averant

Solidify™ AutoChecks

Automatic Design Checks

Automatic Checks

- Dead Code
- Deadlock / Livelock
- Clock Domain Crossing (CDC)
- CDC Data Stability & Gray Coding
- Full / Parallel Case Pragma
- Reset Propagation
- Contention
- Array Bounds
- X-Assignment Propagation

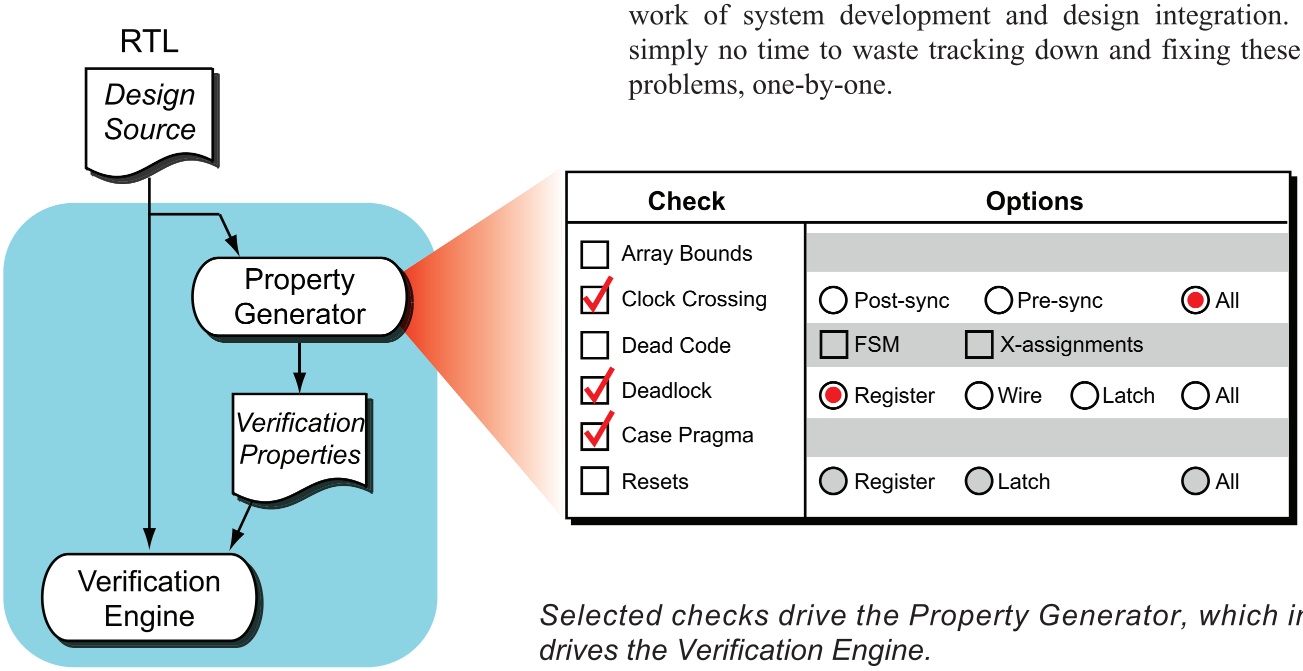
Overview

AutoChecks is a program that automatically checks a circuit for a number of common design problems. Reading only the design source, and with very little input from the user, AutoChecks tracks down such problems as clock domain crossing problems, and a host of other problems that cause a design to function improperly.

AutoChecks is built upon the Solidify formal verification engine, which has proven itself in over 5 years of commercial use. It is fast, thorough, and very easy to use.

Common Problems

Even the most carefully crafted design can be plagued with common yet annoying problems – clock domain synchronization, missing resets, bus contention, lock-up, unreachable states – the list is known to us all. Discovering and fixing these problems is a tedious task that takes time and focus away from the more serious work of system development and design integration. There is simply no time to waste tracking down and fixing these types of problems, one-by-one.



Check	Options
□ Array Bounds	
☑ Clock Crossing	○ Post-sync ○ Pre-sync ● All
□ Dead Code	□ FSM □ X-assignments
☑ Deadlock	● Register ○ Wire ○ Latch ○ All
☑ Case Pragma	
□ Resets	○ Register ○ Latch ○ All

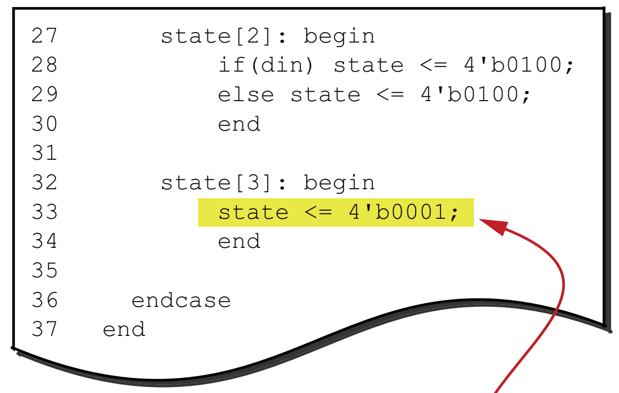
Selected checks drive the Property Generator, which in turn drives the Verification Engine.

Push-button Solution

AutoChecks presents a very simple user interface. The primary inputs are the design source files in Verilog or VHDL (or mixed) format. Also required is a specification of the reset sequence – the sequence applied to the reset signal (if any) that resets the circuit to an initial, stable state. This is typically a single line of code – often just a simple statement.

The user then selects the tests they wish to run. Specific options for some tests – eg., Deadlock check may be applied to registers, wires, latches, outputs, or "all" – may also be chosen. The tool runs automatically and details any problems in the output report.

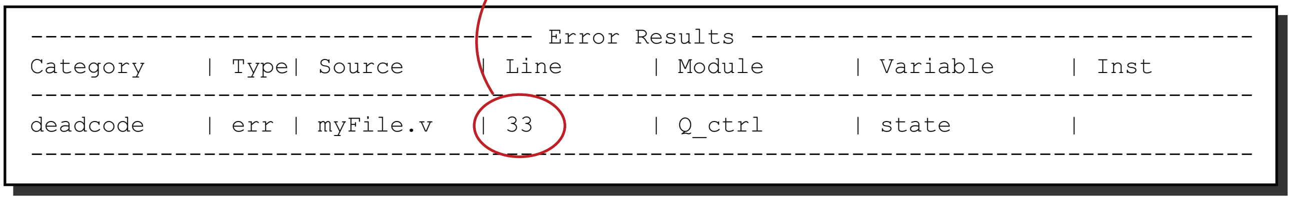
myFile.v



```
27      state[2]: begin
28          if(din) state <= 4'b0100;
29          else state <= 4'b0100;
30          end
31
32      state[3]: begin
33          state <= 4'b0001;
34          end
35
36    endcase
37  end
```

Pinpoint Results

Errors uncovered by the program point to the file, line, and variable that needs attention. The user is directed back into their source files to speed the process of making the fix.



```
----------------------------------- Error Results ------------------------------------
Category     | Type| Source       | Line       | Module        | Variable      | Inst
----------------------------------------------------------------------------------------
deadcode     | err | myFile.v     | 33         | Q_ctrl        | state         |
----------------------------------------------------------------------------------------
```

Solid Benefits

Being able to quickly find and fix a host of common design problems – clock domain crossing, deadlock, deadcode, contention, etc. – is extremely helpful to designers, especially considering the complexity of modern designs and the schedule deadlines that must be met. Employing Solidify AutoChecks from the beginning of a project further reduces the cost of finding and fixing bugs.

AutoChecks is a powerful tool to help you reduce risk, meet time-to-market pressures, and produce solid designs quickly and with confidence.

Averant, Inc.
22320 Foothill Blvd., suite 360
Hayward, CA 94541

averant

Tel: 510-881-8881
Fax: 510-881-8886
www.averant.com