

Verification Mind Games

Jeffrey Montesano, Verilab
 Montreal, Canada
 Jeffrey.Montesano@verilab.com

Mark Litterick, Verilab
 Munich, Germany
 Mark.Litterick@verilab.com

Is it the verification environment's duty to accurately **replicate the real world**?

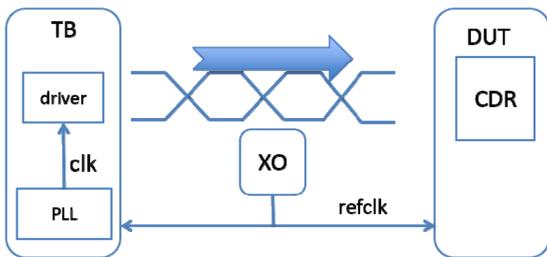
Where is the line between **inter-operating** with the design and **stressing** it?

To what extent must the verification component **follow the design protocol**?

Is it acceptable for the testbench and/or testcases to **make use of design signals**?

Is it worthwhile to target corner **cases that designers consider invalid**?

The DUT is sending data without an accompanying clock - should my VC do Clock-Data-Recovery?...**No**.

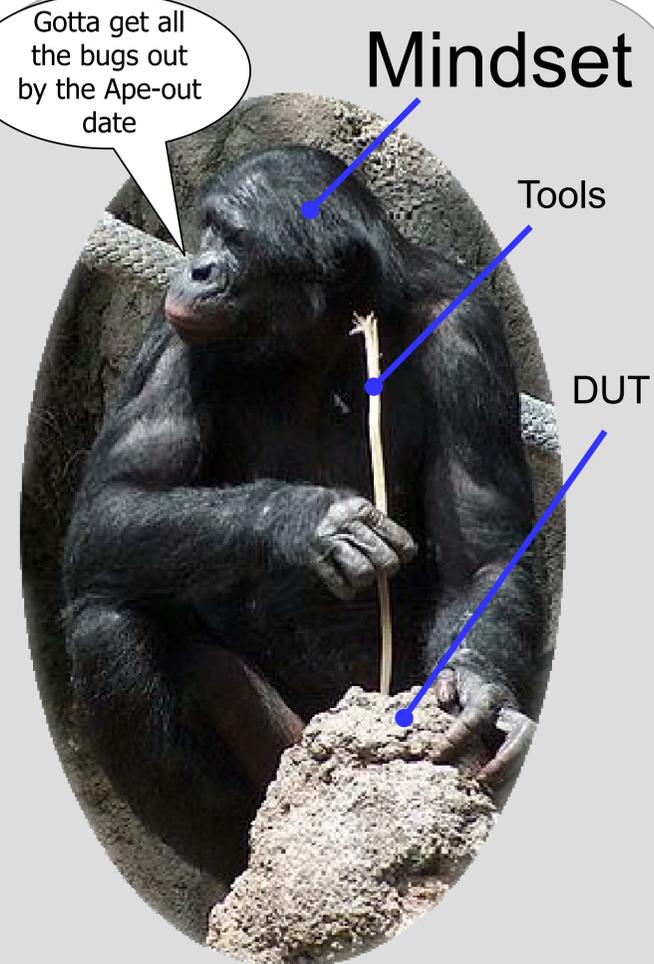


The protocol says that when an error condition is detected, the design must send a NACK packet. Should my VC automatically send NACK too?... **No**.

The design indicates FIFO fullness with signals "full" and "empty". Can my VC or testcase make use of them to prevent overflow/underflow?... **Yes**, but only if checks are made on them.

Is it sufficient to limit error injection to the scenarios for which the DUT has detection capabilities?... **No**.

A design has a register bit that defaults to '0', and causes the DUT to enter low-power mode when written with '1'. Is writing '0' when it is already at '0' important to test?... **Yes**.



The Verification Engineer

Can I snoop the design's internal clock to synchronize my VC to?... **No**.

Should I ask myself "what is it I'm trying to accomplish here?" when embarking on a new verification task... **Yes**.

Is it my responsibility to ensure that the design architect and design engineer are on the same page?... **Yes**.

Should I regularly step back from low-level implementation and take a high-level view of the verification effort as a whole?... **Yes**.

The protocol has bi-directional signals with the potential for multiple masters. Should I split each signal into two at the VC interface level?... **Yes**.

```
interface protocol_if(inout data, inout clk);
    logic clk_o = 1;
    logic clk_i;
    logic data_o = 1;
    logic data_i;

    assign (highz1,strong0) data = data_o;
    assign (highz1,strong0) clk = clk_o;

    assign data_i = data;
    assign clk_i = clk;
endinterface: protocol_if
```

A design draws circles of radius given by an input parameter. Is testing a radius of zero important?... **Yes**.

Should I implement coverage on individual register values?...**No**.

Should I be relying mainly on waveforms to debug my VC?...**No**.

Is coverage closure more important than testcase passing rate?... **Yes**.

Is it necessary to allow a simulation to continue running after it has encountered an error?... **No**.

