

# Sequential Distance Analysis: A Metric For Assertion Coverage

Abhishek Muchandikar  
Synopsys, Inc.  
[mabhi@synopsys.com](mailto:mabhi@synopsys.com)

Thomas Thatcher  
Oracle Corp.  
[thomas.thatcher@oracle.com](mailto:thomas.thatcher@oracle.com)

Xiaolin Chen  
Synopsys, Inc.  
[xiaolin@synopsys.com](mailto:xiaolin@synopsys.com)

Surrendra Dudani  
Synopsys, Inc.  
[dudani@synopsys.com](mailto:dudani@synopsys.com)



## Challenges With Assertion Coverage

- Identifying **logic missed** by a set of assertions
- Detecting **redundant** assertions
- Improving **ad-hoc** analysis to determine whether to add more assertions
- Capturing a comprehensive view of how assertions are **monitoring** the design

Coverage inadequate for assertions

## Solution: A Sequential Distance Metric

- A metric that **measures** sequential distance of elements and assertions
- Helps to **quantify and qualify** completeness of assertions
- Points out **redundant** assertions
- Provides insight on assertion **coverage holes**
- A cost effective **way to monitor** assertion verification progress

## Sequential Distance Analysis Benefit

Report assertion coverage holes

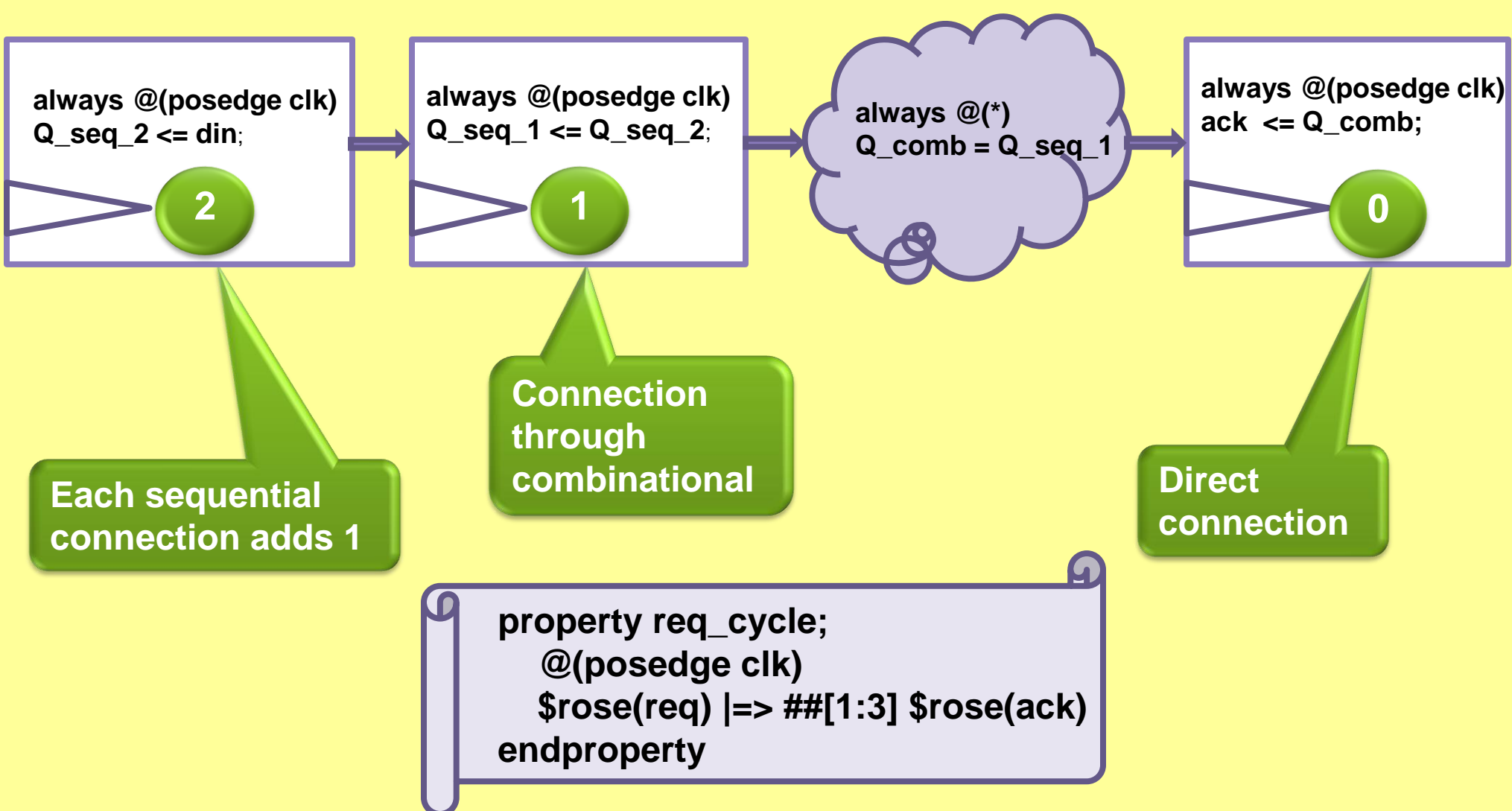
Identify redundant assertions

More effective assertions

Enhance simulation performance

## Definition Of Sequential Distance Of A Sequential Element

- A sequential element is a latch, register or primary input
- A path is considered from an assertion input to a sequential element
- Sequential distance of a path is the number of sequential elements in the path
- Minimum distance of all paths from an assertion to an element is the reported sequential distance



## Example Sequential Distance Report For A FIFO

Lint-[NTL\_COV\_ASSERT01] Unreachable sequential elements  
fifo.sv, 7  
fifo  
Unreachable sequential elements within given sequential depth of design: fifo  
-Within depth of 0  
-<INF> fifo.data\_out(7)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(6)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(5)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(4)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(3)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(2)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(1)(fifo.sv 33 fifo)  
-<INF> fifo.data\_out(0)(fifo.sv 33 fifo)  
-<INF> fifo.data\_in(7)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(6)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(5)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(4)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(3)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(2)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(1)(fifo.sv 27 fifo)  
-<INF> fifo.data\_in(0)(fifo.sv 27 fifo)

Sequential elements "data\_out" & "data\_in" are unreachable:  
No assertion to monitor "data gets through correctly" !

Reports assertion coverage hole:

Lint-[NTL\_COV\_ASSERT05] Assertions with same inputs  
fifo.sv, 7  
fifo  
Groups of assertions with identical inputs for design fifo  
-Group with identical input:  
-fifo.A\_ASSERT\_ALWAYS\_P(fifo.sv 126 fifo)  
-fifo.full\_write\_full(fifo.sv 93 fifo)

Assertion "A\_ASSERT\_ALWAYS" to check full and write & assertion "full\_write" have identical inputs

Identifies redundant assertion

Lint-[NTL\_COV\_ASSERT03] Number of assertions below depth  
fifo.sv, 7  
fifo  
Number of assertions those are reachable within given depth for sequential elements of design fifo  
-Number of assertions with in all possible depth from the sequential  
- 1 fifo.write\_ptr(0)(fifo.sv 34 fifo)  
- 1 fifo.write\_ptr(1)(fifo.sv 34 fifo)  
- 1 fifo.write\_ptr(2)(fifo.sv 34 fifo)  
- 1 fifo.write\_ptr(3)(fifo.sv 34 fifo)  
- 14 fifo.read\_enb(fifo.sv 25 fifo)  
- 14 fifo.reset(fifo.sv 24 fifo)  
- 14 fifo.status\_cnt(0)(fifo.sv 36 fifo)  
- 14 fifo.status\_cnt(1)(fifo.sv 36 fifo)  
- 14 fifo.status\_cnt(2)(fifo.sv 36 fifo)  
- 14 fifo.write\_enb(fifo.sv 26 fifo)

Sequential distance of a set of elements from the assertion set

Monitors effectiveness of assertion set