**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD ID#:** 157

**ISSUE TITLE:** *Parameterize [Driver Schedule]*

**REQUESTER:**  *Arpad Muranyi, Mentor Graphics; Romi Mayder, Xilinx*

**DATE SUBMITTED:** *January 24, 2013*

**DATE REVISED:**

**DATE ACCEPTED BY IBIS OPEN FORUM: Rejected August 21, 2015**

**STATEMENT OF THE ISSUE:**

*The delay values of the [Driver Schedule] keyword are fixed numeric entries in the current IBIS specification. There are situations when a given [Model] might be used at different bit rates, each of which might require different delay values for its [Driver Schedule] keyword. In order to eliminate the need to duplicate the entire [Model], it would be desirable to parameterize the delay entries of [Driver Schedule], so that the EDA tool could pass the appropriate delay values to [Driver Schedule] according to the bit rate the user selected for a simulation.*

**STATEMENT OF THE RESOLVED SPECIFICATIONS:**

*A new subparameter called Parameters shall be added to the [Driver Schedule] keyword to provide a mechanism to declare parameters used as arguments for its four delay parameters (Rise\_on\_dly, Rise\_off\_dly, Fall\_on\_dly and Fall\_off\_dly). The Parameters of the [Driver Schedule] keyword shall be initialized with a constant numeric literal value and/or a reference to a parameter name which is located in a parameter tree in a file. This reference shall begin with the file name, followed by a set of open and close parentheses enclosing the tree root name, any branch names if present and the parameter name. The file reference may point to the .ibs file itself where the reference is made from, or any other file containing one or more parameter trees. External parameter files shall contain only parameter trees and nothing else, and shall be located in the same directory as the .ibs file.*

*When the constant numeric literal and the root name of a parameter tree are both present in such an assignment, the constant numeric literal value shall serve as a default value for the assignment in case the assignment using the reserved word fails for some reason. The file names of any parameter file must follow the rules for file names given in Section 3, GENERAL SYNTAX RULES AND GUIDELINES.*

*Multiple [Driver Schedule] Parameters may be listed on the same line with one assignment, in which case all of the parameters on that line will be assigned the same value.*

*To implement this concept, the IBIS specification shall be changed as outlined below.*

**ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:**

*{TBD}*

**ANY OTHER BACKGROUND INFORMATION:**

*This BIRD depends on the acceptance of BIRD 153 entitled* [*Parameter Tree Keyword*](http://www.eda.org/pub/ibis/birds/bird153.txt) *and is similar in nature to BIRD 117 and 118. The editorial committee should pay special attention to keep them consistent.*

*Keyword:* [Driver Schedule]

*Required:* No

*Sub-Params:* Parameters

*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) shall 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. All of the remaining keywords not shown in the above list, and all of the subparameters will be used from the top-level model and should 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 exist in the .ibs file. The remaining four columns describe delays: Rise\_on\_dly, Rise\_off\_dly, Fall\_on\_dly, and Fall\_off\_dly. These arguments may contain numeric constant literals or parameter names defined in the Parameters subparameter. Any or all of these entries may be defined by parameter names, which must be declared and initialized by one or more Parameters subparameter. 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).

In the above four paragraphs, the word “event” refers to the moment in time when the delay is triggered by the stimulus. This stimulus is provided to the top-level model by the simulation tool. The expiration of delays cannot generate events.

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

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

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.

Only certain numerical entry combinations are permitted to define a complete transition sequence. Table 2 gives the initial scheduled model states for each permitted set of numerical entries. The numerical delay entries, r, r1, and r2 are relative to the internal simulator pulse rising edge, and f, f1, and f2 are the numerical delay entries relative to internal simulator pulse falling edge. For the cases where two delays are given relative to the same edge, the r2 entry is larger than the r1 entry, and the f2 entry is larger than the f1 entry. For cases below, the interchanging of such values corresponds to opposite direction switching. Once the scheduled model is set to its initial state, the switching is controlled by the internal simulator pulse and delays relative to it.

In Table 2, the scheduled model initial states depend on the initial state of the [Model]. This top-level [Model] state (“Low” or “High”) is a function of the stimulus pulse (or simulation control method) and the [Model] Polarity subparameter. For example, if a [Model] Polarity is Inverting and its stimulus pulse starts high, the [Model] initial state is “Low” and all scheduled model initial states follow the settings under the “Low” column. Two possible four-data ordering combinations are omitted because their initial states are ambiguous. Special rules to select the initial states would produce sequencing equivalent to the two-data combinations shown in the first two lines of the table.

Table – Scheduled Model Initial State

| **Table Numerical Delay Entries** | **[Model] Initial State** |
| --- | --- |
| **Rise\_on** | **Rise\_off** | **Fall\_on** | **Fall\_off** | **Low** | **High** |
| R | NA | f | NA | Low | High |
| NA | R | NA | f | High | Low |
| r1 | r2 | NA | NA | Low | Low |
| r2 | r1 | NA | NA | High | High |
| NA | NA | f1 | f2 | High | High |
| NA | NA | f2 | f1 | Low | Low |
| r1 | r2 | f2 | f1 | Low | Low |
| r2 | r1 | f1 | f2 | High | High |

The delay numbers r, r1, r2, and f, f1, f2 plus the associated model transitions should fit within the corresponding pulse width durations. Smaller pulse width stimuli may change the switching sequencing and is not supported.

Subparameter Definitions:

Parameters:

This optional subparameter lists and initializes parameter names to be used as arguments for the delay arguments of the [Driver Schedule] keyword under which it appears. The list of Parameters may span several lines by using the word Parameters at the start of each line. Any delay argument which is entered as a parameter must be declared and initialized with the Parameters subparameter.

Parameters are locally scoped under each [Driver Schedule] keyword, i. e. the same parameter under two different [Driver Schedule]s will have independent values.

The Parameters subparameter may contain one or more parameter names, which must be followed by an equal sign and a constant numeric literal and/or a reference to a parameter name which is located in a parameter tree. The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses. The file reference may point to the .ibs file itself where the reference is made from, or any other file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter files must follow the rules for file names given in Section 3, GENERAL SYNTAX RULES AND GUIDELINES. External parameter files may only contain parameter trees using the tree syntax described in the IBIS specification.

When a parameter reference and a constant numeric literal are both present in an assignment, they must be separated by at least one white space. In this case, the EDA tool should attempt to make the assignment using parameter reference first. If that fails (for example if the file doesn't exist) the constant numeric literal shall be used for the assignment. When multiple converter parameters are listed on a single line with one assignment, all of the parameters on that line shall be assigned the same value by the EDA tool.

The EDA tool may provide additional means to the user to make assignments to Parameters. This may include the option to override the values provided in the .ibs file, or to allow the user to make selections for multi-valued parameters in the parameter tree.

*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.

*Examples:*

[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

[Driver Schedule]

|

| List of parameters

Parameters RiseOnDly FallOnDly = thisfile.ibs(TreeRootName(Drf\_on)) 0.0

|

| Model\_name Rise\_on\_dly Rise\_off\_dly Fall\_on\_dly Fall\_off\_dly

 MODEL\_OUT RiseOnDly NA FallOnDly NA

|