Section 16.8.3, page 321 (IEEE Std 1800™-2008 Draft 3a)

Replace on page 321

```
$sampled(expression [, clocking_event])
```

With

```
$sampled(expression [, clocking_event])
```

Replace on page 321

The use of these functions is not limited to assertion features; they can be used as expressions in procedural code as well. The clocking event, although optional as an explicit argument to the functions, is required for their semantics. The clocking event is used to sample the value of the argument expression.

With

The use of these functions is not limited to assertion features; they can be used as expressions in procedural code as well. The clocking event, although optional as an explicit argument to the functions `$past`, `$rose`, `$stable` and `$fell`, is required for their semantics. The clocking event is used to sample the value of the argument expression.
The function `$sampled` does not use a clocking event. The value of `$sampled` is updated in the preponed scheduling region in every simulation time step.

Replace on page 322

Function `$sampled` returns the sampled value of the expression with respect to the last occurrence of the clocking event. When `$sampled` is invoked prior to the occurrence of the first clocking event, the value of X is returned. The use of `$sampled` in assertions, although allowed, is redundant, as the result of the function is identical to the sampled value of the expression itself used in the assertion.

With

The `$sampled` function returns the value of the expression sampled in the preponed region of the simulation time step in which the function is called. The value is stable throughout the simulation step.

The value of an expression sampled in the preponed region corresponding to time 0 is the result of evaluating the expression using the initial values of the variables comprising the expression. The initial value of a static variable is the value assigned in its declaration, or in the absence of such an assignment it is the default (or uninitialized) value of the corresponding type. The initial value of any other variable or signal is the default value of the corresponding type. For example, if `$sampled(y)` is called at time 0, and `y` is of type logic, the value returned is `X`.

The use of `$sampled` in assertions, although allowed, is redundant, as the result of the function is identical to the sampled value of the expression itself used in the assertion.

Replace

When these functions are called at or before the first clock tick of the clocking event, the results are computed by comparing the current sampled value of the expression to X.

With

When these functions are called at or before the first clock tick of the clocking event, the results are computed by comparing the current sampled value of the expression to ~~X~~ the result of the expression evaluated using the initial values of the variables comprising the expression. The initial value of a static variable is the value assigned in its declaration, or in the absence of such an assignment it is the default (or uninitialized) value of the corresponding type. The initial value of any other variable or signal is the default value of the corresponding type.

Replace on page 323

A clock tick is based on clocking_event. If the specified clock tick in the past is before the start of simulation, the returned value from the `$past` function is a value of X.

With

A clock tick is based on clocking_event. ~~If the specified clock tick in the past is before the start of simulation, the returned value from the `$past` function is a value of X.~~
If the specified clock tick in the past is before the start of simulation, the returned value from the `$past` function is the result of evaluating the expression using the initial values of the variables comprising the expression. The initial value of a static variable is the value assigned in its declaration, or in the absence of such an assignment it is the default (or uninitialized) value of the corresponding type. The initial value of any other variable or signal is the default value of the corresponding type.

Replace on page 460, clause 19.11

```
$sampled(expression [, clocking_event])
```

With

```
$sampled(expression ~~[, clocking_event]~~)
```

Add to C.2 as C.2.2 (Note to the editor: Please adjust numbering as necessary.)

C.2.2 $sampled with a clocking event argument

The IEEE Std 1800-2005 required in 17.7.3 that an explicit or inferred clocking event argument be provided for the `$sampled` assertion system function. In this version of the standard, the semantics of `$sampled` have been changed to a form that does not depend on a clocking event. Therefore the syntax for defining the clocking event argument to `$sampled` is deprecated and does not appear in this version of the standard.