```
|==============================================================================
|==============================================================================
| I/O Buffer Information Specification (IBIS) Version 4.0 (July 19, 2002)
|
| IBIS is a standard for electronic behavioral specifications of integrated
| circuit input/output analog characteristics.
|==============================================================================
|==============================================================================
|
|                    T A B L E   O F   C O N T E N T S
|
|==============================================================================
|==============================================================================
|
```

Section 1

G E N E R A L   I N T R O D U C T I O N

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 this document, IBIS Version 4.0 (ratified in July 2002).

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 4 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 described in separate .ebd files.

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.

Section 2

S T A T E M E N T O F I N T E N T

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 4.0 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.
|
|==============================================================================
```

==

Section 3

G E N E R A L   S Y N T A X   R U L E S   A N D   G U I D E L I N E S

==
==

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.
```

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 twenty 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 80 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 throughout 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 80 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.

==

```
|==============================================================================
|
|                                  Section 4
|
|                 F I L E    H E A D E R    I N F O R M A T I O N
|
|==============================================================================
|==============================================================================
|     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]      4.0                     | 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 throughout 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 ".ebd".  The file name must be the actual name of the file.
|------------------------------------------------------------------------------
[File Name]     ver4_0.ibs
|
|==============================================================================
|     Keyword:  [File Rev]
|    Required:  Yes
| Description:  Tracks the revision level of a particular .ibs file.
| Usage Rules:  Revision level is set at the discretion of the engineer
|               defining the file.  The following guidelines are recommended:
|                   0.x     silicon and file in development
|                   1.x     pre-silicon file data from silicon model only
|                   2.x     file correlated to actual silicon measurements
|                   3.x     mature product, no more changes likely
|------------------------------------------------------------------------------
[File Rev]      1.0                     | Used for .ibs file variations
|
|==============================================================================
|    Keywords:  [Date], [Source], [Notes], [Disclaimer], [Copyright]
|    Required:  No
| Description:  Optionally clarifies the file.
| Usage Rules:  The keyword arguments can contain blanks, and be of any
|               format.  The [Date] keyword argument is limited to a maximum
|               of 40 characters, and the month should be spelled out for
|               clarity.
```

```
|
|                 Because IBIS model writers may consider the information in
|                 these keywords essential to users, and sometimes legally
|                 required, design automation tools should make this information
|                 available.  Derivative models should include this text
|                 verbatim.  Any text following the [Copyright] keyword must be
|                 included in any derivative models verbatim.
|------------------------------------------------------------------------------
[Date]            July 19, 2002          | The latest file revision date
|
[Source]          Put originator and the source of information here.  For
                  example:

                  From silicon level SPICE model at Intel.
                  From lab measurement at IEI.
                  Compiled from manufacturer's data book at Quad Design, etc.
|
[Notes]           Use this section for any special notes related to the file.
|
[Disclaimer]      This information is for modeling purposes only, and is not
                  guaranteed.                      | May vary by component
|
[Copyright]       Copyright 2002, XYZ Corp., All Rights Reserved
|
|==============================================================================
```

```
|==============================================================================
|
|                                  Section 5
|
|                   C O M P O N E N T   D E S C R I P T I O N
|
|==============================================================================
|==============================================================================
|     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 manufacture.
| 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:  Yes
| Description:  Associates the component's I/O models to its various external
|               pin names and signal names.
```

Sub-Params: signal_name, model_name, R_pin, L_pin, C_pin

Usage Rules: All pins on a component must be specified. The first column must contain the pin name. The second column, signal_name, gives the data book name for the signal on that pin. The third column, model_name, maps a pin to a specific I/O buffer model or model selector name. Each model_name must have a corresponding model or model selector name listed in a **[Model]** or **[Model Selector]** keyword below, unless it is a reserved model name (POWER, GND, or NC).

Each line must contain either three or six columns. A pin line with three columns only associates the pin's signal and model. Six columns can be used to override the default package values (specified under **[Package]**) FOR THAT PIN ONLY. When using six columns, the headers R_pin, L_pin, and C_pin must be listed. If "NA" is in columns 4 through 6, the default packaging values must be used. The headers R_pin, L_pin, and C_pin may be listed in any order.

Column length limits are:

[Pin]	5 characters max
model_name	20 characters max
signal_name	20 characters max
R_pin	9 characters max
L_pin	9 characters max
C_pin	9 characters max

```
[Pin]   signal_name     model_name      R_pin   L_pin   C_pin
|
  1     RAS0#           Buffer1         200.0m  5.0nH   2.0pF
  2     RAS1#           Buffer2         209.0m  NA      2.5pF
  3     EN1#            Input1          NA      6.3nH   NA
  4     A0              3-state
  5     D0              I/O1
  6     RD#             Input2          310.0m  3.0nH   2.0pF
  7     WR#             Input2
  8     A1              I/O2
  9     D1              I/O2
 10     GND             GND             297.0m  6.7nH   3.4pF
 11     RDY#            Input2
 12     GND             GND             270.0m  5.3nH   4.0pF
|  .
|  .
|  .
 18     Vcc3            POWER
 19     NC              NC
 20     Vcc5            POWER           226.0m  NA      1.0pF
|
```

Keyword: **[Package Model]**

Required: No

Description: Indicates the name of the package model to be used for the component.

Usage Rules: The package model name is limited to 40 characters. Spaces are allowed in the name. The name should include the company name or initials to help ensure uniqueness. The EDA tool will search for a matching package model name as an argument to a **[Define Package Model]** keyword in the current IBIS file first. If a match is not found, the EDA tool will next look for a match in an external .pkg file. If the matching package model is in an external .pkg file, it must be located in the same directory as the .ibs file. The file names of .pkg files must follow the rules for file names given in Section 3, GENERAL SYNTAX RULES AND GUIDELINES.

Other Notes:: Use the **[Package Model]** keyword within a **[Component]** to indicate which package model should be used for that component. The specification permits .ibs files to contain **[Define Package Model]** keywords as well. These are described in the "Package Modeling" section near the end of this specification. When package model definitions occur within a

```
|                .ibs file, their scope is "local", i.e., they are known only
|                within that .ibs file and no other.  In addition, within that
|                .ibs file, they override any globally defined package models
|                that have the same name.
|------------------------------------------------------------------------------
[Package Model]     QS-SMT-cer-8-pin-pkgs
|
|==============================================================================
|    Keywords:  [Alternate Package Models], [End Alternate Package Models]
|    Required:  No
| Description:  Used to select a package model from a list of package models.
| Usage Rules:  The [Alternate Package Models] keyword can be used in addition
|               to the [Package Model] keyword.  [Alternate Package Models]
|               shall be used only for components that use the [Package Model]
|               keyword.
|
|               Each [Alternate Package Models] keyword specifies a set of
|               alternate package model names for only one component, which
|               is given by the previous [Component] keyword.  The [Alternate
|               Package Models] keyword shall not appear before the first
|               [Component] keyword in an IBIS file.  The [Alternate Package
|               Models] keyword applies only to the [Component] section in
|               which it appears, and must be followed by an [End Alternate
|               Package Models] keyword.
|
|               All alternate package model names must appear below the
|               [Alternate Package Models] keyword, and above the following
|               [End Alternate Package Models] keyword.  The package model
|               names listed under the [Alternate Package Models] must follow
|               the rules of the package model names associated with the
|               [Package Model] keyword.  The package model names correspond
|               to the names of package models defined by [Define Package
|               Model] keywords.  EDA tools may offer users a facility
|               for choosing between the default package model and any of the
|               alternate package models, when analyzing occurances of the
|               [Component].
|
|               The package model named by [Package Model] can be optionally
|               repeated in the [Alternate Package Models] list of names.
|------------------------------------------------------------------------------
[Alternate Package Models]
|
208-pin_plastic_PQFP_package-even_mode | Descriptive names are shown
208-pin_plastic_PQFP_package-odd_mode
208-pin_ceramic_PQFP_package-even_mode
208-pin_ceramic_PQFP_package-odd_mode
|
[End Alternate Package Models]
|
|==============================================================================
|     Keyword:  [Pin Mapping]
|    Required:  No
| Description:  Used to indicate which power and ground buses a given driver,
|               receiver, or terminator is connected to.
|  Sub-Params:  pulldown_ref, pullup_ref, gnd_clamp_ref, power_clamp_ref
| Usage Rules:  Each power and ground bus is given a unique name that must
|               not exceed 15 characters.  The first column contains a pin
|               name.  Each pin name must match one of the pin names declared
|               previously in the [Pin] section of the IBIS file.  The second
|               column, pulldown_ref, designates the ground bus connections
|               for that pin.  Here the term ground bus can also mean another
|               power bus.  The third column pullup_ref designates the power
|               bus connection.  The fourth and fifth columns gnd_clamp_ref
|               and power_clamp_ref contain entries, if needed, to specify
|               different ground bus and power bus connections than those
|               previously specified.
|
|               If the [Pin Mapping] keyword is present, then the bus
|               connections for EVERY pin listed in the [Pin] section must
|               be given.
```

```
|
|                 Each line must contain either three or five columns.  Use the
|                 NC reserved word for entries that are not needed or that
|                 follow the conditions below:
|
|                 All entries with identical labels are assumed to be connected.
|                 Each unique entry label must connect to at least one pin whose
|                 model_name is POWER or GND.
|
|                 If a pin has no connection, then both the pulldown_ref and
|                 pullup_ref subparameters for it will be NC.
|
|                 GND and POWER pin entries and buses are designated by entries
|                 in either the pulldown_ref or pullup_ref columns.  There is no
|                 implied association to any column other than through explicit
|                 designations in other pins.
|
|                 For any other type of pin, the pulldown_ref column contains
|                 the power connection for the [Pulldown] table for non-ECL type
|                 [Model]s.  This is also the power connection for the [GND
|                 Clamp] table and the [Rgnd] model unless overridden by a
|                 specification in the gnd_clamp_ref column.
|
|                 Also, the pullup_ref column contains the power connection for
|                 the [Pullup] table and, for ECL type models, the [Pulldown]
|                 table.  This is also the power connection for the [POWER
|                 Clamp] table and the [Rpower] model unless overridden by a
|                 specification in the power_clamp_ref column.
|
|                 The column length limits are:
|                     [Pin Mapping]      5 characters max
|                     pulldown_ref     15 characters max
|                     pullup_ref       15 characters max
|                     gnd_clamp_ref    15 characters max
|                     power_clamp_ref  15 characters max
|
|                 When 5 columns are specified, the headings gnd_clamp_ref and
|                 power_clamp_ref must be used.  Otherwise, these headings can
|                 be omitted.
|-----------------------------------------------------------------------------
[Pin Mapping]  pulldown_ref     pullup_ref   gnd_clamp_ref  power_clamp_ref
|
1              GNDBUS1          PWRBUS1     | Signal pins and their associated
2              GNDBUS2          PWRBUS2     | ground and power connections
3              GNDBUS1          PWRBUS1      GNDCLMP        PWRCLAMP
4              GNDBUS2          PWRBUS2      GNDCLMP        PWRCLAMP
5              GNDBUS2          PWRBUS2      NC             PWRCLAMP
6              GNDBUS2          PWRBUS2      GNDCLMP        NC
                                            | Some possible clamping connections
|  .                                        | are shown above for illustration
|  .                                        | purposes.
|  .
11             GNDBUS1          NC          | One set of ground connections
12             GNDBUS1          NC          | NC indicates no connection to
13             GNDBUS1          NC          | power bus
|  .
21             GNDBUS2          NC          | Second set of ground connections
22             GNDBUS2          NC
23             GNDBUS2          NC
|  .
31             NC               PWRBUS1     | One set of power connections
32             NC               PWRBUS1     | NC indicates no connection to
33             NC               PWRBUS1     | ground bus
|  .
41             NC               PWRBUS2     | Second set of power connections
42             NC               PWRBUS2
43             NC               PWRBUS2
|  .
51             GNDCLMP          NC          | Additional power connections
52             NC               PWRCLMP     | for clamps
```

```
|==============================================================================
|     Keyword:  [Diff Pin]
|    Required:  No
| Description:  Associates differential pins, their differential threshold
|               voltages, and differential timing delays.
|  Sub-Params:  inv_pin, vdiff, tdelay_typ, tdelay_min, tdelay_max
| Usage Rules:  Enter only differential pin pairs.  The first column, [Diff
|               Pin], contains a non-inverting pin name.  The second column,
|               inv_pin, contains the corresponding inverting pin name for
|               I/O output.  Each pin name must match the pin names declared
|               previously in the [Pin] section of the IBIS file.  The third
|               column, vdiff, contains the specified output and differential
|               threshold voltage between pins if the pins are Input or I/O
|               model types.  For output only differential pins, the vdiff
|               entry is 0 V.  The fourth, fifth, and sixth columns,
|               tdelay_typ, tdelay_min, and tdelay_max, contain launch delays
|               of the non-inverting pins relative to the inverting pins. The
|               values can be of either polarity.
|
|               If a pin is a differential input pin, the differential input
|               threshold (vdiff) overrides and supersedes the need for Vinh
|               and Vinl.
|
|               If vdiff is not defined for a pin that is defined as requiring
|               a Vinh by its [Model] type, vdiff is set to the default value
|               of 200 mV.
|
| Other Notes::  The output pin polarity specification in the table overrides
|               the [Model] Polarity specification such that the pin in the
|               [Diff Pin] column is Non-Inverting and the pin in the inv_pin
|               column is Inverting.  This convention enables one  [Model] to
|               be used for both pins.
|
|               The column length limits are:
|                   [Diff Pin]     5 characters max
|                   inv_pin        5 characters max
|                   vdiff          9 characters max
|                   tdelay_typ     9 characters max
|                   tdelay_min     9 characters max
|                   tdelay_max     9 characters max
|
|               Each line must contain either four or six columns.  If "NA" is
|               entered in the vdiff, tdelay_typ, or tdelay_min columns, its
|               entry is interpreted as 0 V or 0 ns.  If "NA" appears in the
|               tdelay_max column, its value is interpreted as the tdelay_typ
|               value.  When using six columns, the headers tdelay_min and
|               tdelay_max must be listed.  Entries for the tdelay_min column
|               are based on minimum magnitudes; and tdelay_max column,
|               maximum magnitudes.  One entry of vdiff, regardless of its
|               polarity, is used for difference magnitudes.
|------------------------------------------------------------------------------
[Diff Pin]  inv_pin  vdiff  tdelay_typ tdelay_min tdelay_max
|
 3           4       150mV    -1ns       0ns      -2ns  | Input or I/O pair
 7           8          0V     1ns        NA        NA  | Output* pin pair
 9          10          NA      NA        NA        NA  | Output* pin pair
16          15       200mV     1ns    | Input or I/O pin pair
20          19          0V      NA    | Output* pin pair, tdelay = 0
22          21          NA      NA    | Output*, tdelay = 0
                                      | * Could be Input or I/O with vdiff = 0
|
|==============================================================================
|     Keyword:  [Series Pin Mapping]
|    Required:  No
| Description:  Used to associate two pins joined by a series model.
|  Sub-Params:  pin_2, model_name, function_table_group
| Usage Rules:  Enter only series pin pairs.  The first column, [Series Pin
|               Mapping], contains the series pin for which input impedances
|               are measured.  The second column, pin_2, contains the other
```

connection of the series model. Each pin must match the pin names declared previously in the **[Pin]** section of the IBIS file. The third column, model_name, associates the Series or Series_switch model for the pair of pins in the first two columns. The fourth column, function_table_group, contains an alphanumeric designator string to associate those sets of Series_switch pins that are switched together.

Each line must contain either three or four columns. When using four columns, the header function_table_group must be listed.

One possible application is to model crossbar switches where the straight through On paths are indicated by one designator and the cross over On paths are indicated by another designator. If the model referenced is a Series model, then the function_table_group entry is omitted.

The column length limits are:

```
    [Series Pin Mapping]       5 characters max
    pin_2                      5 characters max
    model_name                20 characters max
    function_table_group      20 characters max
```

Other Notes:: If the model_name is for a non-symmetrical series model, then the order of the pins is important. The **[Series Pin Mapping]** and pin_2 entries must be in the columns that correspond with Pin 1 and Pin 2 of the referenced model.

This mapping covers only the series paths between pins. The package parasitics and any other elements such as additional capacitance or clamping circuitry are defined by the model_name that is referenced in the **[Pin]** keyword. The model_names under the **[Pin]** keyword that are also referenced by the **[Series Pin Mapping]** keyword may include any legal model or reserved model except for Series and Series_switch models. Normally the pins will reference a **[Model]** whose Model_type is 'Terminator'. For example, a Series_switch model may contain Terminator models on EACH of the pins to describe both the capacitance on each pin and some clamping circuitry that may exist on each pin. In a similar manner, Input, I/O or Output models may exist on each pin of a Series model that is serving as a differential termination.

```
|------------------------------------------------------------------------------
[Series Pin Mapping]  pin_2    model_name      function_table_group
|
 2                     3       CBTSeries       1      | Four independent groups
 5                     6       CBTSeries       2
 9                     8       CBTSeries       3
 12                   11       CBTSeries       4
|
 22                   23       CBTSeries       5      | Straight through path
 25                   26       CBTSeries       5
 22                   26       CBTSeries       6      | Cross over path
 25                   23       CBTSeries       6
|
 32                   33       Fixed_series           | No group needed
|
|==============================================================================
```

Keyword: **[Series Switch Groups]**

Required: No

Description: Used to define allowable switching combinations of series switches described using the names of the groups in the **[Series Pin Mapping]** keyword function_table_group column.

Sub-Params: On, Off

Usage Rules: Each state line contains an allowable configuration. A typical state line will start with 'On' followed by all of the on-state group names or an 'Off' followed by all of the off-state group names. Only one of 'On' or 'Off' is required since the undefined states are presumed to be opposite of the

```
|                explicitly defined states.  The state line is terminated with
|                the slash '/', even if it extends over several lines to fit
|                within the 80 character column width restriction.
|
|                The group names in the function_table_group are used to
|                associate switches whose switching action is synchronized by
|                a common control function.  The first line defines the assumed
|                (default) state of the set of series switches.  Other sets of
|                states are listed and can be selected through a user interface
|                or through automatic control.
|------------------------------------------------------------------------------
[Series Switch Groups]
| Function Group States
On 1 2 3 4 /             | Default setting is all switched On
|
Off 1 2 3 4 /            | All Off setting
On 1 /                   | Other possible combinations below
On 2 /
On 3 /
On 4 /
On 1 2 /
On 1 3 /
On 1 4 /
On 2 3 /
On 2 4 /
On 3 4 /
On 1 2 3 /
On 1 2 4 /
On 1 3 4 /
On 2 3 4 /
| Off 4 /                | The last four lines above could have been replaced
| Off 3 /                | with these four lines with the same meaning.
| Off 2 /
| Off 1 /
|
On 5 /                   | Crossbar switch straight through connection
On 6 /                   | Crossbar cross over connection
Off 5 6 /                | Crossbar open switches
|
|==============================================================================
```

Keyword: **[Model Selector]**

Required: No

Description: Used to pick a **[Model]** from a list of **[Model]**s for a pin which uses a programmable buffer.

Usage Rules: A programmable buffer must have an individual **[Model]** section for each one of its modes used in the .ibs file. The names of these **[Model]**s must be unique and can be listed under the **[Model Selector]** keyword and/or pin list. The name of the **[Model Selector]** keyword must match the corresponding model name listed under the **[Pin]** or **[Series Pin Mapping]** keyword and must not contain more than 20 characters. A .ibs file must contain enough **[Model Selector]** keywords to cover all of the model selector names specified under the **[Pin]** and **[Series Pin Mapping]** keywords.

The section under the **[Model Selector]** keyword must have two fields. The two fields must be separated by at least one white space. The first field lists the **[Model]** name (up to 20 characters long). The second field contains a short description of the **[Model]** shown in the first field. The contents and format of this description is not standardized, however it shall be limited in length so that none of the descriptions exceed the 80-character length of the line that it started on. The purpose of the descriptions is to aid the user of the EDA tool in making intelligent buffer mode selections and it can be used by the EDA tool in a user interface dialog box as the basis of an interactive buffer selection mechanism.

The first entry under the **[Model Selector]** keyword shall be

```
|                considered the default by the EDA tool for all those
|                pins which call this [Model Selector].
|
|                The operation of this selection mechanism implies that a group
|                of pins which use the same programmable buffer (i.e., model
|                selector name) will be switched together from one [Model] to
|                another.  Therefore, if two groups of pins, for example an
|                address bus and a data bus, use the same programmable buffer,
|                and the user must have the capability to configure them
|                independently, one can use two [Model Selector] keywords with
|                unique names and the same list of [Model] keywords; however,
|                the usage of the [Model Selector] is not limited to these
|                examples.  Many other combinations are possible.
|-----------------------------------------------------------------------------
[Pin]   signal_name     model_name      R_pin   L_pin   C_pin
|
  1     RAS0#           Progbuffer1     200.0m  5.0nH   2.0pF
  2     EN1#            Input1          NA      6.3nH   NA
  3     A0              3-state
  4     D0              Progbuffer2
  5     D1              Progbuffer2     320.0m  3.1nH   2.2pF
  6     D2              Progbuffer2
  7     RD#             Input2          310.0m  3.0nH   2.0pF
|  .
|  .
|  .
  18     Vcc3            POWER
|
[Model Selector]        Progbuffer1
|
OUT_2       2 mA buffer without slew rate control
OUT_4       4 mA buffer without slew rate control
OUT_6       6 mA buffer without slew rate control
OUT_4S      4 mA buffer with slew rate control
OUT_6S      6 mA buffer with slew rate control
|
[Model Selector]        Progbuffer2
|
OUT_2       2 mA buffer without slew rate control
OUT_6       6 mA buffer without slew rate control
OUT_6S      6 mA buffer with slew rate control
OUT_8S      8 mA buffer with slew rate control
OUT_10S    10 mA buffer with slew rate control
|
|=============================================================================
```

==

Section 6

M O D E L   S T A T E M E N T

==

==

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 20 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.

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

Input I/O I/O_open_drain I/O_open_sink I/O_open_source	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.
Input_ECL I/O_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.
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.
Output	This model type indicates that an output always sources and/or sinks current and cannot be disabled.
3-state	This model type indicates that an output can be disabled, i.e., put into a high impedance state.
Open_sink Open_drain	These model types indicate that the output has an OPEN side (do not use the **[Pullup]** keyword, or if it must be used, set I = 0 mA for all voltages specified) and the output SINKS current. Open_drain model type is retained for backward compatibility.
Open_source	This model type indicates that the output has an OPEN side (do not use the **[Pulldown]**

	keyword, or if it must be used, set I = 0 mA for all voltages specified) and the output SOURCES current.
Input_ECL Output_ECL I/O_ECL 3-state_ECL	These model types specify that the model represents an ECL type logic that follows different conventions for the **[Pulldown]** keyword.
Series	This model type is for series models that can be described by **[R Series]**, **[L Series]**, **[Rl Series]**, **[C Series]**, **[Lc Series]**, **[Rc Series]**, **[Series Current]** and **[Series MOSFET]** keywords.
Series_switch	This model type is for series switch models that can be described by **[On]**, **[Off]**, **[R Series]**, **[L Series]**, **[Rl Series]**, **[C Series]**, **[Lc Series]**, **[Rc Series]**, **[Series Current]** and **[Series MOSFET]** keywords.

The Model_type subparameter is required.

The C_comp subparameter is required only when C_comp_pullup, C_comp_pulldown, C_comp_power_clamp, and C_comp_gnd_clamp are not present. If the C_comp subparameter is not present, at least one of the C_comp_pullup, C_comp_pulldown, C_comp_power_clamp, or C_comp_gnd_clamp subparameters is required. It is not illegal to include the C_comp subparameter together with one or more of the remaining C_comp_* subparameters, but in that case the simulator will have to make a decision whether to use C_comp or the C_comp_pullup, C_comp_pulldown, C_comp_power_clamp, and C_comp_gnd_clamp subparameters. Under no circumstances should the simulator use the value of C_comp simultaneously with the values of the other C_comp_* subparameters.

C_comp_pullup, C_comp_pulldown, C_comp_power_clamp, and C_comp_gnd_clamp are intended to represent the parasitic capacitances of those structures who's I-V characteristics are described by the **[Pullup]**, **[Pulldown]**, **[POWER Clamp]** and **[GND Clamp]** I-V tables. For this reason, the simulator should generate a circuit netlist so that, if defined, each of the C_comp_* capacitors are connected in parallel with their corresponding I-V tables, whether or not the I-V table exists. That is, the C_comp_* capacitors are positioned between the signal pad and the nodes defined by the **[Pullup Reference]**, **[Pulldown Reference]**, **[POWER Clamp Reference]** and **[GND Clamp Reference]** keywords, or the **[Voltage Range]** keyword and GND.

The C_comp and C_comp_* subparameters define die capacitance. These values should not include the capacitance of the package. C_comp and C_comp_* are allowed to use "NA" for the min and max values only.

The Polarity, Enable, Vinl, Vinh, Vmeas, Cref, Rref, and Vref subparameters are optional. The Polarity subparameter can be defined as either Non-Inverting or Inverting, and the Enable subparameter can be defined as either Active-High or Active-Low.

The Cref and Rref subparameters correspond to the test load that the semiconductor vendor uses when specifying the propagation delay and/or output switching time of the model. The Vmeas subparameter is the reference voltage level that the semiconductor vendor uses for the model. Include Cref, Rref, Vref, and Vmeas information to facilitate board-level timing simulation. The assumed connections for Cref, Rref, and Vref are shown in the following diagram:

```
     ___________
    |           |
    |       |\  |                 Rref
    |Driver |  \|------o----/\/\/\----o Vref
    |       |  /|      |
    |       |/  |     === Cref
    |___________|      |
                       |
                      GND
```

Other Notes:: A complete **[Model]** description normally contains the following keywords: **[Voltage Range]**, **[Pullup]**, **[Pulldown]**, **[GND Clamp]**, **[POWER Clamp]**, and **[Ramp]**. A Terminator model uses one or more of the **[Rgnd]**, **[Rpower]**, **[Rac]**, and **[Cac]** keywords. However, some models may have only a subset of these keywords. For example, an input structure normally only needs the **[Voltage Range]**, **[GND Clamp]**, and possibly the **[POWER Clamp]** keywords. If one or more of **[Rgnd]**, **[Rpower]**, **[Rac]**, and **[Cac]** keywords are used, then the Model_type must be Terminator.

```
|--------------------------------------------------------------------------------
| Signals       CLK1, CLK2,...          | Optional signal list, if desired
[Model]         Clockbuffer
Model_type      I/O
Polarity        Non-Inverting
Enable          Active-High
Vinl = 0.8V                             | Input logic "low" DC voltage, if any
Vinh = 2.0V                             | Input logic "high" DC voltage, if any
Vmeas = 1.5V                | Reference voltage for timing measurements
Cref = 50pF                 | Timing specification test load capacitance value
Rref = 500                  | Timing specification test load resistance value
Vref = 0                    | Timing specification test load voltage
| variable         typ              min              max
C_comp             7.0pF            5.0pF            9.0pF
|
C_comp_pullup      3.0pF            2.5pF            3.5pF | These four can be
C_comp_pulldown    2.0pF            1.5pF            2.5pF | used instead of
C_comp_power_clamp 1.0pF            0.5pF            1.5pF | C_comp
C_comp_gnd_clamp   1.0pF            0.5pF            1.5pF
|
|================================================================================
```

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, Pulse_high, Pulse_low, Pulse_time, Vmeas, Vref, Cref, Rref, Cref_rising, Cref_falling, Rref_rising, Rref_falling, Vref_rising, Vref_falling, Vmeas_rising, Vmeas_falling

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

The following subparameters are defined:

Vinh	Input voltage threshold high
Vinl	Input voltage threshold low
Vinh+	Hysteresis threshold high max Vt+
Vinh-	Hysteresis threshold high min Vt+
Vinl+	Hysteresis threshold low max Vt-
Vinl-	Hysteresis threshold low min Vt-
S_overshoot_high	Static overshoot high voltage
S_overshoot_low	Static overshoot low voltage
D_overshoot_high	Dynamic overshoot high voltage
D_overshoot_low	Dynamic overshoot low voltage
D_overshoot_time	Dynamic overshoot time
Pulse_high	Pulse immunity high voltage
Pulse_low	Pulse immunity low voltage
Pulse_time	Pulse immunity time
Vmeas	Measurement voltage for timing measurements
Vref	Timing specification test load voltage

Cref	Timing specification capacitive load
Rref	Timing specification resistance load
Cref_rising	Timing specification capacitive load for rising edges
Cref_falling	Timing specification capacitive load for falling edges
Rref_rising	Timing specification resistance load for rising edges
Rref_falling	Timing specification resistance load for falling edges
Vref_rising	Timing specification test load voltage for rising edges
Vref_falling	Timing specification test load voltage for falling edges
Vmeas_rising	Measurement voltage for rising edge timing measurements
Vmeas_falling	Measurement voltage for falling edge timing measurements

Usage Rules: **[Model Spec]** must follow all other subparameters under the **[Model]** keyword.

For each subparameter contained in the first column, the remaining three hold its typical, minimum and maximum values. The entries of typical, minimum and maximum must be placed on a single line and must be separated by at least one white space. All four columns are required under the **[Model Spec]** keyword. However, data is required only in the typical column. If minimum and/or maximum values are not available, the reserved word "NA" must be used indicating the typical value by default.

The minimum and maximum values are used for specifications subparameter values that may track the min and max operation conditions of the **[Model]**. Usually it is related to the Voltage Range settings.

Unless noted below, no subparameter requires having present any other subparameter.

Vinh, Vinl rules:

The threshold subparameter lines provide additional min and max column values, if needed. The typ column values are still required and would be expected to override the Vinh and Vinl subparameter values specified elsewhere. Note: the syntax rule that require inserting Vinh and Vinl under models remains unchanged even if the values are defined under the **[Model Spec]** keyword.

Vinh+, Vinh-, Vinl+, Vinl- rules:

The four hysteresis subparmeters (used for Schmitt trigger inputs for defining two thresholds for the rising edges and two thresholds for falling edges) must all be defined before independent input thresholds for rising and falling edges of the hysteresis threshold rules become effective. Otherwise the standard threshold subparameters remain in effect. The hysteresis thresholds shall be at the Vinh+ and Vinh- values for a low-to-high transition, and at the Vinl+ and Vinl- values for a high-to-low transition.

```
       |
       |           Receiver Voltage with Hysteresis Thresholds
       |
       |
       |       Rising Edge                             Falling Edge
       |    Switching Region   oo    o              Switching Region
       |                  |  o    oo  ooooooooo           |
       |                  V o                    o        |
Vinh+  - - - - - - - - - - x                        o     |
Vinh-  - - - - - - - - -  x                           o   |
       |                 o                             o  |
       |                o                               o |
       |               o                                 oV
Vinl+  - - - - - - - -  o - - - - - - - - - - - - - - - - - x
Vinl-  - - - - - - - - o  - - - - - - - - - - - - - - - - -  x
       |             o                                         o
       |          o                                               o
       |oooooo-----------------------------------------------------oooooooo

                                Time -->
```

S_overshoot_high, S_overshoot_low rules:

The static overshoot subparameters provide the DC voltage values for which the model is no longer guaranteed to function correctly. Typically these are voltages that would cause the physical component to be destroyed.

D_overshoot_high, D_overshoot_low, D_overshoot_time rules:

The dynamic overshoot values provide a time window during which the overshoot may exceed the static overshoot limits but be below the dynamic overshoot limits. D_overshoot_time is required for dynamic overshoot testing. In addition, if D_overshoot_high is specified, then S_overshoot_high is necessary for testing beyond the static limit. Similarly, if D_overshoot_low is specified, then S_overshoot_low is necessary for testing beyond the static limit.

```
       |
       |     Receiver Voltage with Static and Dynamic Overshoot Limits
       |
       |
       |   D_overshoot_time ->|      |<-
       |                      |      |
D_overshoot_high - - - - - - -+ - - -+
       |                      | oo   |  Passes - Does Not Exceed Bounds
       |                      |o  o  |
S_overshoot_high - - - - - - -x    o +- - - - - - - - - - - - - - - - - - -
       |                     o      o ooooooooo
       |                    o        o         o
       |                   o                    o
       |                  o                      o
       |                 o                        o
       |                o                          o
       |               o                            o
       |              o                              o        Fails -
       |             o                                o    Exceeds Bounds
       |           o                                   o      |   |  |
       |        o                                       o     V   V  V
       |oooooo-------------------------------------------o---------o---oooo
S_overshoot_low - - - - - - - - - - - - - - - - - - - - - x      +x x x - -
       |                                                  |o      x   x
       |                                                  | o   o|
D_overshoot_low - - - - - - - - - - - - - - - - - - - - - + -x x-+
       |                                                  |   x  |
                                       D_overshoot_time ->|      |<-

                                Time -->
```

Pulse_high, Pulse_low, Pulse_time rules:

The pulse immunity values provide a time window during which a rising pulse may exceed the nearest threshold value but be below the pulse voltage value and still not cause the input to switch. Pulse_time is required for pulse immunity testing. A rising response is tested only if Pulse_high is specified. Similarly, a falling response is tested only if Pulse_low is specified. The rising response may exceed the Vinl value, but remain below the Pulse_high value.

Similarly, the falling response may drop below the Vinh value, but remain above the Pulse_low value. In either case the input is regarded as immune to switching if the responses are within these extended windows. If the hysteresis thresholds are defined, then the rising response shall use Vinh- as the reference voltage, and the falling response shall use Vinl+ as the reference voltage.

```
           |
           |      Receiver Voltage with Pulse Immunity Thresholds
           |
           |
           |          Switching                        No Switching
           |              |                                |
           |              |      oo    o                   |   Switching
           |              |    o    oo  ooooooooo          |      |
           |              |   o                   o        |      |
           |             V  o                       o      V   oooV
  Vinh - - - - - - - - - -  x - - - - - - - - - - - - - x   o + -x
           | Pulse_time ->|  o  |<-                      |ooo  |   o
  Pulse_high - - - - -  + o - +          Pulse_low  - + - - +    o
           |            |o    |          Pulse_time ->|     |<-   o
  Vinl - - - - - - - -  x     + - - - - - - - - - - - - - - - - -  x
           |           o                                           o
           |         o                                              o
           |      o                                                   o
           |oooooo-------------------------------------------------------o

                                  Time -->
```

Vmeas, Vref, Cref, Rref rules:

The Vmeas, Vref, Cref and Rref values under the **[Model Spec]** keyword override their respective values entered elsewhere. Note that a Vmeas, Vref, Cref or Rref subparameters may not be used if its edge specific version (*_rising or *_falling) is used.

Cref_rising, Cref_falling, Rref_rising, Rref_falling, Vref_rising, Vref_falling, Vmeas_rising, Vmeas_falling rules:

Use these subparameters when specifying separate timing test loads and voltages for rising and falling edges. If one 'rising' or 'falling' subparameter is used, then the corresponding 'rising' or 'falling' subparameter must be present. The values listed in these subparameters override any corresponding Cref, Vref, Rref or Vmeas values entered elsewhere.

```
|------------------------------------------------------------------------------
[Model Spec]
|   Subparameter           typ        min        max
|
| Thresholds
|
Vinh                       3.5        3.15       3.85     | 70% of Vcc
Vinl                       1.5        1.35       1.65     | 30% of Vcc
|
| Vinh                     3.835       3.335     4.335    | Offset from Vcc
| Vinl                     3.525       3.025     4.025    | for PECL
```

```
|
| Hysteresis
|
Vinh+                     2.0        NA         NA       | Overrides the
Vinh-                     1.6        NA         NA       | thresholds
Vinl+                     1.1        NA         NA
Vinl-                     0.6        NA         NA       | All 4 are required
|
| Overshoot
|
S_overshoot_high          5.5        5.0        6.0      | Static overshoot
S_overshoot_low          -0.5        NA         NA
D_overshoot_high          6.0        5.5        6.5      | Dynamic overshoot
D_overshoot_low          -1.0       -1.0       -1.0      | requires
|                                                        | D_overshoot_time
D_overshoot_time          20n        20n        20n      | & static overshoot
|
| Pulse Immunity
|
Pulse_high                3V         NA         NA       | Pulse immunity
Pulse_low                 0          NA         NA       | requires
Pulse_time                3n         NA         NA       | Pulse_time
|
| Timing Thresholds
|
Vmeas                     3.68       3.18       4.68     | A 5 volt PECL
|                                                        | example
|
| Timing test load voltage reference example
|
Vref                      1.25       1.15       1.35     | An SSTL-2 example
|
|
| Rising and falling timing test load example (values from PCI-X
| specification)
|
Cref_falling              10p        10p         10p
Cref_rising               10p        10p         10p
Rref_rising               25         500         25   | typ value not specified
Rref_falling              25         500         25   | typ value not specified
Vref_rising               0          1.5         0
Vref_falling              3.3        1.5         3.6
Vmeas_rising              0.941      0.885       1.026 | vmeas = 0.285(vcc)
Vmeas_falling             2.0295     1.845       2.214 | vmeas = 0.615(vcc)
|
|==============================================================================
|     Keyword:  [Receiver Thresholds]
|    Required:  No
|  Sub-Params:  Vth, Vth_min, Vth_max, Vinh_ac, Vinh_dc, Vinl_ac, Vinl_dc,
|               Threshold_sensitivity, Reference_supply, Vcross_low,
|               Vcross_high, Vdiff_ac, Vdiff_dc, Tslew_ac, Tdiffslew_ac
| Description:   The [Receiver Thresholds] keyword defines both a set of
|               receiver input thresholds as well as their sensitivity to
|               variations in a referenced supply.  The subparameters are
|               defined as follows:
|
|               Vth, Vth_min and Vth_max are the ideal input threshold
|               voltages at which the output of a digital logic receiver
|               changes state.  Vth is the nominal input threshold voltage
|               under the voltage, temperature and process conditions that
|               define 'typ'.  Vth_min is the minimum input threshold
|               voltage at 'typ' conditions while Vth_max is the maximum
|               input threshold voltage at 'typ' conditions.
|
|               Vinh_ac is the voltage that a low-to-high going input
|               waveform must reach in order to guarantee that the
|               receiver's output has changed state.  In other words,
|               reaching Vinh_ac is sufficient to guarantee a receiver
|               state change. Vinh_ac is expressed as an offset from Vth.
|
```

Vinh_dc is the voltage that an input waveform must remain above (more positive than) in order to guarantee that a receiver output will NOT change state. Vinh_dc is expressed as an offset from Vth.

Vinl_ac is the voltage that a high-to-low going input waveform must reach in order to guarantee that the receiver's output has changed state. In other words, reaching Vinl_ac is sufficient to guarantee a receiver state change. Vinl_ac is expressed as an offset from Vth.

Vinl_dc is the voltage that an input waveform must remain below (more negative than) in order to guarantee that a receiver's output will NOT change state. Vinl_dc is expressed as an offset from Vth.

Threshold_sensitivity is a unit less number that specifies how Vth varies with respect to the supply voltage defined by the Reference_supply subparameter. Threshold_sensitivity is defined as:

```
                        change in input threshold voltage
Threshold_sensitivity = -----------------------------------
                        change in referenced supply voltage
```

Threshold_sensitivity must be entered as a whole number or decimal, not as a fraction.

Reference_supply indicates which supply voltage Vth tracks; i.e., it indicates which supply voltage change causes a change in input threshold. The legal arguments to this subparameter are as follows:

Power_clamp_ref	The supply voltage defined by the **[POWER Clamp Reference]** keyword
Gnd_clamp_ref	The supply voltage defined by the **[GND Clamp Reference]** keyword
Pullup_ref	The supply voltage defined by the **[Pullup reference]** keyword
Pulldown_ref	The supply voltage defined by the **[Pulldown reference]** keyword
Ext_ref	The supply voltage defined by the **[External Reference]** keyword

Tslew_ac and Tdiffslew_ac measures the absolute difference in time between the point at which an input waveform crosses Vinl_ac and the point it crosses Vinh_ac. The purpose of this parameter is to document the maximum amount of time an input signal may take to transition between Vinh_ac and Vinl_ac and still allow the device to meet its input setup and hold specifications. Tslew_ac is the parameter used for single ended receivers while Tdiffslew_ac must be used for receivers with differential inputs.

Vcross_low is the least positive voltage at which a differential receivers' input signals may cross while switching and still allow the receiver to meet its timing and functional specifications. Vcross_low is specified with respect to 0 V.

Vcross_high is the most positive voltage at which a differential receivers' input signals may cross while switching and still allow the receiver to meet its timing and functional specifications. Vcross_high is specified with respect to 0 V.

Vdiff_dc is the minimum voltage difference between the inputs of a differential receiver that guarantees the receiver will not change state.

Vdiff_ac is the minimum voltage difference between the inputs of a differential receiver that guarantees the receiver will change state.

Usage Rules: The **[Receiver Thresholds]** keyword is valid if the model type includes any reference to input or I/O. For single ended receivers the Vinh_ac, Vinh_dc, Vinl_ac, Vinh_dc, Vth and Tslew_ac subparameters are required and override the Vinh, Vinl, Vinh+/- and Vinl+/- subparameters declared under the **[Model]** or **[Model Spec]** keywords. For single ended receivers the Vth_min, Vth_max, Threshold_sensitivity and Reference_supply subparameters are optional. However, if the Threshold_sensitivity subparameter is present then the Reference_supply subparameter must also be present.

For differential receivers (i.e., the **[Receiver Thresholds]** keyword is part of a **[Model]** statement that describes a pin listed in the **[Diff Pin]** keyword) then the Vcross_low, Vcross_high, Vdiff_ac, Vdiff_dc and Tdiffslew_ac subparameters are required. The rest of the subparameters are not applicable. The Vdiff_ac and Vdiff_dc values override the value of the vdiff subparameter specified by the **[Diff Pin]** keyword. Note that Vcross_low and Vcross_high are valid over the device's minimum and maximum operating conditions.

Subparameter **Usage Rules**:
Numerical arguments are separated from their associated subparameter by an equals sign (=); white space around the equals sign is optional. The argument to the Reference_supply subparameter is separated from the subparameter by white space.

Vth at Minimum or Maximum Operating Conditions:
As described above, the Vth_min and Vth_max subparameters define the minimum and maximum input threshold values under typical operating conditions. There is no provision for directly specifying Vth under minimum or maximum operating conditions. Instead, these values are calculated using the following equation:

Vth(min/max) = Vth* + **[(Threshold_sensitivity) X (change in supply voltage)]**

where Vth* is either Vth, Vth_min or Vth_max as appropriate, and the supply voltage is the one indicated by the Reference_Supply subparameter.

A basic 3.3 V single ended receiver using only the required subparameters.

[Receiver Thresholds]
Vth = 1.5V
Vinh_ac = +225mV
Vinh_dc = +100mV
Vinl_ac = -225mV
Vinl_dc = -100mV
Tslew_ac = 1.2ns

A single ended receiver using an external threshold reference. In this case the input threshold is the external reference voltage so Threshold_sensitivity equals 1.

[Receiver Thresholds]
Vth = 1.0V
Threshold_sensitivity = 1
Reference_Supply Ext_ref
Vinh_ac = +200mV
Vinh_dc = +100mV
Vinl_ac = -200mV
Vinl_dc = -100mV
Tslew_ac = 400ps

```
|
| A fully specified single ended 3.3 V CMOS receiver
|
[Receiver Thresholds]
Vth = 1.5V
Vth_min = 1.45V
Vth_max = 1.53V
Threshold_sensitivity = 0.45
Reference_supply Power_clamp_ref
Vinh_ac = +200mV
Vinh_dc = +100mV
Vinl_ac = -200mV
Vinl_dc = -100mV
Tslew_ac = 400ps
|
| A differential receiver
|
[Receiver Thresholds]
Vcross_low = 0.65V
Vcross_high = 0.90V
Vdiff_ac = +200mV
Vdiff_dc = +100mV
Tdiffslew_ac = 200ps
|
|==============================================================================
```

Keyword: **[Add Submodel]**

Required: No

Description: References a submodel to be added to an existing model.

Usage Rules: The **[Add Submodel]** keyword is invoked within a model to add the functionality that is contained in the submodel or list of submodels in each line that follows. The first column contains the submodel name. The second column contains a submodel mode under which the submodel is used.

If the top-level model type is one of the I/O or 3-state models, the submodel mode may be Driving, Non-Driving, or All. For example, if the submodel mode is Non-Driving, then the submodel is used only in the high-Z state of a 3-state model. Set the submodel mode to All if the submodel is to be used for all modes of operation.

The submodel mode cannot conflict with the top-level model type. For example, if the top-level model type is an Open or Output type, the submodel mode cannot be set to Non-Driving. Similarly, if the top-level model type is Input, the submodel mode cannot be set to Driving.

The **[Add Submodel]** keyword is not defined for Series or Series_switch model types.

Refer to the ADD SUBMODEL DESCRIPTION section in this document for the descriptions of available submodels.

```
|------------------------------------------------------------------------------
[Add Submodel]
| Submodel_name        Mode
Bus_Hold_1             Non-Driving   | Adds the electrical characteristics of
                                     | [Submodel] Bus_Hold_1 for receiver or
                                     | high-Z mode only.
Dynamic_clamp_1        All           | Adds the Dynmanic_clamp_1 model for
                                     | all modes of operation.
|
|==============================================================================
```

Keyword: **[Driver Schedule]**

Required: No

Description: Describes the relative model switching sequence for referenced models to produce a multi-staged driver.

Usage Rules: The **[Driver schedule]** keyword establishes a hierarchical order between models and should be placed under the **[Model]** which acts as the top-level model. The scheduled models are then referenced from the top-level model by the **[Driver Schedule]**

keyword.

When a multi-staged buffer is modeled using the **[Driver Schedule]** keyword, all of its stages (including the first stage, or normal driver) have to be modeled as scheduled models.

If there is support for this feature in a EDA tool, the **[Driver Schedule]** keyword will cause it to use the **[Pulldown]**, **[Pulldown Reference]**, **[Pullup]**, **[Pullup Reference]**, **[Voltage Range]**, **[Ramp]**, **[Rising Waveform]** and **[Falling Waveform]** keywords from the scheduled models instead of the top-level model, according to the timing relationships described in the **[Driver Schedule]** keyword. Consequently, the keywords in the above list will be ignored in the top-level model. Also, all other keywords not shown in the above list will be ignored in the scheduled model(s).

However, both the top-level and the scheduled model(s) have to be complete models, i.e., all of the required keywords must be present and follow the syntactical rules.

For backwards compatibility reasons and for EDA tools which do not support multi-staged switching, the keywords in the above list can be used in the top-level **[Model]** to describe the overall characteristics of the buffer as if it was a composite model. It is not guaranteed, however, that such a top-level model will yield the same simulation results as a full multi-stage model. It is recommended that a "golden waveform" for the device consisting of a **[Rising Waveform]** table and a **[Falling Waveform]** table be supplied in the top-level model to serve as a reference for validation.

Even though some of the keywords are ignored in the scheduled model, it may still make sense in some cases to supply correct data with them. One such situation would arise when a **[Model]** is used both as a regular top-level model as well as a scheduled model.

The **[Driver Schedule]** table consists of five columns. The first column contains the model names of other models that exists in the .ibs file. The remaining four columns describe delays: Rise_on_dly, Rise_off_dly, Fall_on_dly, and Fall_off_dly. The t=0 time of each delay is the event when the EDA tool’s internal pulse initiates a rising or falling transition. All specified delay values must be equal to or greater than 0. There are only five valid combinations in which these delay values can be defined:

```
1)  Rise_on_dly    with   Fall_on_dly
2)  Rise_off_dly   with   Fall_off_dly
3)  Rise_on_dly    with   Rise_off_dly
4)  Fall_on_dly    with   Fall_off_dly
5)  All four delays defined
    (be careful about correct sequencing)
```

The four delay parameters have the meaning as described below. (Note that this description applies to buffer types which have both pullup and pulldown structures. For those buffer types which have only a pullup or pulldown structure, the description for the missing structure can be omitted.)

Rise_on_dly is the amount of time that elapses from the internal simulator pulse initiating a RISING edge to the t = 0 time of the waveform or ramp that turns the I-V table of the PULLUP device ON, and the t = 0 time of the waveform or ramp that turns the I-V table of the PULLDOWN device OFF (if they were not already turned ON and OFF, respectively, by another event).

Rise_off_dly is the amount of time that elapses from the internal simulator pulse initiating a RISING edge to the t = 0 time of the waveform or ramp that turns the I-V table of the PULLUP device OFF, and the t = 0 time of the waveform or ramp that turns the I-V table of the PULLDOWN device ON (if they were not already turned ON and OFF, respectively, by another event).

Fall_on_dly is the amount of time that elapses from the internal simulator pulse initiating a FALLING edge to the t = 0 time of the waveform or ramp that turns the I-V table of the PULLDOWN device ON, and the t = 0 time of the waveform or ramp that turns the I-V table of the PULLUP device OFF (if they were not already turned ON and OFF, respectively, by another event).

Fall_off_dly is the amount of time that elapses from the internal simulator pulse initiating a FALLING edge to the t = 0 time of the waveform or ramp that turns the I-V table of the PULLDOWN device OFF, and the t = 0 time of the waveform or ramp that turns the I-V table of the PULLUP device ON (if they were not already turned ON and OFF, respectively, by another event).

Note that some timing combinations may only be possible if the two halves of a complementary buffer are modeled separately as two open_* models.

Use 'NA' when no delay value is applicable. For each scheduled model the transition sequence must be complete, i.e., the scheduled model must return to its initial state.

No **[Driver Schedule]** table may reference a model which itself has within it a **[Driver Schedule]** keyword.

Other Notes:: The added models typically consist of Open_sink (Open_drain) or Open_source models to provide sequentially increased drive strengths. The added drive may be removed within the same transition for a momentary boost or during the opposite transition.

The syntax also allows for reducing the drive strength.

Note that the Rise_on_dly, Rise_off_dly, Fall_on_dly, Fall_off_dly parameters are single value parameters, so typical, minimum and maximum conditions cannot be described with them directly. In order to account for those effects, one can refer to the fastest waveform table with the delay number and then insert an appropriate amount of horizontal lead in section in those waveforms which need more delay.

Notice that the C_comp parameter of a multi-stage buffer is defined in the top-level model. The value of C_comp therefore includes the total capacitance of the entire buffer, including all of its stages. Since the rising and falling waveform measurements include the effects of C_comp, each of these waveforms must be generated with the total C_comp present, even if the various stages of the buffer are characterized individually.

Note: In a future release, the **[Driver Schedule]** keyword may be replaced by a newer method of specification that is consistent with some other planned extensions. However, the **[Driver Schedule]** syntax will continue to be supported.

```
|------------------------------------------------------------------------------
[Driver Schedule]
| Model_name     Rise_on_dly  Rise_off_dly  Fall_on_dly  Fall_off_dly
  MODEL_OUT      0.0ns        NA            0.0ns        NA
|
|                                  Examples of added multi-staged transitions
```

```
  M_O_SOURCE1     0.5ns        NA           0.5ns        NA
|               low (high-Z) to high       high to low (high-Z)
  M_O_SOURCE2    0.5n         1.5n         NA           NA
|                low to high to low         low (high-Z)
  M_O_DRAIN1     1.0n         NA           1.5n         NA
|               low to high (high-Z)       high (high-Z) to low
  M_O_DRAIN2     NA           NA           1.5n         2.0n
|                  high (high-Z)           high to low to high
|
|==============================================================================
```

Keyword: **[Temperature Range]**
Required: Yes, if other than the preferred 0, 50, 100 degree Celsius range
Description: Defines the temperature range over which the model is to operate.
Usage Rules: List the actual die temperatures (not percentages) in the typ, min, max format. "NA" is allowed for min and max only.
Other Notes:: The **[Temperature Range]** keyword also describes the temperature range over which the various I-V tables and ramp rates were derived. Refer to NOTES ON DATA DERIVATION METHODS for rules on which temperature values to put in the 'min' and 'max' columns.

```
|------------------------------------------------------------------------------
| variable              typ             min             max
[Temperature Range]     27.0            -50             130.0
|
|==============================================================================
```

Keyword: **[Voltage Range]**
Required: Yes, if **[Pullup Reference]**, **[Pulldown Reference]**, **[POWER Clamp Reference]**, and **[GND Clamp Reference]** are not present
Description: Defines the power supply voltage tolerance over which the model is intended to operate. It also specifies the default voltage rail to which the **[Pullup]** and **[POWER Clamp]** I-V data is referenced.
Usage Rules: Provide actual voltages (not percentages) in the typ, min, max format. "NA" is allowed for the min and max values only.
Other Notes:: If the **[Voltage Range]** keyword is not present, then all four of the keywords described below must be present: **[Pullup Reference]**, **[Pulldown Reference]**, **[POWER Clamp Reference]**, and **[GND Clamp Reference]**. If the **[Voltage Range]** is present, the other keywords are optional and may or may not be used as required. It is legal (although redundant) for an optional keyword to specify the same voltage as specified by the **[Voltage Range]** keyword.

```
|------------------------------------------------------------------------------
| variable              typ             min             max
[Voltage Range]         5.0V            4.5V            5.5V
|
|==============================================================================
```

Keyword: **[Pullup Reference]**
Required: Yes, if the **[Voltage Range]** keyword is not present
Description: Defines a voltage rail other than that defined by the **[Voltage Range]** keyword as the reference voltage for the **[Pullup]** I-V data.
Usage Rules: Provide actual voltages (not percentages) in the typ, min, max format. "NA" is allowed for the min and max values only.
Other Notes:: This keyword, if present, also defines the voltage range over which the typ, min, and max dV/dt_r values are derived.

```
|------------------------------------------------------------------------------
| variable              typ             min             max
[Pullup Reference]      5.0V            4.5V            5.5V
|
|==============================================================================
```

Keyword: **[Pulldown Reference]**
Required: Yes, if the **[Voltage Range]** keyword is not present
Description: Defines a power supply rail other than 0 V as the reference voltage for the **[Pulldown]** I-V data. If this keyword is not present, the voltage data points in the **[Pulldown]** I-V table are referenced to 0 V.
Usage Rules: Provide actual voltages (not percentages) in the typ, min, max

format. "NA" is allowed for the min and max values only.

Other Notes:: This keyword, if present, also defines the voltage range over which the typ, min, and max dV/dt_f values are derived.

```
 variable               typ            min            max
[Pulldown Reference]    0V             0V             0V
```

Keyword: **[POWER Clamp Reference]**

Required: Yes, if the **[Voltage Range]** keyword is not present

Description: Defines a voltage rail other than that defined by the **[Voltage Range]** keyword as the reference voltage for the **[POWER Clamp]** I-V data.

Usage Rules: Provide actual voltages (not percentages) in the typ, min, max format. "NA" is allowed for the min and max values only.

Other Notes:: Refer to the "Other Notes" section of the **[GND Clamp Reference]** keyword.

```
 variable               typ            min            max
[POWER Clamp Reference] 5.0V           4.5V           5.5V
```

Keyword: **[GND Clamp Reference]**

Required: Yes, if the **[Voltage Range]** keyword is not present

Description: Defines a power supply rail other than 0 V as the reference voltage for the **[GND Clamp]** I-V data. If this keyword is not present, the voltage data points in the **[GND Clamp]** I-V table are referenced to 0 V.

Usage Rules: Provide actual voltages (not percentages) in the typ, min, max format. "NA" is allowed for the min and max values only.

Other Notes:: Power Supplies: It is intended that standard TTL and CMOS models be specified using only the **[Voltage Range]** keyword. However, in cases where the output characteristics of a model depend on more than a single supply and ground, or a **[Pullup]**, **[Pulldown]**, **[POWER Clamp]**, or **[GND Clamp]** table is referenced to something other than the default supplies, use the additional 'reference' keywords.

```
 variable               typ            min            max
[GND Clamp Reference]   0V             0V             0V
```

Keyword: **[External Reference]**

Required: Yes, if a receiver's input threshold is determined by an external reference voltage

Description: Defines a voltage source that supplies the reference voltage used by a receiver for its input threshold reference.

Usage Notes: Provide actual voltages (not percentages in the typ, min max format. "NA" is allowed for the min and max values only. Note that the numerically largest value should be placed in 'max' column, while the numerically smallest value should be placed in the 'min' column.

```
 variable              typ          min           max
[External Reference]   1.00V        0.95V         1.05V
```

Keywords: **[TTgnd], [TTpower]**

Required: No

Description: These keywords specify the transit time parameters used to estimate the transit time capacitances or develop transit time capacitance tables for the **[GND Clamp]** and **[POWER Clamp]** tables.

Usage Rules: For each of these keywords, the three columns hold the transit values corresponding to the typical, minimum and maximum **[GND Clamp]** or **[POWER Clamp]** tables, respectively. The entries for TT(typ), TT(min), and TT(max) must be placed on a single line and must be separated by at least one white space. All three columns are required under these keywords. However, data is required only in the typical column. If minimum and/or maximum values are not available, the reserved word "NA" must be used indicating the TT(typ) value by default.

Other Notes:: The transit time capacitance is added to C_comp. It is in a SPICE reference model as Ct = TT * d(Id)/d(Vd) where d(Id)/d(Vd) defines the DC conductance at the incremental DC operating point of the diode, and TT is the transit time. This expression does not include any internal series resistance. Such a resistance is assumed to be negligible in practice. Assume that the internal diode current (Id) – voltage (Vd) relationship is Id = Is * (exp(q(Vd)/kT) - 1) where Is is the saturation current, q is electron charge, k is Boltzmann's constant, and T is temperature in degrees Kelvin. Then d(Id)/d(Vd) is approximately (q/kT) * Id when the diode is conducting, and zero otherwise. This yields the simplification Ct = TT * (q/kT) * Id. The Id is found from the **[GND Clamp]** and **[POWER Clamp]** operating points, and the corresponding TTgnd or TTpower is used to calculate the Ct value. If the **[Temperature Range]** keyword is not defined, then use the default "typ" temperature for all Ct calculations.

The effective TT parameter values are intended to APPROXIMATE the effects. They may be different from the values found in the SPICE diode equations. Refer to the NOTES ON DATA DERIVATION METHOD for extracting the effective values.

```
|------------------------------------------------------------------------------
| variable      TT(typ)          TT(min)          TT(max)
[TTgnd]         10n              12n              9n
[TTpower]       12n              NA               NA
|
|==============================================================================
```

Keywords: **[Pulldown], [Pullup], [GND Clamp], [POWER Clamp]**

Required: Yes, if they exist in the model

Description: The data points under these keywords define the I-V tables of the pulldown and pullup structures of an output buffer and the I-V tables of the clamping diodes connected to the GND and the POWER pins, respectively. Currents are considered positive when their direction is into the component.

Usage Rules: In each of these sections, the first column contains the voltage value, and the three remaining columns hold the typical, minimum, and maximum current values. The four entries, Voltage, I(typ), I(min), and I(max) must be placed on a single line and must be separated by at least one white space.

All four columns are required under these keywords. However, data is only required in the typical column. If minimum and/or maximum current values are not available, the reserved word "NA" must be used. "NA" can be used for currents in the typical column, but numeric values MUST be specified for the first and last voltage points on any I-V table. Each I-V table must have at least 2, but not more than 100, rows.

Other Notes:: The I-V table of the **[Pullup]** and the **[POWER Clamp]** structures are 'Vcc relative', meaning that the voltage values are referenced to the Vcc pin. (Note: Under these keywords, all references to 'Vcc' refer to the voltage rail defined by the **[Voltage Range]**, **[Pullup Reference]**, or **[POWER Clamp Reference]** keywords, as appropriate.) The voltages in the data tables are derived from the equation: Vtable = Vcc – Voutput.

Therefore, for a 5 V model, -5 V in the table actually means 5 V above Vcc, which is +10 V with respect to ground; and 10 V means 10 V below Vcc, which is -5 V with respect to ground. Vcc-relative data is necessary to model a pullup structure properly, since the output current of a pullup structure depends on the voltage between the output and Vcc pins and not the voltage between the output and ground pins. Note that the **[GND Clamp]** I-V table can include quiescent input currents, or the currents of a 3-stated output, if so desired.

When tabulating data for ECL models, the data in the **[Pulldown]** table is measured with the output in the 'logic low' state. In other words, the data in the table represents the I-V characteristics of the output when the output is at the most negative of its two logic levels. Likewise, the data in the **[Pullup]** table is measured with the output in the 'logic one' state and represents the I-V characteristics when the output is at the most positive logic level. Note that in BOTH of these cases, the data is referenced to the Vcc supply voltage, using the equation: Vtable = Vcc - Voutput.

Monotonicity Requirements:

To be monotonic, the I-V table data must meet any one of the following 8 criteria:

1- The CURRENT axis either increases or remains constant as the voltage axis is increased.
2- The CURRENT axis either increases or remains constant as the voltage axis is decreased.
3- The CURRENT axis either decreases or remains constant as the voltage axis is increased.
4- The CURRENT axis either decreases or remains constant as the voltage axis is decreased.

5- The VOLTAGE axis either increases or remains constant as the current axis is increased.
6- The VOLTAGE axis either increases or remains constant as the current axis is decreased.
7- The VOLTAGE axis either decreases or remains constant as the current axis is increased.
8- The VOLTAGE axis either decreases or remains constant as the current axis is decreased.

An IBIS syntax checking program shall test for non-monotonic data and provide a maximum of one warning per I-V table if non-monotonic data is found. For example:

"Warning: Line 300, Pulldown I-V table for model DC040403 is non-monotonic! Most simulators will filter this data to remove the non-monotonic data."

It is also recognized that the data may be monotonic if currents from both the output stage and the clamp diode are added together as most simulators do. To limit the complexity of the IBIS syntax checking programs, such programs will conduct monotonicity testing only on one I-V table at a time.

It is assumed that the simulator sums the clamp tables together with the appropriate **[Pullup]** or **[Pulldown]** table when a buffer is driving high or low, respectively. From this assumption and the nature of 3-statable buffers, it follows that the data in the clamping table sections are handled as constantly present tables and the **[Pullup]** and **[Pulldown]** tables are used only when needed in the simulation.

The clamp tables of an Input or I/O buffer can be measured directly with a curve tracer, with the I/O buffer 3-stated. However, sweeping enabled buffers results in tables that are the sum of the clamping tables and the output structures. Based on the assumption outlined above, the **[Pullup]** and **[Pulldown]** tables of an IBIS model must represent the difference of the 3-stated and the enabled buffer's tables. (Note that the resulting difference table can demonstrate a non-monotonic shape.) This requirement enables the simulator to sum the tables, without the danger of double counting, and arrive at an accurate model in both the 3-stated and enabled conditions.

```
|
|                 Since in the case of a non 3-statable buffer, this difference
|                 table cannot be generated through lab measurements (because
|                 the clamping tables cannot be measured alone), the [Pullup]
|                 and [Pulldown] tables of an IBIS model can contain the sum of
|                 the clamping characteristics and the output structure.  In
|                 this case, the clamping tables must contain all zeroes, or the
|                 keywords must be omitted.
|-----------------------------------------------------------------------------
[Pulldown]
|  Voltage   I(typ)    I(min)    I(max)
|
   -5.0V    -40.0m    -34.0m    -45.0m
   -4.0V    -39.0m    -33.0m    -43.0m
|    .
|    .
    0.0V      0.0m      0.0m      0.0m
|    .
|    .
    5.0V     40.0m     34.0m     45.0m
   10.0V     45.0m     40.0m     49.0m
|
[Pullup]                                | Note: Vtable = Vcc - Voutput
|
|  Voltage   I(typ)    I(min)    I(max)
|
   -5.0V     32.0m     30.0m     35.0m
   -4.0V     31.0m     29.0m     33.0m
|    .
|    .
    0.0V      0.0m      0.0m      0.0m
|    .
|    .
    5.0V    -32.0m    -30.0m    -35.0m
   10.0V    -38.0m    -35.0m    -40.0m
|
[GND Clamp]
|
|  Voltage   I(typ)    I(min)    I(max)
|
   -5.0V  -3900.0m  -3800.0m  -4000.0m
   -0.7V    -80.0m    -75.0m    -85.0m
   -0.6V    -22.0m    -20.0m    -25.0m
   -0.5V     -2.4m     -2.0m     -2.9m
   -0.4V      0.0m      0.0m      0.0m
    5.0V      0.0m      0.0m      0.0m
|
[POWER Clamp]                           | Note: Vtable = Vcc - Voutput
|
|  Voltage   I(typ)    I(min)    I(max)
|
   -5.0V   4450.0m       NA        NA
   -0.7V     95.0m       NA        NA
   -0.6V     23.0m       NA        NA
   -0.5V      2.4m       NA        NA
   -0.4V      0.0m       NA        NA
    0.0V      0.0m       NA        NA
|
|=============================================================================
```

Keywords: **[Rgnd], [Rpower], [Rac], [Cac]**

Required: Yes, if they exist in the model

Description: The data for these keywords define the resistance values of Rgnd and Rpower connected to GND and the POWER pins, respectively, and the resistance and capacitance values for an AC terminator.

Usage Rules: For each of these keywords, the three columns hold the typical, minimum, and maximum resistance values. The three entries for R(typ), R(min), and R(max), or the three entries for C(typ), C(min), and C(max) must be placed on a single line and must be separated by at least one white space. All three columns are required under these keywords. However, data is only required in the typical column. If minimum and/or maximum values are not available, the reserved word "NA" must be used indicating the R(typ) or C(typ) value by default.

Other Notes:: It should be noted that **[Rpower]** is connected to 'Vcc' and **[Rgnd]** is connected to 'GND'. However, **[GND Clamp Reference]** voltages, if defined, apply to **[Rgnd]**. **[POWER Clamp Reference]** voltages, if defined, apply to **[Rpower]**. Either or both **[Rgnd]** and **[Rpower]** may be defined and may coexist with **[GND Clamp]** and **[POWER Clamp]** tables. If the terminator consists of a series R and C (often referred to as either an AC or RC terminator), then both **[Rac]** and **[Cac]** are required. When **[Rgnd]**, **[Rpower]**, or **[Rac]** and **[Cac]** are specified, the Model_type must be Terminator.

```
       |<-------------TERMINATOR Model--------------->|

           [Voltage Range] or
        [POWER Clamp Reference]
                  o
                  |
       POWER_ o---o---o
       clamp  |       |
           |--o--|    \
           |     |    /
           | I-V |    \ Rpower    [Package] Keyword
           |     |    /            Subparameters *
           |--o--|    |         |<----------------->|
              |       |
              |       |                               PIN
        o-----o-------o-----o-----/\/\/\--@@@@@@---o--o
        |     |GND_   |     |      R_pkg   L_pkg   |
        |     |clamp  |     |                      |
        |  |--o--|    |     |                      |
        |  |     |    \     |                      |
        |  | I-V |    /Rgnd |                      |
        |  |     |    \     \                      |
        |  |--o--|    /     / Rac                  |
        |     |       |     \                      |
        |     o---o---o     /                      |
        |         |         |                      |
C_comp ===        o        === Cac          C_pkg ===
        |      GND or       |                      |
        |   [GND Clamp      |                      |
        |   Reference]      |                      |
        o-------------------o----------------------o
                            |
                            o
                           GND
```

* Note: More advanced package parameters are available within this standard, including more detailed power and ground net descriptions.

---

variable	R(typ)	R(min)	R(max)	
[Rgnd]	*330ohm*	*300ohm*	*360ohm*	Parallel Terminator
[Rpower]	*220ohm*	*200ohm*	*NA*	
[Rac]	*30ohm*	*NA*	*NA*	

variable	C(typ)	C(min)	C(max)	
[Cac]	*50pF*	*NA*	*NA*	AC terminator

Keywords: **[On]**, **[Off]**

Required: Yes, both **[On]** and **[Off]** for Series_switch Model_types only

Description: The 'On' state electrical models are positioned under **[On]**. The 'Off' state electrical models are positioned under **[Off]**.

Usage Rules: These keywords are only valid for Series_switch Model_types. Only keywords associated with Series_switch electrical models are permitted under **[On]** or **[Off]**. The Series electrical models describe the path for one state only and do not use the **[On]** and **[Off]** keywords.

In Series_switch models, **[On]** or **[Off]** must be positioned before any of the **[R Series]**, **[L Series]**, Rl Series], **[C Series]**, **[Lc Series]**, **[Rc Series]**, **[Series Current]**, and **[Series MOSFET]** keywords. There is no provision for any of these keywords to be defined once, but to apply to both states.

|--

[On]
| ... On state keywords such as **[R Series]**, **[Series Current]**, **[Series MOSFET]**
[Off]
| ... Off state keywords such as **[R Series]**, **[Series Current]**
|

|==

Keywords: **[R Series]**, **[L Series]**, **[Rl Series]**, **[C Series]**, **[Lc Series]**, **[Rc Series]**

Required: Yes, if they exist in the model

Description: The data for these keywords allow the definition of Series or Series_switch R, L or C paths.

Usage Rules: For each of these keywords, the three columns hold the typical, minimum, and maximum resistance values. The three entries must be placed on a single line and must be separated by at least one white space. All three columns are required under these keywords. However, data is only required in the typical column. If minimum and/or maximum values are not available, the reserved word "NA" must be used.

Other Notes:: This series RLC model is defined to allow IBIS to model simple passive models and/or parasitics.

These keywords are valid only for Series or Series_switch Model_types.

The model is:

```
                R Series
            +---/\/\/\/\--------------------+
            |                               |
  Pin 1     |   L Series   Rl Series        |  Pin 2
     <---+---@@@@@@@@---/\/\/\/\----------+--->
            |                               |
            |   | |                         |
            +---| |---@@@@@@@@@---/\/\/\/\---+
                | |   Lc Series   Rc Series
              C Series
```

[Rl Series] shall be defined only if **[L Series]** exists. **[Rl Series]** is 0 ohms if it is not defined in the path.

[Rc Series] and **[Lc Series]** shall be defined only if **[C Series]** exists. **[Rc Series]** is 0 ohms if it is not defined in the path. **[Lc Series]** is 0 henries if it is not defined in the path.

C_comp values are ignored for series models.

|--

variable	R(typ)	R(min)	R(max)	
[R Series]	*8ohm*	*6ohm*	*12ohm*	
variable	L(typ)	L(min)	L(max)	
[L Series]	*5nH*	*NA*	*NA*	
variable	R(typ)	R(min)	R(max)	
[Rl Series]	*4ohm*	*NA*	*NA*	
variable	C(typ)	C(min)	C(max)	The other elements
[C Series]	*50pF*	*NA*	*NA*	are 0 impedance

==

Keyword: **[Series Current]**

Required: Yes, if they exist in the model

Description: The data points under this keyword define the I-V tables for voltages measured at Pin 1 with respect to Pin 2. Currents are considered positive if they flow into Pin 1. Pins 1 and 2 are listed under the **[Series Pin Mapping]** keyword under columns **[Series Pin Mapping]** and pin_2, respectively.

Usage Rules: The first column contains the voltage value, and the remaining columns hold the typical, minimum, and maximum current values. The four entries, Voltage, I(typ), I(min), and I(max) must be placed on a single line and must be separated by at least one white space.

All four columns are required under these keywords. However, data is only required in the typical column. If minimum and/or maximum current values are not available, the reserved word "NA" must be used. "NA" can be used for currents in the typical column, but numeric values MUST be specified for the first and last voltage points on any I-V table. Each I-V table must have at least 2, but not more than 100 rows.

Other Notes:: There is no monotonicity requirement. However the model supplier should realize that it may not be possible to derive a behavioral model from non-monotonic data.

This keyword is valid only for Series or Series_switch Model_types.

The model is:

```
                 Table Current
                    ------>
              +  Table Voltage  -
           Pin 1  |---------|  Pin 2
              <---+         +--->
                  |---------|
```

C_comp values are ignored for **[Series Current]** models.

```
[Series Current]
|  Voltage   I(typ)     I(min)     I(max)
  -5.0V  -3900.0m  -3800.0m  -4000.0m
  -0.7V    -80.0m     -75.0m     -85.0m
  -0.6V    -22.0m     -20.0m     -25.0m
  -0.5V     -2.4m      -2.0m      -2.9m
  -0.4V      0.0m       0.0m       0.0m
   5.0V      0.0m       0.0m       0.0m
|
```

==

Keyword: **[Series MOSFET]**

Required: Yes, for series MOSFET switches

Description: The data points under this keyword define the I-V tables for voltages measured at Pin 2 for a given Vds setting. Currents are considered positive if they flow into Pin 1. Pins 1 and 2 are listed under the **[Series Pin Mapping]** keyword under **[Series Pin Mapping]** and pin_2 columns, respectively.

Sub-Params: Vds

Usage Rules: The first column contains the voltage value, and the three remaining columns hold the typical, minimum, and maximum current values. The four entries, Voltage, I(typ), I(min), and I(max) must be placed on a single line and must be separated by at least one white space.

All four columns are required under these keywords. However, data is only required in the typical column. If minimum and/or maximum current values are not available, the reserved word "NA" must be used. "NA" can be used for currents in the typical column, but numeric values MUST be specified for the first and last voltage points on any I-V table. Each I-V table must have at least 2, but not more than 100 rows.

Other Notes:: There is no monotonicity requirement. However the model supplier should realize that it may not be possible to derive a behavioral model from non-monotonic data.

```
                 Table Current
                    ------->  Ids
                +     Vds     -

                     Vcc
                      | g
                    __|__
                    -----  NMOS
             Pin 1 |     |    Pin 2
               <---+     +--->  +  Voltage = Vcc - Vs
               d   |_____|   s
              PMOS  --+--       Vs
                      | g
                     GND        -
```

Either of the FET's could be removed (or have zero current contribution. Thus this model covers all four conditions, off, single NMOS, single PMOS and parallel NMOS/PMOS.

```
       Voltage = Table Voltage = Vtable = Vcc - Vs
       Ids = Table Current for a given Vcc and Vds
```

Internal Logic that is generally referenced to the power rail is used to set the NMOS MOSFET switch to its 'ON' state. Internal logic likewise referenced to ground is used to set the PMOS device to its 'ON' state if the PMOS device is present. Thus the **[Voltage Range]** settings provide the assumed gate voltages. If the **[POWER Clamp Reference]** exists, it overrides the **[Voltage Range]** value. The table entries are actually Vgs values of the NMOS device and Vcc - Vgs values of the PMOS device, if present. The polarity conventions are identical with those used for other tables that are referenced to power rails. Thus the voltage column can be viewed as a table defining the source voltages Vs according to the convention: Vtable = Vcc - Vs. This convention remains even without the NMOS device.

If the switch is used in an application such as interfacing between 3.3 V and 5.0 V logic, the Vcc may be biased at a voltage (such as 4.3 V) that is different from a power rail voltage (such as 5.0 V) used to create the model. Just readjust the **[Voltage Range]** entries (or **[POWER Clamp Reference]** entries).

One fundamental assumption in the MOSFET switch model is that it operates in a symmetrical manner. The tables and expressions are given assuming that Vd >= Vs. If Vd < Vs, then apply the same relationships under the assumption that the source and drain nodes are interchanged. A consequence of this assumption is that the Vds subparameter is constrained to values Vds > 0. It is assumed that with Vds = 0 the currents will be 0 mA. A further consequence of this assumption that would be embedded in the analysis process is that the voltage table is based on the side of the model with the lowest voltage (and that side is defined as the source). Thus the analysis must allow current to flow in both directions, as would occur due to reflections when the switch is connected in series with an unterminated transmission line.

The model data is used to create an On state relationship between the actual drain to source current, ids, and the actual drain to source voltage, vds:

ids = f(vds).

This functional relationship depends on the actual source voltage Vs and can be expressed in terms of the corresponding table currents associated with Vs (and expressed as a function of Vtable).

If only one **[Series MOSFET]** table is supplied (as a first order approximation), the functional relationship is assumed to be linearly related to the table drain to source current, Ids, for the given Vds subparameter value and located at the existing gate to source voltage value Vtable. This table current is denoted as Ids(Vtable, Vds). The functional relationship becomes:

ids = Ids(Vtable, Vds) * vds/Vds.

More than one **[Series MOSFET]** table is permitted, but it is simulator dependent how the data will be used. Each successive **[Series MOSFET]** table must have a different subparameter value for Vds. The number of tables must not exceed 100.

C_comp values are ignored for **[Series MOSFET]** models.

```
|------------------------------------------------------------------------------
| An NMOS Example
|
[On]
[Series MOSFET]
Vds = 1.0
| Voltage   I(typ)    I(min)    I(max)
   5.0V    257.9m    153.3m    399.5m    | Defines the Ids current as a
   4.0V    203.0m    119.4m    317.3m    | function of Vtable, for Vds = 1.0
   3.0V    129.8m     74.7m    205.6m
   2.0V     31.2m     16.6m     51.0m
   1.0V     52.7p     46.7p     56.7p
   0.0V      0.0p      0.0p      0.0p
|

| A PMOS/NMOS Example
|
[On]
[Series MOSFET]
Vds = 0.5
| Voltage   I(typ)    I(min)     I(max)
0.0 48.6ma NA NA
0.1 47.7ma NA NA
0.2 46.5ma NA NA
0.3 46.1ma NA NA
0.4 45.3ma NA NA
0.5 44.4ma NA NA
0.6 42.9ma NA NA
0.7 42.3ma NA NA
0.8 41.2ma NA NA
0.9 39.7ma NA NA
1.0 38.6ma NA NA
1.1 38.1ma NA NA
1.2 38.6ma NA NA
1.3 40.7ma NA NA
1.4 45.0ma NA NA
1.5 49.2ma NA NA
1.6 52.3ma NA NA
1.7 55.1ma NA NA
1.8 57.7ma NA NA
1.9 58.8ma NA NA
2.0 58.9ma NA NA
2.1 59.2ma NA NA
2.2 59.3ma NA NA
2.3 59.4ma NA NA
2.4 59.8ma NA NA
```

```
2.5 60.1ma NA NA
2.6 61.8ma NA NA
2.7 62.3ma NA NA
2.8 63.4ma NA NA
2.9 64.4ma NA NA
3.0 65.3ma NA NA
3.1 66.0ma NA NA
3.2 66.8ma NA NA
3.3 68.2ma NA NA
|
|==============================================================================
```

Keyword: **[Ramp]**

Required: Yes, except for inputs, terminators, Series and Series_switch model types

Description: Defines the rise and fall times of a buffer. The ramp rate does not include packaging but does include the effects of the C_comp or C_comp_* parameters.

Sub-Params: dV/dt_r, dV/dt_f, R_load

Usage Rules: The rise and fall time is defined as the time it takes the output to go from 20% to 80% of its final value. The ramp rate is defined as:

```
dV          20% to 80% voltage swing
-- = --------------------------------------
dt   Time it takes to swing the above voltage
```

The ramp rate must be specified as an explicit fraction and must not be reduced. The **[Ramp]** values can use "NA" for the min and max values only. The R_load subparameter is optional if the default 50 ohm load is used. The R_load subparameter is required if a non-standard load is used.

```
|------------------------------------------------------------------------------
[Ramp]
| variable      typ              min              max
dV/dt_r         2.20/1.06n       1.92/1.28n       2.49/650p
dV/dt_f         2.46/1.21n       2.21/1.54n       2.70/770p
R_load = 300ohms
|
|==============================================================================
```

Keywords: **[Rising Waveform], [Falling Waveform]**

Required: No

Description: Describes the shape of the rising and falling edge waveforms of a driver.

Sub-Params: R_fixture, V_fixture, V_fixture_min, V_fixture_max, C_fixture, L_fixture, R_dut, L_dut, C_dut

Usage Rules: Each **[Rising Waveform]** and **[Falling Waveform]** keyword introduces a table of voltage versus time points that describe the shape of an output waveform. These voltage versus time points are taken under the conditions specified by the R/L/C/V_fixture and R/L/C_dut subparameters. The table itself consists of one column of time points, then three columns of voltage points in the standard typ, min, and max format. The four entries must be placed on a single line and must be separated by at least one white space. All four columns are required. However, data is only required in the typical column. If minimum or maximum data is not available, use the reserved word "NA". The first value in the time column need not be '0'. Time values must increase as one parses down the table. The waveform table can contain a maximum of 1000 data rows. A maximum of 100 waveform tables are allowed per model.

Note that for backward compatibility, the existing **[Ramp]** keyword is still required. The data in the waveform table is taken with the effects of the C_comp parameter included.

A waveform table must include the entire waveform; i.e., the first entry (or entries) in a voltage column must be the DC voltage of the output before switching and the last entry (or entries) of the column must be the final DC value of the output after switching. Each table must contain at least two entries. Thus, numerical values are required for the first and last entries of any column containing numerical data.

The data in all of the waveform tables should be time correlated. In other words, the edge data in each of the tables (rising and falling) should be entered with respect to a single point in time when the input stimulus is assumed to have initiated a logic transition. All waveform extractions should reference a common input stimulus time in order to provide a sufficiently accurate alignment of waveforms. The first line in each waveform table should be assumed to be the reference point in time corresponding to a logic transition. For example, assume that some internal rising edge logic transition starts at time = 0. Then a rising edge voltage-time table might be created starting at time zero. The first several table entries might be some "lead-in" time caused by some undefined internal buffer delay before the voltage actually starts transitioning. The falling edge stimulus (for the purpose of setting reference time for the voltage-time table) should also start at time = 0. And, the falling edge voltage-time table would be created starting at time zero with a possibly different amount of "lead-in" time caused by a possibly different but corresponding falling edge internal buffer delay. Any actual device differences in internal buffer delay time between rising and falling edges should appear as differing lead-in times between the rising and the falling waveforms in the tables just as any differences in actual device rise and fall times appear as differing voltage-time entries in the tables.

A **[Model]** specification can contain more than one rising edge or falling edge waveform table. However, each new table must begin with the appropriate keyword and subparameter list as shown below. If more than one rising or falling edge waveform table is present, then the data in each of the respective tables must be time correlated. In other words, the rising (falling) edge data in each of the rising (falling) edge waveform tables must be entered with respect to a common reference point on the input stimulus waveform.

The 'fixture' subparameters specify the loading conditions under which the waveform is taken. The R_dut, C_dut, and L_dut subparameters are analogous to the package parameters R_pkg, C_pkg, and L_pkg and are used if the waveform includes the effects of pin inductance/capacitance. The diagram below shows the interconnection of these elements.

```
                    PACKAGE             |    TEST FIXTURE
                                        |
  __________                            |
 |   DUT    |     L_dut   R_dut         | L_fixture  R_fixture
 |   die    |---@@@@@--/\/\/\--o------- |--@@@@---o---/\/\/\----- V_fixture
 |__________|                  |        |         |
                               |        |         |
                               |        |         |
                       C_dut ===        |        === C_fixture
                               |        |         |
                               |        |         |
                              GND       |        GND
```

NOTE: The use of L_dut, R_dut, and C_dut is strongly discouraged in developing Waveform data from simulation models. Some simulators may ignore these parameters because they may introduce numerical time constant artifacts.

Only the R_fixture and V_fixture subparameters are required, the rest of the subparameters are optional. If a subparameter is not used, its value defaults to zero. The subparameters must appear in the text after the keyword and before the first row of the waveform table.

V_fixture defines the voltage for typ, min, and max supply conditions. However, when the fixture voltage is related to the power supply voltages, then the subparameters V_fixture_min and V_fixture_max can be used to further specify the fixture voltage for min and max supply voltages.

NOTE: Test fixtures with R_fixture and V_fixture, V_fixture_min, and V_fixture_max only are strongly encouraged because they provide the BEST set of data needed to produce the best model for simulation. C_fixture and L_fixture can be used to produce waveforms which describe the typical test case setups for reference.

NOTE: In most cases two **[Rising Waveform]** tables and two **[Falling Waveform]** tables will be necessary for accurate modeling.

All tables assume that the die capacitance is included. Potential numerical problems associated with processing the data using the effective C_comp (or C_comp_* values as appropriate) for effective die capacitance may be handled differently among simulators.

```
|------------------------------------------------------------------------------
[Rising Waveform]
R_fixture = 50
V_fixture = 0.0
| C_fixture = 50p        | These are shown, but are generally not recommended
| L_fixture = 2n
| C_dut = 7p
| R_dut = 1m
| L_dut = 1n
| Time            V(typ)              V(min)              V(max)
   0.0000s       25.2100mV           15.2200mV           43.5700mV
   0.2000ns       2.3325mV           -8.5090mV           23.4150mV
   0.4000ns       0.1484V            15.9375mV            0.3944V
   0.6000ns       0.7799V             0.2673V             1.3400V
   0.8000ns       1.2960V             0.6042V             1.9490V
   1.0000ns       1.6603V             0.9256V             2.4233V
   1.2000ns       1.9460V             1.2050V             2.8130V
   1.4000ns       2.1285V             1.3725V             3.0095V
   1.6000ns       2.3415V             1.5560V             3.1265V
   1.8000ns       2.5135V             1.7015V             3.1600V
   2.0000ns       2.6460V             1.8085V             3.1695V
| ...
  10.0000ns       2.7780V             2.3600V             3.1670V
|
[Falling Waveform]
R_fixture = 50
V_fixture = 5.5
V_fixture_min = 4.5
V_fixture_max = 5.5
| Time            V(typ)              V(min)              V(max)
   0.0000s        5.0000V             4.5000V             5.5000V
   0.2000ns       4.7470V             4.4695V             4.8815V
   0.4000ns       3.9030V             4.0955V             3.5355V
   0.6000ns       2.7313V             3.4533V             1.7770V
   0.8000ns       1.8150V             2.8570V             0.8629V
   1.0000ns       1.1697V             2.3270V             0.5364V
   1.2000ns       0.7539V             1.8470V             0.4524V
   1.4000ns       0.5905V             1.5430V             0.4368V
   1.6000ns       0.4923V             1.2290V             0.4266V
   1.8000ns       0.4639V             0.9906V             0.4207V
   2.0000ns       0.4489V             0.8349V             0.4169V
| ...
  10.0000ns       0.3950V             0.4935V             0.3841V
|
|==============================================================================
```

Keywords: **[Test Data]**

Required: No

Description: Indicates the beginning of a set of Golden Waveforms and

```
|                   references the conditions under which they were derived. An
|                   IBIS file may contain any number of [Test Data] sections
|                   representing different driver and load combinations.
|                   Golden Waveforms are a set of waveforms simulated
|                   using known ideal test loads.  They are useful in verifying
|                   the accuracy of behavioral simulation results against the
|                   transistor level circuit model from which the IBIS model
|                   parameters originated.
|   Sub-Params:     Test_data_type, Driver_model, Driver_model_inv, Test_load
|  Usage Rules:     The name following the [Test Data] keyword is required.  It
|                   allows a tool to select which data to analyze.
|
|                   The Test_data_type subparameter is required, and its value
|                   must be either "Single_ended" or "Differential."  The value of
|                   Test_data_type must match the value of Test_load_type found in
|                   the load called by Test_load.
|
|                   The Driver_model subparameter is required.  Its value
|                   specifies the "device-under-test" and must be a valid [Model]
|                   name.  Driver_model_inv is only legal if Test_data_type is
|                   Differential.  Driver_model_inv is not required but may be
|                   used in the case in which a differential driver uses two
|                   different models for the inverting and non-inverting pins.
|
|                   The Test_load subparameter is required and indicates which
|                   [Test Load] was used to derive the Golden Waveforms. It must
|                   reference a valid [Test Load] name.
|------------------------------------------------------------------------------
[Test Data] Data1
Test_data_type Single_ended
Driver_model Buffer1
Test_load Load1
|
|==============================================================================
|     Keywords:     [Rising Waveform Near], [Falling Waveform Near],
|                   [Rising Waveform Far], [Falling Waveform Far],
|                   [Diff Rising Waveform Near], [Diff Falling Waveform Near],
|                   [Diff Rising Waveform Far], [Diff Falling Waveform Far]
|     Required:     At least one Rising/Falling waveform is required under the
|                   scope of the [Test Data] keyword.
|  Description:     Describes the shape of the rising and falling Golden Waveforms
|                   of a given driver and a given [Test Load] measured at the
|                   driver I/O pad (near) or receiver I/O pad (far).  A model
|                   developer may use the [Rising Waveform Near/Far] and [Falling
|                   Waveform Near/Far] keywords to document Golden Waveforms whose
|                   purpose is to facilitate the correlation of reference
|                   waveforms and behavioral simulations.
|  Usage Rules:     The process, temperature, and voltage conditions under which
|                   the Golden Waveforms are generated must be identical to those
|                   used to generate the I-V and V-T tables. The Golden Waveforms
|                   must be generated using unpackaged driver and receiver models.
|                   The simulator must NOT use the Golden Waveform tables in the
|                   construction of its internal stimulus function.
|
|                   The tables must conform to the format described under the
|                   [Rising Waveform] and [Falling Waveform] keywords.
|
|                   Both differential and single-ended waveforms are allowed
|                   regardless of the value of Test_data_type.  If Test_data_type
|                   is Singled_ended then differential waveforms will be ignored.
|                   If Test_data_type is Differential, a single-ended waveform
|                   refers to the model specified by Driver_model and the
|                   non-inverting driver output.
|------------------------------------------------------------------------------
[Rising Waveform Far]
| Time            V(typ)              V(min)              V(max)
   0.0000s        25.2100mV           15.2200mV           43.5700mV
   0.2000ns        2.3325mV           -8.5090mV           23.4150mV
   0.4000ns        0.1484V            15.9375mV            0.3944V
   0.6000ns        0.7799V             0.2673V             1.3400V
```

```
   0.8000ns       1.2960V             0.6042V             1.9490V
   1.0000ns       1.6603V             0.9256V             2.4233V
   1.2000ns       1.9460V             1.2050V             2.8130V
   1.4000ns       2.1285V             1.3725V             3.0095V
   1.6000ns       2.3415V             1.5560V             3.1265V
   1.8000ns       2.5135V             1.7015V             3.1600V
   2.0000ns       2.6460V             1.8085V             3.1695V
| ...
  10.0000ns       2.7780V             2.3600V             3.1670V
|
[Falling Waveform Far]
| Time            V(typ)              V(min)              V(max)
   0.0000s        5.0000V             4.5000V             5.5000V
   0.2000ns       4.7470V             4.4695V             4.8815V
   0.4000ns       3.9030V             4.0955V             3.5355V
   0.6000ns       2.7313V             3.4533V             1.7770V
   0.8000ns       1.8150V             2.8570V             0.8629V
   1.0000ns       1.1697V             2.3270V             0.5364V
   1.2000ns       0.7539V             1.8470V             0.4524V
   1.4000ns       0.5905V             1.5430V             0.4368V
   1.6000ns       0.4923V             1.2290V             0.4266V
   1.8000ns       0.4639V             0.9906V             0.4207V
   2.0000ns       0.4489V             0.8349V             0.4169V
| ...
  10.0000ns       0.3950V             0.4935V             0.3841V
```

|==

Keywords: **[Test Load]**

Required: No

Description: Defines a generic test load network and its associated electrical parameters for reference by Golden Waveforms under the **[Test Data]** keyword. The Golden Waveform tables correspond to a given **[Test Load]** which is specified by the Test_load subparameter under the **[Test Data]** keyword.

Sub-Params: Test_load_type, C1_near, Rs_near, Ls_near, C2_near, Rp1_near, Rp2_near, Td, Zo, Rp1_far, Rp2_far, C2_far, Ls_far, Rs_far, C1_far, V_term1, V_term2, Receiver_model, Receiver_model_inv, R_diff_near, R_diff_far.

Usage Rules: The Test_load_type subparameter is required, and its value must be either "Single_ended" or "Differential."

The subparameters specify the electrical parameters associated with a fixed generic test load. The diagram below describes the single_ended test load.

All subparameters except Test_load_type are optional. If omitted, series elements are shorted and shunt elements are opened by default.

```
                                 V_term1
                              o-----------o
                              |           |
                              \           \    receiver_model_name
  _______                     /           /                          _______
 |       |  NEAR       Rp1_near \           \ Rp1_far          FAR   |       |
 | |\    |                    /           /                         | |\    |
 | | \   |   Rs_near  Ls_near |   _____   |     Ls_far  Rs_far      | | \   |
 | |  >-|---o--/\/\--@@@@--o----o--O_____)--o----o--@@@@--/\/\--o---|-|  >  |
 | | /   |   |             |  |   Td      |    |               |    | | /   |
 | |/    |   | C1_near     |  \   Zo      \    | C2_far        |    | |/    |
 |_______|  ===           ===  /           /   ===             ===   |_______|
             |     C2_near |  \           \    |         C1_far |
             |             |  /           /    |               |
             |             |  |  V_term2  |    |               |
             o-------------o  o-----------o    o---------------o
             |              Rp2_near    Rp2_far                |
            GND                                               GND
```

```
|                If the Td subparameter is present, then the Zo subparameter
|                must also be present.  If the Td subparameter is not present,
|                then the simulator must remove the transmission line from
|                the network and short the two nodes to which it was
|                connected.
|
|                V_term1 defines the termination voltage for parallel
|                termination resistors Rp1_near and Rp1_far.  This voltage
|                is not related to the [Voltage Range] keyword.
|                If either Rp1_near or Rp1_far is used, then V_term1 must
|                also be used.
|
|                V_term2 defines the termination voltage for parallel
|                termination resistors Rp2_near and Rp2_far.
|                If either Rp2_near or Rp2_far is used, then V_term2 must
|                also be used.
|
|                Receiver_model is optional and indicates which, if any,
|                receiver is connected to the far end node. If not used, the
|                network defaults to no receiver.
|
|                Receiver_model_inv is not required but may be used in the
|                case in which a differential receiver uses two different
|                models for the inverting and non-inverting pins.
|                Receiver_model_inv is ignored if Test_load_type is
|                Single-ended.
|
|                If Test_load_type is Differential, then the test load is a
|                pair of the above circuits.  If the R_diff_near or R_diff_far
|                subparameter is used, a resistor is connected between the
|                near or far nodes of the two circuits.  If Test_load_type is
|                Single_ended, R_diff_near and R_diff_far are ignored.
|------------------------------------------------------------------------------
[Test Load] Load1
Test_load_type Single_ended
C1_near     = 1p
Rs_near     = 10
Ls_near     = 1n
C2_near     = 1p
Rp1_near    = 100
Rp2_near    = 100
Td          = 1ns
Zo          = 50
Rp1_far     = 100
Rp2_far     = 100
C2_far      = 1p
Ls_far      = 1n
Rs_far      = 10
C1_far      = 1p
R_diff_far  = 100
Receiver_model Input1
| variable      typ            min            max
Vterm1          1.5            1.4            1.6
Vterm2          0.0            0.0            0.0
|
| Example of a transmission line and receiver test load
|
[Test Load] Tline_rcv
Td          = 1n
Zo          = 50
Receiver_model Input1
|
|==============================================================================
```

==

Section 6a

A D D   S U B M O D E L   D E S C R I P T I O N

==
==

The **[Add Submodel]** keyword can be used under a top-level **[Model]** keyword to to add special-purpose functionality to the existing top-level model. This section describes the structure of the top-level model and the submodel.

TOP-LEVEL MODEL:

When special-purpose functional detail is needed, the top-level model can call one or more submodels. The **[Add Submodel]** keyword is positioned after the initial set of required and optional subparameters of the **[Model]** keyword and among the keywords under **[Model]**.

The **[Add Submodel]** keyword lists of name of each submodel and the permitted mode (Driving, Non-Driving or All) under which each added submodel is used.

SUBMODEL:

A submodel is defined using the **[Submodel]** keyword. It contains a subset of keywords and subparameters used for the **[Model]** keyword along with other keywords and subparameters that are needed for the added functionality.

The **[Submodel]** and **[Submodel Spec]** keywords are defined first since they are used for all submodels.

The only required subparameter in **[Submodel]** is Submodel_type to define the list of submodel types. No subparameters under **[Model]** are permitted under the **[Submodel]** keyword.

The following set of keywords that are defined under the **[Model]** keyword are supported by the **[Submodel]** keyword:

[Pulldown]
[Pullup]
[GND Clamp]
[POWER Clamp]
[Ramp]
[Rising Waveform]
[Falling Waveform]

The **[Voltage Range], [Pullup Reference], [Pulldown Reference], [GND Clamp Reference],** and **[POWER Clamp Reference]** keywords are not permitted. The voltage settings are inherited from the top-level model.

These additional keywords are used only for the **[Submodel]** are documented in this section:

[Submodel Spec]
[GND Pulse Table]
[POWER Pulse Table]

The application of these keywords depends upon the Submodel_type entries listed below:

Dynamic_clamp
Bus_hold
Fall_back

Permitted keywords that are not defined for any of these submodel types are ignored. The rules for what set of keywords are required are found under the Dynamic Clamp, Bus Hold, and Fall Back headings of this section.

==

Keyword: **[Submodel]**
Required: No
Description: Used to define a submodel, and its attributes.
Sub-Params: Submodel_type
Usage Rules: Each submodel must begin with the keyword **[Submodel].** The submodel name must match the one that is listed under an **[Add Submodel]** keyword and must not contain more than 20 characters. A .ibs file must contain enough **[Submodel]** keywords to cover all of the model names specified under the **[Add Submodel]** keyword.

Submodel_type subparameter is required and must be one of the following:

Dynamic_clamp, Bus_hold, Fall_back

The C_comp subparameter is not permitted under the **[Submodel]** keyword. The total effective die capacitance including the submodel contributions are provided in the top-level model.

Other Notes:: The following list of keywords that are defined under the **[Model]** keyword can be used under **[Submodel]: [Pulldown], [Pullup], [GND Clamp], [POWER Clamp], [Ramp], [Rising Waveform]**, and **[Falling Waveform].**

The following list of additional keywords can be used: **[Submodel Spec], [GND Pulse Table]**, and **[POWER Pulse Table].**

```
[Submodel]        Dynamic_clamp1
Submodel_type     Dynamic_clamp
```

===

Keyword: **[Submodel Spec]**
Required: No
Description: The **[Submodel Spec]** keyword defines four columns under which specification and information subparameters are defined for submodels.
Sub-Params: V_trigger_r, V_trigger_f, Off_delay
Usage Rules: The **[Submodel Spec]** is to be used only with submodels.

The following subparameters are used:

V_trigger_r	Rising edge trigger voltage
V_trigger_f	Falling edge trigger voltage
Off_delay	Turn-off delay from V_trigger_r or V_trigger_f

For each subparameter contained in the first column, the remaining three hold its typical, minimum and maximum values. The entries of typical, minimum and maximum be must be placed on a single line and must be separated by at least one white space. All four columns are required under the **[Submodel Spec]** keyword. However, data is required only in the typical column. If minimum and/or maximum values are not available, the reserved word "NA" must be used to indicate the typical value by default.

The values in the minimum and maximum columns usually correspond to the values in the same columns for the inherited top-level voltage range or reference voltages in the top-level model. The V_trigger_r and V_trigger_f subparameters should hold values in the minimum and maximum columns that correspond to the voltage range or reference voltages of the top-level model. The Off_delay subparameter, however, is an exception to this rule because in some cases it may be completely or or partially independent from supply voltages and/or manufacturing process variations. Therefore the minimum and maximum entries for the Off_delay subparameter should be ordered simply by their magnitude.

Unless noted, each **[Submodel Spec]** subparameter is independent

```
|               of any other subparameter.
|
|               V_trigger_r, V_trigger_f rules:
|
|               The voltage trigger values for the rising and falling edges
|               provide the starting time when an action is initiated.
|
|               Off_delay rules:
|
|               The functionality of the Off_delay subparameter is to provide
|               an additional time related mechanism to turn off circuit
|               elements.
|
|------------------------------------------------------------------------------
| Dynamic Clamp Example:
|
[Submodel Spec]
|   Subparameter          typ        min        max
|
V_trigger_r               3.6        2.9        4.3 | Starts power pulse table
V_trigger_f               1.4        1.2        1.6 | Starts gnd pulse table
|
| Bus Hold Example:
|
[Submodel Spec]
|   Subparameter          typ        min        max
V_trigger_r               3.1        2.4        3.7 | Starts low to high
                                                    | bus hold transition
V_trigger_f               1.8        1.6        2.0 | Starts high to low
                                                    | bus hold transition
|
| Bus_hold application with pullup structure triggered on and then clocked off:
|
[Submodel Spec]
|   Subparameter          typ        min        max
V_trigger_r               3.1        2.4        3.7 | Low to high transition
                                                    | triggers the turn on
                                                    | process of the pullup
V_trigger_f             -10.0      -10.0      -10.0 | Not used, so trigger
                                                    | voltages are set out
                                                    | of range
Off_delay                 5n         4n         6n  | Time from rising edge
                                                    | trigger at which the
                                                    | pullup turned off
|
|==============================================================================
|
| Dynamic Clamp:
|
| When the Submodel_type subparameter under the [Submodel] keyword is set to
| Dynamic_clamp, the submodel describes the dynamic clamp functionality.
|
| The [GND Pulse Table] and [POWER Pulse Table] keywords are defined.  An
| example for a complete dynamic clamp model is provided.
|
|==============================================================================
```

Keywords: **[GND Pulse Table], [POWER Pulse Table]**

Required: No

Description: Used to specify the offset voltage versus time of **[GND Clamp]** and **[POWER Clamp]** tables within submodels.

Usage Rules: Each **[GND Pulse Table]** and **[POWER Pulse Table]** keyword introduces a table of voltage vs. time points that describe the shape of an offset voltage from the **[GND Clamp Reference]** voltage (or default ground) or the **[POWER Clamp Reference]** voltage (or default **[Voltage Range]** voltage). Note, these voltage values are inherited from the top-level model.

The table itself consists of one column of time points, then three columns of voltage points in the standard typ, min, and max format. The four entries must be placed on a single line and must be separated by at least one white space. All four columns are required. However, data is only required in the typical column. If minimum or maximum data is not available, use the reserved word "NA". Time values must increase as one parses down the table. The waveform table can contain of maximum of 100 rows.

Each table must contain at least two entries. Thus, numerical values are required for the first and last entries of any column containing numerical data.

The voltage entries in both the **[Gnd Pulse Table]** and **[POWER Pulse Table]** tables are directly measured offsets. At each instance, the **[Gnd Pulse Table]** voltage is ADDED to the **[GND Clamp]** table voltages to provide the shifted table voltages. At each instance, the **[POWER Pulse Table]** voltage is SUBTRACTED (because of polarity conventions) from the **[POWER Clamp]** table voltages to provide the shifted table voltages.

Only one **[GND Pulse Table]** and one **[POWER Pulse Table]** are allowed per model.

The **[GND Pulse Table]** and **[POWER Pulse Table]** interact with **[Submodel Spec]** subparameters V_trigger_f and V_trigger_r. Several modes of operation exist based on whether a pulse table and its corresponding trigger subparameter are given. These modes are classified as triggered and static.

Triggered Mode:

For triggered mode a pulse table must exist and include the entire waveform; i.e., the first entry (or entries) in a voltage column must be equal to the last entry.

Also, a corresponding **[Submodel Spec]** V_trigger_* subparameter must exist. The triggered interaction is described:

The V_trigger_f subparameter under **[Submodel Spec]** is used to detect when the falling edge waveform at the die passes the trigger voltage. At that time the **[Gnd Pulse Table]** operation starts. Similarly, the V_trigger_r subparameter is used to detect when the rising edge waveform at the die passes the trigger voltage. At that time **[POWER Pulse Table]** operation starts. The **[GND Pulse Table]** dependency is shown below:

```
|                                Waveform at Die
|
|           o o o o
|                   o
|                    o
|                     o -------
|                     |o      ^
|                     | o     | V_trigger_f
|                     |  o    v               time
|                     |    o o-------------------->
|                     |
|                     |
|                     |       [GND Pulse Table]
|                     |
|                     |          o o o o
|                     |         o       o
|                     |        o          o
|                     |       o             o
|                     |      o                o
|                     |     o                   o          time
|                     o o o o o                   o o o -------->
|
|                     ^
|                     |_  [GND Pulse Table] operation starts at this time
|
| The V_trigger_r and [POWER Pulse Table] operate in a similar manner.  When
| the V_trigger_r voltage value is reached on the rising edge, the [POWER
| Pulse Table] is started.  Normally the offset voltage entries in the [POWER
| Pulse Table] are negative.
|
| Static Mode:
|
| When the [GND Pulse Table] keyword does not exist, but the added model
| [GND Clamp] table does exist, the added model [GND Clamp] is used directly.
| Similarly, when the [POWER Pulse Table] keyword does not exist, but the
| added model [POWER Clamp] table does exist, the added model [POWER Clamp]
| is used directly.
|
| This mode provides additional fixed clamping to an I/O_* buffer or a
| 3-state buffer when it is used as a driver.
|-----------------------------------------------------------------------------
|
| Example of Dynamic_clamp Model with both dynamic GND and POWER clamps:
|
[Submodel]        Dynamic_Clamp_1
Submodel_type     Dynamic_clamp
|
[Submodel Spec]
|   Subparameter          typ        min        max
|
V_trigger_f               1.4        1.2        1.6   | Falling edge trigger
V_trigger_r               3.6        2.9        4.3   | Rising edge trigger
|
|                         typ        min        max
| [Voltage Range]           5.0        4.5        5.5
| Note, the actual voltage range and reference voltages are inherited from
| the top-level model.
|
[GND Pulse Table]                                     | GND Clamp offset table
|    Time          V(typ)       V(min)        V(max)
|
      0              0            0             0
    1e-9             0            0             0
    2e-9           0.9          0.8           1.0
   10e-9           0.9          0.8           1.0
   11e-9             0            0             0
|
|
|
|
```

```
[GND Clamp]                                         | Table to be offset
|
|  Voltage        I(typ)       I(min)       I(max)
|
    -5.000     -3.300e+01   -3.000e+01   -3.500e+01
    -4.000     -2.300e+01   -2.200e+01   -2.400e+01
    -3.000     -1.300e+01   -1.200e+01   -1.400e+01
    -2.000     -3.000e+00   -2.300e+00   -3.700e+00
    -1.900     -2.100e+00   -1.500e+00   -2.800e+00
    -1.800     -1.300e+00   -8.600e-01   -1.900e+00
    -1.700     -6.800e-01   -4.000e-01   -1.100e+00
    -1.600     -2.800e-01   -1.800e-01   -5.100e-01
    -1.500     -1.200e-01   -9.800e-02   -1.800e-01
    -1.400     -7.500e-02   -7.100e-02   -8.300e-02
    -1.300     -5.750e-02   -5.700e-02   -5.900e-02
    -1.200     -4.600e-02   -4.650e-02   -4.550e-02
    -1.100     -3.550e-02   -3.700e-02   -3.450e-02
    -1.000     -2.650e-02   -2.850e-02   -2.500e-02
    -0.900     -1.850e-02   -2.100e-02   -1.650e-02
    -0.800     -1.200e-02   -1.400e-02   -9.750e-03
    -0.700     -6.700e-03   -8.800e-03   -4.700e-03
    -0.600     -3.000e-03   -4.650e-03   -1.600e-03
    -0.500     -9.450e-04   -1.950e-03   -3.650e-04
    -0.400     -5.700e-05   -2.700e-04   -5.550e-06
    -0.300     -1.200e-06   -1.200e-05   -5.500e-08
    -0.200     -3.000e-08   -5.000e-07    0.000e+00
    -0.100      0.000e+00    0.000e+00    0.000e+00
     0.000      0.000e+00    0.000e+00    0.000e+00
     5.000      0.000e+00    0.000e+00    0.000e+00
|
[POWER Pulse Table]                                | POWER Clamp offset table
                           |
|    Time          V(typ)       V(min)       V(max)
|
       0             0            0            0
    1e-9             0            0            0
    2e-9          -0.9         -1.0         -0.8
   10e-9          -0.9         -1.0         -0.8
   11e-9             0            0            0
|
[POWER Clamp]                                      | Table to be offset
|
|  Voltage        I(typ)        I(min)       I(max)
|
    -5.000      1.150e+01    1.100e+01    1.150e+01
    -4.000      7.800e+00    7.500e+00    8.150e+00
    -3.000      4.350e+00    4.100e+00    4.700e+00
    -2.000      1.100e+00    8.750e-01    1.300e+00
    -1.900      8.000e-01    6.050e-01    1.000e+00
    -1.800      5.300e-01    3.700e-01    7.250e-01
    -1.700      2.900e-01    1.800e-01    4.500e-01
    -1.600      1.200e-01    6.850e-02    2.200e-01
    -1.500      3.650e-02    2.400e-02    6.900e-02
    -1.400      1.200e-02    1.100e-02    1.600e-02
    -1.300      6.300e-03    6.650e-03    6.100e-03
    -1.200      4.200e-03    4.750e-03    3.650e-03
    -1.100      2.900e-03    3.500e-03    2.350e-03
    -1.000      1.900e-03    2.450e-03    1.400e-03
    -0.900      1.150e-03    1.600e-03    7.100e-04
    -0.800      5.500e-04    9.150e-04    2.600e-04
    -0.700      1.200e-04    4.400e-04    5.600e-05
    -0.600      5.400e-05    1.550e-04    1.200e-05
    -0.500      1.350e-05    5.400e-05    1.300e-06
    -0.400      8.650e-07    7.450e-06    4.950e-08
    -0.300      6.250e-08    7.550e-07    0.000e+00
    -0.200      0.000e+00    8.400e-08    0.000e+00
    -0.100      0.000e+00    0.000e-08    0.000e+00
     0.000      0.000e+00    0.000e+00    0.000e+00
|
|===============================================================================
```

==

Bus Hold:

When the Submodel_type subparameter under the **[Submodel]** keyword is set to Bus_hold, the added model describes the bus hold functionality. However, while described in terms of bus hold functionality, active terminators can also be modeled.

Existing keywords and subparameters are used to describe bus hold models. The **[Pullup]** and **[Pulldown]** tables both are used to define an internal buffer that is triggered to switch to its opposite state. This switching transition is specified by a **[Ramp]** keyword or by the **[Rising Waveform]** and **[Falling Waveform]** keywords. The usage rules for these keywords are the same as under the **[Model]** keyword. In particular, at least either the **[Pullup]** or **[Pulldown]** keyword is required. Also, the **[Ramp]** keyword is required, even if the **[Rising Waveform]** and **[Falling Waveform]** tables exist. However, the voltage ranges and reference voltages are inherited from the top-level model.

For bus hold submodels, the **[Submodel Spec]** keyword, V_trigger_r, and V_trigger_f are required. The Off_delay subparameter is optional, and can only be used if the submodel consists of a pullup or a pulldown structure only, and not both. Devices which have both pullup and pulldown structures controlled in this fashion can be modeled using two submodels, one for each half of the circuit.

The transition is triggered by action at the die using the **[Submodel Spec]** V_trigger_r and V_trigger_f subparameters is described next. In all subsequent discussions, "low" means the pulldown structure is on or active, and the pullup structure is off or inactive if either or both exist. The opposite settings are referred to as "high".

If the starting voltage is below V_trigger_f, then the bus hold model is set to the low state causing additional pulldown current. If the starting voltage is above V_trigger_r, the bus hold model is set to the high state for additional pullup current.

Under some unusual cases, the above conditions can be both met or not met at all. To resolve this, the EDA tool should compute the starting voltage with the bus hold model set to low. If the starting voltage is equal to or less than the average of V_trigger_r and V_trigger_f, keep the bus hold model in the low state. Otherwise, set the bus hold model to the high state.

When the input passes through V_trigger_f during a high-to-low transition at the die, the bus hold output switches to the low state. Similarly, when the input passes though V_trigger_r during a low-to-high transition at the die, the bus hold output switches to the high state.

If the bus hold submodel has a pullup structure only, V_trigger_r provides the time when its pullup is turned on and V_trigger_f or Off_delay provides the time when it is turned off, whichever occurs first. Similarly, if the submodel has a pulldown structure only, V_trigger_f provides the time when its pulldown is turned on and V_trigger_r or Off_delay provides the time when it is turned off, whichever occurs first. The required V_trigger_r and V_trigger_f voltage entries can be set to values outside of the input signal range if the pullup or pulldown structures are to be held on until the Off_delay turns them off.

The starting mode for each of the submodels which include the Off_delay subparameter of the **[Submodel Spec]** keyword is the off state. Also, while two submodels provide the desired operation, either of the submodels may exist without the other to simulate turning on and off only a pullup or a pulldown current.

The following tables summarizes the bus hold initial and switching transitions:

BUS HOLD WITHOUT OFF_DELAY:

```
|     Initialization:
|
|       Initial Vdie Value                 Initial Bus Hold
|                                          Submodel State
|       --------------------------------   ----------------
|       <= V_trigger_r & < V_trigger_f     low
|       => V_trigger_f & > V_trigger_r     high
|
|       <= (V_trigger_f + V_trigger_r)/2   low    | Recommendations if neither
|       > (V_trigger_f + V_trigger_r)/2    high   | or both conditions above
|                                                 | are satisifed
|
|
|     Transitions:
|
|       Prior Bus Hold    Vdie transition  Bus Hold
|       Submodel State    through          Transition
|                         V_trigger_r/f
|       --------------    ---------------  -----------
|       low               V_trigger_r      low-to-high
|       low               V_trigger_f      no change
|       high              V_trigger_r      no change
|       high              V_trigger_f      high-to-low
|
|   BUS HOLD WITH OFF_DELAY (REQUIRES EITHER [PULLUP] or [PULLDOWN] ONLY):
|
|     Initialization:
|
|       [Pullup] or       Initial Bus Hold
|       [Pulldown] Table  Submodel State (Off Mode)
|       ----------------  -------------------------
|       [Pullup]          low
|       [Pulldown]        high
|
|     Transitions:
|
|       Prior Bus Hold    Vdie transition  Bus Hold      Off_delay
|       Submodel State    through          Transition    Transition
|                         V_trigger_r/f
|       --------------    ---------------  -----------   -----------
|       low               V_trigger_r      low-to-high   high-to-low
|       low               V_trigger_f      no change     no change
|       high              V_trigger_r      no change     no change
|       high              V_trigger_f      high-to-low   low-to-high
|
|     Note, if Vdie passes again through the V_trigger_r/f thresholds
|     before the Off_delay time is reached, the bus hold state follows the
|     change documented in the first table, overriding the Off_delay
|     transition.
|
| No additional keywords are needed for this functionality.
|-----------------------------------------------------------------------------
|
| Complete Bus Hold Model Example:
|
[Submodel]        Bus_hold_1
Submodel_type     Bus_hold
|
[Submodel Spec]
|   Subparameter           typ        min        max
|
V_trigger_f                1.3        1.2        1.4  | Falling edge trigger
V_trigger_r                3.1        2.6        4.6  | Rising edge trigger
|
|                          typ        min        max
 [Voltage Range]             5.0        4.5        5.5
| Note, the actual voltage range and reference voltages are inherited from
| the top-level model.
|
|
```

```
[Pulldown]
|
-5V     -100uA     -80uA     -120uA
-1V     -30uA      -25uA     -40uA
0V       0          0          0
1V       30uA       25uA     40uA
3V       50uA       45uA     50uA
5V       100uA      80uA     120uA
10v      120uA      90uA     150uA
|
[Pullup]
|
-5V      100uA      80uA     120uA
-1V      30uA       25uA     40uA
0V       0          0          0
1V      -30uA      -25uA    -40uA
3V      -50uA      -45uA    -50uA
5V      -100uA     -80uA    -120uA
10v     -120uA     -90uA    -150uA
|
|------------------------------------------------------------------------------
|
[Ramp]
|                        typ              min              max
dV/dt_r                  2.0/0.50n        2.0/0.75n        2.0/0.35n
dV/dt_f                  2.0/0.50n        2.0/0.75n        2.0/0.35n
R_load = 500
|
|------------------------------------------------------------------------------
|
| Complete Pulldown Timed Latch Example:
|
[Submodel]       Timed_pulldown_latch
Submodel_type    Bus_hold
|
[Submodel Spec]
|   Subparameter           typ        min        max
|
V_trigger_r                3.1        2.6        4.6    | Rising edge trigger
                                                        | Values could be set out
                                                        | of range to disable the
                                                        | trigger
V_trigger_f                1.3        1.2        1.4    | Falling edge trigger



Off_delay                  3n         2n         5n     | Delay to turn off the
                                                        | pulldown table
|
| Note that if the input signal goes above the V_trigger_r value, the
| pulldown structure will turn off even if the timer didn't expire yet.
|
|                          typ        min        max
| [Voltage Range]          5.0        4.5        5.5
| Note, the actual voltage range and reference voltages are inherited from
| the top-level model.
|
[Pulldown]
|
-5V     -100uA     -80uA     -120uA
-1V      -30uA     -25uA     -40uA
0V       0          0          0
1V       30uA       25uA     40uA
3V       50uA       45uA     50uA
5V       100uA      80uA     120uA
10v      120uA      90uA    150uA
|
| [Pullup] table is omitted to signal Open_drain functionality.
|
|------------------------------------------------------------------------------
```

```
|
[Ramp]
|                         typ            min            max
dV/dt_r                   2.0/0.50n      2.0/0.75n      2.0/0.35n
dV/dt_f                   2.0/0.50n      2.0/0.75n      2.0/0.35n
R_load = 500
|
```

==

Fall Back:

When the Submodel_type subparameter under the **[Submodel]** keyword is set to Fall_back, the added model describes the fall back functionality. This submodel can be used to model drivers that reduce their strengths and increase their output impedances during their transitions. The fall back submodel is specified in a restrictive manner consistent with its intended use with a driver model operating only in Driving mode. In a Non-Driving mode, no action is specified. For example, a fall back submodel added to and Input or Terminator model would be inactive.

Existing keywords and subparameters are used to describe fall back models. However, only one **[Pullup]** or **[Pulldown]** table, but not both, is allowed. The switching transition is specified by a **[Ramp]** keyword or by the **[Rising Waveform]** and **[Falling Waveform]** keywords. The **[Ramp]** keyword is required, even if the **[Rising Waveform]** and **[Falling Waveform]** tables exist. However, the voltage ranges and reference voltages are inherited from the top-level model.

For fall back submodels, the **[Submodel Spec]** keyword, V_trigger_r, and V_trigger_f are required. Unlike the bus hold model, the Off_delay subparameter is not permitted. Devices which have both pullup and pulldown structures can be modeled using two submodels, one for the rising cycle and one for the falling cycle.

In all following discussion, "low" means the pulldown structure is on or active, and the pullup structure is off or inactive. The opposite settings are referred to as "high".

The transition is triggered by action at the die using the **[Submodel Spec]** V_trigger_r and V_trigger_f subparameters. The initialization and transitions are set as follows:

INITIAL STATE:

[Pullup] or [Pulldown] Table	Initial Fall Back Submodel State (Off Mode)
[Pullup]	low
[Pulldown]	high

DRIVER RISING CYCLE:

Prior State	Vdie	Rising Edge Transition	Vdie > V_trigger_r Transition
low	<= V_trigger_r	low-to-high	high-to-low
	> V_trigger_r	stays low	stays low
high	<= V_trigger_r	stays high	high-to-low
	> V_trigger_r	stays high	stays high

DRIVER FALLING CYCLE:

Prior State	Vdie	Falling Edge Transition	Vdie < V_trigger_f Transition
high	=> V_trigger_f	high-to-low	low-to-high
	< V_trigger_f	stays high	stays high
low	=> V_trigger_f	stays low	low-to-high

```
|            < V_trigger_f   stays low     stays low
|
|
| One application is to configure the submodel with only a pullup structure.
| At the beginning of the rising edge cycle, the pullup is turned on to the
| high state.  When the die voltage passes V_trigger_r, the pullup structure
| is turned off.  Because only the pullup structure is used, the off state is
| low corresponding to a high-Z state.  During the falling transition, the
| pullup remains in the high-Z state if the V_trigger_f is set out of range to
| avoid setting the submodel to the high state.  So a temporary boost in drive
| occurs only during the first part of the rising cycle.
|
| A similar submodel consisting of only a pulldown structure could be
| constructed to provide added drive strength only at the beginning of the
| falling cycle.  The complete IBIS model would have both submodels to give
| added drive strength for both the start of the rising and the start of the
| falling cycles.
|
| No additional keywords are needed for this functionality.
|------------------------------------------------------------------------------
|
| Complete Dynamic Output Model Example Using Two Submodels:
|
[Submodel]        Dynamic_Output_r
Submodel_type     Fall_back
|
[Submodel Spec]
|   Subparameter          typ        min        max
|
V_trigger_f              -10.0      -10.0      -10.0  | Falling edge trigger
                                                      | set out of range to
                                                      | disable trigger
V_trigger_r                3.1        2.6        4.6  | Rising edge trigger
|
|                           typ        min        max
| [Voltage Range]           5.0        4.5        5.5
| Note, the actual voltage range and reference voltages are inherited from
| the top-level model.
|
[Pullup]
|
-5V      100mA      80mA      120mA
0V       0           0         0
10v     -200mA     -160mA    -240mA
|
| [Pulldown] table is omitted to signify Open_source functionality.
|
|------------------------------------------------------------------------------
|
[Ramp]
|                        typ              min              max
dV/dt_r                  1.5/0.50n        1.43/0.75n       1.58/0.35n
dV/dt_f                  1.5/0.50n        1.43/0.75n       1.58/0.35n
R_load = 50
|
|------------------------------------------------------------------------------
|
[Submodel]        Dynamic_Output_f
Submodel_type     Fall_back
|
[Submodel Spec]
|   Subparameter          typ        min        max
|
V_trigger_r               10.0       10.0       10.0  | Rising edge trigger
                                                      | set out of range to
                                                      | disable trigger
V_trigger_f                1.3        1.2        1.4  | Falling edge trigger
|
|                           typ        min        max
| [Voltage Range]           5.0        4.5        5.5
```

```
| Note, the actual voltage range and reference voltages are inherited from
| the top-level model.
|
[Pulldown]
|
-5V     -100mA     -80mA     -120mA
0V       0           0          0
10v      200mA      160mA     240mA
|
| [Pullup] table is omitted to signify Open_drain functionality.
|
|-----------------------------------------------------------------------------
|
[Ramp]
|                        typ             min             max
dV/dt_r                  1.5/0.50n       1.43/0.75n      1.58/0.35n
dV/dt_f                  1.5/0.50n       1.43/0.75n      1.58/0.35n
R_load = 50
|
|=============================================================================
```

==

Section 7

P A C K A G E   M O D E L I N G

==
==

The **[Package Model]** keyword is optional. If more than the default RLC package model is desired, use the **[Define Package Model]** keyword.

Use the **[Package Model]** keyword within a **[Component]** to indicate the package model for that component. The specification permits .ibs files to contain the following additional list of package model keywords. Note that the actual package models can be in a separate <package_file_name>.pkg file or can exist in the IBIS files between the **[Define Package Model] ... [End Package Model]** keywords for each package model that is defined. For reference, these keywords are listed below. Full descriptions follow. EDA tools that do not support these keywords will ignore all entries between the **[Define Package Model]** and **[End Package Model]** keywords.

[Define Package Model]	Required if the **[Package Model]** keyword is used
[Manufacturer]	(note 1)
[OEM]	(note 1)
[Description]	(note 1)
[Number Of Sections]	(note 2)
[Number Of Pins]	(note 1)
[Pin Numbers]	(note 1)
[Model Data]	(note 2)
[Resistance Matrix]	Optional when **[Model Data]** is used
[Inductance Matrix]	(note 3)
[Capacitance Matrix]	(note 3)
[Bandwidth]	Required (for Banded_matrix matrices only)
[Row]	(note 3)
[End Model Data]	(note 2)
[End Package Model]	(note 1)

(note 1) Required when the **[Define Package Model]** keyword is used
(note 2) Either the **[Number Of Sections]** or the **[Model Data]**/**[End Model Data]** keywords are required. Note that **[Number of Sections]** and the **[Model Data]**/**[End Model Data]** keywords are mutually exclusive.
(note 3) Required when the **[Define Package Model]** keyword is used and the **[Number Of Sections]** keyword is not used.

When package model definitions occur within a .ibs file, their scope is "local" -- they are known only within that .ibs file and no other. In addition, within that .ibs file, they override any globally defined package models that have the same name.

USAGE RULES FOR THE .PKG FILE:

Package models are stored in a file whose name looks like:

<filename>.pkg.

The <filename> provided must adhere to the rules given in Section 3, GENERAL SYNTAX RULES AND GUIDELINES. Use the ".pkg" extension to identify files containing package models. The .pkg file must contain all of the required elements of a normal .ibs file, including **[IBIS Ver]**, **[File Name]**, **[File Rev]**, and the **[End]** keywords. Optional elements include the **[Date]**, **[Source]**, **[Notes]**, **[Disclaimer]**, **[Copyright]**, and **[Comment Char]** keywords. All of the elements follow the same rules as those for a normal .ibs file.

Note that the **[Component]** and **[Model]** keywords are not allowed in the .pkg file. The .pkg file is for package models only.

|==

Keyword: **[Define Package Model]**
Required: Yes
Description: Marks the beginning of a package model description.
Usage Rules: If the .pkg file contains data for more than one package, each section must begin with a new **[Define Package Model]** keyword. The length of the package model name must not exceed 40 characters in length. Blank characters are allowed. For every package model name defined under the **[Package Model]** keyword, there must be a matching **[Define Package Model]** keyword.

|--

[Define Package Model]   QS-SMT-cer-8-pin-pkgs

|==

Keyword: **[Manufacturer]**
Required: Yes
Description: Declares the manufacturer of the component(s) that use this package model.
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 and .pkg files.

|--

[Manufacturer] Quality Semiconductors Ltd.

|==

Keyword: **[OEM]**
Required: Yes
Description: Declares the manufacturer of the package.
Usage Rules: The length of the manufacturer's name must not exceed 40 characters (blank characters are allowed). In addition, each manufacturer must use a consistent name in all .ibs and .pkg files.
Other Notes:: This keyword is useful if the semiconductor vendor sells a single IC in packages from different manufacturers.

|--

[OEM]   Acme Packaging Co.

|==

Keyword: **[Description]**
Required: Yes
Description: Provides a concise yet easily human-readable description of what kind of package the **[Package Model]** is representing.
Usage Rules: The description must be less than 60 characters in length, must fit on a single line, and may contain spaces.

|--

[Description]   220-Pin Quad Ceramic Flat Pack

|==

Keyword: **[Number Of Sections]**
Required: No
Description: Defines the maximum number of sections that make up a 'package stub'. A package stub is defined as the connection between the die pad and the corresponding package pin; it can include (but is not limited to) the bondwire, the connection between the bondwire and pin, and the pin itself. This keyword must be used if a modeler wishes to describe any package stub as other than a single, lumped L/R/C. The sections of a package stub are assumed to connect to each other in a series fashion.
Usage Rules: The argument is a positive integer greater than zero. This keyword, if used, must appear in the specification before the **[Pin Numbers]** keyword. The maximum number of sections includes sections between the Fork and Endfork subparameters.

|--

[Number Of Sections] 3

```
|==============================================================================
     Keyword:  [Number Of Pins]
    Required:  Yes
 Description:  Tells the parser how many pins to expect.
 Usage Rules:  The field must be a positive decimal integer.  The [Number
               Of Pins] keyword must be positioned before the [Pin Numbers]
               keyword.
|------------------------------------------------------------------------------
[Number Of Pins]   128
|
|==============================================================================
     Keyword:  [Pin Numbers]
    Required:  Yes
 Description:  Tells the parser the set of names that are used for the
               package pins and also defines pin ordering.  If the [Number Of
               Sections] keyword is present it also lists the elements for
               each section of a pin's die to pin connection.
  Sub-Params:  Len, L, R, C, Fork, Endfork
 Usage Rules:  Following the [Pin Numbers] keyword, the names of the pins are
               listed.  There must be as many names listed as there are pins
               (as given by the preceding [Number Of Pins] keyword).  Pin
               names can not exceed 5 characters in length.  The first pin
               name given is the "lowest" pin, and the last pin given is the
               "highest."  If the [Number Of Sections] keyword is used then
               each pin name must be followed by one or more of the legal
               subparameter combinations listed below.  If the [Number Of
               Sections] keyword is not present then subparameter usage is
               NOT allowed.

               Subparameters:

               The Len, L, R, and C subparameters specify the length,
               inductance, capacitance and resistance of each section of each
               stub on a package.

               The Fork and Endfork subparameters are used to denote branches
               from the main package stub.

               Len     The length of a package stub section.  Lengths are
                       given in terms of arbitrary 'units'.
               L       The inductance of a package stub section, in terms of
                       'inductance/unit length'.  For example, if the total
                       inductance of a section is 3.0nH and the length of the
                       section is 2 'units', the inductance would be listed
                       as L = 1.5nH  (i.e. 3.0 / 2).
               C       The capacitance of a package stub section, in terms of
                       capacitance per unit length.
               R       The DC (ohmic) resistance of a package stub section,
                       in terms of ohms per unit length.
               Fork    This subparameter indicates that the sections
                       following (up to the Endfork subparameter) are part of
                       a branch off of the main package stub.  This
                       subparameter has no arguments.
               Endfork This subparameter indicates the end point of a branch.
                       For every Fork subparameter there must be a
                       corresponding Endfork subparameter.  As with the Fork
                       subparameter, the Endfork subparameter has no
                       arguments.

               Specifying a Len or L/R/C value of zero is allowed.  If
               Len = 0 is specified, then the L/R/C values are the total for
               that section.  If a non-zero length is specified, then the
               total L/R/C for a section is calculated by multiplying the
               value of the Len subparameter by the value of the L, R, or C
               subparameter.  However, if a non-zero length section is
               specified, the L and C for that section should be treated as
               distributed elements.

               Using The Subparameters to Describe Package Stub Sections:
```

A section description begins with the Len subparameter and ends with the slash (/) character. The value of the Len, L, R, and C subparameters and the subparameter itself are separated by an equals sign (=); white space around the equals sign is optional. The Fork and Endfork subparameters are placed between section descriptions (i.e., between the concluding slash of one section and the 'Len' parameter that starts another). A particular section description can contain no data (i.e., the description is given as 'Len = 0 /').

Legal Subparameter Combinations for Section Descriptions:

A) A single Len = 0 subparameter, followed by a slash. This is used to describe a section with no data.

B) Len, and one or more of the L, R and C subparameters. If the Len subparameter is given as zero, then the L/R/C subparameters represent lumped elements. If the Len subparameter is non-zero, then the L/R/C subparameters represent distributed elements.

C) Single Fork or Endfork subparameter. Normally, a package stub is described as several sections, with the Fork and Endfork subparameters surrounding a group of sections in the middle of the complete package stub description. However, it is legal for the Fork/Endfork subparameters to appear at the end of a section description. The package pin is connected to the last section of a package stub description not surrounded by a Fork/Endfork statements. See the examples below.

Package Stub Boundaries:

A package stub description starts at the connection to the die and ends at the point at which the package pin interfaces with the board or substrate the IC package is mounted on. Note that in the case of a component with through-hole pins, the package stub description should include only the portion of the pin not physically inserted into the board or socket. However, it is legal for a package stub description to include both the component and socket together if this is how the component is intended to be used.

```
|
| A three-section package stub description that includes a bond wire (lumped
| inductance), a trace (treated as a transmission line with DC resistance),
| and a pin modeled as a lumped L/C element.
|
[Pin Numbers]
A1 Len=0 L=1.2n/ Len=1.2 L=2.0n C=0.5p R=0.05/ Len=0 L=2.0n C=1.0p/
|
| Pin A2 below has a section with no data
|
A2 Len=0 L=1.2n/ Len=0/ Len=1.2 L=2.0n C=0.5p R=0.05/ Len=0 L=2.0n C=1.0p/
|
| A section description using the Fork and Endfork subparameters.  Note that
| the indentation of the Fork and Endfork subparameters are for readability
| are not required.
|
A1 Len=0 L=2.3n /            | bondwire
Len=1.2 L=1.0n C=2.5p /      | first section
 Fork                        | indicates the starting of a branch
 Len=1.0 L=2.0n C=1.5p /     | section
 Endfork                     | ending of the branch
Len=0.5 L=1.0 C=2.5p/        | second section
Len=0.0 L=1.5n /             | pin
|
| Here is an example where the Fork/Endfork subparameters are at the end of a
| package stub description.
|
B13 Len=0 L=2.3n /           | bondwire
Len=1.2 L=1.0n C=2.5p /      | first section
```

Len=0.5 L=1.0 C=2.5/	second section, pin connects here
Fork	indicates the starting of a branch
Len=1.0 L=2.0n C=1.5p /	section
Endfork	ending of the branch

==

Keyword: **[Model Data]**

Required: Yes

Description: Indicates the beginning of the formatted package model data, that can include the **[Resistance Matrix]**, **[Inductance Matrix]**, **[Capacitance Matrix]**, **[Bandwidth]**, and **[Row]** keywords.

[Model Data]

==

Keyword: **[End Model Data]**

Required: Yes

Description: Indicates the end of the formatted model data.

Other Notes:: In between the **[Model Data]** and **[End Model Data]** keywords is the package model data itself. The data is a set of three matrices: the resistance (R), inductance (L), and capacitance (C) matrices. Each matrix can be formatted differently (see below). Use one of the matrix keywords below to mark the beginning of each new matrix.

[End Model Data]

==

Keywords: **[Resistance Matrix]**, **[Inductance Matrix]**, **[Capacitance Matrix]**

Required: **[Resistance Matrix]** is optional. If it is not present, its entries are assumed to be zero. **[Inductance Matrix]** and **[Capacitance Matrix]** are required.

Sub-Params: Banded_matrix, Sparse_matrix, or Full_matrix

Description: The subparameters mark the beginning of a matrix, and specify how the matrix data is formatted.

Usage Rules: For each matrix keyword, use only one of the subparameters. After each of these subparameters, insert the matrix data in the appropriate format. (These formats are described in detail below.)

Other Notes:: The resistance, inductance, and capacitance matrices are also referred to as "RLC matrices" within this specification.

When measuring the entries of the RLC matrices, either with laboratory equipment or field-solver software, currents are defined as ENTERING the pins of the package from the board (General Syntax Rule #11). The corresponding voltage drops are to be measured with the current pointing "in" to the "+" sign and "out" of the "-" sign.

```
                I1   +-----+     I2
              ----->  |     |  <------
       board o--------| Pkg |---------o board
               +  V1 - |     | -  V2  +
                     +-----+
```

It is important to observe this convention in order to get the correct signs for the mutual inductances and resistances.

[Resistance Matrix] *Banded_matrix*
[Inductance Matrix] *Sparse_matrix*
[Capacitance Matrix] *Full_matrix*

==

RLC MATRIX NOTES:

For each **[Resistance Matrix]**, **[Inductance Matrix]**, or **[Capacitance Matrix]** a different format can be used for the data. The choice of formats is provided to satisfy different simulation accuracy and speed requirements. Also, there are many packages in which the resistance matrix can have no coupling terms at all. In this case, the most concise format (Banded_matrix) can be used.

There are two different ways to extract the coefficients that are reported in the capacitance and inductance matrices. For the purposes of this specification, the coefficients reported in the capacitance matrices shall be the 'electrostatic induction coefficients' or 'Maxwell's capacitances'. The Maxwell capacitance Kij is defined as the charge induced on conductor "j" when conductor "i" is held at 1 volt and all other conductors are held at zero volts. Note that Kij (when i /= j) will be a negative number and should be entered as such. Likewise, for the inductance matrix the coefficients for Lij are defined as the voltage induced on conductor "j" when conductor "i"'s current is changed by 1amp/sec and all other conductors have no current change.

One common aspect of all the different formats is that they exploit the symmetry of the matrices they describe. This means that the entries below the main diagonal of the matrix are identical to the corresponding entries above the main diagonal. Therefore, only roughly one-half of the matrix needs to be described. By convention, the main diagonal and the UPPER half of the matrix are provided.

In the following text, we use the notation **[I, J]** to refer to the entry in row I and column J of the matrix. Note that I and J are allowed to be alphanumeric strings as well as integers. An ordering of these strings is defined in the **[Pin Numbers]** section. In the following text, "Row 1" means the row corresponding to the first pin.

Also note that the numeric entries of the RLC matrices are standard IBIS floating point numbers. As such, it is permissible to use metric "suffix" notation. Thus, an entry of the C matrix could be given as 1.23e-12 or as 1.23p or 1.23pF.

Full_matrix:

When the Full_matrix format is used, the couplings between every pair of elements is specified explicitly. Assume that the matrix has N rows and N columns. The Full_matrix is specified one row at a time, starting with Row 1 and continuing down to Row N.

Each new row is identified with the Row keyword.

==

Keyword: **[Row]**
Required: Yes
Description: Indicates the beginning of a new row of the matrix.
Usage Rules: The argument must be one of the pin names listed under the **[Pin Numbers]** keyword.

[Row]   3

==

Following a **[Row]** keyword is a block of numbers that represent the entries for that row. Suppose that the current row is number M. Then the first number listed is the diagonal entry, **[M,M]**. Following this number are the entries of the upper half of the matrix that belong to row M: **[M, M+1]**, **[M, M+2]**, ... up to **[M,N]**.

For even a modest-sized package, this data will not all fit on one line. You can break the data up with new-line characters so that the 80 character line length limit is observed.

An example: suppose the package has 40 pins and that we are currently working on Row 19. There is 1 diagonal entry, plus 40 - 19 = 21 entries in the upper half of the matrix to be specified, for 22 entries total. The data might be formatted as follows:

```
[Row]    19
5.67e-9  1.1e-9  0.8e-9  0.6e-9  0.4e-9  0.2e-9   0.1e-9   0.09e-9
8e-10    7e-10   6e-10   5e-10   4e-10   3e-10    2e-10    1e-10
```

9e-11 8e-11 7e-11 6e-11 5e-11 4e-11

In the above example, the entry 5.67e-9 is on the diagonal of row 19.

Observe that Row 1 always has the most entries, and that each successive row has one fewer entry than the last; the last row always has just a single entry.

Banded_matrix:

A Banded_matrix is one whose entries are guaranteed to be zero if they are farther away from the main diagonal than a certain distance, known as the "bandwidth." Let the matrix size be N x M, and let the bandwidth be B. An entry **[I,J]** of the matrix is zero if:

$$| I - J | > B$$

where |.| denotes the absolute value.

The Banded_matrix is used to specify the coupling effects up to B pins on either side. Two variations are supported. One allows for the coupling to circle back on itself. This is technically a simple form of a bordered block diagonal matrix. However, its data can be completely specified in terms of a Banded_matrix for an N x M matrix consisting of N rows and M = N + B columns. The second variation is just in terms of an N x N matrix where no circle back coupling needs to be specified.

The bandwidth for a Banded_matrix must be specified using the **[Bandwidth]** keyword:

Keyword:	**[Bandwidth]**
Required:	Yes (for Banded_matrix matrices only)
Description:	Indicates the bandwidth of the matrix.
Usage Rules:	The bandwidth field must be a non-negative integer. This keyword must occur after the **[Resistance Matrix]**, etc., keywords, and before the matrix data is given.

[Bandwidth] 10

Specify the banded matrix one row at a time, starting with row 1 and working up to higher rows. Mark each row with the **[Row]** keyword, as above. As before, symmetry is exploited: do not provide entries below the main diagonal.

For the case where coupling can circle back on itself, consider a matrix of N pins organized into N rows 1 ... N and M columns 1 ... N, 1 ... B. The first row only needs to specify the entries **[1,1]** through **[1,1+B]** since all other entries are guaranteed to be zero. The second row will need to specify the entries **[2,2]** through **[2,2+B]**, and so on. For row K the entries **[K,K]** through **[K,K+B]** are given when K + B is less than or equal to the size of the matrix N. When K + B exceeds N, the entries in the last columns 1 ... B specify the coupling to the first rows. For row K, the entries **[K,K] ... [K,N] [K,1] ... [K,R]** are given where R = mod(K + B - 1, N) + 1. All rows will contain B + 1 entries. To avoid redundant entries, the bandwidth is limited to B <= int((N - 1) / 2).

For the case where coupling does not circle back on itself, the process is modified. Only N columns need to be considered. When K + B finally exceeds the size of the matrix N, the number of entries in each row starts to decrease; the last row (row N) has only 1 entry. This construction constrains the bandwidth to B < N.

As in the Full_matrix, if all the entries for a particular row do not fit into a single 80-character line, the entries can be broken across several lines.

It is possible to use a bandwidth of 0 to specify a diagonal matrix (a matrix with no coupling terms.) This is sometimes useful for resistance

```
| matrices.
|
| Sparse_matrix:
|
| A Sparse_matrix is expected to consist mostly of zero-valued entries, except
| for a few nonzeros.  Unlike the Banded_matrix, there is no restriction on
| where the nonzero entries can occur.  This feature is useful in certain
| situations, such as for Pin Grid Arrays (PGAs).
|
| As usual, symmetry can be exploited to reduce the amount of data by
| eliminating from the matrix any entries below the main diagonal.
|
| An N x N Sparse_matrix is specified one row at a time, starting with row 1
| and continuing down to row N.  Each new row is marked with the [Row]
| keyword, as in the other matrix formats.
|
| Data for the entries of a row is given in a slightly different format,
| however.  For the entry [I, J] of a row, it is necessary to explicitly list
| the name of pin J before the value of the entry is given.  This
| specification serves to indicate to the parser where the entry is put into
| the matrix.
|
| The proper location is not otherwise obvious because of the lack of
| restrictions on where nonzeros can occur.  Each (Index, Value) pair is
| listed upon a separate line.  An example follows.  Suppose that row 10 has
| nonzero entries [10,10], [10,11], [10,15], and [10,25].  The following row
| data would be provided:
|
[Row]   10
| Index          Value
10               5.7e-9
11               1.1e-9
15               1.1e-9
25               1.1e-9
|
| Note that each of the column indices listed for any row must be greater than
| or equal to the row index, because they always come from the upper half of
| the matrix.  When alphanumeric pin names are used, special care must be
| taken to ensure that the ordering defined in the [Pin Numbers] section is
| observed.
|
| With this convention, please note that the Nth row of an N x N matrix has
| just a single entry (the diagonal entry).
|
|==============================================================================
|     Keyword:  [End Package Model]
|    Required:  Yes
| Description:  Marks the end of a package model description.
| Usage Rules:  This keyword must come at the end of each complete package
|               model description.
|
|               Optionally, add a comment after the [End Package Model]
|               keyword to clarify which package model has just ended.  For
|               example,
|
|               [Define Package Model]   My_Model
|               |
|               |  ... content of model ...
|               |
|               [End Package Model]      | end of My_Model
|------------------------------------------------------------------------------
[End Package Model]
|
|==============================================================================
|                             Package Model Example
|
| The following is an example of a package model file following the
| package modeling specifications.  For the sake of brevity, an 8-pin package
| has been described.  For purposes of illustration, each of the matrices is
| specified using a different format.
```

```
|
|==============================================================================
|
[IBIS Ver]       4.0
[File Name]      example.pkg
[File Rev]       0.1
[Date]           June 1, 2002
[Source]         Quality Semiconductors.  Data derived from Helmholtz Inc.'s
                 field solver using 3-D Autocad model from Acme Packaging.
[Notes]          Example of couplings in packaging
[Disclaimer]     The models given below may not represent any physically
                 realizable 8-pin package.  They are provided solely for the
                 purpose of illustrating the .pkg file format.
|
|==============================================================================
|
[Define Package Model]  QS-SMT-cer-8-pin-pkgs
[Manufacturer]          Quality Semiconductors Ltd.
[OEM]                   Acme Package Co.
[Description]           8-Pin ceramic SMT package
[Number Of Pins]        8
|
[Pin Numbers]
1
2
3
4
5
6
7
8
|
[Model Data]
|
| The resistance matrix for this package has no coupling
|
[Resistance Matrix]     Banded_matrix
[Bandwidth]             0
[Row]   1
10.0
[Row]   2
15.0
[Row]   3
15.0
[Row]   4
10.0
[Row]   5
10.0
[Row]   6
15.0
[Row]   7
15.0
[Row]   8
10.0
|
| The inductance matrix has loads of coupling
|
[Inductance Matrix]     Full_matrix
[Row]   1
3.04859e-07      4.73185e-08      1.3428e-08     6.12191e-09
1.74022e-07      7.35469e-08     2.73201e-08     1.33807e-08
[Row]   2
3.04859e-07      4.73185e-08      1.3428e-08     7.35469e-08
1.74022e-07      7.35469e-08     2.73201e-08
[Row]   3
3.04859e-07      4.73185e-08     2.73201e-08     7.35469e-08
1.74022e-07      7.35469e-08
[Row]   4
3.04859e-07      1.33807e-08     2.73201e-08     7.35469e-08
1.74022e-07
```

```
[Row]   5
4.70049e-07     1.43791e-07     5.75805e-08     2.95088e-08
[Row]   6
4.70049e-07     1.43791e-07     5.75805e-08
[Row]   7
4.70049e-07     1.43791e-07
[Row]   8
4.70049e-07
|
| The capacitance matrix has sparse coupling
|
[Capacitance Matrix]    Sparse_matrix
[Row]   1
1       2.48227e-10
2       -1.56651e-11
5       -9.54158e-11
6       -7.15684e-12
[Row]   2
2       2.51798e-10
3       -1.56552e-11
5       -6.85199e-12
6        -9.0486e-11
7       -6.82003e-12
[Row]   3
3       2.51798e-10
4       -1.56651e-11
6       -6.82003e-12
7        -9.0486e-11
8       -6.85199e-12
[Row]   4
4       2.48227e-10
7       -7.15684e-12
8       -9.54158e-11
[Row]   5
5       1.73542e-10
6       -3.38247e-11
[Row]   6
6       1.86833e-10
7       -3.27226e-11
[Row]   7
7       1.86833e-10
8       -3.38247e-11
[Row]   8
8       1.73542e-10
|
[End Model Data]
[End Package Model]
|
|================================================================================
```

==

Section 8

E L E C T R I C A L   B O A R D   D E S C R I P T I O N

==
==

A "board level component" is the generic term to be used to describe a printed circuit board (PCB) or substrate which can contain components or even other boards, and which can connect to another board through a set of user visible pins. The electrical connectivity of such a board level component is referred to as an "Electrical Board Description". For example, a SIMM module is a board level component that is used to attach several DRAM components on the PCB to another board through edge connector pins. An electrical board description file (a .ebd file) is defined to describe the connections of a board level component between the board pins and its components on the board.

A fundamental assumption regarding the electrical board description is that the inductance and capacitance parameters listed in the file are derived with respect to well-defined reference plane(s) within the board. Also, this current description does not allow one to describe electrical (inductive or capacitive) coupling between paths. It is recommended that if coupling is an issue, then an electrical description be extracted from the physical parameters of the board.

What is, and is not, included in an Electrical Board Description is defined by its boundaries. For the definition of the boundaries, see the Description section under the **[Path Description]** Keyword.

USAGE RULES:

A .ebd file is intended to be a stand-alone file, not associated with any .ibs file. Electrical Board Descriptions are stored in a file whose name looks like <filename>.ebd, where <filename> must conform to the naming rules given in the General Syntax Section of this specification. The .ebd extension is mandatory.

CONTENTS:

A .ebd file is structured similar to a standard IBIS file. It must contain the following keywords, as defined in the IBIS specification: **[IBIS Ver]**, **[File Name]**, **[File Rev]**, and **[End]**. It may also contain the following optional keywords: **[Comment Char]**, **[Date]**, **[Source]**, **[Notes]**, **[Disclaimer]**, and **[Copyright]**. The actual board description is contained between the keywords **[Begin Board Description]** and **[End Board Description]**, and includes the keywords listed below:

[Begin Board Description]
[Manufacturer]
[Number Of Pins]
[Pin List]
[Path Description]
[Reference Designator Map]
[End Board Description]

More than one **[Begin Board Description]**/**[End Board Description]** keyword pair is allowed in a .ebd file.

==

Keyword: **[Begin Board Description]**
Required: Yes
Description: Marks the beginning of an Electrical Board Description.
Usage Rules: The keyword is followed by the name of the board level component. If the .ebd file contains more than one **[Begin Board Description]** keyword, then each name must be unique. The length of the component name must not exceed 40 characters in length, and blank characters are allowed. For every

```
|                 [Begin Board Description] keyword there must be a matching
|                 [End Board Description] keyword.
|------------------------------------------------------------------------------
[Begin Board Description]  16Meg X 8 SIMM Module
|
|==============================================================================
|      Keyword:  [Manufacturer]
|     Required:  Yes
|  Description:  Declares the manufacturer of the components(s) that use this
|                .ebd file.
|  Usage Rules:  Following the keyword is the manufacturer's name.  It must not
|                exceed 40 characters, and can include blank characters.  Each
|                manufacturer must use a consistent name in all .ebd files.
|------------------------------------------------------------------------------
[Manufacturer] Quality SIMM Corp.
|
|==============================================================================
|      Keyword:  [Number Of Pins]
|     Required:  Yes
|  Description:  Tells the parser the number of pins to expect.  Pins are any
|                externally accessible electrical connection to the component.
|  Usage Rules:  The field must be a positive decimal integer.  Note: The
|                simulator must not limit the Number Of Pins to any value less
|                than 1,000.  The [Number Of Pins] keyword must be positioned
|                before the [Pin List] keyword.
|------------------------------------------------------------------------------
[Number Of Pins] 128
|
|==============================================================================
|      Keyword:  [Pin List]
|     Required:  Yes
|  Description:  Tells the parser the pin names of the user accessible pins.
|                It also informs the parser which pins are connected to power
|                and ground.
|   Sub-Params:  signal_name
|  Usage Rules:  Following the [Pin List] keyword are two columns.  The first
|                column lists the pin name while the second lists the data book
|                name of the signal connected to that pin. There must be as
|                many pin_name/signal_name rows as there are pins given by the
|                preceding [Number Of Pins] keyword.  Pin names must be the
|                alphanumeric external pin names of the part.  The pin names
|                cannot exceed eight characters in length.  Any pin associated
|                with a signal name that begins with "GND" or "POWER" will be
|                interpreted as connecting to the boards ground or power plane.
|                In addition, NC is a legal signal name and indicates that the
|                Pin is a 'no connect'.  As per the IBIS standard "GND",
|                "POWER" and "NC" are case insensitive.
|------------------------------------------------------------------------------
|  A SIMM Board Example:
|
[Pin List]  signal_name
 A1          GND
 A2          data1
 A3          data2
 A4          POWER5     | This pin connects to 5 V
 A5          NC         | a no connect pin
| .
| .
 A22         POWER3.3   | This pin connects to 3.3 V
 B1          casa
| .
| .
|etc.
|
|==============================================================================
|      Keyword:  [Path Description]
|     Required:  Yes
|  Description:  This keyword allows the user to describe the connection
|                between the user accessible pins of a board level component
|                and other pins or pins of the ICs mounted on that board.  Each
```

pin to node connection is divided into one or more cascaded "sections", where each section is described in terms of its L/R/C per unit length. The Fork and Endfork subparameters allow the path to branch to multiple nodes, or another pin. A path description is required for each pin whose signal name is not "GND", "POWER" or "NC".

Board Description and IC Boundaries:

In any system, each board level component interfaces with another board level component at some boundary. Every electrical board description must contain the components necessary to represent the behavior of the board level component being described within its boundaries. The boundary definition depends upon the board level component being described.

For CARD EDGE CONNECTIONS such as a SIMM or a PC Daughter Card plugged into a SIMM Socket or Edge Connector, the boundary should be at the end of the board card edge pads as they emerge from the connector.

For any THROUGH-HOLE MOUNTED COMPONENT, the boundary will be at the surface of the board on which the component is mounted.

SURFACE MOUNTED COMPONENT models end at the outboard end of their recommended surface mount pads.

If the board level component contains an UNMATED CONNECTOR, the unmated connector will be described in a separate file, with its boundaries being as described above for the through-hole or surface mounted component.

Sub-Params: Len, L, R, C, Fork, Endfork, Pin, Node

Usage Rules: Each individual connection path (user pin to node(s)) description begins with the **[Path Description]** keyword and a path name, followed by the subparameters used to describe the path topology and the electrical characteristics of each section of the path. The path name must not exceed 40 characters, blanks are not allowed, and each occurrence of the **[Path Description]** keyword must be followed by a unique path name. Every signal pin (pins other than POWER, GND or NC) must appear in one and only one path description per **[Begin Board Description]**/**[End Board Description]** pair. Pin names do not have to appear in the same order as listed in the **[Pin List]** table. The individual subparameters are broken up into those that describe the electrical properties of a section, and those that describe the topology of a path.

Section Description Subparameters:

The Len, L, R, and C subparameters specify the length, the series inductance, resistance, and the capacitance to ground of each section in a path description.

Len The physical length of a section. Lengths are given in terms of arbitrary 'units'. Any non-zero length requires that the parameters that follow must be interpreted as distributed elements by the simulator.

L The series inductance of a section, in terms of 'inductance/unit length'. For example, if the total inductance of a section is 3.0 nH and the length of the section is 2 'units', the inductance would be listed as L = 1.5 nH (i.e. 3.0 / 2).

C The capacitance to ground of a section, in terms of capacitance per unit length.

R The series DC (ohmic) resistance of a section, in terms of ohms per unit length.

Topology Description Subparameters:

The Fork and Endfork subparameters denote branches from the main pin-to-node or pin-to-pin connection path. The Node subparameter is used to reference the pin of a component or board as defined in a .ibs or .ebd file. The Pin subparameter is used to indicate the point at which a path connects to a user visible pin.

Fork This subparameter indicates that the sections following (up to the Endfork subparameter) are part of a branch off of the main connection path. This subparameter has no arguments.

Endfork This subparameter indicates the end point of a branch. For every Fork subparameter there must be a corresponding Endfork subparameter. As with the Fork subparameter, the Endfork subparameter has no arguments. The Fork and Endfork parameters must appear on separate lines.

Node reference_designator.pin
This subparameter is used when the connection path connects to a pin of another, externally defined component. The arguments of the Node subparameter indicate the pin and reference designator of the external component. The pin and reference designator portions of the argument are separated by a period ("."). The reference designator is mapped to an external component description (another .ebd file or .ibs file) by the **[Reference Designator Map]** Keyword. Note that a Node MUST reference a model of a passive or active component. A Node is not an arbitrary connection point between two elements or paths.

Pin This subparameter is used to mark the point at which a path description connects to a user accessible pin. Every path description must contain at least one occurrence of the Pin subparameter. It may also contain the reserved word NC. The value of the Pin subparameter must be one of the pin names listed in the **[Pin List]** section.

Note: The reserved word NC can also be used in path descriptions in a similar manner as the subparameters in order to terminate paths. This usage is optional.

Using The Subparameters to Describe Paths:

A section description begins with the Len subparameter and ends with the slash (/) character. The value of the Len, L, R, and C subparameters and the subparameter itself are separated by an equals sign (=); white space around the equals sign is optional. The Fork, Endfork, Node and Pin subparameters are placed between section descriptions (i.e., between the concluding slash of one section and the 'Len' parameters that starts another). The arguments of the Pin and Node subparameter are separated by white space.

Specifying a Len or L/R/C value of zero is allowed. If Len = 0 is specified, then the L/R/C values are the total for that section. If a non-zero length is specified, then the total L/R/C for a section is calculated by multiplying the value of the Len subparameter by the value of the L, R, or C subparameter. However, as noted below, if a non-zero length is specified, that section MUST be interpreted as distributed elements.

Legal Subparameter Combinations for Section Descriptions:

A) Len, and one or more of the L, R and C subparameters. If the Len subparameter is given as zero, then the L/R/C subparameters represent lumped elements. If the Len subparameter is non-zero, then the L/R/C subparameters represent distributed elements and both L and C must be specified, R is optional. The segment Len ..../ must not be split; the whole segment must be on one line.

B) The first subparameter following the **[Path Description]** keyword must be 'Pin', followed by one or more section descriptions. The path description can terminate in a Node, another pin or the reserved word, NC. However, NC may be optionally omitted.

Dealing With Series Elements:

A discrete series R or L component can be included in a path description by defining a section with Len=0 and the proper R or L value. A discrete series component can also be included in a path description by writing two back to back node statements that reference the same component (see the example below). Note that both ends of a discrete, two terminal component MUST be contained in a single **[Path Description]**. Connecting two separate **[Path Description]**s with a series component is not allowed.

An Example Path For a SIMM Module:

```
[Path Description] CAS_2
Pin J25
Len = 0.5 L=8.35n C=3.34p R=0.01 /
Node u21.15
Len = 0.5 L=8.35n C=3.34p R=0.01 /
Node u22.15
Len = 0.5 L=8.35n C=3.34p R=0.01 /
Node u23.15
```

```
        +--------------------------------------------------------------
        |
        |         _______         _______         _______
  J25 <---------O_______O---+---O_______O---+---O_______O---+
        |         Len=0.5   |     Len=0.5   |     Len=0.5   |
        |                   |               |               |
        |                +--+--+         +--+--+         +--+--+
        |                |Pin15|         |Pin15|         |Pin15|
        |                |     |         |     |         |     |
        |                | U21 |         | U22 |         | U23 |
        |                |     |         |     |         |     |
        |
```

A Description Using The Fork and Endfork Subparameters:

```
[Path Description] PassThru1
Pin B5
Len = 0   L=2.0n /
Len = 2.1 L=6.0n C=2.0p /
 Fork
 Len = 1.0 L = 1.0n C= 2.0p /
 Node u23.16
 Endfork
Len = 1.0 L = 6.0n C=2.0p /
Pin A5
```

```
       +-------------------------------------------------------------
       |
       |                ______________
   A5 <----------------O______________O-------------+
       |                   Len=1.0                  |
       |                                            |
       |           _____________________________    |
   B5 <---@@@@@---O_____________________________O---+
       |   2 nH               Len=2.1               |
       |                                            |
       |                          ______________    |
       |                     +---O______________O---+
       |                     |         Len=1.0
       |                     |
       |                  +--+--+
       |                  |Pin16|
       |                  |     |
       |                  | U23 |
       |                  |     |
       |
```

A Description Including a Discrete Series Element:

[Path Description] sig1
Pin B27
Len = 0 L=1.6n /
Len = 1.5 L=6.0n C=2.0p /
Node R2.1
Node R2.2
Len = 0.25 L=6.0n C=2.0p /
Node U25.6

```
       +-------------------------------------------------------------
       |
       |                                 +-----------+
       |           __________________    |Pin     Pin|    ___
  B27 <---@@@@@---O__________________O---+ 1       2 +---O___O---+
       |  1.6 nH        Len=1.5          |    R2     |  Len=0.25 |
       |                                 +-----------+           |
       |                                                      +--+--+
       |                                                      |Pin 6|
       |                                                      |     |
       |                                                      | U25 |
       |                                                      |     |
       |
```

==

Keyword: **[Reference Designator Map]**

Required: Yes, if any of the path descriptions use the Node subparameter

Description: Maps a reference designator to a component or electrical board description contained in an .ibs or .ebd file.

Usage Rules: The **[Reference Designator Map]** keyword must be followed by a list of all of the reference designators called out by the Node subparameters used in the various path descriptions. Each reference designator is followed by the name of the .ibs or .ebd file containing the electrical description of the component or board, then the name of the component itself as given by the .ibs or .ebd file's **[Component]** or **[Begin Board Description]** keyword respectively. The reference designator, file name and component name terms are separated by white space. By default the .ibs or .ebd files are assumed to exist in the same directory as the calling .ebd file. It is legal for a reference designator to point to a component that is contained in the calling .ebd file.

The reference designator is limited to ten characters.

[Reference Designator Map]

External Part References:

```
|
| Ref Des  File name   Component name
u23        pp100.ibs   Pentium(R)__Pro_Processor
u24        simm.ebd    16Meg X 36 SIMM Module
u25        ls244.ibs   National 74LS244a
u26        r10K.ibs    My_10K_Pullup
|
```

|==

Keyword: **[End Board Description]**

Required: Yes

Description: Marks the end of an Electrical Interconnect Description.

Usage Rules: This keyword must come at the end of each complete electrical interconnect model description.

Optionally, a comment may be added after the **[End Electrical Description]** keyword to clarify which board model has ended.

|--

```
[End Board Description]        | End: 16Meg X 8 SIMM Module
|
```

|==

Keyword: **[End]**

Required: Yes

Description: Defines the end of the .ibs, .pkg, or .ebd file.

|--

```
[End]
|
```

|==

==

Section 9

N O T E S   O N   D A T A   D E R I V A T I O N   M E T H O D

==
==

This section explains how data values are derived. It describes certain assumed parameter and table extraction conditions if they are not explicitly specified. It also describes the allocation of data into the "typ", "min", and "max" columns under variations of voltage, temperature, and process.

The required "typ" column for all data represents typical operating conditions. For most **[Model]** keyword data, the "min" column describes slow, weak performance, and the "max" column describes the fast, strong performance. It is permissible to use slow, weak components or models to derive the data for the "min" column, and to use fast, strong components or models to derive the data in the "max" columns under the corresponding voltage and temperature derating conditions for these columns. It is also permissible to use typical components or models derated by voltage and temperature and optionally apply proprietary "X%" and "Y%" factors described later for further derating. This methodology has the nice feature that the data can be derived either from semiconductor vendor proprietary models, or typical component measurement over temperature/voltage.

The voltage and temperature keywords and optionally the process models control the conditions that define the "typ", "min", and "max" column entries for all I-V table keywords **[Pulldown]**, **[Pullup]**, **[GND Clamp]**, and **[POWER Clamp]**; all **[Ramp]** subparameters dV/dt_r and dV/dt_f; and all waveform table keywords and subparameters **[Rising Waveform]**, **[Falling Waveform]**, V_fixture, V_fixture_min, and V_fixture_max.

The voltage keywords that control the voltage conditions are **[Voltage Range]**, **[Pulldown Reference]**, **[Pullup Reference]**, **[GND Clamp Reference]**, and **[POWER Clamp Reference]**. The entries in the "min" columns contain the smallest magnitude voltages, and the entries in the "max" columns contain the largest magnitude voltages.

The optional **[Temperature Range]** keyword will contain the temperature which causes or amplifies the slow, weak conditions in the "min" column and the temperature which causes or amplifies the fast, strong conditions in the "max" column. Therefore, the "min" column for **[Temperature Range]** will contain the lowest value for bipolar models (TTL and ECL) and the highest value for CMOS models. Default values described later are assumed if temperature is not specified.

The "min" and "max" columns for all remaining keywords and subparameters will contain the smallest and largest magnitude values. This applies to the **[Model]** subparameter C_comp as well even if the correlation to the voltage, temperature, and process variations are known because information about such correlation is not available in all cases.

C_comp is considered an independent variable. This is because C_comp includes bonding pad capacitance, which does not necessarily track fabrication process variations. The conservative approach to using IBIS data will associate large C_comp values with slow, weak models, and the small C_comp values with fast, strong models.

The default temperatures under which all I-V tables are extracted are provided below. The same defaults also are stated for the **[Ramp]** subparameters, but they also apply for the waveform keywords.

The stated voltage ranges for I-V tables cover the most common, single supply cases. When multiple supplies are specified, the voltages shall extend similarly to values that handle practical extremes in reflected wave simulations.

For the **[Ramp]** subparameters, the default test load and voltages are provided. However, the test load can be entered directly by the R_load subparameter. The allowable test loads and voltages for the waveform keywords are stated by required and optional subparameters; no defaults are needed. Even with waveform keywords, the **[Ramp]** keyword continues to be required so that the IBIS model remains functional in situations which do not support waveform processing.

The following discussion lists test details and default conditions.

1) I-V Tables:

I-V tables for CMOS models:

typ = typical voltage, typical temp deg C, typical process
min = minimum voltage, max temp deg C, typical process, minus "X%"
max = maximum voltage, min temp deg C, typical process, plus "X%"

I-V tables for bipolar models:

typ = typical voltage, typical temp deg C, typical process
min = minimum voltage, min temp deg C, typical process, minus "X%"
max = maximum voltage, max temp deg C, typical process, plus "X%"

Nominal, min, and max temperature are specified by the semiconductor vendor. The default range is 50 deg C nom, 0 deg C min, and 100 deg C max temperatures.

X% should be statistically determined by the semiconductor vendor based on numerous fab lots, test chips, process controls, etc.. The value of X need not be published in the IBIS file, and may decrease over time as data on the I/O buffers and silicon process increases.

Temperatures are junction temperatures.

2) Voltage Ranges:

Points for each table must span the voltages listed below:

Table	Low Voltage	High Voltage
[Pulldown]	GND - POWER	POWER + POWER
[Pullup]	GND - POWER	POWER + POWER
[GND Clamp]	GND - POWER	GND + POWER
[POWER Clamp]	POWER	POWER + POWER
[Series Current]	GND - POWER	GND + POWER
[Series MOSFET]	GND	GND + POWER

As described in the **[Pulldown Reference]** keyword section, the I-V tables of the **[Pullup]** and the **[POWER Clamp]** structures are 'Vcc relative', using the equation: Vtable = Vcc - Voutput.

For example, a model with a 5 V power supply voltage should be characterized between (0 - 5) = -5 V and (5 + 5) = 10 V; and a model with a 3.3 V power supply should be characterized between (0 - 3.3) = -3.3 V and (3.3 + 3.3) = 6.6 V for the **[Pulldown]** table.

When tabulating output data for ECL type models, the voltage points must span the range of Vcc to Vcc - 2.2 V. This range applies to both the **[Pullup]** and **[Pulldown]** tables. Note that this range applies ONLY when characterizing an ECL output.

These voltage ranges must be spanned by the IBIS data. Data derived from lab measurements may not be able to span these ranges as such and so may need to be extrapolated to cover the full range. This data must not be left for the simulator to provide.

3) Ramp Rates:

The following steps assume that the default load resistance of 50 ohms is used. There may be models that will not drive a load of only 50 ohms into any useful level of dynamics. In these cases, use the semiconductor vendor's suggested (nonreactive) load and add the load subparameter to the **[Ramp]** specification.

The ramp rate does not include packaging but does include the effects of the C_comp parameter; it is the intrinsic output stage rise and fall time only.

The ramp rates (listed in AC characteristics below) should be derived as follows:

a. If starting with the silicon model, remove all packaging. If starting with a packaged model, perform the measurements as outlined below. Then use whatever techniques are appropriate to derive the actual, unloaded rise and fall times.

b. If: The Model_type is one of the following: Output, I/O, or 3-state (not open or ECL types);
   Then: Attach a 50 ohm resistor to GND to derive the rising edge ramp. Attach a 50 ohm resistor to POWER to derive the falling edge ramp.

   If: The Model_type is Output_ECL, I/O_ECL, 3-state_ECL;
   Then: Attach a 50 ohm resistor to the termination voltage (Vterm = VCC - 2 V). Use this load to derive both the rising and falling edges.

   If: The Model_type is either an Open_sink type or Open_drain type;
   Then: Attach either a 50 ohm resistor or the semiconductor vendor suggested termination resistance to either POWER or the suggested termination voltage. Use this load to derive both the rising and falling edges.

   If: The Model_type is an Open_source type;
   Then: Attach either a 50 ohm resistor or the semiconductor vendor suggested termination resistance to either GND or the suggested termination voltage. Use this load to derive both the rising and falling edges.

c. Due to the resistor, output swings will not make a full transition as expected. However the pertinent data can still be collected as follows:
   1) Determine the 20% to 80% voltages of the 50 ohm swing.
   2) Measure this voltage change as "dV".
   3) Measure the amount of time required to make this swing "dt".

d. Post the value as a ratio "dV/dt". The simulator extrapolates this value to span the required voltage swing range in the final model.

e. Typ, Min, and Max must all be posted, and are derived at the same extremes as the I-V tables, which are:

```
Ramp rates for CMOS models:
   typ = typical voltage, typical temp deg C, typical process
   min = minimum voltage, max temp deg C, typical process, minus "Y%"
   max = maximum voltage,  min temp deg C, typical process, plus  "Y%"

Ramp rates for bipolar models:
   typ = typical voltage, typical temp deg C, typical process
   min = minimum voltage, min temp deg C, typical process, minus "Y%"
   max = maximum voltage,  max temp deg C, typical process, plus  "Y%"
```

   where nominal, min, and max temp are specified by the semiconductor vendor. The preferred range is 50 deg C nom, 0 deg C min, and 100 deg C max temperatures.

   Note that the derate factor, "Y%", may be different than that used for the I-V table data. This factor is similar to the X% factor described above. As in the case of I-V tables, temperatures are junction temperatures.

f. During the I-V measurements, the driving waveform should have a rise/fall time fast enough to avoid thermal feedback. The specific choice of sweep time is left to the modeling engineer.

4) Transit Time Extractions:
The transit time parameter is indirectly derived to be the value that produces the same effect as that extracted by the reference measurement or reference simulation.

The test circuit consists of the following:

a) A pulse source (10 ohms, 1 ns at full duration ramp) or equivalent and transitioning between Vcc and 0 V,
b) A 50 ohm, 1 ns long trace or transmission line,
c) A 500 ohm termination to the ground clamp reference voltage for TTgnd extraction and to the power clamp reference voltage for TTpower extraction (to provide a convenient, minimum loading 450 ohm - 50 ohm divider for high-speed sampling equipment observation of the component denoted as the device under test), and
d) The device under test (DUT).

```
                                        DUT with [GND Clamp]
                                           | /|
                      _____________        |/ |
          o---/\/\/\--O_____________)---o--|< |--o GND
             10 ohms    Z0 = 50 ohm     |  | \|  |
                        TD = 1 ns       |          |
                                        |-/\/\/\-|
                                        | 500 ohm Load for Probing
 Vcc ---\                 ------\       |
        \                       \      o      /--\
 0 V      \------                 \          /    \-------
        | |                        \---------/
        1 ns                    |<----------->|
   1 ns, 10 ohm             Choose TTgnd that matches the measured
   Source Signal            delay with the IBIS model simulation delay
```

Example of TTgnd Extraction Setup

The TTgnd extraction will be done only if a **[GND Clamp]** table exists. A high to low transition that produces a positive "glitch", perhaps several nanoseconds later indicates a stored charge in the ground clamp circuit. The test circuit is simulated using the complete IBIS model with C_comp and the Ct model defined under the **[TTgnd]** and **[TTpower]** keywords. An effective TTgnd value that produces a "glitch" with the same delay is extracted.

Similarly, the TTpower extraction will be done only if a **[POWER Clamp]** table exists. A low to high transition that produces a negative "glitch", perhaps several nanoseconds later indicates a stored charge in the power clamp circuit. An effective TTpower value that produces a glitch with the same delay is extracted.

It is preferred to do the extractions with the package parameters removed. However, if the extraction is done from measurements, then the package model should be included in the IBIS based simulation.

5) Series MOSFET Table Extractions:
An extraction circuit is set up according to the figure below. The switch is configured into the 'On' state. This assumes that the Vcc voltage will be applied to the gate by internal logic. Designate one pin of the switch as the source node, and the other pin as the drain node. The Table Currents designated as Ids are derived directly as a function of the Vs voltage at the source node as Vs is varied from 0 to Vcc. This voltage is entered as a Vgs value as a consequence of the relationship Vtable = Vgs = Vcc - Vs. Vds is held constant by having a fixed voltage Vds between the drain and source nodes. Note, Vds > 0 V. The current flowing into the drain is tabulated in the table for the corresponding Vs points.

```
     +------------------------------------------+
     |                                          |
     |                   Ids = Table Current    |
     |                   --->                   |
     +---<---|     |--->----------+             |
         d   |_____| - s          | +           |
               --+-- Vgs      +---+---+  +----+----+
                 | g  +       | Sweep |  |   Vs +  |
                              |   Vs  |  |Fixed Vds|
                              +---+---+  +----+----+
                                  | -         |
                                 GND         GND
```

Example of Series MOSFET Table Extraction

It is expected that this data will be created from semiconductor vendor proprietary silicon models, and later correlated with actual component measurement.

==
==