

NX-SPICE

User's Manual

August 1986

p-cad[®]
PERSONAL CAD SYSTEMS INC.



NX-SPICE

User's Manual

August 1986

p-cad[®]
PERSONAL CAD SYSTEMS INC.

000-0058-01

COPYRIGHT

Copyright (c) 1986 by Personal CAD Systems, Inc.
(P-CAD).

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Personal CAD Systems, Inc.

Personal CAD Systems, Inc. provides this manual "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. P-CAD may make improvements and/or changes in the product(s) and/or the program(s) described in this manual at any time and without notice.

Although P-CAD has gone to great effort to verify the integrity of the information herein, this publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes will be incorporated in new editions of this publication.

TRADEMARKS

NX-SPICE, PC-CAPS, PC-LINK, and PC-NODES are trademarks of Personal CAD Systems, Inc. (P-CAD).

SPICE is a trademark of The Regents of the University of California.

HSPICE is a trademark of Meta Software, Inc.

PSPICE is a trademark of MicroSim, Inc.

ABOUT THIS MANUAL

This manual describes NX-SPICE, a program that converts P-CAD schematic netlist files into a format that is compatible with SPICE simulation programs.

Chapter 1, INTRODUCTION, provides an overview of NX-SPICE and installation instructions.

Chapter 2, PREPARING THE SCHEMATIC DATABASE, gives instructions for using PC-CAPS to create a schematic database to be used with NX-SPICE.

Chapter 3, PREPARING THE INPUT FILES, gives instructions for creating the files to be input into NX-SPICE.

Chapter 4, USING NX-SPICE, gives instructions for configuring and running NX-SPICE.

Chapter 5, VIEWING THE OUTPUT FILES, tells how to view, print, and interpret the NX-SPICE output files.

Appendix A, ERROR MESSAGES, explains error messages that might be produced during NX-SPICE program operation.

Appendix B, NX-SPICE SYMBOL LIBRARY, describes the SPICE symbol library.



NOTATION

This manual gives step-by-step procedures and examples. To make it easy for you to follow these procedures, we use the following notation.

<xxxx> Angle brackets around lowercase letters indicate a variable name that may be entered by the system or by you. For example:

`<filename>.SCH`

[] Square brackets indicate the name of a key. For example:

`[Return]`

[Return] `[Return]` indicates the key that is used to execute a command or accept an option. This key may be labeled differently depending on your system. For example:

`[RETURN], [↵], [Enter],
[Enter↵], [ENTER]`

[]-[] Square brackets connected with a hyphen indicate keys that must be pressed simultaneously. For example:

Press `[Ctrl]-[Alt]-[Del]`.

UPPER Uppercase letters indicate a command or an element that must be typed as shown. For example:

Type PCPLOTS and press [Return]

/ A forward slash separates main menu and submenu command combinations. For example:

DRAW/ARC

* An asterisk in a filename or in a filename extension indicates that any character(s) can occupy that position and all the remaining positions in the filename or extension. For example, the DOS command

DIR *.SYM

displays a list of all the filenames with the extension .SYM in the current directory.

TESTFILE TESTFILE is a sample filename that you must replace with the filename you intend to use. For example:

Database Filename :TESTFILE.SCH
Netlist Filename :TESTFILE.NLT

CONTENTS

CHAPTER 1. INTRODUCTION	1-1
Overview	1-1
System Requirements	1-4
Installation	1-4
CHAPTER 2. PREPARING THE SCHEMATIC	
DATABASE	2-1
Schematic Database Structure	2-1
Hierarchical Structure	2-1
Multisheet Structure	2-1
Special PC-CAPS Input For NX-SPICE	2-2
Creating SPICE-compatible Symbols	2-2
Handling Multisheet Schematics	2-3
Sheet Identification	2-3
Hierarchically Embedded Sheets	2-4
NX-SPICE Attributes	2-4
The <filename>.SPI Statements	2-4
Assigning Attributes	2-5
CARDx	2-6
INCLx	2-7
PNAME	2-8
SNAME	2-11
SPGN (Global Node)	2-12
SPMI (Mutual Inductor)	2-13
SPPx	2-14
SPTI (Title)	2-16
CHAPTER 3. PREPARING THE INPUT FILES. .	3-1
Extracting the Binary Netlists	3-1
Creating the Batch Control File	3-2
CHAPTER 4. USING NX-SPICE	4-1
Interactive Mode	4-2
Configuring NX-SPICE	4-3
Running NX-SPICE	4-7

CONTENTS (Continued)

Batch Processing.	4-9
Command Line Mode	4-11
CHAPTER 5. VIEWING THE OUTPUT FILES. . .	5-1
Viewing and Printing Files.	5-1
The <filename>.SPI File.	5-2
Control Statements.	5-2
Element Statements	5-3
Subcircuit Definitions.	5-5
Appendix A. ERROR MESSAGES.	A-1
File Related Error Messages	A-1
Processing Error Messages.	A-3
Appendix B. NX-SPICE SYMBOL LIBRARY.	B-1
Symbol Graphics.	B-3

FIGURES

1-1. NX-SPICE Data Flow.	1-3
2-1. Inverter Subcircuit	2-9
3-1. Sample Batch Control File.	3-4
4-1. NX-SPICE Opening Menu	4-2
4-2. NX-SPICE Configuration Screen.	4-4
4-3. NX-SPICE Program Screen.	4-7
4-4. Sample Program Screen.	4-8
4-5. Batch Control Menu Screen	4-10

TABLE

B-1. HSPICE Elements.	B-1
-------------------------------	-----

CHAPTER 1. INTRODUCTION

The NX-SPICE interface program translates P-CAD circuit schematics to files that are compatible with SPICE simulation programs.

NX-SPICE is compatible with HSPICE, PSPICE, Berkeley SPICE 2G and equivalent SPICE programs. NX-SPICE fully supports multisheet and hierarchical designs.

This chapter provides an overview of NX-SPICE, and describes system requirements and installation procedures.

OVERVIEW

NX-SPICE, together with PC-NODES, enables you to produce files in SPICE format from a PC-CAPS created schematic.

There are four stages in this process:

1. Using PC-CAPS, you prepare the schematic design. This design consists of one or more schematic files (<filename>.SCH), which are created by interconnecting basic schematic symbols (<filename>.SYM). The types of symbols you can use are described in Chapter 2, "Preparing the Schematic Database."
2. Using PC-NODES, you extract a binary netlist file (<filename>.NLT) from each schematic file in your design. The netlist file contains component and interconnection information for the schematic. If a

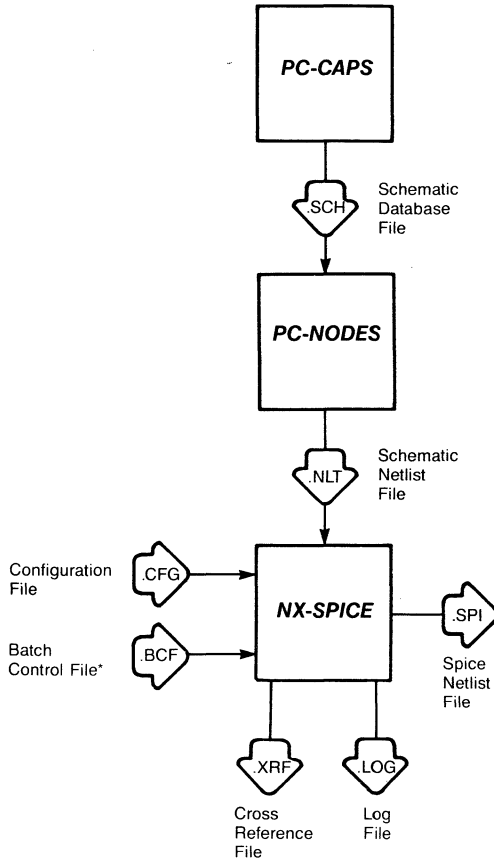
design consists of a single schematic, you can input the corresponding netlist file directly to NX-SPICE, in which case you do not have to perform step 3.

3. If your design is multisheet or hierarchical, you next use a text editor to create a batch control file (<filename>.BCF). This file contains the names of all netlist files to be processed. NX-SPICE will repeatedly read the batch control file and process the next netlist file until all the files have been translated.
4. You then input the netlist file or the batch control file into NX-SPICE. NX-SPICE uses the netlist(s) to produce three files:
 - **The SPICE Netlist File (<filename>.SPI)** is a SPICE-format netlist file. This file is described in detail in Chapter 5, "Viewing the Output Files."
 - **The NX-SPICE Log File (NXSPICE.LOG)** contains all messages displayed during the execution of NX-SPICE. This file is useful for examining any errors that are reported by NX-SPICE.

NOTE: Each time you run NX-SPICE the previous log file is overwritten. If you want to save a log file, you must rename it, or print it out.

- **The Cross Reference File (<filename>.XRF)** matches nonnumeric node names in the original schematic to numeric names in the <filename>.SPI file. NX-SPICE only creates the cross reference file if the "Create Numeric Node Names" configuration option is active.

Figure 1-1 shows the process of creating a SPICE netlist from a PC-CAPS created schematic.



*Optional File

Figure 1-1. NX-SPICE Data Flow

SYSTEM REQUIREMENTS

Before you install NX-SPICE, your computer system must have the following minimum configuration:

- IBM PC/XT, PC/AT, TI Professional, Tandy TRS 2000, or equivalent
- 640K of RAM
- P-CAD supported graphics board and monitor
- DOS 2.0 or higher operating system
- The CONFIG.SYS file in the root directory, containing a BUFFERS value of at least 12 and a FILES value of at least 20
- The PCADDRV.SYS file (created automatically when you use the P-CAD INSTALL program) in the root directory and the appropriate loadable device driver files in the appropriate directory

INSTALLATION

Your NX-SPICE diskette contains the following files:

Program file:

NXSPICE.EXE

Symbol files:

VFM.SYM

IDC.SYM

ISIN.SYM

IFM.SYM

CCCS.SYM

IAC.SYM

IEXP.SYM

VCCS.SYM

CCVS.SYM

IPULSE.SYM

IPWL.SYM

VCVS.SYM

VCCSNL.SYM

VCVSNL.SYM	CCCSNL.SYM	CCVSNL.SYM
RES.SYM	CAP.SYM	CAPNL.SYM
IND.SYM	INDNL.SYM	TLINE.SYM
DIODE.SYM	NPN.SYM	PNP.SYM
NPNS.SYM	PNPS.SYM	NJFET.SYM
PJFET.SYM	NMOSB.SYM	PMOS.SYM
NMOS.SYM	PMOSB.SYM	VAC.SYM
VDC.SYM	VPULSE.SYM	VSIN.SYM
VEXP.SYM	VPWL.SYM	

Appendix B, "NX-SPICE Symbol Library," lists and describes the symbols provided with NX-SPICE.

To install NX-SPICE on your hard disk, insert the diskette in drive A and use the following procedures. These procedures assume that you are using the P-CAD recommended directory structure.

Install the program file in your \PCAD\EXE directory. To do this, first change to the \PCAD\EXE directory by typing:

```
CD \PCAD\EXE [Return]
```

Copy the file by typing:

```
COPY A:*.EXE [Return]
```

Next, create a new directory for the SPICE symbol library by typing:

```
MD \PCAD\SYM\SPICE [Return]
```

Install the symbol library in the new directory. First, change directories by typing:

```
CD \PCAD\SYM\SPICE [Return]
```

Then copy the files by typing:

```
COPY A:*.SYM [Return]
```


CHAPTER 2. PREPARING THE SCHEMATIC DATABASE

Before running NX-SPICE, you must use PC-CAPS to create a schematic database. This chapter describes PC-CAPS schematic structure and gives guidelines for entering the special information required to construct a SPICE-compatible schematic. To ensure that your database is compatible with NX-SPICE, read this chapter before constructing your schematic.

SCHEMATIC DATABASE STRUCTURE

A PC-CAPS schematic consists of an interconnected group of predefined symbols. Each symbol is a logical and pictorial representation of a component.

With NX-SPICE, P-CAD supplies a library of SPICE-compatible symbols to use as the basic components of a schematic (see Appendix B, "NX-SPICE Symbol Library").

Hierarchical Structure

Hierarchical structure is the usage of symbols to represent subcircuits of a schematic. With PC-CAPS, you can design a symbol to identify a circuit, then use the symbol to represent the circuit in another schematic. PC-CAPS and NX-SPICE support hierarchical structure.

Multisheet Structure

A multisheet schematic is created by producing a number of single-sheet schematics. You indicate

connections between sheets by giving connected wires the same net name on each sheet. NX-SPICE allows up to 100 sheets in a multisheet schematic.

NX-SPICE supports multisheet schematics at all levels of the design hierarchy. You must use NX-SPICE batch mode, not PC-LINK, to process a multisheet schematic.

SPECIAL PC-CAPS INPUT FOR NX-SPICE

As you assemble your schematic database, you must input certain information for NX-SPICE. This information and the PC-CAPS commands used to enter it are described in this section.

Creating SPICE-Compatible Symbols

The primitive symbols provided with NX-SPICE are fully compatible with SPICE. They have preassigned pin order, component type ID, and in some cases component attributes (see Appendix B, "NX-SPICE Symbol Library").

You might want to create new graphics for some of the NX-SPICE primitive symbols. To do so, in PC-CAPS, draw the new symbol with the PC-CAPS DRAW commands. Then enter pins with the ENTR/PIN command and assign a component type ID with the SCMD/SCAT command. Refer to the appropriate primitive symbol in Appendix B, "NX-SPICE Symbol Library," when assigning pin order and component type ID.

You can create symbols for hierarchical subcircuits in a PC-CAPS schematic in SYMB mode. The PC-CAPS default extension for a hierarchical subcircuit is .SYM;

however, you can use either .SYM, .SCH, or any other extension you choose.

After you create a hierarchical symbol, use the ENTR/PIN command in SYMB mode to enter the pins for the symbol. NX-SPICE assigns pin order in the same order that you enter the pins.

Handling Multisheet Schematics

NX-SPICE requires certain information, described below, for multisheet schematics.

Sheet Identification

To indicate that a given sheet is a single-sheet of a multisheet schematic, assign the SHEET attribute to the sheet. In SYMB mode, use the ATTR/ACOM command and enter, anywhere in the design, the following attribute:

SHEET=<sheet id>

where:

sheet id is a one, two, or three character, alphanumeric identifier.

Each sheet must have a unique <sheet id> identifier. For example, you could assign the following attributes to three sheets that constitute one design:

SHEET=01
SHEET=02
SHEET=03

Hierarchically Embedded Sheets

If a multisheet schematic is embedded within a hierarchical design, you must represent the schematic with a symbol. To do so, use one sheet as the primary sheet and place the symbol for the entire multisheet module in this primary sheet. When placing the symbol on the schematic, use the filename of the primary sheet.

NOTE: The primary sheet must be listed first in the list of sheet names included in the batch control file that you submit to NX-SPICE.

NX-SPICE ATTRIBUTES

NX-SPICE uses attributes as information in the <filename>.SPI output file. This section describes the types of statements contained in the output file and how to use attributes to add information to the file.

The <filename>.SPI Statements

The <filename>.SPI file contains four types of statements generated from the information in the PC-CAPS schematic netlist:

Control statements pertain to an entire circuit description. Control Statements are generated by CARDx, INCLx, SPTI, and SPGN attributes assigned to the root circuit in SYMB mode.

Primitive element statements describe instances of low-level components in a circuit. SPPx attributes attached to a low-level component in DETL mode append parameters to the corresponding element statement in the <filename>.SPI file.

Subcircuit calls describe instances of subcircuits in a higher level circuit. SPPx attributes attached to a subcircuit instance in DETL mode append attributes to the corresponding subcircuit call in the <filename>.SPI file. SNAME and PNAME attributes rename a subcircuit call.

Subcircuit definitions define the generic content of a subcircuit but do not describe specific instances of that subcircuit. SPPx attributes attached to the subcircuit in SYMB mode append parameters to the subcircuit definition in the <filename>.SPI file. SNAME and PNAME attributes generate subcircuit definitions with variable parameter lists.

The next section explains how to assign specific NX-SPICE attributes.

Assigning Attributes

This section lists the NX-SPICE attributes in alphabetical order and explains the type of information that can be assigned with each attribute. To assign an attribute, use the PC-CAPS ATTR/ACOM command in either SYMB or DETL mode as specified below. Refer to the *PC-CAPS User's Manual* for details on assigning attributes.

NOTE: In PC-CAPS, when an attribute value contains spaces, you must enclose the value in double quotation marks or in parentheses. NX-SPICE only allows the double quotation marks. The quotation marks are optional for values that do not have spaces.

CARDx

CARDx attributes are assigned to a circuit or subcircuit in SYMB mode. The value of the attribute is copied directly into the <filename>.SPI output file in the order of the CARDx suffix. Each CARDx attribute generates a new line in the <filename>.SPI file. The value appears in the file prior to the element records for the symbol being processed.

The range for CARDx attributes is CARD1 ... CARD99. You do not have to assign CARDx attributes in sequence. If an attribute is missing in the sequence, processing continues with the next CARDx attribute.

The CARDx notation is typically used to generate statements such as .LIB, .OPT, and .WIDTH, and to place comments in the <filename>.SPI file. For example,

```
CARD1=".LIB 33"
CARD2=".LIB 44"
CARD3=".OPT ccc ccc"
CARD4=".WIDTH IN=80 OUT=80"
```

generates the following lines:

```
.LIB 33
.LIB 44
.OPT ccc ccc
.WIDTH IN=80 OUT=80
```

NOTE: If you do not specify a .WIDTH statement NX-SPICE automatically generates a default statement as follows:

```
.WIDTH IN=80 OUT=132.
```

CARDx also supports a decimal notation that enables you to generate lines longer than the 39 characters allowed for attribute values. When the decimal notation is used, the values of CARDx with the same base integer are concatenated onto the same line in the .SPI output file, with a space between each one. For example,

```
CARD1.1="* Now is the time"
CARD1.2="for all good women"
CARD1.3="and men to come to"
CARD1.4="the aid of their party."
```

generates the following line:

```
* Now is the time for all good women and men to
come to the aid of their party.
```

The range for the decimal notation is CARDx.1 ... CARDx.4. You may intermix the decimal notation with normal CARDx notation.

When you use CARDx to place comments in the <filename>.SPI file, begin each line with an asterisk (*); SPICE ignores all text from the * to the end of the line. For example:

```
CARD1="* This is a comment"
```

Because the value of the CARDx attribute is copied literally into the output file, NX-SPICE performs no error checking.

INCLx

The INCLx attribute is attached to a circuit or subcircuit in SYMB mode. Its value is the name of a file that is to be copied literally into the <filename>.SPI

file in the order of the INCLx suffix. The contents of the specified file appears in the <filename>.SPI file immediately after the element record for the circuit or subcircuit being processed.

The INCLx attribute is typically used to copy a model file into the SPICE file. For example, the following attribute specifies that the contents of the CMOS1.MOD file be copied into the <filename>.SPI file:

```
INCL1=CMOS1.MOD
```

The range for INCLx is INCL1 ... INCL99. NX-SPICE processes the INCLx attributes in order of the numeric suffixes, however you do not have to enter the attributes in any order. If attributes are missing in the sequence, NX-SPICE continues processing with the next INCLx attribute.

NX-SPICE performs no error checking on the contents of the file being copied into the <filename>.SPI file.

PNAME

Assign the PNAME attribute to subcircuit symbols in DETL or SYMB mode to rename a subcircuit call. By default, NX-SPICE assigns the subcircuit filename (without path or extension) to the subcircuit call. PNAME overrides the default name, using the attribute value as the subcircuit name in the subcircuit call. In addition, if a subcircuit definition does not already exist, PNAME generates a subcircuit definition, using the symbol netlist file as a template, but using the PNAME attribute value as the subcircuit name.

You can also create special attributes that are passed down from the symbol detail to subcircuit definitions. This parameter passing is useful for versions of SPICE

that do not allow parameters to be appended to the end of subcircuit calls (see "SPPx", below).

Example:

This example shows how to use the PNAME attribute to pass parameters to a subcircuit definition.

Figure 2-1 shows the detail and symbol representations of an inverter subcircuit.

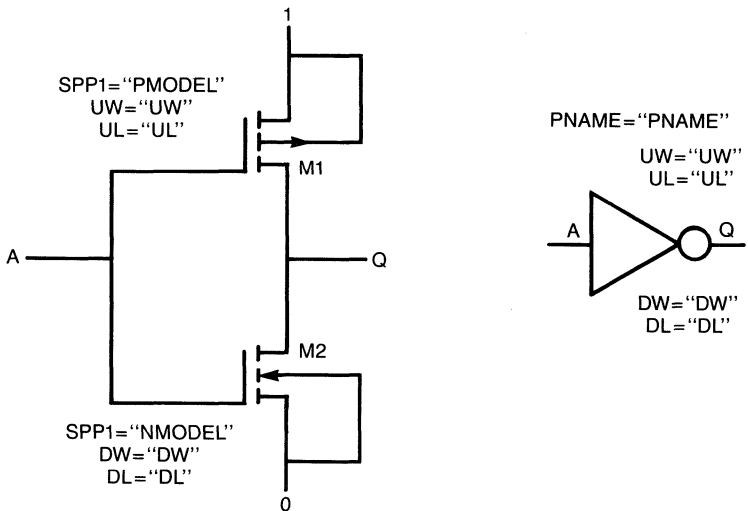


Figure 2-1. Inverter Subcircuit

The following attributes have been attached to the primitives in the subcircuit: SPP1 to call two different model names, PMODEL and NMODEL; and the

specially created attributes, UW and DW, and UL and DL to designate two separate sets of W and L parameters.

The inverter symbol representation has been assigned attributes as prompts (PNAME="PNAME", UW="UW" and so on) to remind the user of the names of the special attributes to assign to the instance of the inverter.

In a schematic database the following values are attached to one instance of the inverter shown above in Figure 2-1:

```
PNAME=INV21
UL="L=10U"
UW="W=21U"
DL="L=15U"
DW="W=31U"
```

In another instance of the inverter symbol, the following attributes are attached:

```
PNAME=INV22
UL="L=12U"
UW="W=22U"
DL="L=17U"
DW="W=30U"
```

NX-SPICE passes parameters from the subcircuit instance to the subcircuit definition in the following order: first the SPP1 attribute, then any specially created attributes, and finally any SPP2 ... SPP99 attributes. The above example generates the following subcircuit definitions in the <filename>.SPI file:

```
.SUBCKT INV21 Q A
M1 1 A Q PMODEL L=10U W=21U
M2 Q A 0 NMODEL L=15U W=31U
```

```

.ENDS
*
*
.SUBCKT INV22 Q A
M1 1 A Q PMODEL L=12U W=22U
M2 Q A 0 NMODEL L=17U W=30U
.ENDS
*
*

```

As you can see from this example, PNAME enables you both to specify parameters common to several subcircuit definitions, using SPPx, and to create special attributes to specify variable values.

Remember, to create a new subcircuit definition, you must change the parameter values and the PNAME attribute value.

SNAME

Assign the SNAME attribute to a symbol in DETL or SYMB mode to rename the subcircuit in the subcircuit call. By default, NX-SPICE assigns the subcircuit filename (without path or extension) to the subcircuit call. SNAME overrides the default name, using the attribute value as the name of the subcircuit call. For example, suppose DS78L12.SYM generates the following subcircuit call in the <filename>.SPI file:

```
XC000001 UN000001 UN000000 DS78L12
```

If you assign the attribute, SNAME=OPAMP, to DS78L12.SYM, the following subcircuit call is generated in the <filename>.SPI file:

```
XC000001 UN000001 UN000000 OPAMP
```

You can attach attributes to the symbol part of the symbol definition to serve as prompts that appear when the component is entered.

SPGN (Global Node)

Assign the SPGN attribute to the root circuit in SYMB mode to specify global nodes. You can specify multiple nodes with a single attribute, and assign SPGN more than once, however the order in which the global nodes appear in the .GLOBAL statement in the <filename>.SPI file is unpredictable. For example,

```
SPGN="1 2"  
SPGN="15"
```

generates the following statement:

```
.GLOBAL 1 2 15
```

or the statement:

```
.GLOBAL 15 1 2
```

If the "Create Numeric Node Names" configuration option is active, NX-SPICE assigns numeric nodes in the range 9000-9999 to nonnumeric nodes assigned as global with the SPGN attribute. NX-SPICE does not change valid numeric nodes assigned as global with the SPGN attribute.

You can also assign nodes as global with the CARDx attribute and a .GLOBAL statement. For example,

```
CARD1=".GLOBAL 1 2 15"
```

When assigning global nodes using CARDx, be certain that the referenced nodes are valid numeric nodes that will not have aliases generated, because NX-SPICE does no checking on CARDx attributes.

Node 0 is global by definition and requires no explicit global assignment.

SPMI (Mutual Inductor)

You assign the SPMI attribute to a circuit or subcircuit in SYMB mode to specify a mutual inductor. The SPMI attribute value specifies the name to be given the mutual inductor, the names of the coupled inductors, and the coupling coefficient.

For example, to specify a mutual inductor named KL1L2 with a coupling coefficient of 0.37, assign the following attribute to the circuit or subcircuit in which the inductors are found:

```
SPMI="KL1L2 L1 L2 0.37"
```

The attribute specified above generates the following lines in the <filename>.SPI file:

```
KL1L2 L1 L2 0.37
L1 3 4 10U
L2 5 6 5U
```

NX-SPICE validates all three element names in the attribute string, but only corrects the mutual inductor name if the "Correct Invalid Component Names" configuration option is active.

You can assign the SPMI attribute as many times as necessary to a single subcircuit symbol.

SPPx

You can assign the SPPx attribute to a primitive in DETL mode, to a subcircuit in DETL mode, or to a subcircuit in SYM mode.

NX-SPICE passes the value of the SPPx attribute directly into the <filename>.SPI file without any error checking.

The SPPx attribute range is SPP1 ... SPP99. NX-SPICE processes the attributes in order of the numeric suffixes and concatenates the attribute values onto a single line in the file with a space between each value. If attributes are missing in the sequence, NX-SPICE continues processing with the next SPPx attribute.

Primitive Symbol Component. When you assign an SPPx attribute to a primitive symbol component, NX-SPICE appends the attribute value to the element statement generated for the component in the <filename>.SPI file. For example, to assign a value of 1.2K to a resistor attached to nodes 1 and 2, assign the following attribute to the resistor symbol in DETL mode:

```
SPP1=1.2K
```

The following line is generated in the <filename>.SPI file:

```
RC000001 1 2 1.2K
```

Most of the SPICE primitive symbols supplied with the NX-SPICE program have been preassigned SPPx attributes with variables as values to prompt you to enter specific values. For example, the resistor symbol, RES.SYM, has the attribute SPP1="VALUE"; MOSFETs and JFETs have the attribute SPP1="MODEL". You can replace "VALUE" with a specific value, and "MODEL"

with a specific model name using the ATTR/SCHG command, or you can add additional parameters using the ATTR/ACOM command.

Refer to your SPICE manual for details on the types of parameters and the order in which to enter them for specific types of primitive components.

NOTE: NX-SPICE symbols for the various voltage and current sources produce a single SPICE output. You must specify attributes to make a source AC, DC, SINE, and so on.

Subcircuit Instance. When you assign an SPPx attribute to a subcircuit component instance in DETL mode, NX-SPICE appends the attribute value to the end of the subcircuit call generated for the component in the <filename>.SPI file. For example, if you assign the following attributes to an instance of a subcircuit, OPAMP1, that is connected to nodes 9, 10, 11, and 12,

```
SPP1="R1=2.2K"
SPP2="V(5)=5"
```

the following subcircuit call is generated in the <filename>.SPI file:

```
XC14 9 10 11 12 OPAMP1 R1=2.2K V(5)=5
```

NOTE: Some versions of SPICE do not recognize parameters that are attached to the end of subcircuit calls. In these versions of SPICE, use the PNAME attribute (see "PNAME" above).

Because attributes associated with the subcircuit definition are copied to all instances of the subcircuit when it is used as a circuit element, subcircuit calls and circuit definitions have duplicate parameter lists. If

this redundancy is a problem, delete the redundant attributes from the subcircuit instance using the ATTR/DATR command in PC-CAPS.

Subcircuit Definitions. When you assign an SPPx attribute in SYMB mode to a subcircuit symbol, NX-SPICE appends the attribute value to the subcircuit definition generated for the symbol in the <filename>.SPI file.

For example, if you assign the following attributes to the OPAMP1 subcircuit symbol, containing nodes 1, 2, 3, and 4,

```
SPP1="R1=5.5K"
SPP2="V(5)=7.2 V(6)=2"
```

the following subcircuit definition is generated in the <filename>.SPI file:

```
.SUBCKT OPAMP1 1 2 3 4 R1=5.5K
+ V(5)=7.2 V(6)=2
```

SPTI (Title)

The SPTI attribute is assigned to the root circuit in SYMB mode to specify a title for the <filename>.SPI file.

If you do not assign an SPTI attribute, the title is the root circuit filename without path or extension. The title is the first line of the <filename>.SPI file.

For example,

```
SPTI="TEST CIRCUIT"
```


generates the following title as the first line of the
<filename>.SPI file:

TEST CIRCUIT



CHAPTER 3. PREPARING THE INPUT FILES

After your schematic is completed, you must use PC-NODES to extract binary netlists for each sheet and hierarchical element.

If your schematic consists of just one schematic file, you can input the netlist directly into NX-SPICE, or you can use a batch control file to process it with other files.

If the schematic is hierarchical and you want to submit the entire circuit, or if the schematic has multiple sheets, you must create a batch control file to input all the netlists to NX-SPICE.

This chapter describes how to prepare the input files for NX-SPICE and contains instructions for creating the binary netlist and the batch control file.

EXTRACTING THE BINARY NETLISTS

Run PC-NODES to extract binary netlists from each sheet or hierarchical element of your schematic. Use the instructions in the *PC-NODES User's Manual*.

Do not use PC-LINK to link the netlists because NX-SPICE handles hierarchy differently and independently of the P-CAD system; you must create a batch control file to process the netlists for multisheet and hierarchical schematic designs.

CREATING THE BATCH CONTROL FILE

In the batch control file you list all the files to be processed during an NX-SPICE batch mode session. Use a text editor such as the DOS EDLIN program to create the file. You can give the file any name you choose, but the extension must be .BCF.

The format for a batch control file is as follows:

```
CIRCUIT [<output filename>]
<root circuit filenames>
[<subcircuit filenames>]
[<subcircuit filenames>]
.
.
[<subcircuit filenames>]
```

```
CIRCUIT [<output filename>]
<root circuit filenames>
[<subcircuit filenames>]
[<subcircuit filenames>]
.
.
[<subcircuit filenames>]
.
.
```

where:

output filename is an optional filename to name the <filename>.SPI file.

root circuit filenames is the main circuit filename of the schematic design or a list of filenames if you have a multisheet schematic.

subcircuit filenames is a subcircuit schematic filename of a hierarchical design or a list of filenames if you have a multisheet schematic.

NOTE: The square brackets indicate optional items, and the angle brackets indicate variables to be replaced by specific filenames. Do not enter either type of brackets in the batch control file.

For each schematic design to be processed, type the keyword **CIRCUIT** followed by an optional output filename. If you don't specify it, the output filename is the same as the first sheet of the main (root) circuit and has the **.SPI** extension.

On the line following **CIRCUIT** specify the root circuit of the design, and on consecutive lines list any subcircuits. **NX-SPICE** assumes **.NLT** extensions for the input (circuit and subcircuit) filenames, and **.SPI** for the output files that are created during **NX-SPICE** operation unless you specify a different extension.

You can put comments in the batch control file by using the percent sign (%). All text to the end of the line, including the percent sign, is ignored by **NX-SPICE**.

If you want to process a multisheet schematic, you must specify all of the sheets on a single line of the batch control file, or continue lines with the plus sign (+).

The following example shows the format for root circuit or subcircuit entries in a batch control file.

```
file1 file2 file3           % 3 sheet module

file4 file5 +
file6 file7 +             % 5 sheet module
file8

file9                       % 1 sheet module
```

Figure 3-1 shows a sample batch control file for two separate hierarchical schematic designs.

```
% multisheet counter
CIRCUIT Count_3
Counter Counter2 Counter3
Full
Half 1
Half 2
Inv 1
Inv 2

% single-sheet counter
CIRCUIT Count_1
Counter
Full
Half 1
Half 2
```

Figure 3-1. Sample Batch Control File

Use NX-SPICE batch processing mode to process the batch control file (see "Batch Processing" in Chapter 4).

CHAPTER 4. USING NX-SPICE

This chapter describes the required conditions and procedures for configuring and running NX-SPICE.

Before running NX-SPICE, be sure that:

- Your system is correctly configured.
- You have installed the NX-SPICE program file (NXSPICE.EXE).
- You have assembled the schematic circuit.
- You have extracted the netlist files.

NX-SPICE has three operating modes: interactive, command line, and batch. In interactive mode, NX-SPICE displays a series of screens and you select options and specify filenames. In command line mode, you specify filenames on the command input line when you start the program, and NX-SPICE processes the input file automatically. In batch mode, you specify the name of a batch control file that instructs NX-SPICE to process several files automatically. Use batch mode to process multisheet and hierarchical schematic designs.

The following sections describe how to use interactive mode to start and configure NX-SPICE, how to use command line mode, and how to process batch control files.

INTERACTIVE MODE

In interactive mode, you must first start NX-SPICE to display the Opening Menu.

To start NX-SPICE, be sure you are in the appropriate project directory, then type:

NXSPICE [Return]

When the NX-SPICE Title Screen appears, press any key to continue. The system displays the Opening Menu as shown in Figure 4-1.

NX-SPICE

Options:

Configure NX-SPICE

Batch Processing

>> Run NX-SPICE <<

Exit NX-SPICE

Press: [SPACE] for next option; [RETURN] to accept

Figure 4-1. NX-SPICE Opening Menu

This menu provides the following options.

Configure NX-SPICE - Allows you to view or change NX-SPICE configuration options.

Batch Processing - Allows you to process several netlists at once.

NOTE: If you have a hierarchical or multisheet design you must use batch mode to process the files. You cannot use PC-LINK to flatten hierarchical netlists or to link multisheet schematics, because SPICE handles hierarchy in a different way and independently of the P-CAD system.

Run NX-SPICE - Allows you to translate a P-CAD netlist into SPICE format.

Exit NX-SPICE - Returns you to DOS.

If you want to view or change the configuration, use the procedure in "Configuring NX-SPICE." If you do not want to view or change the configuration, continue to "Running NX-SPICE" and "Batch Processing."

Configuring NX-SPICE

NX-SPICE allows you to set several configuration options. You can use a configuration for just one operating session or you can save it to a configuration file that automatically sets the configuration options each time you run NX-SPICE.

Select **Configure NX-SPICE** at the Opening Menu to view or modify the current configuration.

To configure NX-SPICE, follow the steps below.

1. At the NX-SPICE Opening Menu, use the space bar or the cursor keys to select **Configure NX-SPICE**, then press [Return]. The system displays the Configuration Screen and the prompt and default value for the first configuration option, **Create Numeric Node Names**, as shown in Figure 4-2.

NX-SPICE
Configuration
Create Numeric Node Names ? : NO

Press [SPACE] for the next option; [RETURN] to accept

Figure 4-2. NX-SPICE Configuration Screen

2. This option converts nonnumeric node names to SPICE numerical node names in the <filename>.SPI file during NX-SPICE operation. NX-SPICE assigns aliases in the range 8000-8999 for local nonnumeric names and in the range 9000-9999 for global nodes (assigned with the SPGN attribute). If you have given a node a numeric name, NX-SPICE does not

assign an alias. However, you must use a number in the range 0-7999; otherwise NX-SPICE treats the number as a nonnumeric name and assigns an alias.

This option also creates the cross reference file (<filename>.XRF), which matches nonnumeric node names in the original schematic to numeric node names in the <filename>.SPI file.

Press [Return] to accept the default value, "NO" or press the spacebar and [Return] to specify "YES."

If you specify "NO," the system displays the following additional option:

Report Nonnumeric Node Names : YES

This option reports nonnumeric node names as error messages on the screen during NX-SPICE operation, and records these messages in the NXSPICE.LOG file.

Press [Return] to accept the default value "YES" or press the spacebar and [Return] to specify "NO."

3. The next prompt appears, as follows:

Correct Invalid Component Names : YES

This option attaches a valid SPICE letter prefix to invalid component names in the <filename>.SPI file. Where the original component name is more than eight characters long, the system replaces the first character with a valid SPICE character. The NXSPICE.LOG file lists any invalid names and their respective corrected names.

Press [Return] to accept the default value "YES" or press the spacebar and [Return] to specify "NO."

4. The next prompt appears, as follows:

Line Numbers : YES

This option numbers the statements in the <filename>.SPI file. P-CAD recommends using this option for the first NX-SPICE run of any netlist because flagged errors are more easily referenced by the NXSPICE.LOG file.

Press [Return] to accept the default value "YES" or press the spacebar and [Return] to specify "NO."

After you have specified the last option, the system prompts you to save the configuration, as follows:

Update Configuration File : YES

Press [Return] to specify "YES" and save the configuration to a disk file for future sessions of NX-SPICE.

Press the spacebar and then [Return] to specify "NO." The configuration is then used for the current session only.

NOTE: NX-SPICE creates a new configuration file (NXSPICE.CFG) whenever you update the configuration options. If you have modified the original configuration but want to restore it, at the DOS prompt, delete NXSPICE.CFG. When you start NX-SPICE, the program automatically creates a new configuration file containing the original default values.

Running NX-SPICE

Select **Run NX-SPICE** to process one single-sheet netlist. To run NX-SPICE, follow the steps below.

1. At the Opening Menu, use the space bar or the cursor keys to select **Run NX-SPICE** and press [Return]. The system prompts you for the name of the netlist file, as shown in Figure 4-3.

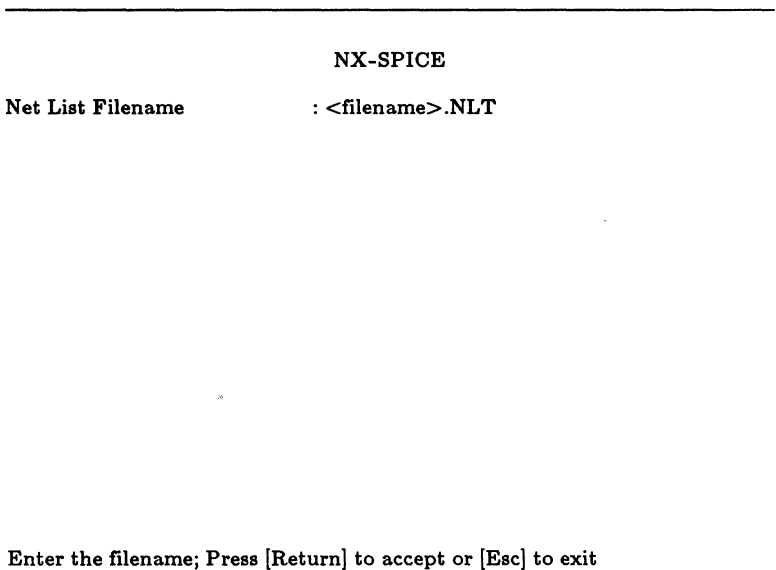


Figure 4-3. NX-SPICE Program Screen

2. Enter the name of the P-CAD netlist file to be read by NX-SPICE and press [Return]. You do not need to enter the .NLT extension; the system

automatically inserts it (see Figure 4-4), unless you specify a different extension.

NX-SPICE

Net List Filename : TESTFILE.NLT
SPICE Filename : TESTFILE.SPI

Enter the filename; Press [Return] to accept or [Esc] to exit

Figure 4-4. Sample Program Screen

3. The system prompts for the name of the SPICE-compatible output file to be created. The default is the netlist filename with the .SPI extension. Press [Return] to accept the default or enter another name and extension and press [Return].

NOTE: At either of these two fields, if you decide not to proceed with the program, you can press [Esc] to quit and return to the Opening Menu.

When you enter the output filename, NX-SPICE begins file processing. Progress reports and any error messages appear at the bottom of the screen.

When processing is complete, the system displays the Opening Menu. You can select **Exit NX-SPICE** to exit to DOS or one of the other options to process another file.

Batch Processing

Use NX-SPICE batch mode to process hierarchical and multisheet schematics, or to process several files consecutively without user intervention. A schematic design may be both hierarchical and multisheet.

NOTE: Do not use PC-LINK on hierarchical or multisheet schematics, because it flattens the databases. SPICE handles hierarchy differently and independently of the P-CAD system.

When processing several separate circuits, NX-SPICE batch mode saves processing time over command line mode (described in the next section) because NX-SPICE is only loaded into memory once.

Use a text editor to create a batch control file (<filename>.BCF). You can give the file any name you choose, but the extension must be .BCF. The batch control file lists the netlist files to be processed by NX-SPICE. See "Creating the Batch Control File" in Chapter 3 for specific information on the format for the batch control file.

To process batch control files, start NX-SPICE and then select **Batch Processing** from the NX-SPICE Opening Menu. The screen in Figure 4-5 is displayed.

NX-SPICE

Batch Control File: <filename>.BCF

Enter the filename; Press [Return] to accept or [Esc] to exit

Figure 4-5. Batch Control Menu Screen

Type the name of the batch control file, and press [Return]. NX-SPICE displays the output filename (based on the main circuit of the design) and begins processing. Processing and error messages appear at the bottom of the screen and in the NXSPICE.LOG file. When all the files specified in the batch control file have been processed, the system returns you to the Opening Menu.

If you process more than one design, messages for each design are appended to the NXSPICE.LOG file.

Press [Esc] to abort the batch job and return to DOS.

You can also run a batch control file from the DOS prompt without first starting NX-SPICE by using the at sign (@). To do so, type:

```
NXSPICE @<filename> [Return]
```

Do not leave a space between the @ and the filename. You do not have to enter the .BCF extension; NX-SPICE assumes it. When processing is complete, the system returns you to the DOS prompt.

COMMAND LINE MODE

Command line mode allows you to specify the input netlist filename and, if you want, the output filename when you start NX-SPICE.

To use command line mode, at the DOS prompt, type NXSPICE followed by the filenames, then press [Return].

The input filename is typed first, after NXSPICE, and is required. If you do not include the filename extension, NX-SPICE assumes the extension to be .NLT.

The output filename is typed next, and is optional. If you don't specify an extension, NX-SPICE assigns .SPI. If you do not specify an output filename, NX-SPICE gives the output file the same name as the input file with an .SPI extension.

Examples of command lines are shown below.

```
NXSPICE TESTFILE [Return]  
NXSPICE TESTFILE TEST [Return]
```

After you enter the command line, the program runs automatically with no further input from you. Error and processing messages appear at the bottom of the screen and in the NXSPICE.LOG file. When processing is complete, the program returns you to DOS.

You can abort program operation and return to DOS by pressing [Esc].

CHAPTER 5. VIEWING THE OUTPUT FILES

NX-SPICE produces three output files: the SPICE-compatible netlist file (<filename>.SPI), the log file (NXSPICE.LOG), and optionally the cross-reference file (<filename>.XRF).

This chapter describes how to view and print the output files and how to interpret the <filename>.SPI file.

Error messages that might be in the log file are listed and described in Appendix A. The cross-reference file (<filename>.XRF) matches nonnumeric node names in the original schematic to numeric aliases created in the <filename>.SPI file.

VIEWING AND PRINTING FILES

To display or print an output file, you can use the DOS TYPE and [Ctrl]-[P] commands. For example, to display a file, type:

```
TYPE TESTFILE.LOG [Return]
```

or, to print a file, type:

```
TYPE TESTFILE.LOG [Ctrl]-[P] [Return]
```

When the file is printed, press [Ctrl]-[P] again to turn off the print function.

You can also use the DOS PRINT command to print a file.

THE <filename>.SPI FILE

The <filename>.SPI file consists of three sections: control statements, element statements, and subcircuit definitions. These three sections are preceded by the P-CAD header, which provides copyright information and identifies the input and output files.

If you run batch mode and specify a single <filename>.SPI output file, the file contains all sections of the first file, then all sections of the second file, and so on for each file specified in the batch control file.

This section describes the information in each part of the file and tells where the information originates in the PC-CAPS schematic.

Control Statements

The first part of the file, after the P-CAD header, contains control statements and comments that apply to the schematic as a whole. Control statements begin with a period and a keyword and contain different types of information, depending on the type of statement. For example, the following is a control statement that specifies 1 and 2 as global nodes:

```
.GLOBAL 1 2
```

Control statements are generated by SPTI, SPGN, INCLx, and CARDx attributes assigned to the main circuit in SYMB mode. SPTI generates the title for the file. SPGN generates a global node statement. INCLx specifies files to be copied into the file (after the element statements). CARDx is a general attribute; the value of CARDx is passed directly into the file. The majority of the control statements, such as .LIB, .NODESET, and so on are generated by CARDx

attributes. Some control statements, such as the .WIDTH statement, are automatically generated by NX-SPICE and given default values if an appropriate CARDx attribute value is not specified in the PC-CAPS schematic.

Comment lines begin with an asterisk (*). In this part of the file comments are also generated by CARDx attributes assigned in SYMB mode to the main circuit. Comments can also be assigned to elements and subcircuit definitions.

NX-SPICE automatically generates some comment lines, such as ELEMENTS, to provide a header for a section of the file.

The plus sign (+) at the beginning of a line indicates that the line is a continuation of the previous line.

Line numbering is optional; it is controlled by the **Line Numbers** configuration option (see "Configuring NX-SPICE," in Chapter 4).

Element Statements

The next part of the file is the elements section. It contains one record for each component in the circuit. The format for an element record is as follows:

```
<instance name> <node list> <param list>
```

where:

<instance name> is assigned using the NAME/COMP command in DETL mode; or if a name is not assigned, PC-NODES assigns a name in the form UCsssnnn, where sss is the sheet

identifier, and nnn is a three-digit number assigned sequentially beginning with 000. If the **Correct Invalid Component Names** configuration option is active, NX-SPICE automatically assigns a SPICE valid prefix to the instance name; that is, X to indicate a subcircuit component, R to indicate a resistor, C a capacitor, and so on (see Table B-1 in Appendix B, NX-SPICE Symbol Library). The NXSPICE.LOG file correlates the original component names in the schematic to the SPICE-compatible names in the <filename>.SPI file.

If the **Correct Invalid Component Names** configuration option is not active, NX-SPICE appends error messages to the original component names. You must correct the component names before inputting the file to SPICE.

<node names> is the names of nodes attached to the component in the schematic. Nodes in the range 0-7999 are user assigned. If nodes are nonnumeric, and the **Create Numeric Node Names** option is active, NX-SPICE renames nodes in the 8000-8999 range, and global nodes in the 9000-9999 range.

<param list> is optional parameters and initial conditions. You assign the SPPx attribute in DETL mode to generate optional parameters to append to element statements. The value of the SPPx parameter is appended directly to the element statement.

NOTE: The element record for a subcircuit call also contains the subcircuit filename. See the examples that follow.

The following are two examples of element records:

```
XC000001 UN000002 UN000003 DS78L12
RC000004 UN000001 UN000000 1.4K 1.5
```

The first line above is a subcircuit call; it describes a subcircuit component in the schematic. Note the subcircuit filename, DS78L12. The second line describes an NX-SPICE resistor.

Subcircuit Definitions

The last part of the file contains subcircuit definitions. Subcircuit definitions are generated from subcircuit files listed in the batch control file. The format for subcircuit definitions is as follows:

* Comment line

```
.SUBCKT <filename> <nodelist> <param list>
Element statements
.ENDS
```

where:

.SUBCKT <filename> is the **.SUBCKT** keyword and the subcircuit filename.

<node list> is a list of the nodes in the subcircuit.

<param list> is a list of optional parameters and initial conditions.

Element statements is the element records that describe the components in the subcircuit. See "Element Statements" in this chapter for a description of the format for element statements.

SPPx attributes that you assign to the subcircuit symbol in SYMB mode generate the parameters and initial conditions.

The following example shows two subcircuit definitions:

```
*
* SUBCIRCUIT NOR75
*
.SUBCKT NOR75 8001 8002 8003
*
MC000000 8001 8002 0 0 N W=7.5
MC000001 8001 8003 0 0 N W=7.5
MC000002 8004 8003 8001 1 P W=7.5
MC000003 1 8002 8004 1 P W=7.5
*
*
.ENDS NOR75
*
* SUBCIRCUIT INVI4832
*
.SUBCKT INVI4832 8001 8002
*
MT2 8001 8002 0 0 N W=32
M1 1 8002 8001 1 PLT W=48
*
*
.ENDS INVI4832
*
*
.END
```

The .END keyword indicates the end of the <filename>.SPI file.

APPENDIX A. ERROR MESSAGES

This appendix lists and describes NX-SPICE error messages. Errors are grouped into two types: file related errors and processing errors.

FILE RELATED ERROR MESSAGES

Errors in this group are caused by system problems. You cannot run the program until the error is corrected.

Message: ** Database Error <filename> Database File Not Found**

Cause: The user specified netlist file does not exist.

Action: Make sure you typed in the correct filename. Verify that the file exists in the directory path specified.

Message: ** Database Error <filename> Incompatible Database**

Cause: The netlist file being processed does not have a format compatible with the output of PC-NODES.

Action: Make sure you typed in the correct filename. If the filename is correct, rerun PC-NODES to recreate the binary netlist file.

Message: ** Database Error <filename>
Disk is Full**

Cause: Your disk does not have enough space to process files.

Action: Create more disk space by deleting unnecessary files or by outputting files to an empty floppy diskette in drive A or B.

Message: ** Batch Control File Does not Exist <batch control filename>**

Cause: The user specified batch control file does not exist.

Action: Make sure you typed in the correct filename. If the filename is correct, check your directory to make sure file actually exists.

Message: ** Invalid NXSPICE.CFG file format Press Any Key to exit**

Cause: The NXSPICE.CFG configuration file is not in the format readable by the current version of NX-SPICE.

Action: Delete the existing NXSPICE.CFG file, then create a new one using the Configuration Mode of NX-SPICE.

PROCESSING ERROR MESSAGES

Errors in this group occur during processing.

Message: ** Missing Keyword 'CIRCUIT' in Batch File**

Cause: The batch control file does not start with the keyword CIRCUIT to define a circuit.

Action: Open the batch control file with a text editing program and put in the keyword CIRCUIT for each schematic design to be processed (see "Creating the Batch Control File" in Chapter 3).

Message: ** Main Circuit Name Missing in Batch File**

Cause: The first sheet of the main (root) circuit was not specified along with the keyword CIRCUIT in the batch control file.

Action: Open the batch control file with a text editing program and put in the name of the root circuit file on the line following the keyword CIRCUIT (see "Creating the Batch Control File" in Chapter 3).

Message: ** Spice File (<filename>) Generation Failed (rc)**

Cause: NX-SPICE was unable to complete processing. Another specific error message should have been issued prior to this message.

Action: Review the other error messages issued.

Message: ** Memory Allocation Error**

Cause: Insufficient disk space is available or the disk environment is incorrect.

Action: Verify that the diskette has adequate available memory space, and use the DOS CHKDSK command to verify that the disk is not bad.

Message ** Invalid Block pointer
Free Failed**

Cause: An attempt to free disk memory failed.

Action: Test the disk with the DOS CHKDSK command.

Message: *** Error creating <modulename> Module

Cause: An error occurred while this file <modulename> was being processed; another message should have been issued prior to this.

Action: Review the previous error messages.

Message: ** Database Error <XXX>**

Cause: Accessing part of the binary netlist was unsuccessful. <XXX> is internal information to P-CAD that flags the location of the error.

Action: Rerun PC-NODES; if the error persists, verify the completeness and integrity of the original schematic.

Message: * Incorrect Component Type (<component type>) Processing Terminated**

Cause: The component type of a hierarchical subcircuit symbol (or of main circuit) is not 256.

Action: Verify that the component attribute number of the particular schematic is 256. If necessary, use the SCMD/SCAT command in SYMB mode to assign a value of 256 to the circuit or subcircuit.

Message: * No Value given for attribute <attribute keyword>**

Cause: The named attribute has no value.

Action: Verify the existence of the attribute value in the schematic. If the attribute value is missing, modify or add the attribute with the ATTR/SCHG or ATTR/ACOM commands, respectively.

Message: ** String Too long <string>**

Cause: The string of characters being written to the <filename>.SPI file is too long for NX-SPICE. The complete string is shown in the NXSPICE.LOG file.

Action: Limit all attribute entries in your schematic to less than 100 characters (especially CARDx.1-x.4 sets).

Message: ** Unconnected Node Detected**

Cause: An unconnected pin was detected.

Action: See the <filename>.SPI file for the entry that has an unconnected node error message. Verify that the pin is connected on the schematic.

Message: ** Invalid Node Name : <nodename>
See .LOG file**

Cause: A nonnumeric node name is encountered and NX-SPICE is configured to **Create Numeric Node Names** in the configuration. The possible cause is an unnamed net.

Action: Either rename the node using NAME/NET and make sure all nodes are named; or modify your NX-SPICE configuration by selecting **YES** for **Create Numeric Node Names**.

Message: ** Invalid Initialization Attribute : <attr value>**

Cause: The format of the SPNS or SPIC attributes is not as specified for creating .NODESET and .IC values.

Action: These attributes are present for compatibility with 1.24 databases. Reenter these attributes as SPPx attributes.

Message: Dependent Source Error
See .SPI file

Cause: A voltage controlled dependent source was specified with an incorrect number of controlling nodes with the SPVC attribute.

Action: The SPVC attribute is present only for compatibility with 1.24 databases. Reenter the controlling nodes as SPPx attributes.

Message: ** Invalid Element Name : <element name>**

Cause: An element name has an incorrect first character.

Action: Rename schematic symbol for the element using NAME/COMP or configure NX-SPICE to automatically correct invalid component names.

Message: ** Invalid Element Type**
See .SPI file

Cause: A primitive symbol type is set to a value other than the allowable NX-SPICE element type.

Action: Change the symbol component attribute type to a legal value (see Appendix B, "NX-SPICE Symbol Library") with the SCMD/SCAT command in PC-CAPS. Then run PC-NODES to reextract the netlist.

Message: Invalid Mutual Inductor : <MI Specification>

Cause: The SPMI attribute assigned for mutual inductor statements is not set in the form allowable by NX-SPICE.

Action: Correct the value in your schematic using the ATTR/SCHG command of PC-CAPS. See "SPMI (Mutual Inductor)" in Chapter 2. Then run PC-NODES to reextract the netlist.

APPENDIX B.NX-SPICE SYMBOL LIBRARY

This appendix provides a listing of the symbols provided with NX-SPICE.

Table B-1 lists the symbols and their corresponding type ID, SPICE type prefix, and filename.

Table B-1. HSPICE Elements

TYPE ID	ELEMENT TYPE	PREFIX	.SYM NAME
9	Resistor	R	RES.SYM
86	Capacitor	C	CAP.SYM
87	Nonlinear capacitor	C	CAPNL.SYM
114	Inductor	L	IND.SYM
115	Nonlinear inductor	L	INDNL.SYM
*	Mutual inductor	K	
88	Transmission line	T	TLINE.SYM
89	Junction diode	D	DIODE.SYM
90	NPN BJT	Q	NPN.SYM
90	NPN BJT (substrate)	Q	NPNS.SYM
91	PNP BJT	Q	PNP.SYM
91	PNP BJT (substrate)	Q	PNPS.SYM
93	N-channel JFET/MESFET	J	NJFET.SYM
94	P-channel JFET/MESFET	J	PJFET.SYM
16	N-channel MOSFET	M	NMOS.SYM
16	N-channel MOSFET (bulk)	M	NMOSB.SYM
17	P-channel MOSFET	M	PMOS.SYM
17	P-channel MOSFET (bulk)	M	PMOSB.SYM
113	Ind. voltage source AC	V	VDC.SYM
113	Ind. voltage source DC	V	VAC.SYM
113	Ind. voltage source (pulse)	V	VPULSE.SYM
113	Ind. foltage source (sinusoidal)	V	VSIN.SYM

* specify mutual inductors with SPMI attribute.

Table B-1 Continued

TYPE ID	ELEMENT TYPE	PREFIX	.SYM NAME
113	Ind. voltage source (exponential)	V	VEXP.SYM
113	Ind. voltage source (piecewise linear)	V	VPWL.SYM
113	Ind. voltage source FM	V	VFM.SYM
95	Ind. current source AC	I	IAC.SYM
95	Ind. current source DC	I	IDC.SYM
95	Ind. current source (pulse)	I	IPULSE.SYM
95	Ind. current source (sinusoidal)	I	ISIN.SYM
95	Ind. current source (exponential)	I	IEXP.SYM
95	Ind. current source (piecewise linear)	I	IPWL.SYM
95	Ind. current source FM	I	IFM.SYM
96	VCCS (linear)	G	VCCS.SYM
96	VCCS (nonlinear)	G	VCCSNL.SYM
97	VCVS (linear)	E	VCVS.SYM
97	VCVS (nonlinear)	E	VCVSNL.SYM
98	CCCS (linear)	F	CCCS.SYM
98	CCCS (nonlinear)	F	CCCSNL.SYM
99	CCVS (linear)	H	CCVS.SYM
99	CCVS (nonlinear)	H	CCVSNL.SYM
256	<subcircuit>	X	

SYMBOL GRAPHICS

This section shows the SPICE primitive graphics. Symbol filename, type ID, SPICE prefix, pin ordering, and predefined attributes, if any, are listed.

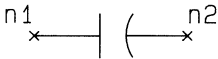
RESISTOR
RES.SYM



Type ID: 9
Prefix: R
Pin Order: n1,n2
Predefined Attributes:

SPP1="VALUE"

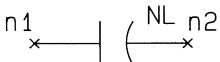
CAPACITOR
CAP.SYM



Type ID: 86
Prefix: C
Pin Order: n1,n2
Predefined Attributes:

SPP1="VALUE"

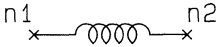
CAPACITOR
(NONLINEAR)
CAPNL.SYM



Type ID: 114
Prefix: C
Pin Order: n1,n2
Predefined Attributes:

SPP1="POLY"
SPP2="COEFF"

**INDUCTOR
IND.SYM**

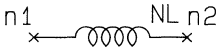


Type ID: 114
Prefix: L
Pin Order: n1,n2
Predefined Attributes:

SPP1="VALUE"



**INDUCTOR (NONLINEAR)
INDNL.SYM**



Type ID: 115
Prefix: L
Pin Order: n1,n2
Predefined Attributes:

SPP1="POLY"
SPP2="COEFF"



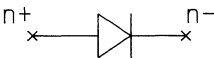
**TRANSMISSION LINE
TLINE.SYM**



Type ID: 88
Prefix: T
Pin Order: n1,n2,n3,n4
Predefined Attributes:

SPP1="Z0=VALUE"

**DIODE
DIODE.SYM**

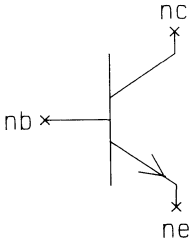


Type ID: 89
Prefix: D
Pin Order: n+,n-
Predefined Attributes:

SPP1="MODEL"



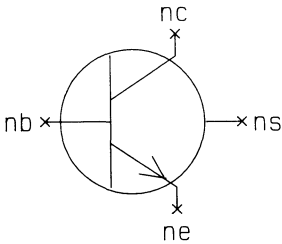
NPN BJT
NPN.SYM



Type ID: 90
Prefix: Q
Pin Order: nc,nb,ne
Predefined Attributes:

SPP1="MODEL"

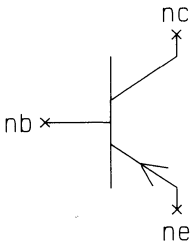
NPN BJT (SUBSTRATE)
NPNS.SYM



Type ID: 90
Prefix: Q
Pin Order: nc,nb,ne,ns
Predefined Attributes:

SPP1="MODEL"

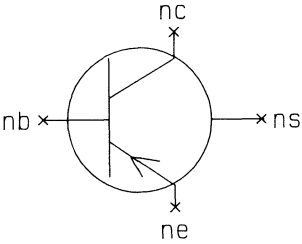
PNP BJT
PNP.SYM



Type ID: 91
Prefix: Q
Pin Order: nc,nb,ne
Predefined Attributes:

SPP1="MODEL"

**PNP BJT (SUBSTRATE)
PNPS.SYM**

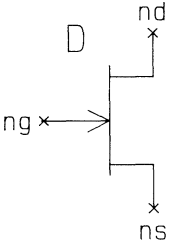


Type ID: 91
Prefix: Q
Pin Order: nc,nb,ne,ns
Predefined Attributes:

SPP1="MODEL"



**N-CHANNEL JFET
NJFET.SYM**

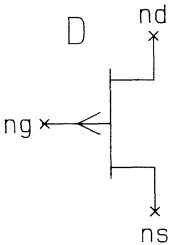


Type ID: 93
Prefix: J
Pin Order: nd,ng,ns
Predefined Attributes:

SPP1="MODEL"



**P-CHANNEL JFET
PJFET.SYM**

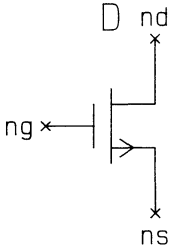


Type ID: 94
Prefix: J
Pin Order: nd,ng,ns
Predefined Attributes:

SPP1="MODEL"



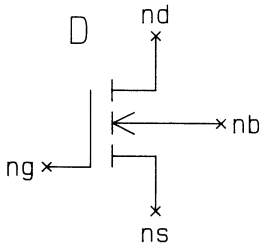
**N-CHANNEL MOSFET
NMOS.SYM**



Type ID: 16
Prefix: M
Pin Order: nd,ng,ns
Predefined Attributes:

SPP1="MODEL"

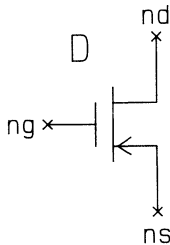
**N-CHANNEL MOSFET
(BULK)
NMOSB.SYM**



Type ID: 16
Prefix: M
Pin Order: nd,ng,ns,nb
Predefined Attributes:

SPP1="MODEL"

**P-CHANNEL MOSFET
PMOS.SYM**

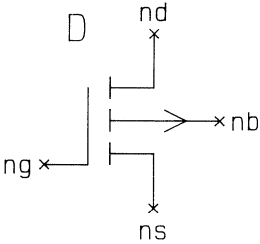


Type ID: 17
Prefix: M
Pin Order: nd,ng,ns
Predefined Attributes:

SPP1="MODEL"

P-CHANNEL MOSFET
(BULK)

PMOSB.SYM



Type ID: 17

Prefix: M

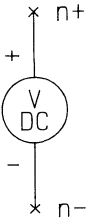
Pin Order: nd,ng,ns,nb

Predefined Attributes:

SPP1="MODEL"

IND VOLTGE SOURCE
(DC)

VDC.SYM



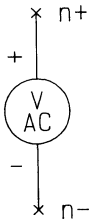
Type ID: 113

Prefix: V

Pin Order: n+,n-

IND VOLTGE SOURCE
(AC)

VAC.SYM



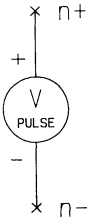
Type ID: 113

Prefix: V

Pin Order: n+,n-

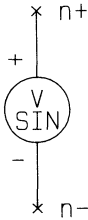
IND VOLTGE SOURCE
(PULSE)
VPULSE.SYM

Type ID: 113
Prefix: V
Pin Order: n+,n-



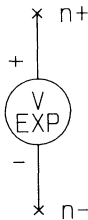
IND VOLTGE SOURCE
(SINUSOIDAL)
VSIN.SYM

Type ID: 113
Prefix: V
Pin Order: n+,n-



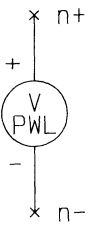
IND VOLTGE SOURCE
(EXPONENTIAL)
VEXP.SYM

Type ID: 113
Prefix: V
Pin Order: n+,n-



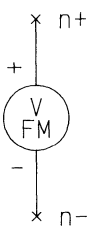
IND VOLTGE SOURCE
(PIECEWISE LINEAR)
VPWL.SYM

Type ID: 113
Prefix: V
Pin Order: n+,n-



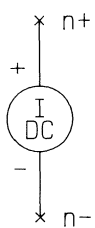
IND VOLTGE SOURCE
(FM)
VFMSYM

Type ID: 113
Prefix: V
Pin Order: n+,n-



IND CURRENT SOURCE
(DC)
IDC.SYM

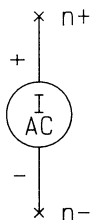
Type ID: 95
Prefix: I
Pin Order: n+,n-



IND CURRENT SOURCE
(AC)

IAC.SYM

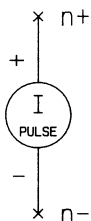
Type ID: 95
Prefix: I
Pin Order: n+,n-



IND CURRENT SOURCE
(PULSE)

IPULSE.SYM

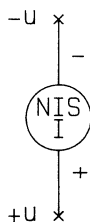
Type ID: 95
Prefix: I
Pin Order: n+,n-



IND CURRENT SOURCE
(SINUSOIDAL)

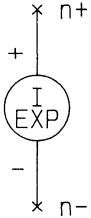
ISIN.SYM

Type ID: 95
Prefix: I
Pin Order: n+,n-



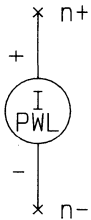
IND CURRENT SOURCE
(EXPONENTIAL)
IEXP.SYM

Type ID: 95
Prefix: I
Pin Order: n+,n-



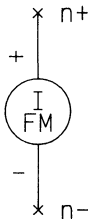
IND CURRENT SOURCE
(PIECEWISE LINEAR)
IPWL.SYM

Type ID: 95
Prefix: I
Pin Order: n+,n-

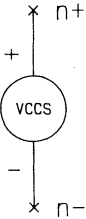


IND CURRENT SOURCE
(FM)
IFM.SYM

Type ID: 95
Prefix: I
Pin Order: n+,n-



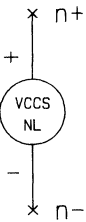
VCCS
VCCS.SYM



Type ID: 96
Prefix: G
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROLS"
SPP2="VALUE"

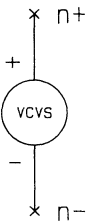
VCCS (NONLINEAR)
VCCSNL.SYM



Type ID: 96
Prefix: G
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROLS"
SPP2="COEFF"

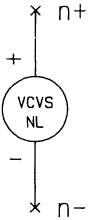
VCVS
VCVS.SYM



Type ID: 97
Prefix: E
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROLS"
SPP2="VALUE"

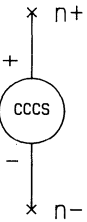
VCVS (NONLINEAR)
VCVSNL.SYM



Type ID: 97
Prefix: E
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROLS"
SPP2="COEFF"

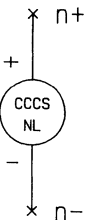
CCCS
CCCS.SYM



Type ID: 98
Prefix: F
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROL"
SPP2="VALUE"

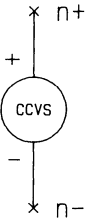
CCCS (NONLINEAR)
CCCSNL.SYM



Type ID: 98
Prefix: F
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROL"
SPP2="COEFF"

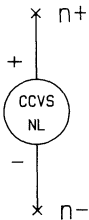
CCVS
CCVS.SYM



Type ID: 99
Prefix: H
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROL"
SPP2="VALUE"

CCVS (NONLINEAR)
CCVSNL.SYM



Type ID: 99
Prefix: H
Pin Order: n+,n-
Predefined Attributes:

SPP1="CONTROL"
SPP2="COEFF"

