CR-5000

SD^{+} Component Designer

Contents

Introduction: Component Designer Overview

With GUI function in the Component Designer tool, you can import component information that is defined in various formats and output (export) the information after editing, checking and converting it into any formats.

<Component Designer functions>

1.Import function

You can import information from the following files.

-LCDB
-Symbol
-CSV format file (SBS)
-Hardware description language (Verilog-HDL, VHDL)
-Pin report file (ACTEl, ALTERA, LATTICE, XILIMX)
-Part and Symbol on schematic sheet

2.Editing function

You can edit the following information in the Component

-Components
-Pins
-Symbol properties
-Parts properties

3.Consistency check

You can check the following.

* Data check in a component.

-Duplication of pin name and pin number
-Definition of required properties and pin data.
-Latest update of internal local symbol file.

*Comparison with external file (LCDB)

-Equality of pin number.
-Equal correspondence of properties for pin number or pin mane.

4.Export function

You can export information to the following files.

-LCDB
-Symbol
-CSV format file (SBS)
-Hardware description language (Verilog-HDL, VHDL)
-Pin restriction file (ACTEL, ALTERA, XILINX, LATTICE)
-Part Library ASCII file (.cdf)
-FDL file (VISULA)

NOTE

Processing contents vary according to the "Editing Mode" in the Component Designer.
Refer to P.5-3.

<Main operation items>

1.Importing external data (Option)

Take in external data as required and convert it into component information to edit in the Component Designer.

2.Composing a component

Recompose parts information based on component unit. Component composition is represented as component tree that consists of package component and sub-components.
If you want to divide symbols, change the editing mode to "Split Component Mode".

3.Defining components information.

Define component information such as pin information for each component.

4.Defining symbol / parts properties

Define properties for symbols or parts.

5.Confirming consistency

Check if your edited components information is correct.

6.Editing / auto-generating symbols

Auto-generate a symbol that matches your edited component or edit a symbol shape.

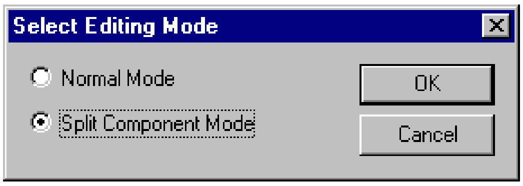
7.Outputting symbol file and LCDB. Outputting external data. (Option)

Output edited information as symbol file or LDCB data. Output edited information as other external data if necessary.

[Split Component Mode]

With the Component Designer, you can split a component into peaces and place them on a schematic data. To do it, you need to change the editing mode to [Split Component Mode]

<How to specify>
[Environment] - [Editing Mode] ---


Select Editing Mode

○ Normal Mode
● Split Component Mode

Select the mode in the dialog and click <<OK>>

The following is ordinal process in the split component mode.

1.Importing external data (Option)

Import external data as required and convert it into component information to edit in the Component Designer.
You can also import symbol information as one of split components.

2.Composing a component

Represent component information as component tree that consists of package component and split components. Only package component is required on this point.

3.Defining component information.

Define component information such as pin information for each component.
To create split components, classify the pins into groups.

4. Composing split components.

Synthesize split components based on the groups defined in the pin definition table.
(Select [Component] - [(Re) generate Package Component])

5.Defining symbol / part properties

Defining symbol / part properties

6.Confirming consistency

Check if your edited component information is correct.
You can also check consistency with LCDB.

7.Auto-generating / editing symbols

Auto-generate split symbols. You can also edit symbol shape as required.

8.Outputting symbol file

Output your created split symbols as symbol file.

Reference

For details, see split mark in each chapter. Split

1.Importing from external data.

You can retrieve information from the data of the System Designer or other tools into the Component Designer using the import wizard. Also retrieve information directly into the Component Designer without using the wizard by specifying the component or symbol on the Schematic Sheet.

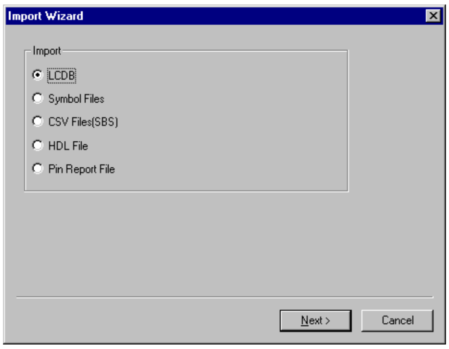
Importing data by means of wizard

Import information into the Component Designer from the System Designer or other tools by means of the import wizard.

<Data to be imported>

You can import information from the following data.

-LCDB
-Symbol
-CSV format file (SBS)
-Hardware description language
(Verilog-HDL, VHDL)
-Pin report file(ACTEL, ALTERA, LATTICE, XILINX)



Import Wizard

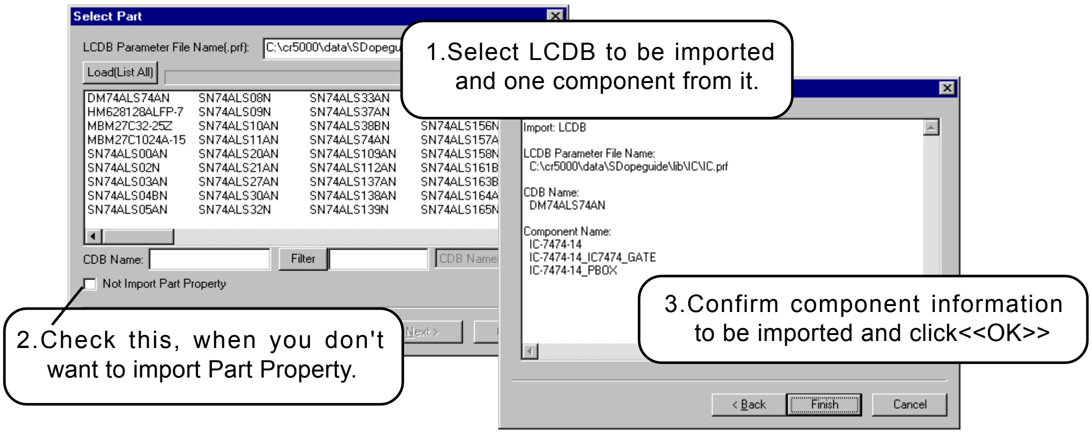
Import
● LCDB
○ Symbol Files
○ CSV Files(SBS)
○ HDL File
○ Pin Report File

<Operation process>

[File] - [Import]
Specify file type to be imported in the Wizard.

1.Importing from LCDB

Select one component from specified LCDB and import the information into the ComponentDesigner.



Select Part

LCDB Parameter File Name(.prf): C:\cr5000\data\SDopegu

```
DM74ALS74AN        SN74ALS08N    SN74ALS33AN
HM628128ALFP-7     SN74ALS09N    SN74ALS37AN
MBM27C32-25Z       SN74ALS10AN   SN74ALS38BN    SN74ALS156N
MBM27C1024A-15     SN74ALS11AN   SN74ALS74AN    SN74ALS157A
SN74ALS00AN        SN74ALS20AN   SN74ALS109AN   SN74ALS158N
SN74ALS02N         SN74ALS21AN   SN74ALS112AN   SN74ALS161B
SN74ALS03AN        SN74ALS27AN   SN74ALS137AN   SN74ALS163B
SN74ALS04BN        SN74ALS30AN   SN74ALS138AN   SN74ALS164A
SN74ALS05AN        SN74ALS32N    SN74ALS139N    SN74ALS165N
```

CDB Name:
□ Not Import Part Property

```
Import: LCDB

LCDB Parameter File Name:
  C:\cr5000\data\SDopeguide\lib\IC\IC.prf

CDB Name:
  DM74ALS74AN

Component Name:
  IC-7474-14
  IC-7474-14_IC7474_GATE
  IC-7474-14_PBOX
```

1.Select LCDB to be imported and one component from it.

2.Check this, when you don't want to import Part Property.

3.Confirm component information to be imported and click<<OK>>

<Contents of process>

The following information is displayed in each display area that composes the ComponentDesigner

[Component tree] -Places package component at the top of the tree and imports other gates as sub-components. Imports CDB name and part name.

[Symbol viewer] -Displays symbol sheets registered in LCDB.

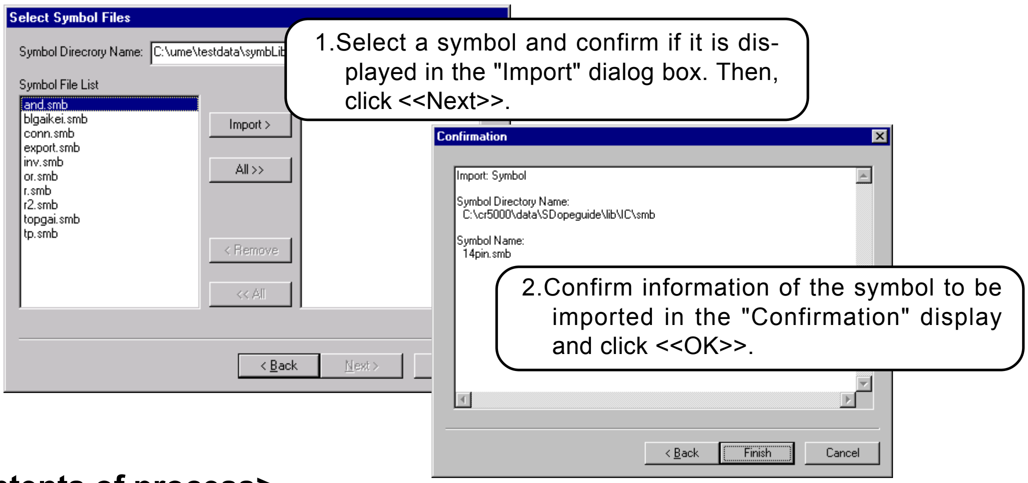
[Component] -Imports pin information or gate number for each component.

[Symbol property] -Imports property information from symbol sheets registered in LCDB.

[Parts property] -Imports property information registered in LCDB component information.

2.Importing from symbols

Import specified symbol file information to the Component Designer.



Symbol Direcroy Name: C:\ume\testdata\symbLib

and.smb
blgaikei.smb
conn.smb
export.smb
inv.smb
or.smb
r.smb
r2.smb
topgai.smb
tp.smb

Import: Symbol

Symbol Directory Name:
C:\cr5000\data\SDopeguide\lib\IC\smb

Symbol Name:
14pin.smb

1.Select a symbol and confirm if it is displayed in the "Import" dialog box. Then, click <<Next>>.

2.Confirm information of the symbol to be imported in the "Confirmation" display and click <<OK>>.

<Contents of process>

The following information is displayed in each area that composes the Component Designer.

[Component tree]

*When symbol component type is "Part"

-A symbol is imported as a package component as it is.

*When symbol component type is "Gate"

Vacant package named "part_package" is created and all symbols are imported as sub-components.

When the editing mode is "Normal mode"

A symbol is imported as a component that has component type.
Component type that was set in a symbol sheet property is used here.

When the editing mode is "Split Component mode"

A symbol is imported as "Split component".

Split

CAUTION

When component type is "Gate", symbol is imported as a sub-component and no property for package is created.

[Symbol viewer]

Package

-Symbol name "part_package.smb" is given.(Symbol shape is not displayed)

Sub-component

Vacant package named "part_package" is created and all symbols are imported as sub-components.

[Component] **-** Component type, function type, pin information, etc. are imported.

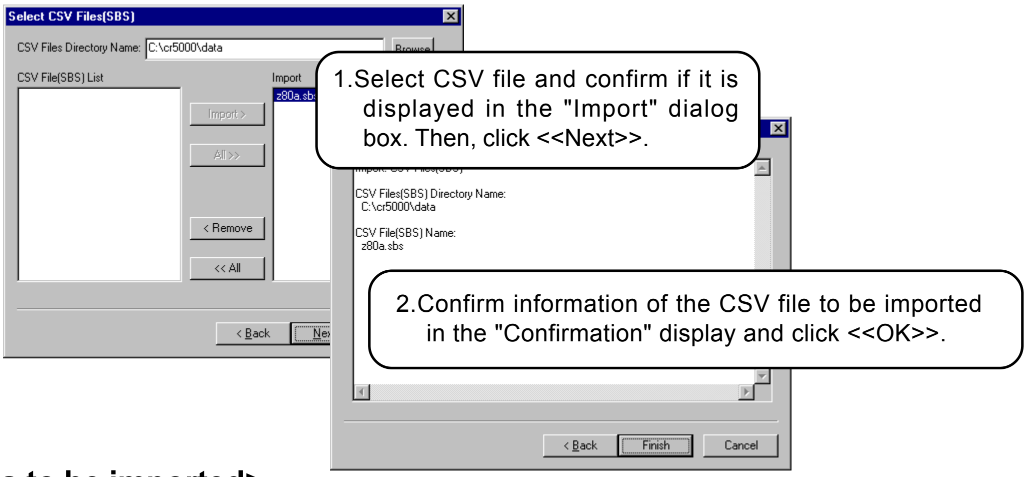
Component name	symbol name is imported.
Component type	"Component type" of sheet property on imported symbol sheet is imported. It is displayed only when the editing mode is "Normal mode".
Function type	"Function type" of sheet property on imported symbol sheet is imported.
Number of pins, Pin No.	Symbol pin property is imported.

[Symbol property] - Sheet property of symbol sheet is imported.

[Parts property] - This is not imported.

3.Importing from CSV file (SBS)

Import information of CSV file created in prescribed format to the Component Designer.



Select CSV Files(SBS)

CSV Files Directory Name: C:\cr5000\data

CSV File(SBS) List

Import: z80a.sbs

CSV Files(SBS) Directory Name:
C:\cr5000\data

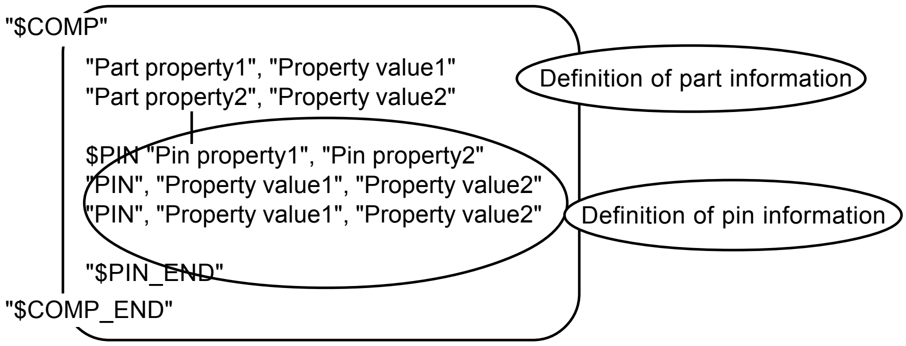
CSV File(SBS) Name:
z80a.sbs

1.Select CSV file and confirm if it is displayed in the "Import" dialog box. Then, click <<Next>>.

2.Confirm information of the CSV file to be imported in the "Confirmation" display and click <<OK>>.

<Files to be imported>

The following are CSV file formats to be imported.



```
"$COMP"
    "Part property1", "Property value1"
    "Part property2", "Property value2"

    $PIN "Pin property1", "Pin property2"
    "PIN", "Property value1", "Property value2"
    "PIN", "Property value1", "Property value2"

    "$PIN_END"
"$COMP_END"
```

Definition of part information

Definition of pin information

CSV file sample

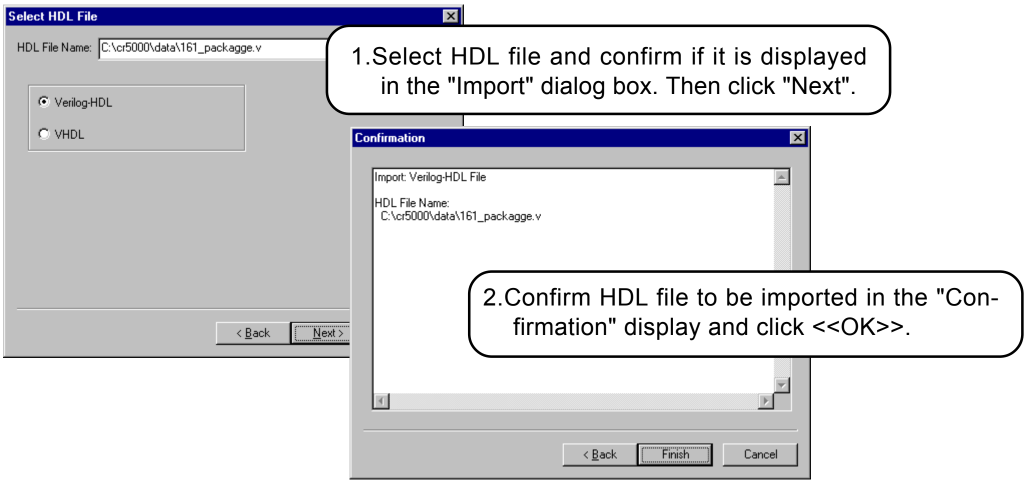
```
"$COMP"
"sbsSymbolOrigin","UpperLeft"
"isvGate","NO"
"sbsPinIdentificationProperty","sbsPinId"
"sbsRegerateShape","YES"
"sbsPinArrangement","QUAD"
"cdbName","74LS04-DIP"
"componentType","gate"
"partName","74ALS04-DIP"
"useWithSchema","YES"
"function","INVERTER"
"componentName","04_INVERTER"

"$PIN","sbsPinId","bitCount","pinNumber","cirPinLabel","pinLabel","io","sbsPinPosition"
"pin","1","1","1,3,5,9,11,13","","A","INPUT","LEFT"
"pin","2","1","2,4,6,8,10,12","","Y","OUTPUT","RIGHT"
"pin","3","1","14,14,14,14,14,14","","VCC","VCC","TOP"
"pin","4","1","7,7,7,7,7,7","","GND","GND","BOTTOM"
"$PIN_END"

"$COMP_END"
```

<Contents of process>

Part/Pin properties defined in CSV are imported to corresponding items in the ComponentDesigner.

4.Importing from hardware description language (Verilog-HDL, VHDL)

Import information of specified hardware description language (Verilog-HDL) to the ComponentDesigner.



Select HDL File

HDL File Name: C:\cr5000\data\161_packagge.v

● Verilog-HDL
○ VHDL

Confirmation

Import: Verilog-HDL File

HDL File Name:
C:\cr5000\data\161_packagge.v

1.Select HDL file and confirm if it is displayed in the "Import" dialog box. Then click "Next".

2.Confirm HDL file to be imported in the "Confirmation" display and click <<OK>>.

<Contents of process>

The following information is displayed in each display area that composes the ComponentDesigner.

[Component] Input/Output port name is imported to pin property "User label".
Input/Output port mode is imported to pin property "IO property".
Input/Output port vector width is imported to pin property "Bus width".

[Parts property] Module name of Verilog-HDL design and entity name of VHDL design are imported to "Entity name"

<Internal processing when importing from Verilog-HDL>

The following process is done when importing from Verilog-HDL.

1.Manipulating text macro
2.Importing module names
3.Manipulating port statement.

1.Manipulating text macro

When importing bit range from port statement later and if MSB or LSB of the port has been replaced with text macro, return its value to default one. Memorize relation of the default text macro for future reference.

<e.g..>

Verilog-HDL design:	Internal memory:	
	[Text macro name]	[Macro text]
'define length 16	length	16
'define REG 2'b01	REG	2'b01
:		

2.Importing module name

Import design module name as parts property "Entity name" of "Component Designer". Remove "\" and space from module name before import.

<e.g..>

Verilog-HDL design:	Component Designer:
	[Entity name]
Module goodct (-----)	
:	

3.Manipulating port statement

Import port name, mode and pit range in the port statement to user label, IO property and bus width that are pin properties of the Component Designer.

-Basically, port name is imported to user label as it is. However, if a port name has "\" or space, remove them before import. If bit range is specified in a port name, the bit range is added to the end of the port name for import. At this point, consistency check with port list is performed. Template with array notation to be added to user label can be specified from System Designer resource file (landata.rsc).

-For IO property, "INPUT", "OUTPUT" and "BIDIRECT" is imported when each mode is "input", "output" and "inout"

-For bus width, bit number calculated from bit range is imported. If text macro is used for MSB or LSB of bit range, it is replaced with value referring to stored text macro setting.

Verilog-HDL design:		Internal memory:		
		[User label]	[IO Property]	[Bus width]
output [15:0]	\Y-OUT ;	Y-OUT [15:0]	OUTPUT	16
input	D, UPC;	UPC	INPUT	1
inout [' length:1]	X_BID;	X_BID [16:1]	BIDIRECT	16
:				

<Internal processing when importing from VHDL>

The following process is done when importing form VHDL.

1.Importing entity name
2.Manipulating port statement

1.Importing entity name

Import design entity name as parts property "Entity name" of "Component Designer".
If module name is put between "\" and space, remove them before import.

<e.g..>

VHDL design:	Component Designer:
	[Entity name]
Module goodct is	goodct
:	

2.Manipulating port statement

Import port name, mode and pit range in the port statement to user label, IO property and bus width that are pin properties of the Component Designer.

-Basically, port name is imported to user label as it is. However, if a port name has "\" or space, remove them before import. If bit range is specified in port name, the bit range is added to the end of the port name for import. At this point, consistency check with port list is performed. Template with array notation to be added to user label can be specified from System Designer resource file (landata.rsc).

-For IO property, "INPUT", "OUTPUT" and "BIDIRECT" is imported when each mode is "input", "output" and "inout".

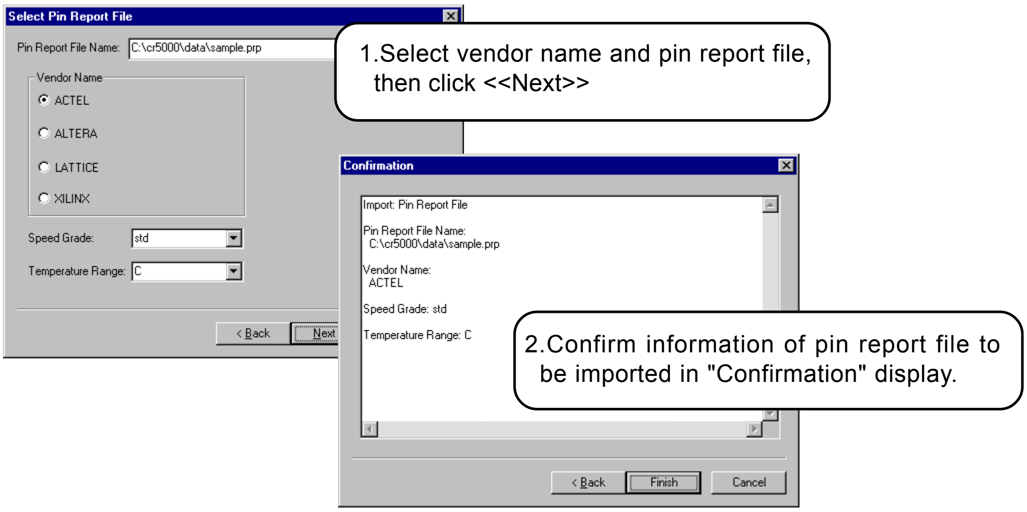
-For bus width, bit number calculated from bit range is imported. If text macro is used for MSB or LSB of bit range, it is replaced with value referring to stored text macro setting

VHDL design:	Internal memory:	
	[User label]	[IO Property]
port (A: out std_logic_vector (7 downto 0):	A[7:0]	OUTPUT
MCLK, MCLR : in std_logic:	MCLK	INPUT
\B&\: inout std_logic_vector (0 to 7)):	MCLR	INPUT
:	B&[0:7]	BIDIRECT

5.Importing from pin report file

The following vendor's pin report files can be imported

-ACTEL
-ALTERA
-LATTICE
-XILINX



Pin Report File Name: C:\cr5000\data\sample.prp

Vendor Name
● ACTEL
○ ALTERA
○ LATTICE
○ XILINX

Speed Grade: std

Temperature Range: C

```
Import: Pin Report File

Pin Report File Name:
 C:\cr5000\data\sample.prp

Vendor Name:
 ACTEL

Speed Grade: std

Temperature Range: C
```

1.Select vendor name and pin report file, then click <<Next>>

2.Confirm information of pin report file to be imported in "Confirmation" display.

<Contents of process>

The following is correspondence between pin report file information and Component Designeritems.

[Component] Logic pin name is imported to pin property "User label"
Physical pin number is imported to pin property "Pin number".

[Parts property] Part name is imported to "Entity name".
Design name is imported to "asicDesignName".
Vendor name is imported to "asicVendorName".
Created date is imported to "asicDesignDate"

Reference

For details of specification converted from pin report file, see the online help.
Online help [Tool menu] - [Component Designer]
[File] - [Import] - [Component Designer Pin report file Import]

Importing from Schematic Sheet

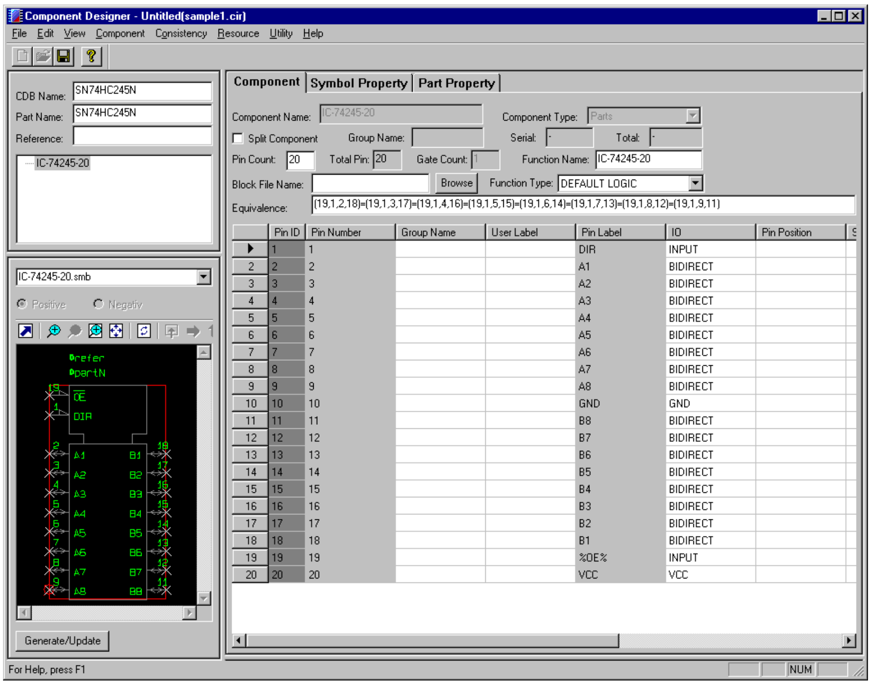
By specifying the component or the symbol listed on the Schematic Sheet of the System Designer, information is directly imported into the Component Designer.

1. Importing component selected on the Schematic Sheet

The component selected on the Schematic Sheet is imported into the Component Designer by reading LCDB information being referenced.

<Operation>

Select the component on the Schematic Sheet.
With the component selected, choose [Utility] - [Component Designer] from the menu bar.


Component Designer - Untitled(sample1.cir)

CDB Name: SN74HC245N
Part Name: SN74HC245N
Reference:
IC-74245-20

IC-74245-20.smb
● Positive ○ Negativ

Component Name: IC-74245-20
Component Type: Parts
□ Split Component
Group Name:
Serial: -
Total: -
Pin Count: 20
Total Pin: 20
Gate Count: 1
Function Name: IC-74245-20
Block File Name:
Function Type: DEFAULT LOGIC
Equivalence: (19,1,2,18)=(19,1,3,17)=(19,1,4,16)=(19,1,5,15)=(19,1,6,14)=(19,1,7,13)=(19,1,8,12)=(19,1,9,11)

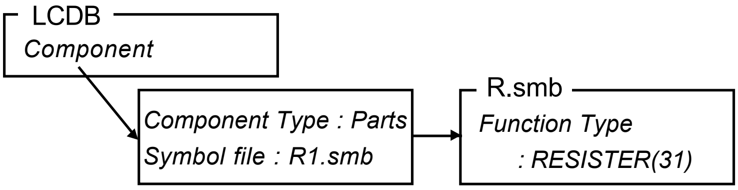
	Pin ID	Pin Number	Group Name	User Label	Pin Label	IO	Pin Position
▶	1	1			DIR	INPUT	
2	2	2			A1	BIDIRECT	
3	3	3			A2	BIDIRECT	
4	4	4			A3	BIDIRECT	
5	5	5			A4	BIDIRECT	
6	6	6			A5	BIDIRECT	
7	7	7			A6	BIDIRECT	
8	8	8			A7	BIDIRECT	
9	9	9			A8	BIDIRECT	
10	10	10			GND	GND	
11	11	11			B8	BIDIRECT	
12	12	12			B7	BIDIRECT	
13	13	13			B6	BIDIRECT	
14	14	14			B5	BIDIRECT	
15	15	15			B4	BIDIRECT	
16	16	16			B3	BIDIRECT	
17	17	17			B2	BIDIRECT	
18	18	18			B1	BIDIRECT	
19	19	19			%OE%	INPUT	
20	20	20			VCC	VCC	

The Component Designer starts and the contents of LCDB will be imported with the CDB name (part name if there is no CDB Name) of the selected component used as the key.

<Processing of the Information to be imported>

[Processing of Function Type]

Import the function type of the symbol referred to by [parts] component registered on the LCDB. If there are two or more symbols referred to by [parts] component, the function of the symbol described at the beginning of the components will be imported.



2. Importing the split symbol selected on the Schematic Sheet

The split symbol selected on the Schematic Sheet will be imported into the ComponentDesigner.

Based on the selected symbol file name (XXX.smb), the split symbol of same version will be searched for and invoked in the "split mode".

Reference "Version" is created from the same symbol and is used to represent a group of symbols. For details of assignment of the version, refer to "Automatic Creation of Symbol Name" in "Symbol Export" online help.

<Operation>

Select the split symbol on the Schematic Sheet.

With the symbol selected, choose [Utility]-[Component Designer].

The Component Designer starts, and at the same time, other split symbols of the same component as the selected split symbol will also be imported.

<Processing of Information to be imported>

[Processing of Function Type]

If all symbols to be imported are of the same function type, that function type will be imported. If different function types are present, the function type will be imported with the number set to "0".

[Processing of Component Name]

If all symbols have the same function type, that component name will be imported as "LCDB Reference Component Name".

If there is a different component name, LCDB reference component name will be "part_package".

LCDB reference component name is used when LCDB is checked, a symbol is created or updated.

The component name referred to here means that used in LCDB rather than "component name" inherently assigned to each split component on the Component Designer.

[Processing of Parts Properties]

If all symbols to be imported have the same value for a certain property, that property will be imported as the parts property. If there is a different value, the value of that property will not be imported.

Note that the following properties will not be copied from symbol properties to parts properties.

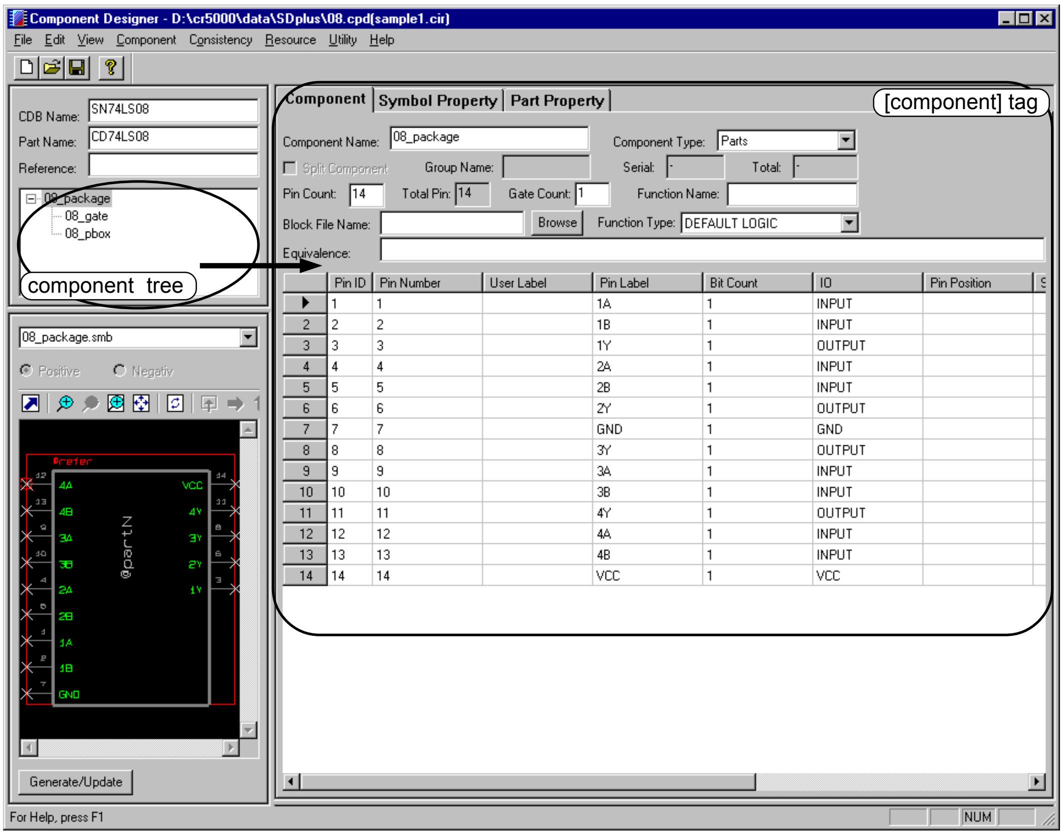
Component Type (componentType)
Function Name (function)
Block File Name (blockName)
Gate Count (gateCount)
Reference Designater (reference)
Split Component (isvGate)

2. Composition of a component

Define composition of a component used in the Component Designer.

<Composition of a component>

Component information selected in component tree is displayed in [Component] tag.



Component Designer - D:\cr5000\data\SDplus\08.cpd(sample1.cir)

CDB Name: SN74LS08
Part Name: CD74LS08
Reference:

- 08_package
  - 08_gate
  - 08_pbox

08_package.smb

● Positive ○ Negativ

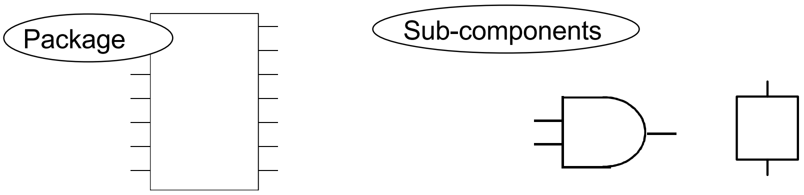
Component Name: 08_package
Component Type: Parts
□ Split Component
Group Name:
Serial: -
Total: -
Pin Count: 14
Total Pin: 14
Gate Count: 1
Function Name:
Block File Name:
Function Type: DEFAULT LOGIC
Equivalence:

	Pin ID	Pin Number	User Label	Pin Label	Bit Count	IO	Pin Position
▶	1	1		1A	1	INPUT	
2	2	2		1B	1	INPUT	
3	3	3		1Y	1	OUTPUT	
4	4	4		2A	1	INPUT	
5	5	5		2B	1	INPUT	
6	6	6		2Y	1	OUTPUT	
7	7	7		GND	1	GND	
8	8	8		3Y	1	OUTPUT	
9	9	9		3A	1	INPUT	
10	10	10		3B	1	INPUT	
11	11	11		4Y	1	OUTPUT	
12	12	12		4A	1	INPUT	
13	13	13		4B	1	INPUT	
14	14	14		VCC	1	VCC	

For Help, press F1

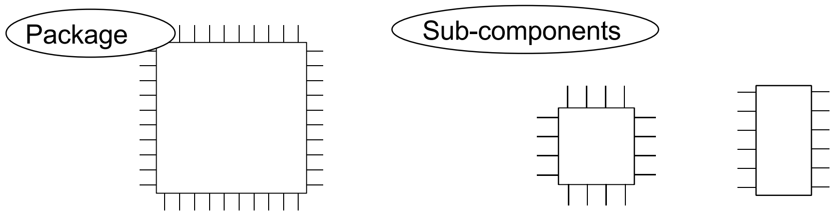
Component tree for a component consists of package component and sub-component. (Sub-component is not always necessary)

- When the editing mode is "Normal mode", sub-components represent gate or power box



Split

- When the editing mode is "Split Component Mode", sub-component represents each split component after the package is divided.



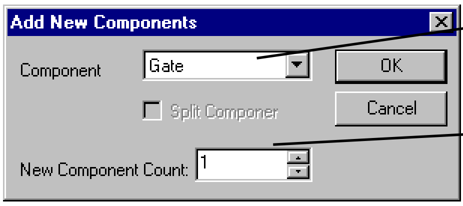
Components are automatically synthesized when [Import] function is used.
It can be added or deleted by user.
To add, it can be loaded from external file.
(External file = symbol file, VerilogHDL, VHDL)

<Additional operation>

(1)Adding new component

[Component] - [Add New Component]

Editing Mode=Normal Mode



Add New Components

Component: Gate

☐ Split Componer

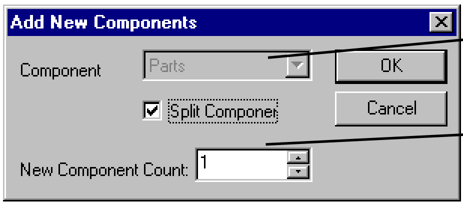
New Component Count: 1

Specify component type.

Specify number of component you want to add.

Split

Editing Mode=Split Component Mode



Add New Components

Component: Parts

☑ Split Componer

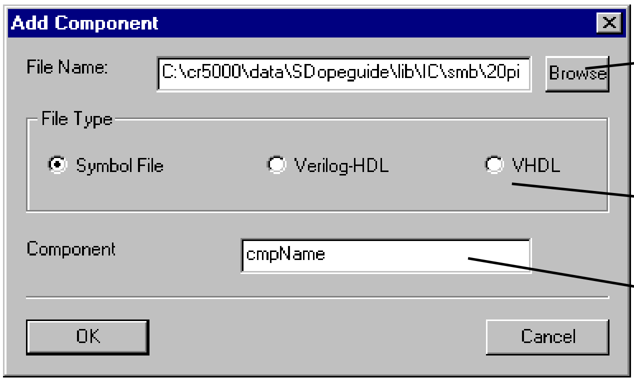
New Component Count: 1

Specify whether to select split component or not.

Specify number of component you want to add.

(2)Adding from external file

[Component] - [Add Component File]



Add Component

File Name: C:\cr5000\data\SDopeguide\lib\IC\smb\20pi

File Type

● Symbol File ○ Verilog-HDL ○ VHDL

Component: cmpName

Specify file name.

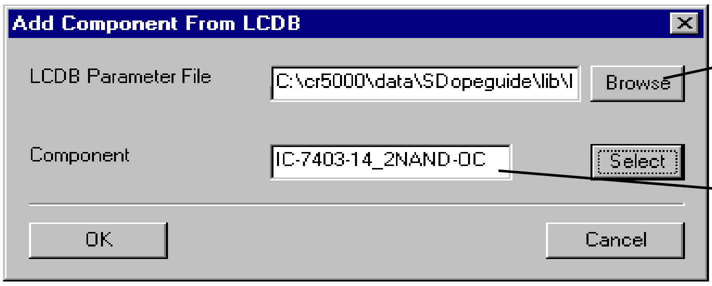
Specify loading external file type.

Specify a component name if you want.

(3)Adding from LCDB

* It is available only when editin mode is "Nomal Mode".

[Component] - [Add Component From LCDB File]



Add Component From LCDB

LCDB Parameter File: C:\cr5000\data\SDopeguide\lib\I

Component: IC-7403-14_2NAND-OC

Specify file name.

Specify a component name you want to load.

Split

Synthesizing split components

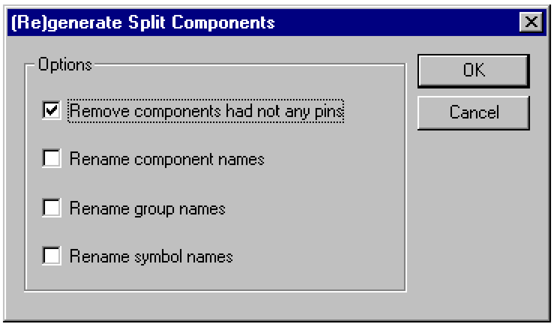
You can (re) synthesize components for each group name with "Split Component Mode".

<Operation method>

Select [Component] - [(Re) Generate Split Components] from the menu bar.

Options

Delete components that do not have pins.
Reassign component name
Reassign group name
Reassign symbol name



(Re)generate Split Components

Options

☑ Remove components had not any pins

□ Rename component names

□ Rename group names

□ Rename symbol names

3.Editing Component information

Edit component information such as pin information with the Component Designer.

- Editing component information

Edit properties displayed in [component] tag as required.

> Displayed items of "Normal mode" and "Split Component Mode" are different from each other.

- Defining pin information

In the pin definition table, 1 pin is described in one line. Property items are displayed at the upper part of the pin definition table.

<Editing contents or operation method>

Adding pin

Adding from the first.

-Enter value in [Pin Count]

-Select [Add pin] from the pop up menu of the pin definition table.

Adding to existing table

-Right-click serial number and select [Add pin] or [Insert pin] from the pop up menu

-To add, specify number of pins you want to add in the dialog.

-Change value of [Pin Count]

Deleting pin

-Right-click serial number of pin you want to delete and select [Delete pin] from the pop up menu

-Change value of [Pin Count]

Move pin

-Right-click serial number of pin you want to move and select [Move Row] from the pop up menu.

-Specify serial number of place where the pin is moved in the dialog.

Defining pin property

-Click a cell and enter value from keyboard.

When there are value candidates, you can select from one of the following methods.

-Click a cell and select from the choices.

-Right-click a cell and select from the pop up menu.

-Enter first numeric letter of candidate value.

(In case of key-input, you only have to enter numeric letter to be recognized)

<Items that can be selected from the assist menu>

Input Seqential data -	Create successive data for more than one cell by specifying first letter, numeral, and increment in the dialog.
Sort Ascending/Sort Descending -	Sort the pin definition table by value of selected property text.
Search -	Search text specified in the dialog from your selected property text.
Copy/ Paste / Clear value -	Copy, Paste and clear property value.
Protecting data / Canceling data protection -	Make it possible or impossible to edit selected property.

ID is always assigned to pin. ID is important property that can be key to load or synthesis of information from external data. So the following functions are added to ID editing. Note that pin ID cannot be edited when the edit mode is "Split Component Mode" where "Edit Disabled" takes effect.
Package information is required because the following functions are used.

Renumbering ID by package component ID.
Select [Renumber ID by package Pin ID] from the assist menu.
Match to ID defined in package component by designating value of pin number or pin name as key word. Property value needs to be entered as keyword.

Renumbering ID to pin number.
Select [Renumber ID by package Pin Number] from the assist menu.
Match to defined pin number

Reassign ID in ascending order.
Select [Reassign by Line Number] from the assist menu.
Reassign in order of serial number.

NOTE

ID is normally auto-generated in order of pin assignment.

Editing pin property text

Specify to display or hide each property in the pin definition table.

-Click property name and select [Edit pin property text] from the assist menu.

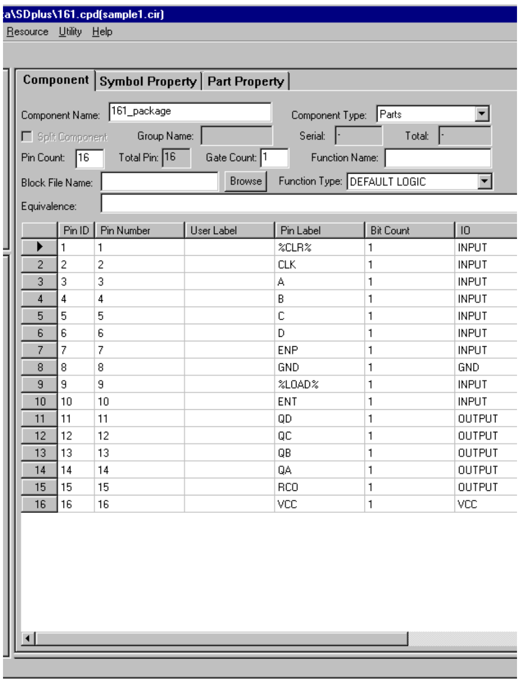
Split

Defining group name when dividing pin

Property [Group Name] is displayed only when the editing mode is "Split Component Mode" A combination of split pins is determined by group name.

-Specify group name in [Group Name] in the pin definition table.Enter the same group name for the pin you want to place in the same group.

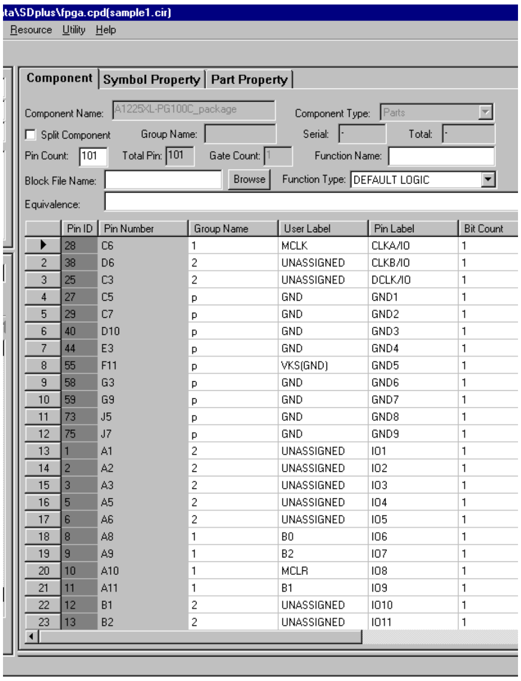
<Normal Mode>


a\SDplus\161.cpd(sample1.cir)

Component Name: 161_package
Component Type: Parts
□ Split Component
Group Name:
Serial: -
Total: -
Pin Count: 16
Total Pin: 16
Gate Count: 1
Function Name:
Block File Name:
Function Type: DEFAULT LOGIC
Equivalence:

	Pin ID	Pin Number	User Label	Pin Label	Bit Count	IO
▶	1	1		%CLR%	1	INPUT
2	2	2		CLK	1	INPUT
3	3	3		A	1	INPUT
4	4	4		B	1	INPUT
5	5	5		C	1	INPUT
6	6	6		D	1	INPUT
7	7	7		ENP	1	INPUT
8	8	8		GND	1	GND
9	9	9		%LOAD%	1	INPUT
10	10	10		ENT	1	INPUT
11	11	11		QD	1	OUTPUT
12	12	12		QC	1	OUTPUT
13	13	13		QB	1	OUTPUT
14	14	14		QA	1	OUTPUT
15	15	15		RCO	1	OUTPUT
16	16	16		VCC	1	VCC

<Split Component Mode>


ta\SDplus\fpga.cpd(sample1.cir)

Component Name: A1225XL-PG100C_package
Component Type: Parts
□ Split Component
Group Name:
Serial: -
Total: -
Pin Count: 101
Total Pin: 101
Gate Count: 1
Function Name:
Block File Name:
Function Type: DEFAULT LOGIC
Equivalence:

	Pin ID	Pin Number	Group Name	User Label	Pin Label	Bit Count
▶	28	C6	1	MCLK	CLKA/IO	1
2	38	D6	2	UNASSIGNED	CLKB/IO	1
3	25	C3	2	UNASSIGNED	DCLK/IO	1
4	27	C5	p	GND	GND1	1
5	29	C7	p	GND	GND2	1
6	40	D10	p	GND	GND3	1
7	44	E3	p	GND	GND4	1
8	55	F11	p	VKS(GND)	GND5	1
9	58	G3	p	GND	GND6	1
10	59	G9	p	GND	GND7	1
11	73	J5	p	GND	GND8	1
12	75	J7	p	GND	GND9	1
13	1	A1	2	UNASSIGNED	IO1	1
14	2	A2	2	UNASSIGNED	IO2	1
15	3	A3	2	UNASSIGNED	IO3	1
16	5	A5	2	UNASSIGNED	IO4	1
17	6	A6	2	UNASSIGNED	IO5	1
18	8	A8	1	B0	IO6	1
19	9	A9	1	B2	IO7	1
20	10	A10	1	MCLR	IO8	1
21	11	A11	1	B1	IO9	1
22	12	B1	2	UNASSIGNED	IO10	1
23	13	B2	2	UNASSIGNED	IO11	1

4.Editing symbol property / parts property

Edit symbol property and parts property.

[Symbol property]

Display and edit symbol property of symbol that is referred to for editing component.Properties that are displayed in symbol property tag can be assigned to symbol sheet using [Generate / Update symbol] dialog.

[Parts property]

Display and edit parts property of a component that is being edited in the ComponentDesigner.

<Operation process>

(This is common to all tags)

Click a cell of a property you want to define and enter value form keyboard. If there are value candidates, you can select from one of the following methods.

-Click a cell and select from the choices.
-Right-click a cell and select from the popup menu.
-Enter first numeric letter of candidate value.
(In case of key-input, you only have to enter numeric letter to be recognized.)

<Items that can be selected from the assist menu>

Operation for call

Clearing value - Clear property value
Protecting data / Canceling data protection - Make it possible or impossible to edit selected property value.

Operation for line

Protecting data / Canceling data protection - Make it possible or impossible to edit selected property value.
(The following items are applied to [Parts property] only)
Moving line - Move line to the line of specified serial number.
Changing line -Switch selected 2 lines.
Deleting line - Delete selected property.

New property can be added in [Parts property] only.

Select property you want to add from [Add new property] at the bottom of [Property name] row.

NOTE

Displayed property is normally dependent on property definition file ($ZDSROOT/etc/jpn/PropSpec).

5.Consistency Check

Check consistency of information that is being edited in the Component Designer. Results message is displayed in the dialog.
In the case of LCDB check, the error log file is outputted to following place. (The file nameis"partname.log")

- When the component is selected and started from [Utilities] - [Component Designer] in the Schematic Editor of System Designer -> XXX.cir/log/
-When Component Designer is started alone -> $HOME/cr5000/ds/log/

<Process>

Select [Consistency] - [LCDB] or [Internal data] from the menu bar.

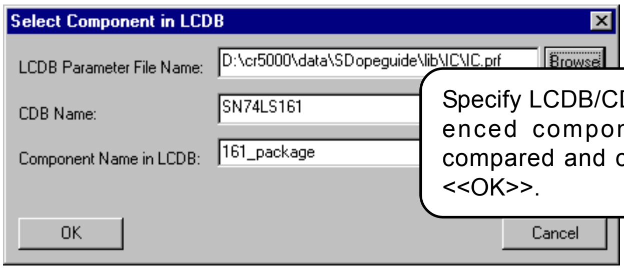
[LCDB] * It is available only when editing mode is "Split Component Mode".

Compare data between Component Designer and LCDB and check them.

Check items

- Equality of pin number.
- Equal correspondence of properties for pin number or pin mane

Plus the same check as [Internal data] (See below)



Select Component in LCDB

LCDB Parameter File Name: D:\cr5000\data\SDopeguide\lib\IC\IC.prf

CDB Name: SN74LS161

Component Name in LCDB: 161_package

Specify LCDB/CDB name and referenced component name to be compared and checked, then click <<OK>>.

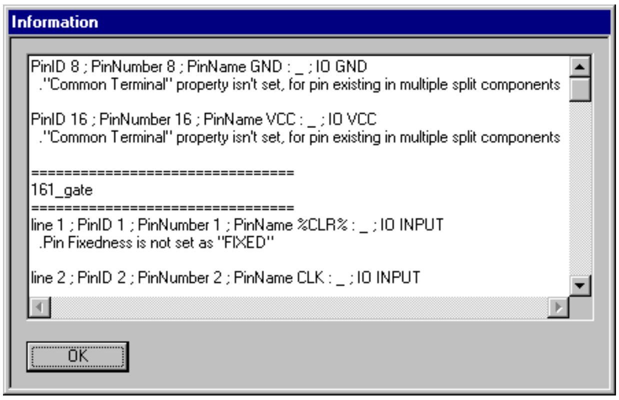
[LCDB]

Select [Consistency check] - [LCDB] or [Internal data] from the menu bar.

Check items

-Duplication of pin name and pin number
-Lack of required properties or pin data.
-Accuracy of property value.
-Latest update of symbol file

Check results are displayed.



```
PinID 8 ; PinNumber 8 ; PinName GND : _ ; IO GND
  ."Common Terminal" property isn't set, for pin existing in multiple split components

PinID 16 ; PinNumber 16 ; PinName VCC : _ ; IO VCC
  ."Common Terminal" property isn't set, for pin existing in multiple split components

================================
161_gate
================================
line 1 ; PinID 1 ; PinNumber 1 ; PinName %CLR% : _ ; IO INPUT
  .Pin Fixedness is not set as "FIXED"

line 2 ; PinID 2 ; PinNumber 2 ; PinName CLK : _ ; IO INPUT
```

6.Symbol auto-generation

Symbol can be auto-generated or edited for each component.

CAUTION

> -Symbol to be generated or edited is temporary one. To use it in schematic sheet, output it as symbol file using export function.
>
> -Symbol cannot be auto-generated without defining the pin definition table. When Component type is gate, pin number is not displayed in generated symbol pin.

1. Symbol auto-generation

Auto-generate a symbol sheet.

<Operation process>

There are 2 methods to auto-generate symbols.

Click [Generate/Update symbol] button in the symbol viewer.

Set the condition in [Generate/Update symbol] dialog and click <<OK>>.

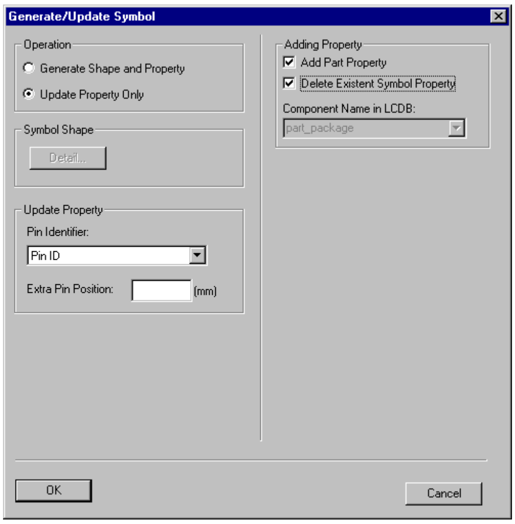
Symbol is generated for your editing component.

Select [Component] - [Generate/Update symbol] from the menu bar.

Set a condition in [Generate/Update symbol] dialog and click <<OK>>.

You can select symbols generated from package component and sub-component.

Start up from the button in the symbol viewer



Generate/Update Symbol

Operation

○ Generate Shape and Property
● Update Property Only

Symbol Shape

Update Property

Pin Identifier: Pin ID

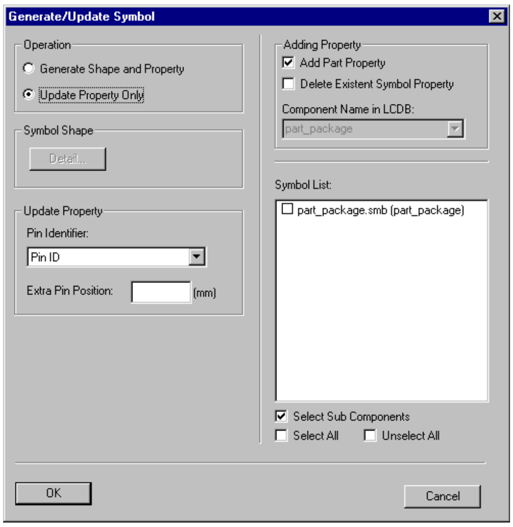
Extra Pin Position: (mm)

Adding Property

☑ Add Part Property
☑ Delete Existent Symbol Property

Component Name in LCDB: part_package

Start up from the menu bar



Generate/Update Symbol

Operation

○ Generate Shape and Property
● Update Property Only

Symbol Shape

Update Property

Pin Identifier: Pin ID

Extra Pin Position: (mm)

Adding Property

☑ Add Part Property
☐ Delete Existent Symbol Property

Component Name in LCDB: part_package

Symbol List:

☐ part_package.smb (part_package)

☑ Select Sub Components
☐ Select All ☐ Unselect All

<Items in the [Generate/Update Symbol] dialog>

[How to Generate / Update]

"Generate shape and Property"
- Genarate/Update symbol.

"Renew property value only"
- Update only property value from [Symbol property]. Symbol is not generated.

[Symbol Shape]

<<Details setting>> Display details setting dialog for symbol shape (Refer to <Item i "Symbol Shape" dialog>)

[Update property]

[Pin Identifire]

When renewing property values, specify pin property as keyword to judge if previous pin and renewal pin are the same. It is available only when "Update Property Only" is ON in [How to Genarate / Update].
You can select pin property from the following.

-Pin ID (ID)
-Pin number (pinNumber)
-Pin name (pinLabel)
-User label (cirPinLabel)

[Extra Pin Position]

Specify distance from existing pin area when adding pin to existing symbol.
It is available only when "Update Property Only" is ON in [How to Genarate / Update].
If no value is specified, the following value is used.
Value used in "Corner Margin" X 3

[Adding Property]

[Add Parts Property]

Turn it ON, then component property and parts property are added as sheet properties of symbol figure. (Symbol property is added in any case)

[Delete Existent Sysmbol Property]

Turn it ON, then existent Symbol properties are deleted.

[Component name in LCDB]

*It can be edited only when the editing mode is "Split Component Mode".
- Specify package component name that includes symbol to be created.
Component name specified here is referred to in LCDB.

[Symbol List]

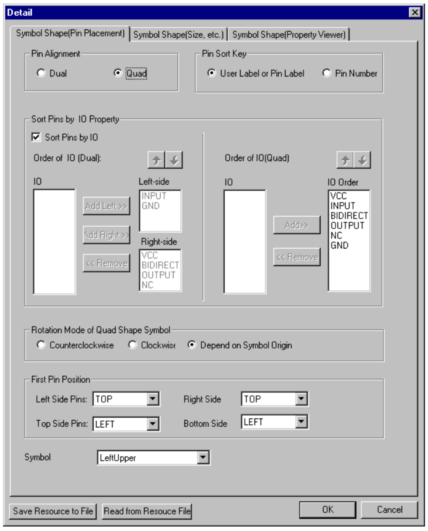
Select symbol to be generated or renewed from all components of editing parts.

[Select Sub Component] - Turn it ON, then all sub-components are selected.

[Select All] - Turn it ON, then all components are selected.

[Unselect All] - Turn it ON, then all selected components are cancelled.

<"Detail" dialog for creating / renewing Symbol Shape>



Detail

Symbol Shape(Pin Placement) | Symbol Shape(Size, etc.) | Symbol Shape(Property Viewer)

Pin Alignment: ○ Dual ● Quad

Pin Sort Key: ● User Label or Pin Label ○ Pin Number

Sort Pins by IO Property

☑ Sort Pins by IO

Order of IO (Dual):

IO

Left-side: INPUT, GND

Right-side: VCC, BIDIRECT, OUTPUT, NC

Order of IO(Quad)

IO

IO Order: VCC, INPUT, BIDIRECT, OUTPUT, NC, GND

Rotation Mode of Quad Shape Symbol: ○ Counterclockwise ○ Clockwise ● Depend on Symbol Origin

First Pin Position

Left Side Pins: TOP

Right Side: TOP

Top Side Pins: LEFT

Bottom Side: LEFT

Symbol: LeftUpper

<Items in "Detail" dialog for Symbol Shape>

[Symbol Shape]

[Pin Alignment]

[Pin alignment]

"Dual": Pins are aligned side by side.

"Quad": Pins are aligned in all directions.

[Pin Sort Key]

Specifies the property to determine the order of pin alignment.

The property serving as the key will vary depending on the presence of property, and pins will be determined according to the following priority.

"Pin name or user pin label":

1.User pin label (Sort is made with the negative logic start and end characters excluded.)

2.Pin name (Sort is made with the negative logic start and end characters excluded.)

3.Pin number

4.Pin ID

"Pin number"

1.Pin number

2.User label (Sort is made with the negative logic start and end characters excluded.)

3.Pin name (Sort is made with the negative logic start and end characters excluded.)

4.Pin ID

[Pin sort by IO properties]

"IO properties are considered"

Specifies whether to consider IO properties for pin sort upon creation of symbol.

If turned ON, the position of IO property will be determined by "Order of IO property alignment (Dual/Quad) ".

If turned OFF, or if no IO property is entered for any pin, all pins will be sorted and aligned into two equal parts in right and left.

[Direction of sides on quad alignment]

When no pin creation position is entered, specify the order of pin alignment where [Pin alignment] is set to "Quad".

[Counterclockwise]: Pins will be aligned counterclockwise from the side on which the symbol origin stays

[Clockwise]: Pins will be aligned clockwise from the side on which the symbol origin stays.

[Follow symbol origin] : Alignment begins with the side containing the symbol origin and turns counterclockwise against the position of the symbol origin. If the symbol origin stays at the left bottom, it follows in the direction of left-top-right-bottom, and if it stays at the upper left, it follows in the direction of left-bottom-right-top.

[First pin position for each side]

[Left side pins]/[right side pins] : Specifies the starting position of pins created from left to right. (TOP BOTTOM)

[Top side pins] / [bottom side pins] :Specifies the starting position of pins created from top to bottom if [Pin alignment] is set to “Quad”. (LEFT RIGHT)

[Symbol origin]

Specifies the position in which symbol origin is placed.

Any of the following items will be selected.

LeftUpper : Places the symbol origin at the left upper corner.

LeftLower : Places the symbol origin at the left lower corner.

RightUpper : Places the symbol origin at the right upper corner.

RightLower : Places the symbol origin at the right lower corner.

UpperLeft : Places the symbol origin at the upper left corner.
It can be specified only when [Pin alignment] is set to “Quad”.

UpperRight : Places the symbol origin at the upper right corner.
It can be specified only when [Pin alignment] is set to “Quad”.

LowerLeft : Places the symbol origin at the lower left corner.
It can be specified only when [Pin alignment] is set to “Quad”.

LowerRight : Places the symbol origin at the lower right corner.
It can be specified only when [Pin alignment] is set to “Quad”.

[Size, others]

[Rectangle]

“Rect width (Dual) “ : Determines the width of the rectangle when pin alignment is set to left to right.

“Length of corner cut (Quad)” : Specifies the length when cutting the corner of a rectangle. It can be specified only when [Pin alignment] is set to “Quad alignment”. The corner most close to “Symbol origin” will be cut by the amount specified.

“Corner Margin” : Specifies the distance from the corner of the rectangle to the first pin.

“Line width of rectangle” : Specifies the line width of the rectangle.

“Line color of rectangle” : Specifies the line color of the rectangle.

"Fit Symbol Shape (Quad)" : Turn it ON, the symbol fit for the number of pins is generated.

[Pin]

“Pin pitch” : Specifies the distance between pins.

“Length of pin” : Specifies the length of the pin (terminal pattern). When set to “0”, the pin will be inserted on the line of the rectangle.

“Generate negative circle” : When turned ON and if [User label (cirPinLabel)] or [Pin name (pinLabel)] is enclosed between the negative logic mark start character and the end character set in the data resource file (landata.rsc), the negative logic mark (a circle that represents negation) will be displayed in the symbol pin.

“Negative Circle Radius” : Specifies the radius of the negative logic mark displayed in a circle in millimeters.

“Width of pin line” : Specifies the line width of the pin.

“Line width of bus pin” : Specifies the line width of bus pin.

“Line color of pin” : Specifies line color of the pin.

[Property viewer]
Specifies the property viewer to be entered for the symbol and pin according to pin alignment.
[Dual] / [Quad]
Symbol property : Lists the symbol properties defined in the property definition file (PropSpec).
Input property viewer : Lists the property viewers to be entered for the symbol.
Pin property : Lists the pin properties defined in the property definition file (PropSpec).
Input property pin viewer : Lists the property viewers to be entered for pins.

NOTE

When pin property "User label (cirPinLabel)" exists, insert property viewer of "User label (cirPinLabel)" and turn the property viewer display OFF. When "User label (cirPinLabel)"exists, "User label (cirPinLabel)" is used to judge if it is negative logic or not instead of "Pin name (pinLabel)".

2. Editing symbol

Open symbol sheet and perform editing as required.

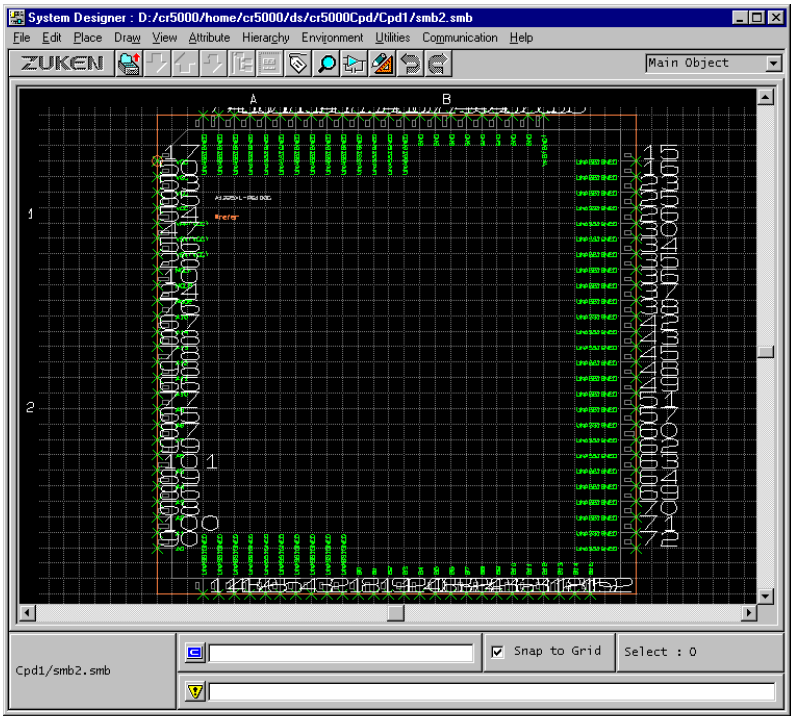
<Operation method>

There are 2 methods to edit symbol. Both have the same processing way.

Double-click symbol viewer.

Select [Component]-[Edit symbol] from the menu bar.

Symbol Editor starts and you can edit temporary symbol sheet.



System Designer : D:/cr5000/home/cr5000/ds/cr5000Cpd/Cpd1/smb2.smb

Main Object

Cpd1/smb2.smb

☑ Snap to Grid

Select : 0

CAUTION

Symbol sheet you edit here is temporary one. So directory name and file name are also temporary. This symbol can be used as symbol by exporting after saving.

7.Export

Component properties and parts properties edited with the Component Designer are converted into data in various formats and exported by means of the export wizard. You can also export the split symbols edited with the Component Designer into the symbol directory used for the Schematic Sheet.

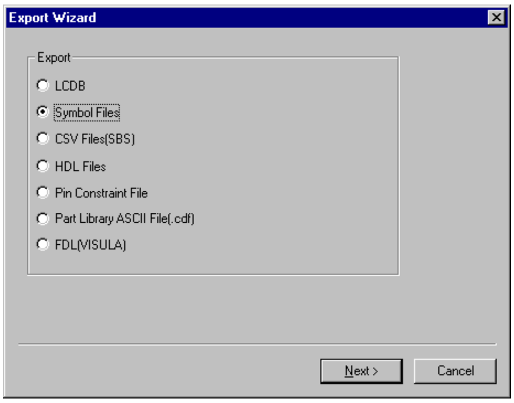
Exporting to various formats

Component properties and parts properties edited with the Component Designer are converted to data in various formats by means of the export wizard.

<Data types to which you can export information>

You can export information to the following data.

-LCDB
-Symbol
-CSV format file (SBS)
-Hardware description language
(Verilog-HDL, VHDL)
-Pin report file
(ACTEL, ALTERA, XILINX, LATTICE)
-Part library ASCII file (cdf)
-FDL(VISULA)



Export Wizard

Export
○ LCDB
● Symbol Files
○ CSV Files(SBS)
○ HDL Files
○ Pin Constraint File
○ Part Library ASCII File(.cdf)
○ FDL(VISULA)

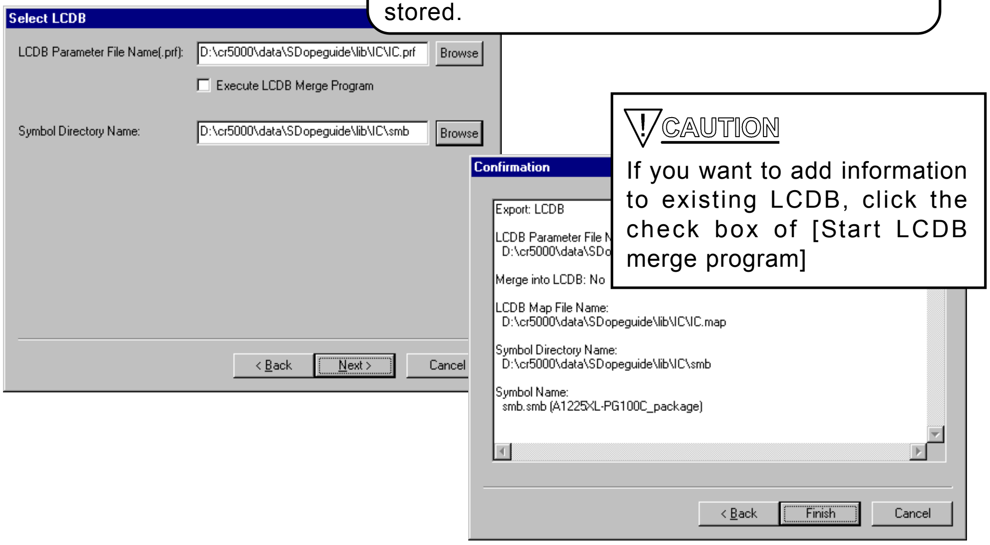
<Operation process>

[File] - [Export]

Specify data type to which you want to export information in the Wizard.

1. LCDB export

Export component property and parts property edited in the Component Designer to LCDB parameter file (.prf). And also, export symbol that the component refers to as symbol sheet (.smb). One component is exported to one LCDB parameter file.



Specify LCDB name to which you want to output information and directory to which symbol is stored.

Select LCDB

LCDB Parameter File Name(.prf): D:\cr5000\data\SDopeguide\lib\IC\IC.prf
□ Execute LCDB Merge Program
Symbol Directory Name: D:\cr5000\data\SDopeguide\lib\IC\smb

Confirmation

Export: LCDB

LCDB Parameter File N
D:\cr5000\data\SDo

Merge into LCDB: No

LCDB Map File Name:
D:\cr5000\data\SDopeguide\lib\IC\IC.map

Symbol Directory Name:
D:\cr5000\data\SDopeguide\lib\IC\smb

Symbol Name:
smb.smb (A1225XL-PG100C_package)

CAUTION

If you want to add information to existing LCDB, click the check box of [Start LCDB merge program]

When exporting, each property is converted as follows.

<Contents of process>

[CDB name], [Part name] - Exported to LCDB parts property [CDB name], [Part name]

[Component tree]

- When the editing mode is "Normal mode", package and all sub-components displayed in the component tree are exported as LCDB components. When the editing mode is "Split component mode", only package displayed in the component tree is exported as LCDB component.

-Exporting package
Package is exported as a component which component type is "Parts".

-Exporting sub-components
When the editing mode is "Normal mode", sub-components are exported as components that have "component type" set in the Component Designer". When the editing mode is "Split component mode", sub-components are not exported.

Symbol - Symbol sheets that each component has are exported to your specified directory.

Properties displayed in [Component] tag.
Exported as properties that LCDB components have. Exported as symbol sheet property as well.

[Component name] - Exported as component name to be registered in LCDB.

[Component type] - Exported as component type that component has.

[Split component]/[Group name]/[Serial]/[Total]/[Function type]
- Not exported.

[Pin property table] property
- Exported as component pin information. However, [User label][Pin creating point] [In order of pin creation] are not exported.

Properties displayed in [Symbol property] tag.
-Exported as symbol sheet properties when properties of [Symbol property] tag are assigned to symbol sheet in [Create/Renew symbol] dialog.
-When symbol property is not assigned to symbol sheet, properties displayed in "Symbol property" tag are not exported.

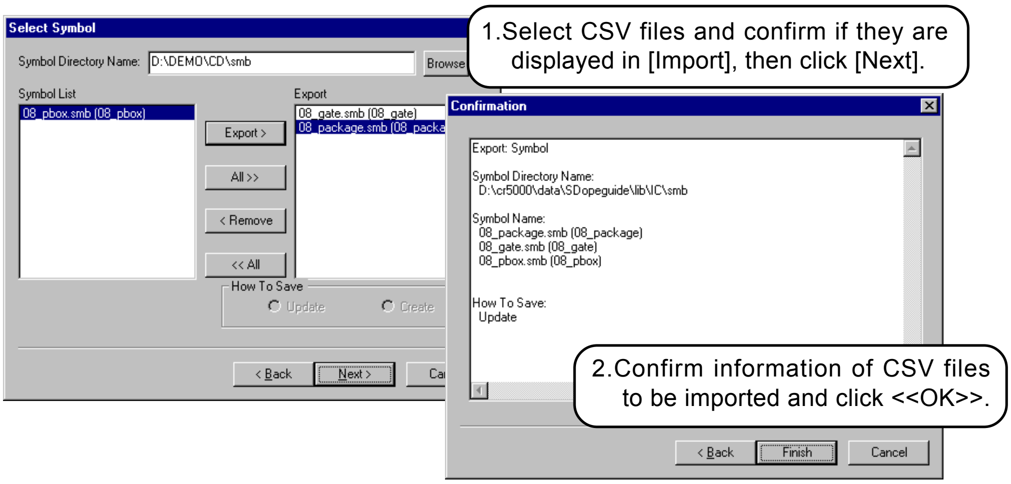
Properties displayed in [Parts property] tag.
Exported as LCDB parts property. Not exported to symbol sheet.

! CAUTION

If you want to add parts edited in the Component Designer to existing LCDB, click the check box of [Start LCDB merge program] in the dialog when exporting.
If you write in existing LCDB without checking it, other parts information is deleted.

2.Exporting symbol

Export symbol figure or symbol property edited in the Component Designer as symbol sheet (.smb).
Only symbol properties assigned to symbol sheet are exported by performing "Generate/ Update symbol". All symbol sheets registered in the Component Designer are covered. Positive / Negative logic symbols are also covered.
If there are repeated symbol file names in the export file when creating a new file, you can export them with file name automatically incremented.



Select Symbol

Symbol Directory Name: D:\DEMO\CD\smb

Symbol List: 08_pbox.smb (08_pbox)

Export: 08_gate.smb (08_gate) / 08_package.smb (08_packa

How To Save: ○ Update ○ Create

Confirmation

Export: Symbol

Symbol Directory Name:
D:\cr5000\data\SDopeguide\lib\IC\smb

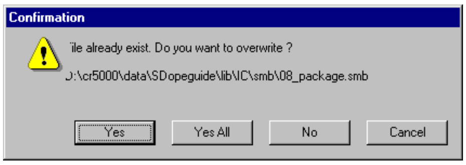
Symbol Name:
08_package.smb (08_package)
08_gate.smb (08_gate)
08_pbox.smb (08_pbox)

How To Save:
Update

1.Select CSV files and confirm if they are displayed in [Import], then click [Next].

2.Confirm information of CSV files to be imported and click <<OK>>.

Select [Update] from [How to save]. Then the file name appears in the export file, or the confirmation dialog appears if there is a duplicate version name.



Confirmation

ile already exist. Do you want to overwrite ?
:\cr5000\data\SDopeguide\lib\IC\smb\08_package.smb

When you click “Yes”, an optional file name will be specified and saved.
When you click “Auto-generate All”, the repeated file name will be automatically incremented and saved.

When exporting, each property is converted as follows.

<Contents of process>

Properties displayed in [Component] tag.

[Component name] - Exported as symbol name.

[Component type][Function name][Function type]
- Exported as symbol sheet properties.

Pin properties displayed in [Pin property table].
- Exported as symbol pin property. However, Pin creating position] [In order of pin creation] are not exported

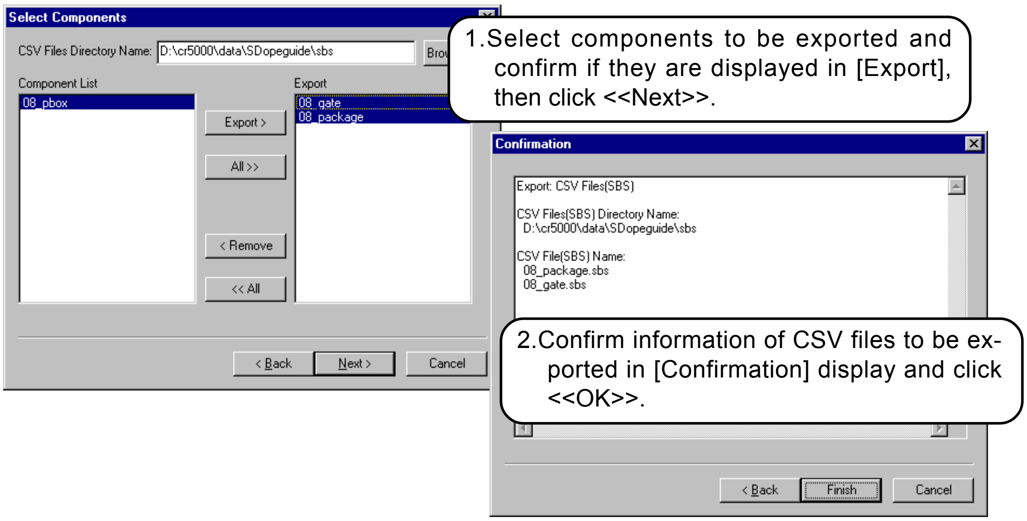
Properties displayed in [Symbol property] tag.
- Properties assigned to symbol sheet are exported by performing [Generate/Update symbol].

Properties displayed in [Parts property] tag.
- Properties assigned to symbol sheet are exported by performing [Generate/Update symbol]

Reference For how to create the symbol file name to be automatically generated, see “Exporting the Symbol” in the online help file.

3.Exporting from CSV file (SBS)

Export information from the Component Designer to fixed CSV file format.



Select Components

CSV Files Directory Name: D:\cr5000\data\SDopeguide\sbs

Component List: 08_pbox

Export: 08_gate, 08_package

1.Select components to be exported and confirm if they are displayed in [Export], then click <<Next>>.

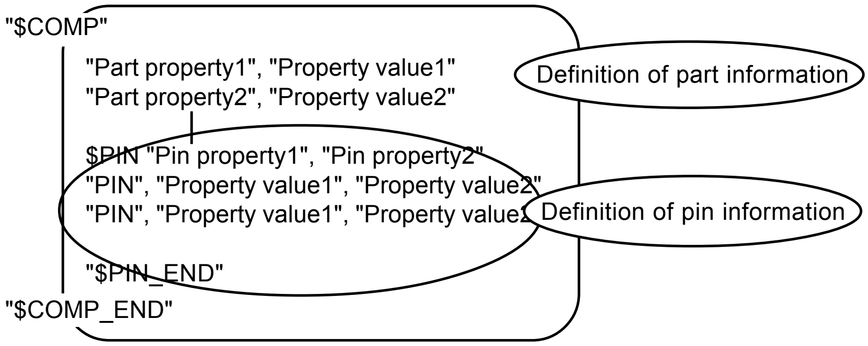
Confirmation

```
Export: CSV Files(SBS)

CSV Files(SBS) Directory Name:
 D:\cr5000\data\SDopeguide\sbs

CSV File(SBS) Name:
 08_package.sbs
 08_gate.sbs
```

2.Confirm information of CSV files to be exported in [Confirmation] display and click <<OK>>.

<Files to be exported>

The following are CSV file formats to be exported.



```
"$COMP"
      "Part property1", "Property value1"
      "Part property2", "Property value2"

      $PIN "Pin property1", "Pin property2"
      "PIN", "Property value1", "Property value2"
      "PIN", "Property value1", "Property value2"

      "$PIN_END"
"$COMP_END"
```

CSV file sample

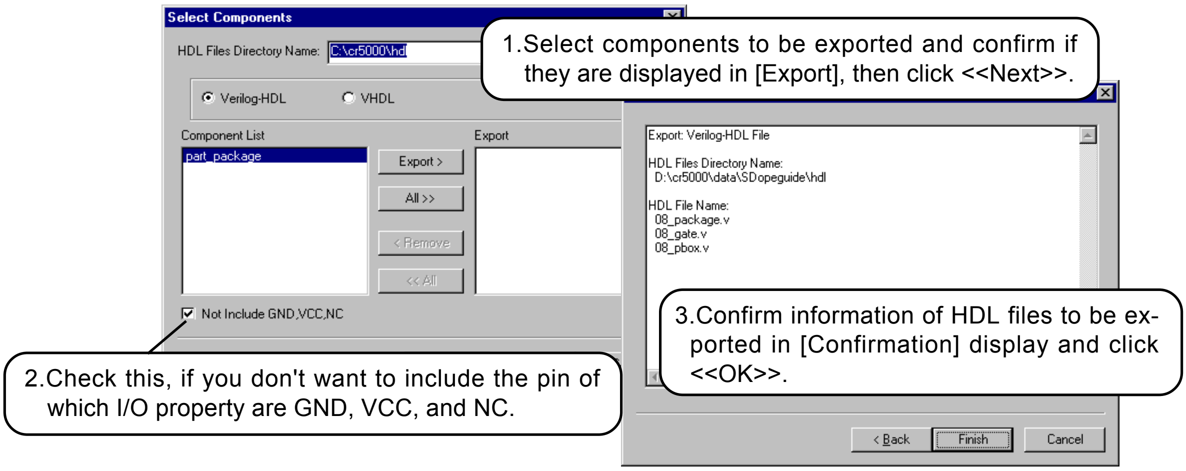
```
"$COMP"
"sbsSymbolOrigin","UpperLeft"
"isvGate","NO"
"sbsPinIdentificationProperty","sbsPinId"
"sbsRegerateShape","YES"
"sbsPinArrangement","QUAD"
"cdbName","74LS04-DIP"
"componentType","gate"
"partName","74ALS04-DIP"
"useWithSchema","YES"
"function","INVERTER"
"componentName","04_INVERTER"

"$PIN","sbsPinId","bitCount","pinNumber","cirPinLabel","pinLabel","io","sbsPinPosition"
"pin","1","1","1,3,5,9,11,13","","A","INPUT","LEFT"
"pin","2","1","2,4,6,8,10,12","","Y","OUTPUT","RIGHT"
"pin","3","1","14,14,14,14,14,14","","VCC","VCC","TOP"
"pin","4","1","7,7,7,7,7,7","","GND","GND","BOTTOM"
"$PIN_END"

"$COMP_END"
```

<Contents of process>

One CSV format file (.sbs) is created for one component.

4.Exporting hardware description language (VerilogHDL, VHDL)

Create hardware description language (Verilog-HDL, VHDL) file from parts property and pin property edited in the Component Designer.



Select Components

HDL Files Directory Name: C:\cr5000\hdl

● Verilog-HDL ○ VHDL

Component List: part_package

Export

☑ Not Include GND,VCC,NC

Export: Verilog-HDL File

HDL Files Directory Name:
D:\cr5000\data\SDopeguide\hdl

HDL File Name:
08_package.v
08_gate.v
08_pbox.v

1.Select components to be exported and confirm if they are displayed in [Export], then click <<Next>>.

2.Check this, if you don't want to include the pin of which I/O property are GND, VCC, and NC.

3.Confirm information of HDL files to be exported in [Confirmation] display and click <<OK>>.

When exported, each property is converted as follows.

<Contents of process>

Pin properties displayed in [Component] tag.

[User label] - Exported to "Input/Output port name".
When array notation is used in user label,
-Verilog-HDL - Array notation is exported to "Input/Output port name".
-VHDL - Array notation is exported to MSB or LSB of the port.

[Pin name] - If [User label] does not exist, it is exported instead.

[IO property] - When IO property is "INPUT" and "OUTPUT", each mode is "input" and "output". In case of the others, mode is "inout".

Parts properties displayed in [Parts property] tag.

[Entity name] - Exported as "Module name".

[Component name] - If [Entity name] does not exist, it is exported instead.

If [Entity name] and [User label] do not meet the following condition, enclose [Entity name] in "\" and [User label] in space and apply each as module name and input/output port.

-First letter of the name is English or "_".
-Nothing except English figure, "_" and "$" is used for the name. (Verilog-HDL)
-Nothing except English figure and "_" is used for the name. (VHDL)

Example of Verilog-HDL file output

```
Module (Port name1, Port name2, Port name3, ...);
 Mode (Vector width)    Port name1:
 Mode (Vector width)    Port name2:
Mode (Vector width)    Port name3:
..
..
//
// Please implement.
//
 endmodule
```

Example of VHDL file output

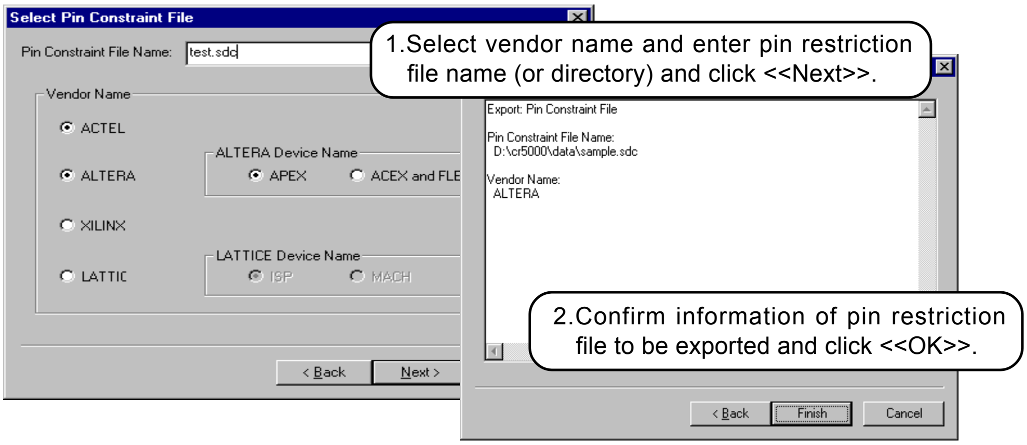
```
LIBRARY   ieee:
USE  ieee.  std_logic_1164,ALL;
Entity Entity name is
Port (Port name1: Mode port type:
        Port name2: Mode port type:
        Port name3: Mode port type:
                   :
        Port name N: Mode port type:
end   Entity name:
architecture RTL of Entity name is
--
--
--
begin
--
--  Please implement.
--
end RTL;
```

5.Exporting pin restriction file.

Create pin restriction file (.sdc) for Symbplify TM from pin properties [User label], [Pin number] edited in the Component Designer.
You can export the following vendor pin restriction file.

-ACTEL
-ALTERA
-XILINX
-LATTICE

Pin restriction file is used to specify (reserve) port of Verilog-HDL/VHDL design in target device physical pin number on Sumplify TM.



Select Pin Constraint File

Pin Constraint File Name: test.sdc

Vendor Name
● ACTEL
● ALTERA
○ XILINX
○ LATTIC

ALTERA Device Name
● APEX ○ ACEX and FLE

LATTICE Device Name
● ISP ○ MACH

< Back Next >

Export: Pin Constraint File

Pin Constraint File Name:
D:\cr5000\data\sample.sdc

Vendor Name:
ALTERA

< Back Finish Cancel

1.Select vendor name and enter pin restriction file name (or directory) and click <<Next>>.

2.Confirm information of pin restriction file to be exported and click <<OK>>.

<Contents of process>

The following are correspondence or formats of properties to be exported.

Properties displayed in [Pin definition table].

[User label] Regarded as interface port with Verilog-HDL/VHDL design that is entered in Synplify TN.

[Pin number] Regarded as physical pin number of target device to set for logic synthesis in Synplify TM.

Format

-Pin restriction file for ACTEL.
Define_attribute {user label} alspin {pin number}

-Pin restriction file for ALTERA (APEX).
Define_attribute {user label} altera_chip_pin_lc {pin number}

-Pin restriction file for ALTERA (ACEX and FLEX).
Define_attribute {user label} altera_chip_pin_lc {@pin number}

-Pin restriction file for XILINX.
Define_attribute {user label} xc_loc {Ppin number}
("P" is not added to the beginning of the Pin Number if it begins with an alphabetical character)

-Pin restriction file for LATTICE (ISP).
Define_attribute {user label} lock {pin number}

-Pin restriction file for LATTICE (MACH).
Define_attribute {user label} loc {pin number}

Pin restriction file contains header or setting statement of message to be output in Symplify TN log file.

-Restriction is not supported for the following pins.

No value in "User label" or "Pin number".

Value of "User label" is defined as power, ground, NC (No Connection) on target device.

Value of "IO" is defined "GND", "VCC", or "NC".

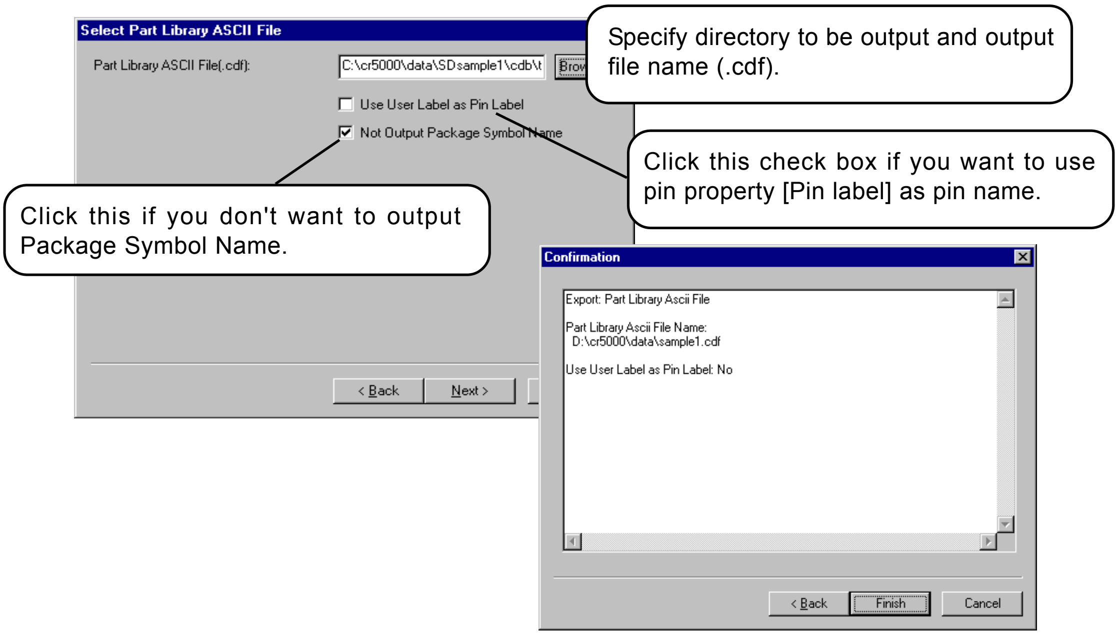
Value of "User label" is defined as default logic pin name on target device.

Power, ground, NC and default logic pin name on target device are determined by each vendor. Power, ground, NC and default logic pin name on regulation of corresponding vendor are as follows.

ACTEL	
Power, supply / ground	- VCC, GND, VSV, VPP, VKS
NC	
Default logical pin name	- UNASSIGNED, MODE
ALTERA	
Power, supply / ground	- VCCIO, VCCINT, GND, GNDINT
NC	- N.C.
Default logical pin name	- RESERVED
XILINX	
Power, supply / ground	- VCC, GND
NC	
Default logical pin name	- TIE
LATTICE	
Power, supply / ground	- GND, VCC, VCCIO
NC	
Default logical pin name	- RESET

6.Exporting Parts Library ASCII file (cdf)

Export component properties and parts properties edited in the Component Designer to Parts Library ASCII file (.cdf).
Outputted Parts Library ASCII file can be parts information of Component Library (CDB) by using "Parts Library ASCII input program".



Select Part Library ASCII File

Part Library ASCII File(.cdf): C:\cr5000\data\SDsample1\cdb\t

□ Use User Label as Pin Label

☑ Not Output Package Symbol Name

Confirmation

Export: Part Library Ascii File

Part Library Ascii File Name:
D:\cr5000\data\sample1.cdf

Use User Label as Pin Label: No

Specify directory to be output and output file name (.cdf).

Click this check box if you want to use pin property [Pin label] as pin name.

Click this if you don't want to output Package Symbol Name.

The following are corresponding items between CDB and Component Designer.

<Contents of process>

[Part name] - Exported to [Part name].

[Component tree]

- Exporting package

  Package is exported as part or pin assignment information.

- Exporting sub-components

  When the editing mode is "Normal mode", sub-components which component type is gate are exported as function or pin assignment (internal function.)
  (If there is a power box, they are exported as properties of pin assignment.)
  When the editing mode is "Split component mode", sub-components are not exported.

Properties displayed in [Component] tag.

- [Component name] - Exported as symbol name.
- [Component type] - Referred to only when export is done.
- [Gate number] - Exported as internal function number of pin assignment.
- [Function name] - Exported as function name or internal function name of pin assignment.
- Pin properties displayed in [Pin property table] - Exported as pin assignment.

Properties displayed in [Symbol property] tag. - Not exported.
Properties displayed in [Parts property] tag. - Exported as parts properties.

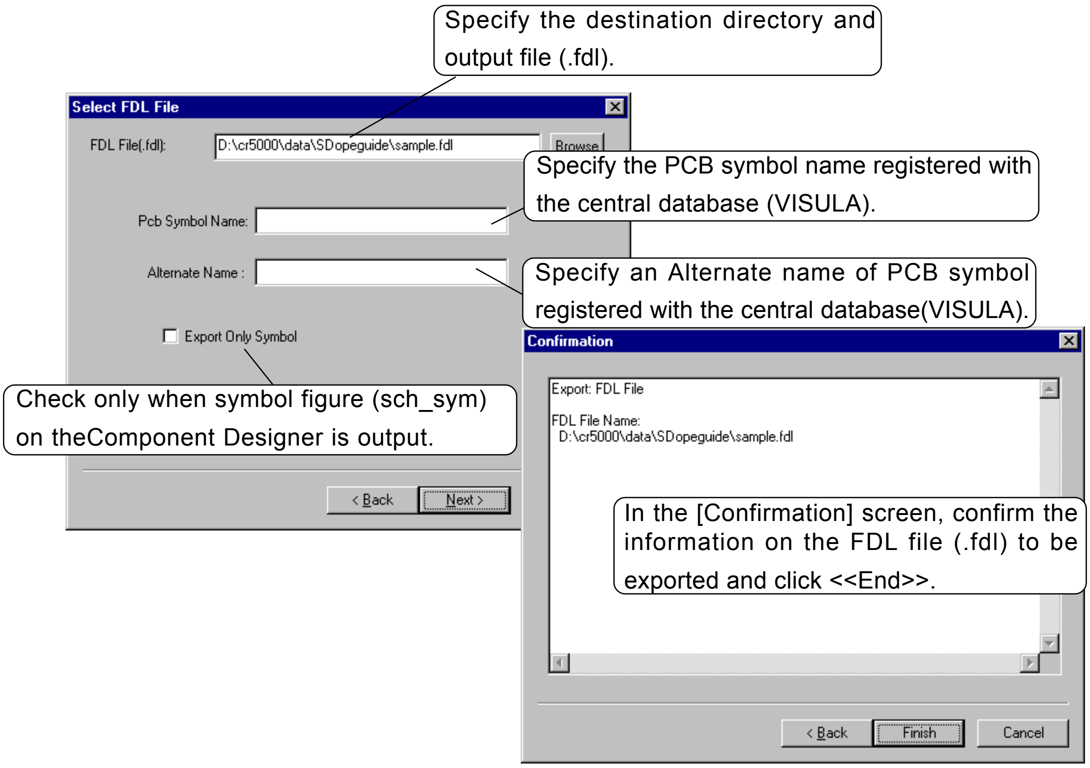
Reference

Basic process is the same as [LCDB export from Component Designer] [Generate parts library from LCDB]

For the details, see "Component Designer Online Help" and "Component Manager User's guide".

7. FDL file (VISULA) export

Export the component property and parts property edited with the Component Designer to FDL format (.fdl), which is an I/O file (ASCII) of the central database of VISULA.



Specify the destination directory and output file (.fdl).

Select FDL File

FDL File(.fdl): D:\cr5000\data\SDopeguide\sample.fdl

Pcb Symbol Name:

Alternate Name :

□ Export Only Symbol

Specify the PCB symbol name registered with the central database (VISULA).

Specify an Alternate name of PCB symbol registered with the central database(VISULA).

Check only when symbol figure (sch_sym) on theComponent Designer is output.

Confirmation

Export: FDL File

FDL File Name:
D:\cr5000\data\SDopeguide\sample.fdl

In the [Confirmation] screen, confirm the information on the FDL file (.fdl) to be exported and click <<End>>.

After export, each property will be converted as follows:

<Contents of process>

The structure of FDL file's main elements to be output by Component Designer.

```
(central
(mark 15)
(issue 1)
   (entities
    (part ...)
     (part_detail ...)
       (element ...)
       (package ...)
       (sch_sym ...)
   )
)
```

Properties to be exported for each element:

-(part) -Parts name, CDB names and other entered parts properties are created.

-(part_detail) -Gate configuration and pin relation of package and element are created.

-(element) -The pin ID is created as an element pin number. Information on equivalence definition of pins is not output.

-(package) -The pin number is created as package pin number.
The pin name is created as the package pin name.

-(sch_sym) -The symbol figure is created.

Exporting the split symbol selected on the schematic

You can edit the symbol retrieved from the schematic as the split symbol, and export it without use of the wizard.

Note that this function is available only when the following conditions are satisfied.

1. When the Component Designer is started by specifying the symbol on the schematic.
2. If a split symbol exists.
3. If the edit mode is "Split Mode".

<Export file>

When you select a normal symbol and start the Component Designer :

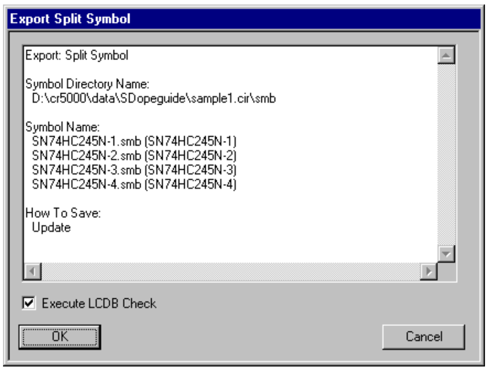
Export the symbol to "smb" directory in the same directory as the circuit directory where the symbol is selected.

When you select a split symbol and start the Component Designer :

Export the symbol to the symbol directory containing the split symbol file selected on the schematic.

<Operation>

Select the symbol on the schematic, specify [Utility] - [Component Designer] and start the Component Designer.
Specify [File] - [Export Split Symbol] – [Overwrite/Save] or [Create New].
To check LCDB for consistency immediately before export, check [LCDB Check].



```
Export: Split Symbol

Symbol Directory Name:
  D:\cr5000\data\SDopeguide\sample1.cir\smb

Symbol Name:
  SN74HC245N-1.smb (SN74HC245N-1)
  SN74HC245N-2.smb (SN74HC245N-2)
  SN74HC245N-3.smb (SN74HC245N-3)
  SN74HC245N-4.smb (SN74HC245N-4)

How To Save:
  Update
```

☑ Execute LCDB Check

Click [OK].

If a file name or a version is repeated during LCDB check, new files will be created by automatically incrementing the file name in the same manner as in the case where a new symbol is created by exporting the symbol by means of the export wizard.
(For how to create the symbol file name to be automatically generated, see "Exporting the symbol" in the online help file.)

When the number of group names are decreased due to "overwrite and save", symbols having an unused group name will be deleted.

8.Other functions

Here you can see the Component Designer and other functions.

* To save the Component Designer file

You can save data being edited in the Component Designer as Component Designer file. Creating file extension is '.cpd'.

<Operation process>

[File] - [Save as]
pecify saving directory and file name in the dialog, then click <<Save>>.

* Component Designer utility functions.

Component Designer equips the following functions as utility.

- Creating schematic data from symbol file

  Auto-generate circuit directory (.cir) and circuit block file (.blk) from symbol file. It is created when internal circuit is created based on circuit block symbol.

- Generating LCDB from schematic data.

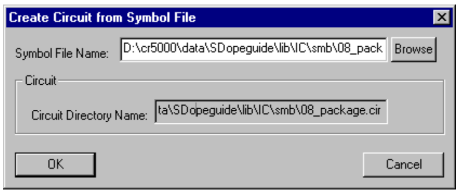
  Generate parts information used in specified schematic data as LCDB.

<Operation process>

- Creating schematic data from symbol file

  [Utility] - [Create schematic data from symbol file]

  On the dialog, specify symbol file name to be referred to and schematic directory name to be output.



Symbol File Name: D:\cr5000\data\SDopeguide\lib\IC\smb\08_pack

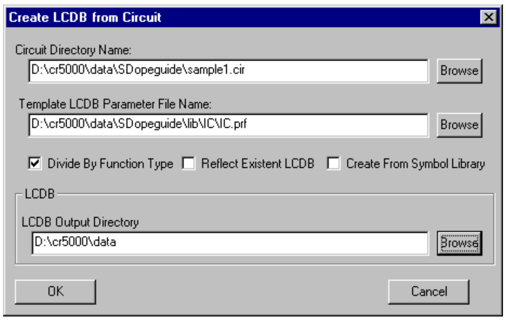
Circuit

Circuit Directory Name: ta\SDopeguide\lib\IC\smb\08_package.cir

- Generating LCDB from schematic data.

  [Utility] - [Generate LCDB from schematic data]

  On the dialog, specify target schematic directory, template LCDB name to be referred to and output directory name.



Circuit Directory Name: D:\cr5000\data\SDopeguide\sample1.cir

Template LCDB Parameter File Name: D:\cr5000\data\SDopeguide\lib\IC\IC.prf

☑ Divide By Function Type ☐ Reflect Existent LCDB ☐ Create From Symbol Library

LCDB

LCDB Output Directory: D:\cr5000\data

9.Split Symbols Precautions

1.Symbolproperties

Symbols of split components are not registered in LCDB. You need to define property values to specify parts in LCDB ["Part name (partName)" or "Part number" (partNumber)] in symbols.

2.Placement in schematic data.

Since split components symbols are not registered in LCDB, parts placement function is not available. Use symbol figure placement function.
-When "Parts Rule Search" is ON, you can add parts information defined in LCDB by specifying part name. In this case, turn "Consistency check between part and symbol" OFF in the details setting for searching parts.
-When "Parts Rule Search" is OFF, parts information defined in LCDB is not automatically added. To take information from LCDB, you need to perform "Reload parts information" after editing schematic data.

Split component symbols cannot be multiple symbols.

3.Reference

Split component symbols that compose one part must have same reference value.
When symbol is placed, reference can be added by [Parts Search dialog].
(Notautomaticallygenerated)
After placement, it can be added by [Change Attribute dialog].
Reference cannot be generated in Circuit Reference Allocator.

4.Rule check

In Circuit Rule Checker, split components symbols are not checked as open gate as they are not registered in LCDB. (In Sheet Rule Checker, they are checked as same as normal parts)

5.ForwardAnnotation

When operating with PWS, net type supports CCF only.
When operating with Board Designer, dedicated flag (isvGate=YES) is automatically added to each split symbol by the Component Designer and output as package symbols.

6.BackAnnotation

Since net type supports CCF only in operation with PWS, back annotation is not available. Split components, among parts information described in back annotation file from Board Designer, are not supported for back-annotation.