# NX-TDL

## User's Manual

August 1986

p-cad®
PERSONAL CAD SYSTEMS INC.

000-0061-03

# NX-TDL

## User's Manual

August 1986

p-cad®
PERSONAL CAD SYSTEMS INC.

000-0061-03

## COPYRIGHT



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

TEGAS and TDL (TEGAS Design Language) are trademarks of General Electric Company, Calma Division.

# ABOUT THIS MANUAL

This manual describes NX-TDL, a program that converts P-CAD schematic database files to ASCII files compatible with the TEGAS-5 simulator system.

**Chapter 1, INTRODUCTION**, provides an overview of NX-TDL 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-TDL.

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

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

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

**Appendix A, ERROR MESSAGES**, describes error messages that may be produced during NX-TDL program operation.

**Appendix B, NX-TDL SCHEMATIC SYMBOLS**, describes and shows the primitive symbols supplied with NX-TDL.

# 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

## CONTENTS (Continued)



## FIGURES



## TABLES

# CHAPTER 1. INTRODUCTION

The NX-TDL interface program translates netlists from P-CAD circuit schematics to files in the TEGAS Design Language (TDL). These files can then be input to the TEGAS-5 or TEXSIM simulation system using the TDL preprocessor.

This chapter provides an overview of NX-TDL and installation instructions.

## OVERVIEW

NX-TDL, together with PC-NODES, enables you to produce a TEGAS-compatible file from a schematic created with PC-CAPS.

There are five 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 symbols (<filename>.SYM and <filename>.SCH). The types of symbols you can use in your schematic are described in Chapter 2, "Preparing the Schematic Database."

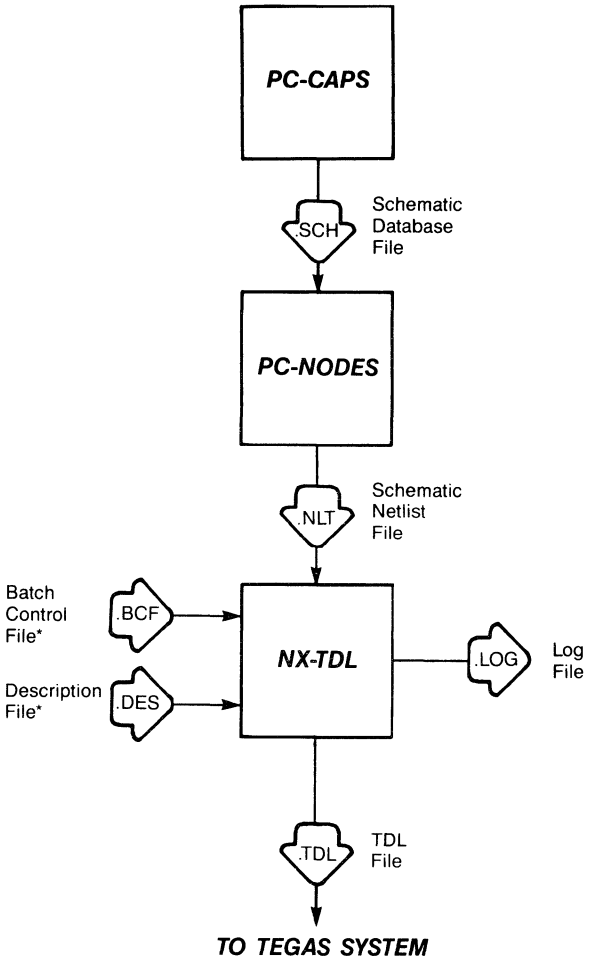
2. Using PC-NODES, you extract a netlist file (<filename>.NLT) from each schematic file. The netlist file contains component and interconnection information for the schematic.

3. If your design contains more than one schematic file, you next use a text editor to create a batch control file (<filename>.BCF). This file contains the names of all the netlist files to be processed.

4. If you want to add description data for any file in the design, you create a corresponding description file (<filename>.DES).

5. You then input the netlist file or batch control file to NX-TDL. NX-TDL translates the netlist(s) into the TDL file (<filename>).TDL), using any applicable description files. It also outputs a log file (<filename>.LOG), which contains all messages displayed during the execution of NX-TDL. This file is useful for examining any error messages that are reported by NX-TDL.

Figure 1-1 shows the process described above.




*Optional File

**Figure 1-1. NX-TDL Input and Output**

## SYSTEM REQUIREMENTS

Before you install NX-TDL, 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

  NOTE: Other P-CAD programs do not require a FILES value of 20. For further information on the CONFIG.SYS file, see the DOS Reference chapter of the *PC-CAPS User's Manual.*

- 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-TDL diskette contains a library of TEGAS-compatible primitive symbols, the NX-TDL program file, NXTDL.EXE, and the following sample files:

TDLDEMO.SCH
TDLDEMO.NLT
TDLDEMO.TDL

You can use the sample data to see how the program works. "Sample TDL File" in Chapter 5 shows the sample schematic and TDL file.

To install NX-TDL 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 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]

Copy the sample data files to your working project directory. For example, if you are using the PROJ0 directory, change to that directory by typing:

CD \PCAD\PROJ0 [Return]

Then copy the files by typing:

COPY A:TDLDEMO.* [Return]

Create a new directory for the TDL symbol library by typing:

MD \PCAD\SYM\TDL [Return]

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

CD \PCAD\SYM\TDL [Return]

Then copy the files by typing:

COPY A:*.SYM [Return]

# CHAPTER 2. PREPARING THE SCHEMATIC DATABASE

Before using NX-TDL, 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 TEGAS-compatible schematic. To ensure that your database is compatible with TEGAS, you should read this chapter before creating 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 schematic component. For example, a 7400 symbol is a single 2-input NAND gate.

With NX-TDL, P-CAD supplies a library of symbols that correspond to TEGAS primitives. You can use these symbols as they are, modify them, modify and use symbols from one of P-CAD's schematic symbol libraries, or create your own symbols. Before modifying or creating symbols, read the NX-TDL requirements in the following sections and in Appendix B, "NX-TDL Schematic Symbols."

Symbols are created in PC-CAPS SYMB mode as files with the .SYM filename extension. You assemble symbols into a schematic in DETL mode, producing a schematic database file with a .SCH filename extension.

NX-TDL supports hierarchical and multisheet schematic structure. These structures are described below.

## Hierarchical Structure

Hierarchical structure is the usage of symbols to represent subcircuits of a schematic. With PC-CAPS, you can create a symbol to represent a subcircuit, then use the symbol to represent the subcircuit in a higher level schematic. Each subcircuit is stored as a separate file.

You can use NX-TDL to handle hierarchical structure in either of two ways or in a combination of the two.

First, you can process each subcircuit individually (lowest levels first), compile the resulting TDL files into TEGAS modules, and store them in a TEGAS library. You can then use a symbol for the subcircuit in a higher-level PC-CAPS schematic. If you use this method for all hierarchical elements in a design, you can input the top level to NX-TDL as a single file.

Alternately, you can submit all the files at all levels of hierarchy to NX-TDL together, using a batch control file. For each file in the design, NX-TDL generates a MODULE command and its associated commands in the Compiler block of the output TDL file. It also generates a Linker block in the TDL file. You can then link the files in the TEGAS system. (Refer to Chapter 3, "Preparing the Input Files," for information about batch control files.)

When you create a subcircuit, use the instructions in the *PC-CAPS User's Manual* to create a symbol and enter input and output pins. If you intend to input the circuit to NX-TDL as a subcircuit of a higher-level design, use the PC-CAPS SCMD/SCAT command to assign a component type ID of 256.

### Multisheet Structure

A multisheet schematic is created by producing a number of single-sheet schematics. Connections between sheets are indicated by giving connected wires the same net name on each sheet. The P-CAD system allows up to 100 sheets in a multisheet schematic.

NX-TDL supports multisheet schematics at all levels of the design hierarchy. You must use a batch control file to input a multisheet design into NX-TDL. (Refer to Chapter 3, "Preparing the Input Files," for information on batch control files.)

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

SHEET=<sheet id>

where:

**sheet id** is one to three characters and is typically a number, with the first sheet being sheet 1 and so on. Sheet IDs in a multisheet schematic must be unique.

This attribute is used by PC-NODES to assign names to unnamed components and nets.

## CREATING TEGAS-COMPATIBLE SYMBOLS

The primitive symbols provided with NX-TDL are described in Appendix B, "NX-TDL Schematic Symbols." You can create more schematic symbols as you need them, provided that corresponding TEGAS primitives or modules exist and that you assign the appropriate

component type IDs, pin numbers, and pin order as described in the following sections. By P-CAD convention, each symbol file has the .SYM filename extension.

### Component Type ID

The component type ID is a number that identifies the component to NX-TDL. NX-TDL uses Table B-1 in Appendix B to find the name of the TEGAS primitive that corresponds to the number. The component type is used in several places in the output TDL file.

The symbols supplied with NX-TDL are already assigned the appropriate component type IDs. If you use any other symbols, you must be sure they correspond to a TEGAS type named in Table B-1 and that you assign the correct ID. To assign the component type ID, use the SCMD/SCAT command in PC-CAPS SYMB mode.

### Pin Order and Pin Names

Use the ENTR/PIN command in SYMB mode to enter each pin for a symbol. The P-CAD pin order is the order in which the pins are entered on the component. Enter the pins in the order used for the corresponding TEGAS primitive and/or use the same pin names as the TEGAS primitive, with the qualifications given below.

- If you run NX-TDL with the **Signal=Pin** configuration option set to "No" (the default), pin names are not used in the output TDL file, so they do not need to be the same as the TEGAS pin names. In this case, the pin order must be the same as the TEGAS pin order.

- If you run NX-TDL with the **Signal=Pin** configuration option set to "Yes," pin names are used in the output TDL file and must be the same as the TEGAS pin names. In this case, the pin order is not significant, so it does not have to be the same as the TEGAS pin order.

## ENTERING TERMINAL PINS

NX-TDL requires you to specify the terminal pins in all schematics in either of the following two ways: creating a symbol in SYMB mode, or entering a component in DETAIL mode.

### Creating a Symbol in SYMB Mode

You can specify terminal pins by creating a symbol for the entire schematic in PC-CAPS SYMB mode. Use the ENTR/PIN command to enter pin information.

For a multisheet schematic, create the symbol on the first sheet only. (The first sheet is the first file listed in the batch control file.)

### Entering a Component in DETAIL Mode

The library supplied with NX-TDL includes three special pin components (symbols). Use the ENTR/COMP command in DETL mode to attach these pin components to the terminal wires of the circuit. The filenames and pin types of these pin components are shown below:

PIN$IN.SYM input terminals
PIN$OUT.SYM output terminals
PIN$IO.SYM I/O terminals

You can use these pin components for only the top level of a design; for a schematic that is to be used as a subcircuit you must enter pins in SYMB mode as described above. For a multisheet schematic, you can enter pin components on as many sheets as necessary.

> NOTE: You cannot use both of the above methods to specify the terminal pins in the same top-level schematic design.

## ASSIGNING ATTRIBUTES

NX-TDL uses several attributes as information in the output TDL file. This section describes and summarizes these attributes. This section also gives the function of each attribute in the TDL file. Refer to the TEGAS documentation for further information on TDL syntax conventions.

None of the attributes listed are required. If you do not assign the attribute, the corresponding field in the output file will be empty, unless otherwise noted below.

To assign an attribute, select the PC-CAPS ATTR/ACOM command in either SYMB or DETL mode as specified below. Type the attribute in the format:

> key=value

where:

> **key** is the attribute keyword that names the attribute.
>
> **value** is the contents of the attribute. If the attribute value contains any blank spaces, it must be enclosed in quotation marks.

### Attributes Assigned to a Schematic

The following attributes are assigned to an entire schematic in SYMB mode. The schematic can be the top level of a design, a subcircuit to be used in other designs, or the first sheet of a multisheet design.

**TLDN** - Specifies the default TDL directory name for use with the TDL file DIRECTORY command.

Example:

```
TLDN=LSTTL
```

**TLCO** - Specifies TDL compiler options for use with the TDL file OPTIONS command. If this attribute is not assigned, NX-TDL uses a value of REPLACE. If you assign a value of "NONE," NX-TDL will not produce the OPTIONS command.

Example:

```
TLCO="REP, NC = 1, XREF"
```

**TLCN** - Specifies a component name to replace the filename of the component in the TDL file. This attribute is useful if the desired component name does not meet DOS requirements (for example, it may be more than eight characters long) or if you want two components stored in separate files to have the same name.

Example:

```
TLCN=VERYLONGNAME
```

**TLMV** - Specifies module version for use with the MODULE command in the TDL file.

Example:

```
TLMV=VER2
```

**TLMD** - Specifies module directory for use with the MODULE command in the TDL file. This attribute is necessary only if you want to specify for a subcircuit a different directory than the default specified with the TLDN attribute.

Example:

```
TLMD=RSY
```

**TLLV** - Specifies the level name of the circuit for use with the LEVEL command in the TDL file.

Example:

```
TLLV=LEVEL2
```

**TLDY** - Specifies a delay statement for use with the DELAYS command in the TDL file. You can assign this attribute more than once to specify more than one delay statement.

Example:

```
TLDY="DEL1/1,2,3,4,5,6/"
TLDY="NOT/DEL1/"
```

**TLWG** - Specifies a wired gate type name (alias) for use with the USE command in the TDL file. You can assign this attribute more than once to name more than one wired gate in the circuit.

Example:

```
TLWG="WIREDOR = WOR(2,1 + SETZZ)"
TLWG="WAND3 = WAND(3,1)"
```

**TLWS** - Specifies a value to be shown with the WIRED command in the TDL file. You can assign this attribute more than once.

Example:

```
TLWS="DEFAULT WIREDOR"
TLWS="USE WAND3 A,B,C"
```

### Attributes Assigned to Components

The following attributes are assigned to individual components (primitives or subcircuits). Except for the TLCN attribute, these attributes are normally assigned in DETL mode to an instance of a component on a schematic. You can also assign these attributes in SYMB mode to a symbol definition so that the attribute will apply to all instances of the symbol whenever it is used in a schematic.

**TLCN** - Specifies a component name to be used in place of the filename in the TDL file. This attribute is assigned in SYMB mode and is used in the same way for a primitive as for a schematic.

**TLPO** - Specifies one or more primitive options to be shown for the primitive after the USE command in the TDL file. You can assign more than one option either by including more than one option in the same attribute value or by assigning the attribute more than once.

Example:

TLPO="COMPPC COMPPS"

**TLCD** - Specifies component delay to be shown for the primitive after the USE command in the TDL file.

Example:

TLCD="/1,2,3/"

**TLPD** - Specifies delays for the output pins of the component for use in the output pin list for the component instance after the DEFINE command in the TDL file. This attribute specifies delays for all the output pins in the component. NX-TDL places the entry for each pin after the corresponding pin name in the TDL file.

The format for a delay value for one pin is:

/<delays>/

The format for delay values for more than one pin is the same as above for each pin, with or without a comma between values. For example, for two pins, the format is either of the following:

/<delays>/,/<delays>/
/<delays>//<delays>/

Delays must be listed using the pin order in which the pins were entered, so that each delay will be assigned to the correct pin in the output pin list. If no delay value is needed for a pin but delay values are needed for subsequent pins, its' place in the list must be marked by two commas (except for before the first delay value given, where one comma holds the place for each previous pin).

The following examples are all valid attribute values for a 5-output component.

TLPD=",/1/,/3/,,"

This value specifies delays on pins 2 and 3.

TLPD=",/1,2/,,/1,2,3/,"

This value specifies delays on pins 2 and 4.

TLPD=",,/1,2,3,4,5,6/"

This value specifies the delay on pin 3 only.

**TLOD** - Specifies delays for the component instance for use as the last entry after the DEFINE command in the TDL file.

Example:

TLOD="/5/"

### Attribute Summary

Table 2-1 lists all the attributes recognized by NX-TDL and gives the TDL file commands to which each attribute applies.

**Table 2-1. Attribute Summary**

| Attribute | Description | TDL Command |
|---|---|---|
| TLCD | Component delay | USE |
| TLCO | TDL compiler options | OPTIONS |
| TLCN | Component name | MODULE<br>USE<br>DEFINE<br>LINK |
| TLDN | TDL directory | DIRECTORY |
| TLDY | Delay definition | DELAYS |
| TLLV | Level | LEVEL |
| TLMD | Module directory | MODULE<br>LINK |
| TLMV | Module version | MODULE<br>LINK |
| TLOD | Occurrence delay | DEFINE |
| TLPD | Primitive pin delay | DEFINE |
| TLPO | Primitive option | USE |
| TLWG | Wired gate alias | USE |
| TLWS | Wired statement | WIRED |

# CHAPTER 3. PREPARING THE INPUT FILES

This chapter describes how to prepare the files that are to be input to NX-TDL.

After your schematic is complete, you must use PC-NODES to extract the schematic netlist(s). If your schematic consists of just one schematic file, you can input the netlist directly into NX-TDL.

If your design has multiple sheets or if it is hierarchical and you want NX-TDL to process all levels into one TDL file, you must create a batch control file to input into NX-TDL. You can also use a batch file to process several designs consecutively.

> NOTE: Do not use PC-LINK to flatten or expand a hierarchical or multisheet design. The expanded netlist output by PC-LINK is not compatible with the TEGAS system.

If you want to include any text in the DESCRIPTION section of the output TDL file, you must create a description file.

The following sections explain how to create the schematic netlist, the batch control file, and the description file.

## EXTRACTING THE SCHEMATIC NETLIST

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

## CREATING THE BATCH CONTROL FILE

The batch control file consists of one or more commands, each of which tells NX-TDL to process a design that may be hierarchical and/or multisheet. Use a text editor to create the batch control file. A batch control file normally has the .BCF filename extension, but you can use any extension.

The format of each batch control file command is shown below. The batch control file contains one command for each design to be processed.

```
OUTFILE <outfile>
<mainfile>
<module>
<module>
<module>
.
.
<module>
```

where:

**outfile** is the name of the output file. If no name is specified, NX-TDL will assign the name of the input netlist file with the .TDL extension. (For a multisheet design, the name of the first netlist file is used.)

**mainfile** is the netlist filename of the highest level of the design. It must begin on a separate line after the output file information. For a multisheet design, all filenames are listed and filenames are separated by a blank space. A plus sign (+) after a blank space at the end of a line indicates that the list continues on the next line. If you do not enter the filename extension, NX-TDL assumes it is .NLT.

**module** is the netlist filename of a subcircuit used in the design. If the subcircuit has more than one sheet, all filenames are listed and filenames are separated by a blank space. A plus sign (+) after a blank space at the end of a line indicates that the list continues on the next line. If you do not enter the filename extension, NX-TDL assumes it is .NLT. Each module name must be listed on a separate line. If the design has no subcircuits, no module names are listed.

You can use the percent sign (%) in the batch control file to indicate a comment. NX-TDL ignores all text from the % to the end of the line.

Figure 3-1 shows an example of a batch control file.

```
OUTFILE TEST1.OUT   % one-sheet nonhierarchical design with
TEST1               % the output file named TEST1.OUT
%
OUTFILE             % five-sheet design with the default
SH1 SH2 SH3 +       % output filename SH1.TDL
SH4 SH5
%
OUTFILE             % hierarchical design with three
DESIGN              % subcircuits, one of which has two
SUB1                % sheets, and the default output filename
SUB2A SUB2B         % DESIGN.TDL
SUB3
```

**Figure 3-1. Sample Batch Control File**

"Running a Batch File" and "Command Line Mode" in Chapter 4 describe the two ways of inputting the batch control file into NX-TDL.

## CREATING THE DESCRIPTION FILE

If you want to specify text to be included in the DESCRIPTION section of the output TDL file, you must use a text editor to create a description file containing the text. This file must have the same filename as the netlist file with the .DES extension. (If the design is multisheet, the name must be that of the first sheet named in the batch control file.)

Each line in the description file should be no longer then 60 characters. If a line is longer, NX-TDL uses only the first 60 characters. The entire file can be no longer than 1023 characters; if it is longer, NX-TDL uses only the first 1023 characters.

# CHAPTER 4. USING NX-TDL

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

Before running NX-TDL, be sure that:

- Your system is correctly configured.
- You have installed the NX-TDL program file (NXTDL.EXE).
- You have assembled the schematic circuit.
- You have extracted the input netlist files and, if necessary, created the batch control file.

NX-TDL has two operating modes, interactive and command line. In either mode, you can process a single netlist or a batch control file. In interactive mode, NX-TDL displays a series of screens and you select options and specify filenames. In command line mode, you specify filenames and options on the command input line when you start the program and NX-TDL processes the input file and produces the desired output file automatically.

The following sections describe how to use interactive mode to start, configure, and run NX-TDL and how to use command line mode.

## INTERACTIVE MODE

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

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

NXTDL [Return]

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

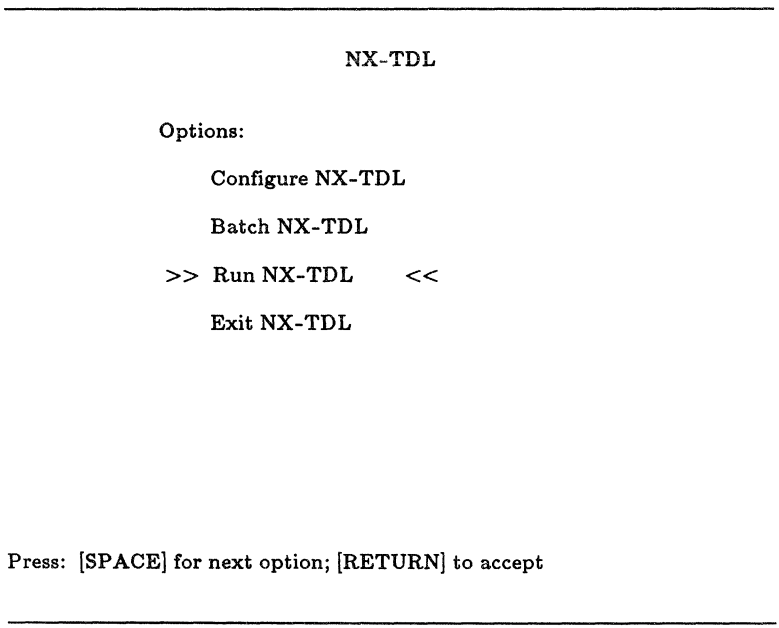



**Figure 4-1. NX-TDL Opening Menu**

This menu provides the following options.

**Configure NX-TDL** - Allows you to set or change NX-TDL configuration options.

**Batch NX-TDL** - Allows you to translate one or more designs using a batch control file.

**Run NX-TDL** - Allows you to translate a single P-CAD netlist to a TDL file.

**Exit NX-TDL** - Returns you to DOS.

If you want to set or change the configuration, use the procedure in "Configuring NX-TDL." If you do not want to set or change the configuration, continue to "Running NX-TDL."

## Configuring NX-TDL

NX-TDL allows you to set several options. These settings are stored in the configuration file, NXTDL.CFG, which is used each time you run the program.

If you do not create the configuration file before the first time you run NX-TDL, the system creates the configuration file using the default settings.

You can use **Configure NX-TDL** to create the configuration file before the first time you run NX-TDL, to change the settings in the configuration file, or to change the settings for just one operating session.

To configure NX-TDL, follow the steps below.

1. At the NX-TDL Opening Menu, press the space bar or use the cursor keys to select **Configure NX-TDL**, then press [Return]. The system displays the Configuration Screen with the default settings as shown in Figure 4-2.

```
                              NX-TDL

                           Configuration

CFG File Name          : NXTDL.CFG

Bidirect               : No

No Connect Value       : 0
Bar Character          : -

Signal = Pin ?         : No
USE Block              : User Modules Only




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

**Figure 4-2. NX-TDL Configuration Screen**

2. The first configuration option allows you to specify how I/O terminal pins are handled. The settings are "No" and "Yes."

The "No" setting causes NX-TDL to enter any I/O pins in a circuit in the OUTPUT statement of the TDL file, to enter a BIDIRECT command in the linker block, and not to enter a BIDIRECT command in the compiler block.

The "Yes" setting causes NX-TDL to enter I/O pins in the INPUT and OUTPUT statements and to enter a BIDIRECT command in the compiler block. The BIDIRECT command will not be given in the linker block.

The default setting is "No." Press [Return] to accept the default or press the space bar to display the "Yes" setting and press [Return].

3. The next option allows you to assign the setting for unconnected input terminal pins. There are five settings: "0" for low, "1" for high, "X" for undefined, "Z" for high impedance, and "(none)."

   The default setting is "0." Press [Return] to accept the default, or press the space bar as many times as necessary to display the desired setting, and then press [Return].

4. The next option allows you to select a character to replace the apostrophe to represent a bar. In PC-CAPS, any name terminated with an apostrophe is displayed with a bar over the entire name. You might want to use a different character to represent the bar in your TDL netlist. Valid characters are: letters, numbers, "-", and "." .

   The default character is "-." Press [Return] to accept the default or type a different character and press [Return].

5. The next option allows you to specify whether or not to include the component pin names with the signal names in the DEFINE section of the TDL file. The settings are "No" and "Yes." The default is "No."

   Press [Return] to accept the default, or press the space bar to display "Yes" and press [Return].

6. The next option allows you to specify which component types are included in the USE block of the TDL file. The settings are "User Modules Only" and "All Components."

   The "User Modules Only" setting causes NX-TDL to include all primitives and to include only those subcircuits that have a version number assigned with the TLMV attribute or a directory name assigned with the TLMV attribute.

   The "All Components" setting causes NX-TDL to inlclude all components.

   The default setting is "User Modules Only." Press [Return] to accept the default, or press the space bar to display the other setting and press [Return].

   After you set the last option, the system displays the following prompt.

   **Configuration File : Update**

7. To save the configuration for this session and future sessions, press [Return]. The system saves the modified configuration to the NXTDL.CFG file.

   To use the configuration for this session only, press the space bar to display the "No Update" setting and press [Return].

When configuration is complete, the system returns you to the Opening Menu.

## Running NX-TDL

**Run NX-TDL** allows you to generate a TDL file from a single P-CAD netlist file. To run NX-TDL, follow the steps below.

1. At the Opening Menu, use the space bar or the cursor keys to select **Run NX-TDL** and press [Return].

   The system displays the NX-TDL Program Screen and prompts for the input netlist filename as shown in Figure 4-3.

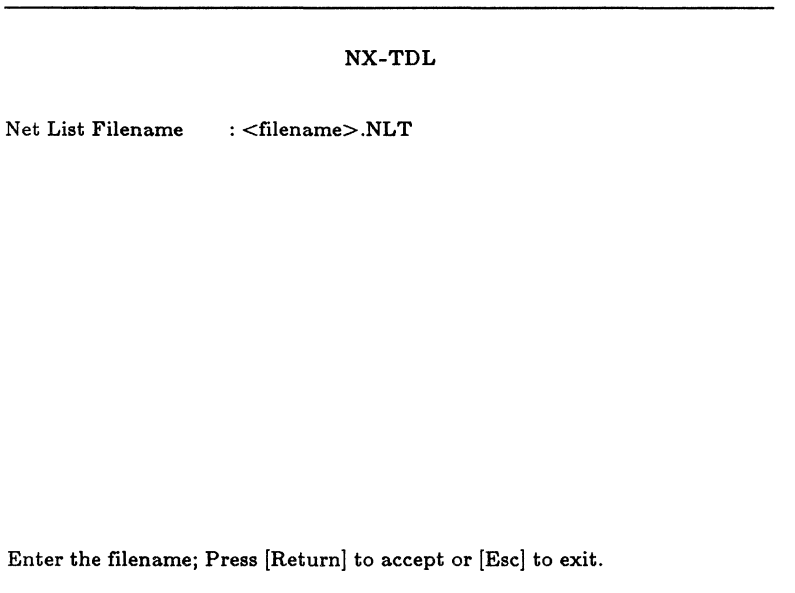



**Figure 4-3. NX-TDL Program Screen**

2. Type the name of the netlist to be read by NX-TDL and press [Return]. If you do not enter the filename extension, NX-TDL assumes it is .NLT. The system prompts for the filename of the output TDL file. The default is the input netlist filename with the .TDL extension as shown in Figure 4-4.

```
                              NX-TDL

Net List Filename      : TESTFILE.NLT
TDL Filename           : TESTFILE.TDL




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

**Figure 4-4. Sample Program Screen**

3. Press [Return] to accept the default or type another filename and press [Return].

   After you enter the last filename, NX-TDL begins file processing. It displays progress reports and error messages, if any, at the bottom of the screen.

When processing is complete, the system displays the Opening Menu.

## Running a Batch File

**Batch NX-TDL** allows you to input a batch control file to NX-TDL to generate one or more TDL files from one or more designs, each of which may be hierarchical or multisheet. To run a batch file, follow the steps below.

1. At the Opening Menu, use the space bar or the cursor keys to select **Batch NX-TDL** and press [Return].

   The system displays the NX-TDL Batch Screen and prompts for the input batch control filename as shown in Figure 4-5.

---

NX-TDL

Batch Control File : <filename>.BCF

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

---

**Figure 4-5. NX-TDL Batch Screen**

2. Type the name of the batch control file to be read by NX-TDL and press [Return]. If you do not enter the filename extension, NX-TDL assumes it is .BCF.

   After you enter the filename, NX-TDL begins file processing. It displays progress reports and error messages, if any, at the bottom of the screen.

When processing is complete, the system displays the Opening Menu.

## COMMAND LINE MODE

Command line mode allows you to specify input and output filenames or a batch control filename when you start NX-TDL.

To use command line mode, at the DOS prompt, type the command line and press [Return].

The format of the command line for a single netlist is shown below.

NXTDL <infile> <outfile>

where:

**NXTDL** initiates the program.

**infile** is the name of the input netlist file. If you do not include the filename extension, NX-TDL uses the .NLT extension.

**outfile** is the name of the output file to be created and is optional. If you specify a name with no extension, NX-TDL uses the .TDL extension. If you do not specify a name, NX-TDL uses the input netlist filename with the .TDL extension.

The format for a command line for a batch control file is shown below.

NXTDL @<filename>

where:

@ signifies that this is a batch file run. No space is allowed between the @ and the filename.

**filename** is the name of the batch control file. If you do not include the filename extension, NX-TDL uses the .BCF extension.

Three examples of command lines are shown below.

```
NXTDL TESTFILE
NXTDL TESTFILE TEST.NEW
NXTDL @TESTFILE
```

The first example causes NX-TDL to create a TDL file from the TESTFILE.NLT file. The output file will have the default filename TESTFILE.TDL.

The second example causes NX-TDL to create a TDL file from the TESTFILE.NLT file and give the output file the filename TEST.NEW.

The third example causes NX-TDL to process the TESTFILE.BCF batch file.

After you enter the command line, the program runs automatically with no further input from you. First, the NX-TDL Title Screen appears, followed by the Program Screen or the Batch Screen.

NX-TDL displays progress reports and error messages, if any, as it processes the file. When processing is complete, the system returns you to DOS.

# CHAPTER 5. VIEWING THE OUTPUT FILES

NX-TDL has two output files, the TEGAS-compatible TDL file (<filename>.TDL) and the log file (<filename>.LOG). These files are both ASCII files that you can examine as necessary.

This chapter describes how to view and print the output files, how to interpret the TEGAS-format file, and provides an example of the TDL file. The log file contains error and information messages displayed during NX-TDL operation. Appendix A describes the error messages.

## VIEWING AND PRINTING FILES

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

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

or, to print the 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 TEGAS DESIGN LANGUAGE FILE

The TDL file (<filename>.TDL) output by NX-TDL consists of three main blocks: compiler, which is always present; linker, which is present only for hierarchical designs; and end, which closes the file. Each block consists of a series of TEGAS Design Language commands that describe the design. The following sections describe the commands used in each block and tell where the information originates in the PC-CAPS schematic.

### Compiler Block

The compiler block contains a series of TDL commands describing each hierarchical file (module) in the design. The module for the lowest hierarchical level is described first; the highest level, or main design, is described last. For a multisheet schematic, all sheets are combined into one module.

The compiler block commands are described below.

**COMPILE** - Begins the compiler block.

**DIRECTORY** - Specifies the directory for the following module. NX-TDL uses the value of the TLDN attribute assigned to the module in PC-CAPS SYMB mode. If the attribute is not assigned, this command is not present.

**OPTIONS** - Specifies compile options for the following module. NX-TDL uses the value of the TLCO attribute assigned to the module in PC-CAPS SYMB mode. If the attribute is not assigned, a value of REPLACE is used.

**MODULE** - Identifies the module, in the format:

<name>//<version>/<dir>

where:

**name** is the module name, assigned to the module using the TLCN attribute in SYMB mode. If the attribute is not assigned, the filename of the module is used.

**version** is the value of the TLMV attribute assigned to the module in SYMB mode. If the attribute is not assigned, no version is given and TDL will use the default, 1.

**dir** is the TDL module directory, assigned to the module using the the TLMD attribute in SYMB mode. If the attribute is not assigned, no directory is given and TDL will use the directory specified in the DIRECTORY command.

**INPUTS** - Names the module's external input pins, specified either by using the ENTR/PIN command in SYMB mode for the entire module or by entering the PIN$IN.SYM pin symbol with the ENTR/COMP command in DETL mode.

**OUTPUTS** - Names the module's external output pins, specified either by using the ENTR/PIN command in SYMB mode for the entire module or by entering the PIN$OUT.SYM pin symbol with the ENTR/COMP command in DETL mode.

**BIDIRECT** - Names the module's I/O pins, specified either by using the ENTR/PIN command in SYMB mode for the entire module or by entering the PIN$IO.SYM pin symbol with the ENTR/COMP command in DETL mode.

If the **Bidirect** configuration option is set to "No" (the default), the command is not present.

**DESCRIPTION** - Describes the module for user reference. This section is present only if NX-TDL finds a description file for the module. (See the section "Creating the Description File" in Chapter 3 for an explanation of the description file.)

**LEVEL** - Specifies the level name for the TEGAS system, assigned using the TLLV attribute in PC-CAPS SYMB mode. If the attribute is not assigned, the LEVEL command is not generated.

**DELAYS** - Specifies delay values, assigned using the TLDY attribute in PC-CAPS SYMB mode. If the attribute is not assigned, the DELAYS command is not generated.

**USE** - Gives information about the wired gates, subcircuits, and primitives used in the module.

Wired gates are listed first. The list of wired gates consists of the contents of each TLWG attribute assigned to the design.

Primitives and subcircuits are listed next. The **USE Block** configuration option determines if all subcircuits are listed or if only those subcircuits with one or both of the applicable attributes are listed.

Primitives and subcircuits are listed in different formats. The format for an entry for a primitive is:

<name>=<tdlname> (<#in>,<#out>+<options>) /<delay>/

where:

> **name** is the P-CAD primitive name assigned using the TLCN attribute in SYMB mode. If the attribute is not assigned, the name is the filename of the primitive without the extension.

**tdlname** is the name of the corresponding TDL primitive. This name is the name in Table B-1 in Appendix B that corresponds to the component type ID assigned using the SCMD/SCAT command in SYMB mode. If the TDL name is the same as the P-CAD name, it is not repeated here.

**#in** is the number of input pins on the primitive.

**#out** is the number of output pins on the primitive.

**options** is any TEGAS primitive options assigned using the TLPO attribute in SYMB mode. If the attribute is not assigned, no options are shown.

**delay** is the component delay assigned using the TLCD attribute in SYMB mode. If the attribute is not assigned, no delay is shown.

The format for an entry for a subcircuit is:

<name>//<version>/<dir>

where:

**name** is the subcircuit name assigned using the TLCN attribute in SYMB mode. If the attribute is not assigned, the filename is used without the extension.

**version** is the value of the TLMV attribute assigned to the subcircuit in SYMB mode. If the attribute is not assigned, no version is shown.

**dir** is the TDL module directory, assigned to the subcircuit using the the TLMD attribute in SYMB mode. If the attribute is not assigned, no directory is shown.

**WIRED** - Specifies wired logic for a subcircuit and is the value of the TLWS attribute assigned in PC-CAPS SYMB mode. All components with either applicable attribute are listed under the WIRED command.

**DEFINE** - Specifies the components and their connections. Each primitive and subcircuit component of the module is listed in the format:

<instance> ( outlist ) = <name> ( inlist ) /delay/

where:

**instance** is the instance name of the component assigned using the NAME/COMP command in PC-CAPS DETL mode or assigned by PC-NODES in the format UCsssnnn.

**outlist** is a list of nets connected to the component's output pins, in the format:

<net> /<delay>/,<net> /<delay>/, ... ,<net> /<delay>/

where:

**net** is the name of the net, assigned using the PC-CAPS NAME/NET command or assigned by PC-NODES in the format UNsssnnn. The nets are listed in the order in which the pins were entered using the PC-CAPS ENTR/PIN command. If you set the **Signal=Pin** configuration option to "Yes," each net name is in the format <net> = <pinname>.

**delay** is the delay value assigned using the TLPD attribute in PC-CAPS DETL mode. The delay value follows the entry for the pin to which it was assigned.

**name** is the P-CAD component name assigned using the TLCN attribute in SYMB mode. If the attribute is not assigned, the name is the component filename without the extension.

**inlist** is a list of the names of the nets connected to all the component's input pins, assigned using the PC-CAPS NAME/NET command or assigned by PC-NODES in the format UNsssnnn. For unconnected pins, instead of the net name, the default value is NC/0/ (if the "No Connect Value" configuration setting is not 0, the current setting is used instead of 0). If you set the **Signal=Pin** configuration option to "Yes," each net name is in the format <net> = <pinname>.

**delay** is the delay value assigned to the component instance using the TLOD attribute in PC-CAPS DETL mode.

**END MODULE** - Ends the information for each module.

**END COMPILE** - Ends the compiler block.

### Linker Block

The linker block provides instructions to the TEGAS system for expanding and linking the modules in the design. It is present in the output file only for hierarchical designs processed using a batch control file. It contains a series of commands, which are described below.

**LINK** - Begins the linker block.

**DIRECTORY** - Specifies the TDL module directory of the top level of the design, assigned using the TLMD attribute in SYMB mode. If this attribute was not assigned, this command is not present.

**MODULE** - Specifies the module name, version, and directory. This information is the same as that in the compiler block in the MODULE command for the top level of the design.

**BIDIRECT** - Lists the names of the I/O pins in the top level of the design, assigned using the ENTR/PIN command in SYMB mode for the entire design. If the top level has no I/O pins, or if the **Bidirect** configuration option is set to "Yes," this command is not present.

**END LINK** - Ends the linker block.

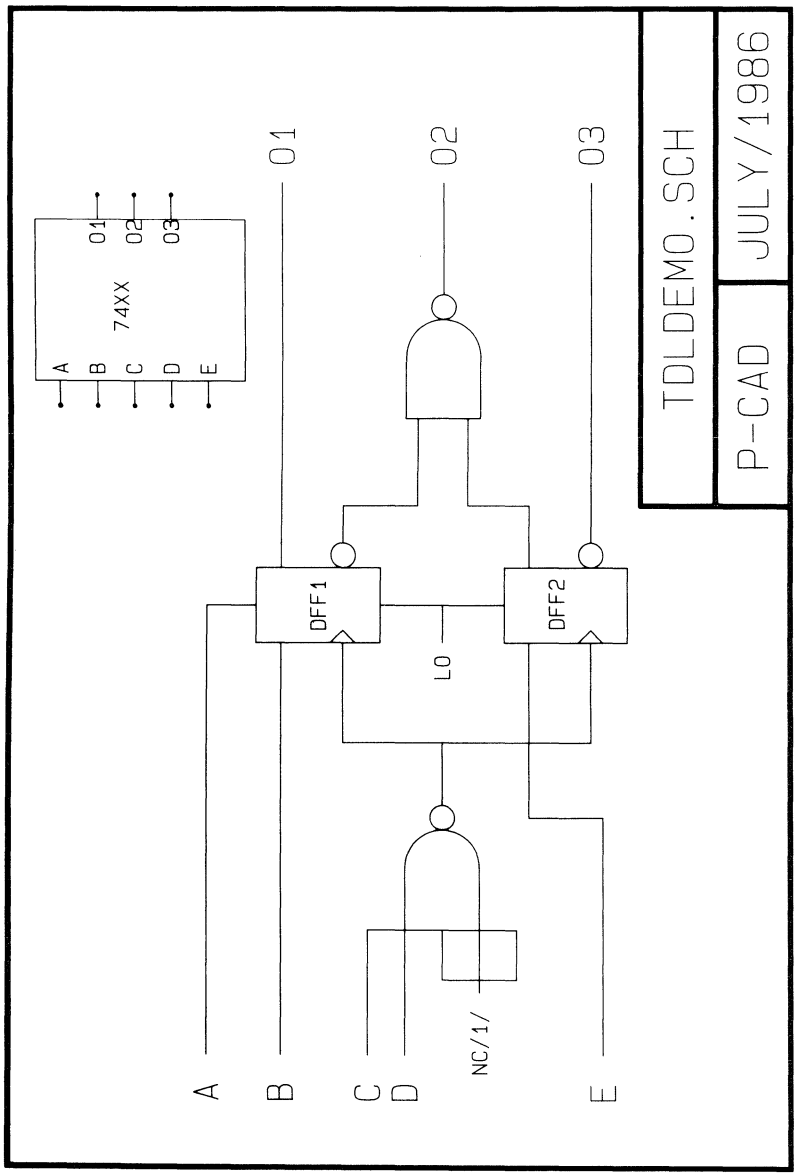
## SAMPLE TDL FILE

The example below shows the TDL file and the schematic file of the sample design included on the NX-TDL diskette. The design is single-sheet and is made up of primitives.

# TDL File: TDLDEMO.TDL

```
/*********************************************************************/
/*                                                                   */
/*                          NX-TDL NETLIST                           */
/*                                                                   */
/*  NX-TDL Version 1.31                                              */
/*  Copyright (C) 1985,1986 - Personal CAD Systems, Inc.             */
/*                                                                   */
/*  Date     : JUN 03 1986                                           */
/*  Time     : 09:09 AM                                              */
/*  File In  : TDLDEMO.NLT                                           */
/*  File Out : TDLDEMO.TDL                                           */
/*  Log File : NXTDL.LOG                                             */
/*                                                                   */
/*********************************************************************/
COMPILE ;
DIRECTORY : DIR1 ;

OPTIONS : REPLACE ;

MODULE : SN74LXXXREV2//VER12/ ;

INPUTS : A, B, C, D, E ;

OUTPUTS : O1, O2, O3 ;

LEVEL : IC;

USE :

DFF1 = DENE (4,2  + COMPPC COMPPS ) /1,2,3/,
DFF2 = DENE (3,2  + COMPPS ) /1,2,3/,
NAND2 = NAND (2,1 ),
NAND5 = NAND (5,1 );

DEFINE :

UC000000 ( O1 /1,2/,UN000000 /4/) =  DFF1 ( UN000002 ,B ,A ,NC/0/ )/5/;
UC000001 ( UN000001 ,O3 ) =  DFF2 ( UN000002 ,E ,NC/0/ );
UC000002 ( O2 ) =  NAND2 ( UN000000 ,UN000001 );
UC000003 ( UN000002 ) =  NAND5 ( C ,D ,NC/1/ ,NC/1/ ,NC/1/ );

END MODULE ;

END COMPILE ;

END TDL ;
```

## Schematic File: TDLDEMO.SCH




000-0061-03

# APPENDIX A. ERROR MESSAGES

NX-TDL displays error messages on the screen and stores them in the log file, which you can view or print.

This appendix lists and describes NX-TDL error messages. Errors are grouped into three types: file access errors, netlist processing errors and other errors. Within each group, error messages are listed in alphabetical order.

## FILE ACCESS ERRORS

Errors in this group occur before NX-TDL can start file processing. You cannot run the program until the error is corrected.

**Message: Batch Control File Does not Exist**

Cause: NX-TDL could not find the batch control file you specified.

Action: Be sure you enter the correct filename, and specify a directory path if necessary.

**Message: Database Error**
**Database File <filename> not found**

Cause: NX-TDL could not find the named input netlist file.

Action: Be sure you enter the correct filename, and specify a directory path if necessary.

**Message:** **Database Error**
**Incompatible Database**

Cause: The named input file was not a netlist produced by PC-NODES.

Action: Be sure you enter the correct filename. If necessary, use PC-NODES to generate a new netlist file from the schematic.

**Message:** **Disk is full--Program terminated. Press any key to exit.**

Cause: Disk was full.

Action: Get more disk space. After you press a key, the program exits to DOS, and the output file is deleted.

**Message:** ****** Error: <filename> used more than once.**

Cause: The input and output filenames were the same.

Action: Change either the input or output filename.

**Message:** **Failed to close <xxx> file**

Cause: This message occurs if a corrupted file pointer exists.

Action: Rerun NX-TDL.

**Message: Failed to flush <xxx> cache**

**Failed to open workfiles**

Cause: These messages occur if the number of files to be simultaneously opened exceed the number specified in the CONFIG.SYS file.

Action: Verify that file=15 in CONFIG.SYS.

**Message: Failed to open log file**

Cause: In the current directory there is either a subdirectory or a read only file named NXTDL.LOG, or the disk is full.

Action: Delete the file or directory NXTDL.LOG if it exists. If the disk is full, delete other files.

**Message: Failed to open <xxx> file**

Cause: This message occurs if the number of files to be simultaneously opened exceed the number specified in the CONFIG.SYS file.

Action: Verify that file=15 in CONFIG.SYS.

**Message: **** <filename> Using illegal File Extension.**

Cause: The output filename specified as a reserved name had the extension .WFK.

Action: Use a different filename extension.

**Message:** **Incorrect NXTDL.CFG file format**
**Press Any Key to exit.**

Cause: The NXTDL.CFG file in the current directory was not usable. It may have been produced using an old version of NX-TDL.

Action: Delete the file. If you want to use options other than the defaults, create a new fle in NX-TDL configuration mode.

**Message:** ****** No batch filename**

Cause: A batch filename was not entered after "NXTDL @" on the command line.

Action: Enter a valid command line.

## NETLIST PROCESSING ERRORS

The following errors are errors in the construction of the input database.

**Message:** **Database Error**
<xxx>

Cause: The input netlist file had an internal error. The file may be damaged.

Action: Use PC-NODES to extract a new netlist file from the schematic, and then rerun NX-TDL. If the error persists, check to make sure the schematic is complete and correct.

**Message: Invalid Name Detected <nametype> : <name>**

Cause: The specified name was longer than twelve characters or contained nonalphanumeric characters.

Action: Use PC-CAPS to correct the name, and then rerun PC-NODES and NX-TDL.

**Message: Invalid TLPD Attribute <compname>**

Cause: The named component instance was assigned an invalid delay value for the TLPD attribute.

Action: Use PC-CAPS to correct the attribute value so that it is in the format specified in Chapter 2, "Preparing the Schematic Database."

**Message: Invalid TLPD Attribute Substring <substring>**

Cause: The specified delay value in the TLPD attribute was not in the correct format for a delay for one pin.

Action: Use PC-CAPS to correct the attribute so that delay values for each pin have both a starting and ending slash (/<delay>/). Then rerun PC-NODES and NX-TDL.

**Message: Missing Keyword in Batch Control File OUTFILE**

Cause: The batch control file did not start with the OUTFILE keyword.

Action: Correct the batch control file.

**Message: Sheet Identifier Error**

**Sheet Identifier Error**
**No Sheet ID**

Cause: These messages appear when NX-TDL cannot find or access the value of the SHEET attribute assigned to a sheet of a multisheet design.

Action: Be sure the schematic is multisheet. If not, correct the batch control file and rerun NX-TDL. If so, use PC-CAPS to assign or correct the SHEET attribute, use PC-NODES to extract a new netlist, and then rerun NX-TDL.

**Message: Unconnected Input Pin**
**Component : <compname> Pin : <pinname>**

Cause: The specified input pin of the named component was not connected to a net.

Action: Use PC-CAPS to connect the pin to a net, and then rerun PC-NODES and NX-TDL.

**Message: Unconnected Module Terminal Pin**
**Component : <compname> Pin : <pinname>**

Cause: The specified terminal pin was not connected to a net on the circuit.

Action: Use PC-CAPS to connect the pin to a net, then rerun PC-NODES and NX-TDL.

**Message: WARNING (USE) : Invalid Primitive Type Detected**
**Type = <type>**

Cause: The specified primitive type was not one of the types recognized by NX-TDL.

Action: Use only the component type IDs that are listed in Table B-1 in Appendix B.

## OTHER ERRORS

The remaining errors are described below.

**Message: **** Invalid bar character <char>.**

Cause: An invalid bar character was entered.

Action: Enter a legal bar character.

# APPENDIX B. NX-TDL SCHEMATIC SYMBOLS

This appendix describes and shows the symbols that are supplied with NX-TDL.

## COMPONENT TYPES

Table B-1 shows the component type IDs recognized by NX-TDL and the corresponding TEGAS primitive names. Each schematic symbol supplied with NX-TDL is already assigned the appropriate ID. Each schematic symbol you create for use with NX-TDL must be assigned one of the IDs in Table B-1.

## Table B-1. NX-TDL Component Type IDs

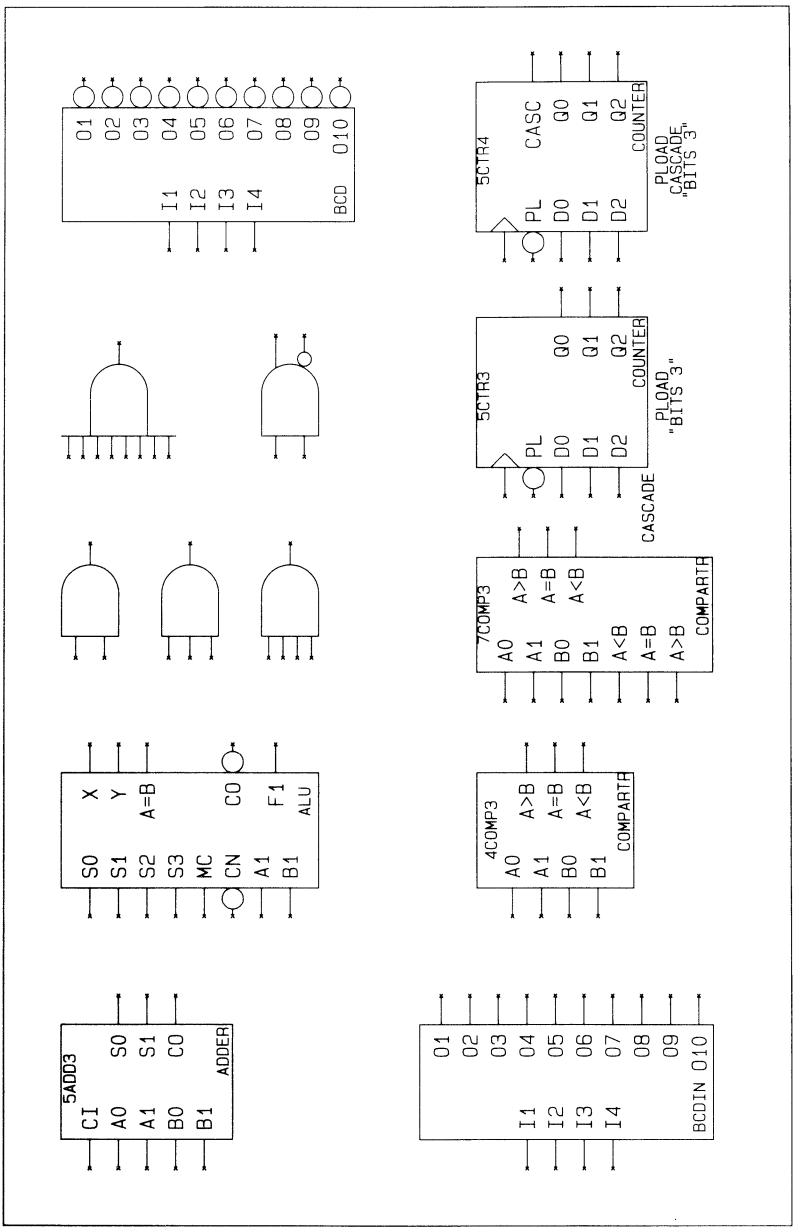
| Type ID | Primitive Name | Type ID | Primitive Name | Type ID | Primitive Name |
|---|---|---|---|---|---|
| 1 | NOT | 42 | SERSHIFT | 65 | SHD10 |
| 2 | NAND | 43 | SHIFT | 66 | SHDALL |
| 3 | NOR | 44 | REGISTER | 67 | BCDIN |
| 4 | AND | 45 | REGFILE | 68 | BCD |
| 5 | OR | 46 | ENCODER | 69 | PUSHPULL |
| 6 | XOR | 47 | COMPARTR | 70 | ORNOR |
| 16 | TRANEPE | 48 | PARITY | 71 | ANDNAND |
| 17 | TRANENE | 49 | MULTIPLY | 72 | DECODER |
| 18 | TRIBUS | 50 | SUBTRACT | 73 | DECODERE |
| 20 | ROM | 51 | COUNTER | 74 | NANDMOUT |
| 21 | RAM | 52 | PLA | 75 | ADDER |
| 30 | PI | 53 | TRANSP | 76 | WAND |
| 31 | PPO | 54 | DELAYT | 77 | WOR |
| 32 | DELAY | 55 | BDSWITCH | 78 | TRIAND |
| 33 | FDETECT | 56 | ALU | 79 | TRIOR |
| 34 | REONESHT | 57 | PWR | 80 | TRINAND |
| 35 | SRNANDL | 58 | GRND | 81 | MUX |
| 36 | SRNORL | 59 | FLTDUMMY | 82 | MUXE |
| 37 | SRENE | 60 | JKMNE | 83 | NXOR |
| 38 | ONESHOT | 61 | SRMNE | 84 | SWITCH |
| 39 | DENE | 62 | CLOCK | 85 | TEST |
| 40 | DMNE | 63 | SREPE | 105 | DEPE |
| 41 | TMNE | 64 | SHD01 | 107 | JKEPE |
| | | | | 109 | JKENE |

## SYMBOLS

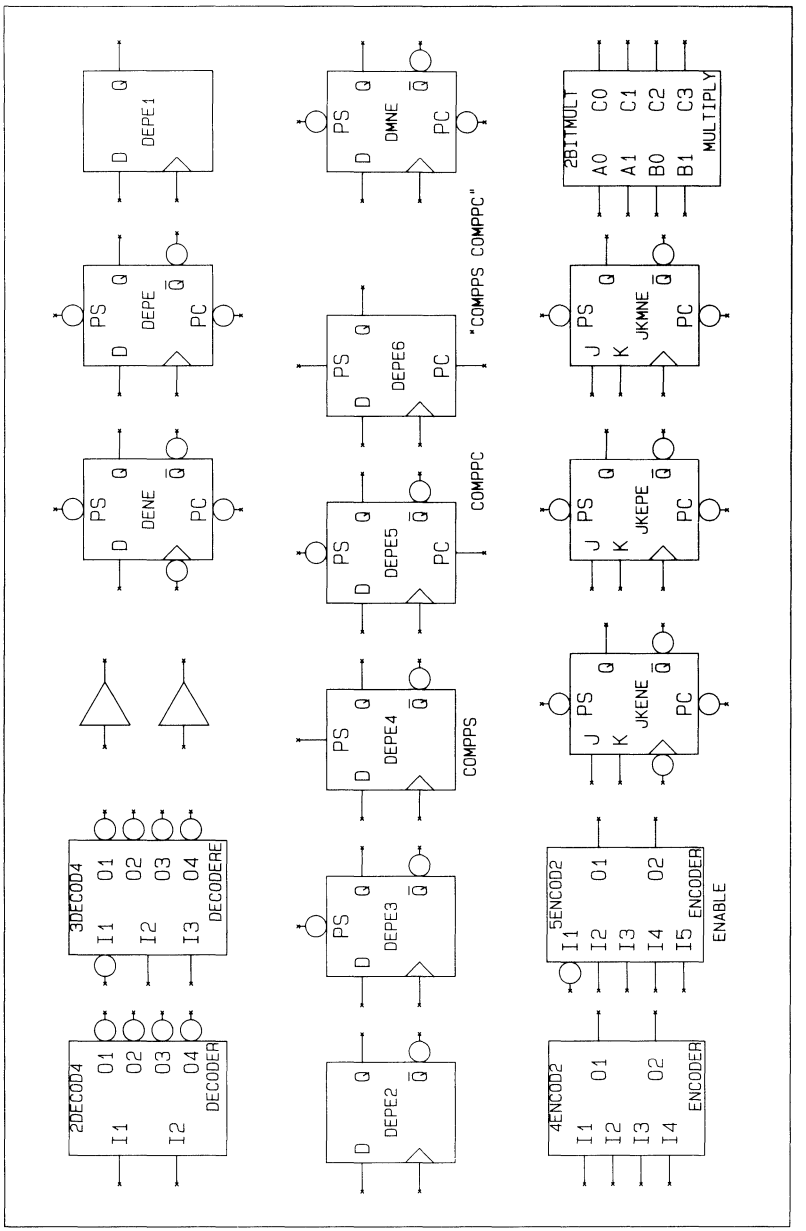
The following pages show the schematic symbols supplied with NX-TDL. Opposite each page of symbols is a list of the symbols and their component type IDs.

These symbols were created using the primitive definitions in the TEGAS manual. You can use them directly in PC-CAPS schematics for use with NX-TDL. You can also use them as models for creating additional TEGAS-compatible symbols; that is, you can copy a symbol file, and then use PC-CAPS in SYMB mode to add, delete, or change pins or graphics.

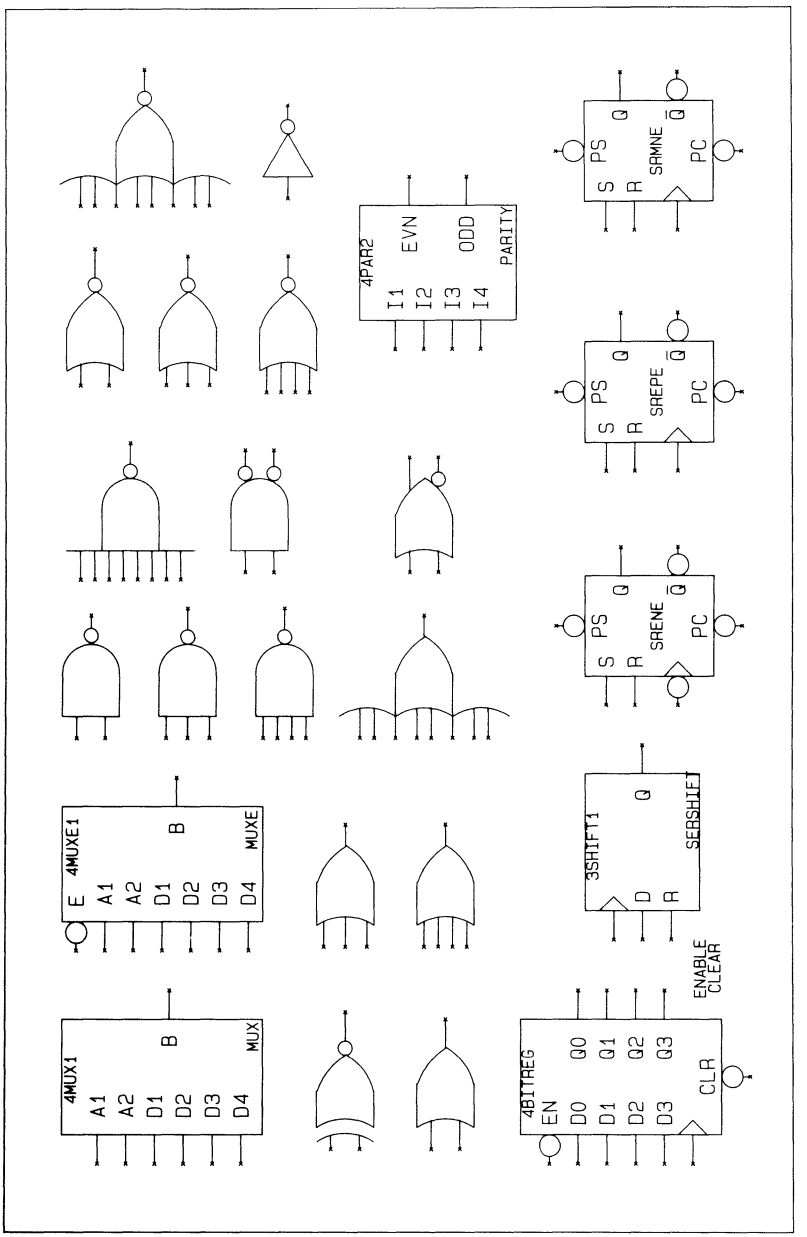
| Symbol Filename | Type ID |
|---|---|
| 5ADD3.SYM | 75 |
| ALU.SYM | 56 |
| AND.SYM | 4 |
| AND3.SYM | 4 |
| AND4.SYM | 4 |
| AND8.SYM | 4 |
| ANDNAND.SYM | 71 |
| BCD.SYM | 68 |
| BCDIN.SYM | 67 |
| 4COMP3.SYM | 47 |
| 7COMP3.SYM | 47 |
| 5CTR3.SYM | 51 |
| 5CTR4.SYM | 51 |


5ADD3
CI
A0
A1
B0
B1
S0
S1
C0
ADDER
S0
S1
S2
S3
MC
CN
A1
B1
X
Y
A=B
C0
F1
ALU
4COMP3
A0
A1
B0
B1
A>B
A=B
A<B
COMPARTR
7COMP3
A0
A1
B0
B1
A<B
A=B
A>B
A>B
A=B
A<B
COMPARTR
5CTR3
PL
D0
D1
D2
Q0
Q1
Q2
COUNTER
CASCADE
PLOAD
"BITS 3"
5CTR4
PL
D0
D1
D2
CASC
Q0
Q1
Q2
COUNTER
PLOAD
CASCADE
"BITS 3"
I1
I2
I3
I4
O1
O2
O3
O4
O5
O6
O7
O8
O9
O10
BCD
BCDIN

| Symbol Filename | Type ID |
|---|---|
| 2DECOD4.SYM | 72 |
| 3DECOD4.SYM | 72 |
| DELAY.SYM | 32 |
| DELAYT.SYM | 54 |
| DENE.SYM | 39 |
| DEPE.SYM | 105 |
| DEPE1.SYM | 105 |
| DEPE2.SYM | 105 |
| DEPE3.SYM | 105 |
| DEPE4.SYM | 105 |
| DEPE5.SYM | 105 |
| DEPE6.SYM | 105 |
| DMNE.SYM | 40 |
| 4ENCOD2.SYM | 46 |
| 5ENCOD2.SYM | 46 |
| JKENE.SYM | 109 |
| JKEPE.SYM | 107 |
| JKMNE.SYM | 60 |
| 2BITMULT.SYM | 49 |


2DECOD4
I1
I2
O1
O2
O3
O4
DECODER
3DECOD4
I1
I2
I3
O1
O2
O3
O4
DECODERE
DEPE2
DEPE3
DEPE4
DEPE5
DEPE6
DEPE1
DENE
DEPE
DMNE
JKENE
JKEPE
JKMNE
COMPPS
COMPPC
"COMPPS COMPPC"
4ENCOD2
5ENCOD2
ENCODER
ENABLE
2BITMULT
A0
A1
B0
B1
C0
C1
C2
C3
MULTIPLY

| Symbol Filename | Type ID |
|---|---|
| 4MUX1.SYM | 81 |
| 4MUXE1.SYM | 82 |
| NAND.SYM | 2 |
| NAND3.SYM | 2 |
| NAND4.SYM | 2 |
| NAND8.SYM | 2 |
| 2NDMOUT2.SYM | 74 |
| NOR.SYM | 3 |
| NOR3.SYM | 3 |
| NOR4.SYM | 3 |
| NOR8.SYM | 3 |
| NOT.SYM | 1 |
| NXOR.SYM | 83 |
| OR.SYM | 5 |
| OR3.SYM | 5 |
| OR4.SYM | 5 |
| OR8.SYM | 5 |
| ORNOR.SYM | 70 |
| 4PAR2.SYM | 48 |
| 4BITREG.SYM | 44 |
| 3SHIFT1.SYM | 43 |
| SRENE.SYM | 37 |
| SREPE.SYM | 63 |
| SRMNE.SYM | 61 |

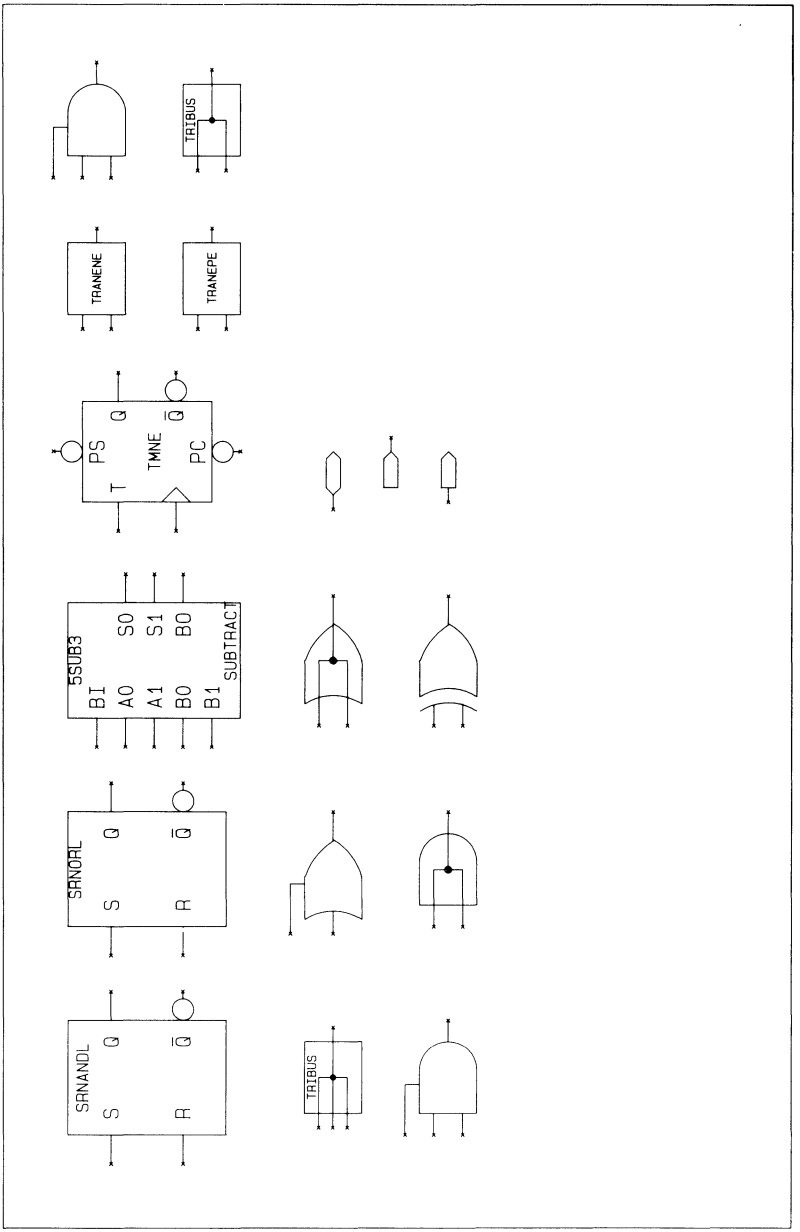

4MUX1
A1
A2
D1
D2
D3
D4
B
MUX
E 4MUXE1
A1
A2
D1
D2
D3
D4
B
MUXE
4BITREG
EN
D0
D1
D2
D3
Q0
Q1
Q2
Q3
CLR
ENABLE
CLEAR
3SHIFT1
D
R
Q
SERSHIFT
4PAR2
I1
I2
I3
I4
EVN
ODD
PARITY
S
R
PS
SAENE
Q
PC
S
R
PS
SAEPE
Q
PC
S
R
PS
SAMNE
Q
PC

000-0061-03

| Symbol Filename | Type ID |
|---|---|
| SRNANDL.SYM | 35 |
| SRNORL.SYM | 36 |
| 5SUB3.SYM | 50 |
| TMNE.SYM | 41 |
| TRANENE.SYM | 17 |
| TRANEPE.SYM | 16 |
| TRIAND3.SYM | 78 |
| TRIBUS.SYM | 18 |
| TRIBUS3.SYM | 18 |
| TRINAND3.SYM | 80 |
| TRIOR.SYM | 79 |
| WAND.SYM | 76 |
| WOR.SYM | 77 |
| XOR.SYM | 6 |

**Pin Component Symbols**

PIN$IO.SYM
PIN$IN.SYM
PIN$OUT.SYM


TRIBUS
TRANENE
TRANEPE
PS
Q
T
TMNE
Q̄
PC
5SUB3
BI
A0
A1
B0
B1
S0
S1
B0
SUBTRACT
SRNORL
S
Q
R
Q̄
SRNANDL
S
Q
R
Q̄
TRIBUS