Motivation

Currently in the LRM the formal semantics of concurrent assertions is defined in terms of a single execution path (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 verification statements complete.

REPLACE

F.3 Semantics

WITH

F.3 Path semantics

This subclause describes the path semantics of assertions: whether assertions are satisfied in a given valuation path. The subclause F.4 describes the global semantics of verification statements.

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 $w^{i\cdot\cdot} \models$ `@` (c) T or $w^{i\cdot\cdot} \models^{\mathrm{d}}$ `@` (c) T.

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

— $w, b \models$ **`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 $w^{i\cdot\cdot} \models$ `@` (c) T or $w^{i\cdot\cdot} \models^{\mathrm{d}}$ `@` (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~~ verification statements is as follows:

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

Rewriting of the top-level properties in the context of assertions and assumptions is defined as follows:

— For $T = P$ or for $T = Q$, $T^a = T$.

— For $T =$**disable iff** (b) P, $T^a =$**accept_on** (b) P.

— For $T =$**disable iff** (b) Q, $T^a =$**accept_on** (b) Q.

Rewriting of the top-level properties in the context of cover statements is defined as follows:

— For $T = P$, $T^c = P$.

— For $T = Q$, $T^c = Q$.

— For $T =$**disable iff** (b) P, $T^c =$**reject_on** (b) P.

— For $T =$**disable iff** (b) Q, $T^c =$**reject_on** (b) Q.

— ~~$w, b \models$ **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 assertions is as follows:

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

— $w, b \models$ **always assert property** T iff $w, b \models$ **always** T^a.

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

— $w, b \models$ **initial assert property** U iff $w, b \models T^a$.

Neutral satisfaction of assumptions is defined as neutral satisfaction of assertions by replacing of **assert property** with **assume property**.

Neutral satisfaction of **cover property** is as follows:

— $w, b \models$ **always @(c) cover property** T iff $w, b \models$@c **s_eventually** T^c.

— $w, b \models$ **always cover property** T iff $w, b \models$ **s_eventually** T^c.

— $w, b \models$ **initial @(c) cover property** T iff $w, b \models$@c T^c.

— $w, b \models$ **initial cover property** U iff $w, b \models T^c$.

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

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

— ~~For $U = Q, w \models U$ 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 $w^{0,i1} \perp^{\omega} \models P$ 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 $w^{0,i1} \perp^{\omega} \models Q$ for i the least index such that $w^{i} \models b$, $0 \leq i < |w|$.~~

In **assert property** or **assume property** context $w \models T$ iff $w \models T^a$.

In **cover property** context $w \models T$ iff $w \models T^c$.

REPLACE

Disabling of top-level properties is defined as follows:

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

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

— For T =**disable iff** (b) $P, w, d \models^{\mathrm{d}} T$ iff some letter of w satisfies b and both $w^{0,i-1}\top^{\omega} \models P$ and $w^{0,i-1}\bot^{\omega} \not\models P$ for i the least index such that $w^i \models b, 0 \leq i < |w|$.

— For U =**disable iff** (b) $Q, w \models^{\mathrm{d}} U$ iff some letter of w satisfies b and both $w^{0,i-1}\top^{\omega} \models Q$ and $w^{0,i-1}\bot^{\omega} \not\models Q$ for i the least index such that $w^i \models b, 0 \leq i < |w|$.

WITH

Disabling of top-level properties in **assert property** or **assume property** context is defined as follows:

— For $T = P$ or $T = Q$, $w \not\models^{\mathrm{d}} T$

— ~~For $U = Q$, $w \not\models^{\mathrm{d}} Q$~~

— For T =**disable iff** (b) $P, w, d \models^{\mathrm{d}} T$ iff some letter of w satisfies b and both $w^{0,i-1}\top^{\omega} \models P$ and $w^{0,i-1}\bot^{\omega} \not\models P$ for i the least index such that $w^i \models b, 0 \leq i < |w|$.

— For ~~U~~T =**disable iff** (b) $Q, w \models^{\mathrm{d}} T$~~$U$~~ iff some letter of w satisfies b and both $w^{0,i-1}\top^{\omega} \models Q$ and $w^{0,i-1}\bot^{\omega} \not\models Q$ for i the least index such that $w^i \models b, 0 \leq i < |w|$.

Disabling of top-level properties in **cover property** context is defined as follows:

— $w \not\models^{\mathrm{d}} T$

ADD

F.4 Global semantics

Note to editor: Shift the numeration of the following subclauses accordingly.

This clause describes the formal semantics of the verification statements in terms of the sets of legal valuation paths of the system. This description is applicable for the formal verification context, where all possible sets of valuation paths are considered. The simulation tools deal with a single valuation path, and the verification statement semantics used in simulation is completely described in F.3.

Each variable v of type t with a value domain $t_1, \ldots, t_n$, may be coded by $\lceil \log n \rceil$ binary variables called *atomic propositions*. Let AP denote a set of all atomic propositions in the model, then any state of the model is unambiguously defined by a set of the atomic propositions having a value *true* at this state $s \subseteq AP$. The set of initial states of the model will be denoted as $I \subseteq 2^{AP}$ The model has a transition function describing which transitions between states are legal in the system: $\rho : 2^{AP} \rightarrow 2^{2^{AP}}$, i.e. for each state it defines the set of the states that the system can transition there at the next step.

Each model M may be represented by the set of all its *valuation paths* W_M, each valuation $w \in W_M$ being a series of states $s_0, s_1, \ldots$ starting with an initial state $s_0 \in I$, s.t. for all $i \geq 0$, $s_{i+1} \in \rho(s_i)$. A valuation path may be considered as an infinite word over the alphabet $\Sigma = 2^{AP}$.

Each assumption m with the enabling condition b is itself also a kind of a model and it defines a set of valuation paths satisfying the assumptions (see F.3.3.1): $W_m = \{w | w, b \models m\}$.

An assumption applied to the system defines a composition of the system with the assumption $M||m$, characterized by the set of valuation paths satisfying both the system and the assumption: $W_{M||m} = W_M \cap W_m$.

We will call *system* Π a composition of the model with all the assumptions $m_1, \ldots, m_n$ defined on it: $\Pi = M \mid\mid \|_{i=1}^{n} m_i$. The system is characterized by the set of valuation paths:

$$W_\Pi = W_M \cap \bigcap_{i=1}^{n} W_{m_i}$$

An assertion *is satisfied* on the system Π if it is satisfied for all system valuation paths $w \in W_\Pi$. A cover statement *is satisfied* on the system Π if it is satisfied for at least one system valuation path $w \in W_\Pi$. See the definition of assertion and cover statement satisfaction on a valuation path in F.3.