sequence_expr **|->** sequence_expr **|->** property_expr
← ----- property ----------→
sequence_expr **|->** sequence_expr **|->** sequence_expr **|->** property_expr
← ----- property ----------→
← ----------property ----------→

The behavior of the implication operator is show in Table 3.9.1.

Table 3.9.1 Implication Operator

Antecedent sequence	Consequent property	Property Evaluation
False	True / false / vacuously true	True, Vacuous; pass count is not incremented
True	False	False; fail count is incremented
True	vacuously true	True, Vacuous; pass count is not incremented[36]
True	True nonvacuously	True; pass count is incremented

Note: Vacuity is related to properties only. A sequence always evaluates to nonvacuous true or false. Thus, an assertion with a sequence as the consequent either succeeds or fail if the antecedent is true.

3.9.1.1 Overlapped implication operator |->

🕮 **Rule:** The "|->"overlapped implication operator allows an *if – then* checking in a property. This operator specifies that if there is a match for the antecedent *sequence_expr*, the condition that follows the operator (i.e., the fulfilling condition or consequent) begins in the same cycle in which the end point of the antecedent becomes true. For example, the following property is demonstrated in Figure 3.9.1.

```
property p_test;    a ##1 b |-> c ##1 d;  endproperty : p_test
ap_test: assert property(@ (posedge clk) p_test);
// Same as above, but without a property declaration. The property expression is inline
ap_test_alternative: assert property(
      @ (posedge clk) a ##1 b |-> c ##1 d);
```

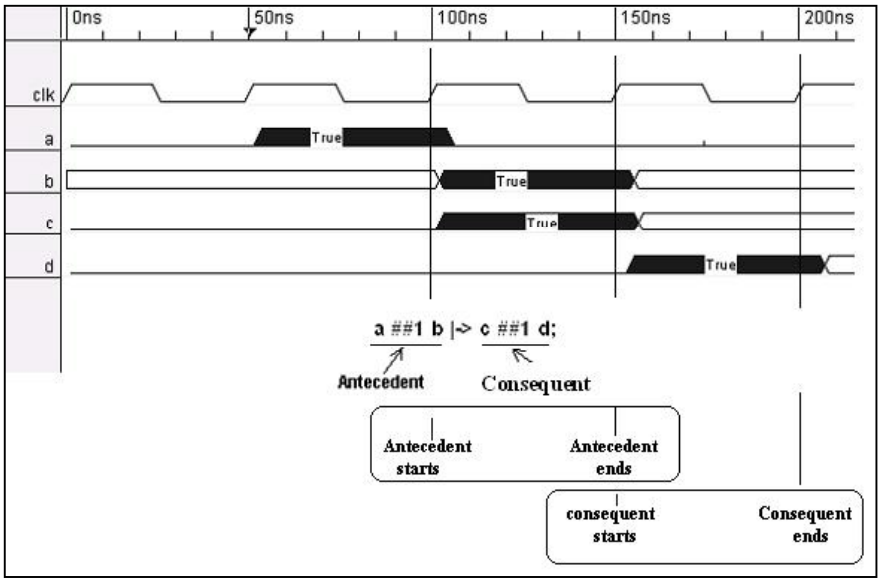

Figure 3.9.1 Timing diagram for property *test*

[36] See file ch3/vac.sv, wave_vac.bmp (waveform), vac.jpg (thread viewer), ac_assertion_report.txt