

Alignment Subcommittee Issues to Consider

##	Date	Section	Issue	Resolved	Who
1	030211		Clarify precedence of logical implication operators (-> and <->) - not in table 2, pg 15		CE
2	030212	4.4.2	Show 'high to low' in precedence table 2		BC
3	030217		Clarify whether (and if so, why we can use literals True, False in Verilog flavor		BC
4	030225	4.2.2	Add -> to table 2		BC
5	030304	B.3.2	Remove "releases" and "V" operators		DB
6	030327	Tbl 2	Fix precedence of [U], [W], to be same as until*		CE
7	030414		Clarify precedence of repetition operators (higher than ';')		CE
8	030514		Clarify that, for all suffix implication operators, and empty match on the LHS doesn't count		CE
9	030615	4.4.6	4.4.6 - finite/infinite - Is there an alternative semantics to the finite paths ? do we force any of the suggested semantics ? If we do, then it must be declared here.		GV
10	030826		1) What is the semantics of a signal value before the "starting point". For instance, can the following property hold? always prev(a)		JA
11	030826		In the formal semantics section of the LRM, weak suffix implication is defined as (p. 109) $\{r_1\} \mid \Rightarrow \{r_2\} = \text{def } \{r_1\} \mid \rightarrow \{T; r_2\}$ <p>This means that r_1 may not be the empty sequence. However, $\mid \Rightarrow$ could equally well (?) have been defined as</p> $\{r_1\} \mid \Rightarrow \{r_2\} = \text{def } \{r_1; T\} \mid \rightarrow \{r_2\}$ <p>which would have allowed the empty sequence. Now you have to write</p> $\{ \text{True} \} \mid \rightarrow \{ r \}$ <p>to get a sequence which starts at time point 0. Is there any specific reason why $\mid \Rightarrow$ is defined the way it is?</p>		JA

- | | | | |
|----|--------|---|------------------------|
| 12 | 030826 | <p>Multiply clocked SERE:s</p> <p>In the informal semantics of clocked sere (p. 33), note 1 on lines 28-30 seem to state that outer clocks are ignored in the case of an inner clock. This seems to contradict the formal semantics on page 104. If the informal semantics were correct, then</p> <p>$\{ a@clk_a \}@clk_b$</p> <p>would be equal to $\{ a@clk_a \}$, which seems inconsistent with</p> <p>$(a@clk_a)@clk_b$</p> <p>which is not equal to $a@clk_a$. My interpretation of this is that the note in the informal semantics only is applicable when the first event of the</p> | JA |
| 13 | 030826 | <p>the last point leads to that the note #1 on p.32, 6.1.1 is incorrect : <code>restrict_guarantee {SERE}</code> , when used as "guarantee" is a property which is built from a plain SERE on its own.</p> <p>If so, should we allow this too : <code>assert { !reset, !reset, reset[*] } ?</code></p> | GV |
| 14 | 030425 | <p>P33L28 Fix note on p33L28 to say that $\{ \{ a;b \}@clk1; c \}@clk2 \} \iff \{ \{ a;b \}@clk1 \}@clk2; c@clk2 \}$</p> | <p>JH</p> <p>Defer</p> |