

Section 17.11

Declaring Properties

p. 232

REPLACE

A **disable iff** clause can be attached to a *property_expr* to yield a *property_spec*

```
disable iff (expression_or_dist) property_expr
```

The expression of the **disable iff** is called the reset expression. The **disable iff** clause allows asynchronous resets to be specified. For an evaluation of the *property_spec*, there is an evaluation of the underlying *property_expr*. If prior to the completion of that evaluation the reset expression becomes true, then the overall evaluation of the *property_spec* is true. Otherwise, the evaluation of the *property_spec* is the same as that of the *property_expr*. The reset expression is tested independently for different evaluation attempts of the *property_spec*. Nesting of **disable iff** clauses, explicitly or through property instantiations, is not allowed.

WITH

A **disable iff** clause can be attached to a *property_expr* to yield a *property_spec*

```
disable iff (expression_or_dist) property_expr
```

The expression of the **disable iff** is called the reset expression. The **disable iff** clause allows **asynchronous pre-emptive** resets to be specified. For an evaluation of the *property_spec*, there is an evaluation of the underlying *property_expr*. If prior to the completion of that evaluation the reset expression becomes true, then the overall evaluation of the *property_spec* is true. Otherwise, the evaluation of the *property_spec* is the same as that of the *property_expr*. The reset expression is tested independently for different evaluation attempts of the *property_spec*. Nesting of **disable iff** clauses, explicitly or through property instantiations, is not allowed.

Section 17.12.3

Clock flow

p. 243

REPLACE

Intuitively, clock flow provides that in a multiply-clocked sequence or property the scope of a clocking event flows left-to-right across linear operators (e.g., repetition, concatenation, negation, implication) and distributes to the operands of branching operators (e.g., conjunction, disjunction, intersection, **if...else**) until it is replaced by a new clocking event.

WITH

Intuitively, clock flow provides that in a multiply-clocked sequence or property the scope of a clocking event flows left-to-right across linear operators (e.g., repetition, concatenation, negation, implication, **disable iff**) and distributes to the operands of branching operators (e.g., conjunction, disjunction, intersection, **if...else**) until it is replaced by a new clocking event.

Annex H

H.3.1 Rewrite rules for clocks

p. 537

REPLACE

- $@(c) R[*1:\$] \rightarrow (@(c) R)[*1:\$] .$
- $@(c) \text{disable iff } (b) \rightarrow P \text{disable iff } (b) @(c) P.$
- $@(c) \text{not } P \rightarrow \text{not } @(c) P.$

WITH

- $@(c) R[*1:\$] \rightarrow (@(c) R)[*1:\$] .$
- ~~$@(c) \text{disable iff } (b) \rightarrow P \text{disable iff } (b) @(c) P.$~~
- $@(c) \text{not } P \rightarrow \text{not } @(c) P.$