Proposal relative to p1800-2008 Draft 4

Motivation

The LRM indicates how the control tasks `$asserton`, `$assertoff` and `$assertkill` affect the behavior of verification statements that are placed in initial blocks. However, if the text is applied, then when `$assertoff` is issued at time 0 to disable assertions during the design reset period which may extend over 1 or more tick of the clocking events of the assertions, then assertions in the initial block would never run. Yet, in most such cases, it is desired to run them on the first clock tick after `$asserton` is issued.
The proposal tries to remedy this situation. In summary, if there is a call to `$assertoff` before any clocking event of the assertion occurs, then the initial assertion will not start until the first clock tick after a `$asserton` is issued. At that point it will execute its unique evaluation attempt.

In 16.14.5

REPLACE

If the statement appears in an **`always`** procedure, the property is always monitored. If the statement appears in an **`initial`** procedure, then the monitoring is performed only on the first clock tick.

WITH

If the statement appears in an **`always`** procedure, the property is always monitored. If the statement appears in an **`initial`** procedure, then the monitoring is performed only on the first clock tick following the instant when assertions are first enabled. This means that it monitors starting on the first clock tick at or after time 0, or if `$assertoff` or `$assertkill` is called before entering the Observed scheduling region that follows the first clocking event, then it will start monitoring on the first clock tick following the issuance of `$asserton`.