Motivation

Currently in the LRM the formal semantics of concurrent assertions is defined in terms of a single execution trace (word). However this approach allows defining semantics of a single assertion only, and is not applicable for describing the whole set of assertions and assumptions. If the definition of the formal semantics for assertions and assumptions in SystemVerilog is rather straightforward, the situation with the **`assert cover`** statement is different since the formal definition of coverage may be done in different ways. Unfortunately the informal definition of coverage in the LRM body does not help understand the formal semantics for coverage.

Consider the following *cover* statement:

```
assert cover(@clk a |=> b);
```

In case when `a` never happens in simulation vacuous coverage will be reported, but what a formal verification tool should report?

In this proposal it is suggested to treat **`assert cover`**`(`T`)` as a CTL directive **EF**T. We also provide a formal definition of assumptions since they are used in the definition of **`assert cover`** and a formal definition of assertions to make the definition of assertion statements complete.

F.3.3.1 Neutral satisfaction

REPLACE

Neutral satisfaction of assertions is as follows:

For the definition of neutral satisfaction of assertions, b denotes the boolean expression representing the enabling condition for the assertion. Intuitively, b is derived from the conditions in the context of a procedural assertion, while b is "1" for a declarative assertion.

— $w, b \models$ **`always`** `@(`c`)` **`assert property`** T iff for every $0 \leq i < |w|$ so that $\overline{w}^i \models c$ and $\overline{w}^i \models b$, either $w^{i..} \models$ `@(`c`)` T or $w^{i..} \models^{\mathrm{d}}$ `@(`c`)` T.

— $w, b \models$ **`always assert property`** U iff for every $0 \leq i < |w|$, if $\overline{w}^i \models b$ then either $w^{i..} \models U$ or $w^{i..} \models^{\mathrm{d}} U$.

— $w, b \models$ **`initial`** `@(`c`)` **`assert property`** T iff for every $0 \leq i < |w|$ so that $\overline{w}^{0,i} \models$`!`$c$ `[*0:$] ##1` c and $\overline{w}^i \models b$, either $w^{i..} \models$ `@(`c`)` T or $w^{i..} \models^{\mathrm{d}}$ `@(`c`)` T.

— $w, b \models$ **`initial assert property`** U iff (if $\overline{w}^0 \models b$ then either $w \models U$ or $w \models^{\mathrm{d}} U$).

Neutral satisfaction of top-level properties is defined as follows:

— For $T = P$ iff, $w \models T$ iff $w \models P$.

— For $U = Q$ iff, $w \models U$ iff $w \models Q$.

— For $T =$**`disable iff`**`(`b`)` $P, w \models T$ iff either

  — $w \models P$ and no letter of w satisfies b, or

  — Some letter of w satisfies b and $w^{0,i1}\bot^{\omega} \models P$ for i the least index such that $w^i \models b$, $0 \leq i < |w|$.

— For $U =$**`disable iff`**`(`b`)` $Q, w \models U$ iff either

  — $w \models Q$ and no letter of w satisfies b, or

— Some letter of w satisfies b and $w^{0,i1}\bot^{\omega} \models Q$ for i the least index such that $w^i \models b$, $0 \leq i < |w|$.

T is said to *pass* on w if $w \models T$. T is said to be *disabled* on w if $w \models^{\mathrm{d}} T$. T is said to *fail* on w if T neither passes nor is disabled on w. It can be proved that T cannot both pass and be disabled on w.

WITH

Neutral satisfaction of ~~assertions~~ assertion statements is as follows:

For the definition of neutral satisfaction of ~~assertions~~ assertion statements, b denotes the boolean expression representing the enabling condition for the ~~assertion~~ assertion statement. Intuitively, b is derived from the conditions in the context of a procedural ~~assertion~~ assertion statement, while b is ~~"~~1~~"~~ for a declarative assertion.

— $w, b \models$ **`always`** `@(`c`)` **`assert property`** T iff for every $0 \leq i < |w|$ so that $\overline{w}^i \models c$ and $\overline{w}^i \models b$, either $w^{i..} \models$ `@(`c`)` T or $w^{i..} \models^{\mathrm{d}}$ `@(`c`)` T.

— $w, b \models$ **`always assert property`** U iff for every $0 \leq i < |w|$, if $\overline{w}^i \models b$ then either $w^{i..} \models U$ or $w^{i..} \models^{\mathrm{d}} U$.

— $w, b \models$ **`initial`** `@(`c`)` **`assert property`** T iff for every $0 \leq i < |w|$ so that $\overline{w}^{0,i} \models$`!`$c$ `[*0:$] ##1` c and $\overline{w}^i \models b$, either $w^{i..} \models$ `@(`c`)` T or $w^{i..} \models^{\mathrm{d}}$ `@(`c`)` T.

— $w, b \models$ **`initial assert property`** U iff (if $\overline{w}^0 \models b$ then either $w \models U$ or $w \models^{\mathrm{d}} U$).

— $w, b \models$ **`always`** `@(`c`)` **`assume property`** T iff $w, b \models$ **`always`** `@(`c`)` **`assert property`** T.

— $w, b \models$ **`always assume property`** U iff $w, b \models$ **`always assert property`** U.

— $w, b \models$ **`initial`** `@(`c`)` **`assume property`** T iff $w, b \models$ **`initial`** `@(`c`)` **`assert property`** T.

— $w, b \models$ **`initial assume property`** U iff $w, b \models$ **`initial assert property`** U.

— $w, b \models$ **`always`** `@(`c`)` **`cover property`** T iff there exists $0 \leq i < |w|$ so that $\overline{w}^i \models c$, $\overline{w}^i \models b$, and $w^{i..} \models$ `@(`c`)` T.

— $w, b \models$ **`always cover property`** U iff there exists $0 \leq i < |w|$ so that $\overline{w}^i \models b$ and $w^{i..} \models U$.

— $w, b \models$ **`initial`** `@(`c`)` **`cover property`** T iff there exists $0 \leq i < |w|$ so that $\overline{w}^{0,i} \models$`!`$c$ `[*0:$] ##1` c, $\overline{w}^i \models b$, and $w^{i..} \models$ `@(`c`)` T.

— $w, b \models$ **`initial cover property`** U iff $\overline{w}^0 \models b$ and $w \models U$.

The neutral satisfaction of assertion statements defined above describes the behavior of an assertion statement on a single trace. The following definitions describe assertion statement satisfaction on a set of traces:

— **`assert property`** statement is *satisfied on a set of traces* if it is satisfied on each trace of the set which satisfies all the assumptions associated with the model.

— **`assume property`** statement is *satisfied on a set of traces* if it is satisfied on each trace of the set.

— **`cover property`** statement is *satisfied on a set of traces* if it is satisfied at least on one trace of the set satisfying all the assumptions associated with the model.

An assertion statement *holds globally* if it is satisfied on the set of all feasible traces.