SV-AC Feedback from Review of D4 Implementation

NOTE: All LRM references are to `P1800-2008-draft4-PROTECTED.pdf`.

FEEDBACK ON 1361 [MK]

19.1, Under Assertion Action control tasks, REPLACE

$assertnonvacuousoff

WITH

~~$assertnonvacuousoff~~ $assertvacuousoff

FEEDBACK ON 1460 [DK]

16.14.2, REPLACE

> When the property for the **`assume`** statement is evaluated to be true, the pass statements of the *action_block* are executed. If it evaluates to false, the fail statements of the action block are executed.

WITH

> When the property for the **`assume`** statement is evaluated to be true, the pass statements of the *action_block* are executed. If it evaluates to false, the fail statements of the ~~action block~~ *action_block* are executed.

RATIONALE: The italicized *action_block* with underscore was specified in the proposal and is used at a number of places in 16.14.

FEEDBACK ON 1674 [EC]

16.14.6, REPLACE

```
a4: assert property(p_multiclock(negedge clk2, ,posedge clk1, a, b, c, d);
```

WITH

```
a4: assert property(p_multiclock(negedge clk2, , posedge clk1, a, b, c, d);
```

The change is to add a space after the second comma.

16.14.6, REPLACE

```
a4: assert property(p_multiclock(negedge clk2, ,posedge clk1, a, b, c, d);
```

WITH

```
a4: assert property(p_multiclock(negedge clk2, , posedge clk1, a, b, c, d);
```

REMARK: This change is in the first version of module m. The change is to add a space after the second comma.

16.14.6, REPLACE

```
always @(posedge clk2 or posedge rst) begin
   if (rst) ... ;
   else if (d)
end
```

WITH

```
always @(posedge clk2 or posedge rst) begin
   if (rst) ... ;
   else if (d);
end
```

RATIONALE: The semicolon is needed for legal syntax. This was an oversight in the proposal.

16.14.6, REPLACE

Assertion `a2` uses explicit reset value `'0` in which case the **`disable iff`** statement could be omitted altogether in the equivalent assertion.

WITH

Assertion `a2` uses explicit reset value ~~`'0`~~ `1'b0` in which case the **`disable iff`** statement could be omitted altogether in the equivalent assertion.

RATIONALE: The assertion specifies `1'b0` rather than `'0`. This was an oversight in the proposal.

16.14.6, REPLACE

The inferred enabling condition for assertion `a3` is `(!bit'(rst!=1'b0) && d)`

WITH

The inferred enabling condition for assertion `a3` is `(!`~~`bit`~~**`bit`**`'(rst!=1'b0) && d)`

RATIONALE: The keyword “bit” should be in bold. This was an oversight in the proposal.

FEEDBACK ON 1768 [LP]

16.14.3, Editor Question:

This code looks like a mix of BNF and example. Should it be all one or the other?

RESPONSE: SV-AC discussed this in our meeting on 2007-10-09 and decided that the existing text is appropriate. The purpose of this text is to describe a general “assert property” that is related to a general “cover sequence” so that the total number of times matched for the general “cover sequence” can be described. We used the style of a RHS in a BNF production in order to achieve the generality of the forms. This passage is not discussing syntax per se, though, so rendering it as a BNF production is not appropriate.

FEEDBACK ON 1734 [JH]

In F.3.3.1, REPLACE in the third item under "Disabling of top-level properties"

$w^{0,i-1}\perp^{\omega}. \not\models P$

WITH

$w^{0,i-1}\perp^{\omega}. \not\models P$

REMARK: There is a stray period that needs to be deleted.

In F.3.6.1, CHANGE

Disabling of top-level properties is defined as follows:

— For $T = P$, $w \not\models^{\mathbf{d}} T$.

— For $U = Q$, $w \not\models^{\mathbf{d}} Q$.

TO

Disabling of top-level properties is defined as follows:

— For $T = P$, $w, L_0 \not\models^{\mathbf{d}} T$.

— For $U = Q$, $w, L_0 \not\models^{\mathbf{d}} Q$.