

# Objective

Proposal for 0003135: Verbal explanation of nexttime and always is misleading for multiple clocks.

The verbal explanations of the operators nexttime/s\_nexttime and their derived operators is correct when there is a single clock, but incorrect in the presence of multiple clocks. (See section 16.13.10, 16.13.11, for instance.) In the presence of multiple clocks, a formula can be evaluated at a cycle which is not a tick of the clock of the nexttime operator. In such a case, the formal semantics dictates, that nexttime will take you to the second tick of the clock of the nexttime operator. It is nexttime[0] that takes us to the first tick. In general nexttime[n] takes us to the n+1-th tick of the clock (where if the current cycle is a tick of the relevant clock then that counts as the first tick).

## Note to the editor:

The text under “AFTER” below is the text from the proposal for 2552. The text from proposal for 3134 should be added after the text for proposal 2552 and before the proposed addition below.

### 16.13.10 Nexttime property

#### AFTER

A property is a *nexttime* if it has one of the following forms, which use the nexttime operators:

- Weak nexttime:

**nexttime** *property\_expr*

The weak nexttime property **nexttime** *property\_expr* evaluates to true if, and only if, either the *property\_expr* evaluates to true beginning at the next clock tick or there is no further clock tick.

- Indexed form of weak nexttime:

**nexttime** [ *constant\_expression* ] *property\_expr*

The indexed weak nexttime property **nexttime** [*constant\_expression*] *property\_expr* evaluates to true if, and only if, either there are not *constant\_expression* clock ticks or *property\_expr* evaluates to true beginning at the last of the next *constant\_expression* clock ticks.

- Strong nexttime:

**s\_nexttime** *property\_expr*

The strong nexttime property **s\_nexttime** *property\_expr* evaluates to true if, and only if, there exists a next clock tick and *property\_expr* evaluates to true beginning at that clock tick.

- Indexed form of strong nexttime:

**s\_nexttime** [ *constant\_expression* ] *property\_expr*

The indexed strong nexttime property **s\_nexttime** [*constant\_expression*] *property\_expr* evaluates to true if, and only if, there exist *constant\_expression* clock ticks and *property\_expr* evaluates to true beginning at the last of the next *constant\_expression* clock ticks.

#### ADD

The above explanations refer to the case where the nexttime property is evaluated in a time step that is a tick of the clock of the nexttime property. When the nexttime property is evaluated in a time step that is not a tick of the clock of the nexttime property then an alignment to the tick of the clock of the nexttime property should be applied before the above description. Thus, it is more precise to say that `s_nexttime`<sub>[n]</sub> `property_expr` evaluates to true if, and only if, there exist  $n+1$  ticks of the clock of the nexttime property, including the current time step, and `property_expr` evaluates to true on the  $n+1$ 'th clock tick, where counting starts at the current time step. In particular `nexttime`[0] and `s_nexttime`[0] act as alignment operators.

### 16.13.11. Always property

#### AFTER

A property is an *always* if it has one of the following forms, which use the always operators:

- Weak always:

**always** `property_expr`

A property **always** `property_expr` evaluates to true if, and only if, `property_expr` holds at every current or future clock tick.

- Ranged form of weak always:

**always** [ `cycle_delay_const_range_expression` ] `property_expr`

A property **always** [ `cycle_delay_const_range_expression` ] `property_expr` evaluates to true if, and only if, `property_expr` holds at every current or future clock tick that is within the range of clock ticks specified by `cycle_delay_const_range_expression`. It is not required that all clock ticks within this range exist.

- Ranged form of strong always:

**s\_always** [ `constant_range` ] `property_expr`

A property **s\_always** [ `constant_range` ] `property_expr` evaluates to true if, and only if, all current or future clock ticks specified by `constant_range` exist and `property_expr` holds at each of these clock ticks.

#### ADD

The above explanations refer to the case where the always property is evaluated in a time step that is a tick of the clock of the always property. When the always property is evaluated in a time step that is not a tick of the clock of the always property then an alignment to the tick of the clock of the always property should be applied before the above description. Thus, it is more precise to say that `s_always`<sub>[n:m]</sub> `property_expr` evaluates to true if, and only if, there exists  $m+1$  ticks of the clock of the always property, including the current time step, and `property_expr` evaluates to true beginning in all of the  $n+1$ 'th to  $m+1$ 'th clocks ticks, where counting starts at the current time step.