Chapter 23

Using Meta I/O

Meta I/O converts I/O buffer models between Star-Hspice format and the I/O Buffer Information Specification (IBIS) Version 2.1 format. Meta I/O is not part of Star-Hspice, but is an optional product that operates with the following software:

- Star-Hspice, Versions 95.2 and higher
- Meta Encryptor, Version 1.0
- SPICE to IBIS, Version 1.0, from the University of North Carolina

This chapter explains how to use Meta I/O, gives an example of a Meta I/O application, and describes Meta I/O commands. The following topics are covered in this chapter:

Understanding the Design Flow

Meta I/O Version features include the following:

- Intellectual property protection with IBIS buffer description format
- Library management for improved productivity
- Automatic library conversion and Star-Hspice verification
- Optional Meta Encryption for models other than I/O buffers

You must have the full Star-Hspice product set on your system for Meta I/O to run successfully.

Figure 23-1: shows the design data flow for the IC vendor. The process starts with Star-Hspice or SPICE 2g6 transistor level buffer models. These models are converted by the “SPICE to IBIS” program into IBIS description files. The IBIS description files can then be converted into Star-Hspice behavioral buffer models and verified against the original transistor buffers. Optionally, any Star-Hspice subcircuit can be encrypted with the Meta Encryptor, which is a separate Avant! product.

Figure 23-2: shows the design data flow for the system designer. The process starts with Star-Hspice IBIS templates and IBIS description files from the IC vendor. These files are converted by the “IBIS to Star-Hspice” program into Star-Hspice behavioral level models.

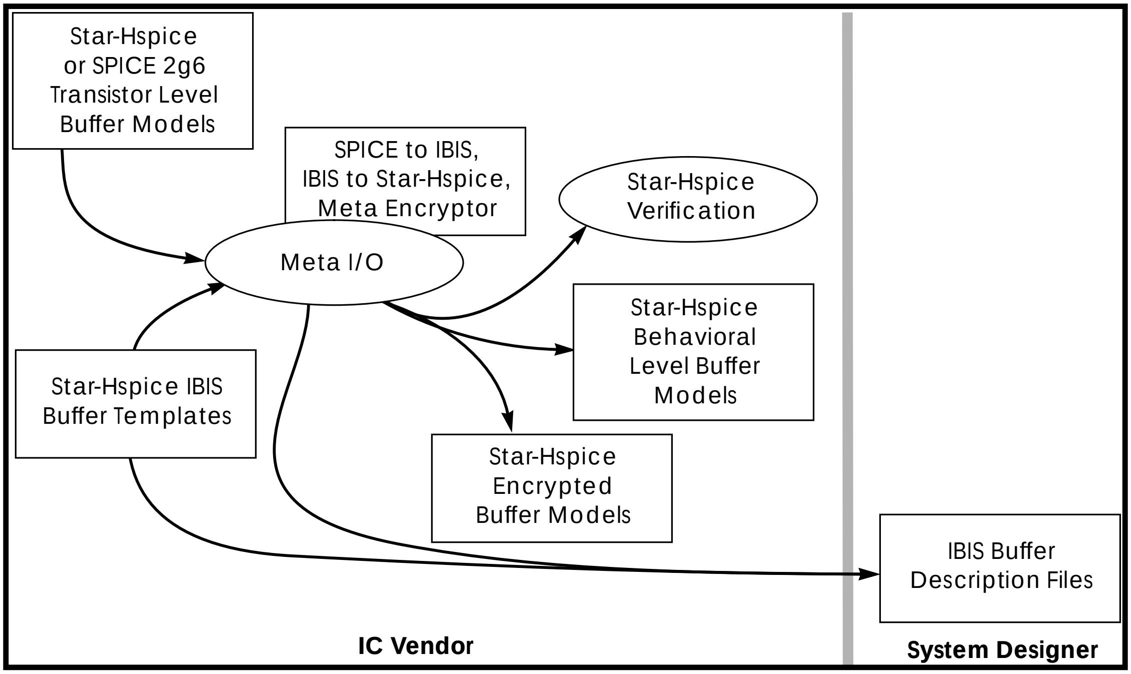

Figure 23-1: IC Vendor Design Data Flow Using Meta I/O

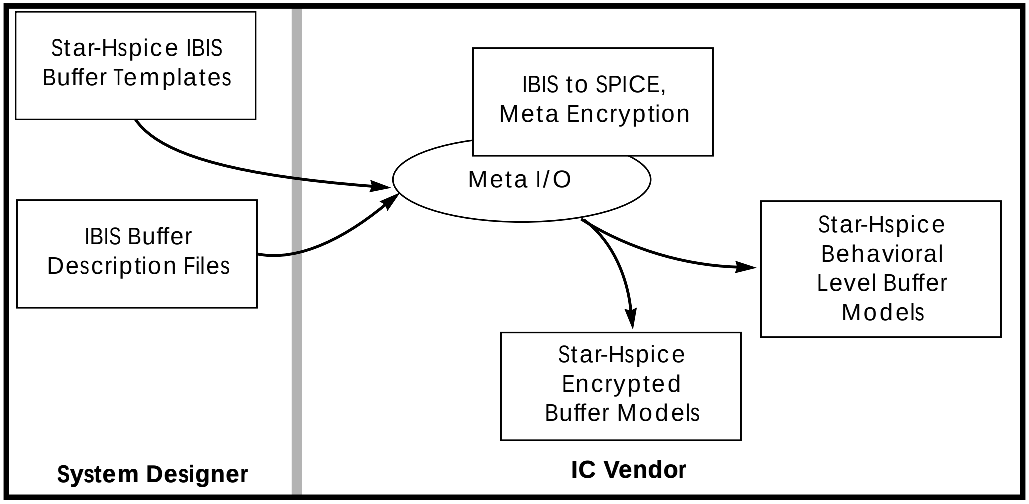

Figure 23-2: – System Design Data Flow Using Meta I/O

Getting Started with IBIS

The example in this section introduces the basic concepts of the Meta I/O product. This example shows how you can use Meta I/O to:

- Select and convert an IBIS description to a set of Star-Hspice behavioral buffers
- Verify the final results with Star-Hspice

The example design is a simple three-buffer component named IbisTest.

Figure 23-3: illustrates the library structure for the IbisTest example. *IbisLib*, *Templates*, *Star-HspiceLib*, and *VerDir* are directories used in the example.

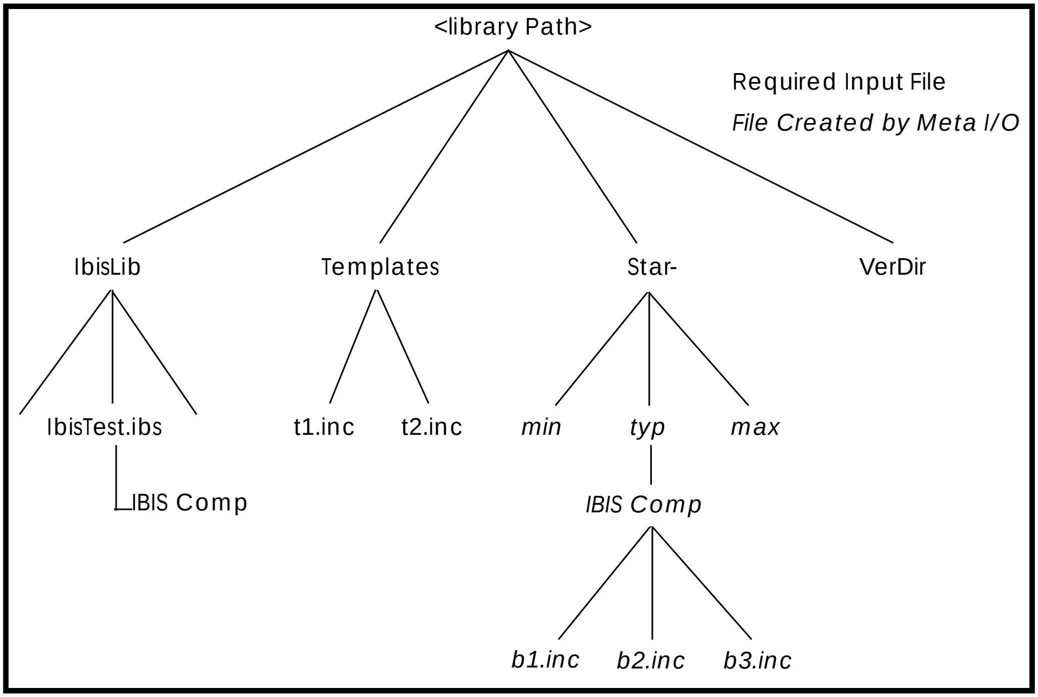

Figure 23-3: – IbisTest Example Library Structure

The Meta I/O Script

Meta I/O derives its control information from a Meta I/O script file that contains all commands required to convert and verify an IBIS-based component library. The script is a sequence of Meta I/O commands in an ASCII text file that you can create using a text editor. The script for this example is presented later in this chapter. All of the Meta I/O commands also are described later in this chapter.

Conversion Function Setup

Set the conversion function to “IBIS to Star-Hspice”. Use the SetConvFunc command with the i2h keyword:

```
SetConvFunc     i2h;
```

Select the desired data from the IBIS description file. Use the SetConvType command with the min keyword:

```
SetConvType     typ;
```

Library Directory Setup

Use the *SetIbisDescripDir* command to select the source of the IBIS description file. This command accepts a string parameter that defines the directory containing the IBIS description file for the IBIS to Star-Hspice process.

Use the *SetIbisTemplateDir* command to select the source directory for the behavioral level buffer model template files. This command accepts a string parameter that defines the directory containing the template files for the IBIS to Star-Hspice process.

Use the *SetStar-HspiceDir* command to select the destination for the behavioral level buffer model files. This command accepts a string parameter that defines the directory to hold the files created during the IBIS to Star-Hspice process.

Use the *SetVerDir* command to select the location for running the Star-Hspice verification process. This command accepts a string parameter that defines the directory to hold the Star-Hspice intermediate files during verification.

The section of the IbisTest script that sets up directories is:

```
  SetIbisDescripDir./lib/IbisLib;
 SetIbisTemplateDir   ./lib/Templates;
 SetStar-HspiceDir    ./lib/Star-HspiceLib;
 SetVerDir            ./lib/Verify;
```

Template Selection

Meta I/O provides a method of connecting a given IBIS buffer model to an executable Star-Hspice behavioral template with the *SetDefTemplate* and *TemplateMap* template selection commands.

Use the *SetDefTemplate* command to define the filename for all IBIS buffer model types that do not have a special template model. In this example, Meta I/O expects to find the d2a.inc template file in the directory specified by the value of the SetIbisTemplateDir function.

Use the TemplateMap command to specify the filename for specific IBIS buffer model types that require a special template model. In this example, Meta I/O expects to find the *d2a.inc* template file in the directory specified by the value of the *SetIbisTemplateDir* function and uses that template file to convert all I/O_open_drain type models.

```
  SetDefTemplate d2a.inc;
  TemplateMap I/O_open_drain  d3.inc;
```

For further information on *SetDefTemplate* and Requirements, please see the to“Using the Meta I/O Commands” on page 23-10.

File Selection In Meta I/O

Use the *SetFileMask*, *SetInFileSuffix*, *SetOutFileSuffix, and SelectFiles* commands to select the single SPICE transistor level model:

```
SetFileMask       IbisTest;
SetInFileSuffix   .ibs;
SetOutFileSuffix  .inc;
SelectFiles;
```

Use the *SetFileMask* command to describe which files to convert. Meta I/O allows the use of file wildcards in filename, to select multiple files for conversion. In this example, no wildcards are used.

Use the *SetInFileSuffix* command to select only files that end in .ibs.

Use the *SetOutFileSuffix* command to produce output files that have the suffix .inc.

Use the *SelectFiles* command to iterate through the library directory and place each file that matches the select criteria in its internal selected files list.

The selection process is additive, which allows you to set a selection criteria, select files, change criteria, and append new files to the existing list.

Use the *UnselectFiles* command to clear the file selection list.

Conversion

The command Iterate causes Meta I/O to process each file in the selection list according to the conversion function specified earlier. As each file is converted, warning and error messages are echoed to the screen (through stdout) providing a record of the actions taken.

You can select a set of files and then issue many different conversion or verification functions on that list without reselecting the files for each operation.

IbisTest Script Listing

```
# IbisTest Conversion Example
#
SetConvFunc          i2h;
SetConvType          typ;
SetIbisDescripDir    ./lib/IbisLib;
SetIbisTemplateDir   ./lib/Templates;
SetStar-HspiceDir    ./lib/Star-HspiceLib;
SetVerDir            ./lib/Verify;
SetDefTemplate       d2a.inc;
SetFileMask          IbisTest;
SetInFileSuffix      .ibs;
SetOutFileSuffix     .inc;
SelectFiles;
Iterate;
UnselectFiles;
```

Running the Example

Enter the following command to run the IbisTest example:

```
metaio -f IbisTest
```

To capture the results in a log file, use the following command:

```
metaio -f IbisTest > IbisTest.log
```

The Results

The following is a log file that is produced by the example run:

```
0 : #                Meta I/O
0 : #
0 : #    Copyright 1998 Avant!, Inc.
0 : #           All Rights Reserved
0 : #
1 : #
2 : # IbisTest
3 : #
4 :
5 : SetConvFunc             i2h;
6 : SetConvType             typ;
```

```
     7 :
     8 : SetIbisDescripDir       ./lib/IbisLib;
     9 : SetIbisTemplateDir      ./lib/Templates;
    10 : SetStar-HspiceDir            ./lib/Star-Hspice;
    11 :
    12 : SetDefTemplate               d2a.inc;
    13 : Requirement             I/O  d3a.inc;
    14 :
    15 : SetFileMask             *;
    16 : SetInFileSuffix         .ibs;
    17 : SetOutFileSuffix        .inc;
    18 :
    19 : SelectFiles;
    20 : Iterate;
       : i2hConvert = ./lib/IbisLib/ibistest.ibs

Checking ibistest.ibs for IBIS 2.1 <Beta> Compatibility...

WARNING (line  108) - Pulldown Maximum data is non-monotonic
WARNING (line  109) - Pulldown Typical data is non-monotonic
WARNING (line  109) - Pulldown Minimum data is non-monotonic
WARNING (line  152) - Pullup Maximum data is non-monotonic
WARNING (line  155) - Pullup Typical data is non-monotonic
WARNING (line  155) - Pullup Minimum data is non-monotonic
WARNING (line  279) - Pulldown Maximum data is non-monotonic
WARNING (line  280) - Pulldown Typical data is non-monotonic
WARNING (line  281) - Pulldown Minimum data is non-monotonic

IBIS Structure Okay

*** Warning : Line  20 : Can’t get IBIS Package Models for
some reason!

Errors  : 0
Warnings: 9

File Passed

    21 : UnSelectFiles;
    22 :
    23 :
```

Using the Meta I/O Commands

This section is a Meta I/O command reference. All commands, their arguments, and syntax are described.

General Form and Conventions

The general form of a Meta I/O command is:

```
  <CommandName> [<Arg1>[<Arg2> ...]];
```

Every command must be terminated by a semicolon (;).

Command names are case insensitive – they can be entered in any combination of upper- and lowercase letters. Arguments are also case insensitive, except for file names and directory names.

You can enter comments by preceding the comment text by a pound sign (#). A comment can start at the beginning of a line or following the semicolon, as in the following examples:

```
#
# This is a comment
#
SetIbisDescripDir ./lib/Ibis; # So is this
```

Wild Card Characters

Meta I/O supports the use of wild cards for file selection. The allowable wild cards are the asterisk and the character range.

The Asterisk

Use the asterisk (*) to match zero or more of any characters.

The wild card

```
  buf*
```

matches

```
  buf
```

```
buff
buffer
buf1
```

The Range

Use the Range wild card to match a single ASCII character . The range wild card is delimited by the square brackets and the hyphen character.

The syntax for the range wild card is:

```
‘[’ <Char> ‘-’ <Char> ‘]’
```

A range wild card

```
buf[0-2]
```

matches

```
buf0
buf1
buf2
```

but does not match

```
buff
buf6
```

Directory Management Commands

SetIbisDescripDir <Path>;

Sets the directory that Meta I/O refers to when accessing IBIS description files for conversion.

The <Path> parameter is a string.

GetIbisDescripDir;

Returns a string describing the directory where Meta I/O accesses IBIS description files.

SetIbisTemplateDir <Path>;

Sets the directory that Meta I/O refers to when accessing IBIS template files for use in conversion.

The <Path> parameter is a string.

GetIbisTemplateDir;

Returns a string describing the directory where Meta I/O accesses IBIS template files.

SetStar-HspiceDir <Path>;

Sets the directory that Meta I/O refers to when producing Star-Hspice results files during conversion.

The <Path> parameter is a string.

The actual path is suffixed with the current value of the function *GetConvType*.

GetStar-HspiceDir;

Returns a string describing the directory where Meta I/O writes Star-Hspice files during conversion.

The returned path does not include the current value of the function *GetConvType*.

SetVerDir <Path>;

Sets the directory that Meta I/O refers to when running Star-Hspice verifications after conversion.

The <Path> parameter is a string.

GetVerDir;

Returns a string describing the directory where Meta I/O stores temporary files.

SetTmpDir <Path>;

Sets the directory that Meta I/O refers to when running Star-Hspice verifications.

The <Path> parameter is a string.

GetTmpDir;

Returns a string that describing the directory where Meta I/O stores temporary files.

File Management Commands

SetFileMask <WildCard>;

Sets the file selection mask (a regular expression) for files in the current directory that are subject to the current conversion function.

The parameter <WildCard> for example, might be a string of the form “74ls*”, which matches all files starting with the literal “74ls” that contain zero or more characters.

GetFileMask;

Returns a string representing the current value of file selection mask.

SetInFileSuffix <WildCard>;

Sets the input file suffix mask regular expression thatselects files in the current input directory subject to the current conversion function.

The parameter <WildCard>, for example, might be a string of the form “.ibs”, which would match all files with the literal “.ibs” suffix that contain one or more characters up to the literal “.”.

GetInFileSuffix;

Returns a string that represents the current value of the input file suffix mask.

SetOutFileSuffix <String>;

Sets the output file suffix mask (a regular expression) used to create files in the current output directory subject to the current conversion function.

The parameter <String>, for example, might be a string of the form ".inc", which matches all files with the literal ".inc" suffix that contain one or more characters up to the literal ".".

GetOutFileSuffix;

Returns a string that represents the current value of the output file suffix mask.

SelectFiles;

Selects the files in the current conversion directory based on the values of the:

- applicable directory
- SetFileMask
- SetInFileSuffix strings

The function returns the number of files in the selection.

Example

```
SetIbisDescDir/lib/dir1;
SetFileMask 74ls*;
SetInFileSuffix.ibs;
SelectFiles;
```

with the following files in /lib/dir1:

```
7400.ibs
7402.ibs
74ls00.ibs
74ls02.ibs
74ls04.ibs
83333.ibs
83334.ibs
```

sets the current selection list to contain

```
74ls00.ibs
74ls02.ibs
74ls04.ibs
```

To append the new filenames to the selection list, change the wild cards and reissue a SelectFiles command.

UnSelectFiles;

Clears the current selection list.

Conversion Management Commands

SetConvFunction <ConvFunc>;

This command sets the conversion function to one of the predefined conversions: Spice to IBIS, IBIS to Star-Hspice, Star-Hspice Encryption, and Verification.

Table 23-1: – SetConvFunc Conversion Types

Conversion Function	Token
Spice to IBIS	s2i
IBIS to Star-Hspice	i2h
Encryption	Enc
Verification	Ver

GetConvFunction;

Returns a string that indicates the current conversion function.

SetConvType <ConvType>;

Use this command to select between the three corners of the IBIS description. These corners are defined as Minimum, Typical, and Maximum and represent different values (scalars and vectors) within the IBIS description.

Example:

```
...
SetConvType Min;
Iterate;
...
SetConvType Typ;
Iterate;
...
```

For each of the files that match the selection criteria, this example produces minimum and typical models.

In addition to the selection of the corner, this function should also provide the directory paths outlined in Figure 23-2:. The conversion type string is placed in the output directory path:

```
<OutDir>/<ConvType>/<Compname>/<Buffer><OutFileSuffix>
```

Table 23-2: SetConvType Conversion Types

Conversion Type	Token
All	All
Minimum	Min
Typical	Typ
Maximum	Max

GetConvType;

Returns a string that indicates the current conversion type.

Iterate;

Successively applies the conversion function to each file in the selection list.

Direct Iteration

The Direct Iteration feature allows code within the compiled body of the Meta I/O API to be directly executed. Some functions that exist only within Meta I/O and not as stand-alone programs require successive application to the input files.

This method is primarily used for the IBIS to Star-Hspice behavioral conversion. Because the IBIS Golden Parser is built into Meta I/O directly, an iteration through this conversion function causes Meta I/O to read and parse the IBIS Description, map the relevant pieces into the Star-Hspice behavioral template, and emit the final model into the specified directory tree.

Process Fork

This is the standard process fork found in most operating systems. The primary method is through the system() call.

The appropriate parameters are passed with each call to system().

Execution of the Spice to IBIS and encryption functions is handled by this method.

Verification Management Commands

To ensure accurate conversion of transistor level to behavioral level buffers, Meta I/O provides an automated verification method which casts each type of circuit into a user-defined test harness before Star-Hspice execution.

The test harness provides identical stimuli to each buffer and measures the final outcome. The results can be viewed with MetaWaves or by other user validation means.

The test harnesses are stored in the same directory as the Star-Hspice transistor level modes and have a different suffix to distinguish them from the buffer models.

Name matching is used to connect the test harness with the components under test.

System Management Commands

ExitMetaIoApi <ReturnValue>;

Cleans up and returns the integer <ReturnValue> to calling environment.

Note: ExitMetaIoApi is not implemented in revision 1.0.

```
* A simple test harness

.INCLUDE ‘Voltages’
.INCLUDE ‘Stimili’
.INCLUDE ‘Temp’
.OPTIONS Search= ‘/lib/Star-HspiceLib/typ/comp’

* IBIS Component Comp1
*
x1 ..... b1
x2 ..... b2
x3 ..... b3

.TRAN ....
.OPTIONS Post
.END
```