Ryan Laur, Benjamin Wheeler

Lab 2 Report

Reconfigurable Computing 2

# Bugs with done signal

## Done cleared too early

Preconditions:

* DUT in done state (done == 1)
* Testbench restarted DUT via go == 1

Violation:

* Done not cleared on following cycle (violates design specification)

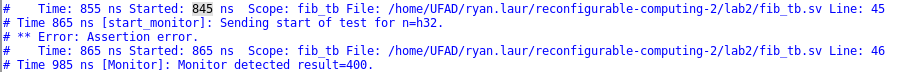
Detected in fib\_tb.sv line 100:

*// if go and done are both asserted, done should be cleared on the next cycle*

  assert property (@(posedge bfm.clk) disable iff (bfm.rst) bfm.go && bfm.done |=> !bfm.done)

  else $error("Time %0t [Assert Property]: Done=1, go=1, done not cleared next cycle.", $time);

Modelsim errors:



Timeline

Description automatically generated

## Done cleared too late

Preconditions:

* DUT in done state (done == 1)
* Testbench restarted DUT via go == 1

Violation:

* Done took two cycles to clear instead of one.

Detected in fib\_tb.sv line 100 (see assertion property for first error).



A picture containing timeline

Description automatically generated

# Infinite loop bugs

## Max number as an input causes state machine to never finish.

Preconditions:

* Circuit fed input of max number (e.g., WIDTH’(1’b1))
  + This input is the largest number the circuit can handle for its specified width.

Violation:

* Circuit never asserts done, gets caught in infinite loop

Detected via a watchdog (monitor.svh line 131) that quits the simulation after done is not seen for 100k cycles.

Graphical user interface, application, Word

Description automatically generated

# Bug: Result register changes when circuit done

Preconditions:

* Circuit reaches done state
* Circuit asserts done signal

Violation:

* Result register changes as done is asserted, which should not happen.

Detected in fib\_tb.sv line 120, 123:

*// upon completion, (ie done = 1), result and overflow retain their values until circuit is restarted*

  assert property (@(posedge bfm.clk) disable iff (bfm.rst) bfm.done && $stable(bfm.done) |-> $stable(bfm.result))

  else $error("Time %0t [Assert Property]: Done=1, result not stable.", $time);

*// upon completion, (ie done = 1), result and overflow retain their values until circuit is restarted*

  assert property (@(posedge bfm.clk) disable iff (bfm.rst) bfm.done && $stable(bfm.done) |-> $stable(bfm.overflow))

  else $error("Time %0t [Assert Property]: Done=1, overflow not stable.", $time);

Graphical user interface, application

Description automatically generated

# Bug: Circuit reacts to changes on the input

Preconditions:

* Circuit is started with go
* Input n is changed before the circuit finishes (before done == 1)

Violation:

* Circuit does not produce correct output

Discovered via scoreboard logic that ensures l, x, and y registers have correct values compared to the model function. Since they were different, this means the DUT is stopping before it should and thus producing incorrect output.

Graphical user interface

Description automatically generated

# Overflow bugs

Text

Description automatically generated

Breaks at n = 0x3f

Text

Description automatically generated