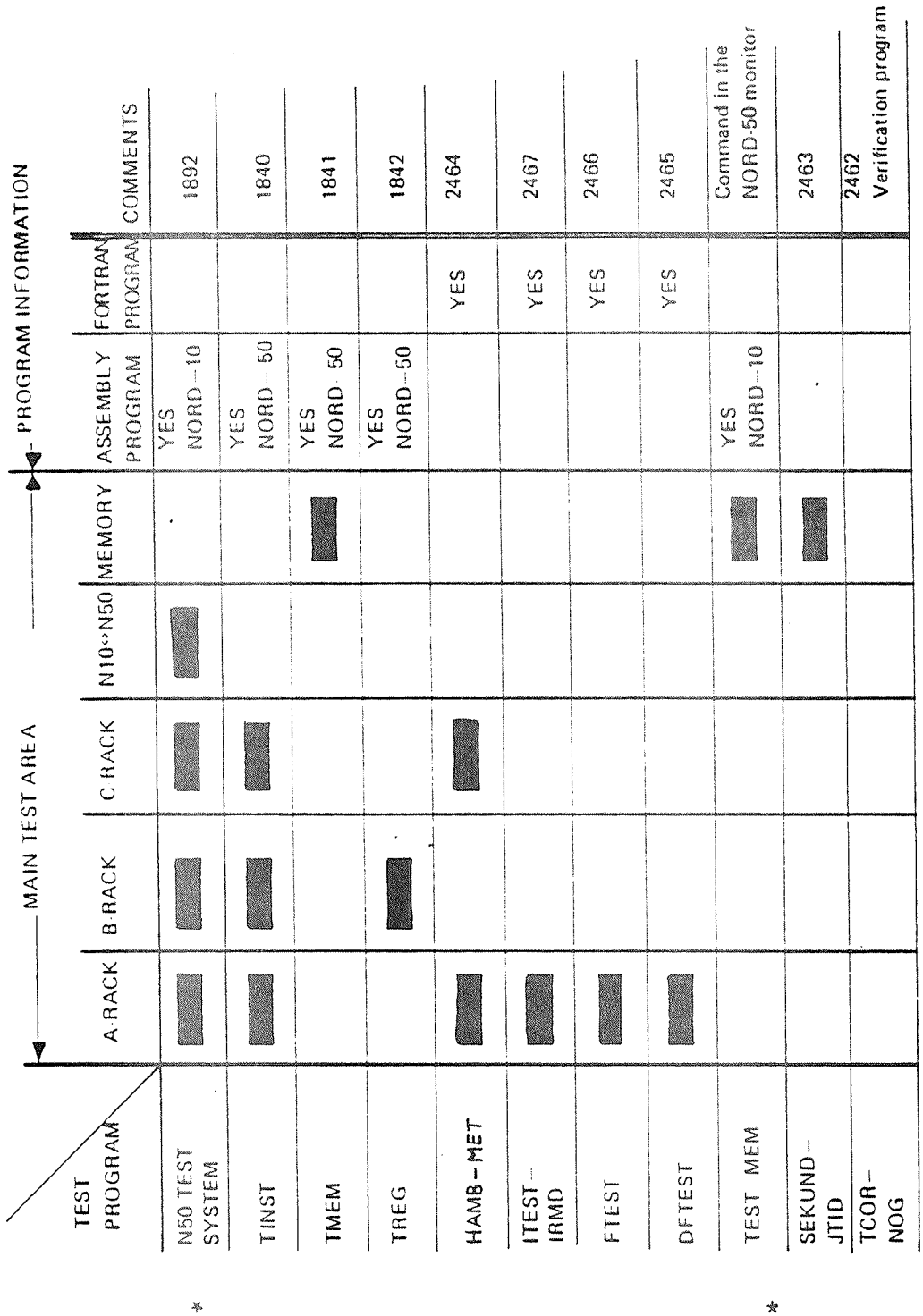
5.1.4. Module Substitution

Modules (also called cards) are exchangeable during test phases. This module exchange technique is very useful in determining whether or not an error will "follow" the module, thus assisting in fault location. It is important to realize that the TEST SYSTEM can not in fact be destroyed by exchanging or by pulling the modules in and out of the Nord-50 since the test program is actually located in the other computer (normally a Nord-10/S).

Errors or faults are easily isolated by using the module substitution technique. The test is repeated with the failing bit in different combinations so that it is traced to its module.

5.1.5. Test Program Overview

Figure 18 contains an overview of the Nord-50 test programs. In using this data it should be kept in mind that N50 TEST SYSTEM and TEST-MEM are run in the Nord-10/S. The remainder of the programs are run in the Nord-50.



* Note. The N50 TEST-SYSTEM and TEST-MEM run in the Nord-10/S.
The remainder of the programs run in the Nord-50.

Fig. 18. NORD-50 test program overview

5.2. Nord-50 Test Program Descriptions

The following Nord-50 test programs are described:

- N50-T-SYS-1892
- TINST-50-1840 % Instruction Test
- TMEM-50-1841 % Memory Test
  Note. Described in Section 6.
- TREG-50-1842 % Register Test
- SEKUND-2463
- JTID-2468
- ITEST-2467 % Integer Multiply/Divide Test
- IRMD-2469
- FTEST-2466 % Floating Test
- DFTEST-2465 % Double Precision Test
- HAMB-MET-2464 % Floating Test
- TCOR-NGO-2462
- TCOR-DATA-2470 % Data file
- N50INITIALIZE-2471 % Example of how to initialize the Nord-50
- N50TEST-BATCH-2472 % Mode file to run all test programs in sequence
- N50TESTP-ASCOL-2473 % Mode file to assemble, compile and load all programs

5.2.1. NORD-50 TEST SYSTEM

NORD-50 TEST SYSTEM HAR-1892

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A stand-alone Hardware Test Program to test Nord-50 communication, memory and instructions with simulated memory.

- Function(s) - Tests all instructions.

- Highlights - Instructions and data are exchanged via the Nord-10/S - Nord/50 communication lines.

  - The result of the test is read back to the Nord-10/S and compared with an "Expect Table" located in the memory of the Nord-10/S.

  - Error reports are given on the Nord-10/S Console Terminal.

  - Up to 16384 combinations of data may be used to test one instruction.

  - The number of data combinations for testing each instruction can be specified. The argument in the AAX instruction found in location 1307 will specify the number of data combinations.

    :1307/AAX<increment>

    The argument in the instruction is used as an index in the test data area. Table 5.1 indicates the number of minutes required to run a complete test for a given number of combinations in Location 1307.

  - The B-register of the Nord-10/S is displaying the test being executed.

LOCATION 1307: .	NUMBER OF DATA COMBINATIONS	NUMBER OF MINUTES TO RUN A COMPLETE TEST:
173402	16384	15.
173410	4096	4.15
173420	2048	2.5
173500	512	1.10
173577	256	0.55

Table 5.1: TEST TIMES FOR DATA COMBINATIONS - LOCATION 1307

- Implementation Instructions - Refer to the NORD-50 Test System Manual, ND-62.008.01. Also see PD (Program Description) Sheet in the NORD SOFTWARE LIBRARY.

  - LOAD and RUN Procedure -

    1. Stop the Nord-10/S.
    2. Insert the Floppy Diskette with Directory 10112 into the Floppy-Drive. (This is only for a single Nord-50.)
    3. Type 1560& on Terminal No. 1 (Console Terminal).
    4. Write LO N50* on the terminal.

    * In Multi Nord-50 configurations, Write:

      LO N51 % For the Test System to Nord-50.1.
      LO N52 % " " " " " Nord-50.2.
      LO N53 % " " " " " Nord-50.3.
      LO N54 % " " " " " Nord-50.4.

  - Restart address = 1000.

    Type: 1000!

Notes:

- 1. Except for the tests: 16, 17, and 50 the Nord-50 memory is not used.

- 2. Temporary errors will not be detected. The execution time in simulated memory is 10 to 15 times longer than normal execution time. The program was designed to detect "hard" errors.

- 3. The TEST SYSTEM responds to the following single letter commands:

  D Display further information about the current error; failing instruction with mnemonic and code.

  C Continue; continues the test with the next input data.

  B Back; restarts the current test with data taken from the beginning of the input data table.

  N Next test; sometimes the same test is started but with a different type of test data.

  P Previous test.

  R Reset; restarts from the first program.

- 4. When an error message is printed on the terminal always print D. A test number is then given followed by failing instruction and code.

- 5. Do not use the test number in the heading of the error message.

- 6. Notes about the error print-out:

  - The test system will print the A-Operand and the B-Operand even if single operand instructions are failing. For INST:SHIFT and bit instructions, only the A-Operand is used.

  - The 32 bit overflow after an integer multiply and the 32 bit remainder after an integer divide will be printed as the least result.

Additional Information:

- 1. Additional information about the Failing Test can be found in:

  Nord-50 Test Programs
  ND-30.007

- 2. The main test areas of the different tests are listed in Table 5.2.

- 3. The Test System accepts only even parity input. This patch is available to accept odd parity input*

OLD:	NEW:
1122/---	44011
1131/---	175235
	124376
	044004
	124001

- 4. The sequence in which instructions are executed in the Test System is shown in Figure 19.

TEST NUMBER (TEST NUMBER APPEARING AFTER D)	TEST AREA:
TO	Nord-10/S - Nord-50 Communication
T16-T17-T50*	Nord-50 Memory EXAMINE-DEPOSIT via Nord-10/S
T 100 - 113 T 144 - 145 T 170 - 171 T 174 - 175 T 215 - 220 T 223 - 224 T 227	C-Rack and Receivers/Transceivers in the B-Rack
T 164 - 167 T 172 - 173 T 176 - 177 T 221 - 222 T 225 - 226	A-Rack and Receiver/Transceivers in the B-Rack
ALL OTHER	B-Rack

* The address area to be tested can be specified in these two memory locations:

6412/--- % Lower Limit
6416/--- % Upper Limit

Only the area between 0-64K can be tested with the test system.

Table 5.2: MAIN TEST AREAS

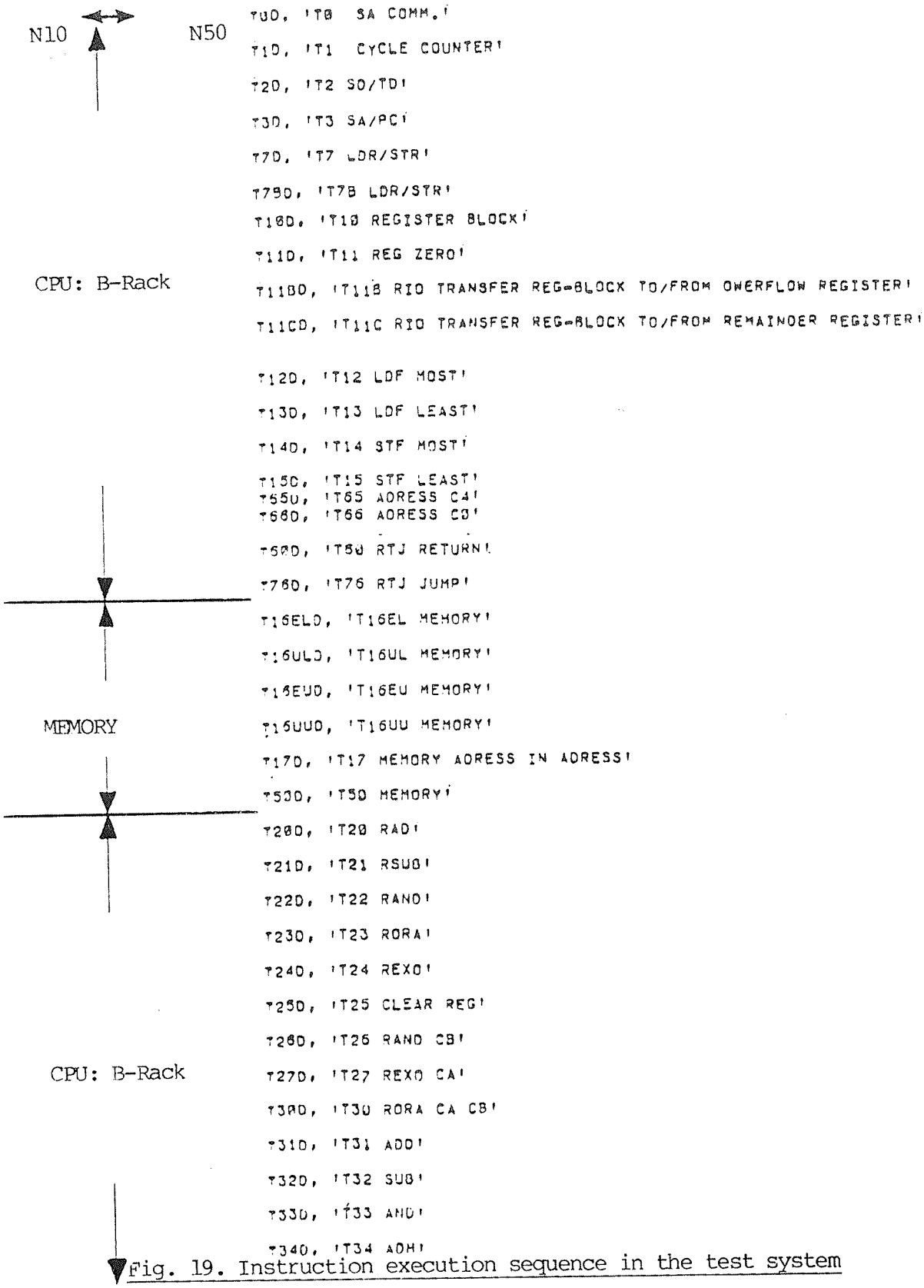


Fig. 19. Instruction execution sequence in the test system

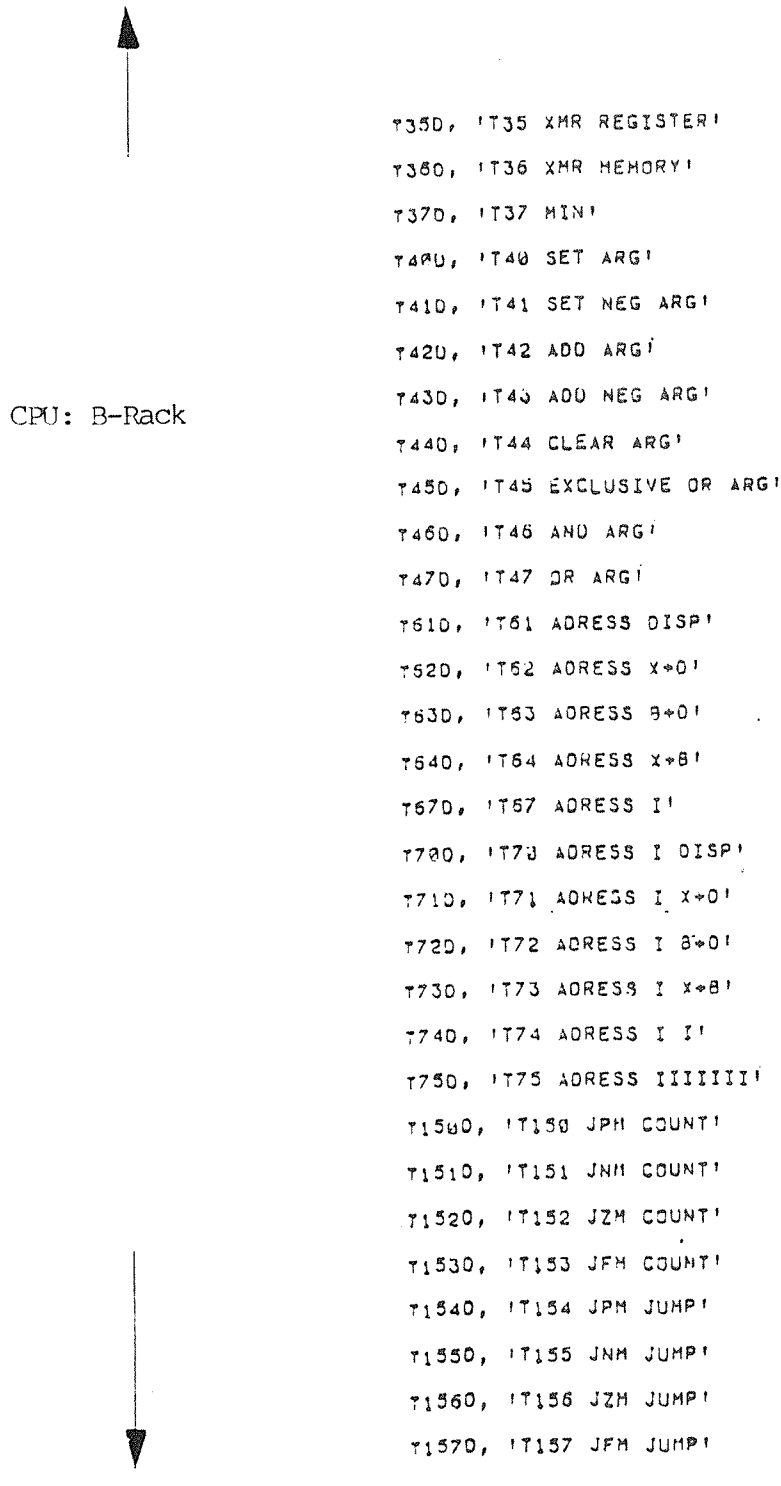


Fig. 20. Instruction execution sequence in the test system

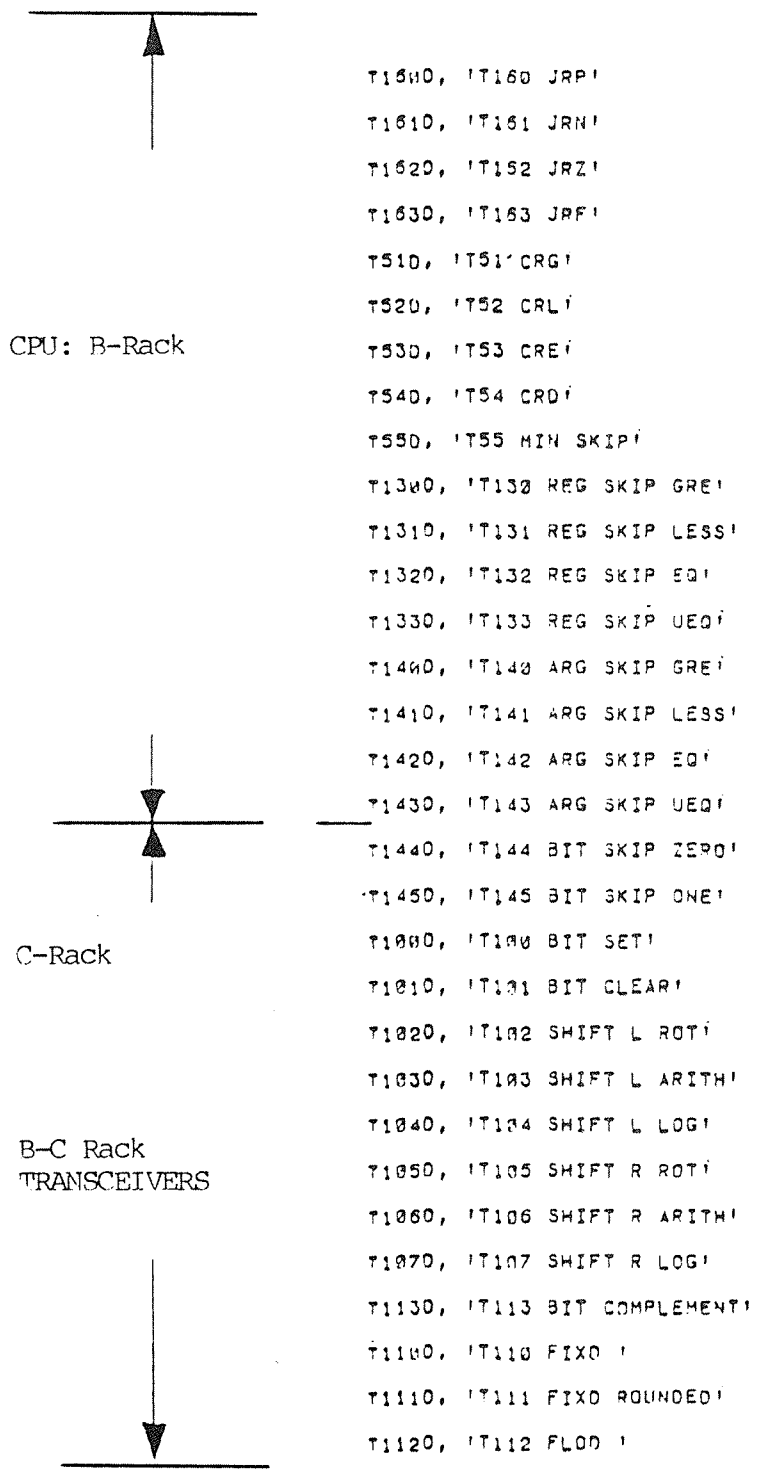


Fig. 21. Instruction execution sequence in the test system

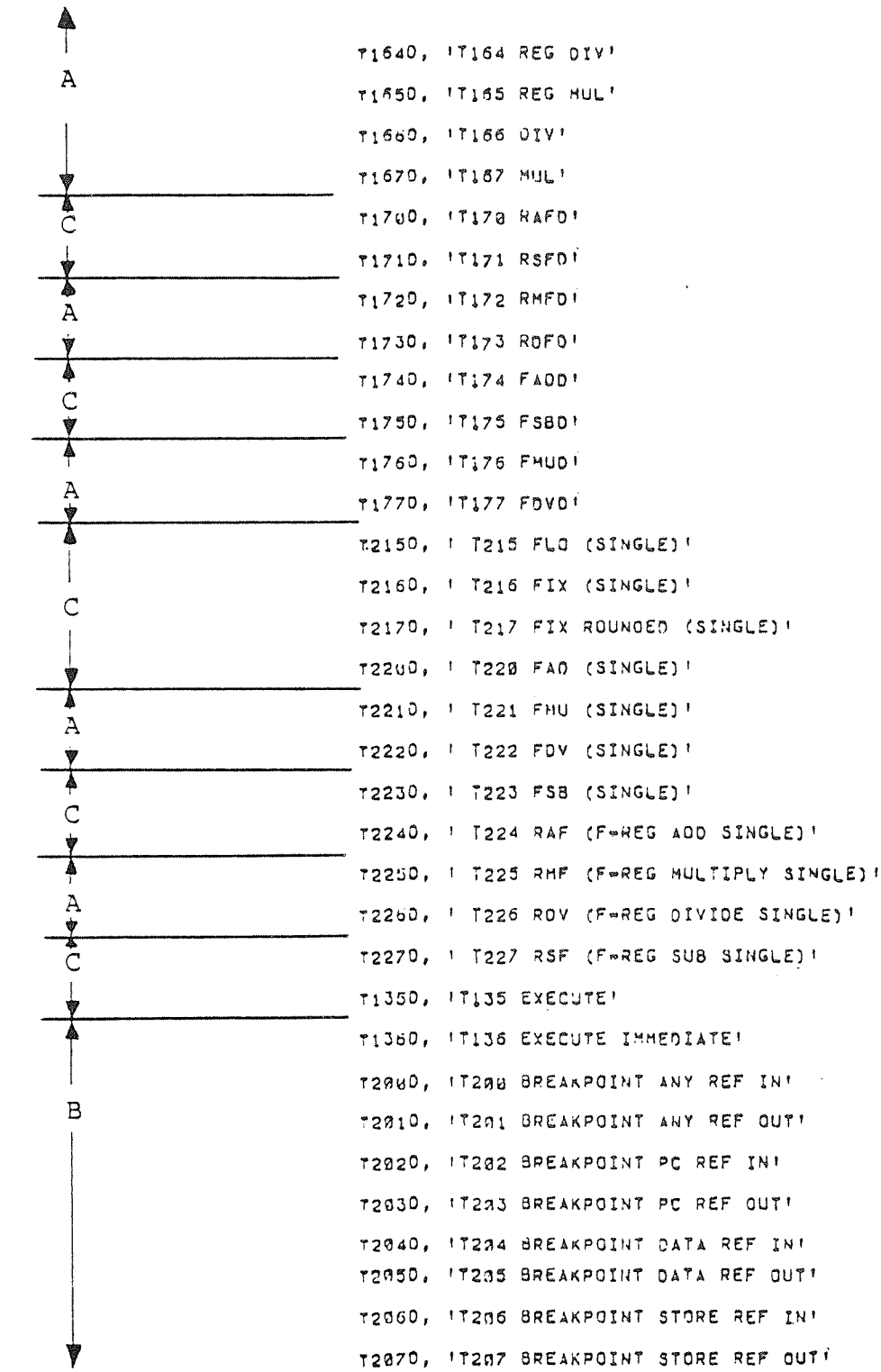


Fig. 22. Instruction execution sequence in the test system

5.2.2. TINST-50

TINST-50 HAR-1840

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A Hardware Test Program for Nord-50 instruction check.
- Highlights - The following instructions are not tested: EXECUTE, FLOATING, DOUBLE SHIFT, CONVERT
  - Loaded, started and examined via the Nord-50 MONITOR.
  - All of the instructions are fetched from the Nord-50 main memory.
  - This program is equivalent to the Nord-10/S Instruction Check Number One (ONE-CHECK).
  - A "WAIT" Number appears of the terminal as opposed to an extensive printout.
- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.
  - Load procedure when the MONITOR has been entered:
    * LOAD TINST or
    * LOAD (N50-TEST) TINST (if not logged on as user TEST).
  - Start address = 27.

Notes:

- 1. The assembly listing is necessary for finding the failing instruction. The program listing is given in Appendix F.

Additional Information:

- 1. The patch for letting the program loop by using the Nord-50 MONITOR is:

```
1107/RTJ 0 27
```

- 2. Do this if the SAVE/UNSAVE routine is failing:

```
* PLACE TINST      or    * PLACE (N50-TEST) TINST
* H-G 27                 * H-G 27
```

Note. The SAVE/UNSAVE routine is bypassed.

- 3. If the Nord-50 is functioning properly, the print-out is:

```
-xxx END XXX - AT : 1107
```

- 4. If the Nord-50 is not functioning properly the program contains the following error steps:

- STOP 0 - at:30	RTJ error
- STOP 0 - at:33	RTJ error
- STOP 0 - at:21	Further information about the error can be found in the locations listed below:

22/Content of register 1 (ACC)
23/Content of register 2 (WORK)
24/Content of register 3 (RES)
25/Address plus 1 of the test causing the error
26/Content of register 21 (ACC + 20_8)

Additional information about MPY and DIV (Address 25 = 763 or Address 25 = 1105) can be found in:

767/Multiplier/Dividend (A operand)
770/Multipleir/Divisor (B operand)
771/Expected result - lower 32 bits
772/Expected result - upper 32 bits
773/Actual result - lower 32 bits
774/Actual result - upper 32 bits

5.2.3. (TMEM-50:SYMB)

TMEM-50:SYMB HAR-1841

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments: This test program is covered in Section 6.

5.2.4. TREG-50:SYMB

TREG-50:SYMB HAR-1842

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A Hardware Test Program to check the Nord-50 register block.
- Functions - Test program for the 64 registers.
  - All registers are checked with 64 different patterns.
- Highlights - Loaded, started and examined via the Nord-50 MONITOR.
- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.
  - Start address = 0.
  - Load procedure from the MONITOR:

```
* LOAD TREG
```

  - If the SAVE/UNSAVE routines are failing, use these commands:

```
* PLACE TREG        % Hardware GOTO address 0
* H-G 0
```

Additional Information:

- 1. If the registers are operating normally the following message is given on the terminal:

  xxx END xxx AT: 136

- 2. If the registers are not operating normally, the following error stops exist in the program:

  1 - STOP - at: 1 or: 4 Register 0 is different from 0

  2 - STOP - at: 17 Register 3 in error
  - STOP - at: 32 Register 2 in error
  - STOP - at: 45 Register 1 in error

  53/Address to expected pattern
  55/Read test pattern

  Note. Program listing is necessary for point 2. The program listing is given in Appendix G.

  3 - STOP - at: 66

  54/Expected test pattern
  55/Test pattern read
  56/Register number (octal)

  Examples of Print-outs:

```
*LOAD TREG
- STOP 0  - AT: 000066  -
*54/
000054: 37773777777     STD   77, 7777, 17, 17,I
000055: 37777777777     FDV   77, 7777, 17, 17,I
000056: 00000000052     STOP  000052
```

5.2.4.1. (SEKUND)

SEKUND-JTID HAR-2463

Directory Name: ND-10112
User Name: FLOPPY-USER

Comment: This test program is a part of test program SEKUND-JTID, HAR-2463. See description for SEKUND-JTID.

5.2.4.2. (JTID)

SEKUND-JTID HAR-2468

Directory Name: ND-10112
User Name: FLOPPY-USER

Comment: This test program is a part of test program SEKUND-JTID, HAR-2463. See description for SEKUND-IRMD.

5.2.5. SEKUND-JTID

SEKUND-JTID HAR-2463

Directory Name: ND-10112
User Name: FLOPPY USER

Comments:

- Purpose - SEKUND-JTID is a hardware test program actually consisting of test program SEKUND used in conjunction with test program JTID. It is (or they are) used to measure execution times for various instructions.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

- Special Consideration - Execution times will be dependent upon the type and configuration of the memory.

Additional Information:

> Special Note. In the F-16 Flight Simulator application, SEKUND-JTID can be run in both the HIGH-SPEED STATIC memory (EXEC-0) and the MULTIPORT memory (EXEC-10) or in a combination of the two.

The following two sets of instructions were referred from the HIGH SPEED STATIC memory (on the left), from the MULTIPORT Memory (on the right).

```
*LOAD EXEC-0

LDR  1.02
STR  1.06
ADD  1.02
LDF  1.57
STF  1.56
FAD  1.41
FADD 1.88
RAF  0.86
RAFD 0.86
FMU  3.99
RMF  3.44
RMFD 6.61
FDV  3.99
FDVD 7.63
RDF  3.44
RAD  0.48
ADDA 0.48
MPY  4.97
DIV  2.00
ADM  1.53
SLR  0.86
SLRD 0.87
JFM- 0.54
EXC' 0.95
EXC" 1.43
JFM+ 1.00
- *** END ***  - AT: 0000474  -
*
```

```
*LOAD (TEST)EXEC-10

LDR  1.45
STR  1.43
ADD  1.45
LDF  2.23
STF  2.04
FAD  1.85
FADD 2.50
RAF  0.88
RAFD 0.88
FMU  4.40
RMF  3.40
RMFD 6.52
FDV  4.40
FDVD 8.16
RDF  3.40
RAD  0.64
ADDA 0.64
MPY  5.34
DIV  2.39
ADM  2.06
SLR  0.89
SLRD 0.89
JFM- 0.64
EXC' 1.12
EXC" 1.93
JFM+ 1.20
- *** END ***  - AT: 0100474  -
*
```

HIGH SPEED STATIC MEMORY MULTIPORT MEMORY

The Load Procedure for single Nord-50 configurations is as follows:

```
*LOAD SEKUND SPACE
```

5.2.5.1. (ITEST)

ITEST-IRMD HAR-2467

Directory Name: ND-10112
User Name: FLOPPY-USER

Comment: This test program is a part of test program ITEST-IRMD, HAR-2467. See description for ITEST-IRMD.

5.2.5.2. (IRMD)

ITEST-IRMD HAR-2469

Directory Name: ND-10112
User Name: FLOPPY-USER

Comment: This test program is a part of test program ITEST-IRMD, HAR-2469. See description for ITEST-IRMD.

5.2.6. ITEST-IRMD

ITEST-IRMD HAR-2469

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A special test program for the reliability test of integer multiply and divide arithmetic (A-Rack).
- Highlight - The number of the test patterns is given by the number of bits per group parameter. See Table 5.2.
- Implementation Instructions - See PD Sheet in the NORD Software Library.
- Special Consideration - The Nord-50 must be in operating condition.

Notes:

- 1. Two files must be opened for WRITE prior to running the program. They are:

  * OPEN TERM 5 W

  * OPEN TERM 7 W

- 2. Self- documenting.

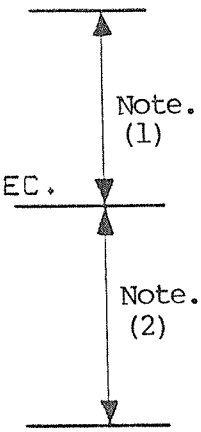
BITS PER GROUP	NUMBER OF TEST DATA	TIME PER RUN (In Seconds)	
1	33	0.191	
2	49	0.420	
3	71	0.882	
4	121	2.562	
5	187	6.120	
6	316	17.475	
7	509	45.339	
8	1021	182.427	
9	1534	411.802	
10	3070	1649.358	
11	4095	2934.579	
12	8191	11741.182	
13	16383	46970.461	% Will give memory out of range with 32K Nord-50 memory.

Table 5.2: BITS PER GROUP PARAMETERS

Additional Information:

- 1. A print-out example of the running of the ITEST is as shown below:

```
ENTER TEST
PASSWORD:
OK
@-50
NORD-50/1 MONITOR - J
*OPEN TERM 5 W
*OPEN TERM 7 W
*LOAD ITEST

NEW RUN DATA? (0 OR 1):    1
LOOP ON ERROR? (0 OR 1):    1
NO. OF BITS PER GROUP (<14):   1
EXPECTED TIME PER RUN IS          0.191 SEC.
NO. OF RUNS :  1000
TIME NEEDED AFTER DATA READY IS        190.575 SEC.
NEW RUN DATA? (0 OR 1):    0
        33 DATA READY
         1 RUN(S) COMPLETED
        10 RUN(S) COMPLETED
       100 RUN(S) COMPLETED
      1000 RUN(S) COMPLETED
END...DO YOU WANT RESTART? (0 OR 1):0
```



Note. (1) = Data prepare Type 1 to new run data?

Note. (2) = Program run Type 0 to new run data?

- 2. Test Information:

The integer IA multiplied with integer IB form the Product IP 1.

The integer IB multiplied with integer IA form the Product IP 2.

There is an error print-out if IP 1 is not equal to IP 2.

The quotient IQ1 = IP1/B.

There is an error print-out if IQ1 is not equal to A.

The quotient IQ2 = IP2/A.

There is an error print-out if IQ2 is not equal to B.

Notes. Products IP1 and IP2 are printed with the overflow register first.

Quotients IQ1 and IQ2 are printed with the remainder first.

5.2.7. FTEST

FTEST HAR-2466

Directory Name: ND-10112;
User Name: FLOPPY-USER

Comments:

- Purpose - A Test Program for reliability testing of the single precision floating multiply and divide. Just for SINGLE PRECISION.
- Highlight - The number of test patterns is given by the number of bits per group parameter. See Table 5.3.
- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.
- Special Considerations - The Nord-50 must be in operating condition.
  - The program needs more than 32K of Nord-50 memory in order to be loaded.

Notes:

- 1. See the HAMB-MET Test for the floating format.
- 2. See the ITEST Test for the method of running the program and for the preparation of data.

Additional Information:

- Error message Information.

Floating product P1 = Floating A x Floating B
product P2 = B x A

- Error Print-out 1 - If P1 does not equal P2,

FLOATING QUOTIENT Q1 = P1/B

- Error Print-out 2 - If Q1 does not equal A,

FLOATING QUOTIENT Q2 = P2/A

- Error Print-out 3 - If Q2 does not equal B,

BITS PER GROUP	NUMBER OF TEST DATA	TIME PER RUN (In Seconds)	
1	47	5.660	
2	73	13.138	
3	113	32.793	
4	181	81.531	
5	311	214.585	
6	505	731.456	
7	1017	1901.221	
8	1531	5865.938	% Memory out of range with only 64K Nord-50 Memory.
9	3067	23549.246	
10	6139	69333.359	

Table 5.3: BITS PER GROUP PARAMETERS

5.2.8. DFTEST

DFTEST HAR-2465

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - This Test Program is for reliability testing of the double precision floating point multiply and divide arithmetic (A-Rack). Just for DOUBLE PRECISION.

- Highlight - The number of test patterns is given by the number of bits per group parameter. This data is identical to the data for the FTEST Test. See Table 5.3.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

- Special Considerations - The Nord-50 must be in operating condition.

  - The program needs more than 32K of Nord-50 memory in order to be loaded.

Notes:

- 1. One file must be opened for WRITE prior to running the program. It is:

  ```
  * OPEN TERM 5 W
  ```

- 2. See the HAMB-MET Test for the floating format.

Additional Information:

- The error message information is the same as for the FTEST Test.

5.2.9. HAMB-MET

HAMB-MET HAR-2464

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A Test Program for floating addition and multiplication, both for SINGLE AND DOUBLE PRECISION.

- Highlights - Floating number: X
  Result of X + X (C-Rack) compared to result of 2X (A-Rack).

  - Failing pattern repeated 10,000 times.

  - Number of errors per 10,000 tries is reported.

  - A random permutation of all possible patterns is used.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

- Special Consideration - The Nord-50 must be in working condition.

Additional Information:

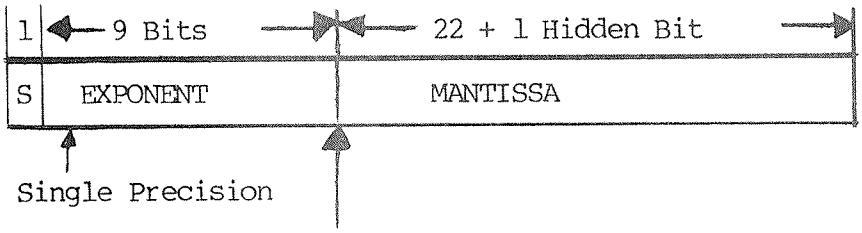
- 1. Example of error print-out:

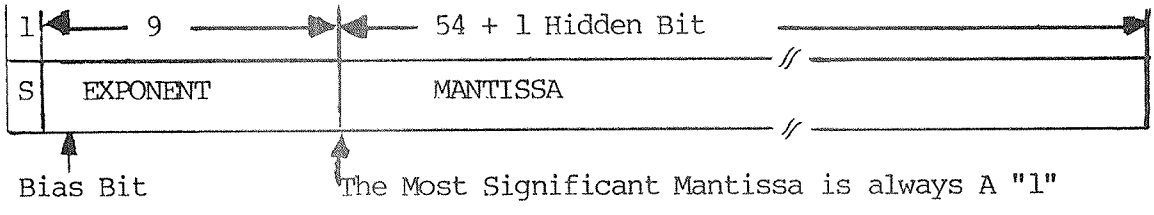
```
*LOAD HAMBURG
FOR HOW MANY MINUTES SHOULD THE TEST RUN ?: 3

   X                  X+X                2*X                FAILING/10000
    10763464416        10763464416        11003464416            1

- *** END ***  - AT: 000015  -
```

The correct answer to this problem is:
The X + X, or the C-Rack failing 2X is obtained by adding one to the exponent.

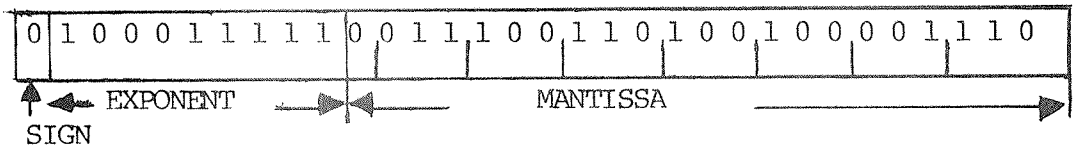
Note. The number 2 which multiplies the X is converted to floating point format in the C-Rack. If both X + X and 2X are failing, check the C-Rack.

- 2. Nord-50 Floating Formats:





THE FLOATING NUMBER X IN THE EXAMPLE WILL THEN APPEAR AS:



5.2.10. TCOR-NGO

TCOR-NGO HAR-2462

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - Verification program to test floating arithmetic.
- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.
- Special Consideration - A data file TCOR-DATA:DATA must be present to run this program.

Additional Information:

- Figure 22 shows a TCOR print-out.

```
*OPEN TCOR-DATADATA 5 R
*OPEN TERR 6 W
*LOAD TCOR-N50

TERRAIN CORRECTION INCREMENT PR. KM RADIUS. CHECK AGAINST UNNORMAL VALUES

DISTANCE KM 0    0.002 0.068  0.23  0.59  0.90  1.28  1.75  2.29  2.87  3.52  5.24  8.44  12.4  18.8  28.8  58.8  99.0 166.7
                                                                                                                    CIRCLE
E28T063      0.00  0.00  0.00  1.07  1.87  0.39  0.82  0.47  0.58  0.44  0.458 0.220 0.075 0.027 0.014 0.009 0.011 0.011  0.00
D28T080      0.00  0.00  0.00  1.42  1.30  0.72  0.95  0.59  0.75  0.55  0.091 0.088 0.040 0.032 0.040 0.019 0.014 0.012  0.00
D28T110      0.00  0.00  0.00  0.21  0.89  0.00  0.86  0.75  1.21  0.72  0.244 0.133 0.073 0.035 0.017 0.016 0.012 0.012  0.00
D28T100      0.00  0.00  0.00  7.57  1.11  6.83  2.70  2.00  0.54  0.73  0.438 0.217 0.057 0.023 0.010 0.009 0.009 0.010  5.30
D28T061      0.00  0.00  0.00  3.87  2.94  4.25  3.32  2.59  1.37  0.83  0.538 0.355 0.208 0.125 0.078 0.037 0.022 0.017  9.08
D28N012      0.00  0.00  0.00  0.18  0.03  0.33  0.12  0.08  0.46  0.74  0.417 0.374 0.260 0.132 0.076 0.006 0.001 0.001  0.00
D28T097      0.00  0.00  0.00  7.15 12.30  0.00  3.99  1.39  0.59  0.45  0.143 0.060 0.054 0.062 0.055 0.020 0.013 0.011 13.18
E28T062      0.00  0.00  0.00  2.42  2.24  1.18  0.61  0.57  0.57  0.52  0.432 0.309 0.085 0.037 0.022 0.012 0.011 0.011  2.77
D28T092      0.00  0.00  0.00  1.75  4.85  2.79  4.54  3.57  2.59  1.93  1.627 0.513 0.136 0.058 0.045 0.025 0.016 0.014  0.00
D28T102      0.00  0.00  0.00  9.45  7.42  6.30  3.43  2.66  1.47  0.97  0.674 0.472 0.239 0.113 0.053 0.034 0.020 0.016  7.19
  STATION  LATITUDE  LONGITUDE  ELEVATION  LAYER  TERRAIN  FREEAIR  BOUGUER ANOMALY

  E28T063  62 11.23   8 13.33   1414.4   159.58   5.99   436.08   -60.62
  D28T080  62 11.62   7 50.32   1507.1   169.99   6.04   464.65   -64.25
  D28T110  62 13.28   7 58.51   1445.8   163.10   5.96   445.76   -63.38
  D28T100  62 13.43   8  7.69   1315.5   140.47  12.24   405.60   -63.75
  D28T061  62 14.36   7 41.21   1755.9   197.90  19.02   541.33   -59.13
  D28N012  62 15.90   8 11.30    572.0    64.70   6.18   176.39   -61.87
  D28T097  62 16.80   7 46.57   1417.0   159.87  17.08   436.08   -59.00
  E28T062  62 17.45   8 16.37   1401.3   158.11   7.73   432.04   -56.75
  D28T092  62 18.54   8  0.32   1573.2   177.41  17.98   485.02   -59.00
  D28T102  62 18.72   7 52.73   1700.2   191.65  19.56   524.16   -50.38
022370 STOP  0
- *** END *** - ATI 0000015 -
*CC  LAST 3 NUMBERS OUT SHOULD BE : 19.56  524.16  -50.38 .
```

FIGURE 22

5.2.11. TCOR-DATA

TCOR-DATA HAR-2470

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - To be used with TCOR-NGO.
- Function - TCOR-DATA is the data file that must be present to run test program TCOR-NGO.
- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

5.2.12. N50 INITIALIZE

N50 INITIALIZE HAR-2471

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - This is an example of how to initialize the Nord-50.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

- Special Consideration - This program must be modified according to the actual system configuration.

5.2.13. N50 TEST-BATCH

N50 TEST-BATCH HAR-2472

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - This is a mode file to run all of the test programs in sequence.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

5.2.14. N50 TESTP-ASCOL

N50 TESTP-ASCOL HAR-2473

Directory Name: ND-10112
User Name: FLOPPY-USER

Comments:

- Purpose - A mode file to assemble, compile, and load all of the test programs.

- Implementation Instructions - See PD Sheet in the NORD SOFTWARE LIBRARY.

6.NORD-50 MEMORY SYSTEM

6.1. Introduction

This section contains:

- General information on the Nord-50 memory.
- The multiport memory as seen from the Nord-50.
- The test possibilities for the Nord-50 memory.
- The identification of failing memory modules.

Notes. - Sections 6.2 and 6.4 will only deal with the multiport memory system as it is found in the "European" version of the Nord-50. The "F-16" memory configuration is not covered in these sections.

- Appendix B contains a step-by-step procedure for the Nord-50 test sequence.

6.2. General Nord-50 Memory Information

The Nord-50 may be physically connected to the same memory as the Nord-10/S via one port in the multiport memory system (a shared memory) or to a separate multiport memory system as a private memory. This is shown in Figure 24.

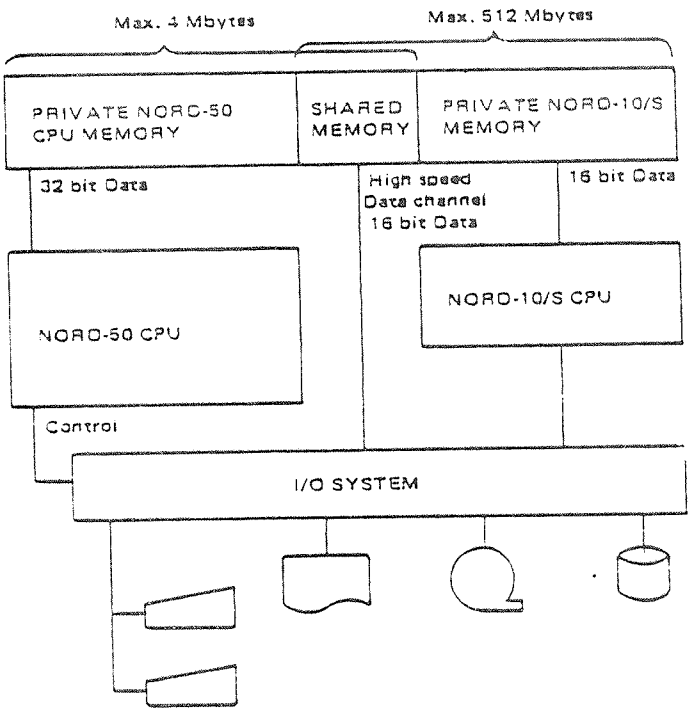


Fig. 24. The NORD-50 computer system

The Nord-50 supplies differential address lines of 20 bits + 2 control signals to the port via a one-to-one cable.

Two cables with differential lines are used for carrying the 32 bit data word. One cable is for bits 0-15 plus 2 parity bits. The other cable is for bits 16-31 plus 2 parity bits.

In memory (the multiport memory system), the Nord-50 occupies one port out of four. The remaining three ports are used by the DMA mass storage devices of the Nord-10/S. The third may be used by a second Nord-50.

The address area each port can utilize is set by the lower and upper limit switches on the address module in the multiport system. This provides for private and shared memory locations for both the Nord-50 and for the Nord-10/S.

The priority for the requests of both the Nord-50 and the Nord-10/S is fixed. The priority is determined by the physical position of the data receiver/driver module and the address receiver module in the multiport rack.

The 32 bit memory word is divided between two 18 bit memory banks with identical memory addresses. The bank in the upper multiport rack takes care of bits 0-15, while the bank in the lower rack handles bits 16-31.

The Nord-50 receives two sets of data ready/address ready signals from the memory modules in the two banks. These signals are supplied to the Nord-50 via the data module in the multiport where the signals are latched - waiting for the last signal to appear.

In a typical Nord-50 configuration, there are two 16 bit multiport channels (i.e., one for the CPU and the other for the DISK) and one 32 bit channel (i.e., for the Nord-50). The address cable of the 16 bit channels are shifted (i.e., interleaved). In this way, all odd addresses are routed to the upper multiport rack and all even addresses are routed to the lower multiport rack. Two 16 bit words, stored in memory into two sequential addresses, are read by the Nord-50 in one read operation.

Figure 25 shows an example of a memory layout in a Nord-10/S - Nord-50 configuration.

6.3. General Precautions for Swapping Nord-50 Multiport Modules

When error messages such as:

MEMORY OUT OF RANGE

or

MEMORY PARITY ERROR

appear on the terminal screen, do not swap modules in the M1 rack with the corresponding module in the M2-rack. The reason for this is that the Nord-50 must have both racks responding when supplying a 32 bit word.

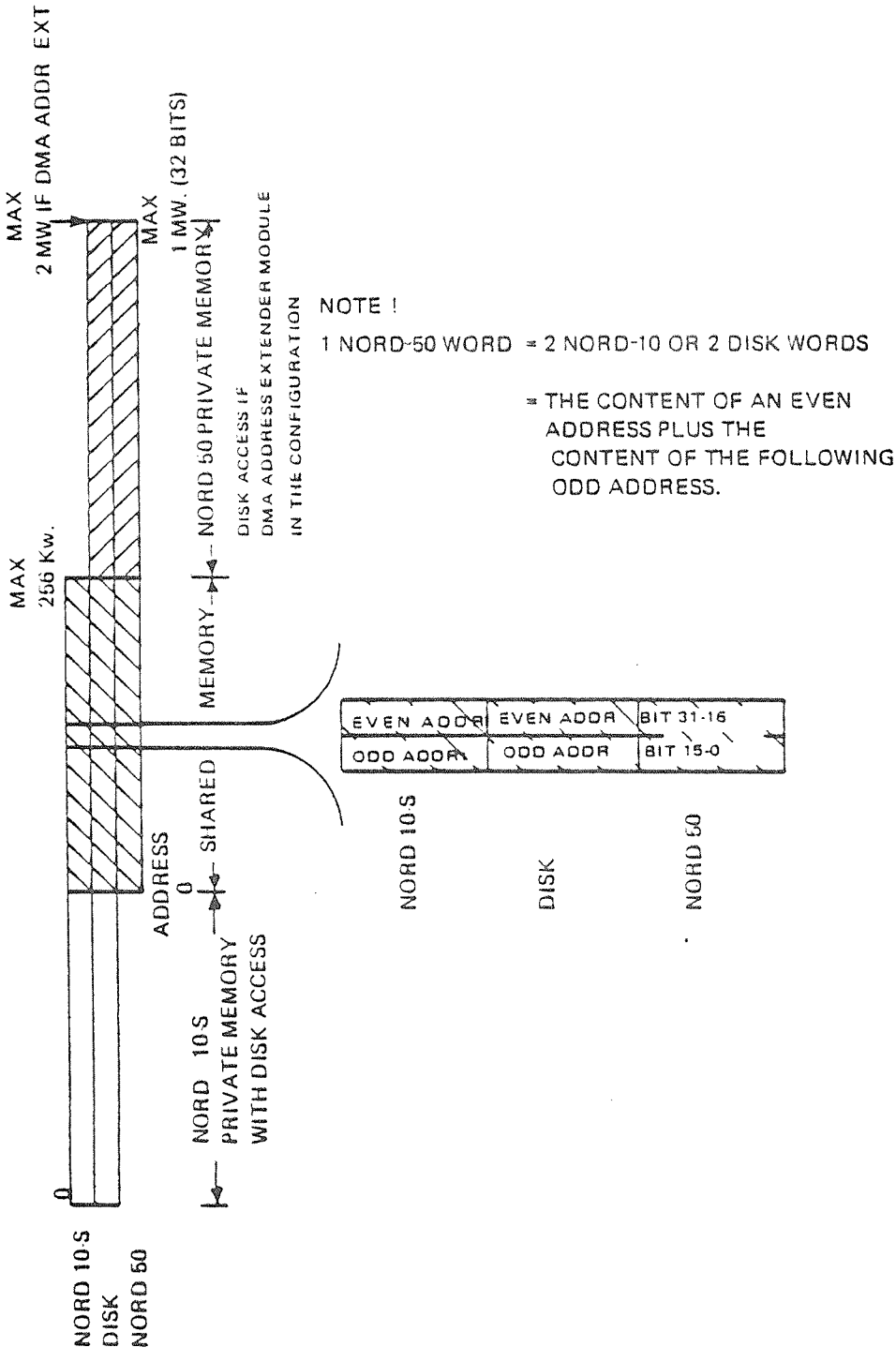


Fig. 25. Memory layout in a NORD-10S - NORD-50 configuration

6.4. Nord-50 Configurations With Multiport Memories

Figure 26 illustrates the Nord-50 configuration with old multiport.

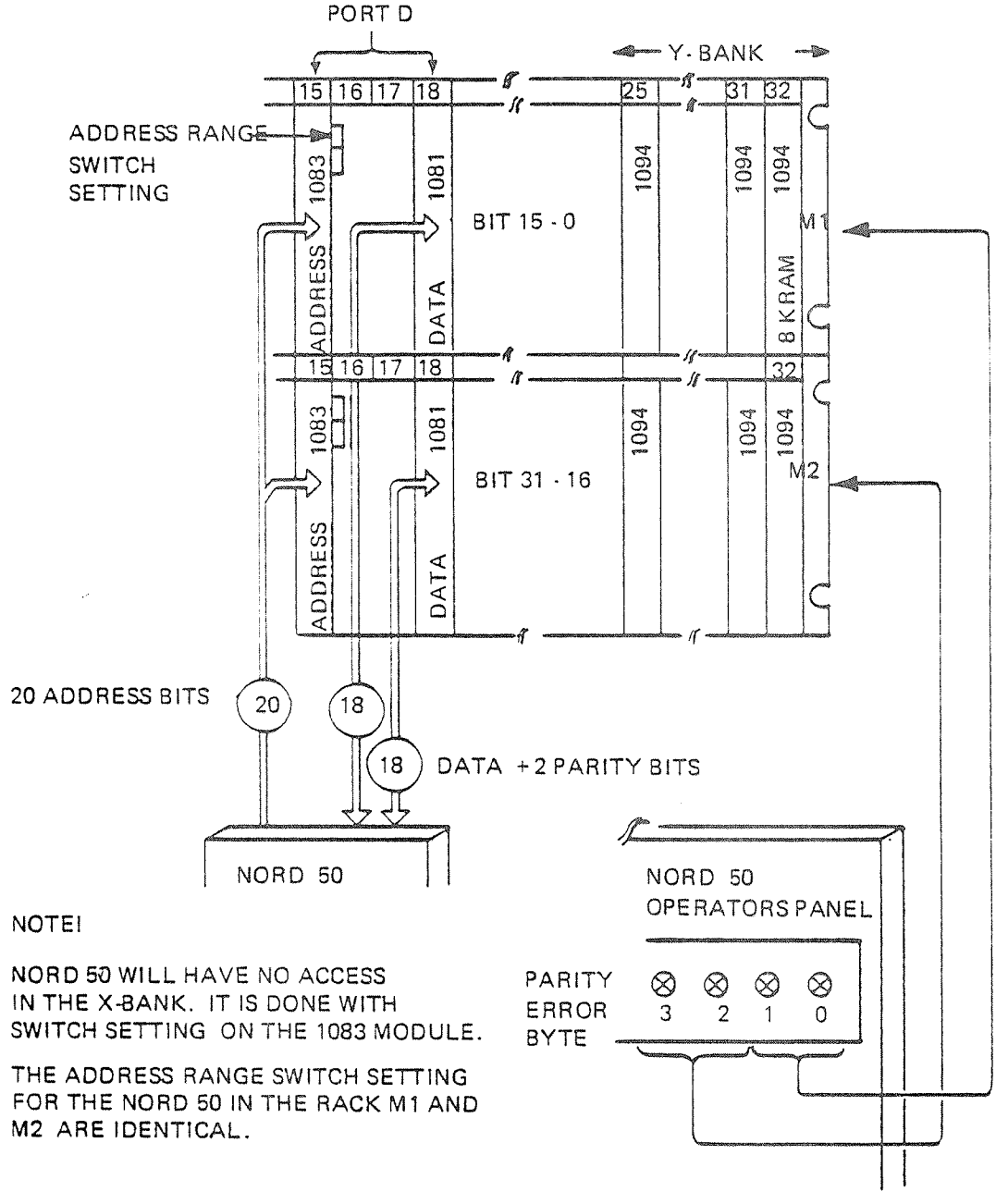


Fig. 26. NORD-50 in configuration with old multiport

General information about the Old Multiport is found in the Nord-10/S Hardware Maintenance Manual.

Figure 27 illustrates the Nord-50 configuration with big multiport.

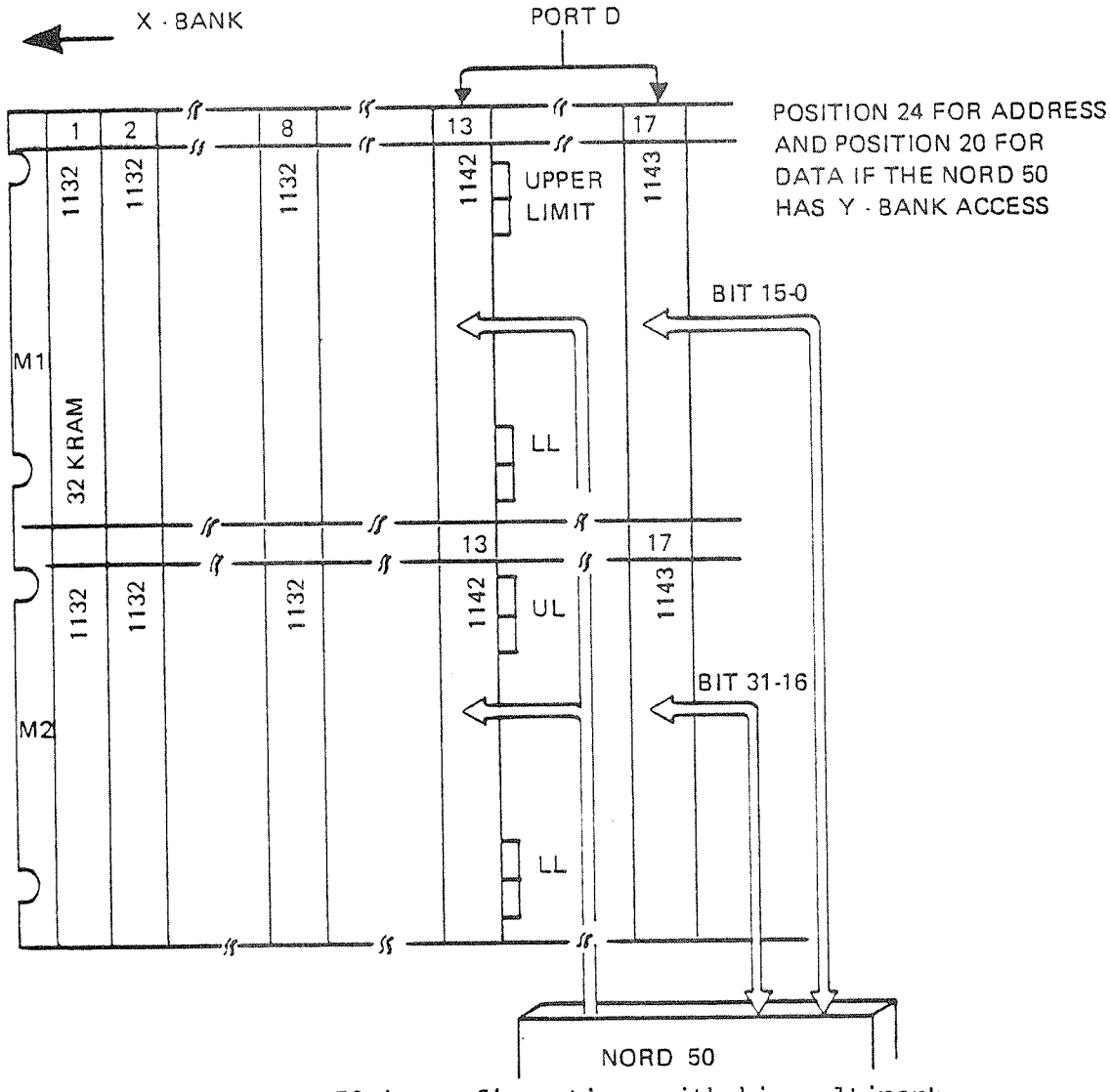


Fig. 27. NORD-50 in configurations with big multiport

Notes ! - Rack-M1 and rack-M2 receive the same Nord-50 address.

- Nord-50 data bit 15-0 to/from rack M1.
- Nord-50 data bit 31-16 to/from rack M2.
- Parity bits 0 and 1 on the Nord-50 operator's panel are referring to rack-M1, bits 2 and 3 are referring to rack-M2.

General information about the Big Multiport is found in the Nord-10/S Hardware Maintenance Manual.

6.5. Types of Tests for the Nord-50 Memory

The types of tests used for the Nord-50 memory are covered in this section.

6.5.1. SHARED MEMORY TESTED FROM THE NORD-10/S

The shared memory (i.e., a memory where both the Nord-10/S and the Nord-50 have access) can be tested by all available Nord-10/S memory test programs (i.e., MOVER, MULTI, T8KMOS, and T32KMOS).Nord-10/S test programs are found in the Test Program Description Manual for the Nord-10/S and the Nord-100.

A Nord-50 address can be converted to a Nord-10/S address in shared memory by using this formula:

```
                    NORD 10 ADDRESS FOR
                    NORD 50 ADDRESS 0
                          +
                   (NORD 50 ADDRESS)*2
                          +
                          0 if bit 31-16
                      or
                          1 if bit 15-0

                 .________________________________.

NORD 10 ADDRESS  =

                          .________________________.
```

6.5.2. SHARED AND NORD-50 PRIVATE MEMORY TESTED FROM THE NORD-50

They are:

1. TEST-MEM : command in the Nord-50 MONITOR
2. TMEM : Nord-50 test program
3. Additional test possibilities

6.5.2.1. TEST-MEM

The command for TEST-MEM is built into the Nord-50 MONITOR. When the command:

*TEST-MEM

is given, the Nord-50 is set in MEMORY EXAMINE/DEPOSIT MODE by the monitor. Data and addresses are then supplied from the Nord-10/S via the I/O system in the Nord-10/S and the Nord-50 CPU to the Nord-50 memory, private and shared.

The test is performed by reversing the data flow and checking the received data.

The following are the highlights of this program:

- This memory test program is in the Nord-50 MONITOR. The command is also in the monitor.

- The Nord-50 memory is tested by depositing/examining the memory via the Nord-10/S Input/Output System.

- The Nord-50 memory is tested in decimal increments of 1K.
- Patterns: 0
  -1

  125252525252
  125252525525
  37700177400
  37777600000
  00000177777

  Address in address test (2 x 16 bits address)

- See Figure 28 for an example of an Address in Address Test.

- Error print-outs:

  Pattern test failing

  Address test failing

- Looping is possible if this command is given before program start

  *LOOP-ON

  Program will loop, and if there is an error, the failing pattern will be repeated

  *LOOP-OFF

  Note. The address protect registers BP and BQ must be reset prior to the start of the test. Reset BP and BQ with this command:

  *B-C ,,P

Examples of this print-out:

```
@ N-50

NORD-50    MONITOR - K
*B-C ,,P

*TEST-MEM

START ADDRESS: 0

NUMBER OF 1K BLOCKS TO BE TESTED : 32

OK!
*
```

The Address in Address Test, indicated in the highlights is used both in TEST MEM and TMEM.

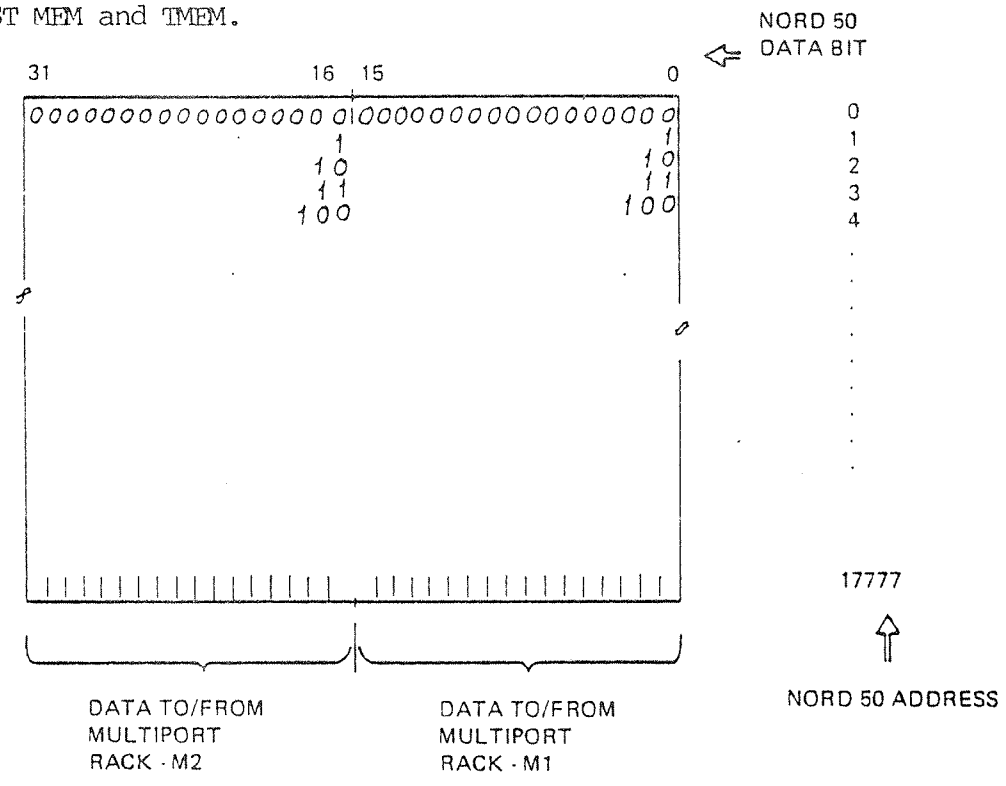


Fig. 28. Address in address test

The two racks in the multiport memory receive the same Nord-50 address. Data bit 15-0 is for the upper rack while data bet 31-16 is for the lower rack.

The Address in Address Test is built up of two blocks of 16 bits each. By using this method it is possible to detect address error both in the M1 and M2 racks . If both racks are failing, suspect the source = Nord-50.

6.5.2.2. TMEM

The Nord-50 memory is tested with the program TMEM. This program is loaded and started with the *LOAD command in the Nord-50 MONITOR. Addresses are specified so that all or any part of the shared memory, the high speed static memory or the Nord-50 private memory can be tested.

The highlights of this test are:

- The test program is used to test and run the Nord-50 memory.
- The program is loaded and started via the Nord-50 MONITOR.
- The following tests may be specified:

Test1*:Store/Read 0
Test2*:Store/Read - 1
Test3*:Address in Address Test (2x16 bits address)
Test4*:Address in Address Test with inverse address
Test5**:Walking tests
a) writes a pattern in the test area
b) writes the pattern inverted in the first location
c) checks that no locations are changed
d) resets inverted pattern

* = Fast Tests
** = Overnight Test

- Start address = 0

- Examples of Print-out:

```
@N-5 1
NORD-50/1 MONITOR - J
*B-C ,,P
*LOAD TMEM

THIS IS YOUR N-50 MOS MEMORY TEST PROGRAM
THE PROGRAM OCCUPIES LOCATIONS: 0000000  TO  0001227
LOWER TEST ADDRESS(OCTAL): 100000
UPPER TEST ADDRESS(OCTAL): 117777
SPECIFY TESTS TO BE RUN BY OCTAL
NUMBERS TERM. BY CR.STOPS WHEN 0 IS TYPED
77 MEANS ALL TESTS
77

WANT TESTS TO LOOP? 0:NO  1:YES
1
ERRORS WILL BE REPORTED IN THE FOLLOWING FORMAT
TEST1,2,3&4: <FAILING ADDR.>,<FOUND DATA>,<EXP.DATA>,<TEST NO>
TEST5: <WALK PATT. ADDR.>,<SAME AS ABOVE>,<READ/WRITE>
```

6.5.2.3. Additional Testing Possibilities

The stand-alone test program

MEM-TEST-2304

is available for testing the memory shared between the universal DMA interface and the Nord-50.

The stand-alone test program

MEMORY-MAP-2317

is used to examine memory configurations. The program will determine the parts of memory that are shared and indicated the manner of program division.

6.6. Failing Memory Module(s) Identification

Figures 28 and 29 will assist in identifiying failing modules.

Figure 28 should be used in conjunction with Section 6.4.

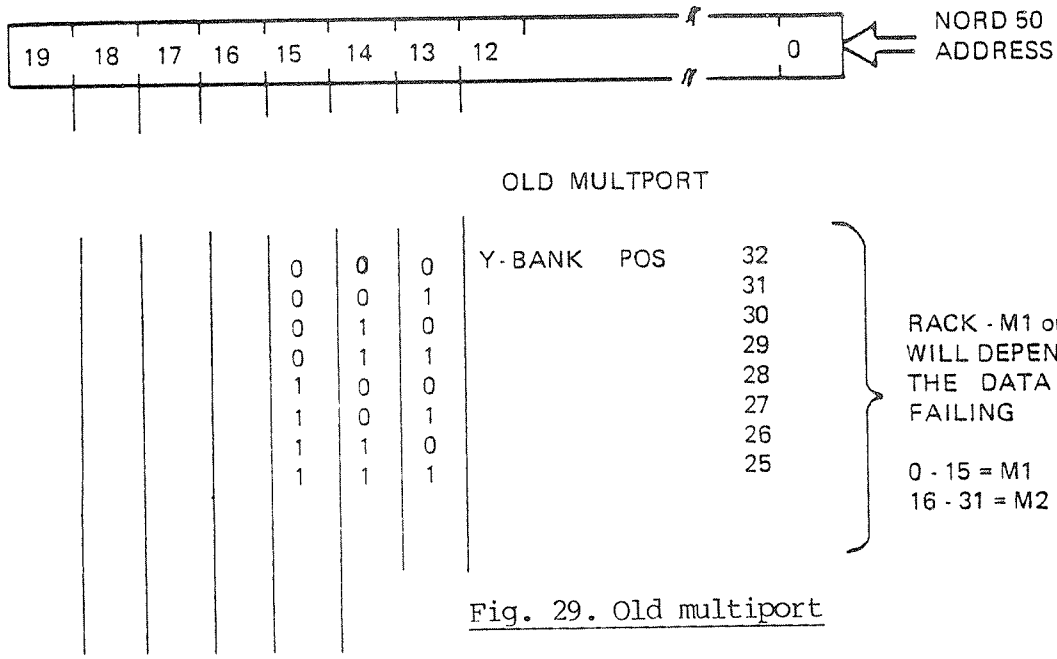


Fig. 29. Old multiport

Figure 27 should be used in conjunction with Section 6.5.

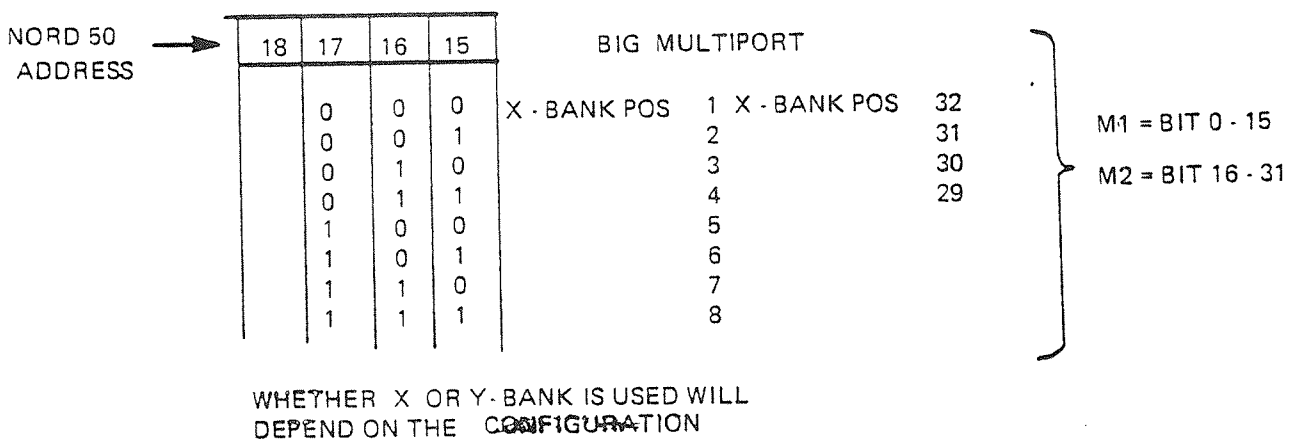


Fig. 30. Big multiport

7.NORD-50 MEMORY EXPANSION

7.1. Introduction

This section is divided into three parts. The first part contains a description of the hardware implementation. The second part contains a description of the software implementation. This part also gives some general remarks about the memory in a Nord-10/S - Nord-50 configuration. In the third part the software commands for changing the memory are covered in detail.

7.2. Memory Expansion Hardware

In configurations with the BIG-MULTIPORT memory, the memory is expanded in 32k x 32 bit increments (2 memory modules). This equals 64k x 16 bit increments or 100 octal pages seen from the disk or the Nord-10/S. In the "old" multiport with 8k (1094) memory modules, the memory can be expanded in 8k x 32 bit or 16k x 16 bit increments.

Note. The memory expansion must always be symmetrical in the M1 and M2 memory racks. Remember to update the address-range switches in the address module of the Nord-50 port of Nord-50 private memory - or all the ports (Nord-50, DMA and Nord-10/S) if expanding the SHARED memory.

7.3.Memory Expansion Software

7.3.1.Nord-50 Memory - General

The absence of a memory management system in the Nord-50 disallows a dynamic swapping of Nord-50 memory during program execution. Therefore, all parts of a program must be in memory prior to its execution, and it must remain there throughout the execution. This also applies to the pages in shared memory. Being a part of the Nord-10/S memory, the pages in shared memory are normally subject to swapping outside the programmer's control.

A way to prevent SINTRAN from swapping a specified physical area is to load a segment and "fix" it in memory using the FIXC monitor call. This is performed automatically by the Nord-50 MONITOR when preparing for the execution of a program. In Figure 31 the Nord-50 program resides in the shaded region.

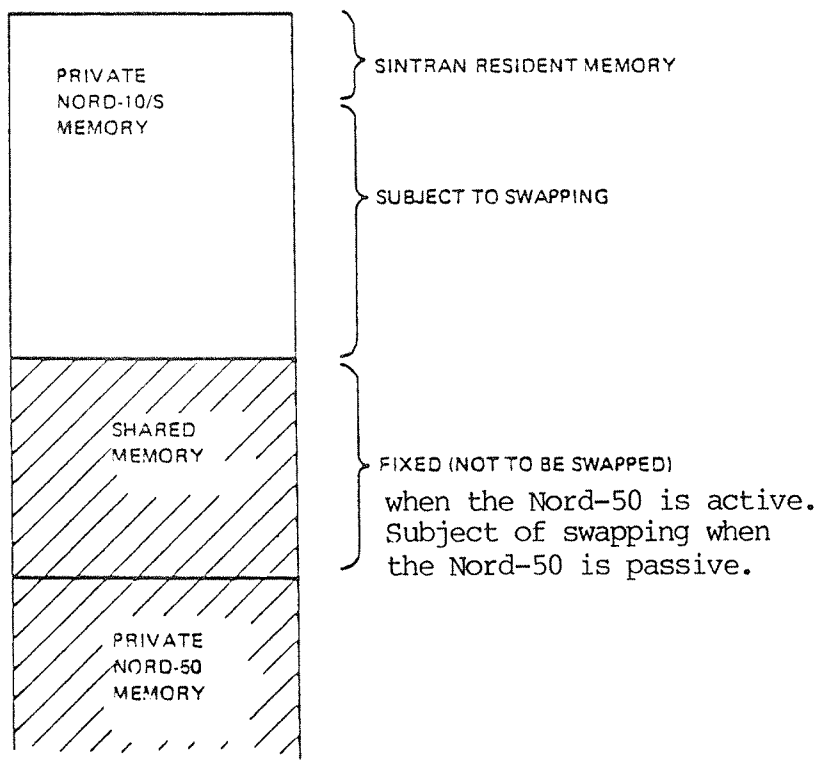


Fig. 31. Physical memory organization

Some Nord-50 programs will not occupy the entire shared memory. In such cases it is uneconomical to fix it all. A better utilization of shared memory is obtained by splitting it into partitions, where each partition is represented by a segment. See Figure 31.

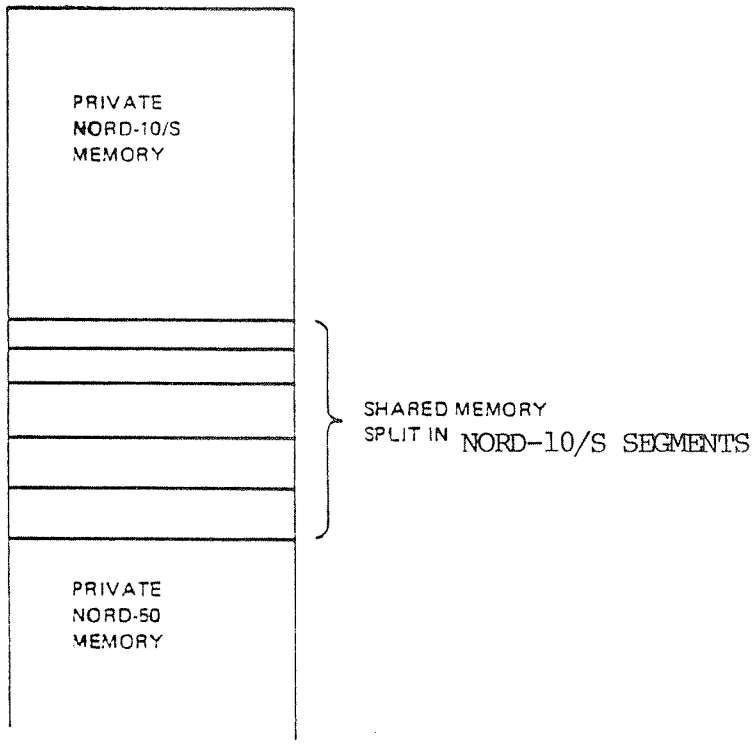


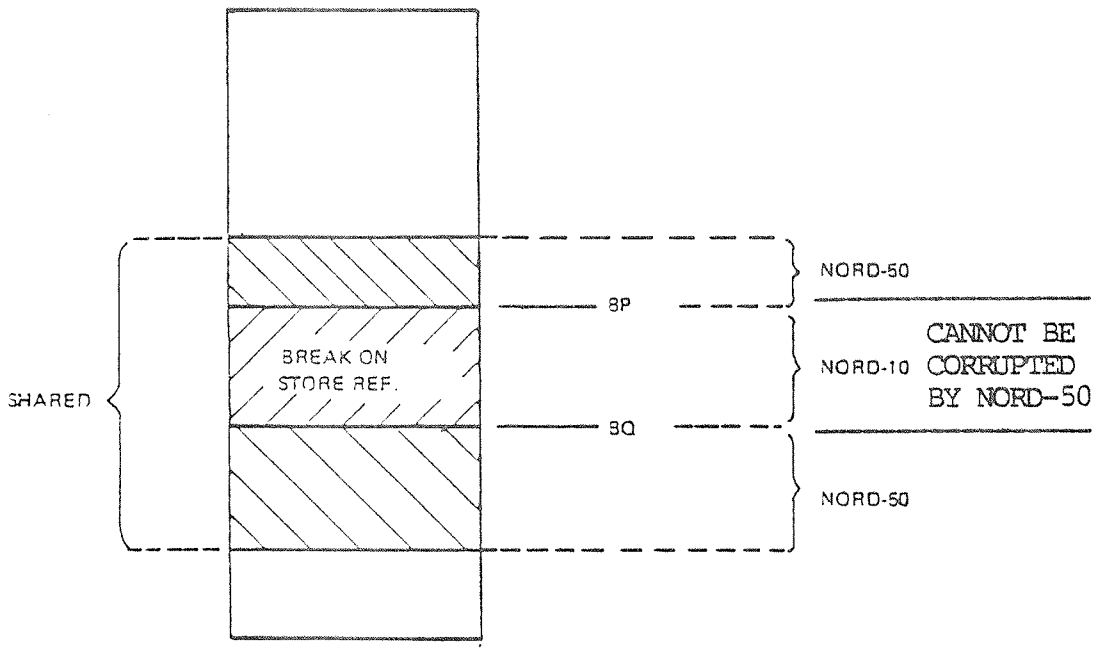
Fig. 32. Shared memory split in segments

When a part of memory is allocated for the Nord-50, the SEGMENT TABLE is used to place the segments in the correct place in the physical memory.

The segments may be of different sizes. Each segment is fixed separately if the corresponding memory space is required by a Nord-50 program.

Important
Note. The finer the partition (i.e., the smaller the segments), the better the utilization of shared memory will be. However, increased administrational overhead is the price we have to pay.* Figure 31 illustrates how shared memory should be utilized.

* Typical size is 16k (16 bits) when 128k of shared memory.



BP and BQ define break area set by:

NORD-50 Loader
*EXIT (stored in block 0 of program)
*BREAK-CONDITIONS

NORD-50 Monitor
*BREAK-CONDITIONS

Fig. 33. Shared memory allocation

In the discussion above, we have assumed all segments in shared memory to be **dynamic** segments. A dynamic segment has the quality that it will only be fixed when required. A **static** segment, however, will be fixed the first time it is required, and will remain fixed whatever program is being run.

An RT common area may also be a part of shared memory. This area is not subject to swapping and allocates no segment for this part of memory. The organization of physical memory is mapped by the Nord-50 Memory Segment Table. This table is illustrated in Figure 34.

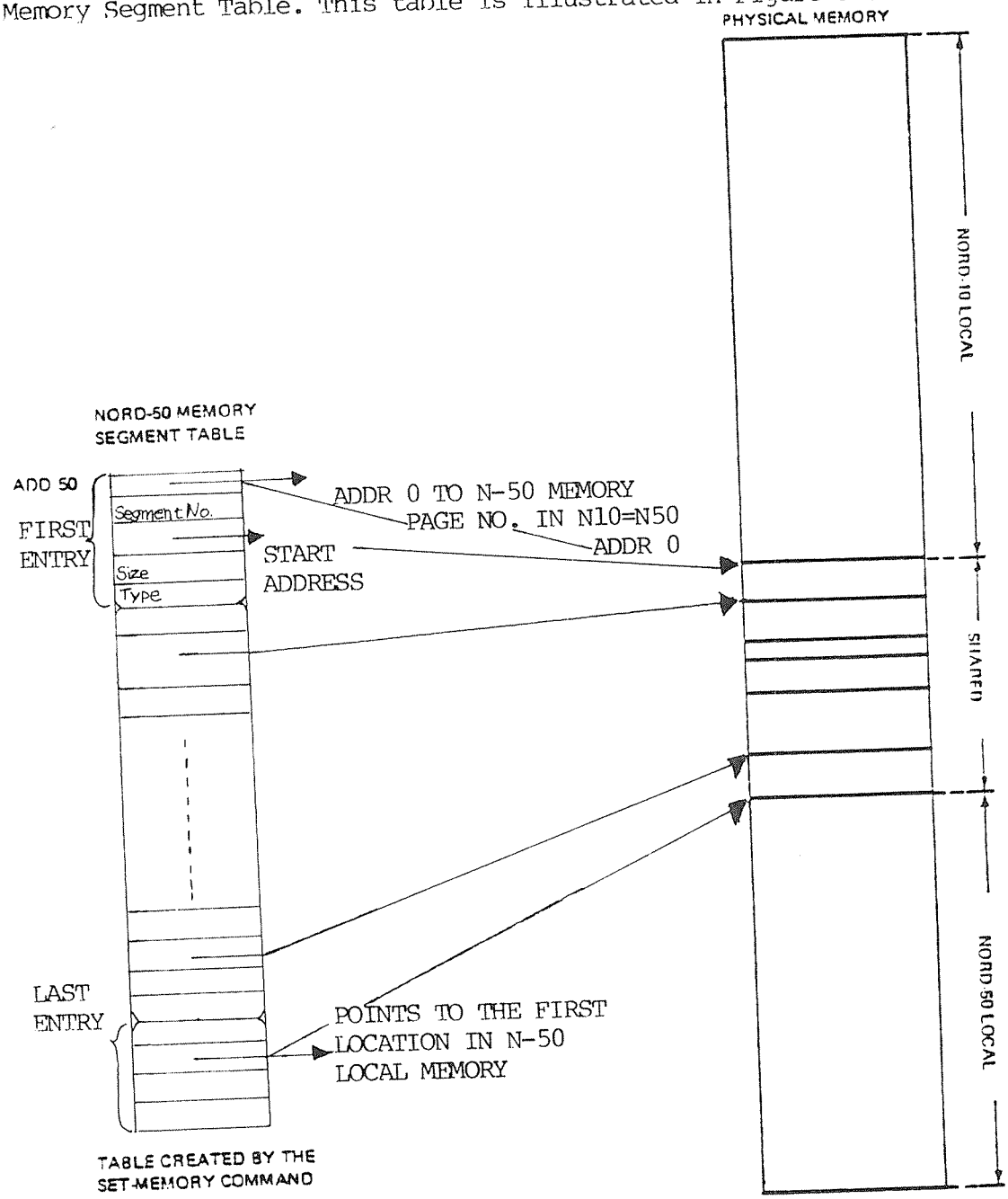
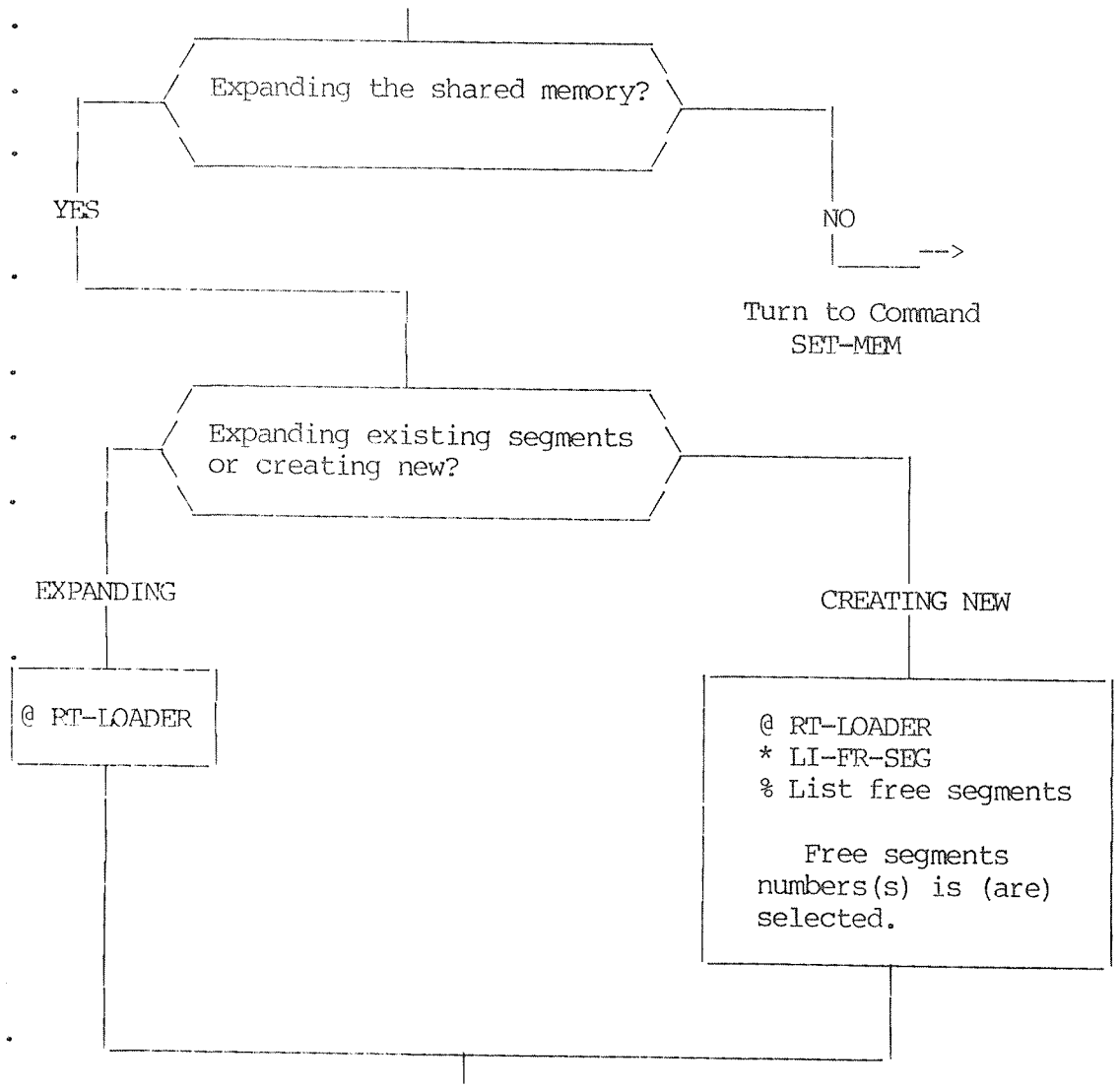
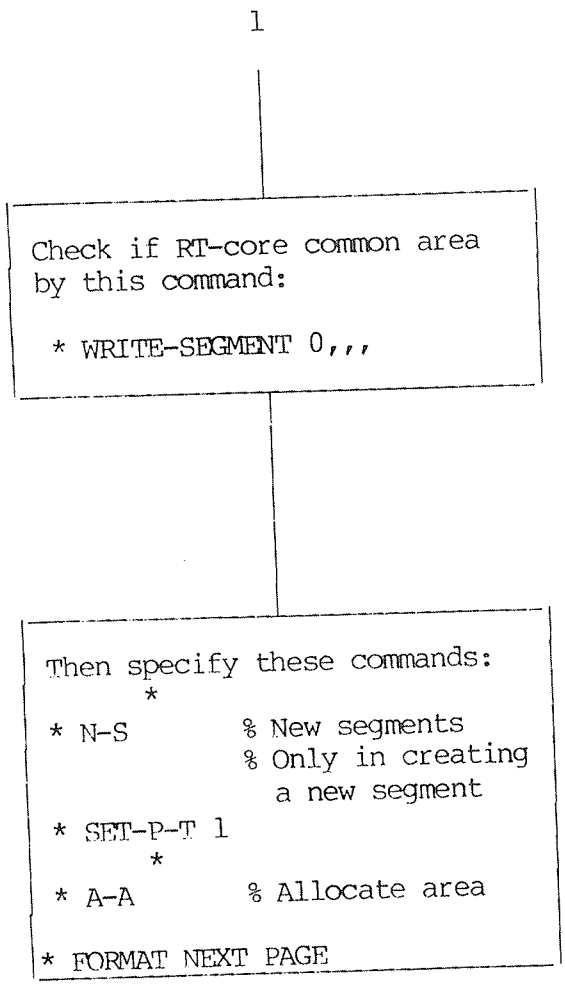


Fig. 34. NORD-50 memory segment table

7.3.2. Memory Expansion

7.3.2.1. Software Commands




1
Check if RT-core common area
by this command:
* WRITE-SEGMENT 0,,,
Then specify these commands:
*
* N-S
% New segments
% Only in creating
a new segment
* SET-P-T 1
*
* A-A
% Allocate area
* FORMAT NEXT PAGE

7.3.2.2. RT-Loader Commands

* NEW-SEGMENT	% SPECIFY A NEW SEGMENT
<segment w>	Number of the segment(s) to be expanded or segment number(s) from LI-FRI-SEG
<ring>	Protect ring 0,1,2 or 3 Default = 0
<segment type>	ND = Non demand DM = Demand
<protect bits>	Permit bits: F = fetch permitted R = read permitted W = write permitted or any combination Default = RFW
<WP/NP>	Written in page bit set WP = SET NP = Not SET (default)

Example:

```
N-S 204 2 ND RW WP
```

A-A	% ALLOCATE AREA
<segment numbers>	Segment number of the segment currently being built
<area size>	Number of 16 bit words in Octal 8 K = 20 000 16 K = 40 000 24 K = 60 000 32 K = 100 000
<lower address>	First address of the area to be allocated.

Note. - 2 pages in Nord-10 = 1 page in Nord-50

- The area size in RT-LOADER is converted to SET-MEM size in N50 MONITOR as indicated below:

RT-LOADER area size = SET-MEM in N50 MONITOR size

<AREA SIZE>/2.1000 = <SIZE>

Example:

AREA SIZE :	40 000	=	SET-MEM SIZE :		20
	60 000	=	"	" :	30
	100 000	=	"	" :	40

The RT common area is not specified here.

7.3.2.3. Nord-50 MONITOR Commands

SET-MEM

Command for generating the Nord-50 Segment Table. It indicates which segment should make which part of physical memory.

<N-10 page no.)

Nord-10 page number for Nord-50 address 0

40	=	32K
100	=	64K
200	=	128K
300	=	192K
400	=	256K

This parameter is found by the CONFIGURATION-INVESTIGATOR test program.

<segment No.>*

Allocated segment number with A-A command in RT-loader

-1 = no segment defined in this part of memory

Used when:
1. RT-common
2. Nord-50 private memory

0 = Denotes end of table

<size>*

Size in Nord-50 pages

10	=	4K N50	=	8K N10
20	=	8K N50	=	16K N10
40	=	16K N50	=	32K N10
100	=	32K N50	=	64K N10
200	=	64K N50	=	128K N10
400	=	128K N50	=	256K N10

<type>*	0	= DYNAMC
	1	= STATIC
	2	= RT-COMMON
	3	= LOCAL N50 MEMORY
	4	= " " " with DMA access
	5	= HOLE

Note. * means repeated until segment number is zero.

When the SET-MEM command is given, the size parameters are added together and tested against the physical memory. If a mismatch, the error messages listed in Section 4 are given. Figure 35 shows an example of a SET-MEM command.

```
KN
@NORD-50 3
*0
KN
SET-MEM 100 -1 100 3 70 40 1 71 40 1 -1 400 5 -1 100 4 -1 [illegible] 4 0
*0
KN
EX
```

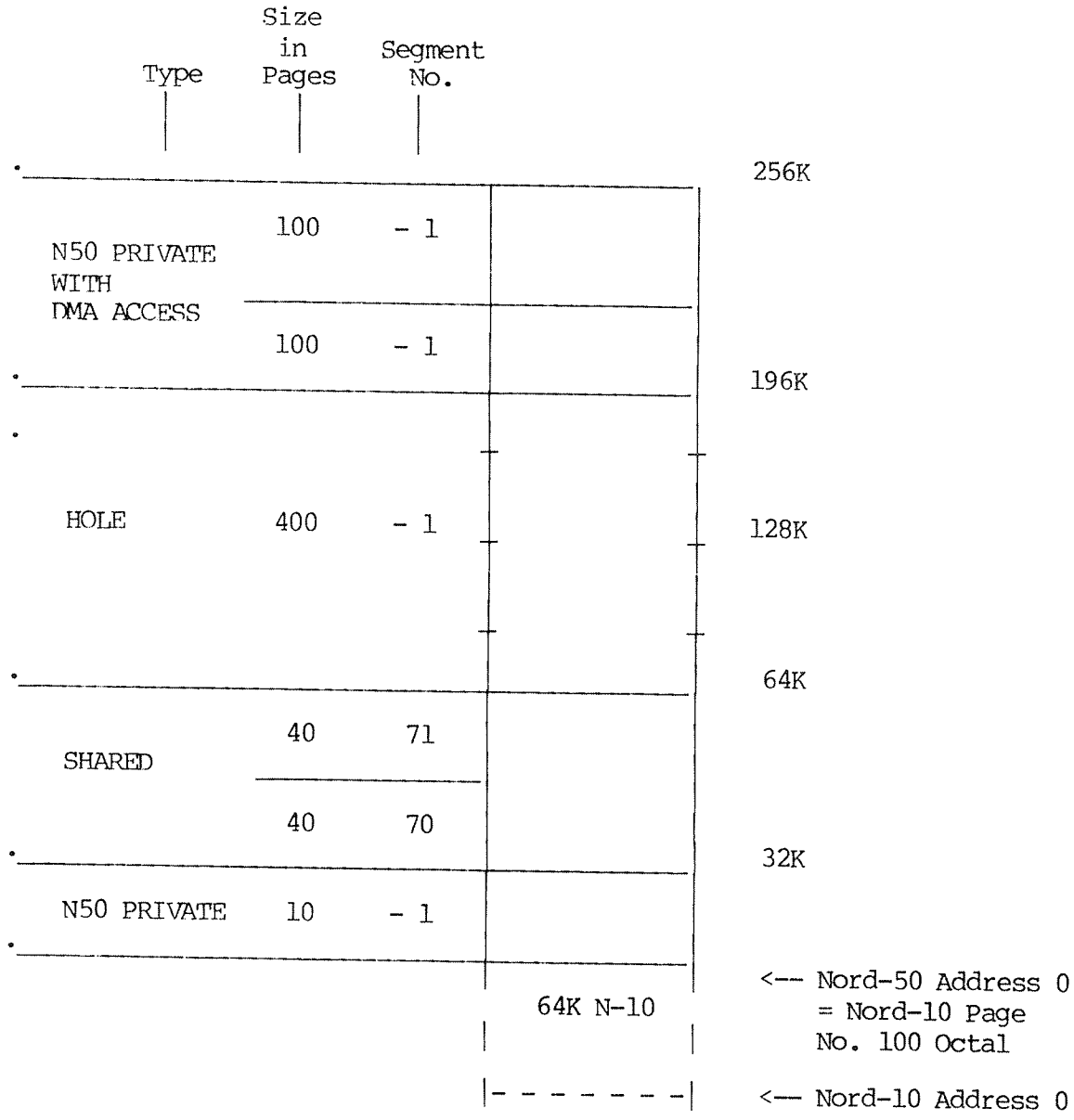


Fig. 35. Example of SET-MEM command

LIST-MEMORY <output file>

This command lists the segment table.

Example:

```
*LIST-MEMORY 1 ↵
SEGMENT NO.   FIRST ADDR.   SIZE   IN USE   TYPE
     061        000000       0020      X     DYNAMIC
     046        020000       0020            DYNAMIC
     047        040000       0020            DYNAMIC
     050        060000       0020            DYNAMIC
     051        100000       0020            DYNAMIC
     052        120000       0020            DYNAMIC
     053        140000       0020            DYNAMIC
     054        160000       0020            DYNAMIC
     055        200000       0020      X     DYNAMIC
     056        220000       0020      X     DYNAMIC
     057        240000       0020      X     DYNAMIC
     060        260000       0020      X     DYNAMIC
    NONE        300000       0020      X     LOCAL MEMORY
                                               FOR NORD-50

NORD-10 PAGE NO. FOR NORD-50 ADDRESS 0:          0100
```

8.NORD-50 POWER SYSTEM

8.1. Introduction

This section covers three topics:

- AC distribution (an example)
- DC distribution (an example)
- DC voltage measurement

For a description of the power supplies in use, refer to the Nord-10/S Hardware Maintenance Manual.

For details of the AC and DC power distribution in the Nord-50, refer to the Nord-10/S technical/mechanical drawings numbers 10/S-26-29 and 10/S 44-65.

8.2. AC Distrubution

8.2.1. Nord-50 Power Sequence

To turn ON the Nord-50, the following conditions should be fulfilled:

- The Nord-10/S must be turned on. (220 V AC to solid state relay for AC sequence control.)
- The power in the Nord-50 operators panel must be pushed. (The key must be released.)
- The main switch in the Nord-50 power supply must be ON.

  Note. The additional multiport memory supply , in some configurations is located in the bottom of the Nord-10/S cabinet.

The Nord-50 is automatically turned OFF when powered down.

Turning OFF the Nord-50 will not affect the Nord-10/S, but the multiport memory power will be turned OFF.

8.2.2.AC Distribution in the Nord-50

Figure 36 illustrates the AC power distribution in the Nord-50.

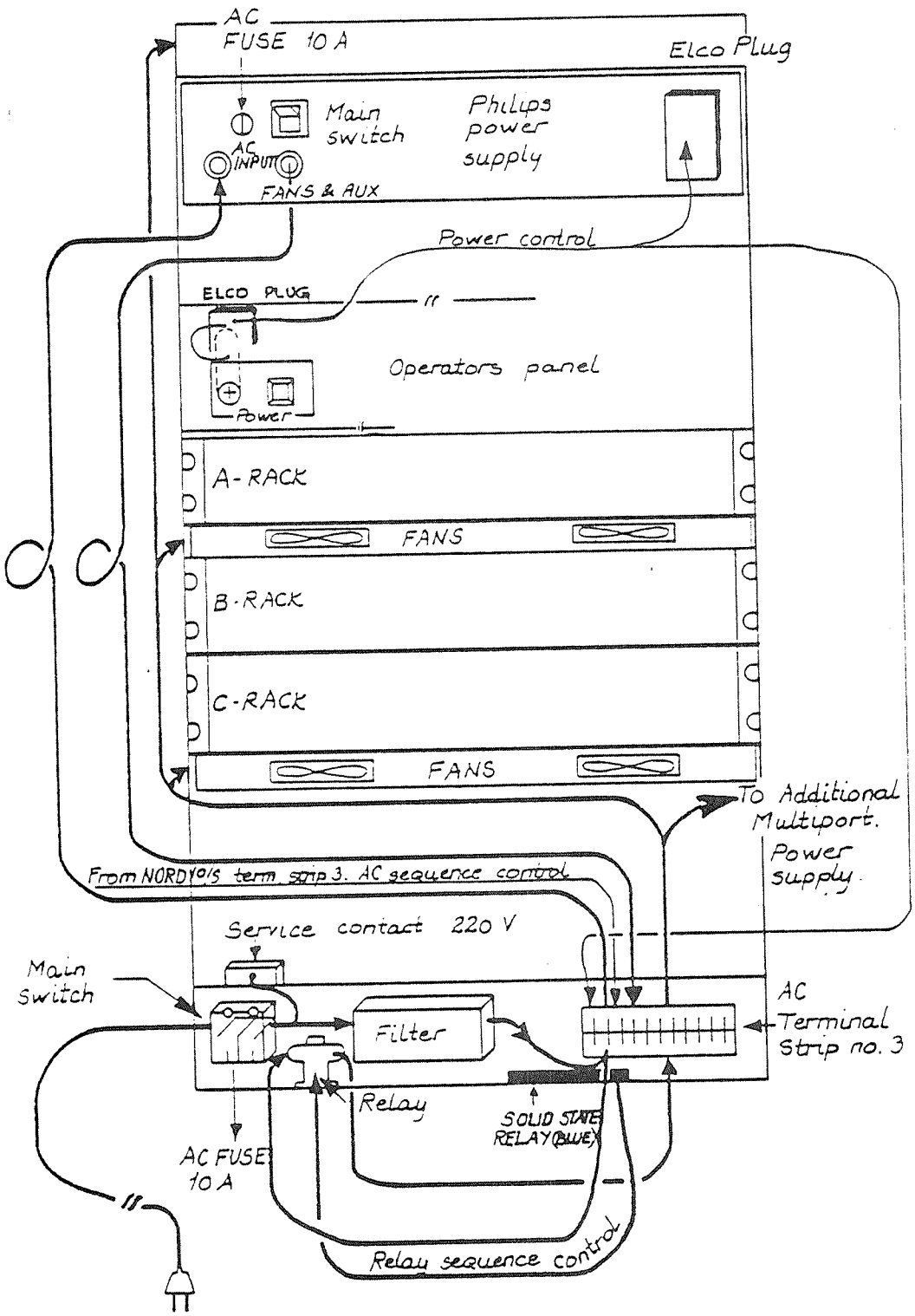


Fig. 36. AC power distribution in the NORD-50

8.3.DC Power Distribution

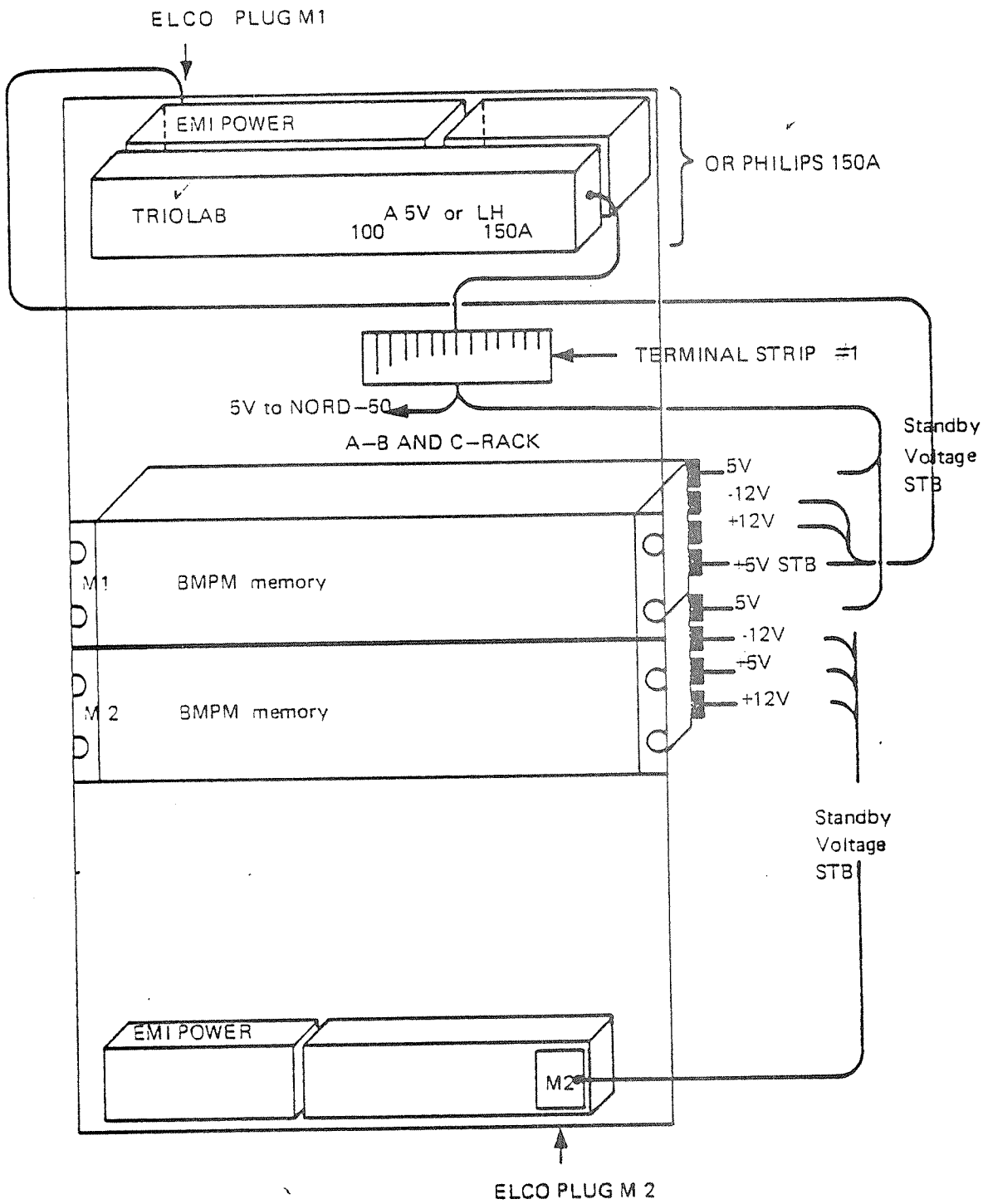
8.3.1.DC Power Distribution General

Standby voltage to the second multiport rack-M2 is supplied from a second multipower supply unit. This may be located in the Nord-50 or in the Nord-10/S cabinet.

In configurations with TRIOLAB 100A 5V power supply this supply is duplicated. One supply distributes 5V to the Nord-50 A and B-racks, while the other supplies the C-rack, and the multiport racks-M1 and M2.

8.3.2.DC Power Distribution in the Nord-50

Figure 37 illustrates the DC power distribution in the Nord-50.



(Nord-50, rear view.)

Fig. 37. DC power distribution in the NORD-50

8.4.DC Voltage Measurements

All DC voltages should be verified on a board extender module.

- Only 5 volts DC is connected to the Nord-50 racks.
- All modules in use in the Nord-50 should be inserted.
- The voltage should be checked in the middle of the rack.
- Check the multiport memory voltage in position 8. If Old multiport and if Big with the X-bank, use position 29 if Big multiport with only Y-bank.

Table 8.1 indicates the various voltages and voltage tolerances.

VOLTAGE	TERMINAL	GROUND TERM	TOLERANCE
+5V	0,1,98,99	2,3,96,97	4.95 - 5.05 VOLT
-5V	58,59	56,57	-4.9 - -5.1 VOLT
+12V * (Standby)	90,91,92,93	96,97	8k modules = 11.8 VOLT 32k modules = 12.0 VOLT
-12V * (Standby)	94,95	96,97	Not Critical
+5V * (Standby)	88,89	96,97	4.95-5.05 VOLT

* The STANDBY voltage is used by the MOS memory systems (local, multiport, and Big multiport) to "refresh" the memory.

Note. Should also be present during normal operation. Lasts for approximately 20 minutes after the AC input drops.

Table 8.1: Voltage Locations and Tolerances

APPENDICES

DETAILED CONTENTS

- - -

APPENDIX A

GLOSSARY

Note. See Section 2.5 for a MNEMONIC List. This list contains the Mnemonics and their meanings associated with the Nord-50 C-Rack.

Appendix A
GLOSSARY

Definitions

COMPUTER TERMS AS USED IN THIS MANUAL

ASC code: ASCII code

assembly test programs: Test programs in Nord-50 assembly language

Block: (As in B-Block.) A group of information units handled as a single group.

board: Circuit board.

board extender module: For extending modules

card: A module.

Central Processing Unit: (CPU) The part of the computer that controls the interpretation and execution of the processing instructions.

Console Terminal: Terminal connected to Nord-10/S

debugging: The removing of a mistake, malfunction, or a fault; for example, removing a mistake in a program, removing a malfunction in a computer, or removing a fault in a power line. disk: A revolving plate upon which data and programs are stored.

dynamic segment: Swapable segment

FLCO Plug: Plug connection in power supply

"European" version of Nord-50: All Nord-50 except F16 Nord-50's

fetch: Same as retrieve.

flip-flop: A circuit or device, usually containing active elements, capable of assuming either one of two stable states.

"hard" error: Repeatable hardware error

high speed static memory: Memory made of flip-flops

latch: As in "to latch" or "the operands are latched in the CPU . . .")

memory, private: A memory in which only one CPU has access.

memory, shared: A memory where both the Nord-10/S and, the DMA and the Nord-50 have access.

Monitor, Nord-50: Program in Nord-10/S, supervising Nord-50

multiport memory: Memory system with 4 access ports

port: (As in PORT D ADDRESS) Communication modules between source and memory modules

RT common area: Area in memory where RT-programs can exchange data

shared memory: See memory, shared.

SINTRAN III Operating System:

static segment: Segments permanent in memory during execution

TRIOLAB: Power supply name (type)

X-Bank: Part of multiport memory

Y-Bank: Part of multiport memory

Appendix A
GLOSSARY

Acronyms

A	A register.
AAX	AAX instruction. See Section 5.2.1.
AC	Alternating current.
ALU	Arithmetic Logic Unit
AR	Address Ready.
BD	Data Bus.
BMPM	(As in "BMPM memory".)
BP	A communication register. (As in <BP address>) (As in BP area.) BP = Lower. BP register.
BQ	A Communication register. (As in <BQ address>) (As in BQ area.) BQ = Upper. BQ register.
CA	Calculated memory address.
CPU	Central Processing Unit.
CR	Carriage Return.

DC	Direct Current.
DIV	Divide.
DMA	Direct Memory Access.
DR	Data Ready.
ECO	Engineering Change Order
EMI	(As in "EMI power".)
FDV	Floating Divide.
FDVD	Floating Divide Double Precision
FIO	Formated Input/Output. A routine.
FMU	Floating Multiply.
FMUD	Floating Multiply Double Precision.
HAR	Hardware Test Programs
I/O	Input/Output.
IOX	Input/Output Execute.
IR_{least}	Least Significant Bit of Instruction Register.
LBP	Breakpoint Register.
LBQ	Breakpoint Register.
LL	Lower Limit.

MA	Memory Address
MAC	Machine Assembly Code.
MD	Memory Data
MILL	(As in ARITHMETIC MILL.) Instruction Processing.
MPY	Multiply.
NC	Not Connected.
ND	Norsk Data, A.S
P	(As in RESULT P.) Result from A-rack.
PC	A communication register. (Program address ???)
PD	Program Description.
PISO	(As in PISO REG.) Parallel input, serial output
POS	Position
RAM	Random Access Memory.
RT	(As in "RT common area".) Real-time.
SA	A communication register. Start address.
SD	A communication register.
SINTRAN	Name of operating system.
STB	Standby. (As in "Standby Voltage STB".)
SUM	(As in "SUM-bus".) Result bus.

TA	A communication register. The last memory reference address. Last memory address.
TD	A communication register. The last data to or from memory.
UL	Upper Limit.
V	Volt.
VS	(As in SINTRAN III/VS.) Virtual Storage.

Abbreviations

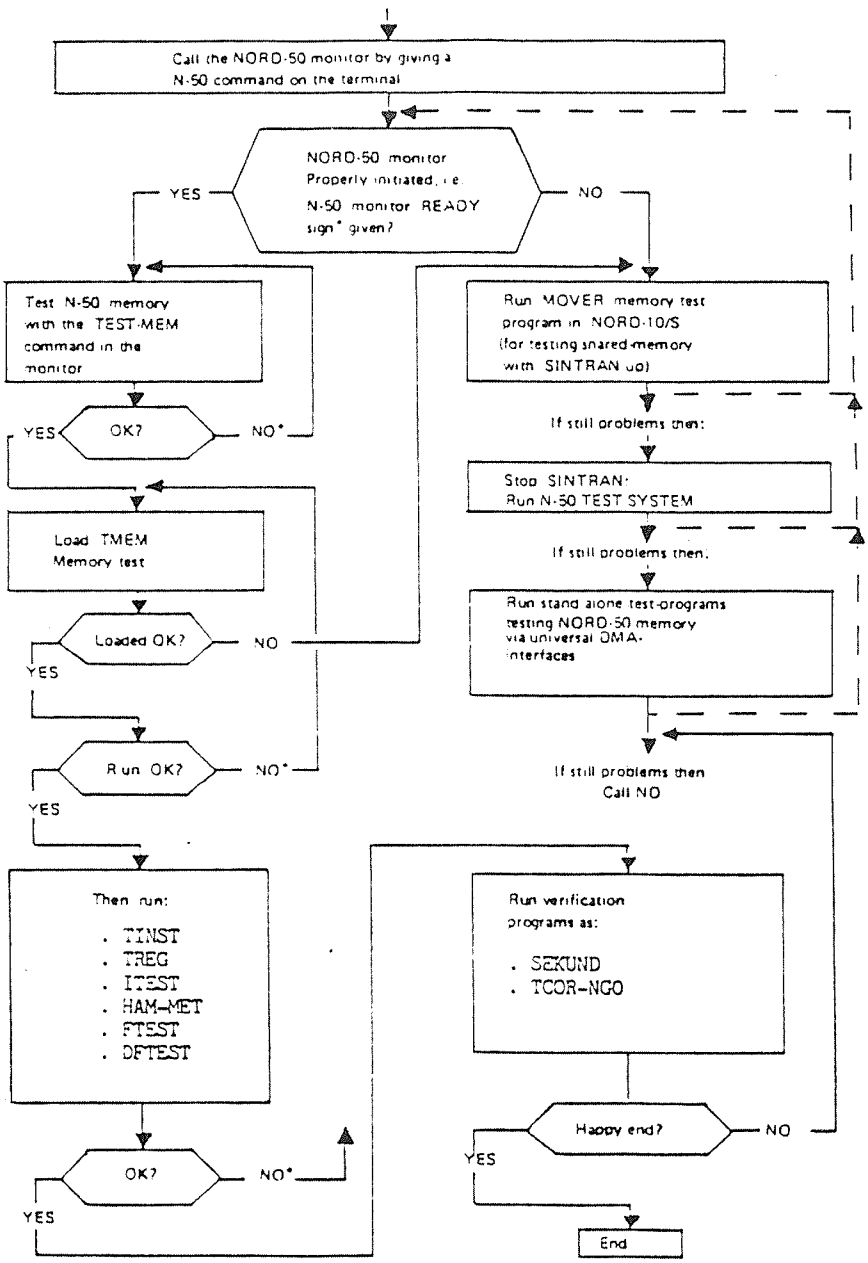
%	"That means."
ADD/SUB	Add/Subtract
ADR	Address (Also ADDR.)
arith	arithmetic (Also ARITHM., ARIT)
AUX	auxiliary
BD	Bus Data
C	Centigrade
CA	Calculated Address
cm	centimeter
ext.	extension
FIXD	fixed
FLOD	floating
Hz	Hertz
instr.	instruction (also INSTR.)
Kg	Kilogram
Kw	Kilowatt

MA	Memory Address
MAX.	maximum
MD	Memory Data Bus
MUL/DIV	Multiply/Divide
Mw	(As in "MAX 2 Mw if ADDR. EXT.") Megaword
PC	Program Counter
PHYS	(As in PHYS REG N-50.) Physical
REF	Reference
REG	(Also reg.) Register.
SD	Simulated Data Register
SW	(As in SE10, SW9)
SWL	(As in SWL10, SWL9,. . . SWL2, SWL 1)
W	watt
W	(As in W15, W14, . . .W1, W0) LED Indicator
WL	(As in WL5, WL4, . . . WL1, WL0) Operator panel light number indicator

Appendix A
GLOSSARY

A P P E N D I X B

NORD-50 TEST SEQUENCE

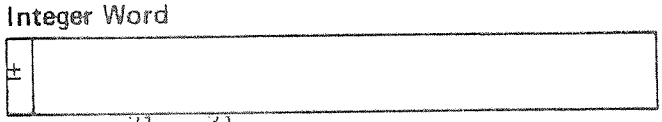


NO* = CALL ND or CHANGE FAILING MODULE IF NEW MODULE AVAILABLE AND REPEAT THE TEST

A P P E N D I X C

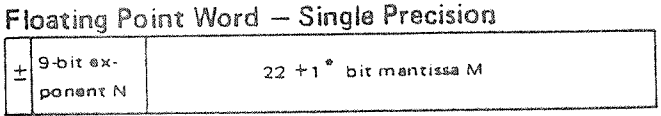
DATA FORMAT SUMMARY

Integer Word



Range: -2^{31} to $2^{31}-1$

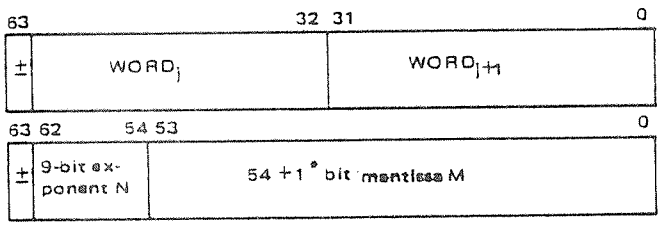
Floating Point Word — Single Precision



Range: 10^{-76} to 10^{76} = 7 decimal digits.
Bit 31 is sign of mantissa. Bias is 400_8.

* The extra bit in the mantissa is the most significant bit and is set to one unless all bits in the exponent are zero. The bit is only used in the arithmetic and is removed in the result.

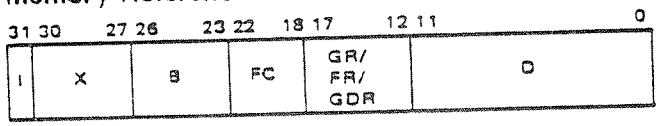
Floating Point Word — Double Precision



Range: 10^{-76} to 10^{76} = 16.3 decimal digits.
Bit 63 is sign of mantissa. Bias is 400_8.

* The extra bit in the mantissa is the most significant bit and is set to one unless all bits in the exponent are zero. The bit is only used in the arithmetic and is removed in the result.

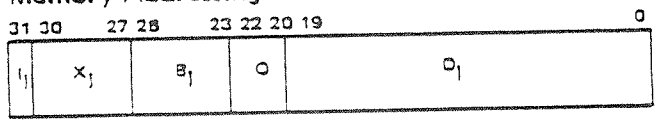
Memory Reference Instruction Word



where

I	–	Indirect addressing flag
X	–	Index register – 1 of 16
B	–	Base register – 1 of 16
FC	–	Function code – 1 of 32 instructions
GR	–	General register number
FR	–	Single precision, floating register number
FDR	–	Double precision, floating register number
D	–	Displacement $0 \leqslant D < 4096$

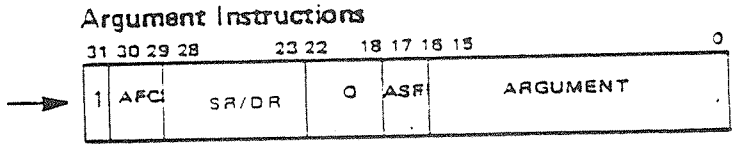
Memory Addressing



Indirect address word – IAW_j at level j.

I_j	–	Indirect addressing flag at level j
X_j	–	Index register for IAW_j
B_j	–	Base register for IAW_j
D_j	–	Displacement for IAW_j $0 \leqslant D < 1048576$

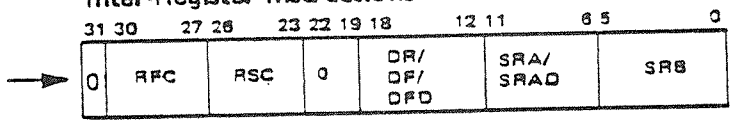
Argument Instructions



where

AFC/ ASF	–	Argument function and subfunction code – direct logical/arithmetical operations, direct test and direct skip
SR/DR	–	Source register/Destination register

Inter-Register Instructions

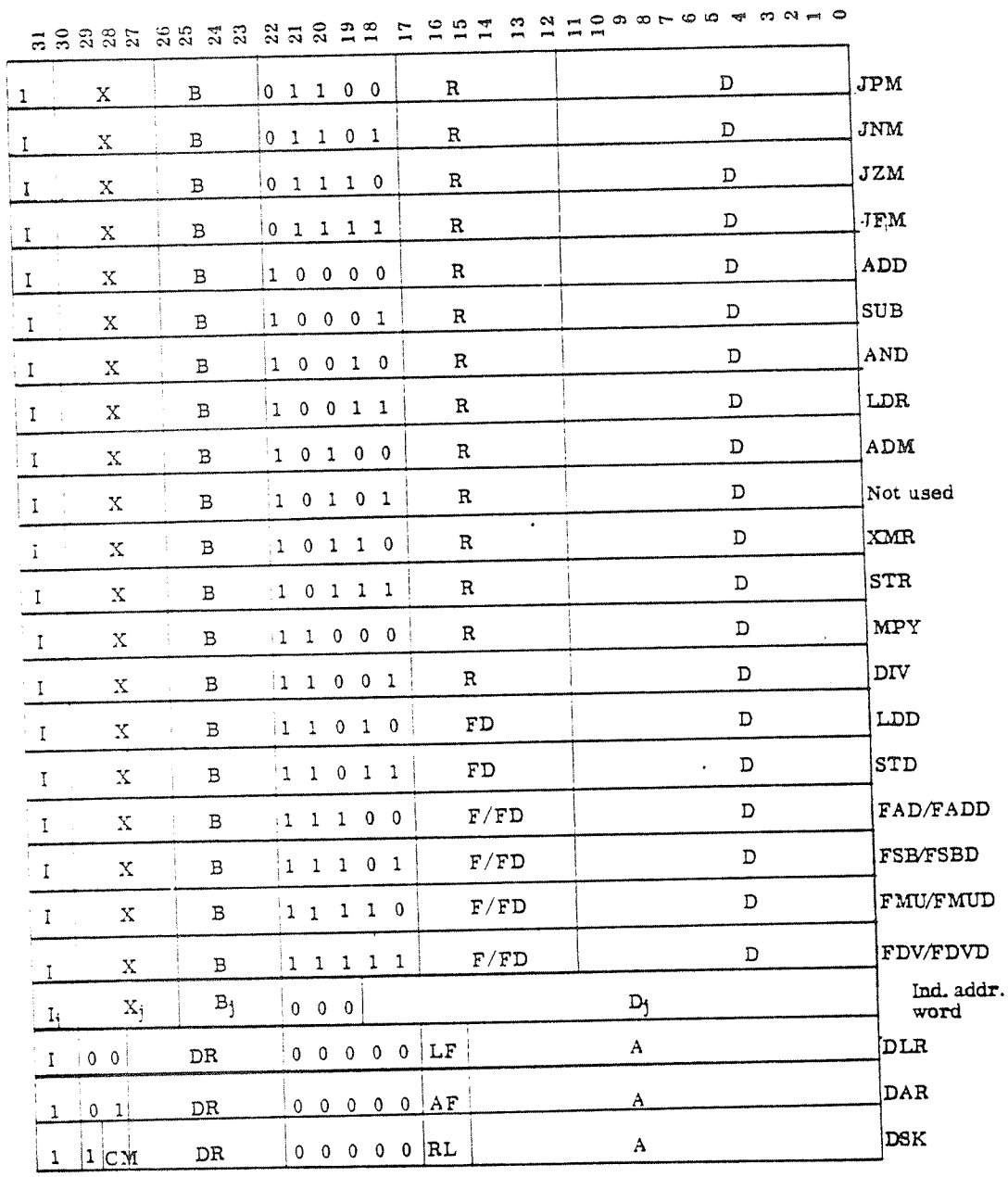


where

RFC/ RSC	–	Inter-register instruction and subfunction code shift, arithmetic, test and skip, logical operations, miscelleneous operations
DR	–	Destination register
DF	–	Destination floating register
DFD	–	Destination double precision floating register
SRA	–	Source register A
SRAD	–	Double precision source register A
SRB	–	Source register B

31 30	27 26	23 22	18 17	12,11	6,5	0	
0	0 0 0 0	0 0 0 0	0 0 0 0 0	MONITOR CODE			STOP (monitor call)
0	0 0 0 1	0 0 0 W	0 0 0 0 0	DR	SRA	EXT. REG. NO.	RIO
0	0 0 1 0	L R SM	0 0 0 0 0	DR	SRA	SHIFT COUNT	SHR
0	0 0 1 1	L R SM	0 0 0 0 0	DFD	SRAD	SHIFT COUNT	SHD
0	0 1 0 0	1 1 0 C	0 0 0 0 0	DR	SRA	BITNO	BST, BCM
0	0 1 0 1	1 1 0 0	0 0 0 0 0	DR	SRA	BITNO	BCL
0	0 1 1 0	1 1 0 0	0 0 0 0 0	DR	SRA	BITNO	BSZ
0	0 1 1 1	1 1 0 0	0 0 0 0 0	DR	SRA	BITNO	BSO
0	1 0 0 0	R 0 0 0	0 0 0 0 0	DR	SRA/SRAD	0 0 0 0 0 0	FIX, RIR FIXD, FIRD
0	1 0 0 1	0 0 1 0	0 0 0 0 0	DF/DFD	SRA	0 0 0 0 0 0	FLO, FLOD
0	1 0 1 0	LO	0 0 0 0 0	DR	SRA	SRB	LRO
0	1 0 1 1		0 0 0 0 0				Not used
0	1 1 0 0	AF	0 0 0 0 0	DR	SRA	SRB	IRO
0	1 1 0 1	0 0 AF	0 0 0 0 0	DF/DFD	SRA/SRAD	SRB/SRBD	FRO
0	1 1 1 0	RL AF	0 0 0 0 0	DR	SRA	SRB	IRS
0	1 1 1 1	RL AF	0 0 0 0 0	DF/DFD	SRA/SRAD	SRB/SRBD	FRS
I	X	B	0 0 0 0 1	R	D		RTJ
I	X	B	0 0 0 1 0	R	D		EXC
I	X	B	0 0 0 1 1	R	D		MIN
I	X	B	0 0 1 0 0	R	D		CRG
I	X	B	0 0 1 0 1	R	D		CRL
I	X	B	0 0 1 1 0	R	D		CRE
I	X	B	0 0 1 1 1	R	D		CRD
I	X	B	0 1 0 0 0	R	D		JRP
I	X	B	0 1 0 0 1	R	D		JRN
I	X	B	0 1 0 1 0	R	D		JRZ
I	X	B	0 1 0 1 1	R	D		JRF

Appendix C
DATA FORMAT SUMMARY



A P P E N D I X D

NORD-50 STATUS REGISTER FORMAT

Appendix D
NORD-50 STATUS REGISTER FORMAT

Status Register Bit Number	Meaning	
0	Program STOP	
1	Address Violation	
2	Instruction hang-up	
3	Overflow	
4	Underflow	
5	Memory Request	Only valid when running in simulated memory mode (used by the TEST SYSTEM).
6	Memory WRITE/READ	
7	Parity Error	
8	Memory hang-up	

A P P E N D I X E

NORD - 50 INSTRUCTIONS

Appendix E
NORD - 50 INSTRUCTIONS

The NORD-50 Assembler is documented in a separate manual (ND-60.075.01). The NORD-50 Reference Manual (ND-05.003.01) may also be helpful when writing assembly programs.

The same format is used in the assembler in the Nord-50-MONITOR.

A summary of all instructions, with the relevant operands for each of them follows.

Summary of Instructions

Memory Reference Instructions:

Mnemonic:		Action:
RTJ	R, D, B, X, I	Return jump
EXC	R, D, B, X, I	Remote execute
MIN	R, D, B, X, I	Memory increment
CRG	R, D, B, X, I	Skip if (R) ⩾ (Ea)
CRL	R, D, B, X, I	Skip if (R) < (Ea)
CRE	R, D, B, X, I	Skip if (R) = (Ea)
CRD	R, D, B, X, I	Skip if (R) ≠ (Ea)
JRP	R, D, B, X, I	Jump if (R) ⩾ 0
JRN	R, D, B, X, I	Jump if (R) < 0
JRZ	R, D, B, X, I	Jump if (R) = 0
JRF	R, D, B, X, I	Jump if (R) ≠ 0
JPM	R, D, B, X, I	Modify (R) and jump if (R) ⩾ 0
JNM	R, D, B, X, I	Modify (R) and jump if (R) < 0
JZM	R, D, B, X, I	Modify (R) and jump if (R) = 0
JFM	R, D, B, X, I	Modify (R) and jump if (R) ≠ 0
ADD	R, D, B, X, I	Add (Ea) to (R)
SUB	R, D, B, X, I	Subtract (Ea) from (R)
AND	R, D, B, X, I	Logical AND between (Ea) and (R)
LDR	R, D, B, X, I	Load (R) with (Ea)
ADM	R, D, B, X, I	Add (R) to (Ea)
XMR	R, D, B, X, I	Exchange (Ea) and (R)
STR	R, D, B, X, I	Store (R) in (Ea)
MPY	R, D, B, X, I	Multiply (R) by (Ea)
DIV	R, D, B, X, I	Divide (R) by (Ea)
LDD	R, D, B, X, I	Load (FD) with (Ea, Ea + 1)
FTD	R, D, B, X, I	Store (FD) in (Ea, Ea + 1)
FAD	R, D, B, X, I	Add (Ea) to (F)
FADD	R, D, B, X, I	Add (Ea, Ea + 1) to (FD)
FSB	R, D, B, X, I	Subtract (Ea) from (F)
FSBD	R, D, B, X, I	Subtract (Ea, Ea + 1) from (FD)
FMU	R, D, B, X, I	Multiply (F) by (Ea)
FMUD	R, D, B, X, I	Multiply (FD) by (Ea, Ea + 1)
FDV	R, D, B, X, I	Divide (F) by (Ea)
FDVD	R, D, B, X, I	Divide (FD) by (Ea, Ea + 1)

R = register
D = displacement
B = base register
X = index register
I = indirect addressing

Memory Reference Instructions Format:

	INSTR	REG NR	DISPLACEMENT	B	X	I
ASSEMBLER:	$0-63_{10}$	0-4 K	0-15	0-15	0 or 1	
MONITOR :	$0-77_{8}$	0-4 K	0-17	0-17	0 or 1	

NORD-50 address calculation:

Ca = (X) + (B) + D

Calculated address	= Content of reg. 0-15 X-field	+	Content of reg. 0-15 B-field	+ Displacement

Inter Register Operations

Shift Instructions:

Mnemonic:		Action:
SLR	DR, SR, SC	Left rotational shift
SRR	DR, SR, SC	Right rotational shift
SLA	DR, SR, SC	Left arithmetical shift
SRA	DR, SR, SC	Right arithmetical shift
SLL	DR, SR, SC	Left logical shift
SRL	DR, SR, SC	Right logical shift
SLRD	DR, SR, SC	Left rotational double register shift
SRRD	DR, SR, SC	Right rotational double register shift
SLAD	DR, SR, SC	Left arithmetical double register shift
SRAD	DR, SR, SC	Right arithmetical double register
SLLD	DR, SR, SC	Left logical double register shift
SRLD	DR, SR, SC	Right logical double register shift

DR = destination register
SR = source register
SC = shift count
$0 \leqslant SC \leqslant 31$ for single register operations
$0 \leqslant SC \leqslant 63$ for double register operations

Miscellaneous Operations:

Mnemonic:		Action:
BST	DR, SR, BN	Bit set
BCM	DR, SR, BN	Bit complement
BCL	DR, SR, BN	Bit clear
BSZ	DR, SR, BN	Bit skip on zero
BSO	DR, SR, BN	Bit skip in one
FIX	DR, SR	Convert floating to integer
FIR	DR, SR	Convert floating to rounded integer
FIXD	DR, SR	Convert double precision floating to integer
FIRD	DR, SR	Convert double precision floating to rounded integer
FLO	DR, SR	Convert integer to floating
FLOD	DR, SR	Convert integer to double precision floating
RIN	DR,, ER	Register input
ROUT	,SR, ER	Register output

BN = bit number
$0 \leq BN \leq 31$
ER = external register
2 — OR (overflow register)
3 — RR (remainder register)

Arithmetic Operations:

Mnemonic:		Action:
RAD	DR, SRA, SRB	Register add
RSB	DR, SRA, SRB	Register subtract
RMU	DR, SRA, SRB	Register multiply
RDV	DR, SRA, SRB	Register divide
RAF	DR, SRA, SRB	Floating register add
RSF	DR, SRA, SRB	Floating register subtract
RMF	DR, SRA, SRB	Floating register multiply
RDF	DR, SRA, SRB	Floating register divide
RAFD	DR, SRA, SRB	Double precision floating register add
RSFD	DR, SRA, SRB	Double precision floating register subtract
RMFD	DR, SRA, SRB	Double precision floating register multiply
RDFD	DR, SRA, SRB	Double precision floating register divide
RAS	DR, SRA, SRB	Register add set carry
RAA	DR, SRA, SRB	Register add add carry
RASA	DR, SRA, SRB	Register add add and set carry
RSS	DR, SRA, SRB	Register subtract set carry
RSA	DR, SRA, SRB	Register subtract add carry
RSSA	DR, SRA, SRB	Register subtract add and set carry

SRA = source register A
SRB = source register B

Test and Skip:

Mnemonic:		Action:
SGR	DR, SRA, SRB	Subtract registers and skip if result ⩾ 0
ASG	DR, SRA, SRB	Add registers and skip if result ⩾ 0
SLE	DR, SRA, SRB	Subtract registers and skip if result < 0
ASL	DR, SRA, SRB	Add registers and skip if result < 0
SEQ	DR, SRA, SRB	Subtract registers and skip if result = 0
ASE	DR, SRA, SRB	Add registers and skip if result = 0
SUE	DR, SRA, SRB	Subtract registers and skip if result ≠ 0
ASU	DR, SRA, SRB	Add registers and skip if result ≠ 0
SGF	DR, SRA, SRB	Subtract floating registers and skip if result ⩾ 0
ASGF	DR, SRA, SRB	Add floating registers and skip if result ⩾ 0
SLF	DR, SRA, SRB	Subtract floating registers and skip if result < 0
ASLF	DR, SRA, SRB	Add floating registers and skip if result < 0
SEF	DR, SRA, SRB	Subtract floating registers and skip if result = 0
ASEF	DR, SRA, SRB	Add floating registers and skip if result ≠ 0
SUF	DR, SRA, SRB	Subtract floating registers and skip if result ≠ 0
ASUF	DR, SRA, SRB	Add floating registers and skip if result ≠ 0
SGD	DR, SRA, SRB	Subtract double precision and skip if result ⩾ 0
AGFD	DR, SRA, SRB	Add double precision and skip if result ⩾ 0
SLD	DR, SRA, SRB	Subtract double precision and skip if result < 0
ALFD	DR, SRA, SRB	Add double precision and skip if result < 0
SED	DR, SRA, SRB	Subtract double precision and skip if result = 0
AEFD	DR, SRA, SRB	Add double precision and skip if result = 0
SUD	DR, SRA, SRB	Subtract double precision and skip if result ≠ 0
AUFD	DR, SRA, SRB	Add double precision and skip if result ≠ 0

Logical Operations:

Mnemonic:		Action:
RND	DR, SRA, SRB	Register AND
RNDA	DR, SRA, SRB	Register AND, use complement of (SRA)
RNDB	DR, SRA, SRB	Register AND, use complement of (SRB)
RXO	DR, SRA, SRB	Register exclusive OR
RXOA	DR, SRA, SRB	Register exclusive OR, use complement of (SRA)
RXOB	DR, SRA, SRB	Register exclusive OR, use complement or (SRB)
ROR	DR, SRA, SRB	Register OR
RORA	DR, SRA, SRB	Register OR, use complement of (SRA)
RORB	DR, SRA, SRB	Register OR, use complement of (SRB)
SZR	DR, SRA, SRB	Set all zeros
RNAB	DR, SRA, SRB	Register AND, use complement of SRA and SRB
ROAB	DR, SRA, SRB	Register OR, use complement of SRA and SRB
RXAB	DR, SRA, SRB	Register exclusive OR, use complement of SRA and SRB

Argument Instructions:

Mnemonic:		Action:
XORA	DR, ARG	Exclusive OR
ANDA	DR, ARG	AND
ORA	DR, ARG	OR
SETA	DR, ARG	Set register
SECA	DR, ARG	Set register to complement
ADDA	DR, ARG	Add
ADCA	DR, ARG	Add complement
DDP	DR, ARG	Skip if (DR) $\geq$ ARG
DDN	DR, ARG	Skip if (DR) $<$ ARG
DDZ	DR, ARG	Skip if (DR) $=$ ARG
DDF	DR, ARG	Skip if (DR) $\neq$ ARG
DSP	DR, ARG	Skip if (DR) $\geq -$ARG
DSN	DR, ARG	Skip if (DR) $<$ $-$ARG
DSZ	DR, ARG	Skip if (DR) $=$ $-$ARG
DSF	DR, ARG	Skip if (DR) $\neq -$ARG

ARG = argument

A P P E N D I X F

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
                                                        MAIN  START
                                0000000001      ACC     EQU   1
                                0000000002      WORK    EQU   2
                                0000000003      RES     EQU   3
                                0000000004      LINK    EQU   4

0000 25252525253                                MIPOS   GCN   =!12525252525
0001 12525252525                                POS     GCN   !12525252525
0002 25252525252                                NEG     GCN   !25252525252
0003 0000000000                                 ZRO     GCN   0
0004 37777777777                                ONES    GCN   =1
0005 0000000001                                 ONE     GCN   1
0006 0000000000                                 STORE   GCN   0
0007 37777777774                                INCR    GCN   =4
0010 0000100 0217 2 00 00 01 00   0217        REMOT   RTJ   0,EXEC+2
0011 0000100000                                 CORTJ   GCN   !100000
                                                        FORM  16,16
0012 0000177777                 0000177777      HALF1   FDAT  0,!177777
0013 37777600000                0000000000      HALF2   FDAT  !177777,0


0014 0002700 0022 2 00 00 27 01   0022        ERROR   STR   ACC,DUMP
0015 0002702 0023 2 00 00 27 02   0023                STR   WORK,DUMP+1
0016 0002703 0024 2 00 00 27 03   0024                STR   RES,DUMP+2
0017 0002704 0025 2 00 00 27 04   0025                STR   LINK,DUMP+3
0020 0002721 0026 2 00 00 27 21   0026                STR   ACC+!20,DUMP+4
0021 0000000000 2 20 00 00 00 000000                  STOP  0
0022 0000000000                                 DUMP    GCN   0
0023 0000000000                                         GCN   0
0024 0000000000                                         GCN   0
0025 0000000000                                         GCN   0
0026 0000000000                                         GCN   0


0027 0000100 0031 2 00 00 01 00   0031        START   RTJ   0,**2
0030 0000000000 2 00 00 00 00 000000                  STOP  0
0031 0000102 0032 2 00 00 01 02   0032        CHRTJ   RTJ   WORK,**1
0032 0000602 1026 2 00 00 06 02   1026                CRE   WORK,=CHRTJ+1
0033 0000000000 2 00 00 00 00 000000                  STOP  0
0034 0002301 0003 2 00 00 23 01   0003                LDR   ACC,ZRO
0035 0001201 0037 2 00 00 12 01   0037                JRZ   ACC,**2
0036 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        JRZ NO JUMP ACC=0
0037 0001031 0041 2 00 00 13 01   0041                JRP   ACC,**2
0040 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        JRP NO JUMP ACC=0
0041 0001301 0043 2 00 00 13 01   0043                JRF   ACC,**2.
0042 0000100 0044 2 00 00 01 00   0044                RTJ   0,**2
0043 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        JRF JUMP ACC=0
0044 0001131 0046 2 00 00 11 01   0046                JRN   ACC,**2
0045 0000100 0047 2 00 00 01 00   0047                RTJ   0,**2
0046 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        JRN JUMP ACC=0
0047 0000701 0005 2 00 00 07 01   0005                CRD   ACC,ONE
0050 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        CRD NO SKIP ACC<>(E
0051 0000601 0005 2 00 00 06 01   0005                CRE   ACC,ONE
0052 0000100 0054 2 00 00 01 20   0054                RTJ   0,**2
0053 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        CRE SKIP ACC<>(EA)
0054 0000501 0005 2 00 00 05 01   0005                CRL   ACC,ONE
0055 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        CRL NO SKIP ACC<(EA)
0056 0000401 0004 2 00 00 04 01   0004                CRG   ACC,ONES
0057 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        CRG NO SKIP ACC>(EA)
0060 0002301 0001 2 00 00 23 01   0001                LDR   ACC,POS
0061 0001001 0063 2 00 00 10 01   0063                JRP   ACC,**2
0062 0000134 0014 2 00 00 01 04   0014                RTJ   LINK,ERROR        JRP NO JUMP ACC>0
0063 0001201 0065 2 00 00 12 01   0065                JRZ   ACC,**2
```

```
00064 0000010000066 0 00 00 01 00   0066    RTJ   0,**2
00065 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRZ JUMP ACC>0
00066 0000013000070 0 00 00 13 01   0070    JRF   ACC,**2
00067 0000010400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRF NO JUMP ACC>0
00070 0000011000072 0 00 00 11 01   0072    JRN   ACC,**2
00071 0000010000073 0 00 00 01 00   0073    RTJ   0,**2
00072 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRN JUMP ACC>0
00073 0000006010001 0 00 00 06 01   0001    CRE   ACC,POS
00074 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRE NO SKIP ACC=+(EA)
00075 0000004010001 0 00 00 04 01   0001    CRG   ACC,POS
00076 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRG NO SKIP ACC=(EA)
00077 0000005010005 0 00 00 05 01   0005    CRL   ACC,ONE
00100 0000010000102 0 00 00 01 00   0102    RTJ   0,**2
00101 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRL SKIP ACC>(EA)
00102 0000023010002 0 00 00 23 01   0002    LDR   ACC,NEG
00103 0000011010105 0 00 00 11 01   0105    JRN   ACC,**2
00104 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRN NO JUMP ACC<0
00105 0000013010107 0 00 00 13 01   0107    JRF   ACC,**2
00106 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRF NO JUMP ACC<0
00107 0000010010111 0 00 00 10 01   0111    JRP   ACC,**2
00110 0000010000112 0 00 00 01 00   0112    RTJ   0,**2
00111 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRP JUMP ACC<0
00112 0000012010114 0 00 00 12 01   0114    JRZ   ACC,**2
00113 0000010000115 0 00 00 01 00   0115    RTJ   0,**2
00114 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          JRZ JUMP ACC<0
00115 0000005010004 0 00 00 05 01   0004    CRL   ACC,ONES
00116 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRL NO SKIP ACC<(EA)
00117 0000007010002 0 00 00 07 01   0002    CRD   ACC,NEG
00120 0000010000122 0 00 00 01 00   0122    RTJ   0,**2
00121 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRD SKIP ACC=(EA)
00122 0000004010005 0 00 00 04 01   0005    CRG   ACC,ONE
00123 0000010000125 0 00 00 01 00   0125    RTJ   0,**2
00124 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          CRG SKIP ACC<(EA)
00125 0000023010004 0 00 00 23 01   0004    LDR   ACC,ONES
00126 0000027010006 0 00 00 27 01   0006    STR   ACC,STORE
00127 0000006010006 0 00 00 06 01   0006    CRE   ACC,STORE
00130 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          STR ERROR
00131 0000023010001 0 00 00 23 01   0001    LDR   ACC,POS
00132 0000027010006 0 00 00 27 01   0006    STR   ACC,STORE
00133 0000006010006 0 00 00 06 01   0006    CRE   ACC,STORE
00134 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          STR ERROR
00135 0000023010002 0 00 00 23 01   0002    LDR   ACC,NEG
00136 0000027010006 0 00 00 27 01   0006    STR   ACC,STORE
00137 0000006010006 0 00 00 06 01   0006    CRE   ACC,STORE
00140 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          STR ERROR
00141 0000023010003 0 00 00 23 01   0003    LDR   ACC,ZRO
00142 0000027010006 0 00 00 27 01   0006    STR   ACC,STORE
00143 0000006010006 0 00 00 06 01   0006    CRE   ACC,STORE
00144 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          STR ERROR
00145 0000023010005 0 00 00 23 01   0005    LDR   ACC,ONE
00146 0000020010004 0 00 00 20 01   0004    ADD   ACC,ONES
00147 0000012010151 0 00 00 12 01   0151    JRZ   ACC,**2
00150 0000010400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          ADD ERROR 1+(-1)
00151 0000023010001 0 00 00 23 01   0001    LDR   ACC,POS
00152 0000021010001 0 00 00 21 01   0001    SUB   ACC,POS
00153 0000012010155 0 00 00 12 01   0155    JRZ   ACC,**2
00154 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          SUB ERROR POS-POS
00155 0000023010001 0 00 00 23 01   0001    LDR   ACC,POS
00156 0000022010001 0 00 00 22 01   0001    AND   ACC,POS
00157 0000006010001 0 00 00 06 01   0001    CRE   ACC,POS
00160 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          AND ERROR POS AND POS
00161 0000023010002 0 00 00 23 01   0002    LDR   ACC,NEG
00162 0000022010001 0 00 00 22 01   0001    AND   ACC,POS
00163 0000012010165 0 00 00 12 01   0165    JRZ   ACC,**2
00164 0000013400014 0 00 00 01 04   0014    RTJ   LINK,ERROR          AND ERROR POS AND NEG
00165 0000023010005 0 00 00 23 01   0005    LDR   ACC,ONE
00166 0000023020003 0 00 00 23 02   0003    LDR   WORK,ZRO
00167 0000024010004 0 00 00 24 01   0004    ADM   ACC,ONES
```

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
00270 0000623020304 0 00 00 06 02    0004          CRE   WORK,ONES
00271 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          ADM ERROR 1+(-1)
00272 0002300011027 0 00 00 23 01    1027          LDR   ACC,=-1
00273 0000273100004 0 00 00 27 01    0004          STR   ACC,ONES
00274 0000230100003 0 00 00 23 01    0003          LDR   ACC,ZRO
00275 0000270100007 0 00 00 27 01    0007          STR   ACC,INCR
00276 0000230110030 0 00 00 23 01    1030          LDR   ACC,=-1
00277 0000230210031 0 00 00 23 02    1031          LDR   WORK,=-1
00300 0000601100007 0 00 00 06 01    0007          XMR   ACC,INCR
00301 0000120100203 0 00 00 12 01    0203          JRZ   ACC,*+2
00302 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          ;XMR ERROR REG FAILED
00303 0000600200007 0 00 00 06 02    0007          CRE   WORK,INCR
00304 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          XMR ERROR (EA) FAILED
00305 0002300011032 0 00 00 23 01    1032          LDR   ACC,=-4
00306 0000270100007 0 00 00 27 01    0007          STR   ACC,INCR
00307 0000230100005 0 00 00 23 01    0005          LDR   ACC,ONE
00310 0000300100007 0 00 00 03 01    0007          MIN   ACC,INCR
00311 0000100000213 0 00 00 01 00    0213          RTJ   0,*+2
00312 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          MIN SKIPPED (EA)=-2
00313 0000300100007 0 00 00 03 01    0007          MIN   ACC,INCR
00314 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          MIN NO SKIP (EA)=0
00315 0000200000010 0 00 00 02 00    0010  EXEC    EXC   0,REMOT
00316 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR
00317 0002300100011 0 00 00 23 01    0011          LDR   ACC,CORTJ
00320 0000420100222 0 00 01 02 01    0222          EXC   1,*+2,ACC
00321 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR
00322 0002321100004 0 00 00 23 21    0004          LDR   ACC*!20,ONES
00323 0002300100005 0 00 00 23 01    0005          LDR   ACC,ONE
00324 0001501100226 0 00 00 15 01    0226          JNM   ACC,*+2
00325 0000100000227 0 00 00 01 00    0227          RTJ   0,*+2
00326 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JNM JUMP ACC=0
00327 0001501100231 0 00 00 15 01    0231          JNM   ACC,*+2
00330 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JNM NO JUMP ACC<0
00331 0001601100233 0 00 00 16 01    0233          JZM   ACC,*+2
00332 0000100000234 0 00 00 01 00    0234          RTJ   0,*+2
00333 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JZM JUMP ACC<0
00334 0002300100005 0 00 00 23 01    0005          ADD   ACC,ONE
00335 0002321100005 0 00 00 23 21    0005          LDR   ACC*!20,ONE
00336 0001601100240 0 00 00 16 01    0240          JZM   ACC,*+2
00337 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JZM NO JUMP ACC=0
00340 0001601100242 0 00 00 16 01    0242          JZM   ACC,*+2
00341 0000100000243 0 00 00 01 00    0243          RTJ   0,*+2
00342 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JZM JUMP ACC>0
00343 0001501100245 0 00 00 15 01    0245          JNM   ACC,*+2
00344 0000100000246 0 00 00 01 00    0246          RTJ   0,*+2
00345 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JNM JUMP ACC>0
00346 0001401100250 0 00 00 14 01    0250          JPM   ACC,*+2
00347 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JPM NO JUMP ACC>0
00350 0002101100033 0 00 00 21 01    1033          SUB   ACC,=2
00351 0002321100004 0 00 00 23 21    0004          LDR   ACC*!20,ONES
00352 0001401100254 0 00 00 14 01    0254          JPM   ACC,*+2
00353 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JPM NO JUMP ACC=0
00354 0001401100256 0 00 00 14 01    0256          JPM   ACC,*+2
00355 0000100000257 0 00 00 01 00    0257          RTJ   0,*+2
00356 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JPM       JUMP ACC<0
00357 0001701100261 0 00 00 17 01    0261          JFM   ACC,*+2
00360 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JFM NO JUMP ACC<0
00361 0002301100005 0 00 00 20 01    0005          ADD   ACC,ONE
00362 0002321100005 0 00 00 23 21    0005          LDR   ACC*!20,ONE
00363 0001701100265 0 00 00 17 01    0265          JFM   ACC,*+2
00364 0000100000266 0 00 00 01 00    0266          RTJ   0,*+2
00365 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JFM JUMP ACC=0
00366 0001701100270 0 00 00 17 01    0270          JFM   ACC,*+2
00367 0000104000014 0 00 00 01 04    0014          RTJ   LINK,ERROR          JFM NO JUMP ACC>0


00270 0002301100001 0 00 00 23 01    0001          LDR   ACC,POS
00271 1400001100001 0 14 00 00 01  00 01           RAD   ACC,0,ACC
```

Appendix F
ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
00272 0000301C021 0 00 00 06 01    0001
00273 0000104C014 0 06 00 01 04    0014
00274 0002301C002 0 00 00 23 01    0002
00275 1440001C101 0 14 01 00 01  01 01
00276 0021200C300 0 00 00 12 00    0300
00277 0000104C014 0 00 00 01 04    0014
00300 0002301C001 0 00 00 23 01    0001
00301 1004000C101 0 16 01 00 00  01 01
00302 0000104C014 0 00 00 01 04    0014
00303 0002300C004 0 00 00 23 02    0004
00304 1004000C302 0 16 01 00 00  00 02
00305 0000104C014 0 00 00 01 04    0014
00306 1004000C001 0 16 01 00 00  00 01
00307 0000100C311 0 00 00 01 00    0311
00310 0000104C014 0 00 00 01 04    0014
00311 1000000C102 0 16 00 00 00  01 02
00312 0002104C014 0 00 00 01 04    0014
00313 0002301C005 0 00 00 23 01    0005
00314 1000000C102 0 16 00 00 00  01 02
00315 0000104C014 0 00 00 01 04    0014
00316 1000000C200 0 16 00 00 00  02 00
00317 0000100C321 0 00 00 01 00    0321
00320 0000104C014 0 00 00 01 04    0014
00321 1024000C002 0 16 05 00 00  00 02
00322 0000100C324 0 00 00 01 00    0324
00323 0000104C014 0 00 00 01 04    0014
00324 1024000C200 0 16 05 00 00  02 00
00325 0000104C014 0 00 00 01 04    0014
00326 1024000C101 0 16 05 00 00  01 01
00327 0000100C331 0 00 00 01 00    0331
00330 0000104C014 0 00 00 01 04    0014
00331 1020000C102 0 16 04 00 00  01 02
00332 0000100C334 0 00 00 01 00    0334
00333 0000104C014 0 00 00 01 04    0014
00334 1020000C001 0 16 04 00 00  00 01
00335 0000100C337 0 00 00 01 00    0337
00336 0000104C014 0 00 00 01 04    0014
00337 1020000C200 0 16 04 00 00  02 00
00340 0000104C014 0 00 00 01 04    0014
00341 0002301C002 0 00 00 23 01    0002
00342 1044000C101 0 16 11 00 00  01 01
00343 0000104C014 0 00 00 01 04    0014
00344 1044000C001 0 16 11 00 00  00 01
00345 0000100C347 0 00 00 01 00    0347
00346 0000104C014 0 00 00 01 04    0014
00347 1044000C100 0 16 11 00 00  01 00
00350 0000100C352 0 00 00 01 00    0352
00351 0000104C014 0 00 00 01 04    0014
00352 0002301C001 0 00 00 23 01    0001
00353 0002302C000 0 00 00 23 02    0000
00354 1040000C102 0 16 10 00 00  01 02
00355 0000104C014 0 00 00 01 04    0014
00356 1040000C100 0 16 10 00 00  01 00
00357 0000100C361 0 00 00 01 00    0361
00360 0000104C014 0 00 00 01 04    0014
00361 1040000C200 0 16 10 00 00  02 00
00362 0000100C364 0 00 00 01 00    0364
00363 0000104C014 0 00 00 01 04    0014
00364 1060000C202 0 16 15 00 00  02 02
00365 0000100C367 0 00 00 01 00    0367
00366 0000104C014 0 00 00 01 04    0014
00367 1060000C200 0 16 15 00 00  02 00
00370 0000104C014 0 00 00 01 04    0014
00371 1060000C100 0 16 15 00 00  01 00
00372 0000104C014 0 00 00 01 04    0014
00373 1060000C102 0 16 14 00 00  01 02
00374 0000100C376 0 00 00 01 00    0376
00375 0000104C014 0 00 00 01 04    0014
```

```
CRE   ACC,POS
RTJ   LINK,ERROR          RAD ERROR POS+0<>POS
LDR   ACC,NEG
RSB   ACC,ACC,ACC
JRZ   0,*+2
RTJ   LINK,ERROR          RSB ERROR NEG=NEG<>0
LDR   ACC,POS
SGR   0,ACC,ACC
RTJ   LINK,ERROR          SGR NO SKIP RESULT=0
LDR   WORK,ONES
SGR   0,0,WORK
RTJ   LINK,ERROR          SGR NO SKIP RESULT>0
SGR   0,0,ACC
RTJ   0,*+2
RTJ   LINK,ERROR          SGR SKIP RESULT<0
ASG   0,ACC,WORK
RTJ   LINK,ERROR          ASG NO SKIP RESULT>0
LDR   ACC,ONE
ASG   0,ACC,WORK
RTJ   LINK,ERROR          ASG NO SKIP RESULT=0
ASG   0,WORK,0
RTJ   0,*+2
RTJ   LINK,ERROR          ASG SKIP RESULT<0
SLE   0,0,WORK
RTJ   0,*+2
RTJ   LINK,ERROR          SLE SKIP RESULT>0
SLE   0,WORK,0
RTJ   LINK,ERROR          SLE NO SKIP RESULT<0
SLE   0,ACC,ACC
RTJ   0,*+2
RTJ   LINK,ERROR          SLE SKIP RESULT=0
ASL   0,ACC,WORK
RTJ   0,*+2
RTJ   LINK,ERROR          ASL SKIP RESULT=0
ASL   0,0,ACC
RTJ   0,*+2
RTJ   LINK,ERROR          ASL SKIP RESULT>0
ASL   0,WORK,0
RTJ   LINK,ERROR          ASL NO SKIP RESULT<0
LDR   ACC,NEG
SEQ   0,ACC,ACC
RTJ   LINK,ERROR          SEQ NO SKIP RESULT=0
SEQ   0,0,ACC
RTJ   0,*+2
RTJ   LINK,ERROR          SEQ SKIP RESULT>0
SEQ   0,ACC,0
RTJ   0,*+2
RTJ   LINK,ERROR          SEQ SKIP RESULT<0
LDR   ACC,POS
LDR   WORK,MIPOS
ASE   0,ACC,WORK
RTJ   LINK,ERROR          ASE NO SKIP RESULT=0
ASE   0,ACC,0
RTJ   0,*+2
RTJ   LINK,ERROR          ASE SKIP RESULT>0
ASE   0,WORK,0
RTJ   0,*+2
RTJ   LINK,ERROR          ASE SKIP RESULT<0
SUE   0,WORK,WORK
RTJ   0,*+2
RTJ   LINK,ERROR          SUE SKIP RESULT=0
SUE   0,WORK,0
RTJ   LINK,ERROR          SUE NO SKIP RESULT<0
SUE   0,ACC,0
RTJ   LINK,ERROR          SUE NO SKIP RESULT>0
ASU   0,ACC,WORK
RTJ   0,*+2
RTJ   LINK,ERROR          ASU SKIP RESULT=0
```

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
00376 1060003001D0 0 16 14 03 00  01 00     ASU   0,ACC,0
00377 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR
00400 1060003D3230 0 16 14 03 00  02 00     ASU   0,#ORK,0
00401 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ASU NO SKIP RESULT<>0
00402 0002301D001  0 00 00 23 01   0001     LDR   ACC,POS
00403 0002302D002  0 00 00 23 02   0002     LDR   #ORK,NEG
00404 1204003D102  0 12 01 03 03  01 02     RND   RES,ACC,#ORK
00405 0003060D003  0 00 00 06 03   0003     CRE   RES,ZRO
00406 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RND ERROR POS AND NEG<>0
00407 1244003D102  0 12 11 00 03  01 02     RNDA  RES,ACC,#ORK
00410 0000603D002  0 00 00 06 03   0002     CRE   RES,NEG
00411 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RNDA ERROR NEG AND POS'<>NEG
00412 1224003D102  0 12 05 00 03  21 02     RND8  RES,ACC,#ORK
00413 0000603D001  0 00 00 06 03   0001     CRE   RES,POS
00414 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RND8 ERROR POS AND NEG'<>POS
00415 1210003D102  0 12 02 00 03  01 02     RXO   RES,ACC,#ORK
00416 0000603D004  0 00 00 06 03   0004     CRE   RES,ONES
00417 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RXO ERROR POS EXOR NEG<>ONES
00420 1210003D101  0 12 02 00 03  01 01     RXO   RES,ACC,ACC
00421 0000603D003  0 00 00 06 03   0003     CRE   RES,ZRO
00422 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RXO ERROR POS EXOR POS<>ZRO
00423 1250003D202  0 12 12 00 03  02 02     RXOA  RES,#ORK,#ORK
00424 0000603D004  0 00 00 06 03   0004     CRE   RES,ONES
00425 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RXOA ERROR NEG' EXOR NEG<>ONES
00426 1230003D102  0 12 06 00 03  21 02     RXO8  RES,ACC,#ORK
00427 0000603D003  0 00 00 06 03   0003     CRE   RES,ZRO
00430 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RXO8 ERROR POS EXOR NEG'<>ZRO
00431 1214003D102  0 12 03 00 03  01 02     ROR   RES,ACC,#ORK
00432 0000603D004  0 00 00 06 03   0004     CRE   RES,ONES
00433 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ROR ERROR POS OR NEG <>ONES
00434 1214003D202  0 12 03 00 03  02 02     ROR   RES,#ORK,#ORK
00435 0000603D002  0 00 00 06 03   0002     CRE   RES,NEG
00436 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ROR ERROR NEG OR NEG <>NEG
00437 1254003D201  0 12 13 00 03  02 01     RORA  RES,#ORK,ACC
00440 0000603D001  0 00 00 06 03   0001     CRE   RES,POS
00441 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RORA ERROR NEG' OR POS<>POS
00442 1234003D201  0 12 07 00 03  02 01     RORB  RES,#ORK,ACC
00443 0000603D002  0 00 00 06 03   0002     CRE   RES,NEG
00444 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          RORB ERROR NEG OR POS'<>NEG
00445 1200001D000  0 12 00 00 01  00 00     SZR   ACC
00446 0000601D003  0 00 00 06 01   0003     CRE   ACC,ZRO
00447 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          SZR ERROR
00450 0002301D001  0 00 00 23 01   0001     LDR   ACC,POS
00451 2004002D000  1 00 01 00 01  000000     XORA  ACC,0
00452 0000601D001  0 00 00 06 01   0001     CRE   ACC,POS
00453 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          XORA ERROR POS EXOR 0<>POS
00454 0002301D002  0 00 00 23 01   0002     LDR   ACC,NEG
00455 2004525252  1 00 02 04 01  125252     ANDA  ACC,'125252
00456 0000601D34  0 00 00 06 01   1034     CRE   ACC,#'125252
00457 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ANDA ERROR NEG AND '125252
00460 0002301D001  0 00 00 23 01   0001     LDR   ACC,POS
00461 2004652525  1 00 03 00 01  052525     ORA   ACC,'52525
00462 0000601D001  0 00 00 06 01   0001     CRE   ACC,POS
00463 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ORA ERROR POS ORA '52525<>POS
00464 2404000D000  1 01 00 00 01  000000     SETA  ACC,0
00465 0000601D003  0 00 00 06 01   0003     CRE   ACC,ZRO
00466 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          SETA ERROR ACC<>0
00467 2404020D001  1 01 01 00 01  000001     SECA  ACC,1
00470 0000601D004  0 00 00 06 01   0004     CRE   ACC,ONES
00471 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          SECA ERROR ACC<>-1
00472 2404040D002  1 01 02 00 01  000002     ADDA  ACC,2
00473 0000601D005  0 00 00 06 01   0005     CRE   ACC,ONE
00474 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ADDA ERROR ACC<>1
00475 2404060D002  1 01 03 00 01  000002     ADCA  ACC,2
00476 0000601D004  0 00 00 06 01   0004     CRE   ACC,ONES
00477 0000104D014  0 00 00 01 04   0014     RTJ   LINK,ERROR          ADCA ERROR ACC<>-1
00500 0002311D035  0 00 00 23 01   1035     LDR   ACC,#'376
00591 3004000D377  1 02 00 00 01  000377     DOP   ACC,'377
```

```
00502 0000123005d4 0 00 00 01 00   0504          RTJ   0,.+2
00503 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDP SKIP ACC<ARG
00504 3000000c0376 1 02 00 00 01 000376          DDP   ACC,!376
00505 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDP NO SKIP ACC=ARG
00506 3000000c0375 1 02 00 00 01 000375          DDP   ACC,!375
00507 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDP NO SKIP ACC>ARG
00510 3000000c0377 1 02 00 00 01 000377          DDN   ACC,!377
00511 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDN NO SKIP ACC<ARG
00512 3000000c0376 1 02 00 00 01 000376          DDN   ACC,!376
00513 0000123d0515 0 00 00 01 00   0515          RTJ   0,.+2
00514 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDN SKP ACC=ARG
00515 3000000c0375 1 02 00 00 01 000375          DDN   ACC,!375
00516 0000123d0520 0 00 00 01 00   0520          RTJ   0,.+2
00517 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDN SKIP ACC>ARG
00520 3000000c0376 1 02 00 00 01 000376          DDZ   ACC,!376
00521 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDZ NO SKIP ACC=ARG
00522 3000000c0377 1 02 00 00 01 000377          DDZ   ACC,!377
00523 0000123d0525 0 00 00 01 00   0525          RTJ   0,.+2
00524 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDZ SKIP ACC<ARG
00525 3000000c0375 1 02 00 00 01 000375          DDZ   ACC,!375
00526 0000123d0530 0 00 00 01 00   0530          RTJ   0,.+2
00527 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDZ SKIP ACC>ARG
00530 3000000c0376 1 02 00 00 01 000376          DDF   ACC,!376
00531 0000123d0533 0 00 00 01 00   0533          RTJ   0,.+2
00532 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDF SKIP ACC=ARG
00533 3000000c0377 1 02 00 00 01 000377          DDF   ACC,!377
00534 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDF NO SKIP ACC<ARG
00535 3000000c0375 1 02 00 00 01 000375          DDF   ACC,!375
00536 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DDF NO SKIP ACC>ARG
00537 0000123d1036 0 00 00 23 01   1036          LDR   ACC,.+!1776
00540 3000000c1776 1 03 00 00 01 001776          DSP   ACC,!1776
00541 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSP NO SKIP ACC=.+ARG
00542 3000000c1777 1 03 00 00 01 001777          DSP   ACC,!1777
00543 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSP NO SKIP ACC>.+ARG
00544 3000000c1775 1 03 00 00 01 001775          DSP   ACC,!1775
00545 0000123d0547 0 00 00 01 00   0547          RTJ   0,.+2
00546 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSP SKIP ACC<.+ARG
00547 3000000c1776 1 03 01 00 01 001776          DSN   ACC,!1776
00550 0000123d0552 0 00 00 01 00   0552          RTJ   0,.+2
00551 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSN SKIP ACC=.+ARG
00552 3000000c1777 1 03 01 00 01 001777          DSN   ACC,!1777
00553 0000123d0555 0 00 00 01 00   0555          RTJ   0,.+2
00554 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSN SKIP ACC>.+ARG
00555 3000000c1775 1 03 01 00 01 001775          DSN   ACC,!1775
00556 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSN NO SKIP ACC<.+ARG
00557 3000000c1776 1 03 02 00 01 001776          DSZ   ACC,!1776
00560 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSZ NO SKIP ACC=.+ARG
00561 3000000c1777 1 03 02 00 01 001777          DSZ   ACC,!1777
00562 0000123d0564 0 00 00 01 00   0564          RTJ   0,.+2
00563 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSZ SKIP ACC>.+ARG
00564 3000000c1775 1 03 02 00 01 001775          DSZ   ACC,!1775
00565 0000123d0567 0 00 00 01 00   0567          RTJ   0,.+2
00566 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSZ SKIP ACC<.+ARG
00567 3000000c1776 1 03 03 00 01 001776          DSF   ACC,!1776
00570 0000123d0572 0 00 00 01 00   0572          RTJ   0,.+2
00571 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSF SKIP ACC=.+ARG
00572 3000000c1777 1 03 03 00 01 001777          DSF   ACC,!1777
00573 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSF NO SKIP ACC>.+ARG
00574 3000000c1775 1 03 03 00 01 001775          DSF   ACC,!1775
00575 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          DSF NO SKIP ACC<.+ARG


00576 0000123d0005 0 00 00 23 01   0005          LDR   ACC,ONE
00577 0000123d0137 0 00 14 03 01   01 37          BSZ   ACC,ACC,31
00600 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          BSZ  NO SKIP BIT=0
00601 0000123d0100 0 00 14 00 01   01 00          BSZ   ACC,ACC,0
00602 0000123d0604 0 00 00 01 00   0604          RTJ   0,.+2
00603 0000123d0014 0 00 00 01 04   0014          RTJ   LINK,ERROR          BSZ SKIP BIT=1
```

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
00604 076003010100 0 07 14 00 01  01 00     BSO   ACC,ACC,0
00605 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BSO NO SKIP BIT=1
00606 076003140137 0 07 14 00 01  01 37     BSO   ACC,ACC,31
00607 0e001000611 0 00 00 01 0C   0611     RTJ   0,+*2
00610 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BSO SKIP BIT=0
00611 0e023011037 0 00 00 23 01   1037     LDR   ACC,=7
00612 046000010103 0 04 14 00 01  01 03     BST   ACC,ACC,3
00613 0e006011040 0 00 00 06 01   1040     CRE   ACC,=15
00614 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BST ERROR
00615 0e023011041 0 00 00 23 01   1041     LDR   ACC,=7
00616 046400010100 0 04 15 00 01  01 00     BCM   ACC,ACC,0
00617 0e006011042 0 00 00 06 01   1042     CRE   ACC,=6
00620 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BCM ERROR
00621 046400010100 0 04 15 00 01  01 00     BCM   ACC,ACC,0
00622 0e006011043 0 00 00 06 01   1043     CRE   ACC,=7
00623 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BCM ERROR
00624 0e023011044 0 00 00 23 01   1044     LDR   ACC,=17777
00625 056000010114 0 05 14 00 01  01 14     BCL   ACC,ACC,12
00626 0e006011045 0 00 00 06 01   1045     CRE   ACC,=17777
00627 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          BCL ERROR

00630 0e023010001 0 00 00 23 01   0001     LDR   ACC,POS
00631 024000101101 0 02 14 00 01  01 01     SLR   ACC,ACC,1
00632 0e006010002 0 00 00 06 01   0002     CRE   ACC,NEG
00633 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SLR ERROR
00634 024000101137 0 02 14 00 01  01 37     SLR   ACC,ACC,31
00635 0e006012001 0 00 00 06 01   0001     CRE   ACC,POS
00636 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SLR ERROR
00637 0e023010012 0 00 00 23 01   0012     LDR   ACC,HALF1
00640 024000101120 0 02 14 00 01  01 20     SLR   ACC,ACC,16
00641 0e006010013 0 00 00 06 01   0013     CRE   ACC,HALF2
00642 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SLR ERROR
00643 0e023010002 0 00 00 23 01   0002     LDR   ACC,NEG
00644 022000101101 0 02 04 00 01  01 01     SRR   ACC,ACC,1
00645 0e006010001 0 00 00 06 01   0001     CRE   ACC,POS
00646 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SRR ERROR
00647 022000101137 0 02 04 00 01  01 37     SRR   ACC,ACC,31
00650 0e006010002 0 00 00 06 01   0002     CRE   ACC,NEG
00651 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SRR ERROR
00652 0e023010013 0 00 00 23 01   0013     LDR   ACC,HALF2
00653 022000101120 0 02 04 00 01  01 20     SRR   ACC,ACC,16
00654 0e006010012 0 00 00 06 01   0012     CRE   ACC,HALF1
00655 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SRR ERROR
00656 0e023010012 0 00 00 23 01   0012     LDR   ACC,HALF1
00657 025000101120 0 02 12 00 01  01 20     SLA   ACC,ACC,16
00660 0e006010013 0 00 00 06 01   0013     CRE   ACC,HALF2
00661 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SLA ERROR
00662 0e023010013 0 00 00 23 01   0013     LDR   ACC,HALF2
00663 023000101120 0 02 06 00 01  01 20     SRA   ACC,ACC,16
00664 0e006010004 0 00 00 06 01   0004     CRE   ACC,ONES
00665 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SRA ERROR
00666 0e023010013 0 00 00 23 01   0013     LDR   ACC,HALF2
00667 025400101120 0 02 13 00 01  01 20     SLL   ACC,ACC,16
00670 0e006010003 0 00 00 06 01   0003     CRE   ACC,ZRO
00671 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SLL ERROR
00672 0e023010013 0 00 00 23 01   0013     LDR   ACC,HALF2
00673 023400101120 0 02 07 00 01  01 20     SRL   ACC,ACC,16
00674 0e006010012 0 00 00 06 01   0012     CRE   ACC,HALF1
00675 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          SRL ERROR
00676 0e032010001 0 00 00 32 01   0001     LDF   ACC,POS
00677 0e006010001 0 00 00 06 01   0001     CRE   ACC,POS
00700 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          LDD ERROR
00701 0e006210002 0 00 00 06 21   0002     CRE   ACC+1,NEG
00702 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          LDD ERROR
00703 0e032010003 0 00 00 32 01   0003     LDF   ACC,ZRO
00704 0e006010003 0 00 00 06 01   0003     CRE   ACC,ZRO
00705 0e001040014 0 00 00 01 04   0014     RTJ   LINK,ERROR          LDD ERROR
00706 0e006210004 0 00 00 06 21   0004     CRE   ACC+1,ONES
```

Appendix F
ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
00707 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          LDD ERROR
00710 0000230100001 0 00 00 23 01   0001          LDR   ACC,POS
00711 0000232100002 0 00 00 23 21   0002          LDR   ACC+!2D,NEG
00712 0000330100006 0 00 00 33 01   0006          STF   ACC,STORE
00713 0000661000006 0 00 00 06 01   0006          CRE   ACC,STORE
00714 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          STD ERROR
00715 0000662100007 0 00 00 06 21   0007          CRE   ACC+!2D,INCR
00716 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          STD ERROR
00717 0000230100003 0 00 00 23 01   0003          LDR   ACC,ZRO
00720 0000232100004 0 00 00 23 21   0004          LDR   ACC+!2D,ONES
00721 0000330100006 0 00 00 33 01   0006          STF   ACC,STORE
00722 0000661000006 0 00 00 06 01   0006          CRE   ACC,STORE
00723 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          STD ERROR
00724 0000662100007 0 00 00 06 21   0007          CRE   ACC+!2D,INCR
00725 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          STD ERROR
00726 0000230210046 0 00 00 23 02   1046          LDR   WORK,=MUDAT
00727 0000250210047 0 00 00 05 02   1047  LOOP1   CRL   WORK,=FINDA
00730 0000010000765 0 00 00 01 00   0765          RTJ   0,MPFIN
00731 0000230100004 0 00 00 23 01   0004          LDR   ACC,ONES
00732 0000270100766 0 00 00 27 01   0766          STR   ACC,ERRFL
00733 0000230100000 0 00 00 23 01   0000          LDR   ACC,0,WORK
00734 0000270100767 0 00 00 27 01   0767          STR   ACC,DATA
00735 2410004000001 1 01 02 00 02   000001        ADDA  WORK,1
00736 0000230100000 0 00 00 23 01   0000          LDR   ACC,0,WORK
00737 0000270100770 0 00 00 27 01   0770          STR   ACC,DATA+1
00740 2410004000001 1 01 02 00 02   000001        ADDA  WORK,1
00741 0000230100000 0 00 00 23 01   0000          LDR   ACC,0,WORK
00742 0000270100771 0 00 00 27 01   0771          STR   ACC,EXP
00743 2410004000001 1 01 02 00 02   000001        ADDA  WORK,1
00744 0000230100000 0 00 00 23 01   0000          LDR   ACC,0,WORK
00745 0000270100772 0 00 00 27 01   0772          STR   ACC,EXP+1
00746 2410006000003 1 01 03 00 02   000003        ADCA  WORK,3
00747 0000230100000 0 00 00 23 01   0000          LDR   ACC,0,WORK
00750 2410004000001 1 01 02 00 02   000001        ADDA  WORK,1
00751 0000300100000 0 00 00 30 01   0000          MPY   ACC,0,WORK

00752 0000270100773 0 00 00 27 01   0773          STR   ACC,RESUL
00753 0000661000771 0 00 00 06 01   0771          CRE   ACC,EXP
00754 0000270000756 0 00 00 27 00   0756          STR   0,ERRFL
00755 0100001000002 0 01 00 00 01   00 02         RIO   ACC,0,2
00756 0000270100774 0 00 00 27 01   0774          STR   ACC,RESUL+1
00757 0000661000772 0 00 00 06 01   0772          CRE   ACC,EXP+1
00760 0000270000756 0 00 00 27 00   0756          STR   0,ERRFL
00761 0000330000756 0 00 00 33 00   0756          MIN   0,ERRFL
00762 0000010400014 0 00 00 01 04   0014          RTJ   LINK,ERROR          MULTIPLY ERROR
00763 2410004000003 1 01 02 00 02   000003        ADDA  WORK,3
00764 0000610000727 0 00 00 01 00   0727          RTJ   0,LOOP1
00765 0000010010500 0 00 00 01 00   1050  MPFIN   RTJ   0,DIVID
00766 37777777777           ERRFL   GCN   -1

00767 0000000000000         DATA    GCN   0           TWO WORDS
00770 0000000000000                 GCN   0           TO BE MULTIPLIED
00771 0000000000000         EXP     GCN   0           EXPECTED RESULT LEAST SIG.
00772 0000000000000                 GCN   0           EXPECTED RESULT MOST SIG.
00773 0000000000000         RESUL   GCN   0           ACTUAL RESULT LEAST SIG.
00774 0000000000000                 GCN   0           ACTUAL RESULT MOST SIG.

01000 0000000000001         MUDAT   GCN   1
01001 0000000000001                 GCN   1
01002 0000000000001                 GCN   1
01003 0000000000000                 GCN   0

01004 37777777777                   GCN   -1
01005 00000000001                   GCN   1
01006 37777777777                   GCN   -1
01007 37777777777                   GCN   -1

01010 37777777777                   GCN   -1
```

ASSEMBLY LISTING FOR INSTRUCTION CHECK PROGRAM (TINST HAR 1840)

```
01036 0000000000000                                    GCN   0
01037 0000000000000                                    GCN   0
01038 0000000000000                                    GCN   0

01041 !7777777777                                      GCN   !17777777777
01042 !0000000000                                      GCN   !00000000000
01043 3000000000000                                    GCN   !30000000000
01044 037777777777                                     GCN   !03777777777

01045 17777777777                                      GCN   !17777777777
01046 3000000000000                                    GCN   !30000000000
01047 1000000000000                                    GCN   !10000000000
01050 3400000000000                                    GCN   !34000000000

01051 2000000000001                                    GCN   !20000000001
01052 3000000000000                                    GCN   !30000000000
01053 3000000000000                                    GCN   !30000000000
01054 037777777777                                     GCN   !03777777777
01055 0000000000000                            FINDA   GCN   0
01056 0000000000002                                    LOR
01057 37777777777
01060 37777777777
01061 37777777777
01062 0000000000002
01063 0000125252
01064 0000000000376
01065 37777776002
01066 0000000000007
01067 0000000002017
01070 0000000000007
01071 0000000000006
01072 0000002000007
01073 0000017777
01074 0000007777
01075 0000000000775
01076 0000000031025
01077 0002300211135 0 00 00 23 02    1135      DIVID  LDR   WORK,=DIVDA
01100 0000500211136 0 00 00 05 02    1136      LOOP2  CRL   WORK,=DAFIN
01101 0000610011107 0 00 00 01 00    1107             RTJ   0,DIFIN
01102 0000230100004 0 00 00 23 01    0004             LDR   ACC,ONES
01103 0000270100766 0 00 00 27 01    0766             STR   ACC,ERRFL
01104 0000230100000 0 00 02 23 01    0000             LDR   ACC,0,WORK
01105 0000270100767 0 00 00 27 01    0767             STR   ACC,DATA
01106 2400400000001 1 01 02 00 02 000001              ADDA  WORK,1
01107 0000230100000 0 00 02 23 01    0000             LDR   ACC,0,WORK
01110 0000270100770 0 00 00 27 01    0770             STR   ACC,DATA+1
01111 2400400000001 1 01 02 00 02 000001              ADDA  WORK,1
01112 0000230100000 0 00 02 23 01    0000             LDR   ACC,0,WORK
01113 0000270100771 0 00 00 27 01    0771             STR   ACC,EXP
01114 2400400000001 1 01 02 00 02 000001              ADDA  WORK,1
01115 0000230100000 0 00 02 23 01    0000             LDR   ACC,0,WORK
01116 0000270100772 0 00 00 27 01    0772             STR   ACC,EXP+1
01117 2410060000003 1 01 03 00 02 000003              ADDA  WORK,3
01120 0000230100000 0 00 02 23 01    0000             LDR   ACC,0,WORK
01121 2400400000001 1 01 02 00 02 000001              ADDA  WORK,1
01122 0000310100000 0 00 02 31 01    0000             DIV   ACC,0,WORK
01123 0000270100773 0 00 00 27 01    0773             STR   ACC,RESUL
01124 0000060100771 0 00 00 06 01    0771             CRE   ACC,EXP
01125 0000270000766 0 00 00 27 00    0766             STR   0,ERRFL
01126 0100030100003 0 01 00 03 01    0003             RIO   ACC,0,3
01127 0000270100774 0 00 00 27 01    0774             STR   ACC,RESUL+1
01130 0000060100772 0 00 00 06 01    0772             CRE   ACC,EXP+1
01131 0000270000766 0 00 00 27 00    0766             STR   0,ERRFL
01132 0000030000766 0 00 00 03 00    0766             MIN   0,ERRFL
01133 0000010400014 0 00 00 01 04    0014             RTJ   LINK,ERROR                DIVIDE ERROR
01134 2410040000003 1 01 02 00 02 000003              ADDA  WORK,3
01135 0000010010051 0 00 00 01 00    1051             RTJ   0,LOOP2
```

```
P1107 0000000134 0 00 00 00 00 000134      DIFIN  STOP   !134


01110 0000000000                           DIVDA  GCN    0
01111 37777777777                                 G N    -1
01112 0000000000                                  GCN    0
01113 0000000000                                  GCN    0

01114 37777777777                                 GCN    -1
01115 0000000001                                  GCN    1
01116 37777777777                                 GCN    -1
01117 0000000000                                  GCN    0


01120 37777777777                                 GCN    -1
01121 37777777777                                 GCN    -1
01122 0000000001                                  GCN    1
01123 0000000000                                  GCN    0

01124 1000000000                                  GCN    !1000000000
01125 17777777777                                 GCN    !17777777777
01126 0000000000                                  GCN    0
01127 1000000000                                  GCN    !1000000000

01130 0000000005                                  GCN    5
01131 0000000002                                  GCN    2
01132 0000000002                                  GCN    2
01133 0000000001                                  GCN    1
01134 0000000000                           DAFIN  GCN    0


01135 0000001110                                  LOR
01136 0000001134
                                                  END
                                                  EOF
```

A P P E N D I X G

ASSEMBLY LISTING FOR NORD-50 REGISTER TEST PROGRAM (TREG -HAR1842)

```
                                                        MAIN TREG
                                                *
                                                *
                                                *       CHECKS GR0
                                                *
                                                *
00000 0000060000007 0 00 00 06 00    0007       TREG    CRE   0,ZRO
00001 0000000000000 0 00 00 00 00 000000                STOP  0               GR0 NONZERO
00002 0000270000006 0 00 00 27 00    0006               STR   0,STORE
00003 0000060000006 0 00 00 06 00    0006               CRE   0,STORE
00004 0000000000000 0 00 00 00 00 000000                STOP  0               STR <GR0> NONZERO (EA)
00005 0000010000010 0 00 00 01 00    0010               RTJ   0,TST
00006 3777777777777                             STORE   GCN   -1
00007 0000000000000                             ZRO     GCN   0
                                                *
                                                *
                                                *       GR 1,2&3 ARE USED AS WORKING REGISTERS INTHE TEST
                                                *       AND SHOULD BE TESTED FIRST
                                                *
                                                *
                                                *

00010 0000230300145 0 00 00 23 03    0145       TST     LDR   3,=PATTS
00011 0000270300053 0 00 00 27 03    0053               STR   3,IPATT
00012 0000230300146 0 00 00 23 03    0146               LDR   3,=-63
00013 0000270300057 0 00 00 27 03    0057               STR   3,PCNT1

00014 2000230300053 1 00 00 23 03    0053       LOOPA   LDR   3,IPATT,0,0,1
00015 0000270300055 0 00 00 27 03    0055               STR   3,FOUND
00016 2000060300053 1 00 00 06 03    0053               CRE   3,IPATT,0,0,1
00017 0000000000000 0 00 00 00 00 000000                STOP  0               GR 3 ERROR
00020 0000030000053 0 00 00 03 00    0053               MIN   0,IPATT
00021 0000030000057 0 00 00 03 00    0057               MIN   0,PCNT1
00022 0000010000014 0 00 00 01 00    0014               RTJ   0,LOOPA
00023 0000230300147 0 00 00 23 03    0147               LDR   3,=PATTS
00024 0000270300053 0 00 00 27 03    0053               STR   3,IPATT
00025 0000230300150 0 00 00 23 03    0150               LDR   3,=-63
00026 0000270300057 0 00 00 27 03    0057               STR   3,PCNT1
00027 2000230200053 1 00 00 23 02    0053       LOOPB   LDR   2,IPATT,0,0,1
00030 0000270200055 0 00 00 27 02    0055               STR   2,FOUND
00031 2000060200053 1 00 00 06 02    0053               CRE   2,IPATT,0,0,1
00032 0000000000000 0 00 00 00 00 000000                STOP  0               GR2 ERROR
00033 0000030000053 0 00 00 03 00    0053               MIN   0,IPATT
00034 0000030000057 0 00 00 03 00    0057               MIN   0,PCNT1
00035 0000010000027 0 00 00 01 00    0027               RTJ   0,LOOPB
00036 0000230200151 0 00 00 23 02    0151               LDR   2,=PATTS
00037 0000270200053 0 00 00 27 02    0053               STR   2,IPATT
00040 0000230200152 0 00 00 23 02    0152               LDR   2,=-63
00041 0000270200057 0 00 00 27 02    0057               STR   2,PCNT1
00042 2000230100053 1 00 00 23 01    0053       LOOPC   LDR   1,IPATT,0,0,1
00043 0000270100055 0 00 00 27 01    0055               STR   1,FOUND
00044 2000060100053 1 00 00 06 01    0053               CRE   1,IPATT,0,0,1
00045 0000000000000 0 00 00 00 00 000000                STOP  0               GR 1 ERROR
00046 0000030000053 0 00 00 03 00    0053               MIN   0,IPATT
00047 0000030000057 0 00 00 03 00    0057               MIN   0,PCNT1
00050 0000010000042 0 00 00 01 00    0042               RTJ   0,LOOPC
00051 0000010000076 0 00 00 01 00    0076               RTJ   0,TSTRG

00052 0000000000254 0 00 00      00000254       FIN     ACN   PATTS+60
00053 0000000000160 0 00 00      00000160       IPATT   ACN   PATTS
00054 0000000000000                             EXP     GCN   0
00055 0000000000000                             FOUND   GCN   0
00056 0000000000000                             RCOUN   GCN   0
00057 0000000000000                             PCNT1   GCN   0
00060 0000000000000                             SAVE    GCN   0
```

ASSEMBLY LISTING FOR NORD-50 REGISTER TEST PROGRAM (TREG -HAR1842)

```
00051 00027010060 0 00 00 27 01   0060      ERROR  STR   1,SAVE
00052 00240007777 1 00 02 00 01 007777            ANDA  1,!7777
00053 00060010000 0 00 01 23 01   0000            LDR   1,0,1
00054 00027010054 0 00 00 27 01   0054            STR   1,EXP
00055 00023010060 0 00 00 23 01   0060            LDR   1,SAVE
00056 00000000000 0 00 00 00 00 000000            STOP  0

                                                  FORM  20,12
00057 00023040254           0000000254    FINI   FDAT  !2304,PATTS+60
00060 00023040160           0000000160    LDREG  FDAT  !2304,PATTS
00061 00027040055           0000000055    STREG  FDAT  !2704,ROUND
00062 00000004C160          0000000160    CHREG  FDAT  !604,PATTS
00063 00020010001                         INCR1  GCN   !10001
00064 00000010000                         INCR2  GCN   !10000
00065 00000000000                         ROUND  GCN   0

00066 00023000153 0 00 00 23 03   0153      TSTRG  LDR   3,#60
00067 00027000057 0 00 00 27 03   0057             STR   3,PCNT1
00070 00023000154 0 00 00 23 03   0154             LDR   3,#3
00071 00027000056 0 00 00 27 03   0056             STR   3,RCOUN
00072 00027000075 0 00 00 27 00   0075             STR   0,ROUND
00073 00023000070 0 00 00 23 03   0070             LDR   3,LDREG
00074 00023020071 0 00 00 23 02   0071             LDR   2,STREG
00075 00023010072 0 00 00 23 01   0072             LDR   1,CHREG
00076 00021000073 0 00 00 21 03   0073      LOOP   SUB   3,INCR1
00077 00020000073 0 00 00 20 03   0073      LOOP1  ADD   3,INCR1
00100 00142010000 0 00 01 02 01   0000             EXC   1,0,3
00101 00003000057 0 00 00 03 00   0057             MIN   0,PCNT1
00102 00001000107 0 00 00 01 00   0107             RTJ   0,LOOP1
00103 00023000155 0 00 00 23 03   0155             LDR   3,#60
00104 00027000057 0 00 00 27 03   0057             STR   3,PCNT1
00105 00023000156 0 00 00 23 03   0156             LDR   3,#3
00106 00027000056 0 00 00 27 03   0056             STR   3,RCOUN
00107 00021020074 0 00 00 21 02   0074             SUB   2,INCR2
00110 00021010073 0 00 00 21 01   0073             SUB   1,INCR1
00111 00020020074 0 00 00 20 02   0074      LOOP2  ADD   2,INCR2
00112 00100010000 0 00 00 01 00   0000             EXC   1,0,2
00113 00003000056 0 00 00 03 00   0056             MIN   0,RCOUN
00114 00020010073 0 00 00 20 01   0073             ADD   1,INCR1
00115 00142010000 0 00 01 02 01   0000             EXC   1,0,1
00116 00001000061 0 00 00 01 00   0061             RTJ   0,ERROR
00117 00003000057 0 00 00 03 00   0057             MIN   0,PCNT1
00120 00001000121 0 00 00 01 00   0121             RTJ   0,LOOP2
00121 00003000075 0 00 00 03 00   0075             MIN   0,ROUND
00122 00023000070 0 00 00 23 03   0070             LDR   3,LDREG
00123 00020000075 0 00 00 20 03   0075             ADD   3,ROUND
00124 00006000067 0 00 00 06 03   0067             CRE   3,FINI
00125 00001000137 0 00 00 01 00   0137             RTJ   0,*+2
00126 00000000134 0 00 00 00 00 000134             STOP  !134
00127 00023010072 0 00 00 23 01   0072             LDR   1,CHREG
00130 00020010075 0 00 00 20 01   0075             ADD   1,ROUND
00131 00023020157 0 00 00 23 02   0157             LDR   2,#60
00132 00027000057 0 00 00 27 02   0057             STR   2,PCNT1
00133 00023020071 0 00 00 23 02   0071             LDR   2,STREG
00134 00001000106 0 00 00 01 00   0106             RTJ   0,LOOP

00135 00000000160                         LDR
00136 37777777701
00137 00000000160
00140 37777777701
00141 00000000160
```

```
00152 3777777701
00153 3777777704
00154 0000000003
00155 3777777704
00156 0000000003
00157 3777777704


                                                    FORM  4,4,4,4,4,4,4,4
00160 0000000000        0000000000      PATTS FDAT  0,0,0,0,0,0,0,0
00161 0210421042;       0000000001            FDAT  1,1,1,1,1,1,1,1
00162 0421042104?       0000000002            FDAT  2,2,2,2,2,2,2,2
00163 0631463146?       0000000003            FDAT  3,3,3,3,3,3,3,3
00164 1042104210?       0000000004            FDAT  4,4,4,4,4,4,4,4
00165 1252525252?       0000000005            FDAT  5,5,5,5,5,5,5,5
00166 1463146314?       0000000006            FDAT  6,6,6,6,6,6,6,6
00167 1673567356?       0000000007            FDAT  7,7,7,7,7,7,7,7
00170 2104210421?       0000000010            FDAT  8,8,8,8,8,8,8,8
00171 2314631463?       0000000011            FDAT  9,9,9,9,9,9,9,9
00172 2525252525?       0000000012            FDAT  10,10,10,10,10,10,10,10
00173 2735673567?       0000000013            FDAT  11,11,11,11,11,11,11,11
00174 3146314631?       0000000014            FDAT  12,12,12,12,12,12,12,12
00175 3356735673?       0000000015            FDAT  13,13,13,13,13,13,13,13
00176 3567356735?       0000000016            FDAT  14,14,14,14,14,14,14,14
00177 3777777777        0000000017            FDAT  15,15,15,15,15,15,15,15
00200 3777777776        0000000016            FDAT  15,15,15,15,15,15,15,14
00201 3777777775        0000000015            FDAT  15,15,15,15,15,15,15,13
00202 3777777773        0000000013            FDAT  15,15,15,15,15,15,15,11
00203 3777777767        0000000007            FDAT  15,15,15,15,15,15,15,7
00204 3777777757        0000000017            FDAT  15,15,15,15,15,15,14,15
00205 3777777737        0000000017            FDAT  15,15,15,15,15,15,13,15
00206 3777777677        0000000017            FDAT  15,15,15,15,15,15,11,15
00207 3777777577        0000000017            FDAT  15,15,15,15,15,15,14,15
00210 3777777737        0000000017            FDAT  15,15,15,15,15,15,13,15
00211 3777777577        0000000017            FDAT  15,15,15,15,15,15,7,15
00212 3777777377        0000000017            FDAT  15,15,15,15,15,14,15,15
00213 3777776777        0000000017            FDAT  15,15,15,15,15,13,15,15
00214 3777775777        0000000017            FDAT  15,15,15,15,15,11,15,15
00215 3777773777        0000000017            FDAT  15,15,15,15,15,7,15,15
00216 3777767777        0000000017            FDAT  15,15,15,15,14,15,15,15
00217 3777757777        0000000017            FDAT  15,15,15,15,13,15,15,15
00220 3777737777        0000000017            FDAT  15,15,15,15,11,15,15,15
00221 3777677777        0000000017            FDAT  15,15,15,15,7,15,15,15
00222 3777577777        0000000017            FDAT  15,15,15,14,15,15,15,15
00223 3777377777        0000000017            FDAT  15,15,15,13,15,15,15,15
00224 3776777777        0000000017            FDAT  15,15,15,11,15,15,15,15
00225 3775777777        0000000017            FDAT  15,15,15,7,15,15,15,15
00226 3773777777        0000000017            FDAT  15,15,14,15,15,15,15,15
00227 3767777777        0000000017            FDAT  15,15,13,15,15,15,15,15
00230 3737777777        0000000017            FDAT  15,15,7,15,15,15,15,15
00231 3677777777        0000000017            FDAT  15,14,15,15,15,15,15,15
00232 3577777777        0000000017            FDAT  15,13,15,15,15,15,15,15
00233 3377777777        0000000017            FDAT  15,11,15,15,15,15,15,15
00234 3677777777        0000000017            FDAT  15,7,15,15,15,15,15,15
00235 3577777777        0000000017            FDAT  14,15,15,15,15,15,15,15
00236 3377777777        0000000017            FDAT  13,15,15,15,15,15,15,15
00237 2777777777        0000000017            FDAT  11,15,15,15,15,15,15,15
00240 1777777777        0000000017            FDAT  7,15,15,15,15,15,15,15
00241 0000000017        0000000017            FDAT  0,0,0,0,0,0,0,15
00242 0000000360        0000000000            FDAT  0,0,0,0,0,0,15,0
00243 0000007400        0000000000            FDAT  0,0,0,0,0,15,0,0
00244 0000170000        0000000000            FDAT  0,0,0,0,15,0,0,0
00245 0003600000        0000000000            FDAT  0,0,0,15,0,0,0,0
00246 0074000000        0000000003            FDAT  0,0,15,0,0,0,0,0
00247 0170000000        0000000000            FDAT  0,15,0,0,0,0,0,0
00250 3600000000        0000000000            FDAT  15,0,0,0,0,0,0,0
00251 0000000001        0000000001            FDAT  0,0,0,0,0,0,0,1
00252 0000000020        0000000000            FDAT  0,0,0,0,0,0,1,0
```

ASSEMBLY LISTING FOR NORD-50 REGISTER TEST PROGRAM (TREG -HAR1842)

```
00253 0000000000400     0000000000000     FDAT  0,0,0,0,0,1,0,0
00254 0000010010000     0000000000000     FDAT  0,0,0,0,1,0,0,0
00255 0000020020000     0000000000000     FDAT  0,0,0,1,0,0,0,0
00256 0000040040000     0000000000000     FDAT  0,0,1,0,0,0,0,0
00257 0001000000000     0000000000000     FDAT  0,1,0,0,0,0,0,0
00260 0002000000000     0000000000000     FDAT  1,0,0,0,0,0,0,0

                                          FORM  4,4,4,4,4,4,4,4
00261 0000000000000     0000000000000     FDAT  0,0,0,0,0,0,0,0
00262 0210421042104     0000000000001     FDAT  1,1,1,1,1,1,1,1
00263 0421042104210     0000000000002     FDAT  2,2,2,2,2,2,2,2
00264 0631463146310     0000000000003     FDAT  3,3,3,3,3,3,3,3
00265 1042104210421     0000000000004     FDAT  4,4,4,4,4,4,4,4
00266 1252525252525     0000000000005     FDAT  5,5,5,5,5,5,5,5
00267 1463146314631     0000000000006     FDAT  6,6,6,6,6,6,6,6
00270 1673567356735     0000000000007     FDAT  7,7,7,7,7,7,7,7
00271 2104210421042     0000000000010     FDAT  8,8,8,8,8,8,8,8
00272 2314631463146     0000000000011     FDAT  9,9,9,9,9,9,9,9
00273 2525252525252     0000000000012     FDAT  10,10,10,10,10,10,10,10
00274 2735673567356     0000000000013     FDAT  11,11,11,11,11,11,11,11
00275 3146314631463     0000000000014     FDAT  12,12,12,12,12,12,12,12
00276 3356735673567     0000000000015     FDAT  13,13,13,13,13,13,13,13
00277 3567356735673     0000000000016     FDAT  14,14,14,14,14,14,14,14
00300 3777777777777     0000000000017     FDAT  15,15,15,15,15,15,15,15
00301 3777777777676     0000000000016     FDAT  15,15,15,15,15,15,15,14
00302 3777777777575     0000000000015     FDAT  15,15,15,15,15,15,15,13
00303 3777777777373     0000000000013     FDAT  15,15,15,15,15,15,15,11
00304 3777777777767     0000000000007     FDAT  15,15,15,15,15,15,15,7
00305 3777777777757     0000000000017     FDAT  15,15,15,15,15,15,14,15
00306 3777777777737     0000000000017     FDAT  15,15,15,15,15,15,13,15
00307 3777777777577     0000000000017     FDAT  15,15,15,15,15,15,11,15
00310 3777777777757     0000000000017     FDAT  15,15,15,15,15,15,14,15
00311 3777777777737     0000000000017     FDAT  15,15,15,15,15,15,13,15
00312 3777777777577     0000000000017     FDAT  15,15,15,15,15,15,7,15
00313 3777777777377     0000000000017     FDAT  15,15,15,15,15,14,15,15
00314 3777777767777     0000000000017     FDAT  15,15,15,15,15,13,15,15
00315 3777777757777     0000000000017     FDAT  15,15,15,15,15,11,15,15
00316 3777777737777     0000000000017     FDAT  15,15,15,15,15,7,15,15
00317 3777777677777     0000000000017     FDAT  15,15,15,15,14,15,15,15
00320 3777777577777     0000000000017     FDAT  15,15,15,15,13,15,15,15
00321 3777777377777     0000000000017     FDAT  15,15,15,15,11,15,15,15
00322 3777776777777     0000000000017     FDAT  15,15,15,15,7,15,15,15
00323 3777775777777     0000000000017     FDAT  15,15,15,14,15,15,15,15
00324 3777773777777     0000000000017     FDAT  15,15,15,13,15,15,15,15
00325 3777767777777     0000000000017     FDAT  15,15,15,11,15,15,15,15
00326 3777757777777     0000000000017     FDAT  15,15,15,7,15,15,15,15
00327 3777737777777     0000000000017     FDAT  15,15,14,15,15,15,15,15
00330 3776777777777     0000000000017     FDAT  15,15,13,15,15,15,15,15
00331 3773777777777     0000000000017     FDAT  15,15,7,15,15,15,15,15
00332 3767777777777     0000000000017     FDAT  15,14,15,15,15,15,15,15
00333 3757777777777     0000000000017     FDAT  15,13,15,15,15,15,15,15
00334 3737777777777     0000000000017     FDAT  15,11,15,15,15,15,15,15
00335 3677777777777     0000000000017     FDAT  15,7,15,15,15,15,15,15
00336 3577777777777     0000000000017     FDAT  14,15,15,15,15,15,15,15
00337 3377777777777     0000000000017     FDAT  13,15,15,15,15,15,15,15
00340 2777777777777     0000000000017     FDAT  11,15,15,15,15,15,15,15
00341 1777777777777     0000000000017     FDAT  7,15,15,15,15,15,15,15
00342 0000000000017     0000000000017     FDAT  0,0,0,0,0,0,0,15
00343 0000000003600     0000000000000     FDAT  0,0,0,0,0,0,15,0
00344 0000000074000     0000000000000     FDAT  0,0,0,0,0,15,0,0
00345 0000017000000     0000000000000     FDAT  0,0,0,0,15,0,0,0
00346 0000360000000     0000000000000     FDAT  0,0,0,15,0,0,0,0
00347 0007400000000     0000000000000     FDAT  0,0,15,0,0,0,0,0
00350 0170000000000     0000000000003     FDAT  0,15,0,0,0,0,0,0
00351 3000000000000     0000000000000     FDAT  15,0,0,0,0,0,0,0
00352 0000000000001     0000000000001     FDAT  0,0,0,0,0,0,0,1
00353 0000000003020     0000000000000     FDAT  0,0,0,0,0,0,1,0
00354 0000000000400     0000000000000     FDAT  0,0,0,0,0,1,0,0
```

```
00355 0000001200B0  0000000000B0        FDAT  0,0,0,0,1,0,0,0
00356 0000002000C0  0000000000D0        FDAT  0,0,0,1,0,0,0,0
00357 0000040000E0  0000000000F0        FDAT  0,0,1,0,0,0,0,0
00360 0001000000F0  0000000000E0        FDAT  0,1,0,0,0,0,0,0
00361 0002000000E0  0000000000C0        FDAT  1,0,0,0,0,0,0,0
                                        END
                                        EOF
```

ASSEMBLY LISTING FOR NORD-50 REGISTER TEST PROGRAM (TREG -HAR1842)

ASSEMBLY LISTING FOR NORD-50 REGISTER TEST PROGRAM (TREG -HAR1842)

A P P E N D I X H

DOCUMENTATION REVIEW

Descriptions of manual content within functional groups are as follows:

NORD-50 RELATED MANUALS

1. NORD-50 FUNCTIONAL DESCRIPTION ND-05.007

This manual describes the main building blocks of the Nord-50 and their functions, from a hardware point of view.

It is written for technical and maintenance personnel requiring detailed information about the Nord-50, and should also be of be of interest to software personnel.

Some knowledge of digital techniques and computer systems in general is recommended. Knowledge of the Nord-50 instruction set found in the Nord-50 Reference Manual is necessary.

2. NORD-50 TEST PROGRAMS ND-30.007

This manual contains the descriptions of the test programs used with the Nord-50.

It is written for field service engineers and technical personnel directly involved with testing or maintaining the Nord-50 CPU. It contains detailed information on the contents and structure of each test program and the procedure for their implementation. The manual is the centralized reference document on these test programs.

In-depth knowledge of the functioning of the Nord-50 CPU is required.

3. NORD-10/NORD-50 COMMUNICATION SYSTEM ND-06.005

This manual describes the operation of the communication hardware in the Nord-10/Nord-50 system. It gives a general description of memory communication, and describes communication instructions and simulated memory programming. It is written for technical and maintenance personnel and communication system programmers.

Familiarity with the Nord-10/S amd the Nord-50 at the level described in the Nord-10/S and Nord-50 Functional Descriptions is required, also knowledge of the Nord-10/S and Nord-50 instruction sets, which can be found in the Nord-10/S and Nord-50 Reference Manuals.

4. NORD-50 REFERENCE MANUAL ND-05.003

This manual gives a detailed description of the Nord-50 registers and instructions set. It is intended for programmers and operators. The part concerning registers should be of interest to technical and maintenance personnel.

5. NORD-50 LOADER USER'S GUIDE ND-60.083

This manual gives a brief introduction to the loading process. It describes how to use the Nord-50 Loader, how to load an overlaid program, and how to define the communication between the Nord-50 and the Nord-10. The relocatable code (BRF5) format and the error messages produced by the loader are also described.

It is written for all programmers using the Nord-50. It is also recommended for system supervisors who maintain a library of relocatable modules for the Nord-50 with the Nord-50 BRF Editor. No previous knowledge of losders is required. Familiarity with the Nord-10/S-Nord-50 system is of great advantage for understanding the more advanced use of the Loader.

6. NORD-50 MONITOR USER'S GUIDE AND SYSTEM DOCUMENTATION ND-60.076

The first three chapters of this manual describe the method to be used to run Nord-50 programs. All available commands are explained, including those reserved for the system supervisor. File access and the use of monitor calls from the Nord-50 are described.

Chapters four to six contain the Monitor system architecture and describe how to implement monitor and system documentation.

This manual should be read by all Nord-50 users. The first three chapters are intended for users who execute programs on the Nord-50. A working knowledge of SINTRAN III and with Nord-50 Fortran are required. These can be obtained from the SINTRAN III Timesharing/Batch Guide, ND-60.132 and the Nord-50 Fortran Reference Manual, ND-60.095.

Chapters four to six are not required reading in order to use the Monitor, but should be read by system programmers and analysts and by system supervisors installing the Monitor. Thorough knowledge of the internal operation of SINTRAN III and/or the underlying hardware is required. Less detailed information is required by users reading these chapters as a general description of the Monitor.

7. NORD-50 ASSEMBLER ND-60.075

This manual describes the Nord-50 assembler operating procedure, the assembler directives, macro usage and external program representation. The appendicies contain tables of instructions, BRF directives generated by the assembler and numeric opcodes. It is written for any user of the Nord-50 assembler. Familiarity with assembler programming and sufficient SINTRAN III experience to run the assembler is required. Details of the instruction set are given in the Nord-50 Reference Manual, ND-05.003.

8. NORD-50 BRF Editor ND-60.098
This manual describes the commands available in the NORD-50 BRF editor to maintain and modify NORD-50 BRF files and to disassemble the BRF directives and machine instructions. It is written for those maintaining libraries of relocatable NORD-50 modules, including system supervisors, advanced programmers and system programmers. Some knowledge of the internal BRF format (BRF5) is required. This can be found in the NORD-50 Loader User's Guide, ND-60.083, which also gives the required background information about loading NORD-50 segments.

9. NORD-50 FORTRAN REFERENCE MANUAL ND-60.095

This is a reference manual describing the Nord-50 implementation of Fortran. It also describes external program representation, complier operating procedure and file access under the SINTRAN III Operating System. Appendicies include error message explanations, summaries of loader and monitor commands and a short description of internal data representation and procedure call conventions for interfacing to other languages.

The manual is written for Fortran programmers of the Nord-50. Familiarity with elementary Fortran and SINTRAN III usage are recommended, but the manual contains all the information required to write, compile, load and execute a Fortran program on the Nord-50.

MANUALS FOR THE SPECIAL SYSTEMS INTERFACED OR USED WITH THE NORD-50

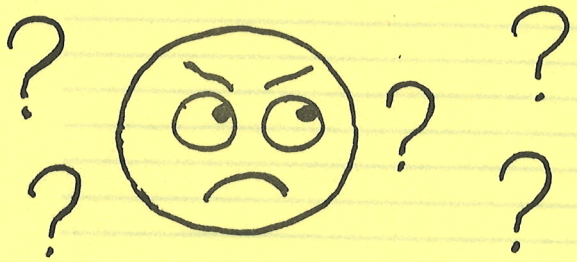
1. BIG MULTIPORT MEMORY SYSTEM ND-06.007

This manual gives a general and detailed description of the Big Multiport Memory System architecture, including memory, service and control channel specifications, and installation.

It is written for technical and maintenance personnel requiring a good understanding of this system, and also for installation personnel. Familiarity with the Nord-10/S and the Nord-50 and the Nord-100 CPUs at the level described in their respective functional descriptions is required.

Index
Index

* * * * * * * * * * SEND US YOUR COMMENTS!!! * * * * * * * * * *

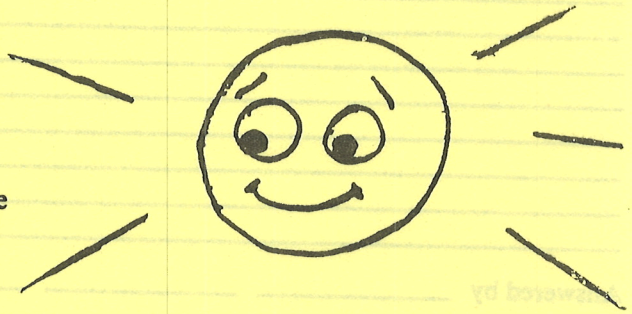


Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card - and an answer to your comments.

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: NORD - 50 Hardware Maintenance Manual

Manual number: ND-30.010.02

What problems do you have? (use extra pages if needed) _____

Do you have suggestions for improving this manual? _____

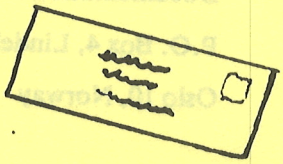
Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

What are you using this manual for? _____

Send to: Norsk Data A.S.
Documentation Department
P.O. Box 4, Lindeberg Gård
Oslo 10, 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 4, Lindeberg Gård
Oslo 10, Norway

– we make bits for the future

NORSK DATA A.S BOX 4 LINDEBERG GARD OSLO 10 NORWAY PHONE: 30 90 30 TELEX: 18661