

IBIS

(I/O Buffer Information Specification)

Version 5.0

Ratified August 29, 2008

Contents (sample)

- 1 General Introduction
- 2 Statement of Intent
- 3 General Syntax Rules and Guidelines
 - 3.1 Keyword Hierarchy
- 4 File Header Information
- 5 Component Description
- 6 Model Statement
 - 6.1 Add Submodel Description
 - 6.2 Multi-Lingual Model Extensions
 - 6.3 Algorithmic Modeling Interface (AMI)
- 7 Package Modeling
- 8 Electrical Board Description
- 9 Notes on Data Derivation Method
- 10 Notes on Algorithmic Modeling Interface and Programming Guide
 - 10.1 Overview
 - 10.2 Applications Scenarios
 - 10.2.1 Linear , Time-invariant Equalization Model
 - 10.2.2 Nonlinear, and / or Time-variant Equalization Model
 - 10.2.3 Reference system Analysis Flow
 - 10.3 Function Signatures
 - .
 - .
 - .
- 11 EMI Parameters

1 General Introduction

This section gives a general overview of the remainder of this document.

Sections 2 and 3 contain general information about the IBIS versions and the general rules and guidelines. Several progressions of IBIS documents are referenced in Section 2 and in the discussion below. They are IBIS Version 1.1 (ratified August 1993), IBIS Version 2.1 (ratified as ANSI/EIA-656 in December 1995), IBIS Version 3.2 (ratified as ANSI/EIA-656-A in October 1999 and renewed on August 17, 2005), IBIS Version 4.2 (ratified as ANSI/EIA-656-B on March 1, 2007), and IBIS Version 5.0 (ratified on August 29, 2008)

The functionality of IBIS follows in Sections 4 through 8. Sections 4 through 6 describe the format of the core functionality of IBIS Version 1.1 and the extensions in later versions. The data in these sections are contained in .ibs files. Section 7 describes the package model format of IBIS Version 2.1 and a subsequent extension. Package models can be formatted within .ibs files or can be formatted (along with the Section file header keywords) as .pkg files. Section 8 contains the Electrical Board Description format of IBIS Version 3.2. Along with Section 4 header information, electrical board descriptions must be contained in separate .ebd files.

Sections 6c, 10, and 11 are new in IBIS Version 5.0 and contain reference and modeling information related to the algorithmic modeling interface support, and EMI parameters

Section 9 contains some notes regarding the extraction conditions and data requirements for IBIS files. This section focuses on implementation conditions based on measurement or simulation for gathering the IBIS compliant data.

2 Statement of Intent

In order to enable an industry standard method to electronically transport IBIS modeling data between semiconductor vendors, EDA tool vendors, and end customers, this template is proposed. The intention of this template is to specify a consistent format that can be parsed by software, allowing EDA tool vendors to derive models compatible with their own products.

One goal of this template is to represent the current state of IBIS data, while allowing a growth path to more complex models/methods (when deemed appropriate). This would be accomplished by a revision of the base template, and possibly the addition of new keywords or categories.

Another goal of this template is to ensure that it is simple enough for semiconductor vendors and customers to use and modify, while ensuring that it is rigid enough for EDA tool vendors to write reliable parsers.

Finally, this template is meant to contain a complete description of the I/O elements on an entire component. Consequently, several models will need to be defined in each file, as well as a table that equates the appropriate buffer to the correct pin and signal name.

Version 5.0 of this electronic template was finalized by an industry-wide group of experts representing various companies and interests. Regular “EIA IBIS Open Forum” meetings were held to accomplish this task.

Commitment to Backward Compatibility. Version 1.0 is the first valid IBIS ASCII file format. It represents the minimum amount of I/O buffer information required to create an accurate IBIS model of common CMOS and bipolar I/O structures. Future revisions of the ASCII file will add items considered to be “enhancements” to Version 1.0 to allow accurate modeling of new, or other I/O buffer structures. Consequently, all future revisions will be considered supersets of Version 1.0, allowing backward compatibility. In addition, as modeling platforms develop support for revisions of the IBIS ASCII template, all previous revisions of the template must also be supported.

Version 1.1 update. The file “ver1_1.ibs” is conceptually the same as the 1.0 version of the IBIS ASCII format (ver1_0.ibs). However, various comments have been added for further clarification.

Version 2.0 update. The file “ver2_0.ibs” maintains backward compatibility with Versions 1.0 and 1.1. All new keywords and elements added in Version 2.0 are optional. A complete list of changes to the specification is in the IBIS Version 2.0 Release Notes document (“ver2_0.rn.txt”).

Version 2.1 update. The file “ver2_1.ibs” contains clarification text changes, corrections, and two additional waveform parameters beyond Version 2.0.

Version 3.0 update. The file “ver3_0.ibs” adds a number of new keywords and functionality. A complete list of functions can be found on eda.org under /pub/ibis/birds/birddir.txt showing the approved Buffer Issue Resolution Documents (BIRDs) that have been approved for Version 3.0.

Version 3.1 update. The file “ver3_1.ibs” contains a major reformatting of the document and a simplification of the wording. It also contains some new technical enhancements that were unresolved when Version 3.0 was approved.

Version 3.2 update. The file “ver3_2.ibs” adds more technical advances and also a number of editorial changes documented in 12 BIRDs and also in responses to public letter ballot comments.

Version 4.0 update. This file “ver4_0.ibs” adds more technical advances and a few editorial changes documented in 11 BIRDs.

Version 4.1 update. This file “ver4_1.ibs” adds more technical advances and a few editorial changes documented in 10 BIRDs.

Version 4.2 Update. This file “ver4_2.ibs” adds more technical advances and and some editorial changes documented in 13 BIRDs.

Version 5.0 Update. This file “ver5_0.ibs” adds more technical advances and and some editorial changes documented in 10 BIRDs.

3 General Syntax Rules and Guidelines

This section contains general syntax rules and guidelines for ASCII IBIS| files:

1. The content of the files is case sensitive, except for reserved words and keywords.
2. The following words are reserved words and must not be used for any other purposes in the document:

POWER - reserved model name, used with power supply pins,

GND - reserved model name, used with ground pins,

NC - reserved model name, used with no-connect pins,

NA - used where data not available, CIRCUITCALL - used for circuit call references in Section 6b.

3. To facilitate portability between operating systems, file names used in the IBIS file must only have lower case characters. File names should have a basename of no more than forty (40) characters followed by a period ('.'), followed by a file name extension of no more than three characters. The file name and extension must use characters from the set (space, ' ', 0x20 is not included):

a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 _ ^ \$ ~ ! # % & - { }) (@ `

The file name and extension are recommended to be lower case on systems that support such names.

4. A line of the file may have at most 120 characters, followed by a line termination sequence. The line termination sequence must be one of the following two sequences: a linefeed character, or a carriage return followed by linefeed character.
5. Anything following the comment character is ignored and considered a comment on that line. The default "|" (pipe) character can be changed by the keyword [Comment Char] to any other character. The [Comment Char] keyword can be used anywhere in the file as desired.
6. Keywords must be enclosed in square brackets, [], and must start in column 1 of the line. No space or tab is allowed immediately after the opening bracket '[' or immediately before the closing bracket ']'. If used, only one space (' ') or underscore ('_') character separates the parts of a multi-word keyword.
7. Underscores and spaces are equivalent in keywords. Spaces are not allowed in subparameter names.

8. Valid scaling factors are:

T = tera	k = kilo	n = nano
G = giga	m = milli	p = pico
M = mega	u = micro	f = femto

When no scaling factors are specified, the appropriate base units are assumed. (These are volts, amperes, ohms, farads, henries, and seconds.) The parser looks at only one alphabetic character after a numerical entry, therefore it is enough to use only the prefixes to scale the parameters. However, for clarity, it is allowed to use full abbreviations for the units, (e.g., pF, nH, mA, mOhm). In addition, scientific notation IS allowed (e.g., 1.2345e-12).

9. The I-V data tables should use enough data points around sharply curved areas of the I-V curves to describe the curvature accurately. In linear regions there is no need to define unnecessary data points.
10. The use of tab characters is legal, but they should be avoided as much as possible. This is to eliminate possible complications that might arise in situations when tab characters are automatically converted to multiple spaces by text editing, file transferring and similar software. In cases like that, lines might become longer than 120 characters, which is illegal in IBIS files.
11. Currents are considered positive when their direction is into the component.
12. All temperatures are represented in degrees Celsius.
13. Important supplemental information is contained in the last section, "NOTES ON DATA DERIVATION METHOD", concerning how data values are derived.
14. Only ASCII characters, as defined in ANSI Standard X3.4-1986, may be used in an IBIS file. The use of characters with codes greater than hexadecimal 07E is not allowed. Also, ASCII control characters (those numerically less than hexadecimal 20) are not allowed, except for tabs or in a line termination sequence. As mentioned in item 10 above, the use of tab characters is discouraged.

3.1 Keyword Hierarchy

ibs FILE

File Header Section

- [IBIS Ver]
- [Comment Char]
- [File Name]
- [File Rev]
- [Date]
- [Source]
- [Notes]
- [Disclaimer]
- [Copyright]

[Component]

Si_location, Timing_location

- [Manufacturer]
- [Package]
- [Pin]

R_pkg, L_pkg, C_pkg
signal_name, model_name, R_pin,
L_pin, C_pin

[Package Model]

[Alternate Package Models]

[End Alternate Package Models]

[Pin Mapping]

pulldown_ref, pullup_ref,
gnd_clamp_ref, power_clamp_ref,
ext_ref

[Diff Pin]

inv_pin, vdiff, tdelay_typ,

tdelay_min, tdelay_max

[Series Pin Mapping]

pin_2, model_name,

function_table_group

[Series Switch Groups]

On, Off

[Node Declarations]

[End Node Declarations]

[Circuit Call]

Signal_pin, Diff_signal_pins,
Series_pins, Port_map

[End Circuit Call]

4 File Header Information

Keyword: [IBIS Ver]

Required: Yes

Description: Specifies the IBIS template version. This keyword informs electronic parsers of the kinds of data types that are present in the file.

Usage Rules: [IBIS Ver] must be the first keyword in any IBIS file. It is normally on the first line of the file, but can be preceded by comment lines that must begin with a "|".

[IBIS Ver] 5.0 | Used for template variations

Keyword: [Comment Char]

Required: No

Description: Defines a new comment character to replace the default "|" (pipe) character, if desired.

Usage Rules: The new comment character to be defined must be followed by the underscore character and the letters "char".

For example:

"_char" redundantly redefines the comment character to be the pipe character. The new comment character is in effect only following the [Comment Char] keyword. The following characters MAY be used:

! " # \$ % & ' () * , : ; < > ? @ \ ^ ` { | } ~

Other Notes: The [Comment Char] keyword can be used anywhere in the file, as desired.

[Comment Char] _char

Keyword: [File Name]

Required: Yes

Description: Specifies the name of the IBIS file.

Usage Rules: The file name must conform to the rules in paragraph 3 of Section 3, GENERAL SYNTAX RULES AND GUIDELINES. In addition, the file name must use the extension ".ibs", ".pkg", or or ".ebd". The file name must be the actual name of the file.

[File Name] ver5_0.ibs

5 Component Description

Keyword: [Component]

Required: Yes

Description: Marks the beginning of the IBIS description of the integrated circuit named after the keyword.

Sub-Params: Si_location, Timing_location

Usage Rules: If the .ibs file contains data for more than one component, each section must begin with a new [Component] keyword. The length of the component name must not exceed 40 characters, and blank characters are allowed.

NOTE: Blank characters are not recommended due to usability issues.

Si_location and Timing_location are optional and specify where the Signal Integrity and Timing measurements are made for the component. Allowed values for either subparameter are 'Die' or 'Pin'. The default location is at the 'Pin'.

```
[Component]    7403398 MC452
|
Si_location     Pin      | Optional subparameters to give measurement
Timing_location Die      | location positions
```

Keyword: [Manufacturer]

Required: Yes

Description: Specifies the name of the component's manufacturer.

Usage Rules: The length of the manufacturer's name must not exceed 40 characters (blank characters are allowed, e.g., Texas Instruments). In addition, each manufacturer must use a consistent name in all .ibs files.

```
[Manufacturer] Intel Corp.
```

Keyword: [Package]

Required: Yes

Description: Defines a range of values for the default packaging resistance, inductance, and capacitance of the component pins.

Sub-Params: R_pkg, L_pkg, C_pkg

Usage Rules: The typical (typ) column must be specified. If data for the other columns are not available, they must be noted with "NA".

Other Notes: If RLC parameters are available for individual pins, they can be listed in columns 4-6 under keyword [Pin]. The values listed in the [Pin] description section override the default values defined here. Use the [Package Model] keyword for more complex package descriptions. If defined, the [Package Model] data overrides the values in the [Package] keyword. Regardless, the data listed under the [Package] keyword must still contain valid data.

```
[Package]
| variable      typ          min          max
R_pkg           250.0m       225.0m       275.0m
L_pkg           15.0nH       12.0nH       18.0nH
C_pkg           18.0pF       15.0pF       20.0pF
```

Keyword: [Pin]

Required:

Description: Associates the component's I/O models to its various external pin names and signal names.

6 Model Statement

Keyword: [Model]

Required: Yes

Description: Used to define a model, and its attributes.

Sub-Params: Model_type, Polarity, Enable, Vinl, Vinh, C_comp, C_comp_pullup, C_comp_pulldown, C_comp_power_clamp, C_comp_gnd_clamp, Vmeas, Cref, Rref, Vref

Usage Rules: Each model type must begin with the keyword [Model]. The model name must match the one that is listed under a [Pin], [Model Selector] or [Series Pin Mapping] keyword and must not contain more than 40 characters. A .ibs file must contain enough [Model] keywords to cover all of the model names specified under the [Pin], [Model Selector] and [Series Pin Mapping] keywords, except for those model names that use reserved words (POWER, GND and NC).

Model_type must be one of the following:

Input, Output, I/O, 3-state, Open_drain, I/O_open_drain, Open_sink, I/O_open_sink, Open_source, I/O_open_source, Input_ECL, Output_ECL, I/O_ECL, 3-state_ECL, Terminator, Series, and Series_switch.

For true differential models documented under Section 6b, Model_type must be one of the following:

Input_diff, Output_diff, I/O_diff, and 3-state_diff

Special usage rules apply to the following. Some definitions are included for clarification:

Input	These model types must have Vinl and Vinh defined. If they are not defined, the parser issues a warning and the default values of Vinl = 0.8 V and Vinh = 2.0 V are assumed.
I/O	
I/O_open_drain	
I/O_open_sink	
I/O_open_source	
Input_ECL	These model types must have Vinl and Vinh defined. If they are not defined, the parser issues a warning and the default values of Vinl = -1.475 V and Vinh = -1.165 V are assumed.
I/O_ECL	
Terminator	This model type is an input-only model that can have analog loading effects on the circuit being simulated but has no digital logic thresholds. Examples of terminators are: capacitors, termination diodes, and pullup resistors.

[bunch of stuff, including a diagram, left out for now. Ed.]

```

| [Model]           External_Model_Diff
Model_type        I/O_diff          | Requires [External Model]
Polarity          Non-Inverting
Enable            Active-High

| The [Diff Pin] vdiff value overrides the thresholds below
Vinl = 0.8V        | Input logic "low" DC voltage, if any
Vinh = 2.0V        | Input logic "high" DC voltage, if any
|
|                   | The true differential measurement point is at
|                   | the crossover voltage
|                   | The Vmeas value is overridden
Vmeas = 1.5V       | Reference voltage for timing measurements
|
|                   | Single-ended timing test load is still permitted
Cref = 5pF          | Timing specification test load capacitance value
Rref = 500           | Timing specification test load resistance value
Vref = 0             | Timing specification test load voltage
|
|                   | These new subparameters are permitted for
|                   | single-ended differential operation based on the
|                   | [Diff Pin] keyword
Rref_diff = 100     | Timing specification differential resistance value
Cref_diff = 5pF      | Timing specification differential capacitance value

```

Keyword: [Model Spec]

Required: No

Sub-Params: Vinh, Vinl, Vinh+, Vinh-, Vinl+, Vinl-, S_overshoot_high, S_overshoot_low, D_overshoot_high, D_overshoot_low, D_overshoot_time, D_overshoot_area_h, D_overshoot_area_l, D_overshoot_ampl_h, D_overshoot_ampl_l, Pulse_high, Pulse_low, Pulse_time, Vmeas, Vref, Cref, Rref, Cref_rising, C_ref_falling, Rref_rising, Rref_falling, Vref_rising, Vref_falling, Vmeas_rising, Vmeas_falling, Rref_diff, Cref_diff

Description: The [Model Spec] keyword defines four columns under which specification subparameters are defined.