

<b>1</b>	<b>PREFACE</b> .....	<b>3</b>
<b>2</b>	<b>OVERVIEW</b> .....	<b>4</b>
2.1	FUNCTIONAL OBJECTIVES .....	4
2.2	FUNCTIONAL OUTLINE .....	5
2.3	SUPPORTED FPGA/CPLD VENDORS AND DEVELOPMENT TOOLS .....	6
<b>3</b>	<b>IMPORT SPECIFICATIONS</b> .....	<b>7</b>
3.1	IMPORT SPECIFICATIONS FOR ACTEL PIN REPORT FILES.....	7
3.1.1	<i>Outline</i> .....	7
3.1.2	<i>General Program Specifications</i> .....	9
3.1.3	<i>Importing the Design Name, Part Name, and data for Date of Creation</i> .....	10
3.1.4	<i>Importing Physical Pin Numbers and Logical Pin Names</i> .....	14
3.2	IMPORT SPECIFICATIONS FOR ALTERA PIN REPORT FILES.....	15
3.2.1	<i>Outline</i> .....	15
3.2.2	<i>General Program Specifications</i> .....	17
3.2.3	<i>Importing the Design Name and Part Name</i> .....	18
3.2.4	<i>Importing Physical Pin Numbers and Logical Pin Names</i> .....	19
3.3	IMPORT SPECIFICATIONS FOR LATTICE PIN REPORT FILES.....	20
3.3.1	<i>Outline</i> .....	20
3.3.2	<i>General Program Specifications</i> .....	21
3.3.3	<i>Importing the design name, part name, date of creation, and vendor name..</i> .....	22
3.3.4	<i>Importing the logical pin name, I/O property, and physical pin numbers</i> .....	24
3.4	IMPORT SPECIFICATIONS FOR XILINX PIN REPORT FILES (FPGA TYPE) .....	25
3.4.1	<i>Outline</i> .....	25
3.4.2	<i>General Program Specifications</i> .....	27
3.4.3	<i>Importing the Design Name</i> .....	28
3.4.4	<i>Importing the Part Name</i> .....	29
3.4.5	<i>Importing Data for Date of Creation</i> .....	30
3.4.6	<i>Importing Physical Pin Numbers and Logical Pin Names</i> .....	31
3.5	IMPORT SPECIFICATIONS FOR XILINX PIN REPORT FILES (CPLD TYPE) .....	33
3.5.1	<i>Outline</i> .....	33
3.5.2	<i>General Program Specifications</i> .....	35
3.5.3	<i>Importing the Design Name and data for Date of Creation</i> .....	36
3.5.4	<i>Importing the Part Name and Pin Properties</i> .....	38
<b>4</b>	<b>ERROR AND WARNING MESSAGES</b> .....	<b>42</b>

4.1	WHEN IMPORTING ACTEL PIN REPORT FILES.....	42
4.2	WHEN IMPORTING ALTERA PIN REPORT FILES.....	45
4.3	WHEN IMPORTING LATTICE PIN REPORT FILES.....	47
4.4	WHEN IMPORTING XILINX PIN REPORT FILES (FPGA TYPE) .....	49
4.5	WHEN IMPORTING XILINX PIN REPORT FILES (CPLD TYPE) .....	52

# 1 PREFACE

This manual describes functional specifications for the *Component Designer Pin Report File Import Program*, which performs one of the *Component Designer* import functions.

Pin report files are created following place and route using the CPLD/FPGA development tool. The report files are text files in vendor-specific format, containing descriptions of physical pin numbers and logical pin names.

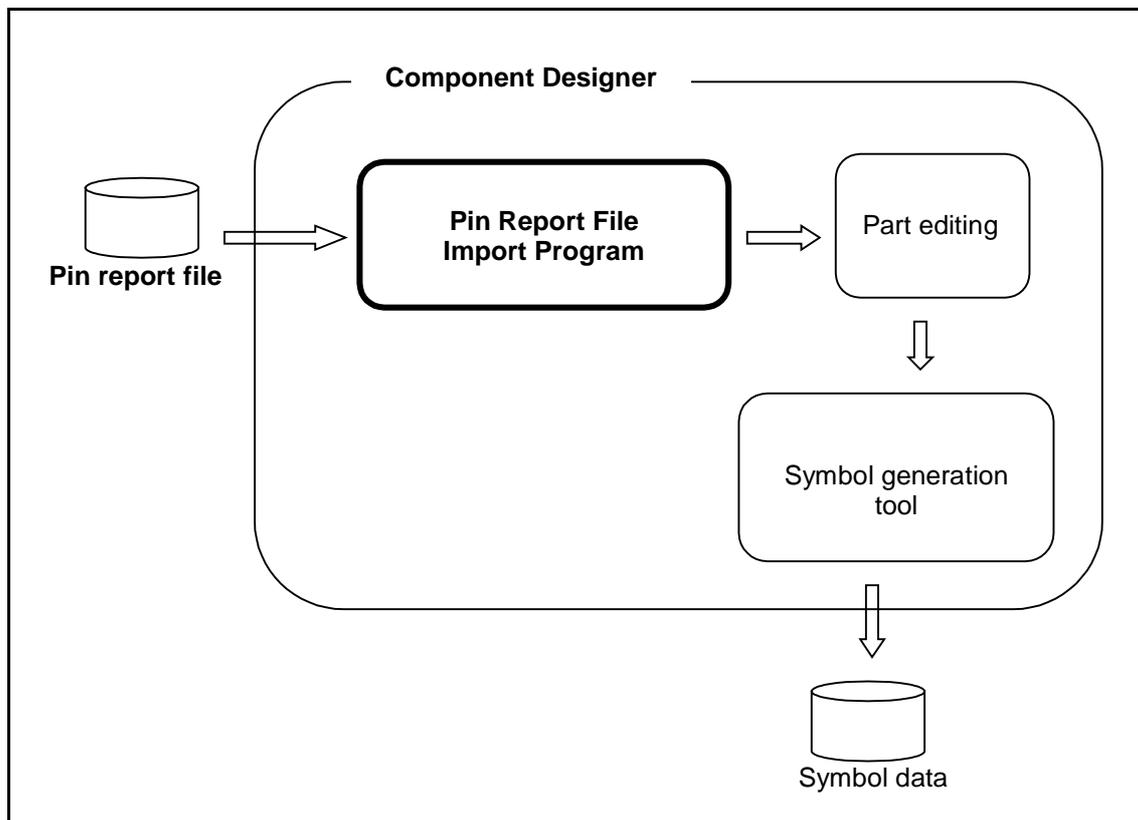
## 2 OVERVIEW

This chapter provides an overview of the *Pin Report File Import Program*.

### 2.1 Functional Objectives

Mainly the current FPGA and CPLD devices being used have over 100 pins, and registering these symbols on schematic editor requires much labor.

This software imports the data required for symbol generation from pin report files created following place and route using the FPGA/CPLD development tool into the *Component Designer*. The above-mentioned work is useful for generating FPGA and CPLD symbols for schematic editor *Sysystem Designer* with a fast and flexible way.



## 2.2 Functional Outline

This software accepts *pin report files* as input, acquires the following data from the file, and causes it to be reflected in the *Component Designer*.

- Design name
- Part name
- Physical pin number
- Logical pin name
- Vendor name
- Date of creation

The following indicates the *Component Designer* target in which the data is reflected.

### **Design name**

Reflected in the device property “asicDesignName” in the *Component Designer*.

### **Part name**

Reflected in the device property “Part Name” in the *Component Designer*.

### **Physical pin number**

Reflected in the pin property “Pin Number” in the *Component Designer*.

### **Logical pin name**

Reflected in the pin property “User Label” in the *Component Designer*.

### **Vendor name**

Reflected in the device property “asicVendorName” in the *Component Designer*.

### **Date of creation**

Reflected in the device property “asicDesignDate” in the *Component Designer*.

## 2.3 Supported FPGA/CPLD Vendors and Development Tools

This program supports the FPGA/CPLD vendors listed below.

- ACTEL
- ALTERA
- LATTICE
- XILINX

This program has been created in conformance with the following FPGA/CPLD development tools and pin report files (extensions).

FPGA/CPLD Vendor Name	Development Tool Name	Version	Pin Report File Extension
ACTEL Company	Designer	R2-1998 R1-2000 ~ R2-2001	.prp .rpt
ALTERA Company	Max+plusII QuartusII Quartus	9.0 ~ 10.1 1.0 ~ 2.0 1999.06 ~ 2000.09	.pin .pin .pin
LATTICE Company	ispEXPERT	8.0.x	.ppn
XILINX Company	M1	1.3,1.5 ~ ISE4	.pad/.rpt

- \* Pin report files output by XILINX's M1 have different file extensions: ".pad" for the FPGA type and ".rpt" for the CPLD type. Pin report files for the CPLD type also contain other data in addition to that usually provided by pin reports.

## 3 IMPORT SPECIFICATIONS

This chapter provides *pin report file* import specifications for each vendor.

### 3.1 Import Specifications for Actel Pin Report Files

#### 3.1.1 Outline

*Actel pin report files* are created by the utility function of the Actel's FPGA development tool "Designer TM (Ver. 1998R2)," after implementation of the tool. The file extension is ".prp" or ".rpt."

- \* Please output *pin report files* listed by number with utility function in which you may select "List By:Number".
- \* Both DOS and Unix formats can be read.

This software acquires the **design name, part name, physical pin numbers, logical pin names, vendor name, and date of creation** from the *Actel pin report file*, then reflects these data in the *Component Designer*.

The default **vendor name** when importing the Actel pin report file is "Actel."

The Actel pin report file does not provide **speed grade** and **temperature range** data for the **part name**. Enter these data from the *Component Designer* when importing the pin report file.

For further details, refer to the documentation shown below.

System Designer Online Help "Tool Menu/Component Designer" - "File" - "Import" - "Pin Report File Import" dialog box.

Specifications for importing **design name, part name, physical pin numbers, logical pin names, and date of creation** from *Actel pin report files* are described in the following pages.

Example: Actel pin report file:

```
*****
Pin Report - Date: Thu Sep 03 19:10:13 1998  Pinchecksum: ded01453_d3078113
  Design Name: goodct  Family: ACT2/1200XL  Die: A1225XL  Package: 100 CPGA
*****

Number  Name          Fixed Load  Level
-----  -
A1  UNASSIGNED
A2  UNASSIGNED
A3  UNASSIGNED
A4  UNASSIGNED
A5  UNASSIGNED
A6  UNASSIGNED
A7  UNASSIGNED
A8  B0
A9  B2
```

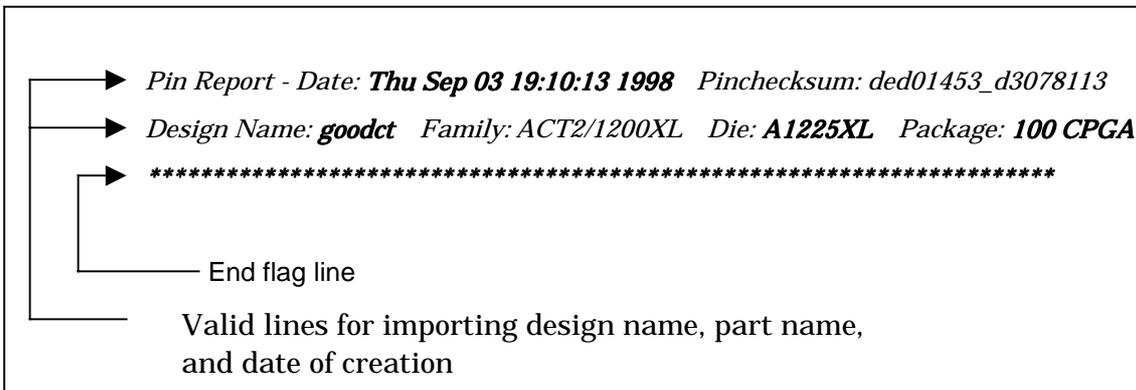
### 3.1.2 General Program Specifications

This program processes each line of the pin report file, starting from the first line.

The program first searches for an occurrence of the character string in which the first half of the line separated by a hyphen (-) is "Pin Report" (case-sensitive, and disregarding spaces, tabs, and line-feeds at the beginning and end of the line).

When the keyword "Pin Report" is found, the program begins importing **design name, part name, and date of creation** from the line. This continues until a line meeting the following conditions is found:

- Disregarding space and tab indicators at the beginning of each line, the line that begins with "\*\*\*."



When the program finishes importing design name, part name, and date of creation, it starts importing pin properties(physical pin numbers and logical pin names) from the next line (the one following the end flag line).

- \* If two or more lines contain the keyword "Pin Report," processing occurs only for the first line found.

### 3.1.3 Importing the Design Name, Part Name, and data for Date of Creation

- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

Starting from the beginning of the Valid lines for importing design name, part name, and date of creation (refer to Section 3.1.2, “General Program Specifications”), the program searches for the following keywords:

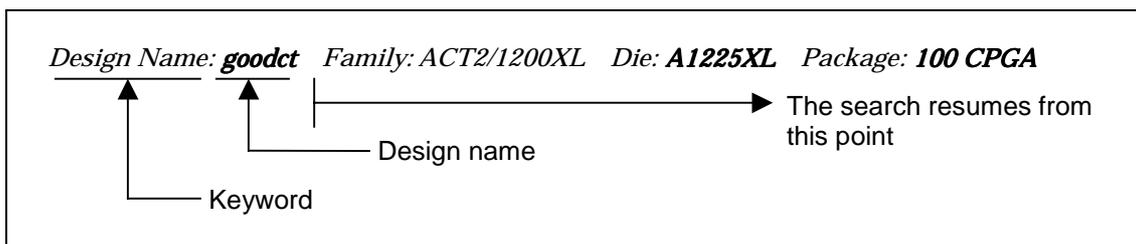
1. Design Name:
2. Die:
3. Package:
4. Date:

- \* All space and tabs preceding the colon (:) are ignored.
- \* The search is case-sensitive.

When keyword 1. is found, the token immediately following is stored in memory as the **design name**.

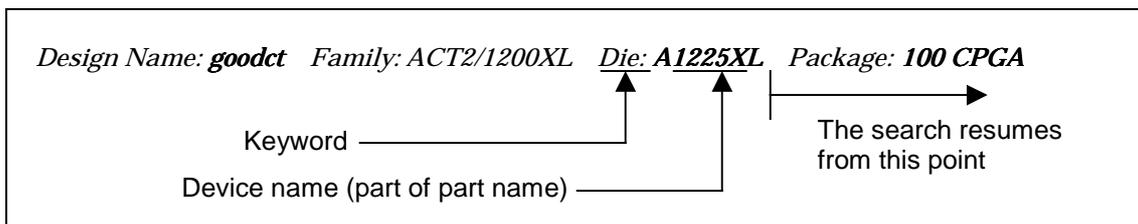
If no token follows the keyword, a null character is stored in memory.

If this token is found, the search continues for keywords 1. through 4. If this token is found, the search continues for keywords 1. through 4. from the point right after the token.



When keyword 2. is found, the token immediately following is stored in memory as **device name** data (which considered part of the part name). If no token follows the keyword, a null character is stored in memory.

If this token is found, the search continues for keywords 1. through 4. If this token is found, the search continues for keywords 1. through 4. from the point right after the token.

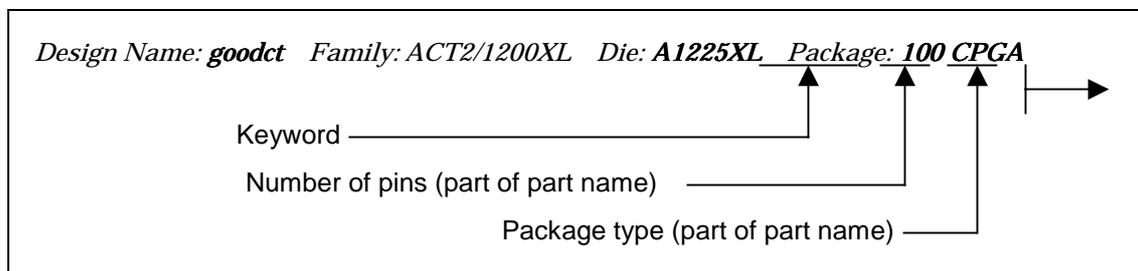


When keyword 3. is found, the program checks to see if the first character of the token immediately following is a number.

- If the first character is a number, the token (i.e., its numeric part) is stored in memory as data for **number of pins** (which considered part of the part name).

If the token immediately following matches the “package type” in the correspondence table shown below, the corresponding element “Apply in Part Name” is stored in memory as **package type**, which is also part of the part name.

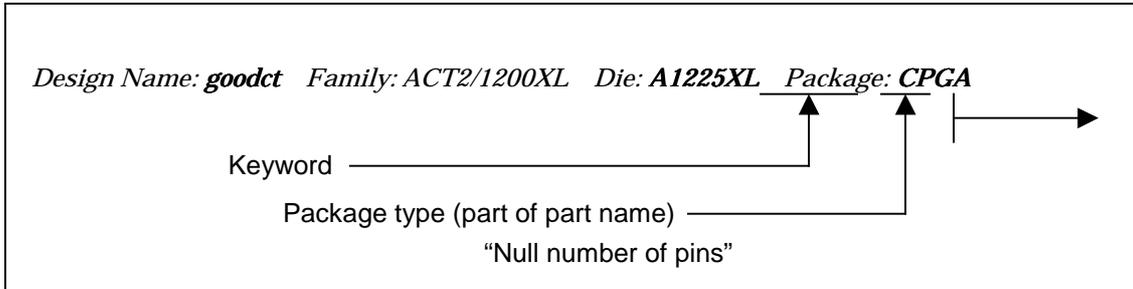
If no token exists after the total pin properties, or if a token is found that fails to match the correspondence table, a null character is stored in memory for package type data.



- If the first character of the token is not a number and the token matches the package type in the correspondence table shown below, the corresponding element “Apply in Part Name” is stored in memory as package type (which considered part of the part name).

If the token does not match the correspondence table, a null character is stored in memory for package type data.

A null character is stored in memory for **number of pins** (which considered part of the part name).



If these tokens are found, the search continues for keywords 1. through 4. from the point right after the token.

Package type correspondence table:

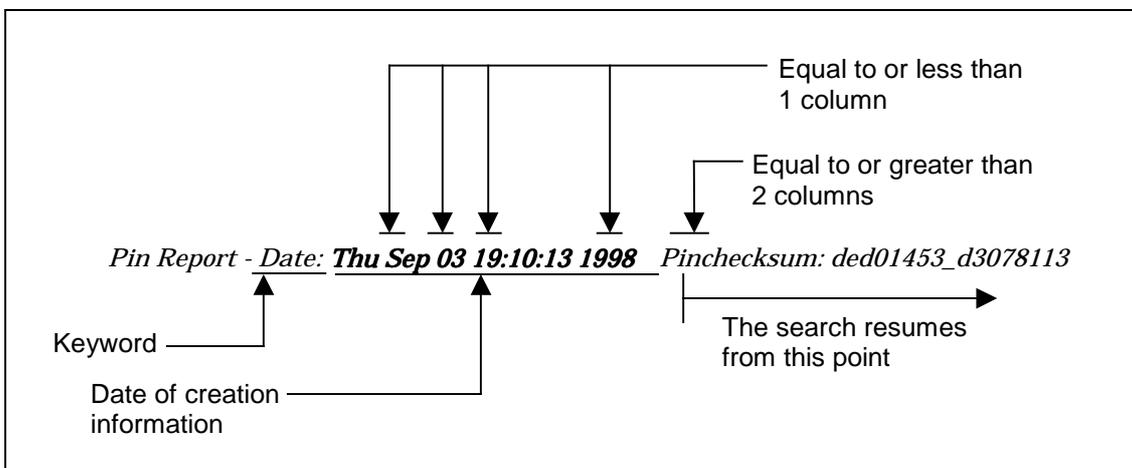
Package Type (Pin Report)	:	Apply in Part Name
PCLL	:	PL
PQFP	:	PQ
CQFP	:	CQ
CPGA	:	PG
VQFP	:	VQ
TQFP	:	TQ
RQFP	:	RQ
BGA	:	BG

When keyword 4. is found, the program reads the character string immediately following the keyword until it reaches a position at which two consecutive space or tabs occur. If this sequence of space or tabs is not found until the end of the line, all characters leading up to the end of the line are read.

The character string read, with all space, tab, and line-feeds at the beginning and end of the string stripped, is stored in memory as **date of creation**.

If no token follows the keyword, a null character is stored in memory.

The search then resumes for keywords 1. through 4. from the point right after the token.



\* If multiple instances of the keyword are found, the first data to be stored applies, with the exception of null.

When the end-flag line (refer to Section 3.1.2, “General Program Specifications”) is encountered, the program concludes its search for keywords and imports the data for design name, part name, and date of creation, based on the data stored.

For design name and date of creation, data is imported directly, as stored.

For the part name, data is imported after being merged in order of <device name>, <speed grade>, <package type>, <total pin>, and <temperature range>.

\* Data for speed grade and temperature range is entered from the **Component Designer**.

For the sample Actel pin report file, the device name is A1225A, package type is PG (CPGA), and the total pin count is 100. Assuming that the speed grade and temperature range entered from the **Component Designer** are “-1” and “C” (Commercial) respectively, the imported data consists of A1225A-1PG100C.

### 3.1.4 Importing Physical Pin Numbers and Logical Pin Names

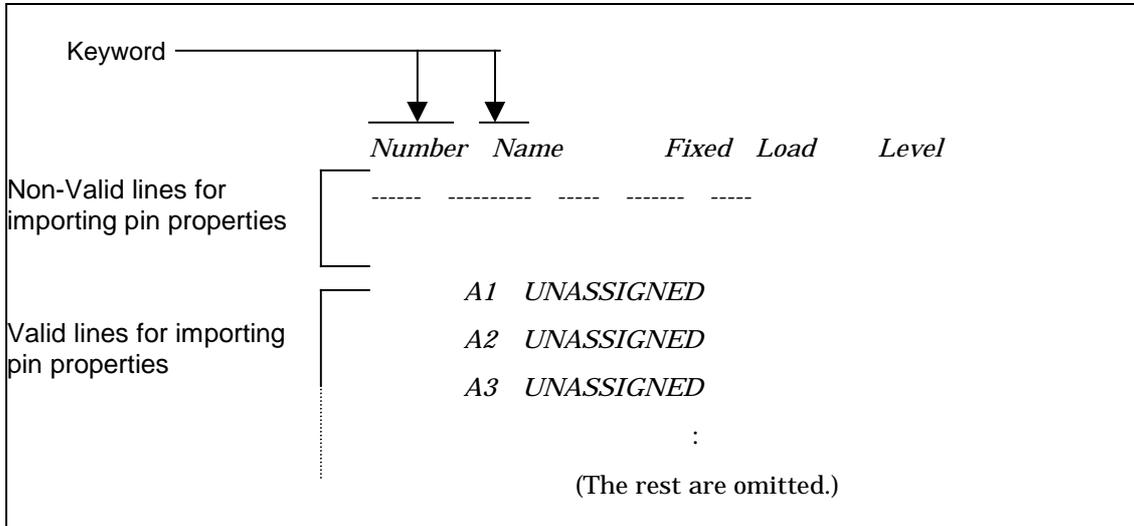
\* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

Starting from the line following the end-of-import flag line for design name, part name, and date of creation (refer to Section 3.1.2, “General Program Specifications”) and until the last line, the program imports pin properties (physical pin numbers and logical pin names) and logical pin names).

The program first searches for a line in which the first token is “Number” (case-sensitive search) and in which the next token is “Name” (case-sensitive search).

When a line is found meeting these conditions, the program imports pin properties (physical pin numbers and logical pin names) from all lines, starting with the next line, excluding those that fail to meet the following conditions.

- Lines consisting of hyphens (-), spaces, tabs, and line-feeds only



Pin properties are imported from one valid line for one pin.

When these properties are imported, the first token of the valid line is imported as a **physical pin number**, and the next token is imported as a **logical pin name**.

When the importing reaches the final line, the program outputs an imported pin count to a log file before closing.

## 3.2 Import Specifications for Altera Pin Report Files

### 3.2.1 Outline

The Altera pin report files are created by Altera's FPGA development tool "MAX+II™ (Ver.9.0)" after implementation of the tool. The file extension is ".pin."

\* Both DOS and Unix formats can be read.

This software acquires the **design name, part name, physical pin numbers, logical pin names, vendor name, and date of creation** from the Altera pin report file and has these data reflected in the Component Designer.

The default **vendor name** when importing the Altera pin report file is "Altera."

Since the Altera pin report file provides no description of **the date of creation**, the updated date for the pin report file is referenced instead.

Specifications regarding data import for **design name, part name, physical pin numbers, and logical pin names** from the Altera pin report file are described in the following pages.

Example: An Altera pin report file:

```
<First part omitted>
:

GND = Dedicated ground pin or unused dedicated input, which MUST be connected to
GND.
RESERVED = Unused I/O pin, which MUST be left unconnected.
-----

CHIP "goodct" ASSIGNED TO AN EPF10K10LC84-3
      RESERVED : 11
      RESERVED : 10
      b3 : 9
      a8 : 8
      a6 : 7
      a7 : 6
      a5 : 5
:
<Remainder omitted>
```

### 3.2.2 General Program Specifications

- \* **Note 1:** The word "token" here refers to a character string isolated by a space, tab, or line-feed.
- \* **Note 2:** Comment statements (the statements following "--") are not processed.

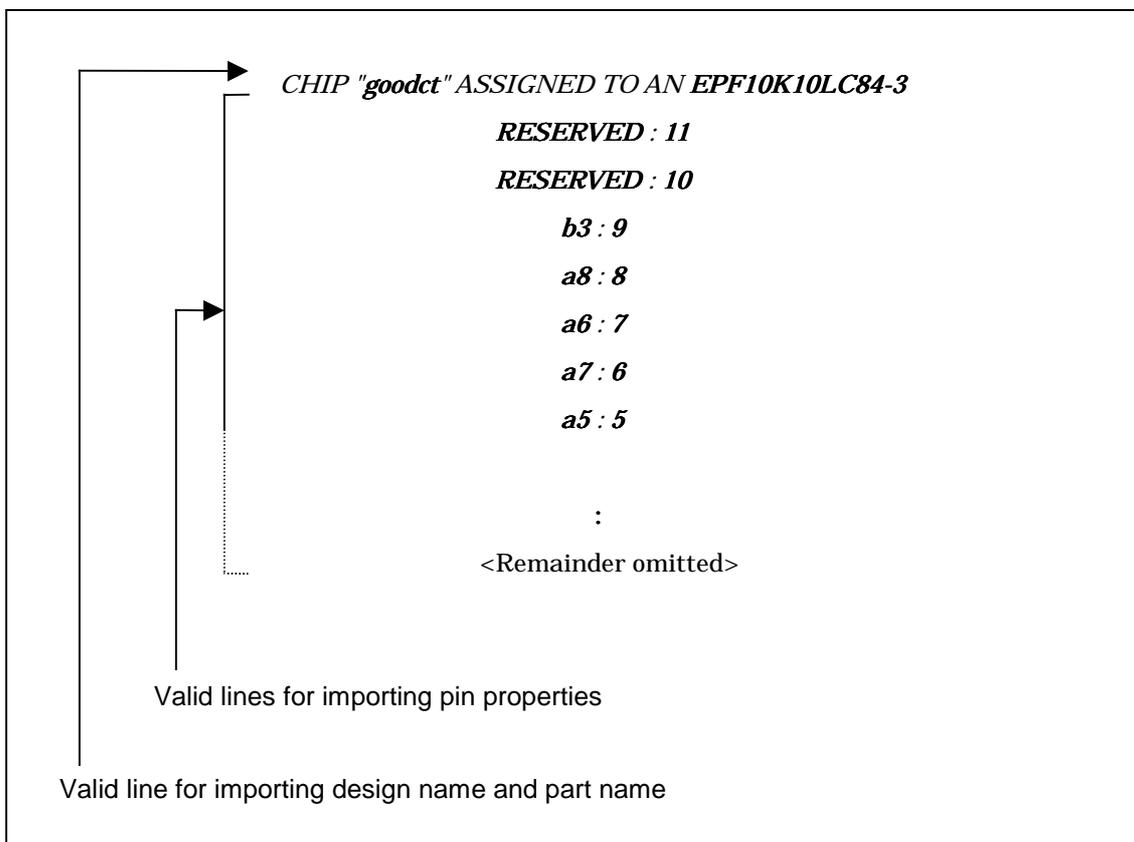
This program processes each line of the pin report file, starting from the first line.

The program first searches for a line in which the first token is the keyword shown below.

- **CHIP**

- \* Case-sensitive

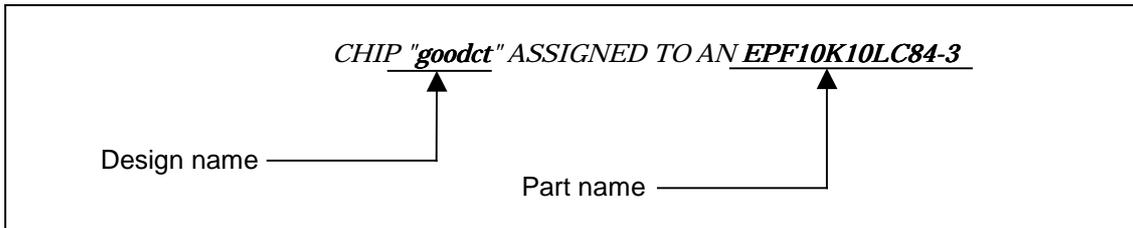
When a line is found that meets these conditions, the program imports design name and part name from the line and imports pin properties (physical pin numbers and logical pin names), starting with the next line and proceeding to the last line of the pin report file.



### 3.2.3 Importing the Design Name and Part Name

- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

Sequentially searching from the beginning of the valid line for importing design name and part name (refer to Section 3.2.2, “General Program Specifications”), the program imports the first character string enclosed in double quotes (") as the **design name** and the last token (ignoring comment lines) as the **part name**.

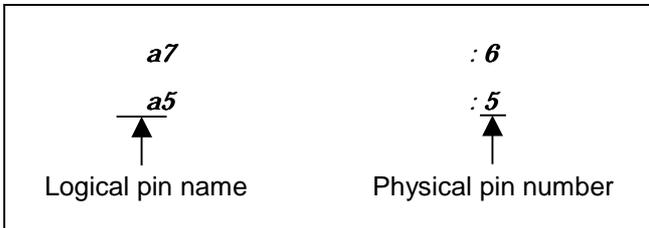


- \* If the valid line for importing design name and part name contains an additional character string, the program may not be able to read the part name correctly. (Comment statements added after the end of this line are not processed.)
- \* The first character of the part name must always be preceded by at least one space or tab.

### 3.2.4 Importing Physical Pin Numbers and Logical Pin Names

Starting from the beginning of the valid lines for importing pin properties (refer to Section 3.2.2, “General Program Specifications”), the program searches for a colon (:). When an occurrence is found, it imports pin properties beginning with that line, with each line providing properties for a single pin.

In each of these lines, the character string in the first half of the line delimited by the colon (:) is stored in memory as **logical pin name**, with all space and tabs at the beginning and end of the string stripped. The character string found in the second half of the line is stored in memory as **physical pin number**, with all space, tab, and line-feeds at the beginning and end of the string stripped.



Unless the physical pin number or logical pin name stored is a null character, the program imports the pin properties into the *Component Designer*.

On reaching the final line, the program outputs the imported pin count to a log file, then closes.

### 3.3 Import Specifications for Lattice Pin Report Files

#### 3.3.1 Outline

Lattice Pin Reports is created by Lattice's development tool, "ispEXPERT TM Ver8.0.x". The file extension is (.ppn).

\* Both DOS and UNIX formats can be read.

This software acquires design name, part name, physical pin numbers, logical pin numbers, I/O property, vendor name, and date of creation from the Lattice pin report file, then reflects these data in the "Component Designer".

Example: LATTICE pin report file

//				
// Software Version 7.2.05				
// Targeted Device : ispLSI5256V-125LB272				
// Platform : Windows NT 4.0				
// Date : Tue Jul 25 13:51:57 2000				
// Elapsed Time : 00:00:00				
//				
	B1	OUT	M3	
	B9	OUT	P3	
	B10	OUT	R2	
	B11	OUT	T1	
	B12	OUT	P4	
	B13	OUT	T2	
	B14	OUT	T3	
	B0	OUT	V1	
	B15	OUT	V2	
	B6	OUT	V3	
	B2	OUT	Y1	

### 3.3.2 General Program Specifications

**\* Note 1: The word "words" here refers to a character string isolated by a space, tab, or line-feed.**

- This program processes the pin report file starting from the first line.
- If a warning or error occurs, the message dialog will appear.
- Even if a warning occurs, the program continues to import a file. However, it may not correctly read the file depending on a nature of warning. Please check the message.
- The Component Designer obtains the following information from the LATTICE Pin Report File.

#### Part Property

“Part Name(partName) “

“Device Name(asicDesignName) “

“Date of Creation (asicDesignDate) “

“Vendor Name (asicVendorName) “

#### Pin Property

“Physical Pin No. (pinNumber) “

“Logical Pin Name (cirPinLabel) “

“Input/Output Property (io) “

- This program does not support the text file generated in the format other than ispEXPERT generation format.

### 3.3.3 Importing the design name, part name, date of creation, and vendor name

The program first checks to see if the first word is //.

- If the program begins with // followed by a keyword, "Targeted Device:", data after "Targeted Device:" is acquired as a part name.
- If the program begins with // followed by the keyword, "Date:", data after "Date:" is acquired as the date of creation.
- If a keyword, "Targeted Device:" was not found, data on a part name is not acquired.
- If a keyword, "Date:" was not found, data on the date of creation is not acquired.
- If multiple instances of keyword, "Targeted Device:" were found, data after the first "Targeted Device:" described is obtained.
- If multiple instances of keyword, "Date:" were found, data after the first "Date:" described is acquired.
- The design name is acquired from the pin report file name (excluding the extension (.ppn)).
- The vendor name will be "LATTICE".
- Keywords are not case-sensitive.
- All space or tabs preceding the colon (":") are deleted before importing the keywords.
- The part name is imported as the part property "partName".
- The device name is imported as the part property "asicDesignName".
- The date of creation is imported as the part property "asicDesignDate".
- The vendor name is imported as the part property "asicVendorName".

Example: Importing the design name, part name, date of creation, and vendor name

```
File name is goodct_lb_80512.ppn <- where "goodct_lb_80512" is the device
name
All vendor names are "LATTICE".

//
//Software Version 7.2.05
//Targeted Device: ispLSI5256V-125LB272 <- "ispLSI5256V-125LB272" is the
part name.
//Platform: Windows NT4.0
//Date: Tue Jul 25 13:51:57 2000 <- "Tue Jul 25 13:51:57 2000" is the date of
creation.
//Elapsed Time: 00:00:00
//
```

### 3.3.4 Importing the logical pin name, I/O property, and physical pin numbers

The program first checks to see if the first word is //.

- If the program does not begin with //, it is considered as the pin information line.
- Words found from the lines that do not begin with // are acquired as "logical pin name", "I/O property", and "physical pin name" in that order.
- If the words found on the pin information line were less than three words, a warning message will be displayed.
- If the words found on the pin information line were more than three words, a warning message will be displayed.
- The I/O property is converted from the LATTICE format to the CR-5000 format according to the following rules.

LATTICE IO	LATTICE PIN NAME	CR-5000
IN	Omitted	INPUT
OUT	Omitted	OUTPUT
SYS	VCC	VCC
SYS	VCCIO	VCC
SYS	GND	GND
SYS	RESET	INPUT
Characters other than above	Omitted	BIDIRECT
No value	Omitted	No value

## 3.4 Import Specifications for Xilinx Pin Report Files (FPGA Type)

### 3.4.1 Outline

Xilinx pin report files (FPGA type) are created by Xilinx's FPGA development tool "M1 TM (Ver.1.3)" after implementation of the tool. The file extension is ".pad."

\* Both DOS and Unix formats can be read.

This software acquires **design name**, **part name**, **physical pin numbers**, **logical pin names**, **vendor name**, and **date of creation** from the Xilinx pin report file (FPGA type), then reflects these data in the *Component Designer*.

The default **vendor name** when importing Xilinx pin report files is "Xilinx."

Since the Xilinx pin report file (FPGA type) lacks **temperature range** data for the **part name**, this data needs to be entered from the *Component Designer* when importing the pin report file.

For further details, refer to the documentation shown below.

System Designer Online Help "Tool Menu/Component Designer" - "File" - "Import" - "Pin Report File Import" dialog box

Specifications regarding data import for **design name**, **part name**, **physical pin numbers**, **logical pin names**, and **date of creation** from the Xilinx pin report file (FPGA type) are described in the following pages.

Example: Xilinx pin report file (FPGA type):

```
PAR: Xilinx Place And Route M1.3.7.
Copyright (c) 1995-1997 Xilinx, Inc. All rights reserved.
Fri Sep 25 16:51:10 1998 rtey

Xilinx PAD Specification File
*****

Input file:      map.ncd
Output file:     goodct.ncd
Part type:      xc4003e
Speed grade:    -1
Package:        pg120

Fri Sep 25 16:51:10 1998

:
<Middle part omitted>
:

Pinout by Pin Number:
+-----+-----+-----+
| Pin Number      | Comp Name      | Constraint      |
+-----+-----+-----+
| A4           |                |                |
| B2           | MCLK        |                |
| B3           |                |                |
| B4           |                |                |
| N9           |                |                |
| N10          |                |                |
| N11          |                |                |
| N13          | A0         |                |
+-----+-----+-----+
```

### 3.4.2 General Program Specifications

- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

This program processes each line of the pin report file, starting from the first line.

The program first checks to see if one of the keywords listed below is found at the beginning of each line (disregarding any space and tabs found at the beginning of the line).

1. Output file:
2. Part type:
3. Speed grade:
4. Package:
5. Pinout by Pin Number:

- \* All space and tabs preceding the colon (:) are ignored.

- \* The search is case-sensitive.

Once the keyword 5. is found, the program begins importing pin properties (physical pin numbers and logical pin names) from the next line, continuing until a line is detected that contains the end-of-pin properties import flag.

**When the program finishes importing pin properties, it also finishes reading the file.** All other necessary data must be written in lines preceding this line (i.e., the line containing keyword 5.).

Among lines preceding the line containing keyword 5. at their beginning,

those containing keyword 1. at their beginning are assumed to be valid lines for importing the **design name**,

those containing keyword 2. at their beginning are assumed to be valid lines for importing part of the part name (**part type**),

those containing keyword 3. at their beginning are assumed to be valid lines for importing part of the part name (**speed grade**), and

those containing keyword 4. at their beginning are assumed to be valid lines for importing part of the part name (**package type**).

Lines in which none of the keywords 1. through 5. is found at the line beginning and in which the first token consists of Sun, Mon, Tue, Wed, Thu, Fri, or Sat, are assumed to be valid lines for the purpose of importing data for **date of creation**.

### 3.4.3 Importing the Design Name

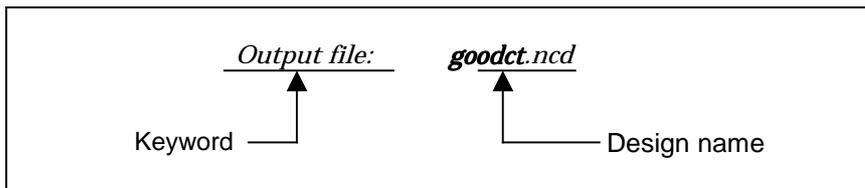
- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

The program retrieves the token found immediately following the keyword “Output file:” from the valid lines for importing design name (refer to Section 3.3.2, “General Program Specifications”). If this token contains a period (.), any characters following the period is removed. The program stores the token in memory as the **design name**,

(If multiple instances of the period (.) are encountered, the last period is considered to the period sought.)

If no period (.) is found in the token, the entire token is directly stored in memory, as is.

If no token follows the keyword “Output file:,” a null character is stored in memory.



The import operation continues until the start-of-pin properties read flag (refer to Section 3.3.2, “General Program Specifications”) is found, or until the last line is reached.

- \* **For multiple instances of the valid line, the first data stored applies, except for null characters.**

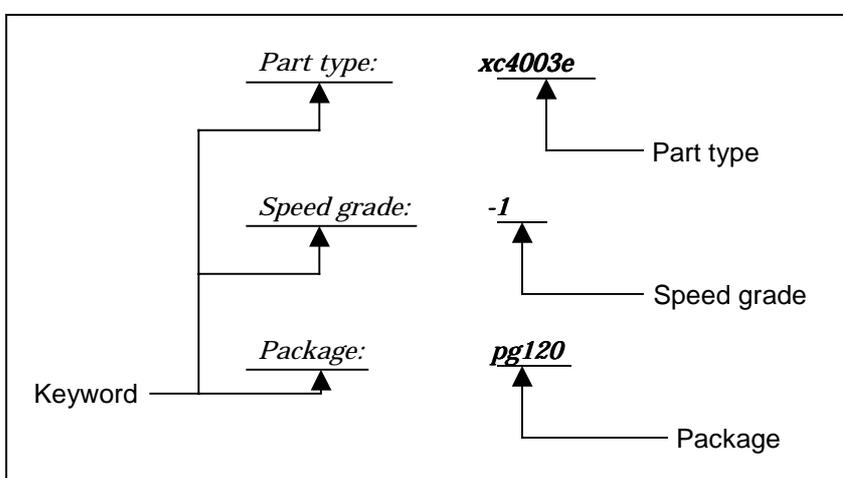
### 3.4.4 Importing the Part Name

- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

The part name is created by importing, then merging the **part type**, **speed grade**, and **package type** from the pin report file.

The tokens found immediately after the keywords “Part type:,” “Speed grade:,” and “Package:” in corresponding valid lines are stored in memory as Part type, speed grade, and package type each.

If no token follows the keyword, a null character is stored in memory.



The **part name** consists of <Part type>, <Speed grade>, <Package>, and <Temperature range>, merged in that order.

- \* The **Temperature range** data needs to be entered from the Component Designer.

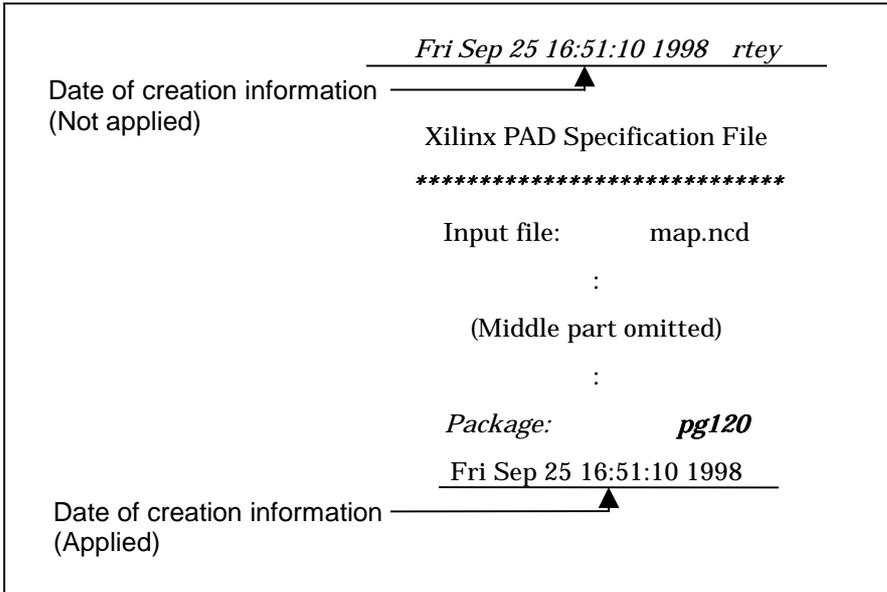
For the example Xilinx pin report file, if the temperature range entered from the Component Designer is “C” (Commercial), the part name consists of “XC4003E-1PG120C.”

The import operation continues until the start-of-pin properties read flag (refer to Section 3.3.2, “General Program Specifications”) is found, or until the last line is reached.

- \* For multiple instances of the valid line, the first data to be stored applies, with the exception of null characters.

### 3.4.5 Importing Data for Date of Creation

The character string in the valid line for importing date of creation data (refer to Section 3.3.2, “General Program Specifications”), with all space, tab, and line-feeds at the beginning and end of the line removed, is stored in memory as the **date of creation**.



The import operation continues until the start-of-pin properties read flag (refer to Section 3.3.2, “General Program Specifications”) is found, or until the last line is reached.

- \* **For multiple instances of the valid line, the last data set applies, rather than the first.** (For all other items, the first data to be stored applies.)

### 3.4.6 Importing Physical Pin Numbers and Logical Pin Names

The program starts searching for pin properties (**physical pin numbers** and **logical pin names**) from the line following the one in which the keyword “Pinout by Pin Number:” is found at the line beginning.

The program first searches for a first line whose first character is “|,” after all space, tab, and line-feeds at the line beginning are removed.

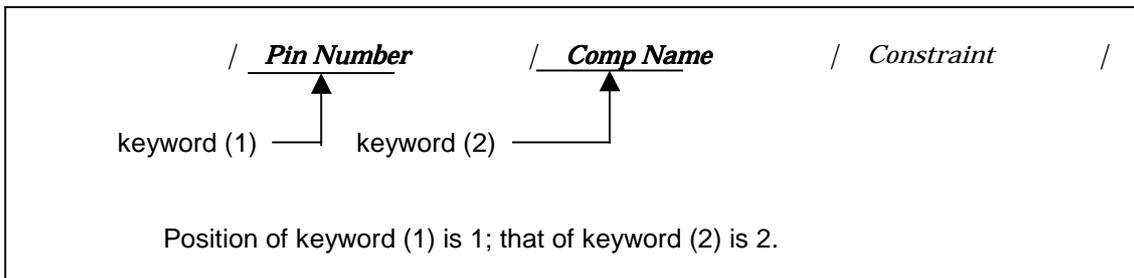
When this line is found, the program divides it into strings by putting “|”s and looks for the following keywords in those strings:

(1) Pin Number

(2) Comp Name

\* Space, tab, and line-feeds at line beginning and line end are ignored.

When keywords are found in a string, the program stores their positions (indicated by the n<sup>th</sup> position of “|”) in memory.



If neither keyword (1) or (2) are found, the program proceeds to the next line and repeats the operation, until it finds keywords (1) and (2).

After storing the positions of keywords (1) and (2) in memory, the program searches subsequent lines for a line having a first character “|”, after all space, tab, and line-feeds at the line beginning are removed. If it encounters such a line, it looks in it for pin properties.

The program divides this line by putting “|”s and retrieves the string created at the position of keyword (1) “Pin Number” by this division as a **physical pin number**, stripping space, tab, and line-feeds at the beginning and end of the string.

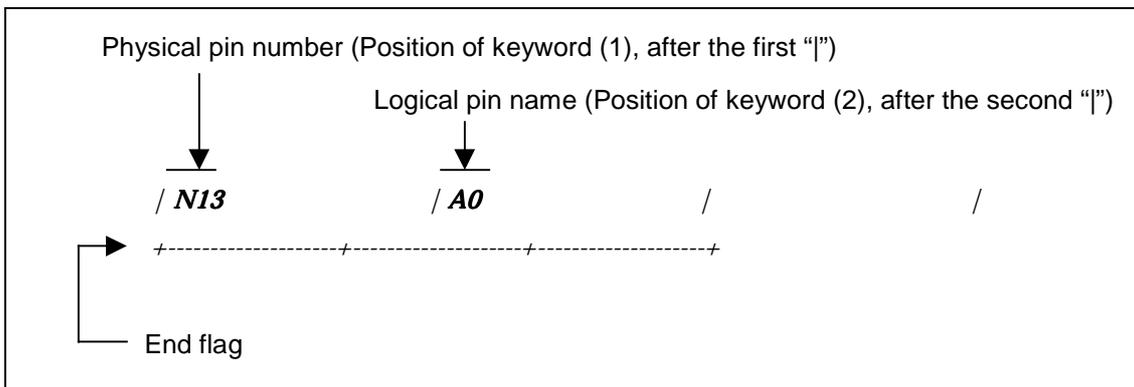
In the same way, the program retrieves the string created at the position of keyword (2) "Comp Name" as a **logical pin name**, stripping space, tab, and line-feeds from the beginning and end of the string.

- \* **When acquiring physical pin numbers or logical pin names, if the data has no positions stored, does not have any stored position, or only contains space, tab, or line-feeds at the stored position, the acquired results consist of null characters.**

Unless the physical pin number or logical pin name consists of null characters, the program imports pin properties from this line, regarding it as properties for a single pin.

The program searches for pin properties from all appropriate lines until arriving at the end flag. The end flag is a line that meets the following conditions.

- **A line, after any one instance of pin properties has been imported, that begins with "+--." (All space, tab, and line-feeds at the beginning of the line are ignored.)**



When the end flag is detected and the program has completed importing pin properties, it outputs the imported pin count to a log file before closing.

## 3.5 Import Specifications for Xilinx Pin Report Files (CPLD Type)

### 3.5.1 Outline

The Xilinx pin report files (CPLD type) are created by the Xilinx's FPGA development tool "M1 TM (Ver.1.3)" after implementation of the tool. The file extension is ".rpt."

\* Both DOS and Unix formats can be read.

This software acquires **design name, part name, physical pin numbers, logical pin names, vendor name, and date of creation** from Xilinx pin report files (CPLD type), then reflects these data in the *Component Designer*.

The default **vendor name** when importing the Xilinx pin report file is "Xilinx."

Since the Xilinx pin report file (CPLD type) lacks **temperature range** data for the **part name**, this data needs to be entered from the Component Designer when importing the pin report file.

For further details, refer to the documentation shown below.

System Designer Online Help "Tool Menu/Component Designer" - "File" - "Import" - "Pin Report File Import" dialog box

Specifications regarding data import for **design name, part name, physical pin numbers, logical pin names, and date of creation** from the Xilinx pin report file (CPLD type) are described in the following pages.

Example: Xilinx pin report file (CPLD type):

```
XACT: version M1.3.7                               Xilinx Inc.
                                                    Fitter Report
  Design Name: goodct
Fitting Status: Successful                          Date: 9-25-98, 5:05PM

***** Resource Summary *****

Design      Device      Macrocells  Product Terms  Pins
:
<Middle part omitted>
:

***** Device Pin Out *****

Device : XC95216-10-PQ160

Pin  Signal      Pin  Signal
No.  Name         No.  Name
  1   VCC         81   VCC
  2   TIE         82   A9
  3   B8          83   TIE
 79   TIE        159  TIE
 80   GND        160  GND

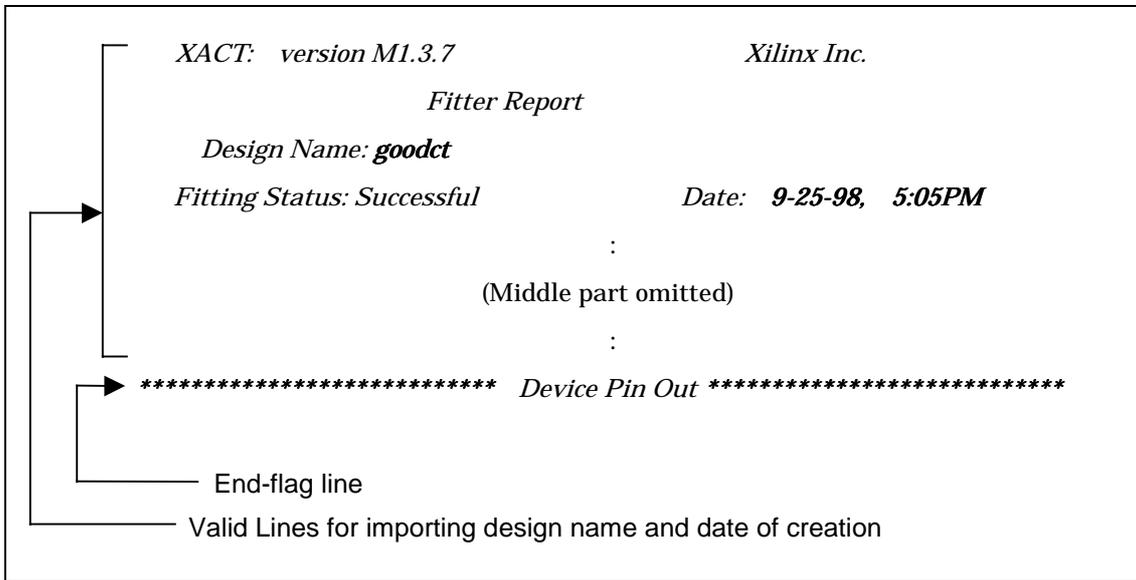
Legend : TIE = Tie pin to GND or board trace driven to valid logic level
        VCC = Dedicated Power Pin
        GND = Dedicated Ground Pin
        TDI = Test Data In, JTAG pin
:
<Rest is omitted>
```

### 3.5.2 General Program Specifications

This program processes each line of the pin report file, starting from the first line.

The program continues to import **design name** and **date of creation** until it finds a line containing the following keyword.

- \* Device Pin Out \*
- \* All space and tabs preceding or following the asterisk "\*" are ignored.
- \* The search is case-sensitive.



When the program finished the data import for design name and date of creation, it begins importing part name and pin properties (physical pin numbers and logical pin names) from the next line (one that follows the end-flag line).

### 3.5.3 Importing the Design Name and data for Date of Creation

- \* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

Starting from the beginning of the valid line for importing design name and date of creation (refer to Section 3.4.2, “General Program Specifications”), the program searches for the following keywords:

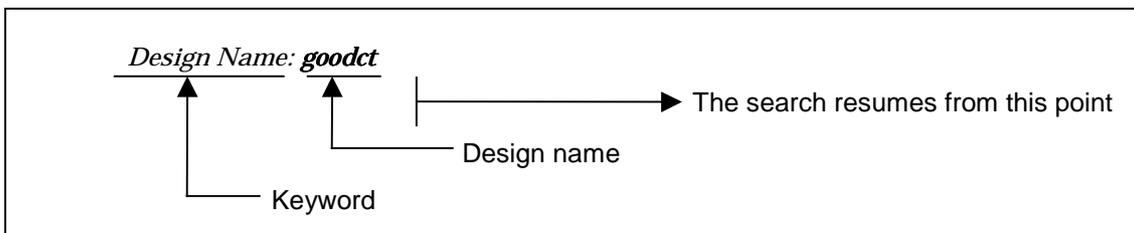
1. Design Name:
2. Date:

- \* Keywords not located at the beginning of the line must be preceded by a space or tab.
- \* All space and tabs preceding the colon (:) are ignored.
- \* The search is case-sensitive.

When keyword 1. is found, the token immediately following is stored in memory as the **design name**.

If no token follows the keyword, a null character is stored in memory.

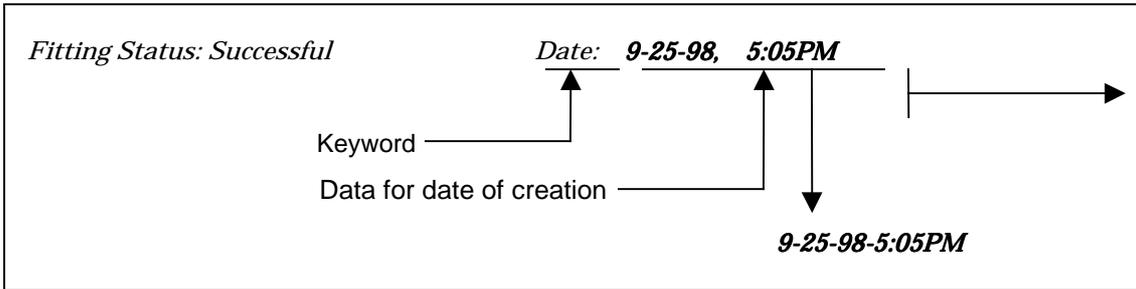
If this token is found, the search continues for keywords 1. and 2.



When keyword 2. is found, the token immediately following is stored in memory as the **date of creation**. If this token ends with a comma “,”, the next token (or null character if nonexistent) is read and merged with the first. The comma “,” is then converted into a hyphen “-”.

If no token follows the keyword, a null character is stored in memory.

The search for keywords 1. and 2 continues after the token is read.



When the end-flag line (refer to Section 3.4.2, “General Program Specifications”) is detected, the program completes its search for the keywords and imports data for design name and date of creation, based on the data stored. Data not stored in memory is processed as null characters.

- \* **If multiple instances of the keyword are found, the first data to be stored applies, with the exception of null characters.**

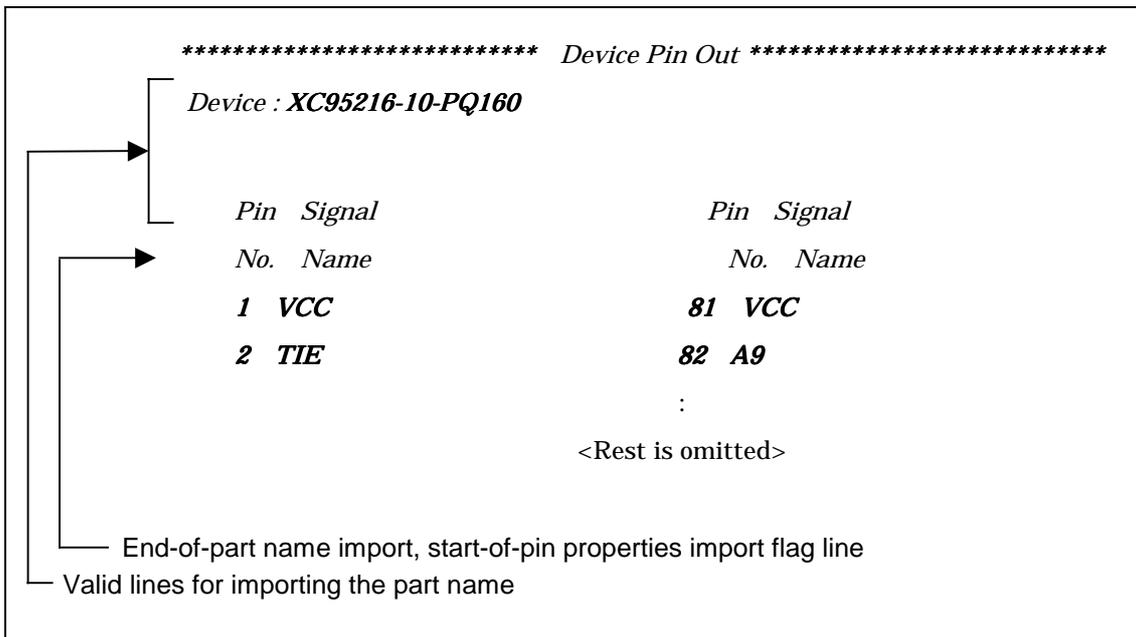
### 3.5.4 Importing the Part Name and Pin Properties

\* **Note:** The word “token” here refers to a character string isolated by a space, tab, or line-feed.

The program imports the part name and pin properties (physical pin numbers and logical pin names), starting with the line following the end-of-import flag line for design name and date of creation data (refer to Section 3.4.2, “General Program Specifications”).

The program continues importing the part name until it finds a line in which tokens occur in the following order:

1. No.
2. Name
3. No.
4. Name



The program scans the beginning of each valid line for importing part name for the following keyword:

- Device:
  - \* All space and tabs preceding the keyword and the colon (:) are ignored.
  - \* The search is case-sensitive.

When a line containing the keyword is found, the token immediately following is stored in memory as the **part name**.

If no token follows the keyword, a null character is stored in memory.

Once the keyword is found in a valid line, the program completes its search from this line.

When it detects the end-of-part name import flag line, the program ends its keyword search and imports the data stored as the part name, after appending <Temperature range> to the end of the data.

If no token is stored in memory, the data is processed as null characters.

- \* The temperature range needs to be entered from the Component Designer.

For the example Xilinx pin report file, if the temperature range entered from the Component Designer is "C" (Commercial), the part name consists of "XC95216-10-PQ160C."

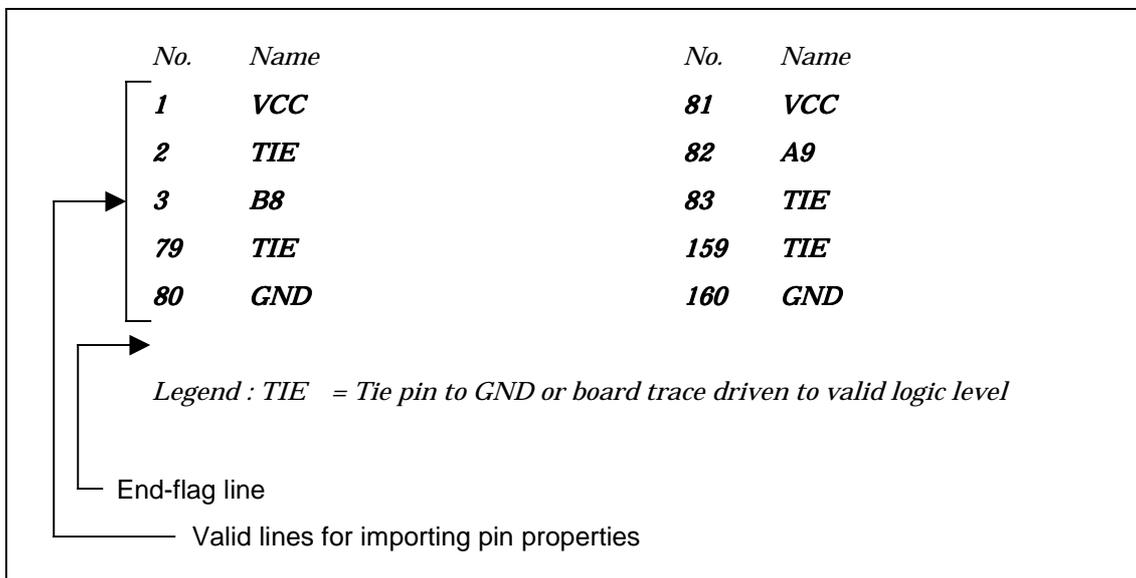
- \* **If multiple instances of the keyword are encountered, the first token stored is applied as the part name, except for null characters.**

Starting with the line next to the end-of-part name import flag line, the program continues importing pin properties (physical pin numbers and logical pin names) until it finds a line that meets the following conditions:

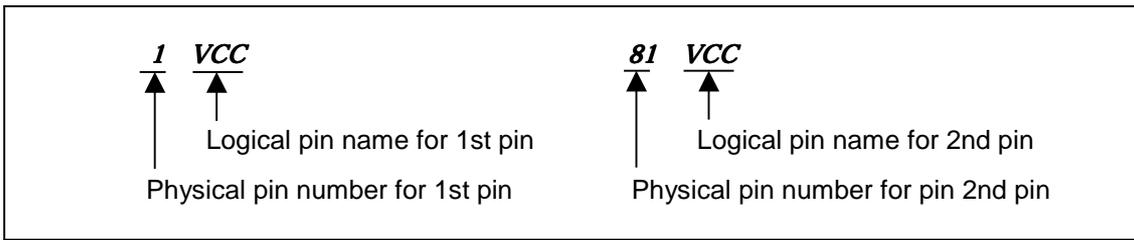
1. A line composed only of space, tab, and line-feeds
2. A line in which the following keyword is found at the beginning

■ Legend:

- \* All space and tabs preceding the keyword and the colon (:) are ignored.
- \* The search is case-sensitive.



From the valid lines for importing pin properties, the program imports pin properties for up to two pins. When importing these properties, the program retrieves the first token as the physical pin number for 1st pin.  
the second token as the logical pin name for 1st pin;  
the third token as the physical pin number for 2nd pin;  
the fourth token as the logical pin name for 2nd pin .



- \* A token must always be present at the position at which a physical pin number or logical pin name is expected. Even an undefined physical pin number or logical pin name must be assigned a temporary number or name.
- \* If the physical pin number or logical pin name contains a space or tab, the program may not be able to import the pin properties correctly.

On reaching the final line, the program outputs the imported pin count to a log file before closing.

## 4 ERROR AND WARNING MESSAGES

This chapter explains various messages output by the software.

Messages are one of two types: for errors, or for warnings. When an error message is output, the file is not imported. When a warning message is output, the file is imported, but the software issues an alert.

Error and warning messages encountered during pin report file imports are described separately below for each file type.

### 4.1 When Importing Actel Pin Report Files

#### Error messages

**ERROR:** **Format error. Keyword “Pin Report” is not found.**

**Cause:** The keyword “Pin Report” can’t be found after searching the pin report file all the way to the last line.

**Solution:** Refer to [Section 3.1.2, “General Program Specifications.”](#)

#### Warning messages

**WARNING:** **Pin properties do not exist. Pin properties are not reflected**

**Cause:** Because the end-of-import flag line is not found in the pin report file during the import of the design name, part name, and data for date of creation, the program can’t perform pin properties import operation.

**Solution:** Refer to [Section 3.1.2, “General Program Specifications.”](#)

**WARNING:** **When try to import Design Name, Keyword “Design Name” is not found. Design name is not reflected in property.**

**Cause:** The keyword “Design Name:” can’t be found in the pin report file.

**Solution:** Refer to [Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”](#)

**WARNING:** **Design Name does not exist. Design name is not reflected in property.**

**Cause:** Although the keyword “Design Name:” exists in the pin report file, the relevant design name can’t be found.

**Solution:** Refer to [Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”](#)

**WARNING: When try to import Part Name, Keyword “Die” is not found. Part name is not reflected correctly in property.**

**Cause:** The keyword “Die:” can’t be found in the pin report file.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: Device Name does not exist. Part name is not reflected correctly in property.**

**Cause:** Although the keyword “Die:” exists in the pin report file, the relevant data for device name can’t be found.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: When try to import Part Name, Keyword “Package” is not found. Part name is not reflected correctly in property.**

**Cause:** The keyword “Package:” can’t be found in the pin report file.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: The data for number of pins does not exist. Part Name is not reflected correctly in property.**

**Cause:** Although the keyword “Package:” exists in the pin report file, the relevant data for number of pins can’t be found.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: Package Type does not exist. Part Name is not reflected correctly in property.**

**Cause:** Although the keyword “Package:” exists in the pin report file, the relevant data for package type can’t be found.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: Speed Grade does not exist. Part Name is not reflected correctly in property.**

**Cause:** The data for speed grade has not been entered from the Component Designer.

**Solution:** Refer to System Designer Online Help “Tool Menu/Component Designer” - “File” - “Import” - “Pin Report File Import” dialog box.

**WARNING: Temperature Range does not exist. Part Name is not reflected correctly in property.**

**Cause:** The data for temperature range has not been entered from the Component Designer.

**Solution:** Refer to System Designer Online Help “Tool Menu/Component Designer” - “File” - “Import” - “Pin Report File Import” dialog box.

**WARNING: Part Name does not exist. Part Name is not reflected in property.**

**Cause:** The data for device name, number of pins, and package type that comprise the part name all can't be found.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: When try to import the data for date of creation, Keyword “Date” is not found. The data for date of creation is not reflected in property.**

**Cause:** The keyword “Date:” can't be found in the pin report file.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: The data for date of creation does not exist. The data for date of creation is not reflected in property.**

**Cause:** Although the keyword “Date:” exists in the pin report file, the relevant data for date of creation can't be found.

**Solution:** Refer to Section 3.1.3, “Importing the Design Name, Part Name, and data for Date of Creation.”

**WARNING: When try to import Pin properties, Keyword “Number” or “Name” is not found. Pin properties are not reflected.**

**Cause:** The keyword “Number” or “Name” can't be found in the pin report file.

**Solution:** Refer to Section 3.1.4, “Importing Physical Pin Numbers and Logical Pin Names.”

**WARNING: Physical Pin Number XX does not have Logical Pin Number.**

**Cause:** The logical pin name corresponding to physical pin number XX can't be found in the pin report file.

**Solution:** Refer to Section 3.1.4, “Importing Physical Pin Numbers and Logical Pin Names.”

## 4.2 When Importing Altera Pin Report Files

### Error messages

**ERROR:** **Format error. Keyword “CHIP” is not found.**

**Cause:** The keyword “CHIP” can’t be found after searching the pin report file all the way to the last line.

**Solution:** Refer to [Section 3.2.2, “General Program Specifications.”](#)

### Warning messages

**WARNING:** **Keyword “CHIP” exist plural. The imported data may be not correct.**

**Cause:** Multiple instances of keyword “CHIP” exist. The first instance of the keyword found is considered valid.

**Solution:** Refer to [Section 3.2.2, “General Program Specifications.”](#)

**WARNING:** **Design Name does not exist. Design Number is not reflected in property.**

**Cause:** No character string enclosed in double quotes (") is found after the keyword “CHIP.”

**Solution:** Refer to [Section 3.2.3, “Importing the Design Name and Part Name.”](#)

**WARNING:** **Part Name does not exist. Part Name is not reflected in property.**

**Cause:** No token is found after the keyword “CHIP.” Or no token follows the token containing the design name.

**Solution:** Refer to [Section 3.2.3, “Importing the Design Name and Part Name.”](#)

**WARNING:** **The Separate word “:” exist plural. The imported pin properties may be not correct.**

**Cause:** The valid line for importing pin properties contains multiple instances of delimiter (:).

**Solution:** Refer to [Section 3.2.3, “Importing the Design Name and Part Name.”](#)

**WARNING:** **Wrong line exist. This line is not imported.**

**Cause:** Although the valid line for importing pin properties contains the delimiter (:), no physical pin numbers and logical pin names are found.

**Solution:** Refer to [Section 3.2.3, “Importing the Design Name and Part Name.”](#)

**WARNING: Logical Pin Name XX does not have Physical Pin Number.**

**Cause:** The physical pin number corresponding to logical pin name XX can't be found in the pin report file.

**Solution:** Refer to Section 3.2.3, "Importing the Design Name and Part Name."

**WARNING: Physical Pin Number XX does not have Logical Pin Name.**

**Cause:** The logical pin name corresponding to physical pin number XX can't be found in the pin report file.

**Solution:** Refer to Section 3.2.3, "Importing the Design Name and Part Name."

## 4.3 When Importing Lattice Pin Report Files

### Warning messages

**WARNING:(line:%d)Duplicate keyword "Target Device" exist. "Target Device" data found first in the pin report file is imported as Part Name.**

**Cause:** In LATTICE pin report files, multiple instances of the keyword "Targeted Device" to import as data on date of creation exist. The first instance of "Targeted Device" is imported as the property "Part Name".

**Solution:** Refer to "3.3.3 Importing the design name, part name, date of creation, and vendor name".

**WARNING: Keyword "Target Device" is not found. Part Name is not imported.**

**Cause:** In LATTICE pin report files, no keyword "Targeted Device" to import as a part name is found. The property "Part Name" is not imported.

**Solution:** Refer to "3.3.3 Importing the design name, part name, date of creation, and vendor name".

**WARNING: (line:%d)Duplicate keyword "Date" exist. "Date" data found first in the pin report file is imported as the date of creation.**

**Cause:** In LATTICE pin report files, multiple instances of the keyword "Date" to import as data on date of creation exist. The first instance of "Date" is imported as the part property "asicDesignDate".

**Solution:** Refer to "3.3.3 Importing the design name, part name, date of creation, and vendor name".

**WARNING: "Date" is not found. The data for date of creation is not imported.**

**Cause:** In LATTICE pin report files, no keyword "Date" to import as data on date of creation is found. The part property "asicDesignDate" is not imported.

**Solution:** Refer to "3.3.3 Importing the design name, part name, date of creation, and vendor name".

**WARNING: (line:%d) There are too many pin property. Check the pin report file.**

**Cause:** In LATTICE pin report files, more than three pin properties exist for one pin.

**Solution:** Refer to "3.3.4 Importing the logical pin name, I/O property, and physical pin numbers".

**WARNING: (line:%d) Physical pin label of the pin (pinID:%d) is empty.**

**Cause:** The value is not set for the pin name.

**Solution:** Refer to "3.3.4 Importing the logical pin name, I/O property, and physical pin numbers".

**WARNING: (line:%d) IO property of the pin (pinID:%d) is empty.**

**Cause:** The value is not set for the I/O property.

**Solution:** Refer to "3.3.4 Importing the logical pin name, I/O property, and physical pin numbers".

**WARNING: (line:%d) Physical pin number of the pin (pinID:%d) is empty.**

**Cause:** The value is not set for the pin number.

**Solution:** Refer to "3.3.4 Importing the logical pin name, I/O property, and physical pin numbers".

## 4.4 When Importing Xilinx Pin Report Files (FPGA Type)

### Error messages

**ERROR:** **Format Error.**

**Cause:** None of the keywords “Output file:,” “Part type:,” “Speed grade:,” “Package:,” or “Pinout by Pin Number:” is found when searching the pin report file all the way to the last line.

**Solution:** Refer to [Section 3.3.2, “General Program Specifications.”](#)

### Warning messages

**WARNING:** **When try to import Pin properties, Keyword “Pinout by Pin Number” is not found. Pin properties is are not reflected.**

**Cause:** The keyword “Pinout by Pin Number:” can’t be found in the pin report file.

**Solution:** Refer to [Section 3.3.2, “General Program Specifications.”](#)

**WARNING:** **When try to import Design Name, Keyword “Output file” is not found. Design name is not reflected in property.**

**Cause:** The keyword “Output file:” can’t be found in the pin report file.

**Solution:** Refer to [Section 3.3.3, “Importing the Design Name.”](#)

**WARNING:** **Design Name does not exist. Design Name is not reflected in property.**

**Cause:** Although the keyword “Output file:” exists in the pin report file, design name is not found.

**Solution:** Refer to [Section 3.3.3, “Importing the Design Name.”](#)

**WARNING: When try to import Part Name, Keyword “Part type” is not found. Part Name is not reflected correctly in property.**

**Cause:** The keyword “Part type:” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: Part Type does not exist. Part Name is not reflected correctly in property.**

**Cause:** Although the keyword “Part type:” exists in the pin report file, the data for part type is not found.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: When try to import Part Name, Keyword “Speed grade” is not found. Part Name is not reflected correctly in property.**

**Cause:** The keyword “Speed grade:” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: Speed Grade does not exist. Part Name is not reflected correctly in property.**

**Cause:** Although the keyword “Speed grade:” exists in the pin report file, the data for speed grade is not found.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: When try to import Part Name, Keyword “Package” is not found. Part Name is not reflected correctly in property.**

**Cause:** The keyword “Package:” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: Package does not exist. Part Name is not reflected correctly in property.**

**Cause:** Although the keyword “Package:” exists in the pin report file, the data for package is not found.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: Temperature Range does not exist. Part Name is not reflected correctly in property.**

**Cause:** The data for temperature range has not been entered from the Component Designer.

**Solution:** Refer to System Designer Online Help “Tool Menu/Component Designer” - “File” - “Import” - “Pin Report File Import” dialog box.

**WARNING: Part Name does not exist. Part Name is not reflected in property.**

**Cause:** None of the data for part type, speed grade, and package is found.

**Solution:** Refer to Section 3.3.4, “Importing the Part Name.”

**WARNING: The data for date of creation does not exist. The data for date of creation is not reflected in property**

**Cause:** The data for the date of creation is not found in the pin report file.

**Solution:** Refer to Section 3.3.5, “Importing Data for Date of Creation.”

**WARNING: When try to import Pin properties, Keyword “Pin Number” is not found. Physical Pin Number is not reflected in property.**

**Cause:** The keyword “Pin Number” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.6, “Importing Physical Pin Numbers and Logical Pin Names.”

**WARNING: When try to import Pin properties, Keyword “Comp Name” is not found. Logical Pin Name is not reflected in property**

**Cause:** The keyword “Comp Name” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.6, “Importing Physical Pin Numbers and Logical Pin Names.”

**WARNING: When try to import Pin properties, Keywords “Pin Number” or “Comp Name” is not found. Pin properties are not reflected.**

**Cause:** The keywords “Pin Number” and “Comp Name” can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.6, “Importing Physical Pin Numbers and Logical Pin Names.”

**WARNING: Logical Pin Name XX does not have Physical Number.**

**Cause:** The physical pin number corresponding to logical pin name XX can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.6, “Importing Physical Pin Numbers and Logical Pin Names.”

**WARNING: Physical Pin Number XX does not have Logical Pin Name.**

**Cause:** The logical pin name corresponding to physical pin number XX can’t be found in the pin report file.

**Solution:** Refer to Section 3.3.6, “Importing Physical Pin Numbers and Logical Pin Names.”

## 4.5 When Importing Xilinx Pin Report Files (CPLD Type)

### Error messages

**ERROR:** **Format Error. Keyword "Device Pin Out" is not found.**

**Cause:** The keyword "\* Device Pin Out \*" can't be found by searching the pin report file all the way to the last line.

**Solution:** Refer to Section 3.4.2, "General Program Specifications."

### Warning messages

**WARNING:** **When try to import Design Name, Keyword "Design Name" is not found. Design Name is not reflected in property**

**Cause:** The keyword "Design Name:" can't be found in the pin report file.

**Solution:** Refer to Section 3.4.3, "Importing the Design Name and data for Date of Creation"

**WARNING:** **Design Name does not exist. Design Name is not reflected in property**

**Cause:** Although the keyword "Design Name:" exists in the pin report file, design name is not found.

**Solution:** Refer to Section 3.4.3, "Importing the Design Name and data for Date of Creation"

**WARNING:** **When try to import the data for date of creation, Keyword "Date" is not found. The data for date of creation is not reflected in property**

**Cause:** The keyword "Date:" can't be found in the pin report file.

**Solution:** Refer to Section 3.4.3, "Importing the Design Name and data for Date of Creation"

**WARNING: The data for date of creation does not exist. The date for date of creation is not reflected in property.**

**Cause:** Although the keyword "Date:" exists in the pin report file, data for date of creation is not found.

**Solution:** Refer to Section 3.4.3, "Importing the Design Name and data for Date of Creation"

**WARNING: Pin properties do not found. Pin properties are not reflected**

**Cause:** No lines are found in which the tokens "No.", "Name," "No.," and "Name" are written in that order.

**Solution:** Refer to Section 3.4.4, "Importing the Part Name and Pin Properties."

**WARNING: When try to import Part Name, Keyword "Device" is not found. Part Name is not reflected in property**

**Cause:** Keyword "Device" can't be found in teh pin report file.

**Solution:** Refer to Section 3.4.4, "Importing the Part Name and Pin Properties."

**WARNING: Part Name does not exist. Part Name is not reflected in property.**

**Cause:** Although the keyword "Device:" exists in the pin report file, no part name is found.

**Solution:** Refer to Section 3.4.4, "Importing the Part Name and Pin Properties."

**WARNING: Temperature Range does not exist. Part Name is not reflected correctly in property**

**Cause:** The temperature range has not been entered from the **Component Designer**.

**Solution:** Refer to **System Designer Online Help "Tool Menu/Component Designer"** - "File" - "Import" - "Pin Report File Import" dialog box.

**WARNING: The number of Pin properties of a line is more than 5. 5th data and more are not objects of import.**

**Cause:** The valid line for importing pin properties includes more than 5 tokens.

**Solution:** Refer to Section 3.4.4, "Importing the Part Name and Pin Properties."

**WARNING: Physical Pin Number XX does not have Logical Pin Name.**

**Cause:** The logical pin name corresponding to physical pin number XX can't be found in the pin report file.

**Solution:** Refer to Section 3.4.4, "Importing the Part Name and Pin Properties."