Hi, All -

I have reviewed Tom's Notes-proposal (Mantis 658) and the majority of the proposed changes look good.

I do have some friendly amendments to offer to this proposal in the attached file.

Regards - Cliff
This file: Notes_1364_CliffReview.pdf

----------

## Cliff-notes on Tom's 1364-Notes Proposals

4.9
NOTE—Array size does not affect the element size..
No change required

No change required *to text but delete extra "." at the end of the sentence (Friendly amendment)*

----------

5.1.3 - I do not understand what the second "Deleted:" box is referring to?

----------

5.4.3
NOTE—Multiplication without losing any overflow bits is still possible simply by assigning the result to something wide enough to hold it.
No change required

Should be 5.4.1 *and I would like this to be a normative paragraph -*

WAS: NOTE—Multiplication without losing any overflow bits is still possible simply by assigning the result to something wide enough to hold it.

PROPOSED: It is possible to perform multiplication without losing any overflow bits by simply assigning the result to something wide enough to hold it.

----------

11.2
Change to:
The Verilog HDL is defined in terms of a discrete event execution model. The discrete event simulation is described in more detail in this section to provide a context to describe the meaning and valid interpretation of Verilog HDL constructs. These resulting definitions provide the standard Verilog reference model for simulation, which all compliant simulators shall implement. However, there is a great deal of choice in the definitions that follow, and differences in some details of execution are to be expected between different simulators. In addition, Verilog HDL simulators are free to use different algorithms than those described in this section, provided the user-visible effect is consistent with the reference model.

*Friendly amendment - change punctuation to remove fractured sentence*
These resulting definitions provide the standard Verilog reference model for simulation, which all compliant simulators shall implement; however, there is a great deal of choice in the definitions that follow, and differences in some details of execution are to be expected between different simulators.

*For grammatical details, see:*

*Semicolons with Clauses*
*http://englishplus.com/grammar/00000094.htm*

*Subordinate Conjunctions:*
*http://web.cn.edu/kwheeler/grammar_subordinate.html*

----------

12.2.2.1
This is already an informative example so there really is no need to change the text, but I am not opposed to the proposed change.

----------

12.4
Change "note" to "recognize"

----------

12.4.1 - Example 1 *(and globally true for all examples)*
This is already an informative example so there really is no need to change the text, but I am not opposed to the proposed change.

----------

12.8.1 (last par) - I disagree with the proposal. I believe this should be normative (see below)

WAS:

*2) The hierarchy below each starting point is expanded as much as possible without elaborating generate constructs. All parameters encountered during this expansion are given their final values by applying initial values, parameter overrides and **defparam** statements.*

This means that any **defparam** statement whose target can be resolved within the hierarchy elaborated so far must have its target resolved and its value applied. **defparam** statements whose target cannot be resolved are deferred until the next iteration of this step.

*3) Each generate construct encountered in step 2 is revisited, and the generate scheme is evaluated. The resulting generate block instantiations make up the new list of starting points. If the new list of starting points is not empty, go to step 2.*

~~Note that no **defparam** inside the hierarchy below a generate construct is allowed to refer to a parameter outside the generate construct. Therefore it is possible for parameters to get their final values in step 2.~~

PROPOSED:

*2) The hierarchy below each starting point is expanded as much as possible without elaborating generate constructs. All parameters encountered during this expansion are given their final values by applying initial values, parameter overrides and **defparam** statements.*

This means that any **defparam** statement whose target can be resolved within the hierarchy elaborated so far must have its target resolved and its value applied. **defparam** statements whose target cannot be resolved are deferred until the next iteration of this step. Since no **defparam** inside the hierarchy below a generate construct is allowed to refer to a parameter outside the generate construct, it is possible for parameters to get their final values before going to step 3.

*3) Each generate construct encountered in step 2 is revisited, and the generate scheme is evaluated. The resulting generate block instantiations make up the new list of starting points. If the new list of starting points is not empty, go to step 2.*

----------
17.5 - I disagree with the proposal. I believe this should be normative (see below)

WAS: ~~NOTE—~~The input terms can be nets or variables whereas the output terms shall only be variables.
PROPOSED: The input terms can be nets or variables whereas the output terms shall only be variables.

----------
18.2.4
This is just printed text in an example - do not change.

----------