Automatic Bug Fixing
Bug Fixes

Added new code segment in a commit:

```verilog
always_comb
begin : MUX
  if (sel == 1'b0)
    begin
      mux_out = din_0;
    end
  else begin
    mux_out = din_1;
  end
end
```

2 Ways to Fix

- Fix Root Cause
- Unroll Entire Commit

`begin` Syntax Error

```verilog
always_comb
begin : MUX
  if (sel == 1'b0) begin
    mux_out = din_0;
  end else begin
    mux_out = din_1;
  end
end
```

```verilog
always_comb
begin : MUX
  if (sel == 1'b0) begin
    mux_out = din_0;
  end else begin
    mux_out = din_1;
  end
end
```
Regression Test Setups

**Continuous Integration**
- Gate keeper
- Short directed tests

**Post-Integration Testing**
- Large Test Suites
- Random/Directed Tests

**Automatic Debug**
- Inhouse Solution (e.g. using GIT Bisect)
  
  ![PinDown](image)

- Test passed
  Change is integrated into the Revision Control System (RCS)

- Test failed
  Bug report sent back to engineer

Change
PinDown Automated Debug Process

Automated, Validated Identification of Faulty Code

Regression Test Failures → Pin-Point → Validate → Assign Bug Report

- Random Tests
- Directed Tests
- Single Cause
- Multiple Cause
- Build Failures
- Test Failures
- Intermittent

PinDown Patented Algo to find Bugs:
- Smart Test Grouping
- Run Debug Tests
- Smart Pin-Pointing

Fixing the Bugs

Pass

Fail

Revision Control System

Committing Fix (optional)
PinDown Debug Algorithm

1. Grouping
   - Bug Group 1: "Error: RX value wrong" 12 test failures
   - Bug Group 2: "Address is XXXXXXX" 14 test failures

2. Smart Selection
   - Identify possible bad commits
   - Fastest Test

3. Unroll Changes
   - 10 selections in parallel

4. Verify The Rest
   - Test passes
   - No pass

Revision Control System

Next Bug

Bug No: 1 (new bug)
Test: test34_vm_seed_0x45679832
Build: dtlb_thdr
Error: Error: RX value wrong
Committer: johndoe
Commit Message: 979301: RX performance improvement
Committed Files:
//depot/bigproject/dv/dtlb/dtlb_rqe/common/interface/dtlb_rqe_tb_if.sv
Log:
The Principle is Simple

...but the reality is difficult

PinDown supports all scenarios
Solution: Before committing a fix, check if a fix has already been committed. Both humans and tools need to do this.
Solution: The tool must not unroll its own fixes. In those cases just send a bug report.
Partial Introduction of a New Feature

Solution: Don’t allow partial introduction of a feature which makes a test fail. This rule applies to a continuous integration gate keeper as well.

Constrained random tests may find issues weeks after inserting
PinDown Automated Debug Process

Automated, Validated Identification of Faulty Code

Regression Test Failures ➔ Pin-Point ➔ Validate ➔ Assign Bug Report

- PinDown Patented Algo to find Bugs:
  - Smart Test Grouping
  - Run Debug Tests
  - Smart Pin-Pointing

Fixing the Bugs

- Automatically fixing bugs locally is advanced and has no side-effects

Revision Control System

Automatically committing the fix is easy to implement but has serious side-effects (human-tool race, fault oscillation, implications for partial introduction of new failures)
Summary

Continuous Integration
Gate Keeper

Test Suites
Short Test Suites
Directed Tests Only

Policy Implications
No partial checkins that causes tests to fail

Automatic Bug Fix
Local

Test Suites
Any Test Suite Size
Random, Directed Tests

Policy Implications
None

Automatic Bug Fix
with Commit

Test Suites
Any Test Suite Size
Random, Directed Tests

Policy Implications
- No partial checkins that causes tests to fail
- Check tool/RCS before commit