PDIF™

User's Manual

October 1986

PERSONAL CAD SYSTEMS INC.

000-0114-01

COPYRIGHT

TRADEMARKS

000-0114-01

ABOUT THIS MANUAL

This manual describes the P-CAD Database Interchange Format (PDIF), which is an ASCII format for P-CAD design databases, and the PDIF-IN and PDIF-OUT programs that provide the ability to transfer databases into and out of the P-CAD system.

Chapter 1, INTRODUCTION, provides an overview of PDIF, including explanations of several applications, and gives installation instructions.

Chapter 2, USING PDIF-IN, gives instructions for running PDIF-IN.

Chapter 3, USING PDIF-OUT, gives instructions for running PDIF-OUT.

Chapter 4, THE PDIF FILE, gives detailed specifications for the PDIF ASCII file and for each section and subsection of the file.

Chapter 5, PDIF KEYWORD REFERENCE, gives detailed specifications for using all the keywords introduced in Chapter 4.

Appendix A, PDIF-IN ERROR MESSAGES, describes error messages produced by PDIF-IN.

Appendix B, PDIF-OUT ERROR MESSAGES, describes error messages produced by PDIF-OUT.

Appendix C, NETLIST IMPORT, explains how to use PDIF-IN for importing netlists from other CAD systems.

Appendix D, P-CAD TO PDIF CROSS-REFERENCE, contains a cross-reference table of P-CAD commands and PDIF data.

Appendix E, SAMPLE PDIF FILES, shows sample files produced by PDIF-OUT from various types of design databases.

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.

<xxxxx> 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 .SYM extension 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)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

FIGURES

FIGURES (Continued)

TABLES

CHAPTER 1. INTRODUCTION

The P-CAD Database Interchange Format (PDIF) is an ASCII format for P-CAD design databases. It is designed to work with both schematic databases and printed circuit board (PCB) databases.

The PDIF software includes the PDIF-IN and PDIF-OUT programs. PDIF-IN enables you to input a design in the PDIF ASCII format to the P-CAD system and PDIF-OUT enables you to translate a P-CAD design into the PDIF format.

This chapter gives an overview of the PDIF programs and their applications and gives installation instructions.

OVERVIEW

This section describes the input and output files for the PDIF programs and lists several applications for the programs.

PDIF Input and Output Files

Both PDIF programs can work with any P-CAD schematic or PCB design file. With PDIF-OUT, you input a file produced by PC-CAPS or PC-CARDS and produce the PDIF file. With PDIF-IN, you input a PDIF file and produce a file in PC-CAPS or PC-CARDS format. All these files are described in this section.

The P-CAD Files

The P-CAD files that can be input to PDIF-OUT and output by PDIF-IN are listed below.

The **symbol file** (<filename>.SYM) produced by PC-CAPS contains the representation of a schematic component.

The **schematic database file** (<filename>.SCH) produced by PC-CAPS contains all the information for a schematic circuit. This circuit can be an entire design, an element of a hierarchical design, or a sheet of a multisheet design. Hierarchical and multisheet schematic structures are discussed in Chapter 4, "The PDIF File."

The **part file** (<filename>.PRT) produced by PC-CARDS contains the representation of a physical component to be used in PCB design.

The **PCB database file** (<filename>.PCB) produced by PC-CARDS contains all the applicable information for a PCB design.

> NOTE: PCB database files produced by P-CAD's PC-PACK, PC-PLACE, and PC-ROUTE programs also can be input into PDIF-OUT.

The **padstack file** (<filename>.PS) produced by PC-CARDS contains the representation of a padstack and is used to specify padstack information for plotting a PCB design.

The PDIF File

The PDIF file (<filename>.PDF) is output by PDIF-OUT and input to PDIF-IN. It has sections for all the graphic, connectivity, and text information that is in a P-CAD file.

The PDIF file is an ASCII file that you can examine to check the database information and that you can create or edit using a text editor. The format of the PDIF file is described in detail in Chapter 4, "The PDIF File."

PDIF Data Flow

The PDIF file enables you to move data between the P-CAD files discussed above and other systems and other applications. The flow charts in this section show how PDIF can be used. The applications are discussed in the next section, "PDIF Applications."

Figure 1-1 shows data flow into and out of the PDIF file.

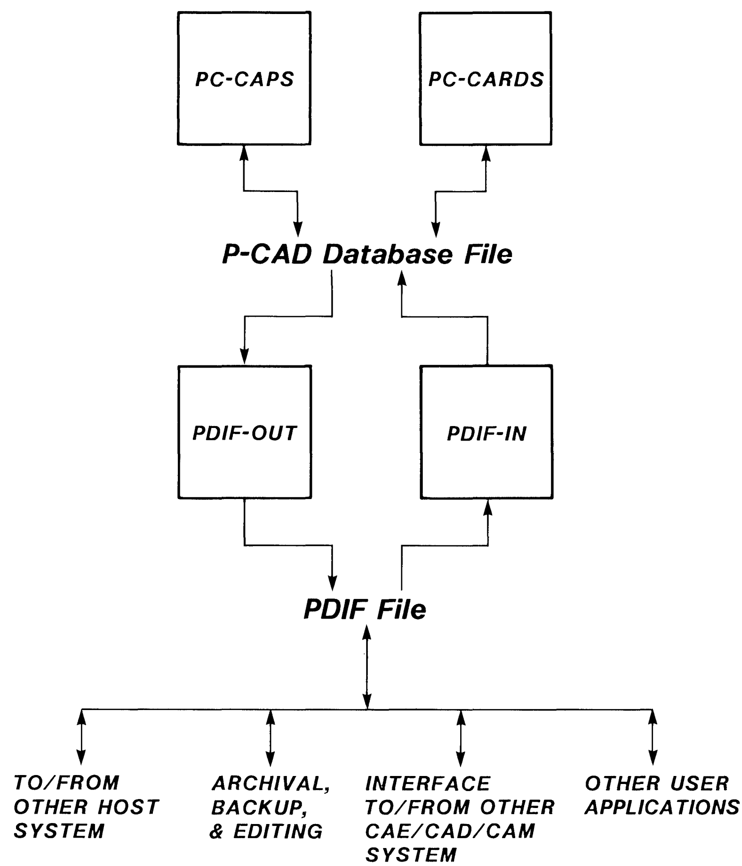


Figure 1-1. PDIF Data Flow

PDIF-IN allows you to translate a PDIF file into P-CAD database format. The PDIF file can be created by using PDIF-OUT, by using another program, or by using a text editor. The output P-CAD database file can be used with PC-CAPS (schematic databases) or PC-CARDS (PCB databases).

Figure 1-2 shows PDIF-IN data flow.

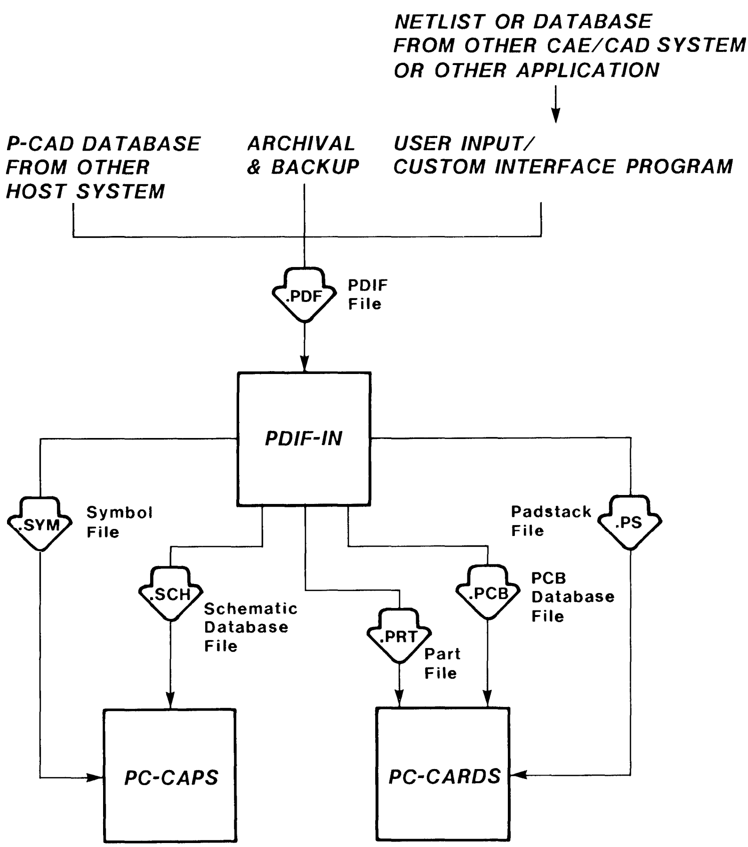


Figure 1-2. PDIF-IN Data Flow

PDIF-OUT translates a PC-CAPS schematic database or a PC-CARDS PCB database into a PDIF file, which then can be used for the appropriate application. Figure 1-3 shows PDIF-OUT data flow.

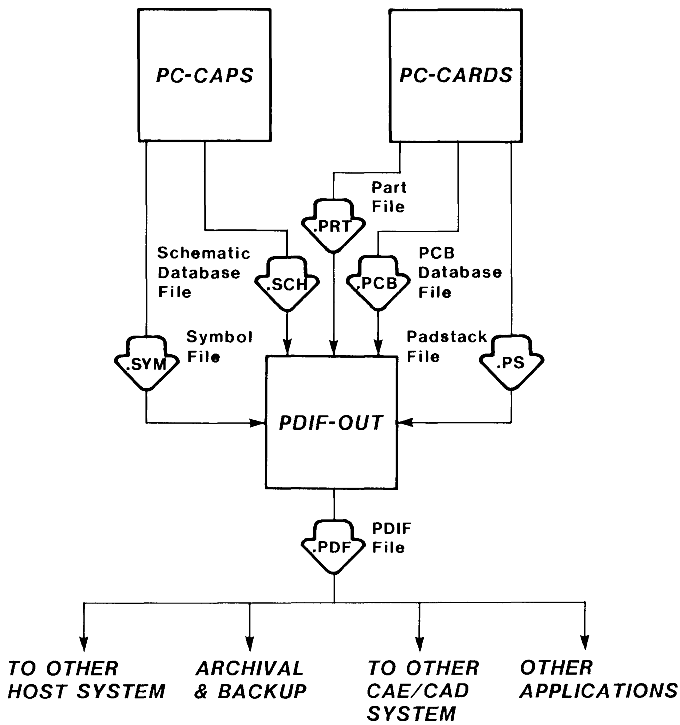


Figure 1-3. PDIF-OUT Data Flow

PDIF Applications

PDIF provides a standard format that can be used for working with P-CAD databases and for interfacing the P-CAD system with other CAE/CAD/CAM systems. Some examples of PDIF applications are given below.

Schematic Database Interface - PDIF allows you to transfer a design database between the P-CAD system and another CAE/CAD system. For example, you might want to transfer a database created in another system into PC-CAPS for editing. For this application, you can develop a translator to convert databases between your other system and the PDIF format.

Schematic Netlist Interface - PDIF can be used to develop a custom netlist interface between P-CAD schematics and another system's simulator or PCB CAD system. You can develop programs to extract the necessary netlist information from the PDIF file and to input the resulting netlist to the other system.

PCB Database Interface - PDIF allows you to use the P-CAD system to produce a PCB design and do partial layout editing, then use another PCB CAD system for autoplacement and autorouting. You can develop programs to translate the P-CAD PCB database into the format required by the autoplace or autoroute program and, if needed, to translate the placed or routed database back into PDIF format for final P-CAD manual editing.

In the same manner, you can use PDIF to develop a custom interface with a CAM system; for example, an interface to drilling machines, component auto-insertion systems, or automatic test equipment.

Netlist Import - You can use the PDIF format to import design netlists from another system into P-CAD format. This application is most common for PCB netlist import but can also be used for schematics. For this application, you can either develop a custom translator or use a text editing program to create the PDIF file. The PDIF file information required for this application is described in Appendix C, "Netlist Import."

Library Import - You can develop a custom interface with another program to import all your library parts to P-CAD. This application enables you to transfer all the parts without errors and without having to rebuild them.

Transferring Data Between Host Systems - The PDIF file provides a means of transferring P-CAD data files between different hardware systems and different versions of the P-CAD software.

Archival, Backup, and Editing - You may want to store your P-CAD databases in PDIF format. PDIF also provides a quick way to edit a design database. For example, the PDIF format enables you to use a text editor to do a global edit of trace widths in a PCB database. You can also use PDIF to view pin names and pin orders for P-CAD symbols and parts and to make any necessary corrections.

SYSTEM REQUIREMENTS

Before you install PDIF, your computer system must have the following minimum configuration:

- o IBM PC/AT, PC/XT, HP Vector, TI Business Pro, TI Professional, or equivalent
- o 640K of RAM (512 for the TI Professional)
- o Color or monochrome monitor
- o DOS 2.0 or higher operating system

INSTALLATION

The PDIF program is supplied on two diskettes. One diskette contains the PDIF-IN program file, PDIFIN.EXE. The other diskette contains the PDIF-OUT program file, PDIFOUT.EXE. In addition, both diskettes contain the following sample data files:

74LS00.SYM	74LS00.PDF
74LS04.SYM	74LS04.PDF
EX1.SCH	EX1.PDF
EX2.SYM	EX2.PDF
EX3.SCH	EX3.PDF
C60R32.PS	C60R32.PDF
7400A.PRT	7400A.PDF
EX4.PCB	EX4.PDF
7404T.PRT	V50R28.PS
N60S32.PS	C60S32.PS
N60R32.PS	EX4.SSF

The sample data files are not required for program operation. You can use them to see how the program works. Refer to Appendix E, "Sample PDIF Files," for more information.

To install PDIF on your hard disk, use the following procedures. These procedures assume that you are using the P-CAD recommended directory structure. (Refer to your *P-CAD Installation Guide* for information on P-CAD directory structure.)

First, insert the PDIF-IN diskette in drive A. Then change to the \PCAD\EXE directory by typing:

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

Copy the PDIF-IN program file by typing:

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

Remove the diskette and insert the PDIF-OUT diskette. Copy the PDIF-OUT program file by typing:

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

Next, 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:*.PDF [Return]
COPY A:*.SYM [Return]
COPY A:*.SCH [Return]
COPY A:*.PS [Return]
COPY A:*.PRT [Return]
COPY A:*.PCB [Return]
COPY A:*.SSF [Return]
```

CHAPTER 2. USING PDIF-IN

This chapter describes the required conditions and procedures for configuring and running PDIF-IN.

Before running PDIF-IN, be sure that:

- Your system is correctly configured.
- You have installed the PDIF-IN program file (PDIFIN.EXE).
- You have created the PDIF file.
- Any layer structure, part, symbol, or padstack file to be accessed by PDIF is present in the current directory or in a directory specified using the **Path** configuration option (described in "Configuring PDIF-IN" in this chapter).

PDIF-IN has two operating modes, interactive and command line. In interactive mode, PDIF-IN displays a series of screens and you select options and specify filenames. In command line mode, you specify filenames when you start the program and PDIF-IN processes the input file automatically.

Before running PDIF-IN, you must decide on your mode of importing components.

The following sections explain component import, describe how to use interactive mode to start, configure, and run PDIF-IN, and describe how to use command line mode.

PREPARING FOR COMPONENT IMPORT

Corresponding to each set of component instances of the same type is a component definition that carries information that is common to all instances. For each schematic symbol or PCB part, you can use either the definition in the COMP_DEF section of the input PDIF file or the definition in the corresponding P-CAD symbol or part file.

Component Import Modes

Importing component definitions from the PDIF file is called internal mode. Importing component definitions from the existing P-CAD files is called external mode. A PDIF-IN configuration option allows you to select the component import mode for all components (refer to "Configuring PDIF-IN").

The Exception File

You can instruct PDIF-IN to use internal mode for some components and external mode for others. To do this, you select one mode in configuration and create a file that lists the names of the components that are to be imported in the alternate mode. This file is called the exception file.

Use a text editor to create the exception file. List the filenames for all components to be imported in the alternate mode, one filename per line. Include the filename extensions. You can give the exception file any filename and extension.

Figure 2-1 shows a sample exception file.

```
74LS00.SYM
74LS04.SYM
```

Figure 2-1. Sample Exception File

For example, if the configuration option for component import mode is set to **Internal** and the exception file in Figure 2-1 is used, PDIF-IN will use the component definitions in the input PDIF file for all components except the 74LS00 and the 74LS04. For these two components, PDIF-IN will use the definition in the corresponding P-CAD symbol files.

A PDIF-IN configuration option allows you to specify the name of the exception file for PDIF-IN to use (refer to "Configuring PDIF-IN").

Importing Component Attributes

In the P-CAD system, component definitions can include user-assigned attributes. These attributes are included in P-CAD symbol and part files but not in the PDIF file generated by PDIF-OUT. However, they can be included in the component definition by editing the PDIF file. PDIF-IN lets you choose whether or not to include a component definition's attributes with all instances of the components in the output P-CAD schematic or PCB design file. (All attributes included in component instances are always included in the design file.)

In most cases, any desired attributes for a component instance are included with the instance in the PDIF file, and PDIF-IN will automatically include the attributes in the output P-CAD file. For this reason, it is generally not advisable to import component attributes unless you want to add an attribute to all instances of a component in the design.

A PDIF-IN configuration option allows you to choose whether or not to import component definition attributes with each component instance (refer to "Configuring PDIF-IN").

Creating P-CAD Components

You can use PDIF-IN to create a P-CAD symbol or part file for each internally imported component of a schematic or PCB design. It is useful to create P-CAD files for components that are used in other designs that you plan to input to PDIF-IN. When you use PDIF-IN for other designs, you can then use external import, which takes less time, for those components.

A PDIF-IN configuration option allows you to choose whether or not to create and save components that are imported internally (refer to "Configuring PDIF-IN"). If you use internal component import, PDIF-IN will create P-CAD files for all components except any listed in the exception file. If you use external import and used an exception file, PDIF-IN will create P-CAD files for all components listed in the exception file.

When you create components, PDIF-IN creates a subdirectory of the current directory and names it INTCMPDR. After the components are created, you should move them to the directory where you store your symbols or parts.

> NOTE: PDIF-IN uses the INTCMPDR subdirectory for internal purposes. Any files on this subdirectory could be erased the next time PDIF-IN is run.

INTERACTIVE MODE

In interactive mode, you must first start PDIF-IN to display the Opening Menu.

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

    PDIFIN [Return]

When the PDIF-IN Title Screen appears, press any key to continue. The system then displays the PDIF-IN Opening Menu as shown in Figure 2-2.

PDIF-IN

Options:

Configure PDIF-IN

>> Run PDIF-IN <<

Exit PDIF-IN

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

Figure 2-2. PDIF-IN Opening Menu

This menu provides the following options.

Configure PDIF-IN - Allows you to examine or change PDIF-IN configuration options.

Run PDIF-IN - Allows you to begin file processing.

Exit PDIF-IN - Returns you to DOS.

If you want to change the configuration, use the procedure in the next section, "Configuring PDIF-IN." If you do not want to change the configuration, continue to "Running PDIF-IN."

Configuring PDIF-IN

Configure PDIF-IN allows you to set several PDIF-IN system options. These settings are stored in the PDIFIN.CFG file, which is used each time you run the program. Default settings are provided and are used if you do not create the configuration file. You can use this menu selection to change the settings at any time.

To configure PDIF-IN, follow the steps below.

1. At the PDIF-IN Opening Menu, press the space bar or use the cursor keys to select **Configure PDIF-IN**, then press [Return]. The system displays the Configuration Screen with the current settings as shown in Figure 2-3.

PDIF-IN Configuration

Directory Path :
Current

System Memory Size 640K
Import Component Attribute .... No
Component Import Mode .... External
Exception File .... <filename>

Enter the path; Press: [RETURN] to accept

Figure 2-3. PDIF-IN Configuration Screen

2. The first configuration option allows you to specify a directory path. This path includes all directories you want the program to search for: symbol files, part files, padstack files, special symbol files, the layer structure database file, and input PDIF files. You must specify the complete path to each directory and separate directory names with a semicolon. For example:

   \PCAD\SYM\TTL;\PCAD\SYM\DISCRETE;
   \PCAD\DESIGNS

The default setting is "Current," which tells the program to look in the current directory only.

Press [Return] to accept the default or type the path, and then press [Return].

3. The next option is **System Memory Size.** The choices are "640K" and "512K." The default is "640K." The "512K" option is only used with the Texas Instruments (TI) Professional 512K system.

   Press [Return] to accept the default. To select "512K," press the space bar to display the "512K" setting, then press [Return].

4. The next option allows you to specify whether or not to import attributes assigned to a component definition. This option is described previously in "Preparing for Component Import." The default is "No."

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

5. The next option is the **Component Import Mode.** This option is described previously in "Preparing for Component Import." The default is "External."

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

6. The next option allows you to specify the name of the exception file to be used for symbol importation. The exception file is explained in the previous section, "Preparing for Component Import."

If you are using an exception file, enter the filename and press [Return]. If you are not using an exception file, just press [Return].

If you set the **Component Import Mode** to "Internal," or if you specified an exception file, the following option appears:

Create Components from Internal COMP_DEF Section No

7. The option shown above allows you to specify whether or not to create P-CAD symbol or part files for components that are internally imported. This option is described previously in "Preparing for Component Import." The default is "No."

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

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

   Save Configuration : YES

8. To save the configuration for this session and future sessions, press [Return] to accept "YES." The system saves the modified configuration as a file named PDIFIN.CFG.

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

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

Running PDIF-IN

Run PDIF-IN allows you to generate a P-CAD file from a PDIF file. To run PDIF-IN, follow the steps below.

1. At the Opening Menu, select **Run PDIF-IN** and press [Return].

   The system displays the PDIF-IN Program Screen and prompts for the PDIF filename as shown in Figure 2-4.

```
                         PDIF-IN

PDIF (ASCII) File   : <filename>.PDF





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

Figure 2-4. PDIF-IN Program Screen

2. Type the filename of the PDIF file to be read by PDIF-IN and press [Return]. If you do not enter the filename extension, PDIF-IN adds the .PDF extension.

   The system prompts for the P-CAD output filename. The default is the database filename specified in the PDIF input file (specified with the dbf parameter after the COMPONENT heading), as shown in Figure 2-5.

```
                    PDIF-IN

PDIF (ASCII) File    : TESTFILE.PDF
P-CAD (Binary) File  : TESTFILE.SYM





Enter the filename; Press [Return] to accept; [Esc] to reject.
```

Figure 2-5. Sample PDIF-IN Program Screen

3. Type the filename of the P-CAD output file to be created and press [Return], or press [Return] to accept the default.

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

   If a file already exists with the same name as the output file, the existing file is given a .OLD filename extension. If the <filename>.OLD file already exists, the system displays the following prompt.

   <filename>.OLD already exists!
   Replace(Y or N): YES

   Press [Return] to replace the file and proceed. If you do not want to proceed with the program until you rename the existing <filename>.OLD file, press the space bar to display "NO" and press [Return]. The system returns to the **PDIF (ASCII) File** prompt. Press [Esc] to exit, then select **Exit PDIF-IN** at the Opening Menu. From DOS, rename the file, then run PDIF-IN again.

   PDIF-IN 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 the input PDIF filename and, if you want, the output filename when you start PDIF-IN.

To use command line mode, at the DOS prompt, type PDIFIN followed by the filename(s), then press [Return].

The format of the command line is:

PDIFIN <infile> <outfile> <options>

where:

PDIFIN initiates the program.

infile is the name of the input netlist file and is required. If you do not include the filename extension, PDIF-IN assigns the .PDF extension.

outfile is the name of the output file to be created. This name is optional. If you do not specify a filename, PDIF-IN uses the output filename specified by the dbf parameter after the COMPONENT heading of the input PDIF file.

options is one or more of the options listed below that correspond to configuration file options. For each option, an "inverted" option is also shown; for example, -na is the inverted option of the -a option. If you do not specify either an option or its inverted option, the configuration file setting is used. If no configuration file exists, the inverted (-n) settings are the defaults.

-a instructs PDIF-IN to import attributes from component definitions.

-na instructs PDIF-IN not to import attributes from component definitions.

-e <filename> instructs PDIF-IN to use the named exception file for component definition import.

-ne instructs PDIF-IN not to use the exception file named in the configuration file for component definition import.

-s instructs PDIF-IN to use the 512K memory size for the TI computer.

-ns instructs PDIF-IN to use the 640K memory size.

-i instructs PDIF-IN to use internal component definition import.

-ni instructs PDIF-IN to use external component definition import.

-c instructs PDIF-IN to create component files from internal component definitions. It is effective only when internal component definition is used.

-nc instructs PDIF-IN to create component files from internal component definitions.

For those options not listed on the command line, configuration file values will be used. If there is no configuration file, the inverted (-n) option will be used.

Three examples of command lines are shown below.

```
PDIFIN TESTFILE  [Return]
PDIFIN TESTFILE.ASC -e TEST.EXC -ni  [Return]
PDIFIN TESTFILE TESTFILE.SCH -a  [Return]
```

The first example causes PDIF-IN to create a P-CAD database file from the TESTFILE.PDF file. The P-CAD file will have the filename specified by the dbf parameter after the COMPONENT heading of the input PDIF file.

The second example causes PDIF-IN to create a P-CAD database file from the TESTFILE.ASC file using external component import and using the TEST.EXC exception file. The P-CAD file will have the filename specified by the dbf parameter after the COMPONENT heading of the input PDIF file.

The third example causes PDIF-IN to create a P-CAD database file from the TESTFILE.PDF file, to import attributes from the component definitions, and to give the P-CAD file the TESTFILE.SCH filename.

After you enter the command line, the program runs automatically with no further input from you.

If a file already exists with the same name as the output file, the existing file is given a .OLD filename extension. If the file already exists with the same filename and .OLD extension, it is replaced.

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

CHAPTER 3. USING PDIF-OUT

This chapter describes the required conditions and procedures for configuring and running PDIF-OUT.

Before running PDIF-OUT, be sure that:

- Your system is correctly configured.
- You have installed the PDIF-OUT program file (PDIFOUT.EXE).
- You have created the database file to input into PDIF-OUT.

  NOTE: If the input file is a PCB database that includes padstack information, be sure that the special symbol file is connected (using the PC-CARDS SCMD/GSSF command). Otherwise, pad information will not be included in the PDIF file.

PDIF-OUT has two operating modes, interactive and command line. In interactive mode, PDIF-OUT displays a series of screens and you select options and specify filenames. In command line mode, you specify filenames when you start the program and PDIF-OUT processes the input file automatically.

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

INTERACTIVE MODE

In interactive mode, you must first start PDIF-OUT to display the Opening Menu.

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

    PDIFOUT [Return]

When the PDIF-OUT Title Screen appears, press any key to continue. The system then displays the PDIF-OUT Opening Menu as shown in Figure 3-1.

```
                    PDIF-OUT

               Options:

                    Configure PDIF-OUT

               >>   Run PDIF-OUT    <<

                    Exit PDIF-OUT



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

Figure 3-1. PDIF-OUT Opening Menu

This menu provides the following options.

Configure PDIF-OUT - Allows you to examine or change PDIF-OUT configuration options.

Run PDIF-OUT - Allows you to begin file processing.

Exit PDIF-OUT - Returns you to DOS.

If you want to change the configuration, use the procedure in the next section, "Configuring PDIF-OUT." If you do not want to change the configuration, continue to "Running PDIF-OUT."

Configuring PDIF-OUT

Configure PDIF-OUT allows you to set several PDIF-OUT system options. These settings are stored in the PDIFOUT.CFG file that is used each time you run the program. Default settings are provided and are used if you do not create the configuration file. You can use this menu selection to change the settings at any time.

To configure PDIF-OUT, follow the steps below.

1. At the PDIF-OUT Opening Menu, press the space bar or use the cursor keys to select **Configure PDIF-OUT**, then press [Return]. The system displays the Configuration Screen with the current settings, as shown in Figure 3-2.

PDIF-OUT Configuration

Output Format .... Compressed
Scan and Tag Reserved Characters .... No
Include Pin Names in SUBCOMP Section .... No

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

Figure 3-2. PDIF-OUT Configuration Screen

2. The first configuration option allows you to choose the format for the ASCII output file. The two formats are "Indented," which indents the levels of data in the file for easy readability, and "Compressed," which does not indent and produces a smaller database size. The default is "Compressed."

   You can accept the default setting of "Compressed" by pressing [Return]. To select "Indented," press the space bar to display the "Indented" setting, then press [Return].

3. The next option allows you to specify whether or not PDIF-OUT is to insert the backslash character before each reserved character encountered in the input file. This option is explained in "Reserved Characters" in Chapter 4. The default is "No."

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

4. The next option allows you to specify whether or not PDIF-OUT is to include pin names in the CN section of the output PDIF file. For more information, refer to the explanation of the CN section in Chapter 4. The default is "No."

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

   After you set the output format, the system displays the following prompt.

   Save Configuration : YES

5. To save the configuration for this session and future sessions, press [Return] to accept "YES." The system saves the modified configuration as a file named PDIFOUT.CFG.

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

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

Running PDIF-OUT

Run PDIF-OUT allows you to generate a PDIF file from a P-CAD file. To run PDIF-OUT, follow the steps below.

1. At the Opening Menu, select **Run PDIF-OUT** and press [Return].

   The system displays the PDIF-OUT Program Screen and prompts for the P-CAD database filename as shown in Figure 3-3.

PDIF-OUT

P-CAD (Binary) File : <Filename>

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

Figure 3-3. PDIF-OUT Program Screen

2. Type the filename of the P-CAD database file to be read by PDIF-OUT and press [Return]. If the filename has an extension, you must include the extension.

   The system prompts for the PDIF output filename. The default is the P-CAD database filename with the .PDF extension, as shown in Figure 3-4.

```
                    PDIF-OUT

P-CAD (Binary) File  : TESTFILE.SCH
PDIF (ASCII) File    : TESTFILE.PDF





 Enter the filename; Press [Return] to accept; [Esc] to cancel.
```

Figure 3-4. Sample PDIF-OUT Program Screen

3. Type the filename of the PDIF output file to be created and press [Return], or press [Return] to accept the default. If you specify a filename without an extension, PDIF-OUT adds the .PDF extension.

If a file already exists with the same name as the output file, the existing file is given a .BAK filename extension. If the <filename>.BAK file already exists, the system displays the following prompt.

<filename>.BAK already exists!
Replace (Y or N) : YES

Press [Return] to replace the file and proceed. If you do not want to proceed with the program until you rename the existing <filename>.BAK file, press the space bar to display "NO" and press [Return]. The system returns to the **P-CAD (Binary) File** prompt. Press [Esc] to exit, then select **Exit PDIF-OUT** at the Opening Menu. From DOS, rename the file, then run PDIF-OUT again.

PDIF-OUT 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 the input P-CAD filename and, if you want, the output PDIF filename when you start PDIF-OUT.

To use command line mode, at the DOS prompt, type PDIFOUT followed by the filename(s), then press [Return].

The format of the command line is:

PDIFOUT <infile> <outfile> <options>

where:

PDIFOUT initiates the program.

infile is the name of the input file and is required. It must include the extension if the filename has an extension.

outfile is the name of the output file to be created. This name is optional. If you do not specify a filename, PDIF-OUT uses the input filename with a .PDF extension. If you specify a filename with no extension, PDIF-OUT assigns the .PDF extension.

options is one or more of the options listed below, which correspond to configuration file options. For each option, an "inverted" option is also shown; for example, -ni is the inverted option of the -i option. If you do not specify either an option or its inverted option, the configuration file setting is used. If no configuration file exists, the inverted (-n) settings are the defaults.

-i instructs PDIF-OUT to create the PDIF file in indented format.

-ni instructs PDIF-OUT to create the output file in compressed format.

-r instructs PDIF-OUT to scan the input file for reserved characters and insert a backslash before each reserved character in the output file.

-nr instructs PDIF-IN not to scan for reserved characters.

-p instructs PDIF-OUT to include pin names in the output file.

-np instructs PDIF-OUT not to include pin names in the output file.

For those options not listed on the command line, configuration file values will be used. If there is no configuration file, the inverted (-n) option will be used.

Three examples of command lines are shown below.

```
PDIFOUT TESTFILE.PCB NEWFILE.ASC [Return]
PDIFOUT TESTFILE.PCB NEWFILE -i [Return]
PDIFOUT TESTFILE.SCH -r -p [Return]
```

The first example causes PDIF-OUT to create a PDIF file from the TESTFILE.PCB file. The PDIF file will have the filename NEWFILE.ASC.

The second example causes PDIF-OUT to create a PDIF file from the TESTFILE.PCB file. The PDIF file will have the filename NEWFILE.PDF and will be in indented format.

The third example causes PDIF-OUT to create a PDIF file from the TESTFILE.SCH file. The PDIF file will have the default filename TESTFILE.PDF, will have backslashes inserted before reserved characters, and will include pin names.

After you enter the command line, the program runs automatically with no further input from you.

If a file already exists with the same name as the output file, the existing file is given a .BAK filename extension. If a file already exists with the same filename and .BAK extension, it is replaced.

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

CHAPTER 4. THE PDIF FILE

The PDIF file is an ASCII file that you can create, examine and edit. It can describe a schematic symbol, a PCB part, a schematic, or a PCB design.

This chapter describes the structure and format of the PDIF file and contains explanations of each section of the file.

The PDIF file format described in this chapter is version 1.31. If you create a PDIF file using a text editor or a custom program, you must include this version number. For more information refer to the explanation of PDIFvrev in Chapter 5, "PDIF Keyword Reference."

FILE STRUCTURE

The PDIF file is designed to represent electronic design databases for schematic and PCB applications.

The PDIF file is organized into two major sections, SYMBOL and DETAIL, which correspond to the two views of a P-CAD design database file. This structure is shown in Figure 4-1.

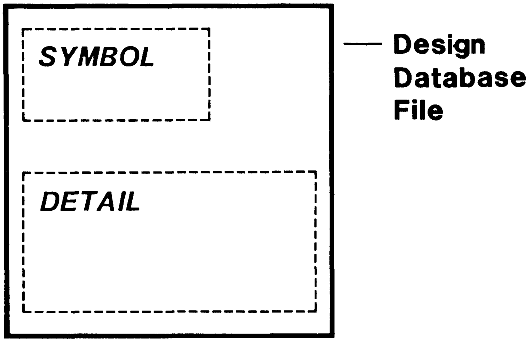
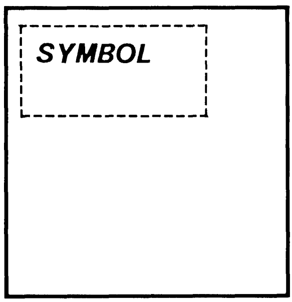


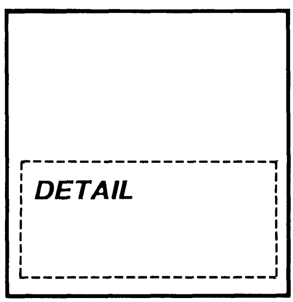
Figure 4-1. PDIF Database Structure

Like a P-CAD database file, a PDIF file can contain data in either or both of these two sections. Figure 4-2 shows the types of databases and the data they contain in each section.



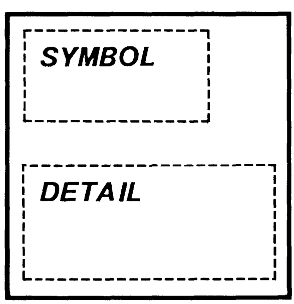
A library component; contains primarily SYMBOL data

Examples:
- A 2-input NAND gate logic symbol
- A 74LS00 14-pin DIP
- An 80286 microprocessor



A flat design; contains primarily DETAIL data

Examples:
- One sheet of a multisheet schematic design
- Top level of a hierarchical schematic design
- A PCB layout



Any database that has a "black-box" representation and the associated circuit; contains both SYMBOL and DETAIL data

Example:
A hierarchical element of a schematic database

Figure 4-2. PDIF File Structure for Different Database Types

Library components are stored in files that contain primarily SYMBOL data, including graphic data and pin information. Schematic and PCB database files contain primarily DETAIL data, including library components, their locations, and their connections.

PCB databases are constructed as flat designs. Schematic databases can be flat but often are constructed using multisheet and/or hierarchical design, which use more than one file per design. The following sections describe flat, multisheet, and hierarchical structure.

Flat Design Structure

A flat design consists of library components placed directly on the design, as shown in Figure 4-3.

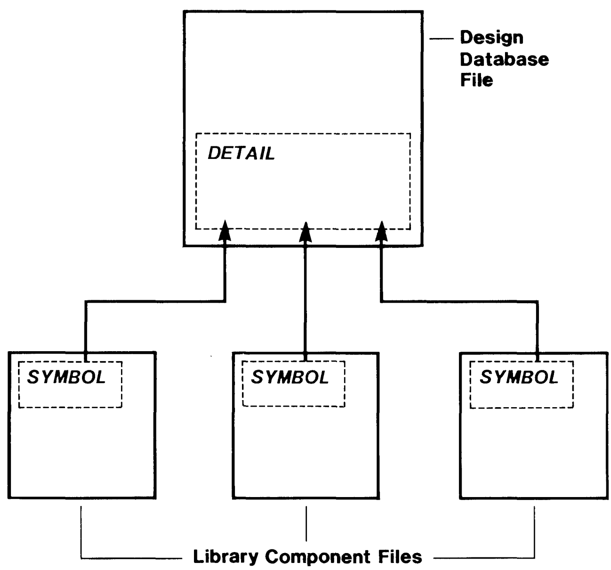


Figure 4-3. Flat Design Structure

The design database file contains primarily DETAIL data, while the library symbols contains primarily SYMBOL data. In the PDIF file, as in the P-CAD file, the SYMBOL data from the component files is contained in the DETAIL section of the design file.

Multisheet Schematic Structure

A schematic design can consist of several different schematic drawings that are interconnected. Each drawing is stored in a separate file and is called a sheet. Figure 4-4 shows multisheet structure.

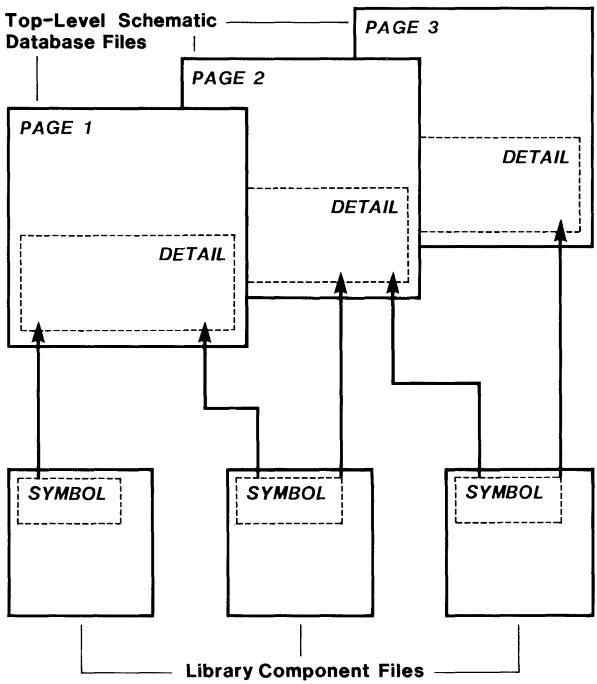


Figure 4-4. Multisheet Schematic Structure

Each sheet is a separate PDIF file. As in flat design, the SYMBOL data from the component file is contained in the DETAIL section of the schematic file.

Hierarchical Schematic Structure

Hierarchical structure uses symbols to represent subcircuits in a schematic design. Figure 4-5 shows this type of structure.

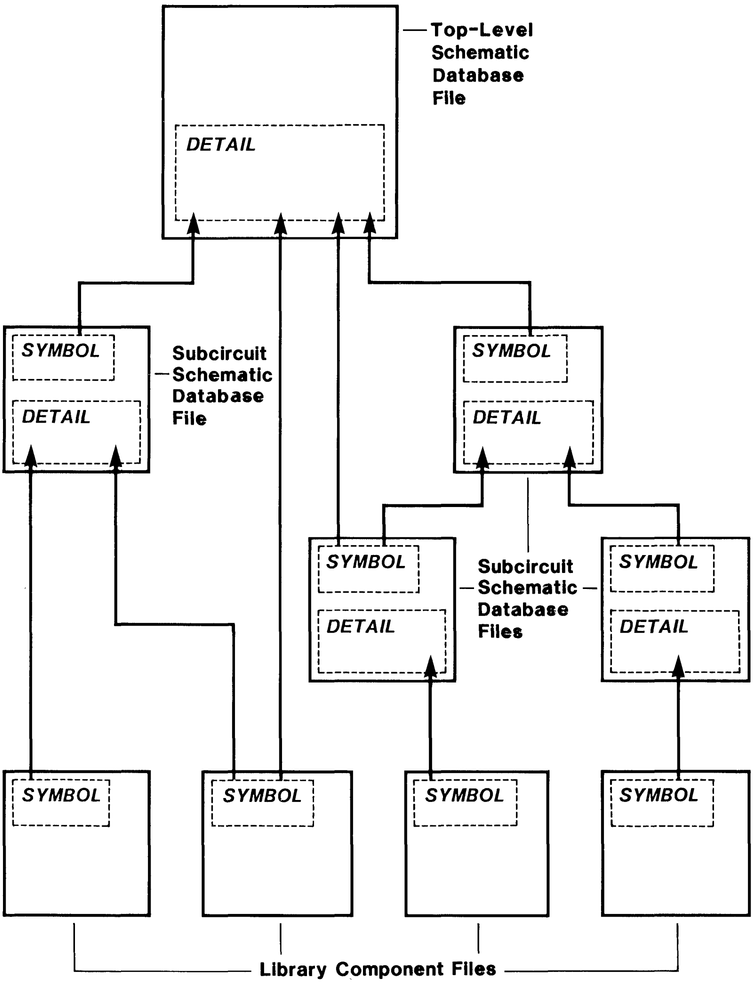


Figure 4-5. Hierarchical Schematic Structure

In Figure 4-5, each file shown is a separate PDIF database file. The top level is flat and contains primarily DETAIL data, while the library components contain primarily SYMBOL data. The user-defined subcircuits contain both SYMBOL and DETAIL data. The SYMBOL data at any level of hierarchy is contained in the DETAIL section of the next higher level.

PDIF Database Substructure

Figure 4-6 shows the PDIF database structure in more detail. The SYMBOL section of the database file contains all the data pertaining to the "black-box" representation of the design, while the DETAIL section contains the actual circuit represented by the "black box."

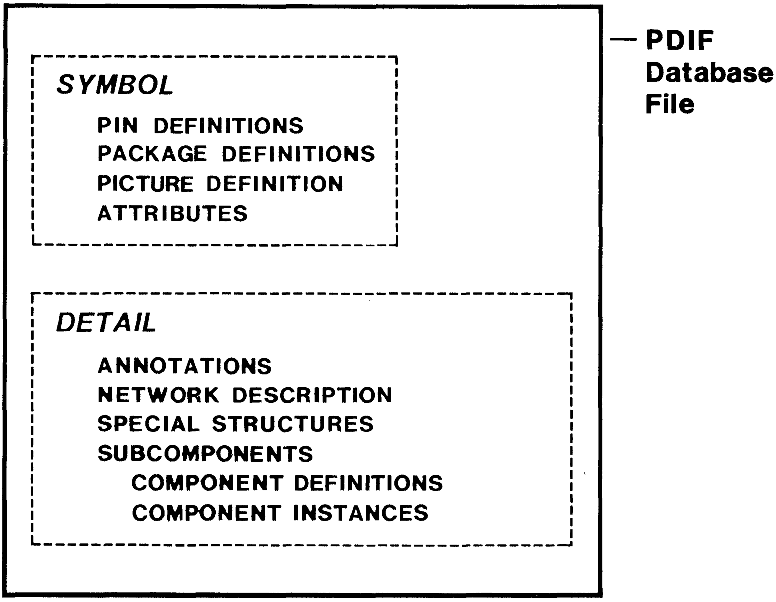


Figure 4-6. PDIF File Substructure

FILE FORMAT

The PDIF file consists of an identifier and five sections in the format shown below.

```
{COMPONENT dbf
 {ENVIRONMENT
  . . .
 }

 {USER
  . . .
 }

 {DISPLAY
  . . .
 }

 {SYMBOL
  . . .
 }

 {DETAIL
  . . .
 }
}
```

where:

COMPONENT identifies the file as a PDIF file.

dbf is the filename of the P-CAD database.

Blank lines and spaces can be inserted anywhere, and are a convenient way of partitioning data for readability.

ENVIRONMENT, **USER**, **DISPLAY**, **SYMBOL**, and **DETAIL** are the five sections of the file.

{ } (curly brackets) enclose the entire file and each section within the file.

. . . (ellipsis) indicates that the section contains more data.

Each section can contain data identified by PDIF keywords and values. A section can also have one or more subsections. Figure 4-7 shows the hierarchy of sections and subsections. (The periods shown have no significance; they are included to clarify the hierarchy only.)

```
COMPONENT
.   ENVIRONMENT
.   USER
.   .   VIEW
.   .   <system>
.   DISPLAY
.   SYMBOL
.   .   PIN_DEF
.   .   .   P
.   .   PKG
.   .   SPKG
.   .   PIC
.   .   ATR
.   .   .   IN
.   .   .   EX
.   DETAIL
.   .   ANNOTATE
.   .   NET_DEF
.   .   .   N
.   .   .   .   DG
.   .   .   .   ATR
.   .   PAD_STACK
.   .   .   PAD_DEF
.   .   .   .   ATR
.   .   .   .   .   IN
.   .   .   .   PIC
.   .   SUBCOMP
.   .   .   COMP_DEF
.   .   .   .   PIN_DEF
.   .   .   .   .   P
.   .   .   .   PKG
.   .   .   .   SPKG
.   .   .   .   PIC
.   .   .   .   ATR
.   .   .   .   .   IN
.   .   .   .   .   EX
.   .   .   I
.   .   .   .   CN
.   .   .   .   ASG
.   .   .   .   ATR
.   .   .   .   .   IN
.   .   .   .   .   EX
```

Figure 4-7. PDIF File Hierarchy

In some cases, the USER and/or DISPLAY sections might not be present. In addition, not all the subsections are used for all types of databases. For example, a schematic database does not contain the SPKG subsection. Some subsections must always be present but can be empty.

Note that some subsections are used more than once in the file. For example, all the subsections of COMP_DEF in the DETAIL section are the same as the subsections of the SYMBOL section.

Table 4-1 lists in alphabetical order the PDIF sections and subsections and the database types for which they are used. The table also shows the hierarchical "paths" in the PDIF file to each subsection, with a slash (/) between each section and the relevant subsection. For example, the COMP_DEF section's path is DETAIL/SUBCOMP, signifying that COMP_DEF is a subsection of SUBCOMP, which is a subsection of DETAIL. The COMPONENT heading, which is not listed in the table, is the "root" of all paths.

Table 4-1. PDIF File Sections and Subsections

Name	Database	Path
ANNOTATE	Sch, PCB	DETAIL
ASG	Sch	DETAIL/SUBCOMP/I
ATR	Sch, PCB	SYMBOL DETAIL/NET__DEF/N DETAIL/PAD__STACK/PAD__DEF DETAIL/SUBCOMP/COMP__DEF DETAIL/SUBCOMP/I
CN	Sch, PCB	DETAIL/SUBCOMP/I
COMP__DEF	Sch, PCB	DETAIL/SUBCOMP
DETAIL	Sch, PCB	
DG	Sch, PCB	DETAIL/NET__DEF/N
DISPLAY	Sch, PCB	
ENVIRONMENT	Sch, PCB	
EX	Sch, PCB	SYMBOL/ATR DETAIL/SUBCOMP/COMP__DEF/ATR DETAIL/SUBCOMP/I/ATR
I	Sch, PCB	DETAIL/SUBCOMP
IN	Sch, PCB	SYMBOL/ATR DETAIL/PAD__STACK/PAD__DEF/ATR DETAIL/SUBCOMP/COMP__DEF/ATR DETAIL/SUBCOMP/I/ATR
N	Sch, PCB	DETAIL/NET__DEF
NET__DEF	Sch, PCB	DETAIL
P	Sch, PCB	SYMBOL/PIN__DEF DETAIL/SUBCOMP/COMP__DEF/PIN__DEF
PAD__DEF	PCB	DETAIL/PAD__STACK
PAD__STACK	PCB	DETAIL
PIC	Sch, PCB	SYMBOL DETAIL/PAD__STACK/PAD__DEF DETAIL/SUBCOMP/COMP__DEF
PIN__DEF	Sch, PCB	SYMBOL DETAIL/SUBCOMP/COMP__DEF
PKG	Sch	SYMBOL DETAIL/SUBCOMP/COMP__DEF
SPKG	PCB	SYMBOL DETAIL/SUBCOMP/COMP__DEF
SUBCOMP	Sch, PCB	DETAIL
SYMBOL	Sch, PCB	
<system>	Sch, PCB	USER
USER	Sch, PCB	
VIEW	Sch, PCB	USER

RESERVED CHARACTERS

In PDIF file syntax, the following characters have specific meanings. They are called reserved characters.

% [ ] { } " blank

Component names, net names, text, or attributes in a P-CAD file sometimes contain reserved characters; for example, an attribute might contain quotation marks and blank spaces, or descriptive text might contain blank spaces.

To allow the use of reserved characters, when a reserved character in a PDIF file is preceded by a backslash (\), it is interpreted as a part of the text. For the blank, \b is used. If a backslash is used as a text character, a double backslash (\\) identifies it as text.

NOTE: When quotation marks are required with keywords for the PDIF syntax, the backslash is not required.

PDIF-OUT includes a configuration option that allows you to choose whether or not PDIF-OUT will search input P-CAD files for reserved characters and insert the \ character before each one. If you use any reserved characters in your P-CAD file, this option should be set to "Yes" before you run PDIF-OUT. If you take care not to use reserved characters, or if you do not plan to input the PDIF file into PDIF-IN, you can set this option to "No" and save time in file processing.

NOTE: A certain error in pin connection produces a "null" character in the P-CAD file. If PDIF-OUT encounters the null character, it generates an "Illegal pin connection" error message. If the configuration option is set to "Yes," the null character is represented by \0; if the configuration option is set to "No," the ? (unconnected pin) character is used for the pin in error.

When PDIF-IN encounters a backslash, it assumes that the next character is a reserved character to be used literally in the name, attribute, text. If PDIF-IN encounters a reserved character without the preceding backslash, an error may result. For example, if an attribute value contains a blank space, the characters after the blank space may be omitted. Or if the } character is used in a net name, PDIF-IN considers the } to be the end of the section.

FILE CONTENTS

This section gives a detailed description of each section and subsection of the PDIF file.

The description of each section has the following parts:

PATH: (Used for subsections only.) The hierarchical path of section and subsections that are "parent" to the subsection.

FUNCTION: Describes the purpose and contents of the section or subsection.

FORMAT: Shows the format of the section or subsection, including alternate formats if allowable and definitions of items if necessary.

KEYWORDS: Gives definitions for all keywords used in the section or subsection. Keywords are predefined words that are used to specify values for certain parameters. All the keywords used in the PDIF file are described in Chapter 5, "PDIF Keyword Reference."

EXAMPLES: Provides examples of the section or subsection.

The notation used in this chapter is shown in Table 4-2.

Table 4-2. PDIF File Notation

Symbol	Meaning
UPPER	Section and subsection names are indicated in uppercase. (PDIF recognizes either uppercase or lowercase.)
_	Keywords are underlined in the "format" listings. (They are not underlined in the PDIF file.)
lower	Variable names representing section, subsection, and keyword parameter values are shown in lowercase. (Actual values can be in uppercase or lowercase.)
" "	Quotation marks are required wherever they are shown.
. . .	The ellipsis is used to indicate that the section or keyword might contain additional data.

Table 4-2 Continued

Symbol	Meaning
%	The percent sign is used to indicate a comment. PDIF-IN ignores all text from the % to the end of the line.
{ }	Curly brackets enclose all sections and subsections and all keywords except for those enclosed in square brackets, which are described below.
[ ]	Square brackets enclose several keywords that define display and graphics information. These keywords are used in the PDIF file in the DISPLAY section to set initial conditions and are used again wherever these conditions change in the design.
x y	The x and y parameters refer to horizontal and vertical grid coordinates.

Sections and subsections are listed in the order used in the PDIF file. Subsections that are used in more than one section of the file are described only at their first occurrence. Some subsections are described with their parent sections.

ENVIRONMENT Section

FUNCTION:

This section describes the database environment used at the creation of the PDIF file.

FORMAT:

The ENVIRONMENT section produced by PDIF-OUT has the following format.

```
{ENVIRONMENT
  {PDIFvrev n.nn }
  {Program "PDIF-OUT Version n.nn"}
  {DBtype "type" }
  {DBvrev n.nn }
  {DBtime "time_stamp" }
  {DBunit "unit" }
  {DBgrid n }
  {Lyrstr "layer" color "layer" color . . . }
}
```

A user-created PDIF file to be input to PDIF-IN can have the same ENVIRONMENT format or it can have an alternate format, which is described below.

When an ENVIRONMENT section in the format above is input into PDIF-IN, the DBtype and Lyrstr information is required; if it is not present, PDIF-IN is unable to process the file.

The alternate format for the ENVIRONMENT section is shown below.

ENVIRONMENT Section

```
{ENVIRONMENT
 file
 {PDIFvrev n.nn}
}
```

where:

file is the name of a file that contains beginning environment settings; for example, a board outline or a layer structure file.

With either format, if the PDIFvrev keyword is not present, PDIF-IN assumes version 1.30.

KEYWORDS:

PDIFvrev	- PDIF version number
Program	- The PDIF-OUT version used to create the file
DBtype	- Database type, schematic or PCB
DBvrev	- Source database version number
DBtime	- Database creation date and time
DBunit	- Database unit
DBgrid	- Database grid definition
Lyrstr	- Layer structure definition

ENVIRONMENT Section

EXAMPLES:

Example 1 - Shows an ENVIRONMENT section produced by PDIF-OUT.

```
{COMPONENT LAT1.SYM
 {ENVIRONMENT
  {PDIFvrev 1.31}
  {Program "PDIF-OUT Version 1.31"}
  {DBtype "Schematic"}
  {DBvrev 1.00}
  {DBtime "Jun. 13, 1986    7:09 p.m.     "}
  {DBunit "MIL"}
  {DBgrid 10}
  {Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2
  "PINFUN" 3  "PINNUM" 1  "PINNAM" 6
  "PINCON" 4 "REFDES" 2 "ATTR" 6 "ATTR2" 1
  "DEVICE" 5 "NETNAM" 7 "CMPNAM" 8 "SHEET" 10}
 }
```

Example 2 - Shows the alternative format.

```
{ENVIRONMENT BOARDB.PCB}
```

USER Section

FUNCTION:

This section includes system-specific data in one or more subsections.

The VIEW subsection contains information that is used only by the P-CAD system. It is always generated by PDIF-OUT.

The <system> subsection, where <system> is the name of another CAE/CAD/CAM system, contains information used only by that system. This subsection is not generated by PDIF-OUT and is not recognized by PDIF-IN; it exists only to enable the user to manually enter information necessary for the other system. The <system> subsection can have as many user-defined subsections and keywords as needed for the relevant data. A PDIF file can contain more than one <system> subsection if necessary.

USER Section

FORMAT:

```
{USER
 {VIEW
  {Mode mode}
  {Nlst St}
  {Vw ix iy is n llx lly urx ury n llx lly urx ury . . . }
  {Lv a s1 s2 s3 . . . }
  {Gs x y}
  }
  {<system>
   {key . . . }
   {sec . . .
    {key . . . }
    . . .
   }
   . . .
  }
}
```

where:

key is a user-defined keyword.

sec is the name of a user-defined subsection.

USER Section

KEYWORDS:

Mode - Database editing mode

Nlst - Netlist status

Vw - Viewing windows

Lv - Layer view status

Gs - Grid spacing

EXAMPLES:

```
{USER
   {VIEW
   {Mode SYMB}
   {Nlst OPEN}
   {Vw 0 0 1 1 -15 -50 515 310}
   {Lv 9 2 0 2 1 0 2 2 2 2 0 0 1 2 }
   {Gs 10 10}
  }
}
```

DISPLAY Section

FUNCTION:

This section defines the initial values for certain parameters that are used in the other sections. These parameters are global; that is, they are "visible" throughout the PDIF file.

The values specified in this section are the initial conditions. They can be changed anywhere in the PDIF database. Once defined, a parameter value is used until it is redefined.

This section is always included in the PDIF file produced by PDIF-OUT. It can be omitted from a PDIF file to be input to PDIF-IN if the ENVIRONMENT section specifies a file that contains beginning parameters.

REMARKS:

Each time PDIF-OUT begins a new section, it will output the display status values even if display status has not changed since the previous section. This facilitates editing the PDIF-OUT file by keeping track of the current status for the user.

FORMAT:

```
{DISPLAY
  [Ly "layer"]
  [Ls "style"][Wd n]
  [Ts n][Ti "hv"][Tr n][Tm "m"]
}
```

DISPLAY Section

KEYWORDS:

Ly - Active layer

Ls - Line style

Wd - Line width

Ts - Text size

Tj - Text justifications

Tr - Text rotation angle

Tm - Text mirror flag

EXAMPLES:

```
{DISPLAY
  [Ly "GATE"]
  [Ls "SOLID"][Wd 0]
  [Ts 25][Tj "LB"][Tr 0][Tm "N"]
}
```

SYMBOL Section

FUNCTION:

This section and the DETAIL section are the major sections of the PDIF database. The SYMBOL section consists of information pertaining to the component, including pin definitions, packaging information, and graphics.

The SYMBOL section is organized into the major subsections shown in the format below. Each subsection is described separately below and examples are shown with the descriptions of the subsections.

The PIN_DEF and PIC subsections are empty when the database has no SYMBOL data, as with a top-level schematic database or a PCB design.

The PKG subsection specifies packaging information for a logic symbol, and therefore is present only for schematic symbol databases and packagable subcircuits.

The SPKG subsection specifies gate-pin names for PCB parts, and therefore is present only for PCB part databases.

The SYMBOL section is not required when importing a netlist into PDIF-IN. (See Appendix C, "Netlist Import.")

SYMBOL Section

FORMAT:

```
{SYMBOL
 {PIN_DEF
   ...
 }
 {PKG        % Schematic databases only
   ...
 }
 {SPKG       % PCB databases only
   ...
 }
 {PIC
   ...
 }
 {ATR
   ...
 }
}
```

PIN_DEF

PATH:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

The PIN_DEF subsection defines the pins in the component. It consists of one P subsection for each pin. Each P subsection gives the name and type of the pin, its logical equivalency, and its location. Pin ordering for the component is the same as the order of the P subsections. The P subsection is described here with PIN_DEF.

FORMAT:

```
{PIN_DEF
  {P pinname {Pt "type"}{Lq n}{Ploc x y}}
  {P pinname {Pt "type"}{Lq n}{Ploc x y}}
  . . .
}
```

where:

pinname is the name of the pin (entered in the P-CAD system using the SYMB:ENTR/PIN command).

PIN_DEF

KEYWORDS:

Pt - Pin type

Lq - Logical equivalency group

Ploc - Pin location coordinates in DBUs

> NOTE: When used in the SYMBOL section, pin locations are relative to the database origin (as is the symbol location, given by SYMBOL/ATTR/IN/ORG).
>
> When used in the DETAIL/SUBCOMP\COMP_DEF section, pin locations are relative to the component origin (set with the SYMB:ENTR/ORG command).

EXAMPLES:

Example 1 - Shows the PIN_DEF subsection for a 2-input NAND gate schematic library symbol database.

```
{PIN_DEF
 [Ly "PINCON"]
 {P Y {Pt "OUTPUT"}{Lq 0}{Ploc 90 -20}}
 {P A {Pt "INPUT"}{Lq 1}{Ploc -90 0}}
 {P B {Pt "INPUT"}{Lq 1}{Ploc -90 -40}}
}
```

Example 2 - Shows the PIN_DEF subsection for the 74LS00 physical part database corresponding to the NAND gate in Example 1.

PIN_DEF

```
{PIN_DEF
 [Ly "PIN"]
 {P 1 {Pt 1}{Lq 1}{Ploc -200 550}}
 {P 2 {Pt 2}{Lq 1}{Ploc -200 450}}
 {P 3 {Pt 2}{Lq 0}{Ploc -200 350}}
 {P 4 {Pt 2}{Lq 2}{Ploc -200 250}}
 {P 5 {Pt 2}{Lq 2}{Ploc -200 150}}
 {P 6 {Pt 2}{Lq 0}{Ploc -200 50}}
 {P 7 {Pt 3}{Lq 0}{Ploc -200 -50}}
 {P 8 {Pt 2}{Lq 0}{Ploc 100 -50}}
 {P 9 {Pt 2}{Lq 3}{Ploc 100 50}}
 {P 10 {Pt 2}{Lq 3}{Ploc 100 150}}
 {P 11 {Pt 2}{Lq 0}{Ploc 100 250}}
 {P 12 {Pt 2}{Lq 4}{Ploc 100 350}}
 {P 13 {Pt 2}{Lq 4}{Ploc 100 450}}
 {P 14 {Pt 4}{Lq 0}{Ploc 100 550}}
}
```

PKG

PATH:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

The PKG subsection is used only with schematic databases and defines packaging information for a schematic symbol. The definition consists of locations where the reference designator and the pin numbers will be shown when the symbol is placed on a schematic and the mappings of logic gates to physical package sections.

REMARKS:

Pnl subsections must be in the same order as the associated P subsections for the component.

FORMAT:

```
{PKG
  {Rdl x y}

  {Pnl x y}
  {Pnl x y}
  . . .

  {Sd s n1 n2 n3 . . . }
  {Sd s n1 n2 n3 . . . }
  . . .
}
```

PKG

KEYWORDS:

Rdl - Location for displaying the reference designator

Pnl - Locations for displaying the pin numbers

Sd - Package section and pin numbers

EXAMPLES:

The example below shows the PKG subsection for a 2-input NAND gate. Notice the section definitions and package pin numbers.

```
{PKG
 [Ly "REFDES"]
 [Ts 15][Tj "CB"]
 {Rdl -10 30}

 [Ly "PINNUM"]
 [Tj "LB"]
 {Pnl 65 -15}
 {Pnl -65 5}
 {Pnl -65 -35}

 {Sd A 3 1 2}
 {Sd B 6 4 5}
 {Sd C 8 9 10}
 {Sd D 11 12 13}
}
```

SPKG

PATH:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

SPKG contains the gate name mapping information for packaged parts. This subsection is present only for PCB databases.

FORMAT:

```
{SPKG
  {Sp spn n1 n2 n3 . . . }
  {Sp spn n1 n2 n3 . . . }
  . . .
}
```

KEYWORDS:

Sp - Symbol pin mapping

EXAMPLES:

```
{SPKG
  {Sp INA 1 4 9 12}
  {Sp INB 2 5 10 13}
  {Sp OUTY 3 6 8 11}
}
```

PIC

PATH:

SYMBOL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

The PIC subsection includes all graphical data that makes up the symbol. This data usually includes the symbol graphics and annotation texts.

FORMAT:

```
{PIC
  {T "string" x y}
  {L x1 y1 x2 y2 x3 y3 . . . }
  {R x1 y1 x2 y2}
  {Fr x1 y1 x2 y2}
  {A cx cy r sa ea}
  {C cx cy r}
  {Fl x y a}
}
```

PIC

KEYWORDS:

T - Text string

L - Line segments

R - Rectangle

Fr - Filled rectangle

A - Arc

C - Circle

Fl - Flash (for photoplotter graphics)

PIC

EXAMPLES:

Example 1 - Describes the picture of a 2-input NAND gate and accompanying text.

```
{PIC
 [Ly "GATE"]
 {L 60 -20 90 -20}
 {A 0 -20 41 284 76}
 {C 50 -20 10}
 {L 10 20 -60 20 -60 -60 10 -60}
 {L -60 0 -90 0}
 {L -90 -40 -60 -40}
 [Ly "PINNAM"]
 [Tj "LT"]
 {T "Y" 65 -25}
 [Tj "LC"]
 {T "A" -55 0}
 {T "B" -55 -40}
 [Ly "DEVICE"]
 [Tj "CT"]
 {T "74LS00" -25 -60}
}
```

Example 2 - Shows the picture of a 14-pin DIP (74LS00 package).

```
{PIC
 [Ly "SLKSCR"]
 {L -100 600 -50 550 0 600}
 {R -150 -100 50 600}
 [Ly "DEVICE"]
 [Ts 275]
 {T "7400" -75  250}
}
```

ATR

PATH:

SYMBOL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

The ATR subsection defines the symbol attributes. The two subsections, IN and EX, are also described in this section.

The IN subsection defines internal attributes and includes the origin (reference point) for the symbol and the symbol's component type.

The EX subsection defines external, or user-defined attributes such as timing or simulation parameters, manufacturing part numbers, and so on. A database can have any number of user-defined attributes.

Note that the ATR subsection under DETAIL/SUBCOMP/I uses different keywords and is described separately in this chapter.

REMARKS:

The Org subsection is not included within the COMP_DEF section.

ATR

FORMAT:

```
{ATR
 {IN
  {Org x y}
  {Ty n}
 }
 {EX
  {At key value x y}
  {At key value x y}
  . . .
 }
}
```

KEYWORDS:

Org - Reference point for the symbol

Ty - Component type

At - User-defined attribute

ATR

EXAMPLES:

```
{ATR
 {IN
  {Org -90 0}
  {Ty 2}
 }
 {EX
  [Ly "ATTR"]
  [Ts 10][Tj "LT"]
  {At PCL (1,1,"D","D") -60 -100}
  {At PART MT-CC08-85065A -60 -80}
 }
}
```

DETAIL Section

FUNCTION:

This section and the SYMBOL section are the major sections of the PDIF database. The DETAIL section consists of all data pertaining to the actual circuit, including all drawing/layout annotations, the electrical wiring or network, and the components and their connectivities.

The DETAIL section is organized into the major subsections shown in the format below. Each subsection is described separately below, and examples are shown with the descriptions of the subsections.

The PAD_STACK subsection is present only for PCB databases, as it contains information about PCB vias (feedthrus) and padstacks, which do not exist in schematic databases.

FORMAT:

```
{DETAIL
 {ANNOTATE        % may be absent
   ...
 }
 {NET_DEF         % may be absent
   ...
 }
 {PAD_STACK       % PCB databases only
   ...
 }
 {SUBCOMP
   ...
 }
}
```

ANNOTATE

PATH:

DETAIL

FUNCTION:

This subsection describes all drawing or layout annotations that are not pictures of the symbols, reference designators or instance names, pin names/numbers or wiring diagrams. For example, this subsection would contain a board outline or text placed on a design for user reference.

The format and the keywords of the ANNOTATE subsection are identical to those in the SYMBOL/PIC subsection described earlier.

FORMAT:

```
{ANNOTATE
  {T "string" x y}
  {L x1 y1 x2 y2 x3 y3 . . . }
  {R x1 y1 x2 y2}
  {Fr x1 y1 x2 y2}
  {A cx cy r sa ea}
  {C cx cy r}
  {Fl x y a}
}
```

ANNOTATE

KEYWORDS:

T - Text string

L - Line segments

R - Rectangle

Fr - Filled rectangle

A - Arc

C - Circle

Fl - Flash (for photoplotter graphics)

EXAMPLES:

```
{ANNOTATE
  {C 20 100 20}
  {C 180 100 20}
  {A 100 20 205 209 331}
  {C 100 20 260}
  {Fr -400 140 -220 260}
  {T "Sheet 1 of 2" 20 -300}
  {L -440 -260 480 -260}
  {R -440 -340 480 300}
}
```

NET_DEF

PATH:

DETAIL

FUNCTION:

NET_DEF describes the circuit "wiring diagram," including wire graphics, signal names, and attributes.

The NET_DEF subsection consists of a series of N subsections, one for each signal in the database. The N subsection is described in the following section, and examples are given there.

FORMAT:

```
{NET_DEF
  {N net_name
    . . .
  }
  {N net_name
    . . .
  }
  . . .
}
```

where:

> **net_name** is the name of the net, entered in the P-CAD system using the DETL:NAME/NET command.

N

PATH:

DETAIL/NET_DEF

FUNCTION:

Each N subsection completely describes one signal. The two subsections, DG and ATR, are also described below.

The DG subsection consists of the pictorial data, including locations for signal names.

The ATR subsection tells whether the net is local or global.

FORMAT:

```
{N net_name
  {DG
    {W x1 y1 x2 y2 . . . }
    {W x1 y1 x2 y2 . . . }
    . . .
    {V x y viatype}
    {V x y viatype}
    . . .
    {Nn x y x y . . . }
  }
  {ATR
    {IN
      {Ns s}
    }
  }
}
```

N

KEYWORDS:

W - Specifies a wire segment or sequence of wire segments

V - Specifies via locations for PCB databases or solder dot locations for schematic databases

Nn - Specifies locations for net names to be displayed

Ns - Specifies if a net is global

EXAMPLES:

Example 1 - Shows the NET_DEF subsection for a schematic.

```
{NET_DEF
 {N INPUT1
  {DG
   [Ly "WIRES"]
   {V 140 140 15}
   {W 0 140 -220 140 -220 120}
   {W -150 140 -310 140}
   [Ly "NETNAM"]
   [Ts 15]
   {Nn -310 140 0 140}
  }
 }
 {N XN00000
  {DG
   [Ly "WIRES"]
   {W -220 -20 -220 -80 -150 -80}
  }
 }
}
```

N

Example 2 - Shows the NET_DEF subsection for a PCB design.

```
{NET_DEF
 {N XN00000
  {DG
   [Ly "COMPNT"]
   {W 200 400 300 400}
   {W 300 100 425 100}
   [Ly "SOLDER"]
   {W 300 400 300 100}
   {V 300 400 0}
  }
 }
 {N XN00001
  {DG
   {W -625 850 -475 850}
   {V -625 850 0}
   {V -475 850 0}
   [Ly "COMPNT"]
   {W -625 575 -625 850}
   {W -475 850 -325 850}
  }
 }
}
```

PAD_STACK

PATH:

DETAIL

FUNCTION:

The PAD_STACK subsection describes all the pads and vias (feedthrus) in a PCB database. This subsection is used with PCB databases only.

The PAD_STACK subsection includes one PAD_DEF subsection for each type of pad or via used in the database. The PAD_DEF subsection is also described below.

Each PAD_DEF subsection has two subsections, PIC, which defines the graphic information, and ATR, which defines attributes. This data includes the pictorial representation of the pad or via and any attributes, and may contain data for photoplotting, drill tapes, and other manufacturing-related information.

The PIC and ATR subsections are also used in the SYMBOL section. Refer to the descriptions of the SYMBOL subsections for further information.

In order for PDIF-OUT to generate pad information, the input P-CAD PCB database file must be linked to the padstacks. (The padstacks are linked by using the PC-CARDS DETL:SCMD/GSSF command.)

PAD_STACK

FORMAT:

```
{PAD_STACK
 {Pad type "name" "name"}
 {Pad type "name" "name"}
 {Pad type "name" "name"}
 . . .
 {PAD_DEF name
  {ATR
   . . .
  }
  {PIC
   . . .
  }
 }
}
```

where:

> **pad_type** is the number that defines a pad or via. In the P-CAD system, pad_type values are defined in the special symbol file.

KEYWORDS:

Pad - Identifies a pad or via

PAD_STACK

EXAMPLES:

```
{PAD_STACK
 {Pad 0 "V50R28.PS" "V50R28.PS"}
 {Pad 1 "C60S32.PS" "N60S32.PS"}
 {Pad 2 "C60R32.PS" "N60R32.PS"}
 {Pad 3 "C60R32.PS" "N60R32.PS"}
 {Pad 4 "C60R32.PS" "N60R32.PS"}
 {Pad 5 "C60R32.PS" "N60R32.PS"}
 {PAD_DEF V50R28.PS
  {ATR
   {IN
    {Org 0 0}
    {Ty 32767}
   }
  }
  {PIC
  [Ly "SLDMSK"]
  [Ls "SOLID"] [Wd 0]
  [Ts 125] [Tj "CC"] [Tr 1] [Tm "N"]
  {C 0 0 40}
  [Ly "FLSMSK"]
  {F1 0 0 17}
  [Ly "FLCOMP"]
  {F1 0 0 15}
  [Ly "FLDRLL"]
  {F1 0 0 23}
  [Ly "DRILL"]
  {Ly -10 0 10 0}
  {L 0 10 0 -10}
  [Ts 60] [Tr 0]
  {T "28" 5 -5}
  [Ly "PADINT"]
  {C 0 0 25}
  }
 }
```

PAD_STACK

```
{PAD_DEF N60S32.PS
 {ATR
  {IN
   {Org 0 0}
   {Ty 32767}
  }
 }

 {PIC
 [Ly "PADCOM"]
 [Ls "SOLID"] [Wd 0]
 [Ts 60] [Tj "CC"] [Tr 0] [Tm "N"]
 {R -30 -30 30 30}
 [Ly "SLDMSK"]
 {C 0 0 45}
 [Ly "FLCOMP"]
 {F1 0 0 10}
 [Ly "FLDRLL"]
 {F1 0 0 23}
 [Ly "FLSMSK"]
 {F1 0 0 19}
 [Ly "DRILL"]
 {L -10 0 10 0}
 {L 0 10 0 -10}
 {T "32" 5 -5}
 [Ly "PIN"]
 {C 0 0 25}
 }
}
{PAD_DEF C60S32.PS
 {ATR
  {IN
   {Org 0 0}
   {Ty 32767}
  }
 }
```

PAD_STACK

```
  {PIC
  [Ly "PADCOM"]
  [Ls "SOLID"] [Wd 0]
  [Ts 60] [Tj "CC"] [Tr 0] [Tm "N"]
  {R -30 -30 30 30}
  [Ly "SLDMSK"]
  {C 0 0 45}
  [Ly "FLCOMP"]
  {Fl 0 0 10}
  [Ly "FLDRLL"]
  {Fl 0 0 23}
  [Ly "FLSMSK"]
  {Fl 0 0 19}
  [Ly "DRILL"]
  {L -10 0 10 0}
  {L 0 10 0 -10}
  {T "32" 5 -5}
  [Ly "PIN"]
  {Fr -25 -25 25 25}
  }
}
{PAD_DEF N60R32.PS
  {ATR
    {IN
      {Org 0 0}
      {Ty 32767}
    }
  }

  {PIC
  [Ly "FLDRLL"]
  [Ls "SOLID"] [Wd 0]
  [Ts 60] [Tj "CC"] [Tr 0] [Tm "N"]
  {Fl 0 0 23}
  [Ly "FLSMSK"]
  {Fl 0 0 19}
```

PAD_STACK

```
  [Ly "FLCOMP"]
  {F1 0 0 9}
  [Ly "SLDMSK"]
  {C 0 0 45}
  [Ly "PADCOM"]
  {C 0 0 30}
  [Ly "DRILL"]
  {L -10 0 10 0}
  {L 0 10 0 -10}
  {T "32" 5 -5}
  [Ly "PIN"]
  {C 0 0 25}
  }
}
{PAD_DEF C60R32.PS
  {ATR
    {IN
      {Org 0 0}
      {Ty 32767}
    }
  }

{PIC
[Ly "FLDRLL"]
[Ls "SOLID"] [Wd 0]
[Ts 60] [Tj "CC"] [Tr 0] [Tm "N"]
{F1 0 0 23}
[Ly "FLSMSK"]
{F1 0 0 19}
[Ly "FLCOMP"]
{F1 0 0 9}
[Ly "SLDMSK"]
{C 0 0 45}
[Ly "DRILL"]
{L -10 0 10 0}
{L 0 10 0 -10}
```

PAD_STACK

```
    {T "32" 5 -5}
    [Ly "PADCOM"]
    {C 0 0 30}
    [Ly "PIN"]
    [Wd 25]
    {C 0 0 15}
    }
  }
}
```

SUBCOMP

PATH:

DETAIL

FUNCTION:

The SUBCOMP subsection defines the components used in the design. It specifies the parts used, their instance names, their pins and the pin connections to the signals defined in the NET_DEF subsection.

The SUBCOMP subsection is organized into the subsections shown in the format below. Each subsection is described separately below, and examples are shown with the descriptions of the subsections.

In the PDIF file for a schematic or physical component, only the SUBCOMP heading is present and none of the subsections are used.

SUBCOMP

FORMAT:

In the PDIF file produced by PDIF-OUT, the SUBCOMP subsection contains the definition of a component followed by all its instances, followed by the definition of the next component, and so forth, as shown below.

```
{SUBCOMP
 {COMP_DEF filename     % Component definition
   . . .
 }
 {I filename inv_name    % Each instance
   . . .
 }
 {I filename inv_name    % Next instance
   . . .
 }
 . . .
 {COMP_DEF filename     % Define next component
   . . .
 }
 . . .
}
```

where:

filename is the name of the component file.

inv_name is the invocation name of the component instance (assigned in the P-CAD system using the DETL:NAME/COMP command).

PDIF-IN also recognizes an alternate format, useful with some CAD systems with which it is more

SUBCOMP

convenient to list all the definitions first, and all the instances afterwards, as shown below. The instances can be in any order.

```
{SUBCOMP

  {COMP_DEF filename      % Component definition
    . . .
  }
  {COMP_DEF filename      % Next component definition
    . . .
  }
  . . .

  {I filename inv_name    % Instances
    . . .
  }
  {I filename inv_name
    . . .
  }
  {I filename inv_name
    . . .
  }
  . . .
}
```

COMP_DEF

PATH:

DETAIL/SUBCOMP

FUNCTION:

Each COMP_DEF subsection defines a component used in a design database. A PDIF file contains as many COMP_DEF subsections as there are unique components used.

The COMP_DEF subsection for a component in a design contains the same information that would be in the SYMBOL section of the PDIF file for the component if it were created.

The COMP_DEF subsection contains all the same subsections as the SYMBOL section. The ATR/IN subsection produced by PDIF-OUT has an additional keyword, Vr (component version number). The ATR/EX subsection is not produced by PDIF-OUT in the COMP_DEF subsection, but is recognized by PDIF-IN. The PKG subsection is present only for schematic databases and the SPKG subsection is present only for PCB part databases. Refer to the descriptions of the SYMBOL subsections for further information.

PDIF-OUT does not produce a COMP_DEF subsection in the PDIF file for a database with no DETAIL data, such as a schematic library symbol or PCB part.

COMP_DEF

When creating a PDIF file to input to PDIF-IN, the COMP_DEF subsection is not necessary, if an external symbol or part file exists for the component and the -ni option is selected when running PDIF-IN.

FORMAT:

```
{COMP_DEF filename
 {PIN_DEF
   . . .
 }
 {PKG
   . . .
 }
 {SPKG
   . . .
 }
 {PIC
   . . .
 }
 {ATR
   . . .
 }
}
```

where:

filename is the name of the component file.

COMP_DEF

EXAMPLES:

Example 1 - Defines a 2-input NAND gate schematic symbol (74LS00).

```
{COMP_DEF 74LS00.SYM
 {PIN_DEF
  [Ly "PINCON"]
  {P INA {Pt "INPUT"}{Lq 1}{Ploc 0 40}}
  {P INB {Pt "INPUT"}{Lq 1}{Ploc 0 0}}
  {P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 170 20}}
 }
 {PKG
  [Ly "REFDES"]
  [Ts 25][Tj "CC"]
  {Rdl 80 20}

  [Ly "PINNUM"]
  [Ts 15][Tj "RC"]
  {Pnl 20 50}
  {Pnl 20 10}
  [Tj "LC"]
  {Pnl 160 30}

  {Sd A 1 2 3}
  {Sd B 4 5 6}
  {Sd C 10 9 8}
  {Sd D 12 13 11}
 }
```

COMP_DEF

```
{PIC
  [Ly "GATE"]
  {C 140 20 10}
  {L 150 20 170 20}
  {A 90 20 40 270 90}
  {L 90 60 30 60 30 -20 90 -20}
  {L 30 40 0 40}
  {L 30 0 0 0}
  [Ly "IEEE"]
  {L 130 30 150 20}
  [Tj "CB"]
  {T "&" 80 40}
  {L 130 20 170 20}
  {L 0 40 30 40}
  {L 0 0 30 0}
  {R 30 -20 130 60}
  [Ly "DEVICE"]
  [Tj "CC"]
  {T "74LS00" 70 -30}
 }
 {ATR
  {IN
   {Org -80 180}
   {Ty 4}
   {Vr 0}
  }
 }
}
```

COMP_DEF

Example 2 - Defines a 7404T PCB part.

```
{COMP_DEF 7404T.PRT
 {PIN_DEF
  [Ly "PIN"]
  {P 1 {Pt 1}{Lq 0}{Ploc 0 0}}
  {P 21 {Pt 2}{Lq 0}{Ploc 0 -100}}
  {P 3 {Pt 2}{Lq 0}{Ploc 0 -200}}
  {P 4 {Pt 2}{Lq 0}{Ploc 0 -300}}
  {P 5 {Pt 2}{Lq 0}{Ploc 0 -400}}
  {P 6 {Pt 2}{Lq 0}{Ploc 0 -500}}
  {P 7 {Pt 3}{Lq 0}{Ploc 0 -600}}
  {P 8 {Pt 2}{Lq 0}{Ploc 300 -600}}
  {P 9 {Pt 2}{Lq 0}{Ploc 300 -500}}
  {P 10 {Pt 2}{Lq 0}{Ploc 300 -400}}
  {P 11 {Pt 2}{Lq 0}{Ploc 300 -300}}
  {P 12 {Pt 2}{Lq 0}{Ploc 300 -200}}
  {P 13 {Pt 2}{Lq 0}{Ploc 300 -100}}
  {P 14 {Pt 4}{Lq 0}{Ploc 300 0}}
 }
 {PIC
  [Ly "SLKSCR"]
  [Wd 15]
  {A 150 50 50 180 0}
  {L 200 50 250 50 250 -650 50 -650 50 50 100 50}
  [Ly "DEVICE"]
  {T "7404" 150 -300}
 }
 {ATR
  {IN
   {Org 150 650}
   {Ty 32767}
   {Vr 0}
  }
 }
}
```

I

PATH:

DETAIL/SUBCOMP

FUNCTION:

The I subsection describes the unique properties of each instance of a component defined in COMP_DEF, as well as the component-to-pin connections.

The I subsection has three major subsections, CN, ASG, and ATR, which are described separately below.

FORMAT:

```
{I filename inv_name
 {CN
   ...
 }
 {ASG                          % Schematic databases only
   ...
 }
 {ATR
   ...
 }
}
```

I

where:

filename is the name of the component, as used in COMP_DEF.

inv_name is the invocation name of the component. In the P-CAD system, this name is assigned using the DETL:NAME/COMP command. If the component was not named, PDIF-OUT assigns a name in the format XCnnnnn, where nnnnn is a five-digit integer and is assigned sequentially starting with 00000.

EXAMPLES:

Example 1 - Shows the I subsection for a schematic database.

```
{I 74107.SYM XC00000
 {CN J XN0001 CP XN00026 K XN00002 RD CLR
Q' QA' Q QA}
 {ASG
  [Ly "REFDES"]
  [Ts 25][Tj "CC"]
  {Rd "A2" 90 100}
  [Ly "PINNUM"]
  [Ts 15][Tj "RC"]
  {Pn "1" 20 130}
  {Pn "12" 10 70}
  {Pn "4" 20 10}
  [Tj "LC"]
  {Pn "13" 100 -40}
  {Pn "2" 160 10}
  {Pn "3" 160 130}
 }
 {ATR
  {IN
   {Pl 410 290
  }
  {EX
   [Ly "ATTR"]
   [Tj "CC"]
   {At PCL (1,1,"d","d") 80 -10}
  }
 }
}
```

I

Example 2 - Shows the I subsection for a PCB database.

```
{I 74107.PRT XC00000
 {CN 1 XN00001 2 QA' 3 QA 4 XN00002  5 QC 6 QC'
  7 GND 8 XN00008 9 XN00025 10 CLR 11 XN00007
  12 XN00025 13 CLR 14 +5V}
 {ATR
  {IN
   {Pl 3750 1150}
  }
 }
}
```

CN

PATH:

DETAIL/SUBCOMP/I

FUNCTION:

The CN subsection defines the component-to-net connectivities. This subsection is essentially the network topology of the database.

FORMAT:

The CN subsection can be in either of two formats. The first format includes pin names as shown below.

```
{CN
  pinname net pinname net pinname net . . .
}
```

where:

> **pinname** is the name of a pin, assigned in the P-CAD system using the SYMB:ENTR/PIN command. PDIF-IN also accepts an asterisk (*) to indicate "don't care" for the pin name.

CN

net is the name of the net to which the pin is connected. In the P-CAD system, the net name is assigned using the DETL:NAME/NET command. PDIF-OUT assigns names to unnamed nets in the format XNnnnnn, where nnnnn is a five-digit integer and is assigned sequentially starting with 00000. If the pin is not connected to a net, a question mark (?) is used instead of the net name.

The sequence of the pins follows the pin order defined in the COMP_DEF/PIN_DEF section for the component type.

The second format does not include pin names. This format is useful in translating databases to or from a CAD system that does not use pin names. This format is shown below.

```
{CN
 net net net . . .
}
```

where:

net is the name of the net to which the pin is connected, as described above.

The sequence of the pins follows the pin order defined in the COMP_DEF/PIN_DEF section for the component type.

CN

PDIF-OUT can create the CN section in either format, depending on the setting of the **Include pin names in CN** configuration option. PDIF-IN recognizes the CN section in either format. It ignores the pin names, however, and uses the pin names listed in the COMP_DEF section for the component.

EXAMPLES:

Example 1 - Shows that the first pin of the component is connected to a signal called CHIPSEL', the second pin to ADDR15, the third to ADDR14, and the fourth pin is unconnected.

```
{CN
  Y CHIPSEL' A ADDR15 B ADDR14 C ?
}
```

Example 2 - Shows the same component as Example 1, with no pin names.

```
{CN
  CHIPSEL'  ADDR15  ADDR14  ?
 }
```

ASG

PATH:

DETAIL/SUBCOMP/I

FUNCTION:

This subsection specifies the packaging information of a schematic component, including its reference designator, designator location, and the package pin numbers corresponding to its logical pins. The ASG subsection exists in schematic databases only.

The order of Pn subsections is the same as the order of P subsections in the COMP_DEF/PIN_DEF section for the component type.

FORMAT:

```
{ASG
 {Rd "rd" x y}
 {Pn "n" x y}
 {Pn "n" x y}
 {Pn "n" x y}
 . . .
}
```

KEYWORDS:

Rd - Reference designator

Pn - Package pin numbers

ASG

EXAMPLES:

```
{ASG
 {Rd "A2" 90 100}
 [Ly "PINNUM"]
 {Pn "1" 20 130}
 {Pn "12" 10 70}
 {Pn "4" 20 10}
 {Pn "13" 100 -40}
 {Pn "2" 160 10}
 {Pn "3" 160 130}
}
```

ATR

PATH:

DETAIL/SUBCOMP/I

FUNCTION:

This subsection defines a component's instance-specific internal and external attributes, in the IN and EX subsections respectively. This subsection uses different keywords and gives different information than the ATR subsections in the SYMBOL, PAD_STACK, and COMP_DEF sections.

FORMAT:

```
{ATR
  {IN
    {Pl x y}
    {Sc sx sy}
    {Ro n}
    {Mr "m"}
    {Nl x y}
  }
  {EX
    {At key value x y}
    {At key value x y}
    . . .
  }
}
```

ATR

KEYWORDS:

Pl - Placement coordinates

Sc - X and Y scaling factors

Ro - Placement angle

Mr - Placement mirroring status

Nl - Name location of component instance

At - User-defined attributes

EXAMPLES:

```
{ATR
  {IN
    {Pl 410 290}
  }
  {EX
    [Ly "ATTR"]
    [Tj "CC"]
    {At PCL (1,1,"d","d") 80 -10}
  }
}
```

CHAPTER 5. PDIF KEYWORD REFERENCE

This chapter provides an alphabetical reference to the PDIF keywords and their usage. The keywords serve as database creation commands and define the various aspects that make up the database.

For each keyword, this chapter gives a detailed explanation with the following parts:

DATABASE: Gives the database types for which the keyword is used. The types are schematic (Sch) and printed circuit board (PCB).

SECTIONS: Shows the sections and subsections of the PDIF file in which the keyword is used.

PATHS: Shows the paths in the PDIF file to the subsections in which the keyword is used. (This part is not present for keywords used in the five main sections of the PDIF file.)

FUNCTION: Provides a brief description of the function of the keyword.

FORMAT: Shows the format used for the keyword.

REMARKS: Gives additional information about the usage of the keyword.

EXAMPLES: Provides specific examples of the keyword.

P-CAD CROSS-REFERENCE: Gives the P-CAD command or function that corresponds to the keyword. (For more information about the P-CAD commands and status line parameters, refer to Appendix D, "P-CAD to PDIF Cross-Reference," and to the "Commands" chapter of the *PC-CAPS User's Manual* or *PC-CARDS User's Manual.*).

The notation used in this chapter is shown in Table 5-1.

Table 5-1. Keyword Reference Notation

Symbol	Meaning
lower	Variable names representing keyword parameter values are shown in lowercase. (Actual values can be in uppercase or lowercase.)
" "	Quotation marks are required wherever they are shown.
. . .	The ellipsis is used to indicate that the keyword might contain additional data.
{ }	Curly brackets enclose all keywords except for those enclosed in square brackets, which are described below.
[ ]	Square brackets enclose several keywords that define display and graphics information. These keywords are used in the PDIF file in the DISPLAY section to set initial conditions and are used again wherever these conditions change in the design.
x y	The x and y parameters refer to horizontal and vertical grid coordinates.

A

Database:
Sch, PCB

Arc

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies an arc.

FORMAT:

{A cx cy r sa ea}

where:

> **cx** is the center x-coordinate (integer).
>
> **cy** is the center y-coordinate (integer).
>
> **r** is the radius (integer).
>
> **sa** is the start angle in degrees (integer).
>
> **ea** is the end angle in degrees (integer).

A

REMARKS:

An arc is identified using the center coordinates cx and cy, the radius r, and the starting and ending angles, sa and ea. The arc is drawn counterclockwise from sa to ea.

EXAMPLES:

{A 0 -20 41 284 76}

P-CAD CROSS-REFERENCE:

DRAW/ARC (SYMB or DETL)

At

Database:
Sch, PCB

User-defined attribute

SECTIONS:

EX

PATHS:

SYMBOL/ATR
DETAIL/SUBCOMP/COMP_DEF/ATR
DETAIL/SUBCOMP/I/ATR

FUNCTION:

Shows a user-specified component attribute and its location.

FORMAT:

{At key value x y}

where:

key is the attribute keyword.

value is the user-defined value.

x is the x-coordinate (integer).

y is the y-coordinate (integer).

At

REMARKS:

The attribute keyword is a text string that defines the value. It can be up to 8 characters and must start with a letter.

The value is a text string up to 39 characters. If it begins with a quotation mark or an open parenthesis, then it consists of everything up to and including the matching quotation mark or close parenthesis.

In the SYMBOL section, the attribute location is the absolute location coordinate.

In the DETAIL section, the location given is relative to the component origin and assumes no scaling, mirroring, or rotation. Absolute database location is calculated by using values of Pl, Sc, Mr and Ro for the component instance.

EXAMPLES:

```
{At PRT "7402.PRT" 50 -75}
{At PCL (1, 1, "d", "d") 185 -55}
{At Limit -12 50 -15}
{At HIVOLT VCC 100 50}
{At PI 3.1415926 1075 2080}
```

P-CAD CROSS-REFERENCE:

ATTR/ACOM (SYMB or DETL)

C

Database:
Sch, PCB

Circle

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies a circle.

FORMAT:

{C cx cy r}

where:

cx is the center x-coordinate (integer).

cy is the center y-coordinate (integer).

r is the radius (integer).

C

REMARKS:

A circle is identified by the center coordinates cx and cy, and the radius r.

EXAMPLES:

{C 50 -20 10}

P-CAD CROSS-REFERENCE:

DRAW/CIRC (SYMB or DETL)

DBgrid

Database:
Sch, PCB

Database grid

SECTIONS:

ENVIRONMENT

FUNCTION:

Specifies the database grid.

FORMAT:

{DBgrid n}

where:

> **n** is the number of units per DBU (integer). (Unit is defined by the DBunit keyword.)

REMARKS:

This keyword, together with the DBunit keyword, defines the relationship of the grid (working area) to a unit of measurement. In the P-CAD system, these values are not user-changeable. These keywords are provided for interfacing to other systems that use other values.

For any P-CAD design, the total working area is 60000 by 60000 database units (DBUs).

DBgrid

In the P-CAD system, the DBunit value is always "MIL". For a schematic database, DBgrid is always 10. These values specify that, if the schematic is printed or plotted at a scale of 1, then the P-CAD DBU is 10 mils.

For a PCB database, DBgrid is 1, so if the DBunit is "MIL", the DBU is 1 mil.

EXAMPLES:

{DBgrid 10}

P-CAD CROSS-REFERENCE:

Preset, but can be effectively changed when printing or plotting by changing the scale parameter in PC-PRINT, PC-PLOTS, or PC-PHOTO.

DBtime

Database:
Sch, PCB

Database time-stamp

SECTIONS:

ENVIRONMENT

FUNCTION:

Shows when the database was created.

FORMAT:

{DBtime "time_stamp"}

where:

time_stamp is in the format:

"mmm. dd, yyyy   hh:mm x.m."

where:

mmm. is the month (three characters and a period).

dd is the day (2-digit integer).

yyyy is the year (4-digit integer).

hh is the hour (2-digit integer).

mm is the minutes (2-digit integer).

x.m. is a.m. or p.m.

DBtime

REMARKS:

This section is created by PDIF-OUT. It is not required by PDIF-IN.

EXAMPLES:

{DBtime "Jan. 17, 1986   7:20 p.m.   "}

P-CAD CROSS-REFERENCE:

The information for this section is from the system clock (DOS).

DBtype

Database:
Sch, PCB

Database type

SECTIONS:

ENVIRONMENT

FUNCTION:

Specifies the database type.

FORMAT:

{DBtype "type"}

where:

"type" is "Schematic" or "PC-Board".

EXAMPLES:

{DBtype "Schematic"}

P-CAD CROSS-REFERENCE:

"Schematic" if created using PC-CAPS; "PC-Board" if created using PC-CARDS.

DBunit

Database:
Sch, PCB

Database unit

SECTIONS:

ENVIRONMENT

FUNCTION:

Specifies the unit of measurement in which the database is measured.

FORMAT:

{DBunit "unit"}

where:

"unit" is a unit of measurement.

REMARKS:

In the P-CAD system, the DBU is always measured in mils, so PDIF-OUT always produces a value of "MIL".

For more information, see the section on the DBgrid keyword.

DBunit

EXAMPLES:

{DBunit "MIL"}

P-CAD CROSS-REFERENCE:

Preset, not user-changeable.

DBvrev

Database:
Sch, PCB

Database revision

SECTIONS:

ENVIRONMENT

FUNCTION:

Shows the database version/revision number.

FORMAT:

{DBvrev n.nn}

where:

n.nn is the revision number.

REMARKS:

This keyword refers to the P-CAD database version number. The value for all P-CAD databases is 1.00.

EXAMPLES:

{DBvrev 1.00}

P-CAD CROSS-REFERENCE:

Not user-changeable.

Database:
PCB

Fl
Flash

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies the location and aperture number for a flash.

FORMAT:

{Fl x y a}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

a is the aperture number, 1 to 65.

Fl

REMARKS:

Fl specifies photoplotting requirements in a PCB database. Aperture values for the GTCO photoplotters are 1 through 65; the values for Gerber photoplotters are 1 through 24.

EXAMPLES:

{Fl 600 600 1}

P-CAD CROSS-REFERENCE:

DRAW/FLASH (SYMB or DETL)

Fr

Database:
Sch, PCB

Filled rectangle

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies a filled rectangle.

FORMAT:

{Fr x1 y1 x2 y2}

where:

> **x** is the x-coordinate (integer).
>
> **y** is the y-coordinate (integer).

REMARKS:

The rectangle is identified by two opposite corner coordinate specifications, x1 y1 and x2 y2.

Fr

EXAMPLES:

{Fr 800 25 1075 60}

P-CAD CROSS-REFERENCE:

DRAW/FREC (SYMB or DETL)

Gs

Database:
Sch, PCB

Grid spacing

SECTIONS:

VIEW

PATHS:

USER

FUNCTION:

Specifies the grid spacing.

FORMAT:

{Gs x y}

where:

x is the x-axis spacing (integer).

y is the y-axis spacing (integer).

REMARKS:

This keyword specifies the number of DBUs between visible grid points. This information is used by the P-CAD system only.

Gs

EXAMPLES:

{Gs 10 10}

P-CAD CROSS-REFERENCE:

Status line parameter X:Y

L

Database:
Sch, PCB

Line

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies a line.

FORMAT:

{L x1 y1 x2 y2 ... }

where:

> **x** is the x-coordinate at the end of a segment (integer).
>
> **y** is the y-coordinate at the end of a segment (integer).

L

REMARKS:

Each x y defines a line segment.

EXAMPLES:

{L 1000 0 1000 25}

P-CAD CROSS-REFERENCE:

DRAW/LINE (SYMB or DETL)

Lq

Database:
Sch, PCB

Logical equivalency pin code

PATHS:

SYMBOL/PIN_DEF
DETAIL/SUBCOMP/COMP_DEF/PIN_DEF

FUNCTION:

Specifies logical equivalence of pins.

FORMAT:

{Lq n}

where:

n is the code number (integer).

REMARKS:

The Lq code number shows which pins in the P list are logically equivalent. Pins that are logically equivalent (swappable) have the same code number; pins that are not logically equivalent (unswappable) have an Lq code of 0. PDIF-OUT uses the code numbers that were assigned by the database designer.

Lq

EXAMPLES:

{P A {Pt "INPUT"}{Lq 1}{Ploc -150 50}}
{P B {Pt "INPUT"}{Lq 1}{Ploc -150 10}}
{P C {Pt "OUTPUT"}{Lq 0}{Ploc 20 30}}

P-CAD CROSS-REFERENCE:

SCMD/SPAT (SYMB) in PC-CAPS or status line parameter EQUIV in ENTR/PIN (SYMB) in PC-CARDS

Ls

Database:
Sch, PCB

Line style

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the line style.

FORMAT:

[Ls "style"]

where:

> **"style"** is "SOLID", "DASHED", or "DOTTED".

REMARKS:

The DISPLAY shows the default line style for all lines. A new line style can be specified anywhere in the PDIF file. When a new line style is specified, the new value becomes the default. The legal Ls values are "SOLID", "DASHED", and "DOTTED".

Ls

EXAMPLES:

[Ls "SOLID"]

P-CAD CROSS-REFERENCE:

Status line parameters SOLID, DASHED, and DOTTED

Lv

Database:
Sch, PCB

Layer view status

SECTIONS:

VIEW

PATHS:

USER

FUNCTION:

Shows the active layer and the view status for each layer.

FORMAT:

{Lv a s1 s2 s3 ... sn}

where:

> **a** is the number of the active layer (defined by its position on the layer screen) (integer).
>
> **s1** to **sn** give the layer viewing status:
>
> 0 = OFF
> 1 = ON
> 2 = ABL

Lv

REMARKS:

Layers are listed in the order they appear on the P-CAD layer screen. The layer viewing status information is used by the P-CAD system only.

EXAMPLES:

{Lv 9 2 0 2 1 0 2 2 2 2 0 0 1 2}

P-CAD CROSS-REFERENCE:

VLYR (SYMB or DETL)

Ly

Database:
Sch, PCB

Layer

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the active layer.

FORMAT:

[Ly "layer"]

where:

"layer" is the layer name.

REMARKS:

The "layer" specified in DISPLAY is the active layer, on which all data is placed. Another layer can be specified anywhere in the PDIF file. When another layer is specified, it becomes the new active layer.

Ly

EXAMPLES:

```
{DISPLAY
 [Ly "WIRE"]
 ...
}

{SYMBOL
 {PIN_DEF
  [Ly "PINCON"]
  {P A   ... }
  {P B   ... }
 }
}
```

P-CAD CROSS-REFERENCE:

VLYR (SYMB or DETL), status line parameter LAYER

Lyrstr

Database:
Sch, PCB

Layer structure

SECTIONS:

ENVIRONMENT

FUNCTION:

Defines the layer structure.

FORMAT:

{Lyrstr "layer" n "layer" n ... }

where:

> **"layer"** is a layer name.
>
> **n** is the color code for the corresponding layer (integer).

REMARKS:

This specification includes all the layers that were defined when the database was created.

Lyrstr

EXAMPLES:

{Lyrstr "WIRE" 1 "BUS" 1 "GATE" 2 "IEEE" 2
"PINFUN" 3 "PINNUM" 1 "PINNAM" 6 "REFDES" 2
"ATTR" 6 "SDOT" 1 "DEVICE" 5}

P-CAD CROSS-REFERENCE:

VLYR (SYMB or DETL)

Mode

Database:
Sch, PCB

Database editing mode

SECTIONS:

VIEW

PATHS:

USER

FUNCTION:

Specifies the database editing mode.

FORMAT:

{Mode mode}

where:

> **mode** is the PC-CAPS or PC-CARDS editing mode, either SYMB or DETL.

Mode

REMARKS:

This information is used only by the P-CAD system. If this keyword is not present in a PDIF file, PDIF-IN sets the current mode to SYMB if the database has a symbol view with or without a detail view, or DETL mode if the database has a detail view but no symbol view.

EXAMPLES:

{Mode DETL}

P-CAD CROSS-REFERENCE:

Menu choice SYMB or DETL

Mr

Database:
Sch, PCB

Mirror

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/I/ATR

FUNCTION:

Specifies whether or not the mirroring function is in use.

FORMAT:

{Mr "m"}

where:

> **"m"** is "Y" if mirror-image is invoked, or "N" if orientation is normal.

REMARKS:

This attribute refers to the y-axis orientation of components in the database. It is produced by PDIF-OUT for component instance only if mirroring is specified. It is not required for PDIF-IN if the component is not mirrored.

Mr

EXAMPLES:

{Mr "Y"}

P-CAD CROSS-REFERENCE:

Status line parameter M (Mirror) except during ENTR/TEXT (see Tm keyword)

N1

Database:
Sch, PCB

Name location

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/I/ATR

FUNCTION:

Specifies the location of the component instance name.

FORMAT:

{N1 x y}

where:

> **x** is the x-coordinate (integer).
>
> **y** is the y-coordinate (integer).

Nl

REMARKS:

Nl specifies the location of the name of a specific instance of a component if the name is displayed. The location given is relative to the component origin and assumes no scaling, mirroring, or rotation. Actual database location is calculated using values of Pl, Sc, Mr, and Ro for the component. The name of the component instance is given in the I section.

EXAMPLES:

{Nl 100 55}

P-CAD CROSS-REFERENCE:

NAME/COMP (DETL)

Nlst

Database:
Sch, PCB

SECTIONS:

VIEW

PATHS:

USER

FUNCTION:

Specifies the netlist status.

FORMAT:

{Nlst st}

where:

> **st** is the netlist status, either OPEN if the netlist is free or CLOSE if the netlist is frozen. The default is OPEN.

REMARKS:

This information is used only the the P-CAD system. If this keyword is not present in a PDIF file, PDIF-IN sets the current status to OPEN.

Nlst

EXAMPLES:

{Nlst OPEN}

P-CAD CROSS-REFERENCE:

\SGAT

Nn

Database:
Sch, PCB

Net name

SECTIONS:

DG

PATHS:

DETAIL/NET_DEF/N

FUNCTION:

Specifies the location of the net name.

FORMAT:

{Nn x y x y ... }

where:

> x is the x-coordinate (integer).
>
> y is the y-coordinate (integer).

REMARKS:

Nn specifies one or more locations for the name of the net identified in the N subsection. A net name can be displayed at as many locations as you want.

Nn

EXAMPLES:

{Nn 60 6901}

P-CAD CROSS-REFERENCE:

NAME/NET (DETL), ENTR/WIRE (DETL)

Ns

Database:
Sch, PCB

Net scope

SECTIONS:

ATR

PATHS:

DETAIL/NET_DEF/N

FUNCTION:

Specifies the net scope.

FORMAT:

{Ns s}

where:

s is 0 for a local net or 1 for a global net.

REMARKS:

The Ns keyword is required only for global nets. If a net is local, PDIF-OUT does not generate the Ns keyword. If PDIF-IN does not find the Ns keyword for a net, it assumes the net is local.

Ns

EXAMPLES:

{Ns 1}

P-CAD CROSS-REFERENCE:

SCMD/SNAT (DETL)

Org

Database:
Sch, PCB

Origin

SECTIONS:

IN

PATHS:

SYMBOL/ATR

DETAIL/PAD_STACK/PAD_DEF/ATR

FUNCTION:

Specifies the location of the component origin.

FORMAT:

{Org x y}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

REMARKS:

The origin is the reference point for the placement of a symbol in the database. Placement of an instance is given by Pl.

Org

EXAMPLES:

{Org 170 -190}

P-CAD CROSS-REFERENCE:

ENTR/ORG (SYMB)

Pad

Database:
PCB

Pad or via

SECTIONS:

PAD_STACK

PATHS:

DETAIL

FUNCTION:

Defines a pad or via (feedthru).

FORMAT:

PDIF-OUT always uses the Pad keyword in the format shown below:

{Pad type "name" "name"}

where:

> **type** is the pin type: 0 to 24 (0 is for vias, all other numbers are for pads).
>
> The first "**name**" is the file used for pads for pins connected to nets.
>
> The second "**name**" is the file used for pads for unconnected pins.

Pad

PDIF-IN also recognizes an alternate format that can be used when the same padstack file is used for both connected and unconnected pads:

{Pad type "name"}

where:

"name" is the file used for pads for both connected and unconnected pins.

REMARKS:

The type is used in other subsections of the DETAIL section to specify the via or padstack file named with this keyword.

EXAMPLES:

{Pad 0 "V50R28.PS"}
{Pad 1 "C60S32.PS" "N60S32.PS"}

P-CAD CROSS-REFERENCE:

Specified in the special symbol file.

PDIFvrev

Database:
PCB, Sch

PDIF revision

SECTIONS:

ENVIRONMENT

FUNCTION:

Shows the PDIF version/revision number.

FORMAT:

{PDIFvrev n.nn}

where:

n.nn is the version number (three digits).

REMARKS:

PDIFvrev specifies the version number of the PDIF file format, which is changed occasionally. (This version number is separate from PDIF-IN and PDIF-OUT program version numbers.) PDIF-IN uses the PDIFvrev number to determine each PDIF file's format, and then processes the file accordingly.

PDIF-OUT automatically inserts the correct PDIFvrev number in each PDIF file it creates. If the user creates a PDIF file manually or through a custom program, the correct PDIFvrev number must be included.

PDIFvrev

EXAMPLES:

{PDIFvrev 1.31}

P-CAD CROSS-REFERENCE:

Version number of PDIF is used for the file.

Pl

Database:
PCB, Sch

Place

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/I/ATR

FUNCTION:

Specifies the location of a component.

FORMAT:

{Pl x y}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

REMARKS:

Pl specifies the location for the origin point of a component in the database. It is produced by PDIF-OUT for all databases with DETAIL data. It is not required for PDIF-IN. If PDIF-IN does not find a Pl value for a component, it places the component at 0 0.

Pl

EXAMPLES:

{Pl 180 690}

P-CAD CROSS-REFERENCE:

ENTR/COMP (DETL)

Ploc

Database:
PCB, Sch

Pin location

SECTIONS:

P

PATHS:

SYMBOL/PIN_DEF
DETAIL/SUBCOMP/COMP_DEF/PIN_DEF

FUNCTION:

Specifies the location of a pin.

FORMAT:

{Ploc x y}

where:

> **x** is the x-coordinate (integer).
>
> **y** is the y-coordinate (integer).

REMARKS:

For the DETAIL section, Ploc specifies the location of a given pin relative to the origin of a component (as sent by SYMB:ENTR/ORG).

Ploc

For the SYMBOL section, Ploc specifies the actual location of the pins in the database.

EXAMPLES:

```
{PIN_DEF
 [Ly "PINCON"]
 {P OUTY {Pt "OUTPUT"}{Ploc -10 60}}
 {P INA {Pt "INPUT"}{Ploc -140 60}}
}
```

P-CAD CROSS-REFERENCE:

ENTR/PIN (SYMB)

Pn

Database:
Sch

Pin number

SECTIONS:

ASG

PATHS:

DETAIL/SUBCOMP/I

FUNCTION:

Specifies a pin number and its location.

FORMAT:

{Pn "n" x y}

where:

"n" is the pin number.

x is the x-coordinate (integer).

y is the y-coordinate (integer).

Pn

REMARKS:

Pn specifies a pin number "n" and defines its preassigned location in a schematic database. The location given is relative to the component origin and assumes no scaling, mirroring, or rotation. Actual database location is calculated using values of Pl, Sc, Mr, and Ro for the component instance.

EXAMPLES:

{Pn "3" 175 40}

P-CAD CROSS-REFERENCE:

SCMD/PNUM (DETL)

Pnl

Database:
Sch

Pin number location

SECTIONS:

PKG

PATHS:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies the location of the package pin number.

FORMAT:

{Pnl x y}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

Pnl

REMARKS:

Pnl is a packaging specification for a schematic symbol. It gives the location where a package pin number is displayed when the symbol is used on a schematic. The PDIF file contains a Pnl specification for each pin of the symbol. The order of Pnl sections is the same as the order of P sections for the component type.

EXAMPLES:

{Pnl 65 -15}
{Pnl -65 5}
{Pnl -65 -35}

P-CAD CROSS-REFERENCE:

SCMD/PNLC (SYMB)

Program

Database:
Sch, PCB

Database creation program

SECTIONS:

ENVIRONMENT

FUNCTION:

Shows the PDIF-OUT version used to create the database.

FORMAT:

{Program "PDIF-OUT Version n.nn"}

where

n.nn is the PDIF-OUT version number.

EXAMPLES:

{Program "PDIF-OUT Version 1.31"}

P-CAD CROSS-REFERENCE:

Version of PDIF-OUT

Pt

Database:
Sch, PCB

Pin type definition

SECTIONS:

P

PATHS:

SYMBOL/PIN_DEF
DETAIL/SUBCOMP/COMP_DEF/PIN_DEF

FUNCTION:

Defines the pin type.

FORMAT:

For a schematic database, the format is:

{Pt "type"}

where:

"type" is "INPUT", "OUTPUT", or "I/O".

For a PCB database, the format is:

{Pt n}

where:

n is the pin type, 1 to 24 (integer).

Pt

REMARKS:

For a schematic database, the legal values are "INPUT", "OUTPUT", and "I/O". For a PCB database, Pt is an integer in the range 1 to 24.

EXAMPLES:

Example 1 - Defines the pins for a 2-input NAND gate.

```
{PIN_DEF
 [Ly "PINCON"]
 {P A {Pt "INPUT"}{Lq 1}{Ploc 0 30}}
 {P B {Pt "INPUT"}{Lq 1}{Ploc 0 10}}
 {P Y {Pt "OUTPUT"}{Lq 0}{Ploc 80 20}}
}
```

Pt

Example 2 - Defines the pins for a 7400 14-pin DIP.

```
{PIN_DEF
 [Ly "PIN"]
 {P 1 {Pt 1}{Lq 1}{Ploc -200 550}}
 {P 2 {Pt 2}{Lq 1}{Ploc -200 450}}
 {P 3 {Pt 2}{Lq 0}{Ploc -200 350}}
 {P 4 {Pt 2}{Lq 2}{Ploc -200 250}}
 {P 5 {Pt 2}{Lq 2}{Ploc -200 150}}
 {P 6 {Pt 2}{Lq 0}{Ploc -200 50}}
 {P 7 {Pt 3}{Lq 0}{Ploc -200 -50}}
 {P 8 {Pt 2}{Lq 0}{Ploc 100 -50}}
 {P 9 {Pt 2}{Lq 3}{Ploc 100 50}}
 {P 10 {Pt 2}{Lq 3}{Ploc 100 150}}
 {P 11 {Pt 2}{Lq 0}{Ploc 100 250}}
 {P 12 {Pt 2}{Lq 4}{Ploc 100 350}}
 {P 13 {Pt 2}{Lq 4}{Ploc 100 450}}
 {P 14 {Pt 4}{Lq 0}{Ploc 100 550}}
}
```

P-CAD CROSS-REFERENCE:

Status line parameters I/O, INPUT, OUTPUT, and TYPE

R

Database:
Sch, PCB

Rectangle

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies a rectangle.

FORMAT:

{R x1 y1 x2 y2}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

REMARKS:

A rectangle is identified by two opposite corner coordinate specifications, x1 y1 and x2 y2.

R

EXAMPLES:

{R 800 25 1075 60}

P-CAD CROSS-REFERENCE:

DRAW/RECT (SYMB or DETL)

Rd

Database:
Sch

Reference designator

SECTIONS:

ASG

PATHS:

DETAIL/SUBCOMP/I

FUNCTION:

Specifies the reference designator and its location.

FORMAT:

{Rd "rd" x y}

where:

> **"rd"** is the reference designator.
>
> **x** is the x-coordinate (integer).
>
> **y** is the y-coordinate (integer).

REMARKS:

The location given is relative to the component origin and assumes no scaling, mirroring, or rotation. Actual database location is calculated using values of Pl, Sc, Mr, and Ro for the component instance.

Rd

EXAMPLES:

{Rd "U2" 70 80}

P-CAD CROSS-REFERENCE:

SCMD/PNUM (DETL)

Rdl

Database:
Sch

Reference designator location

SECTIONS:

PKG

PATHS:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies the reference designator location.

FORMAT:

{Rdl x y}

where:

> x is the x-coordinate (integer).
>
> y is the y-coordinate (integer).

REMARKS:

In a schematic symbol database, Rdl specifies where the reference designator will be displayed when the symbol is used in a design database.

Rdl

EXAMPLES:

{Rdl -10 30}

P-CAD CROSS-REFERENCE:

SCMD/PNLC (SYMB)

Ro

Database:
Sch, PCB

Rotation

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/I/ATR

FUNCTION:

Specifies the rotation of a component.

FORMAT:

{Ro n}

where:

n is an integer, 0 to 3:

0 = no rotation
1 = 90 degrees
2 = 180 degrees
3 = 270 degrees

Ro

REMARKS:

Ro is produced by PDIF-OUT for a component instance only if rotation is specified. It is not required by PDIF-IN for nonrotated components. A value that is specified for a component is used for all subsequent components until a new value is specified.

Rotation is counterclockwise in the database.

EXAMPLES:

{Ro 2}

P-CAD CROSS-REFERENCE:

Status line parameter F (Orientation) except during ENTR/TEXT (see Tr keyword)

Sc

Database:
Sch, PCB

Scale

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/I/ATR

FUNCTION:

Shows the scale at which a component symbol is entered in a design database.

FORMAT:

{Sc sx sy}

where:

> **sx sy** is the percentage of the original component size.
>
> **sx** is the x scale (integer, 1-10000).
>
> **sy** is the y scale (integer, 1-10000).

Sc

REMARKS:

Sc specifies the percentage scaling applied to the component graphics. For example, if sx is 100 and sy is 100, then the scale is 100/100, or 100% of the original component size; if sx is 50 and sy is 50, then the scale is 50/50, or half the original component size.

This attribute is produced by PDIF-OUT for a component instance only if the scale is not 100 100. It is not required for PDIF-IN if the scale is 100 100.

EXAMPLES:

{Sc 50 50}

P-CAD CROSS-REFERENCE:

ENTR/COMP (x/y scales) (DETL)

Sd

Database:
Sch

Section definition

SECTIONS:

PKG

PATHS:

SYMBOL
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Defines the logical-to-physical mapping of a schematic symbol, by section.

FORMAT:

{Sd s n1 n2 n3 ... }

where:

> **s** is the package section.
>
> **n** is a package pin number (integer).

Sd

REMARKS:

The PDIF file contains an Sd specification for each package section, s, into which the symbol may be mapped. Each package pin number, n, corresponds to a pin of the schematic symbol.

Pins are listed in the order in which they are defined in the PIN_DEF/P section for the schematic symbol. For example, if a symbol has three pins, and the P section for the pin named INA is listed first, the P section for INB second, and the P section for OUTY third, then in each Sd specification, n1 corresponds to INA, n2 to INB, and n3 to OUTY.

EXAMPLES:

{Sd A 3 1 2}
{Sd B 6 4 5}
{Sd C 8 9 10}
{Sd D 11 12 13}

P-CAD CROSS-REFERENCE:

SCMD/PNLC (SYMB)

Sp

Database:
PCB

Symbol pin map

SECTIONS:

SPKG

PATHS:

SYMBOL/PIN_DEF
DETAIL/SUBCOMP/COMP_DEF/PIN_DEF

FUNCTION:

Shows the gate pin to packaged part mapping.

FORMAT:

{Sp spn n1 n2 n3 ... }

where:

spn is the symbol pin name.

n is a package pin number (integer).

Sp

REMARKS:

This keyword specifies the pins in the packaged parts that parallel the pins in the corresponding schematic symbol.

EXAMPLES:

{Sp INA 1 4 9 12}
{Sp INB 2 5 10 13}
{Sp OUTY 3 6 8 11}

P-CAD CROSS-REFERENCE:

SCMD/SPKG (SYMB)

T

Database:
Sch, PCB

Text

SECTIONS:

PIC, ANNOTATE

PATHS:

SYMBOL
DETAIL
DETAIL/PAD_STACK/PAD_DEF
DETAIL/SUBCOMP/COMP_DEF

FUNCTION:

Specifies text and its location.

FORMAT:

{T "string" x y}

where:

> **"string"** is a the text.
>
> **x** is the x-coordinate (integer).
>
> **y** is the y-coordinate (integer).

T

REMARKS:

T specifies a text string and its location in the schematic or PCB database.

EXAMPLES:

{T "P-CAD" 160 -190}

P-CAD CROSS-REFERENCE:

DRAW/TEXT (SYMB or DETL)

Tj

Database:
Sch, PCB

Text justification

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the text justification.

FORMAT:

[Tj "hv"]

where:

"hv" is a two-letter quoted string.

where:

h is horizontal justification: L, C, or R.

v is vertical justification: T, B, or C.

REMARKS:

Text justification is relative to the beginning cursor position. Horizontal values are L (left), R (right), and C (center); vertical values are T (top), B (bottom), and C (center). Thus, the legal Tj values are LT, LC, LB, CT, CC, CB, RT, RC, and RB.

Tj

The Tj value specified in DISPLAY is the default value for all text. A new value can be specified anywhere in the PDIF file. When a new value is specified, the new value becomes the default.

EXAMPLES:

[Tj "LL"]

P-CAD CROSS-REFERENCE:

Status line parameters B, C, L, R, and T

Tm

Database:
Sch, PCB

Text mirroring

SECTIONS:

DISPLAY, where required

FUNCTION:

Shows whether text mirroring is in use.

FORMAT:

[Tm "m"]

where:

> **"m"** is "Y" if text mirroring is invoked or "N" if text orientation is normal.

REMARKS:

The default Tm value specified in DISPLAY is normally "N" for all text. If "Y" is subsequently specified in the database, then "Y" becomes the default.

EXAMPLES:

[Tm "N"]

Tm

P-CAD CROSS-REFERENCE:

Status line parameter M (Mirror) for ENTR/TEXT only (see Mr keyword).

Tr

Database:
Sch, PCB

Text rotation

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the text rotation.

FORMAT:

[Tr n]

where:

n is an integer, 0 to 3:

0 = no rotation
1 = 90 degrees
2 = 127 degrees
3 = 270 degrees

REMARKS:

The default value shown in DISPLAY is normally 0 for all text. A different value can be specified anywhere in the PDIF file. When a different value is specified, then that value becomes the default.

Rotation is counterclockwise in the database.

Tr

EXAMPLES:

[Tr 0]

P-CAD CROSS-REFERENCE:

Status line parameter F (Orientation) for ENTR/TEXT only (see Ro keyword).

Ts

Database:
Sch, PCB

Text size

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the text size.

FORMAT:

[Ts n]

where:

n is a positive integer, 2 to 5000.

REMARKS:

Ts specifies the height of the text in database units. The Ts value shown in DISPLAY is the default value for all text. A new value can be specified anywhere in the PDIF file. When a new value is specified, the new value becomes the default.

Ts

EXAMPLES:

[Ts 15]

P-CAD CROSS-REFERENCE:

Status line parameter SIZ

Ty

Database:
Sch, PCB

Type

SECTIONS:

IN

PATHS:

SYMBOL/PIC/ATR
DETAIL/SUBCOMP/COMP_DEF/ATR

FUNCTION:

Shows the Component Type ID.

FORMAT:

{Ty n}

where:

n is an integer, -32000 to 32000.

REMARKS:

The component type ID has several uses in the P-CAD system. It is required for the P-CAD PC-LOGS simulator. The PC-CAPS and PC-CARDS programs set a default value, which the designer can change if needed. Refer to the appropriate user's manual for more information.

Ty

EXAMPLES:

{Ty 256}

P-CAD CROSS-REFERENCE:

SCMD/SCAT (SYMB)

V

Database:
Sch, PCB

Via

SECTIONS:

DG

PATHS:

DETAIL/NET_DEF/N

FUNCTION:

Specifies the location of a solder dot in a schematic database or a via (feedthru) in a PCB database.

FORMAT:

{V x y type}

where:

x is the x-coordinate (integer).

y is the y-coordinate (integer).

type is the type of solder dot or via (integer).

V

EXAMPLES:

{V 50 210 0}

P-CAD CROSS-REFERENCE:

ENTR/WIRE (DETL) (when a T junction is made in a schematic database or when an active layer is changed in a PCB database), EDIT/AVIA (DETL)

Vr

Database:
Sch, PCB

Component version number

SECTIONS:

IN

PATHS:

DETAIL/SUBCOMP/COMP_DEF/ATR

FUNCTION:

Gives the component version number.

FORMAT:

{Vr n}

where:

n is the version number (integer).

REMARKS:

This keyword specifies the version or revision number of the component at the time it was incorporated into the current design. Vr 0, the only value currently used, indicates the original component.

Vr

EXAMPLES:

{Vr 0}

P-CAD CROSS-REFERENCE:

Not user-changeable

Vw

Database:
Sch, PCB

Viewing windows

SECTIONS:

VIEW

PATHS:

USER

FUNCTION:

Specifies the viewing windows.

FORMAT:

{Vw ix iy is n llx lly urx ury n llx lly urx ury ... }

where:

> **ix** is the initial x center of screen.
>
> **iy** is the initial y center of screen.
>
> **is** is the initial scale (zoom in/out).
>
> **n llx lly urx ury** is each stored window (optional).
>
> **n** is the user-assigned stored window number, 1-10.
>
> **llx** is the x coordinate at the lower left corner of the window.

Vw

lly is the y coordinate at the lower left corner of the window.

urx is the x coordinate at the upper right corner of window.

ury is the y coordinate at the upper right corner of window.

REMARKS:

This information is used only by the P-CAD system.

EXAMPLES:

{Vw 0 0 1 1 -15 -50 515 310}

P-CAD CROSS-REFERENCE:

STO (SYMB or DETL)

W

Database:
Sch, PCB

Wire

SECTIONS:

DG

PATHS:

DETAIL/NET_DEF/N

FUNCTION:

Specifies a wire.

FORMAT:

{W x1 y1 x2 y2 ... }

where:

> **x** is the x-coordinate at the end of a segment (integer).
>
> **y** is the y-coordinate at the end of a segment (integer).

W

REMARKS:

Each x y defines a segment of wire.

EXAMPLES:

{W 150 650 150 150 180 150}

P-CAD CROSS-REFERENCE:

ENTR/WIRE (DETL)

Wd

Database:
Sch, PCB

Width

SECTIONS:

DISPLAY, where required

FUNCTION:

Specifies the line width.

FORMAT:

[Wd n]

where:

n is an integer from 0 to 250.

REMARKS:

The width specified in DISPLAY is the default value for all lines. A new value can be specified anywhere in the PDIF file. When a new value is specified, then that value becomes the default. If n is 0, then width is the design program default value.

Wd

EXAMPLES:

[Wd 0]

P-CAD CROSS-REFERENCE:

Status line parameter W

APPENDIX A. PDIF-IN ERROR MESSAGES

This appendix shows the error messages generated by PDIF-IN. These error messages are displayed on the screen and stored in the log file.

There are four types of PDIF-IN errors: command line errors, configuration file errors, program initiation errors, and file translation errors. This appendix describes the types of error messages, lists all the error messages, and gives, for each message, the cause of the error and the procedure for correcting it.

COMMAND LINE ERRORS

These errors appear on the screen in command line mode after you enter a command line.

Message: Filename required with -e option

Cause: You specified the -e option on the command line but did not specify an exception filename.

Action: Enter a valid command line.

Message: Illegal option : <xx>

Cause: The command line you entered contains an invalid option letter.

Action: Reenter the command line.

CONFIGURATION FILE ERRORS

These errors involve the PDIFIN.CFG configuration file. They appear on the screen as soon as you start program operation.

Message: **Configuration file error; using default values**

Cause: The configuration file could be damaged.

Action: Delete the configuration file, then use PDIF-IN to create a new configuration file and rerun PDIF-IN.

Message: **Failed to save configuration file**

Cause: Your disk could be full.

Action: Check your disk space, delete files if necessary to make space, and rerun PDIF-IN.

Message: **No data exists in the configuration file**

Cause: The configuration file contains invalid data.

Action: Delete the configuration file, then use PDIF-IN to create a new configuration file and rerun PDIF-IN.

PROGRAM INITIATION ERRORS

These errors occur before PDIF-IN can process the input file.

Message: Cannot open file : <filename>

Cause: Either you specified an incorrect filename or there is a system error.

Action: Be sure you specify the correct filename. If the problem persists, call P-CAD.

Message: Cannot reopen file : <name>

Cause: The system could not close the file, so it was unable to reopen it.

Action: Call P-CAD.

Message: **Device Driver System File not found**
Device Driver System File not complete
Device Driver compatibility error
Device Driver failure

Cause: The P-CAD software was not properly installed, or an installation file has been corrupted or deleted.

Action: Reinstall the software as specified in the *Installation Guide.*

Message: Exception file <name> not found

Cause: You specified an incorrect filename.

Action: Specify a correct filename.

Message: File not found : <filename>

Cause: The named input file does not exist in any directory specified in the **Directory Path** configuration option.

Action: Enter the correct filename or make sure the file is in the correct directory.

Message: Input and output filenames must be different

Cause: You specified identical names for the input and output files.

Action: Use a different name for your output file.

Message: Not a PDIF file : <filename>

Cause: The specified input file is not a PDIF file; or the component section is missing.

Action: Be sure the file you specify is a PDIF file; or check the component section.

FILE TRANSLATION ERRORS

The remaining errors occur during file processing.

Message: **Buffer full : Cannot process more data**

Cause: This is a system error.

Action: Call P-CAD.

Message: **Cannot allocate space for pnconrec.**

Cause: This is an irrecoverable internal error.

Action: Call P-CAD.

Message: **Cannot close file : <filename>**

Cause: PDIF-IN was unable to close the input PDIF file.

Action: Call P-CAD.

Message: **Cannot create new net.**

Cause: The database is too large to fit into available memory.

Action: Create a smaller database.

Message: **Cannot get token**

Cause: Irrecoverable internal error.

Action: Call P-CAD.

Message: **Cannot load ENVIRONMENT database**

Cause: PDIF-IN was unable to load the file specified in the ENVIRONMENT section of the PDIF file.

Action: Call P-CAD.

Message: **Cannot open exception file**

Cause: Either you specified an incorrect filename or there is a system error.

Action: Be sure you specify the correct filename. If the problem persists, call P-CAD.

Message: **Cannot reopen file: <name>**
Fatal error tying pin to net.

Cause: These messages indicate a software failure.

Action: Call P-CAD.

Message: Component directory INTCMPDR exists; files may be destroyed

Cause: The current directory has a subdirectory named INTCMPDR. If you specified "Yes" for the **Create Components** configuration option, PDIF-IN will overwrite any component files in this directory that have the same names as components being created. If you specified "No" for this option, PDIF-IN will erase all the files it created from the input PDIF file and erase this directory if it is empty.

Action: No corrective action is necessary. After PDIF-IN creates component files, be sure to move them from INTCMPDR to another directory for storage.

Message: Component not found : <name>

Cause: PDIF-IN was unable to find the specified component either in the PDIF file or in a P-CAD component file. The component is not included in the output database.

Action: Be sure that the filename is correct in the PDIF file and that either a COMP_DEF section is present in the PDIF file for the component or the corresponding P-CAD file is present in a directory specified in the **Directory Path** configuration option. Rerun PDIF-IN to produce a corrected output file.

Message: Disk full; COMP_DEF symbol not saved

Cause: Your disk does not have enough space to store a component file created from a COMP_DEF section of the PDIF file.

Action: To create the component, delete files from your disk to make space, then rerun PDIF-IN.

Message: Disk full. Database not saved

Cause: Your disk is full.

Action Delete files to make space, then rerun PDIF-IN.

Message: Duplicate COMP_DEF section : <name>

Cause: More than one COMP_DEF section exists in the PDIF file for the splecified component. (This error might be generated by an old version of PDIF-OUT.) The last COMP_DEF section is used to define the component in the output file.

Action: Check to see if the component definition used is satisfactory. If not, delete the extra COMP_DEF section from the PDIF file, then rerun PDIF-IN to produce a corrected output file.

Message: ENVIRONMENT database does not exist

Cause: The file specified in the ENVIRONMENT section of the PDIF file is not present in the current or specified directory.

Action: Make sure the named file exists and is present in a directory specified in the path set during PDIF-IN configuration.

Message: ENVIRONMENT/DBtype statement invalid or not found

Cause: The ENVIRONMENT section of the PDIF file does not contain the DBtype keyword. PDIF-IN cannot process the file.

Action: Add the DBtype keyword to the PDIF file, then rerun PDIF-IN.

Message: ENVIRONMENT/Lyrstr statement invalid or not found

Cause: The ENVIRONMENT section of the PDIF file does not contain the Lyrstr keyword. PDIF-IN cannot process the file.

Action: Add the DBtype keyword to the PDIF file, then rerun PDIF-IN.

Message: ENVIRONMENT section not found

Cause: The PDIF file does not contain an ENVIRONMENT section. PDIF-IN cannot process the file.

Action: Add the ENVIRONMENT section to the PDIF file, then rerun PDIF-IN.

Message: Error building symbol from COMP_DEF

Cause: PDIF-IN was unable to create a component file for the named component from the corresponding COMP_DEF section of the PDIF file due to lack of space on your hard disk.

Action: Delete files as necessary to make space. To create the component file, rerun PDIF-IN.

Message: Error in finding database type

Cause: PDIF-IN cannot determine the database type for the output file.

Action: Call P-CAD.

Message: Error in saving database

Cause: PDIF-IN was unable to save the output database file.

Action: Call P-CAD.

Message: Exception file <name> not found

Cause: The file specified as the exception file does not exist.

Action: Specify the name of an existing file.

Message: External symbol not found : <name>

Cause: PDIF-IN was unable to find the specified component file. The component is not included in the output database.

Action: Be sure that the filename is correct in the PDIF file and that the corresponding file is present in a directory specified in the **Directory Path** configuration option. Alternately, specify internal component import for the component. Rerun PDIF-IN to produce a corrected output file.

Message: Fatal error in tying pin to net

Cause: The pin number or owning net was invalid.

Action: Call P-CAD.

Message: **File error : <name>**

Cause: PDIF-IN was unable to create a component file for the named component from the corresponding COMP_DEF section of the PDIF file due to a problem on your hard disk.

Action: Use the DOS CHKDSK command to check your disk status. Delete files if necessary. To create the component file, rerun PDIF-IN.

Message: **Insufficient memory**

Cause: Your system does not have enough memory. PDIF-IN requires 640K, or 512K if configured for the TI.

Action: Be sure you have enough memory, then rerun PDIF-IN.

Message: **Invalid component name : <name>**

Cause: The specified name found in an I section of the PDIF file does not conform to DOS filename conventions or includes a directory path. The component is not included in the output database.

Action: Correct the component name in the PDIF file, then rerun PDIF-IN to produce a corrected output file.

Message: **Left curly bracket expected**
Warning: Missing '[' or ']'
Warning: Missing '{' or '}'

Cause: These messages indicate that the input PDIF ASCII file has incorrect syntax.

Action: Correct the file. Make sure that for each '{' there is a matching '}' and that for each '[' there is a matching ']'.

Message: **No data in exception file**

Cause: The exception file you specified is empty.

Action: If you want to specify exceptions for component import, add them to the exception file and rerun PDIF-IN.

Message: **Pad stack <name> does not exist**

Cause: The file specified as the pad stack file does not exist.

Action: Specify the name of an existing file.

Message: **{PDIFvrev . . } section not found**

Cause: The ENVIRONMENT section of the PDIF file does not contain the PDIFvrev keyword. PDIF-IN assumes the PDIF file format is the 1.30 version.

Action: No corrective action is necessary unless the PDIF file format is another version.

Message: **Unexpected end of input in <filename>**

Cause: The input PDIF file ends prematurely. This message might be caused by an extra right curly bracket (}).

Action: Check the PDIF file to make sure the curly brackets match properly, then rerun PDIF-IN.

Message: **Using COMP_DEF for <name>**

Cause: You are using external component import but specified the named component in the exception file.

Action: No corrective action is necessary.

Message: **Using external symbol for <name>**

Cause: Either you specified internal component import for the named component but no COMP_DEF section was present in the PDIF file, or you are using internal component import but specified the named component in the exception file.

Action: If the component is correct in the output file, no action is necessary.

Message: Warning: Missing '[' or ']'

Cause: This message indicates that the input PDIF ASCII file has incorrect syntax.

Action: Correct the file. Make sure that for each '{' there is a matching '}' and that for each '[' there is a matching '[.'

Message: Warning: Missing '{' or '}'

Cause: This message indicates that the input PDIF ASCII file has incorrect syntax.

Action: Correct the file. Make sure that for each '{' there is a matching '}' and that for each '[' there is a matching '[.'

APPENDIX B. PDIF-OUT ERROR MESSAGES

This appendix shows the error messages generated by PDIF-OUT. If any of these error messages are displayed on the screen or shown in the PDIF output file, you must correct the error before you can produce a complete PDIF file.

There are four types of PDIF-OUT errors: command line errors, configuration file errors, program initiation errors, and file translation errors. This appendix describes the types of error messages, lists all the error messages, and gives, for each message, the cause of the error and the procedure for correcting it.

COMMAND LINE ERRORS

These errors appear on the screen in command line mode after you enter a command line.

Message: Illegal option : <xx>

Cause: The command line you entered contains an invalid option letter.

Action: Reenter the command line.

CONFIGURATION FILE ERRORS

This group of errors involves the PDIFOUT.CFG configuration file. They appear on the screen as soon as you start program operation.

Message: **Failed to read config. file. Default values used**

Cause: The program was unable to read the PDIFOUT.CFG file.

Action: Run PDIF-OUT in configuration mode to create a new configuration file.

Message: **Failed to save configuration file**

Cause: There is not enough disk space to store the file.

Action: Delete files to make space available.

PROGRAM INITIATION ERRORS

These errors occur before PDIF-OUT can produce the output PDIF file. They appear on the screen during program operation.

Message: **Failed to get database type**

Cause: The program was unable to determine the database type of the input file. The database could be damaged.

Action: Be sure you specified the correct file. Check the file if necessary.

Message: Failed to load the entire <infname> database

Cause: The program encountered an error while loading the input database. There could be insufficient memory or the database could be corrupt.

Action: Be sure you have 640K and rerun the program. If the error persists, check the database; correct or regenerate it if necessary.

Message: Failed to open <outfname>

Cause: The program was unable to create the PDIF output file.

Action: Be sure that the CONFIG.SYS file in the root directory has a FILES value of 15 or more.

Message: Failed to produce PDIF file

Cause: The program was unable to produce the PDIF file. This message accompanies other error messages.

Action: Correct the error noted in the accompanying error message.

Message: <infname> does not exist

Cause: The program was unable to find the input file you specified.

Action: Enter the correct filename. Use a drive or directory designator if the file is not in the current directory.

Message: <infname> format is incompatible

Cause: The input file you specified is not a P-CAD design database.

Action: Enter the correct filename.

Message: Input and output filenames must be different

Cause: You have specified an output filename that is identical to the input filename.

Action: Specify a different filename or extension for the output file.

Message: Insufficient disk space, execution aborted

Cause: There is not enough space on your disk for either the configuration file or the output PDIF file.

Action: Delete files as needed to make space.

Message: Insufficient memory

Cause: The program requires 640K of memory.

Action: Make sure 640K is available.

FILE TRANSLATION ERRORS

The remaining errors occur while PDIF-OUT is processing the input database. They appear in the PDIF output file immediately after the section in which the error was found. When one of these errors occurs, the program stops producing the PDIF file. You must correct the error before you can produce a complete PDIF file.

Message: Database corrupted; Failed to get horizontal justification.

Cause: The input database is damaged.

Action: Correct the database; regenerate it, if necessary.

Message: Database corrupted; Failed to get line style.

Cause: The input database is damaged.

Action: Correct the database; regenerate it, if necessary.

Message: Database corrupted; Failed to get vertical justification.

Cause: The input database is damaged.

Action: Correct the database; regenerate it, if necessary.

Message: **ENVIRONMENT data is incomplete.**
USER data is incomplete.
DISPLAY data is incomplete.
SYMBOL data is incomplete.
DETAIL data is incomplete.

Cause: These messages accompany other error messages that occur in the specified section.

Action: Correct the error noted in the accompanying error message.

Message: **Failed to get attributes for this component.**

Cause: The program was unable to access the attribute information in the database or (if the error is in the SUBCOMP section) in the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get component filename.**

Cause: The program was unable to access the filename of the database or (if the error is in the SUBCOMP section) of the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: Failed to get component instance data block.

Cause: The program was unable to access the instance data for this instance of the component. The database could be damaged.

Action: Correct the database; regenerate it, if necessary.

Message: Failed to get component instance name.

Cause: The program was unable to get the component instance name and was unable to assign an instance name. The input database could be damaged.

Action: Correct the database file; regenerate the database, if necessary.

Message: Failed to get filename.

Cause: The program was unable to find the filename of the database or (if the error is in the SUBCOMP section) in the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get filename of the symbol.**

Cause: The program was unable to get the filename of the symbol. The symbol file is damaged.

Action: Check the symbol file. Correct it if necessary, save it with the desired filename, then rerun PDIF-OUT. If the problem persists, recreate the symbol file.

Message: **Failed to get node tied to pin <pinname>.**

Cause: The program was unable to access data for a node connected to the specified pin of this instance of the component. The database could be damaged.

Action: Correct the database; regenerate it, if necessary.

Message: **Failed to get packaging information for this component.**

Cause: The program was unable to access the packaging information in the database or (if the error is in the SUBCOMP section) in the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: Failed to get parameter block for ASG data.

Cause: The program was unable to access the data for an attribute in the ASG section. The database could be damaged.

Action: Correct the database; regenerate it, if necessary.

Message: Failed to get parameter block for PKG data.

Cause: The program was unable to access the data for an attribute in the PKG section. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: Failed to get picture data for this component.

Cause: The program was unable to access the graphic data for the database or (if the error is in the SUBCOMP section) for the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get picture information for this component.**

Cause: The program was unable to access the graphic information in the database or (if the error is in the SUBCOMP section) in the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get picture of the net.**

Cause: The program was unable to access the wires in the database. The database could be damaged.

Action: Correct the database file; regenerate it, if necessary.

Message: **Failed to get pin information for this component.**

Cause: The program was unable to access the pin information in the database or (if the error is in the SUBCOMP section) in the symbol or part database. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get pin of this component.**

Cause: The program was unable to access data for a pin of this instance of the component. The database could be damaged.

Action: Correct the database; regenerate it, if necessary.

Message: **Failed to get symbol pin.**

Cause: The program was unable to access a pin for the symbol. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failed to get the padstack name.**

Cause: The program was unable to access the filename for the padstack file. The padstack file could be damaged.

Action: Correct the padstack file; regenerate the database, if necessary.

Message: **Failed to get the parameter block.**

Cause: The program was unable to access the data for an attribute in the PDIF file. The input database or symbol/part file could be damaged.

Action: Correct the database or symbol/part file; regenerate the database, if necessary.

Message: **Failure in locating a position in the database.**

Cause: The program was unable to get to a necessary portion of the input database. The database could be damaged.

Action: Check the database; correct it or regenerate it, if necessary.

Message: **Failure in opening the database file.**

Cause: The program was unable to open the input database file. The database could be damaged.

Action: Be sure that the CONFIG.SYS file in the root directory has a FILES value of 15 or more. If so, check the database; correct it or regenerate it, if necessary.

Message: **Failure in reading the database file.**

Cause: The program was unable to read the input database file. The database could be damaged.

Action: Check the database; correct it or regenerate it, if necessary.

Message: Illegal pin connection at <compname> pin <pinname>

Cause: At the named pin in the P-CAD database, you entered a wire from another net, then reversed the direction of the wire, which deleted it. In the PDIF database, the pin is shown as unconnected. However, in the P-CAD system, it is considered connected.

Action: Use PC-CAPS to enter a wire connected to the pin. Then delete the wire. An entire net will disappear. Reenter the net correctly.

Message: Pin number is invalid.

Cause: The program accessed an invalid number for a pin of this instance of the component. The database could be damaged.

Action: Correct the database; regenerate it, if necessary.

Message: **USER VIEW data is incomplete**
SYMBOL PIN_DEF data is incomplete
SYMBOL PKG data is incomplete
SYMBOL SPKG data is incomplete
SYMBOL PIC data is incomplete
SYMBOL ATR data is incomplete
DETAIL ANNOTATE data is incomplete
DETAIL NET_DEF data is incomplete
DETAIL PAD_STACK data is incomplete
DETAIL SUBCOMP data is incomplete
N DG data is incomplete
N ATR data is incomplete
I CN data is incomplete
I ASG data is incomplete
SUBCOMP ATR data is incomplete

Cause: These messages accompany other error messages that occur in the specified subsection.

Action: Correct the error noted in the accompanying error message.

APPENDIX C. NETLIST IMPORT

PDIF enables you to import a netlist containing component and connectivity information into the P-CAD system from another CAD system. You do this by using a text editor to create a PDIF file with the necessary information and inputting the PDIF file into PDIF-IN.

You can import a netlist for any type of P-CAD database file. The most common use of netlist import is to translate a PCB netlist, but the process is equally effective for schematic netlists.

This chapter explains how to construct a PDIF file for this application.

FILE FORMAT

The PDIF file used to import netlists is an abbreviated version of the standard PDIF file that is described in Chapter 4, "The PDIF File." The sections and subsections used are shown below.

```
COMPONENT
.  ENVIRONMENT
.  DETAIL
.  .  SUBCOMP
.  .  .  I
.  .  .  .  CN
.  .  .  .  ATR
.  .  .  .  .  IN
```

The file starts with a header that contains the COMPONENT, ENVIRONMENT, DETAIL, and SUBCOMP identifiers. The rest of the file consists of a series of I and CN records, one set for each component. The ATR section is optional for each component. The file ends with three closing brackets.

The format of the file is shown below.

```
{COMPONENT dbf {ENVIRONMENT file} {DETAIL
{SUBCOMP

  {I filename inv_name
    {CN pin_name net pin_name net . . . }
    {ATR {IN {Pl x y} {Ro n} }}
  }
  {I filename inv_name
    {CN pin_name net pin_name net . . . }
    {ATR {IN . . . }}
  }
  . . .
}}}
```

All the section names are described in Chapter 4, "The PDIF File," and the keywords are described in Chapter 5, "PDIF Keyword Reference." They are reviewed briefly below.

COMPONENT specifies the P-CAD database that is to be created. The dbf value is the filename.

ENVIRONMENT specifies the starting environment for the database. The file value is the name of the layer structure or board outline file that contains the starting environment.

DETAIL and SUBCOMP are the headings for the section of the PDIF file that holds the component data.

The I section shows an instance of a component. The filename value is the name of the P-CAD part or symbol file that identifies the component. The inv_name value is the reference designator in PCB design or the component name in schematic design. Both values must be names recognizable by the P-CAD system.

The CN subsection of the I section shows component-to-net connectivities for the instance of the component. The pin_name values identify each pin, and the net value is the name of the net to which the pin is connected. Pin names are optional and can be omitted. The order in which the pins are entered is significant; see the "CN" section in Chapter 4.

The ATR/IN subsection of the I section can be used to specify component location. You can use the Pl keyword to specify x and y coordinates and the Ro keyword to specify rotation. If location is not specified, all components in the P-CAD database will be located in a "heap" and you can use the P-CAD editing commands to place them wherever you want.

Since there are no internal COMP_DEF sections, PDIF-IN must be run with the -ni option.

EXAMPLE

The following is an example of a PDIF file describing a PCB database.

```
% Project A2345    8/1/86

{COMPONENT MUX.PCB {ENVIRONMENT
LAYS.PCB} {DETAIL {SUBCOMP

 {I 74LS00.PRT U1
  {CN  1 A' 2 B' 3 D00' 4 A' 5 B 6 D01' 7 GND 8 D10'
      9 A 10 B' 11 D11' 12 A 13 B 14 VCC
  }
 }
 {I 74LSO4.PRT U2
  {CN 1 D00' 2 D00 3 D 01' 4 D01 5 D10' 6 D10 7 GND
      8 D11 9 D11' 10 A' 11 A 12 B' 13 B 14 VCC
  }
 }
 {I DECAP.PRT C1 {CN VCC GND} }

 {I DECAP.PRT C2
  {CN * VCC * GND}
  {ATR {IN {Pl 500 500} {Ro 1} }}
 }
 {I CON20.PRT J1
  {CN
      GND GND VCC VCC A D00 B
      D01 ? D10 ? D11
  }
  {ATR {IN {Pl 500 0} {Ro 0} }}
 }
}}}
```

APPENDIX D. P-CAD TO PDIF CROSS-REFERENCE

Table D-1 cross-references P-CAD commands with PDIF data. It lists each P-CAD command or status line parameter, the P-CAD program(s) and mode(s) in which it is used, and the PDIF section(s) and keywords that contain the corresponding data. Commands are listed first in alphabetical order, then status line parameters are listed in alphabetical order.

For further information about the P-CAD commands and status line parameters and for information about the commands with which each status line parameter is used, refer to the "Commands" chapter of the *PC-CAPS User's Manual* or *PC-CARDS User's Manual.*

Table D-1. P-CAD to PDIF Cross-Reference

P-CAD Command	Program	Mode	PDIF Section	PDIF Keyword
ATTR/ACOM	PC-CAPS, PC-CARDS	SYMB	SYMBOL/ATR/EX, DETAIL/SUBCOMP/COMP_DEF/ATR/EX	At
		DETL	DETAIL/SUBCOMP/I/ATR/EX	At
DETL	PC-CAPS, PC-CARDS		USER/VIEW	Mode
DRAW/ARC	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	A
		DETL	DETAIL/ANNOTATE	A
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	A
DRAW/CIRC	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	C
		DETL	DETAIL/ANNOTATE	C
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	C
DRAW/FLSH	PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/PAD_STACK/PAD_DEF/PIC	Fl
		DETL	DETAIL/ANNOTATE	Fl
DRAW/FREC	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	Fr
		DETL	DETAIL/ANNOTATE	Fr
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	Fr
DRAW/LINE	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	L
		DETL	DETAIL/ANNOTATE	L
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	L

Table D-1 Continued

P-CAD Command	Program	Mode	PDIF Section	PDIF Keyword
DRAW/RECT	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	R
		DETL	DETAIL/ANNOTATE	R
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	R
DRAW/TEXT	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIC, DETAIL/SUBCOMP/COMP_DEF/PIC	T
		DETL	DETAIL/ANNOTATE	T
	PC-CARDS	SYMB	DETAIL/PAD_STACK/PAD_DEF/PIC	T
ENTR/COMP	PC-CAPS, PC-CARDS	DETL	DETAIL/SUBCOMP	-
			DETAIL/SUBCOMP/I/ATR/IN	Pl
	PC-CAPS	DETL	DETAIL/SUBCOMP/I ATR/IN	Sc
ENTR/ORG	PC-CAPS, PC-CARDS	SYMB	SYMBOL/ATR/IN, DETAIL/SUBCOMP/COMP_DEF/ATR/IN	Org
ENTR/PIN	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Ploc
			DETAIL/SUBCOMP/I/CN	-
ENTR/RATN	PC-CARDS	DETL	DETAIL/SUBCOMP/I/CN	-
ENTR/WIRE	PC-CAPS, PC-CARDS	DETL	DETAIL/NET_DEF/N	W,V
NAME/COMP	PC-CAPS, PC-CARDS	DETL	DETAIL/SUBCOMP/I	
			DETL/SUBCOMP/I/ATR/IN	Nl
NAME/NET	PC-CAPS, PC-CARDS	DETL	DETAIL/NET_DEF/N	-
			DETAIL/NET_DEF/DG	Nn
SCMD/PNLC	PC-CAPS	SYMB	SYMBOL/PKG, DETAIL/SUBCOMP/COMP_DEF/PKG	Rdl, Pnl,Sd

Table D-1 Continued

P-CAD Command	Program	Mode	PDIF Section	PDIF Keyword
SCMD/PNUM	PC-CAPS	DETL	DETAIL/SUBCOMP/I/ASG	Rd,Pn
SCMD/SCAT	PC-CAPS, PC-CARDS	SYMB	SYMBOL/ATR/IN, DETAIL/SUBCOMP/COMP_DEF/ATR/IN	Ty
SCMD/SNAT	PC-CAPS, PC-CARDS	DETL	DETAIL/NET_DEF/N/ATR	Ns
SCMD/SPAT	PC-CAPS, PC-CARDS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Pt,Lq
SCMD/SPKG	PC-CARDS	SYMB	SYMBOL/SPKG DETAIL/SUBCOMP/SPKG	Sp
/SGAT	PC-CAPS, PC-CARDS	SYMB, DETL	USER/VIEW	Nlst
STO	PC-CAPS, PC-CARDS	SYMB, DETL	USER/VIEW	Vw
SYMB	PC-CAPS, PC-CARDS		USER/VIEW	Mode
VLYR	PC-CAPS, PC-CARDS	SYMB, DETL	ENVIRONMENT	Lyrstr
			USER/VIEW	Lv
			DISPLAY, where required	Ly
B (Bottom Justified)	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Tj
C (Center Justified)	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Tj
DASHED	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Ls
DOTTED	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Ls
EQUIV	PC-CAPS, PC-CARDS	SYMB, DETL	SYMBOL/PIN_DEF/P	Lq

Table D-1 Continued

P-CAD Command	Program	Mode	PDIF Section	PDIF Keyword
F (Orien-tation)	PC-CAPS, PC-CARDS	SYMB, DETL	DETAIL/SUBCOMP/I/ATR/IN	Ro
			DISPLAY, where required	Tr
I/O	PC-CAPS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Pt
INPUT	PC-CAPS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Pt
L (Left Justified)	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Tj
LAYER	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Ly
M (Mirror)	PC-CAPS, PC-CARDS	SYMB, DETL	DETAIL/SUBCOMP/I/ATR	Mr
			DISPLAY, where required	Tm
OUTPUT	PC-CAPS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Pt
R (Right Justified)	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Tj
SIZ	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Ts
SOLID	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Ls
T (Top Justified)	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Tj
TYPE	PC-CARDS	SYMB	SYMBOL/PIN_DEF/P, DETAIL/SUBCOMP/COMP_DEF/PIN_DEF/P	Pt
W	PC-CAPS, PC-CARDS	SYMB, DETL	DISPLAY, where required	Wd
X:Y	PC-CAPS, PC-CARDS	SYMB, DETL	USER/VIEW	Gs

APPENDIX E. SAMPLE PDIF FILES

P-CAD supplies several sample data files with the PDIF programs. This appendix tells how to use these sample files to become familiar with the PDIF programs and file structure. It also includes the complete contents of all the sample PDIF files.

You can use the sample files in the following ways:

- As an aid to understanding PDIF file structure. You can examine the sample PDIF files in this appendix.

- As test input. You can use the sample files as input to PDIF-IN and PDIF-OUT to experiment with running the programs.

- As a reference for specific questions. You can examine the sample P-CAD database files (using PC-CAPS and PC-CARDS) and note how certain features are treated in the PDIF file.

The sample files include all the types of P-CAD design databases and the PDIF files produced by inputting several of the sample P-CAD design databases into PDIF-OUT. Each sample PDIF file has the same filename as the corresponding P-CAD database with a .PDF extension.

Table E-1 lists the types of P-CAD databases and the names of the corresponding sample files. The sections following Table E-1 contain all the sample PDIF files, with plots of the corresponding P-CAD files.

Table E-1. Sample Files

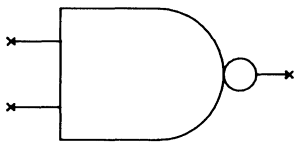
Database Type	Sample P-CAD File	Sample PDIF File
Schematic symbol	74LS00.SYM 74LS04.SYM	74LS00.PDF 74LS04.PDF
Flat schematic	EX1.SCH	EX1.PDF
Hierarchical subcircuit	EX2.SYM	EX2.PDF
Hierarchical schematic	EX3.SCH	EX3.PDF
PCB part	7400A.PRT 7404T.PRT	7400A.PDF *
Padstack	C60R32.PS V50R28.PS N60S32.PS C60S32.PS N60R32.PS	C60R32.PDF * * * *
PCB design	EX4.PCB	EX4.PDF
Special symbol file	EX4.SSF	**

* PDIF files corresponding to these files are not supplied. These files are provided to enable you to use PDIF-IN with the EX4.PDF example.

** The special symbol file is an ASCII file and cannot be converted into a PDIF file. It is provided as a reference for the EX4 example.

SCHEMATIC SYMBOL: 74LS00

Symbol File: 74LS00.SYM



PDIF File: 74LS00.PDF

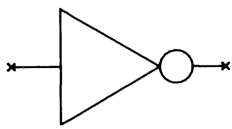
```
%**********************************************************************
%                                                                     *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.            *
%                                                                     *
%     Program  :   PDIF-OUT VERSION 1.31                              *
%     Date     :   Oct 14 1986                                        *
%     Time     :   03:01:01 PM                                        *
%     File In  :   74LS00.SYM                                         *
%     File Out :   74LS00.PDF                                         *
%     Format   :   P-CAD DATABASE INTERCHANGE FORMAT                  *
%                                                                     *
%**********************************************************************

{COMPONENT 74LS00.SYM


{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "Schematic"}
{DBvrev 1.00}
{DBtime "Oct. 26, 1984          8:35 a.m.          "}
{DBunit "MIL"}
{DBgrid 10}
{Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2 "PINFUN" 3 "PINNUM" 1 "PINNAM" 6
"PINCON" 2 "REFDES" 2 "ATTR" 6 "SDOT" 1 "DEVICE" 4}
}

{USER
{VIEW
{Mode SYMB}
{Nlst OPEN}
{Vw -70 30 1}
{Lv 12 0 0 1 0 0 2 0 0 2 1 0 2}
{Gs 10 10}
}
}

{DISPLAY
[Ly "DEVICE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CC"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
[Ly "PINCON"]
{P INA {Pt "INPUT"}{Lq 1}{Ploc -150 50}}
{P INB {Pt "INPUT"}{Lq 1}{Ploc -150 10}}
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 20 30}}
}
{PKG
[Ly "REFDES"]
```

```
[Ts 25][Tj "CC"][Tr 0][Tm "N"]
{Rdl -70 30}

[Ly "PINNUM"]
[Ts 15][Tj "RC"]
{Pnl -130 60}
{Pnl -130 20}
[Tj "LC"]
{Pnl 10 40}

{Sd A 1 2 3}
{Sd B 4 5 6}
{Sd C 10 9 8}
{Sd D 12 13 11}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "LC"][Tr 0][Tm "N"]
{C -10 30 10}
{L 0 30 20 30}
{A -60 30 40 270 90}
{L -60 70 -120 70 -120 -10 -60 -10}
{L -120 50 -150 50}
{L -120 10 -150 10}
[Ly "IEEE"]
{L -20 40 0 30}
[Tj "CB"]
{T "&" -70 50}
{L -20 30 20 30}
{L -150 50 -120 50}
{L -150 10 -120 10}
{R -120 -10 -20 70}
[Ly "DEVICE"]
[Tj "CC"]
{T "74LS00" -80 -20}
}

{ATR
{IN
{Org -150 10}
{Ty 4}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") -70 -30}
}
}
}
```

```
{DETAIL
{ANNOTATE
}

{NET_DEF
{N INA
}
{N INB
}
{N OUTY
}
}

{SUBCOMP
}
}
}
```

SCHEMATIC SYMBOL: 74LS04

Symbol File: 74LS04.SYM



PDIF File: 74LS04.PDF

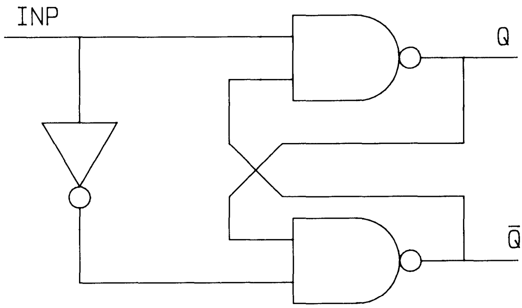
```
%***********************************************************************
%                                                                     *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.            *
%                                                                     *
%     Program  :   PDIF-OUT VERSION 1.31                              *
%     Date     :   Oct 14 1986                                        *
%     Time     :   03:00:49 PM                                        *
%     File In  :   74LS04.SYM                                         *
%     File Out :   74LS04.PDF                                         *
%     Format   :   P-CAD DATABASE INTERCHANGE FORMAT                  *
%                                                                     *
%***********************************************************************

{COMPONENT 74LS04.SYM


{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "Schematic"}
{DBvrev 1.00}
{DBtime "Oct. 26, 1984        8:47 a.m.         "}
{DBunit "MIL"}
{DBgrid 10}
{Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2 "PINFUN" 3 "PINNUM" 1 "PINNAM" 6
"PINCON" 2 "REFDES" 2 "ATTR" 6 "SDOT" 1 "DEVICE" 4}
}

{USER
{VIEW
{Mode SYMB}
{Nlst OPEN}
{Vw -60 30 1}
{Lv 12 0 0 0 2 1 2 1 1 2 1 0 2}
{Gs 10 10}
}
}

{DISPLAY
[Ly "DEVICE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CC"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
[Ly "PINCON"]
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc -10 60}}
{P INA {Pt "INPUT"}{Lq 0}{Ploc -140 60}}
}
```

```
{PKG
[Ly "REFDES"]
[Ts 25][Tj "LC"][Tr 0][Tm "N"]
{Rdl -80 90}
[Ly "PINNUM"]
[Ts 15]
{Pnl -20 70}
[Tj "RC"]
{Pnl -120 70}

{Sd A 2 1}
{Sd B 4 3}
{Sd C 6 5}
{Sd D 8 9}
{Sd E 10 11}
{Sd F 12 13}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "RC"][Tr 0][Tm "N"]
{C -40 60 10}
{L -110 60 -140 60}
{L -110 95 -110 25 -50 60 -110 95}
{L -30 60 -10 60}
[Ly "IEEE"]
[Tj "CC"]
{T "1" -80 70}
{L -30 60 -50 70}
{L -50 60 -10 60}
{R -110 25 -50 95}
{L -110 60 -140 60}
[Ly "DEVICE"]
{T "74LS04" -80 0}
}

{ATR
{IN
{Org -140 60}
{Ty 1}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"D","D") -80 10}
}
}
}
```

```
{DETAIL
{ANNOTATE
}
{NET_DEF
{N OUTY
}
{N INA
}
}
{SUBCOMP
}
}
}
```

FLAT SCHEMATIC: EX1

Schematic File: EX1.SCH



PDIF File: EX1.PDF

```
%*********************************************************************
%                                                                   *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.          *
%                                                                   *
%     Program   :   PDIF-OUT VERSION 1.31                           *
%     Date      :   Oct 14 1986                                     *
%     Time      :   03:01:31 PM                                     *
%     File In   :   EX1.SCH                                         *
%     File Out  :   EX1.PDF                                         *
%     Format    :   P-CAD DATABASE INTERCHANGE FORMAT               *
%                                                                   *
%*********************************************************************

{COMPONENT EX1.SCH

{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "Schematic"}
{DBvrev 1.00}
{DBtime "Jul. 24, 1986        11:13 a.m.         "}
{DBunit "MIL"}
{DBgrid 10}
{Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2 "PINFUN" 3 "PINNUM" 1 "PINNAM" 6
"PINCON" 4 "REFDES" 2 "ATTR" 6 "SDOT" 1 "DEVICE" 5 "OUTLIN" 5
"ATTR2" 6 "NOTES" 6 "NETNAM" 4 "CMPNAM" 5 "BORDER" 5}
}

{USER
{VIEW
{Mode DETL}
{Nlst OPEN}
{Vw -120 70 1}
{Lv 18 2 2 1 0 0 0 1 1 0 0 0 1 1 0 0 2 0 2}
{Gs 10 10}
}
}

{DISPLAY
[Ly "BORDER"]
[Ls "SOLID"][Wd 0]
[Ts 25][Tj "LB"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
}
```

```
{PIC
}
{ATR
{IN
{Org -32767 -32767}
{Ty 255}
}
}
}

{DETAIL
{ANNOTATE
[Ly "BORDER"]
[Ls "SOLID"][Wd 0]
[Ts 25][Tj "LB"][Tr 0][Tm "N"]
{T "EX1.SCH" -150 -150}
{R -380 -110 160 260}
}

{NET_DEF
{N XN00000
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 25][Tj "LB"][Tr 0][Tm "N"]
{W -240 0 -240 -40 -140 -40}
}
}
{N INP
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 25][Tj "LB"][Tr 0][Tm "N"]
{V -240 170 15}
{W -240 130 -240 170}
{W -140 170 -240 170 -350 170}
[Ly "NETNAM"]
[Ts 15]
{Nn -340 190}
}
}
{N Q
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "LB"][Tr 0][Tm "N"]
{W -140 0 -140 40 -100 70 30 70 30 150 100 150}
[Ly "NETNAM"]
```

```
{Nn 90 170}
}
}
{N Q'
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "LB"][Tr 0][Tm "N"]
{W -140 130 -140 70 -100 40 30 40 30 -20 100 -20}
[Ly "NETNAM"]
{Nn 90 0}
}
}
}

{SUBCOMP
{COMP_DEF 74LS00.SYM
{PIN_DEF
[Ly "PINCON"]
{P INA {Pt "INPUT"}{Lq 1}{Ploc 0 40}}
{P INB {Pt "INPUT"}{Lq 1}{Ploc 0 0}}
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 170 20}}
}
{PKG
[Ly "REFDES"]
[Ts 25][Tj "CC"][Tr 0][Tm "N"]
{Rdl 80 20}

[Ly "PINNUM"]
[Ts 15][Tj "RC"]
{Pnl 20 50}
{Pnl 20 10}
[Tj "LC"]
{Pnl 160 30}

{Sd A 1 2 3}
{Sd B 4 5 6}
{Sd C 10 9 8}
{Sd D 12 13 11}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "LC"][Tr 0][Tm "N"]
{C 140 20 10}
{L 150 20 170 20}
{A 90 20 40 270 90}
{L 90 60 30 60 30 -20 90 -20}
{L 30 40 0 40}
```

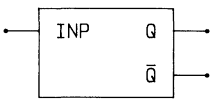
```
{L 30 0 0 0}
[Ly "IEEE"]
{L 130 30 150 20}
[Tj "CB"]
{T "&" 80 40}
{L 130 20 170 20}
{L 0 40 30 40}
{L 0 0 30 0}
{R 30 -20 130 60}
[Ly "DEVICE"]
[Tj "CC"]
{T "74LS00" 70 -30}
}
{ATR
{IN
{Ty 4}
{Vr 0}
}
}
}
{I 74LS00.SYM XC00000
{CN INP Q' Q}
{ATR
{IN
{Pl -140 130}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") 80 -40}
}
}
}
{I 74LS00.SYM XC00001
{CN Q XN00000 Q'}
{ATR
{IN
{Pl -140 -40}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") 80 -40}
}
}
}
{COMP_DEF 74LS04.SYM
{PIN_DEF
[Ly "PINCON"]
```

```
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 130 0}}
{P INA {Pt "INPUT"}{Lq 0}{Ploc 0 0}}
}
{PKG
[Ly "REFDES"]
[Ts 25][Tj "LC"][Tr 0][Tm "N"]
{Rdl 60 30}

[Ly "PINNUM"]
[Ts 15]
{Pnl 120 10}
[Tj "RC"]
{Pnl 20 10}

{Sd A 2 1}
{Sd B 4 3}
{Sd C 6 5}
{Sd D 8 9}
{Sd E 10 11}
{Sd F 12 13}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "RC"][Tr 0][Tm "N"]
{C 100 0 10}
{L 30 0 0 0}
{L 30 35 30 -35 90 0 30 35}
{L 110 0 130 0}
[Ly "IEEE"]
[Tj "CC"]
{T "1" 60 10}
{L 110 0 90 10}
{L 90 0 130 0}
{R 30 -35 90 35}
{L 30 0 0 0}
[Ly "DEVICE"]
{T "74LS04" 60 -60}
}
{ATR
{IN
{Ty 1}
{Vr 0}
}
}
}
{I 74LS04.SYM XC00002
{CN XN00000 INP}
```

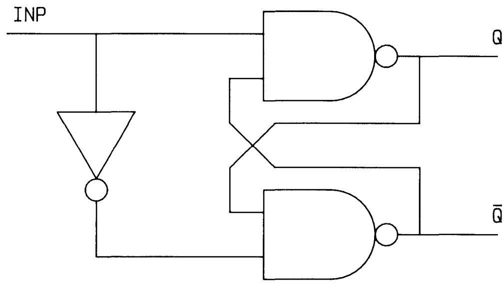
```
{ATR
{IN
{Pl -240 130}
{Ro 3}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"D","D") 60 -50}
}
}
}
}
}
}
```

HIERARCHICAL SUBCIRCUIT: EX2

Symbol File: EX2.SYM



(symbol)



(detail)

PDIF File: EX2.PDF

```
%***********************************************************************
%                                                                      *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.             *
%                                                                      *
%     Program   :   PDIF-OUT VERSION 1.31                              *
%     Date      :   Oct 14 1986                                        *
%     Time      :   03:01:37 PM                                        *
%     File In   :   EX2.SYM                                            *
%     File Out  :   EX2.PDF                                            *
%     Format    :   P-CAD DATABASE INTERCHANGE FORMAT                  *
%                                                                      *
%***********************************************************************

{COMPONENT EX2.SYM

{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "Schematic"}
{DBvrev 1.00}
{DBtime "Nov. 22, 1985          1:55 p.m.          "}
{DBunit "MIL"}
{DBgrid 10}
{Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2 "PINFUN" 3 "PINNUM" 1 "PINNAM" 6
"PINCON" 3 "REFDES" 2 "ATTR" 6 "SDOT" 1 "DEVICE" 5 "NETNAM" 4
"ARTWRK" 6}
}

{USER
{VIEW
{Mode DETL}
{Nlst OPEN}
{Vw 25 -25 2}
{Lv 13 2 0 2 0 0 2 2 2 2 0 0 2 2 2}
{Gs 10 10}
}
}

{DISPLAY
[Ly "NETNAM"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
[Ly "PINCON"]
{P INP {Pt "INPUT"}{Lq 0}{Ploc -140 -80}}
{P Q {Pt "OUTPUT"}{Lq 0}{Ploc 40 -80}}
{P Q' {Pt "OUTPUT"}{Lq 0}{Ploc 40 -120}}
}
```

```
{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{R -110 -140 10 -60}
{L 10 -80 40 -80}
{L 10 -120 40 -120}
{L -140 -80 -110 -80}
[Ly "DEVICE"]
[Tj "CC"]
{T "EX2" -50 -100}
[Ly "PINNAM"]
{T "INP" -80 -80}
{T "Q" -10 -80}
{T "Q'" -10 -120}
}

{ATR
{IN
{Org -140 -80}
{Ty 255}
}
}
}

{DETAIL
{ANNOTATE
[Ly "ARTWRK"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CC"][Tr 0][Tm "N"]
{R -340 -200 240 280}
{T "EX2.SYM" -50 -230}
}

{NET_DEF
{N X
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CC"][Tr 0][Tm "N"]
{W -200 50 -200 20 -80 20}
}
}
{N Q'
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CC"][Tr 0][Tm "N"]
{W -80 180 -80 140 -40 100 90 100 90 40 160 40}
[Ly "NETNAM"]
[Tj "CB"]
{Nn 160 50}
}
```

```
{ATR
{IN
{Ns 1}
}
}
}
{N INP
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W -280 220 -80 220}
{W -200 220 -200 180}
[Ly "NETNAM"]
{Nn -260 230}
}
{ATR
{IN
{Ns 1}
}
}
}
{N Q
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W -80 60 -80 100 -40 140 90 140 90 200 160 200}
[Ly "NETNAM"]
{Nn 160 210}
}
{ATR
{IN
{Ns 1}
}
}
}
}

{SUBCOMP
{COMP_DEF 74LS00.SYM
{PIN_DEF
[Ly "PINCON"]
{P INA {Pt "INPUT"}{Lq 1}{Ploc 0 40}}
{P INB {Pt "INPUT"}{Lq 1}{Ploc 0 0}}
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 170 20}}
}
{PKG
[Ly "REFDES"]
[Ts 25][Tj "CC"][Tr 0][Tm "N"]
{Rdl 80 20}

[Ly "PINNUM"]
[Ts 15][Tj "RC"]
{Pnl 20 50}
```

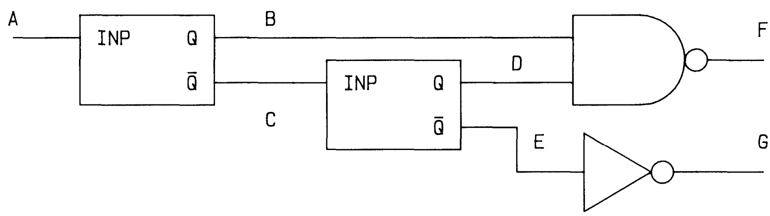
```
{Pnl 20 10}
[Tj "LC"]
{Pnl 160 30}

{Sd A 1 2 3}
{Sd B 4 5 6}
{Sd C 10 9 8}
{Sd D 12 13 11}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "LC"][Tr 0][Tm "N"]
{C 140 20 10}
{L 150 20 170 20}
{A 90 20 40 270 90}
{L 90 60 30 60 30 -20 90 -20}
{L 30 40 0 40}
{L 30 0 0 0}
[Ly "IEEE"]
{L 130 30 150 20}
[Tj "CB"]
{T "&" 80 40}
{L 130 20 170 20}
{L 0 40 30 40}
{L 0 0 30 0}
{R 30 -20 130 60}
[Ly "DEVICE"]
[Tj "CC"]
{T "74LS00" 70 -30}
}
{ATR
{IN
{Ty 4}
{Vr 0}
}
}
}
{I 74LS00.SYM XC00003
{CN INP Q' Q}
{ATR
{IN
{Pl -80 180}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") 80 -40}
}
}
}
{I 74LS00.SYM XC00004
{CN Q X Q'}
```

```
{ATR
{IN
{Pl -80 20}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") 80 -40}
}
}
}
{COMP_DEF 74LS04.SYM
{PIN_DEF
[Ly "PINCON"]
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 130 0}}
{P INA {Pt "INPUT"}{Lq 0}{Ploc 0 0}}
}
{PKG
[Ly "REFDES"]
[Ts 25][Tj "LC"][Tr 0][Tm "N"]
{Rdl 60 30}

[Ly "PINNUM"]
[Ts 15]
{Pnl 120 10}
[Tj "RC"]
{Pnl 20 10}

{Sd A 2 1}
{Sd B 4 3}
{Sd C 6 5}
{Sd D 8 9}
{Sd E 10 11}
{Sd F 12 13}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "RC"][Tr 0][Tm "N"]
{C 100 0 10}
{L 30 0 0 0}
{L 30 35 30 -35 90 0 30 35}
{L 110 0 130 0}
[Ly "IEEE"]
[Tj "CC"]
{T "1" 60 10}
{L 110 0 90 10}
{L 90 0 130 0}
{R 30 -35 90 35}
{L 30 0 0 0}
[Ly "DEVICE"]
{T "74LS04" 60 -60}
}
```

```
{ATR
{IN
{Ty 1}
{Vr 0}
}
}
}
{I 74LS04.SYM XC00005
{CN X INP}
{ATR
{IN
{Pl -200 180}
{Ro 3}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"D","D") 60 -50}
}
}
}
}
}
}
```

HIERARCHICAL SCHEMATIC: EX3

Schematic File: EX3.SCH



PDIF File: EX3.PDF

```
%********************************************************************
%                                                                  *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.         *
%                                                                  *
%     Program  :   PDIF-OUT VERSION 1.31                           *
%     Date     :   Oct 14 1986                                     *
%     Time     :   03:01:42 PM                                     *
%     File In  :   EX3.SCH                                         *
%     File Out :   EX3.PDF                                         *
%     Format   :   P-CAD DATABASE INTERCHANGE FORMAT               *
%                                                                  *
%********************************************************************

{COMPONENT EX3.SCH


{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "Schematic"}
{DBvrev 1.00}
{DBtime "Dec. 5, 1985        12:27 p.m.           "}
{DBunit "MIL"}
{DBgrid 10}
{Lyrstr "WIRES" 1 "BUS" 1 "GATE" 2 "IEEE" 2 "PINFUN" 3 "PINNUM" 1 "PINNAM" 6
"PINCON" 3 "REFDES" 2 "ATTR" 6 "SDOT" 1 "DEVICE" 5 "NETNAM" 4
"ARTWRK" 6}
}

{USER
{VIEW
{Mode DETL}
{Nlst OPEN}
{Vw 25 -25 2}
{Lv 13 2 0 2 0 0 2 2 2 2 0 0 0 2 2}
{Gs 10 10}
}
}

{DISPLAY
[Ly "NETNAM"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
}

{PIC
}
```

```
{ATR
{IN
{Org -32767 -32767}
{Ty 255}
}
}
}

{DETAIL
{ANNOTATE
[Ly "ARTWRK"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{R -360 -140 420 120}
{T "EX3.SCH" 30 -170}
}

{NET_DEF
{N C
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W -100 20 -60 20}
[Ly "NETNAM"]
{Nn -80 -20}
}
}
{N B
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W -100 60 160 60}
[Ly "NETNAM"]
{Nn -80 70}
}
}
{N D
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W 120 20 160 20}
[Ly "NETNAM"]
{Nn 140 30}
```

```
}
}
{N E
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W 120 -20 140 -20 140 -60 170 -60}
[Ly "NETNAM"]
{Nn 160 -40}
}
}
{N A
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W -280 60 -310 60}
[Ly "NETNAM"]
{Nn -310 70}
}
{ATR
{IN
{Ns 1}
}
}
}
{N G
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W 300 -60 360 -60}
[Ly "NETNAM"]
{Nn 360 -40}
}
{ATR
{IN
{Ns 1}
}
}
}
{N F
{DG
[Ly "WIRES"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "CB"][Tr 0][Tm "N"]
{W 330 40 360 40}
```

```
[Ly "NETNAM"]
{Nn 360 60}
}
{ATR
{IN
{Ns 1}
}
}
}
}

{SUBCOMP
{COMP_DEF 74LS00.SYM
{PIN_DEF
 [Ly "PINCON"]
 {P INA {Pt "INPUT"}{Lq 1}{Ploc 0 40}}
 {P INB {Pt "INPUT"}{Lq 1}{Ploc 0 0}}
 {P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 170 20}}
 }
 {PKG
 [Ly "REFDES"]
 [Ts 25][Tj "CC"][Tr 0][Tm "N"]
 {Rdl 80 20}

 [Ly "PINNUM"]
 [Ts 15][Tj "RC"]
 {Pnl 20 50}
 {Pnl 20 10}
 [Tj "LC"]
 {Pnl 160 30}

 {Sd A 1 2 3}
 {Sd B 4 5 6}
 {Sd C 10 9 8}
 {Sd D 12 13 11}
 }

 {PIC
 [Ly "GATE"]
 [Ls "SOLID"][Wd 0]
 [Ts 15][Tj "LC"][Tr 0][Tm "N"]
 {C 140 20 10}
 {L 150 20 170 20}
 {A 90 20 40 270 90}
 {L 90 60 30 60 30 -20 90 -20}
 {L 30 40 0 40}
 {L 30 0 0 0}
 [Ly "IEEE"]
 {L 130 30 150 20}
 [Tj "CB"]
```

```
{T "&" 80 40}
{L 130 20 170 20}
{L 0 40 30 40}
{L 0 0 30 0}
{R 30 -20 130 60}
[Ly "DEVICE"]
[Tj "CC"]
{T "74LS00" 70 -30}
}
{ATR
{IN
{Ty 4}
{Vr 0}
}
}
}
{I 74LS00.SYM XC00006
{CN B D F}
{ASG A
[Ly "REFDES"]
[Ts 25][Tj "CC"][Tr 0][Tm "N"]
{Rd "U1" 80 20}
[Ly "PINNUM"]
[Ts 15][Tj "RC"]
{Pn "1" 20 50}
{Pn "2" 20 10}
[Tj "LC"]
{Pn "3" 160 30}
}
{ATR
{IN
{Pl 160 20}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"d","d") 80 -40}
}
}
}
{COMP_DEF 74LS04.SYM
{PIN_DEF
[Ly "PINCON"]
{P OUTY {Pt "OUTPUT"}{Lq 0}{Ploc 130 0}}
{P INA {Pt "INPUT"}{Lq 0}{Ploc 0 0}}
}
{PKG
[Ly "REFDES"]
[Ts 25][Tj "LC"][Tr 0][Tm "N"]
{Rdl 60 30}
```

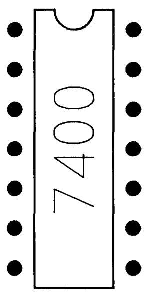
```
[Ly "PINNUM"]
[Ts 15]
{Pnl 120 10}
[Tj "RC"]
{Pnl 20 10}

{Sd A 2 1}
{Sd B 4 3}
{Sd C 6 5}
{Sd D 8 9}
{Sd E 10 11}
{Sd F 12 13}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 15][Tj "RC"][Tr 0][Tm "N"]
{C 100 0 10}
{L 30 0 0 0}
{L 30 35 30 -35 90 0 30 35}
{L 110 0 130 0}
[Ly "IEEE"]
[Tj "CC"]
{T "1" 60 10}
{L 110 0 90 10}
{L 90 0 130 0}
{R 30 -35 90 35}
{L 30 0 0 0}
[Ly "DEVICE"]
{T "74LS04" 60 -60}
}
{ATR
{IN
{Ty 1}
{Vr 0}
}
}
}
{I 74LS04.SYM XC00007
{CN G E}
{ASG A
[Ly "REFDES"]
[Ts 25][Tj "LC"][Tr 0][Tm "N"]
{Rd "U2" 60 30}
[Ly "PINNUM"]
[Ts 15]
{Pn "2" 120 10}
[Tj "RC"]
```

```
{Pn "1" 20 10}
}
{ATR
{IN
{Pl 170 -60}
}
{EX
[Ly "ATTR"]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{At PCL (1,1,"D","D") 60 -50}
}
}
}
{COMP_DEF EX2.SYM
{PIN_DEF
[Ly "PINCON"]
{P INP {Pt "INPUT"}{Lq 0}{Ploc 0 0}}
{P Q {Pt "OUTPUT"}{Lq 0}{Ploc 180 0}}
{P Q' {Pt "OUTPUT"}{Lq 0}{Ploc 180 -40}}
}

{PIC
[Ly "GATE"]
[Ls "SOLID"][Wd 0]
[Ts 10][Tj "CC"][Tr 0][Tm "N"]
{R 30 -60 150 20}
{L 150 0 180 0}
{L 150 -40 180 -40}
{L 0 0 30 0}
[Ly "DEVICE"]
[Ts 15]
{T "EX2" 90 -20}
[Ly "PINNAM"]
{T "INP" 60 0}
{T "Q" 130 0}
{T "Q'" 130 -40}
}
{ATR
{IN
{Ty 255}
{Vr 0}
}
}
}
{I EX2.SYM XC00008
{CN C D E}
```

```
{ATR
{IN
{Pl -60 20}
}
}
}
{I EX2.SYM XC00009
{CN A B C}
{ATR
{IN
{Pl -280 60}
}
}
}
}
}
}
```

PCB PART: 7400A

Part File: 7400A.PRT



PDIF File: 7400A.PDF

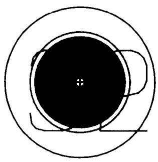
```
%***********************************************************************
%                                                                      *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.             *
%                                                                      *
%     Program  :   PDIF-OUT VERSION 1.31                               *
%     Date     :   Oct 14 1986                                         *
%     Time     :   03:00:15 PM                                         *
%     File In  :   7400A.PRT                                           *
%     File Out :   7400A.PDF                                           *
%     Format   :   P-CAD DATABASE INTERCHANGE FORMAT                   *
%                                                                      *
%***********************************************************************


{COMPONENT 7400A.PRT


{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "PC-Board"}
{DBvrev 1.00}
{DBtime "Dec. 5, 1985        12:34 p.m.          "}
{DBunit "MIL"}
{DBgrid 1}
{Lyrstr "PADCOM" 7 "FLCOMP" 7 "PADSLD" 8 "FLSOLD" 8 "PADINT" 9 "FLINT" 9
"GNDCON" 10 "FLGCON" 10 "GNDCLR" 12 "FLGCLR" 12 "PWRCLR" 13
"FLPCLR" 13 "SLDMSK" 14 "FLSMSK" 14 "DRILL" 15 "FLDRLL" 15 "PIN" 4
"BRDOUT" 4 "FLTARG" 4 "SLKSCR" 6 "DEVICE" 5 "ATTR" 6 "REFDES" 6
"COMP" 1 "SOLDER" 2 "INT1" 3}
}

{USER
{VIEW
{Mode SYMB}
{Nlst CLOSE}
{Vw 750 400 4}
{Lv 21 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 2 2 0 0 2 2 0}
{Gs 50 50}
}
}

{DISPLAY
[Ly "DEVICE"]
[Ls "SOLID"][Wd 5]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
}

{SYMBOL
{PIN_DEF
[Ly "PIN"]
{P 1 {Pt 1}{Lq 1}{Ploc 600 700}}
{P 2 {Pt 2}{Lq 1}{Ploc 600 600}}
{P 3 {Pt 2}{Lq 0}{Ploc 600 500}}
```

```
{P 4 {Pt 2}{Lq 2}{Ploc 600 400}}
{P 5 {Pt 2}{Lq 2}{Ploc 600 300}}
{P 6 {Pt 2}{Lq 0}{Ploc 600 200}}
{P 7 {Pt 3}{Lq 0}{Ploc 600 100}}
{P 8 {Pt 2}{Lq 0}{Ploc 900 100}}
{P 9 {Pt 2}{Lq 3}{Ploc 900 200}}
{P 10 {Pt 2}{Lq 3}{Ploc 900 300}}
{P 11 {Pt 2}{Lq 0}{Ploc 900 400}}
{P 12 {Pt 2}{Lq 4}{Ploc 900 500}}
{P 13 {Pt 2}{Lq 4}{Ploc 900 600}}
{P 14 {Pt 4}{Lq 0}{Ploc 900 700}}
}
{SPKG
{Sp INA 1 4 9 12}
{Sp INB 2 5 10 13}
{Sp OUTY 3 6 8 11}
}

{PIC
[Ly "SLKSCR"]
[Ls "SOLID"][Wd 5]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{A 750 750 50 180 0}
{L 800 750 850 750 850 50 650 50 650 750 700 750}
[Ly "DEVICE"]
{T "7400" 750 400}
}

{ATR
{IN
{Org 600 700}
{Ty 32767}
}
}
}

{DETAIL
{ANNOTATE
}

{NET_DEF
{N 1
}
{N 2
}
{N 3
}
{N 4
}
{N 5
}
{N 6
}
```

```
{N 7
}
{N 8
}
{N 9
}
{N 10
}
{N 11
}
{N 12
}
{N 13
}
{N 14
}
}

{PAD_STACK
}

{SUBCOMP
}
}
}
```

PADSTACK: C60R32

Padstack File: C60R32.PS



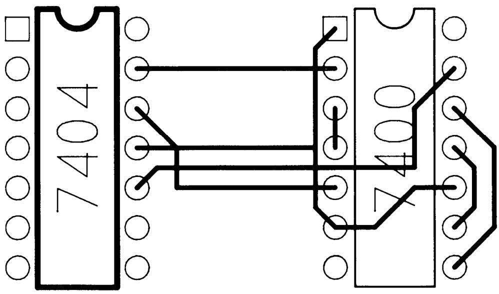
PDIF File: C60R32.PDF

```
%***********************************************************************
%                                                                      *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.             *
%                                                                      *
%     Program  :   PDIF-OUT VERSION 1.31                               *
%     Date     :   Oct 14 1986                                         *
%     Time     :   03:00:35 PM                                         *
%     File In  :   C60R32.PS                                           *
%     File Out :   C60R32.PDF                                          *
%     Format   :   P-CAD DATABASE INTERCHANGE FORMAT                   *
%                                                                      *
%***********************************************************************


{COMPONENT C60R32.PS


{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "PC-Board"}
{DBvrev 1.00}
{DBtime "May. 11, 1984         3:03 p.m.        "}
{DBunit "MIL"}
{DBgrid 1}
{Lyrstr "PADCOM" 7 "FLCOMP" 7 "PADSLD" 8 "FLSOLD" 8 "PADINT" 9 "FLINT" 9
"GNDCON" 10 "FLGCON" 10 "GNDCLR" 12 "FLGCLR" 12 "PWRCLR" 13
"FLPCLR" 13 "SLDMSK" 14 "FLSMSK" 14 "DRILL" 11 "FLDRLL" 11 "PIN" 1
"BRDOUT" 5 "FLTARG" 4 "SLKSCR" 6 "DEVICE" 5 "ATTR" 6 "REFDES" 6
"COMP" 1 "SOLDER" 2 "INT1" 3}
}

{USER
{VIEW
{Mode SYMB}
{Nlst CLOSE}
{Vw 1820 1575 1}
{Lv 17 2 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 2 0 0 0 0 0 0 0 0 0}
{Gs 5 5}
}
}

{DISPLAY
[Ly "PIN"]
[Ls "SOLID"][Wd 25]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
}

{SYMBOL
{PIN_DEF
}

{PIC
[Ly "FLDRLL"]
[Ls "SOLID"][Wd 25]
```

```
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{Fl 1825 1580 23}
[Ly "FLSMSK"]
{Fl 1825 1580 19}
[Ly "FLCOMP"]
{Fl 1825 1580 9}
[Ly "SLDMSK"]
[Wd 0]
{C 1825 1580 45}
[Ly "DRILL"]
{L 1815 1580 1835 1580}
{L 1825 1590 1825 1570}
{T "32" 1830 1575}
[Ly "PADCOM"]
{C 1825 1580 30}
[Ly "PIN"]
[Wd 25]
{C 1825 1580 15}
}

{ATR
{IN
{Org 1825 1580}
{Ty 32767}
}
}
}

{DETAIL
{ANNOTATE
}

{NET_DEF
}

{PAD_STACK
}

{SUBCOMP
}
}
}
```

PCB DESIGN: EX4

PCB File: EX4.PCB



PDIF File: EX4.PDF

```
%***********************************************************************
%                                                                      *
%     Copyright (C) 1983,1986 - Personal CAD Systems, Inc.             *
%                                                                      *
%     Program  :  PDIF-OUT VERSION 1.31                                *
%     Date     :  Oct 14 1986                                          *
%     Time     :  02:59:29 PM                                          *
%     File In  :  EX4.PCB                                              *
%     File Out :  EX4.PDF                                              *
%     Format   :  P-CAD DATABASE INTERCHANGE FORMAT                    *
%                                                                      *
%***********************************************************************

{COMPONENT EX4.PCB

{ENVIRONMENT
{PDIFvrev 1.31}
{Program "PDIF-OUT Version 1.31"}
{DBtype "PC-Board"}
{DBvrev 1.00}
{DBtime "Dec. 5, 1985        12:40 p.m.          "}
{DBunit "MIL"}
{DBgrid 1}
{Lyrstr "PADCOM" 7 "FLCOMP" 7 "PADSLD" 8 "FLSOLD" 8 "PADINT" 9 "FLINT" 9
"GNDCON" 10 "FLGCON" 10 "GNDCLR" 12 "FLGCLR" 12 "PWRCLR" 13
"FLPCLR" 13 "SLDMSK" 14 "FLSMSK" 14 "DRILL" 15 "FLDRLL" 15 "PIN" 4
"BRDOUT" 4 "FLTARG" 4 "SLKSCR" 6 "DEVICE" 5 "ATTR" 6 "REFDES" 6
"COMP" 1 "SOLDER" 2 "INT1" 3}
}

{USER
{VIEW
{Mode DETL}
{Nlst CLOSE}
{Vw 750 350 4}
{Lv 25 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 2 2 0}
{Gs 50 50}
}
}

{DISPLAY
[Ly "SOLDER"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
}

{SYMBOL
{PIN_DEF
}

{PIC
}
```

```
{ATR
{IN
{Org -32767 -32767}
{Ty 255}
}
}
}

{DETAIL
{ANNOTATE
}

{NET_DEF
{N XN00000
{DG
[Ly "SOLDER"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 950 650 900 600 900 200 950 150 1050 150 1150 250 1250 250}
[Ly "COMP"]
{W 900 350 450 350}
{V 900 350 0}
}
}
{N XN00001
{DG
[Ly "COMP"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 550 250 950 250}
{V 550 250 0}
[Ly "SOLDER"]
{W 450 450 550 350 550 250}
}
}
{N XN00002
{DG
[Ly "COMP"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 450 550 950 550}
}
}
{N XN00003
{DG
[Ly "SOLDER"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 1250 350 1300 300 1300 200 1250 150}
}
}
{N XN00004
{DG
[Ly "SOLDER"]
```

```
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 1250 450 1350 350 1350 150 1250 50}
}
}
{N XN00005
{DG
[Ly "COMP"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 1150 300 500 300 450 250}
{V 1150 300 0}
[Ly "SOLDER"]
{W 1250 550 1150 450 1150 300}
}
}
{N XN00006
{DG
[Ly "SOLDER"]
[Ls "SOLID"][Wd 12]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{W 950 450 950 350}
}
}
}

{PAD_STACK
{Pad 0 "V50R28.PS" "V50R28.PS"}
{Pad 1 "C60S32.PS" "N60S32.PS"}
{Pad 2 "C60R32.PS" "N60R32.PS"}
{Pad 3 "C60R32.PS" "N60R32.PS"}
{Pad 4 "C60R32.PS" "N60R32.PS"}
{Pad 5 "C60R32.PS" "N60R32.PS"}
{PAD_DEF V50R28.PS
{ATR
{IN
{Org 0 0}
{Ty 32767}
}
}

{PIC
[Ly "SLDMSK"]
[Ls "SOLID"][Wd 0]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{C 0 0 40}
[Ly "FLSMSK"]
{Fl 0 0 17}
[Ly "FLCOMP"]
{Fl 0 0 15}
[Ly "FLDRLL"]
{Fl 0 0 23}
[Ly "DRILL"]
{L -10 0 10 0}
```

```
{L 0 10 0 -10}
[Ts 60][Tr 0]
{T "28" 5 -5}
[Ly "PADINT"]
{C 0 0 25}
}
}
{PAD_DEF N60S32.PS
{ATR
{IN
{Org 0 0}
{Ty 32767}
}
}

{PIC
[Ly "PADCOM"]
[Ls "SOLID"][Wd 0]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{R -30 -30 30 30}
[Ly "SLDMSK"]
{C 0 0 45}
[Ly "FLCOMP"]
{Fl 0 0 10}
[Ly "FLDRLL"]
{Fl 0 0 23}
[Ly "FLSMSK"]
{Fl 0 0 19}
[Ly "DRILL"]
{L -10 0 10 0}
{L 0 10 0 -10}
{T "32" 5 -5}
[Ly "PIN"]
{C 0 0 25}
}
}
{PAD_DEF C60S32.PS
{ATR
{IN
{Org 0 0}
{Ty 32767}
}
}

{PIC
[Ly "PADCOM"]
[Ls "SOLID"][Wd 0]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{R -30 -30 30 30}
[Ly "SLDMSK"]
{C 0 0 45}
[Ly "FLCOMP"]
{Fl 0 0 10}
```

```
[Ly "FLDRLL"]
{Fl 0 0 23}
[Ly "FLSMSK"]
{Fl 0 0 19}
[Ly "DRILL"]
{L -10 0 10 0}
{L 0 10 0 -10}
{T "32" 5 -5}
[Ly "PIN"]
{Fr -25 -25 25 25}
}
}
{PAD_DEF N60R32.PS
{ATR
{IN
{Org 0 0}
{Ty 32767}
}
}

{PIC
[Ly "FLDRLL"]
[Ls "SOLID"][Wd 0]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{Fl 0 0 23}
[Ly "FLSMSK"]
{Fl 0 0 19}
[Ly "FLCOMP"]
{Fl 0 0 9}
[Ly "SLDMSK"]
{C 0 0 45}
[Ly "PADCOM"]
{C 0 0 30}
[Ly "DRILL"]
{L -10 0 10 0}
{L 0 10 0 -10}
{T "32" 5 -5}
[Ly "PIN"]
{C 0 0 25}
}
}
{PAD_DEF C60R32.PS
{ATR
{IN
{Org 0 0}
{Ty 32767}
}
}

{PIC
[Ly "FLDRLL"]
[Ls "SOLID"][Wd 0]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{Fl 0 0 23}
```

```
[Ly "FLSMSK"]
{Fl 0 0 19}
[Ly "FLCOMP"]
{Fl 0 0 9}
[Ly "SLDMSK"]
{C 0 0 45}
[Ly "DRILL"]
{L -10 0 10 0}
{L 0 10 0 -10}
{T "32" 5 -5}
[Ly "PADCOM"]
{C 0 0 30}
[Ly "PIN"]
[Wd 25]
{C 0 0 15}
}
}
}

{SUBCOMP
{COMP_DEF 7400A.PRT
{PIN_DEF
[Ly "PIN"]
{P 1 {Pt 1}{Lq 1}{Ploc 0 0}}
{P 2 {Pt 2}{Lq 1}{Ploc 0 -100}}
{P 3 {Pt 2}{Lq 0}{Ploc 0 -200}}
{P 4 {Pt 2}{Lq 2}{Ploc 0 -300}}
{P 5 {Pt 2}{Lq 2}{Ploc 0 -400}}
{P 6 {Pt 2}{Lq 0}{Ploc 0 -500}}
{P 7 {Pt 3}{Lq 0}{Ploc 0 -600}}
{P 8 {Pt 2}{Lq 0}{Ploc 300 -600}}
{P 9 {Pt 2}{Lq 3}{Ploc 300 -500}}
{P 10 {Pt 2}{Lq 3}{Ploc 300 -400}}
{P 11 {Pt 2}{Lq 0}{Ploc 300 -300}}
{P 12 {Pt 2}{Lq 4}{Ploc 300 -200}}
{P 13 {Pt 2}{Lq 4}{Ploc 300 -100}}
{P 14 {Pt 4}{Lq 0}{Ploc 300 0}}
}
{SPKG
{Sp INA 1 4 9 12}
{Sp INB 2 5 10 13}
{Sp OUTY 3 6 8 11}
}

{PIC
[Ly "SLKSCR"]
[Ls "SOLID"][Wd 5]
[Ts 60][Tj "CC"][Tr 0][Tm "N"]
{A 150 50 50 180 0}
{L 200 50 250 50 250 -650 50 -650 50 50 100 50}
[Ly "DEVICE"]
[Ts 125][Tr 1]
{T "7400" 150 -300}
}
```

```
{ATR
{IN
{Ty 10000}
{Vr 0}
}
}
}
{I 7400A.PRT XC00000
{CN XN00000 XN00002 XN00006 XN00006 XN00001 XN00000 ? XN00004 XN00003
XN00000 XN00003 XN00004 XN00005 ?}
{ATR
{IN
{Pl 950 650}
}
}
}
{COMP_DEF 7404T.PRT
{PIN_DEF
[Ly "PIN"]
{P 1 {Pt 1}{Lq 0}{Ploc 0 0}}
{P 21 {Pt 2}{Lq 0}{Ploc 0 -100}}
{P 3 {Pt 2}{Lq 0}{Ploc 0 -200}}
{P 4 {Pt 2}{Lq 0}{Ploc 0 -300}}
{P 5 {Pt 2}{Lq 0}{Ploc 0 -400}}
{P 6 {Pt 2}{Lq 0}{Ploc 0 -500}}
{P 7 {Pt 3}{Lq 0}{Ploc 0 -600}}
{P 8 {Pt 2}{Lq 0}{Ploc 300 -600}}
{P 9 {Pt 2}{Lq 0}{Ploc 300 -500}}
{P 10 {Pt 2}{Lq 0}{Ploc 300 -400}}
{P 11 {Pt 2}{Lq 0}{Ploc 300 -300}}
{P 12 {Pt 2}{Lq 0}{Ploc 300 -200}}
{P 13 {Pt 2}{Lq 0}{Ploc 300 -100}}
{P 14 {Pt 4}{Lq 0}{Ploc 300 0}}
}

{PIC
[Ly "SLKSCR"]
[Ls "SOLID"][Wd 15]
[Ts 125][Tj "CC"][Tr 1][Tm "N"]
{A 150 50 50 180 0}
{L 200 50 250 50 250 -650 50 -650 50 50 100 50}
[Ly "DEVICE"]
{T "7404" 150 -300}
}
{ATR
{IN
{Ty 32767}
{Vr 0}
}
}
}
{I 7404T.PRT XC00001
{CN ? ? ? ? ? ? ? ? ? XN00005 XN00000 XN00001 XN00002 ?}
{ATR
```

```
{IN
{Pl 150 650}
}
}
}
}
}
}
```

TOOLS FOR MODERN DESIGN™