What is Verification?

Verification ensures that the RTL performs the correct function.

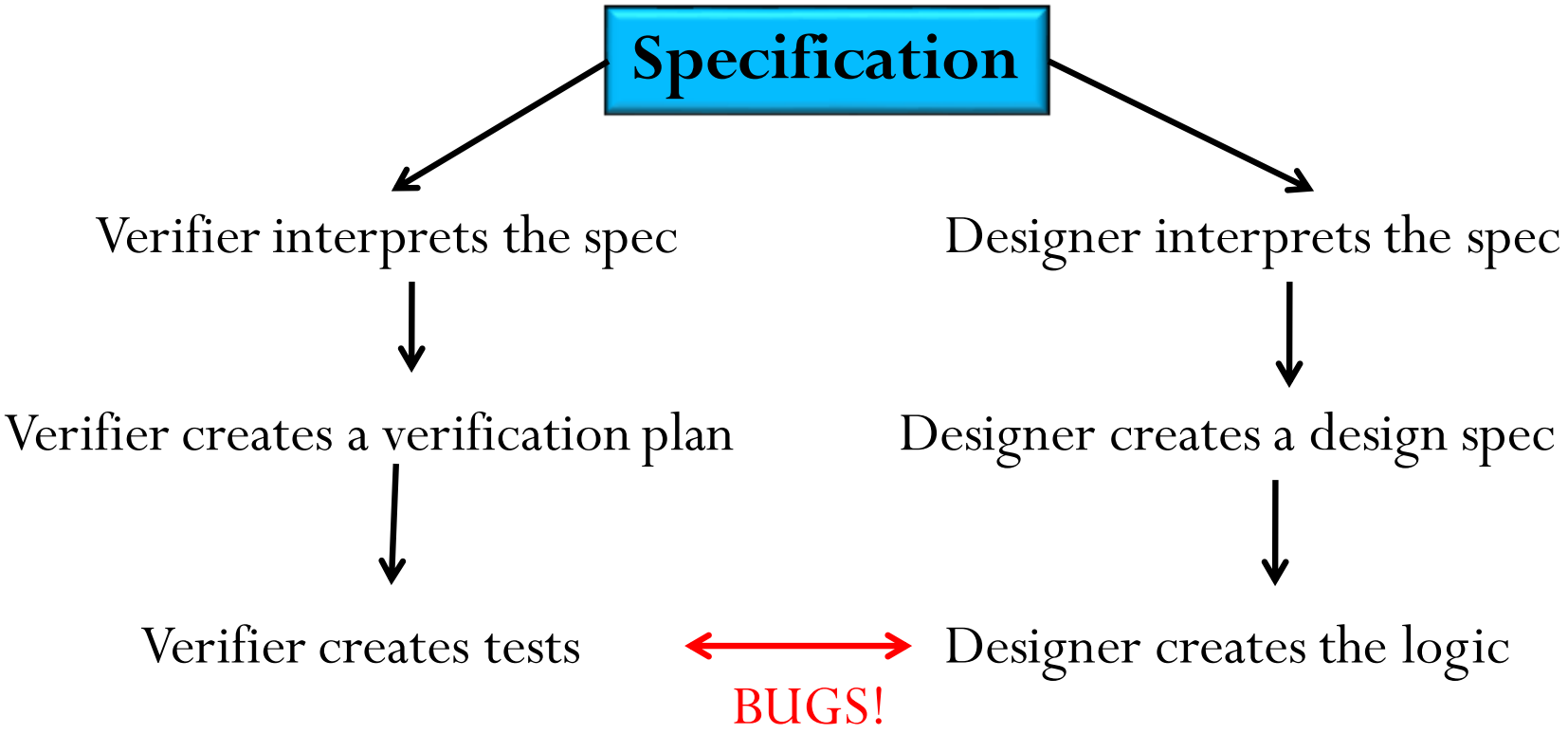
What defines the correct function of the RTL?

Verification ⟵ Specification ⟶ Design

What is the format of the specification?

1. paper document
2. Executable spec
    1. SystemC
    2. C++
    3. Matlab
    4. etc.

The Verification Process



The Verification Process

A hardware design mostly consists of several ***Verilog (.v)*** files with one top module, in which all other sub-modules are instantiated to achieve the desired behavior and functionality.

An environment called ***testbench*** is required for the verification of a given Verilog design and is usually written in ***System Verilog***.

The idea is to drive the design with different stimuli to observe its outputs and compare it with expected values to see if the design is behaving the way it should.

The Verification Process

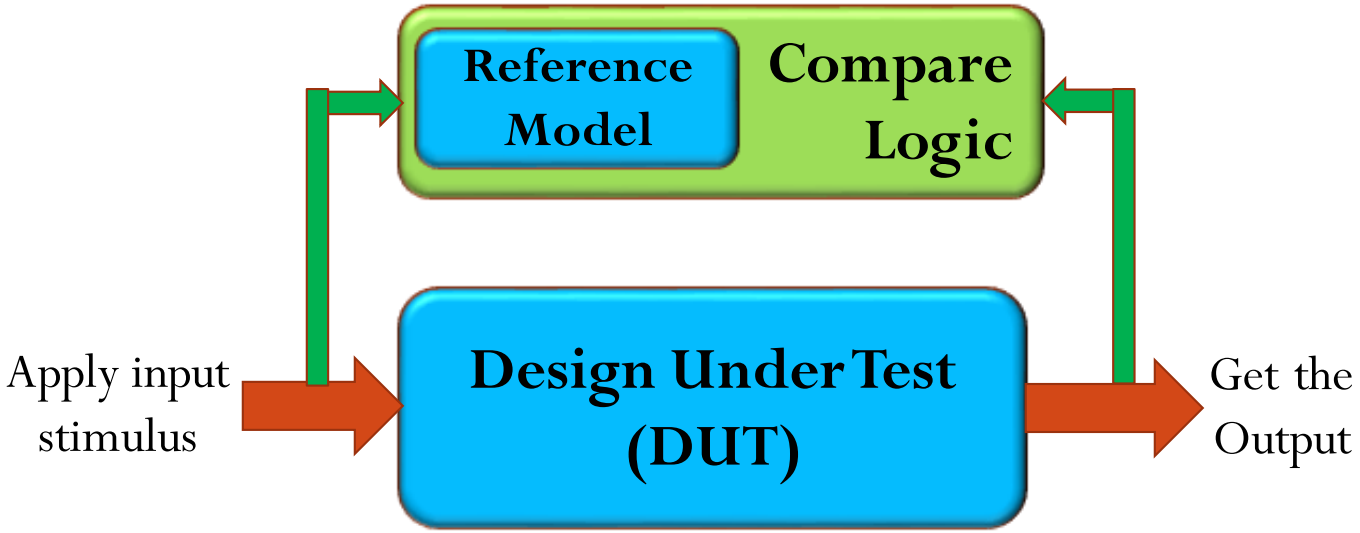
In order to do verification, the top level design module is instantiated within the testbench environment and design input/output ports are connected with the appropriate testbench component signals.

The inputs to the design are driven with certain values for which we know how the design should operate.

The outputs are analyzed and compared with the expected values to see if the design behavior is correct.

How to Verify the Design?

Compare DUT output with expected result.



```systemverilog
// File : d_ff.v
module d_ff (clk, resetn, q, d);

  input clk;
  input resetn;
  input d;
  output q;

  reg q;

  always @ (posedge clk)
    if (! resetn)
      q <= 0;
    else
      q <= d;

endmodule
```

Design

```systemverilog
// File : tb_top.sv
module tb_top ();

  reg clk=0;
  reg resetn;
  reg d;
  wire q;

  // Instantiate the design
  d_ff  d_ff0 (  .clk (clk),
                 .resetn (resetn),
                 .d (d),
                 .q (q));

  // Create a clock
  always #10 clk <= ~clk;

  initial begin
    resetn <= 0;
    d <= 0;

    #10 resetn <= 1;
    #5       d <= 1;
    #8       d <= 0;
    #2       d <= 1;
    #10      d <= 0;
    #50 $finish;
  end
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
endmodule
```

Test Bench