

Clause: 18.3

Description

The second paragraph of clause 18.3 states "The values of variables used in assertions are sampled in the Preponed region of a time slot and the assertions are evaluated during the Observe region. This is explained in Clause 15, Scheduling Semantics".

There is no mention of sampling based on clocking block if the signal used in the assertion is a clocking block signal. As per clause 15.3 "The sampling time of sampled data for property expressions is controlled in the clocking block". It is not very clear from these clauses how the sampling should be done for clocking block variables which are used in the concurrent assertion expression. Do we still continue and sample clocking block variables in the Preponed region irrespective of how the clocking block is sampling? Or are these variables are sampled based on the clocking block declaration ?

Suggested Resolution

The second paragraph of clause 18.3, replace: ~~The values of variables used in assertions are sampled in the Preponed region of a time slot and the assertions are evaluated during the Observe region. This is explained in Clause 15, Scheduling Semantics~~

By

The values of variables used in assertions are sampled in the Preponed region of a time slot and the assertions are evaluated during the Observe region. If a variable used in an assertion is a clocking block input variable, the variable must be sampled by the clocking block with #1step sampling. Any other type of sampling for the clocking block variable shall result in an error. The assertion using the clocking block variable shall not do its own sampling on the variable, but rather use the sampled value produced by the clocking block. This is explained in Clause 15, Scheduling Semantics.

Clause 15.3, remove ~~The sampling time of sampled data for property expressions is controlled in the clocking block~~

Add a new clause 18.17 to explain effect of clocking blocks on assertions.

18.17 Clocking blocks and Concurrent Assertions

If a variable used in a concurrent assertion is a clocking block variable it will be sampled only in the clocking block.

Examples:

```
module A;
  logic a, clk;

  clocking cb_with_input @(posedge clk);
    input a;

  property p1;
    a;
  endproperty
```

```

endclocking

clocking cb_without_input @(posedge clk);
    property p1;
        a;
    endproperty
endclocking

property p1;
    @(posedge clk) a;
endproperty

property p2;
    @(posedge clk) cb_with_input.a;
endproperty

a1: assert property (p1);
a2: assert property (cb_with_input.p1);
a3: assert property (p2);
a4: assert property (cb_without_input.p1);
endmodule

```

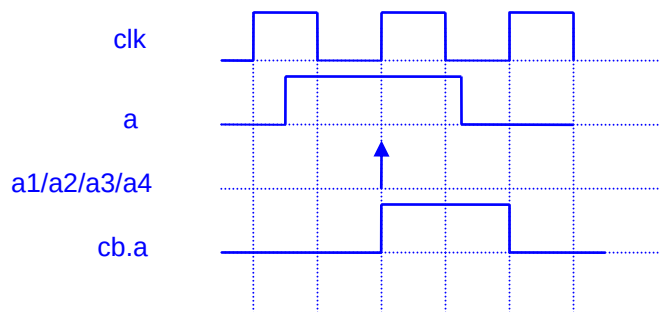


Figure 18-7 - Clocking blocks and Concurrent Assertion

Figure 18-17 explains the behavior of all the assertions. In the above example, a1, a2, a3 and a4 are equivalent.