

Constant_primary as delay in ##N

Modify Syntax 16-2 and 16-3, and A.2.10

from

```
cycle_delay_range ::=
    ## integral_number
    | ## identifier
    | ## ( constant_expression )
    | ## [ cycle_delay_const_range_expression ]
```

to

```
cycle_delay_range ::=
    ## integral_number
    | ## identifier
    | ## ( constant_expression )
    ## constant_primary
    | ## [ cycle_delay_const_range_expression ]
```

Modify 16.6

from

In this syntax, the following statements apply:

- *constant_expression* is computed at compile time and must result in an integer value.
- *constant_expression* can only be 0 or greater.
- The \$ token is used to indicate the end of simulation. For formal verification tools, \$ is used to indicate a finite, but unbounded, range.
- When a range is specified with two expressions, the second expression must be greater than or equal to the first expression.

to

In this syntax, the following statements apply:

- *constant_primary* includes *constant_expression* which ~~*constant_expression*~~ is computed at compile time and must result in an integer value. Furthermore, *constant_expression* and the bounds in *cycle_delay_const_range_expression* can only be 0 or greater.
- ~~— *constant_expression* can only be 0 or greater.~~
- The \$ token is used to indicate the end of simulation. For formal verification tools, \$ is used to indicate a finite, but unbounded, range.
- When a range is specified with two expressions, the second expression must be greater than or equal to the first expression.

- In a *cycle_delay_range*, it shall be illegal for a *constant_primary* to contain a *constant_mintypmax_expression* that is not also a *constant_expression*.