Merged LRM - Draft 2 - Chapter 10 Cliff-Notes

Table 10-6
Do we need to add anything about fields of structs?

Nit: Last example code before 10.2 all of the previous assignments were consistent (a, r, #1) - be consistent with the added example?

```
assign #1 r = a;
```

Page 167 - paragraph before Example 1: (Problem - description added some cases but not all cases - I think we would do well to define either "DRIVER" or "DRIVING SOURCE" and we can reference and keep the driving-source list up to date)

WAS: Nets can be driven by multiple continuous assignments or by a mixture of primitive outputs, module outputs, and continuous assignments. Variables can only be driven by one continuous assignment or by one primitive output or module output,. It shall be an error for a variable driven by a continuous assignment or ~~primitive~~ output to have an initializer in the declaration or any procedural assignment. See also 6.3.

PROPOSED: Nets can be driven by a mixture of one or more continuous assignments and driving sources. Variables can only be driven by one continuous assignment or by one driving source. It shall be an error for a variable driven by a continuous assignment or driving source to have an initializer in the declaration or any procedural assignment. See also 6.3.

PROPOSED: 10.2.2.1 Driving Source
A driving source drives a value onto a net that resolves all driving sources to the appropriate value and strength of all of the combined driving sources. Driving sources are defined as:

- module outputs and inouts.
- program outputs and inouts.
- primitive outputs.
- clocking block outputs and inouts.
- interface outputs and inouts.

Page 170 - I think too much was cut out of this list, unless "singular types" sufficiently covers all the proposed cases.

WAS:~~— **reg, integer, real, realtime,** or **time** data type: an assignment to the name reference of one of these data types.~~
~~— Bit-select of a **reg, integer,** or **time** data type: an assignment to a single bit that leaves the other bits untouched.~~
~~— Part-select of a **reg, integer,** or **time** data type: a part-select of one or more contiguous bits that leaves the rest of the bits untouched.~~
~~— Memory word: a single word of a memory.~~

~~— Concatenation or nested concatenation of any of the above: a concatenation or nested concatenation of any of the previous four forms. Such specification effectively partitions the result of the righthand expression and assigns the partition parts, in order, to the various parts of the concatenation or nested concatenation.~~
— Singular variables, as described in 6.5
— Aggregate variables, as described in Clause 7
— Bit-selects, part-selects and slices of packed arrays
— Slices of unpacked arrays

PROPOSED
— Singular variables, as described in 6.2.
— Bit-select of singular variables.
— Part-select of singular variables.
— Legal concatenations or nested concatenations of one or more singular variables. Such specification effectively partitions the result of the righthand expression and assigns the partition parts, in order, to the various parts of the concatenation or nested concatenation.
— Aggregate variables, as described in Clause 7.
— Bit-selects, part-selects and slices of packed arrays.
— Slices of unpacked arrays.

Page 170 - I believe Stu correctly added Procedural assignment operators, but I think this list is still incomplete.

WAS: SystemVerilog contains three types of procedural assignment statements:
— Blocking procedural assignment statements (see 10.3.1)
— Nonblocking procedural assignment statements (see 10.3.2)
— Procedural assignment operators (see 11.2.7)

Blocking and nonblocking procedural assignment statements specify different procedural flows in sequential
blocks.

PROPOSED: SystemVerilog contains five types of procedural assignment statements:
— Blocking procedural assignment statements (see 10.3.1) including procedural assignment operators (see 11.2.7) and increment and decrement operators (see 11.2.8).
— Nonblocking procedural assignment statements (see 10.3.2)
~~— Procedural assignment operators (see 11.2.7)~~
— Clocking block synchronous drives (see 14.14)

Blocking procedural assignment statements, nonblocking procedural assignment statements and clocking block synchronous drives specify different procedural flows in sequential blocks.

According to Annex C:

This Annex identifies constructs that either:

- Have been deprecated from SystemVerilog and no longer appear in this standard
- Are under consideration for deprecation and might be removed from future versions of this standard

Cliff-Note: Procedural assign and deassign should either be removed from the IEEE Std 1800-2008 Standard or clearly identified in clause 10.5.1 that the construct has been marked for future removal. One of the two following paragraphs should be placed at the beginning of clause 10.5.1.

PROPOSED #1 (Cliff's preference) - Remove this entire clause and replace it with the following normative paragraph (saves answering Stu's questions in this clause - it is time to quit trying to document how these constructs work with new SystemVerilog constructs):

Procedural assign and deassign statements were deprecated in the IEEE Std 1800-2005 and were removed from the IEEE Std 1800-2008 (see C.2.2).

PROPOSED #2 (Cliff's second choice) - add the following paragraph as the first paragraph in clause 10.5.1 (and remove the last paragraph of 10.5.1 - deprecation should be noted first, not last, in a clause).

Procedural assign and deassign statements are under consideration for deprecation (see C.2.2). The description in this clause describes the semantics of the procedural assign and deassign statements prior to their selection for deprecation.

10.5.2 - Force and release procedural statements

Stu asks some good questions (see below). We have already made grouping definitions such as singular and aggregate. Do we need additional grouping definitions that can include singular expressions including bit-select, part-select, etc., so that they can be defined and referenced from one place? This could also help with one of the lists show on page 170 (proposal above). Cliff might need to make a proposal to identify this new grouping-definition.

Stu's questions:

- Does the result of a continuous assignment to a variable update immediately when the variable is released?
- What about unpacked stucts, enums, classes, etc.?

The rest looks good to me.
Regards - Cliff