100% Functional Coverage with Formal Methodology

Suckheui Chung
Advanced Micro Devices (AMD)
suckheui.chung@amd.com

September 24, 2015
SNUG Boston

Agenda

Functional Coverage

Verification Challenge

Implementation

Debug

Conclusion

Functional Coverage

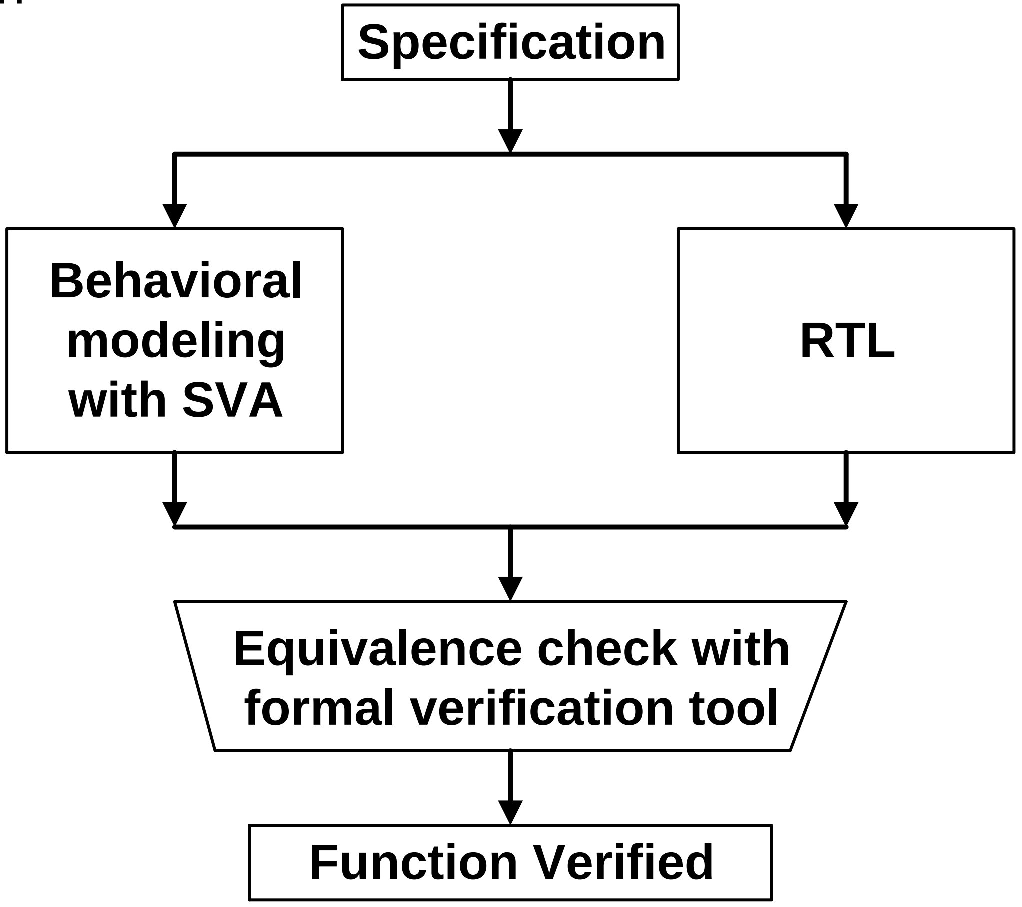
- Challenging goal: 100% functional coverage
  – What does this mean to the formal verification?

- 2 things need to be emphasized

  – Verification of “specification” against RTL
    - Formal verification needs to be updated if the spec is modified
    - Formal verification does not validate the design “intention”
    - Verification fail hole(s) can occur if the spec is not defined for that specific fail case(s)

Functional Coverage

- SVA (System Verilog Assertion): just language to make behavioral models (property) of the spec → No SVA for RTL as design assertion

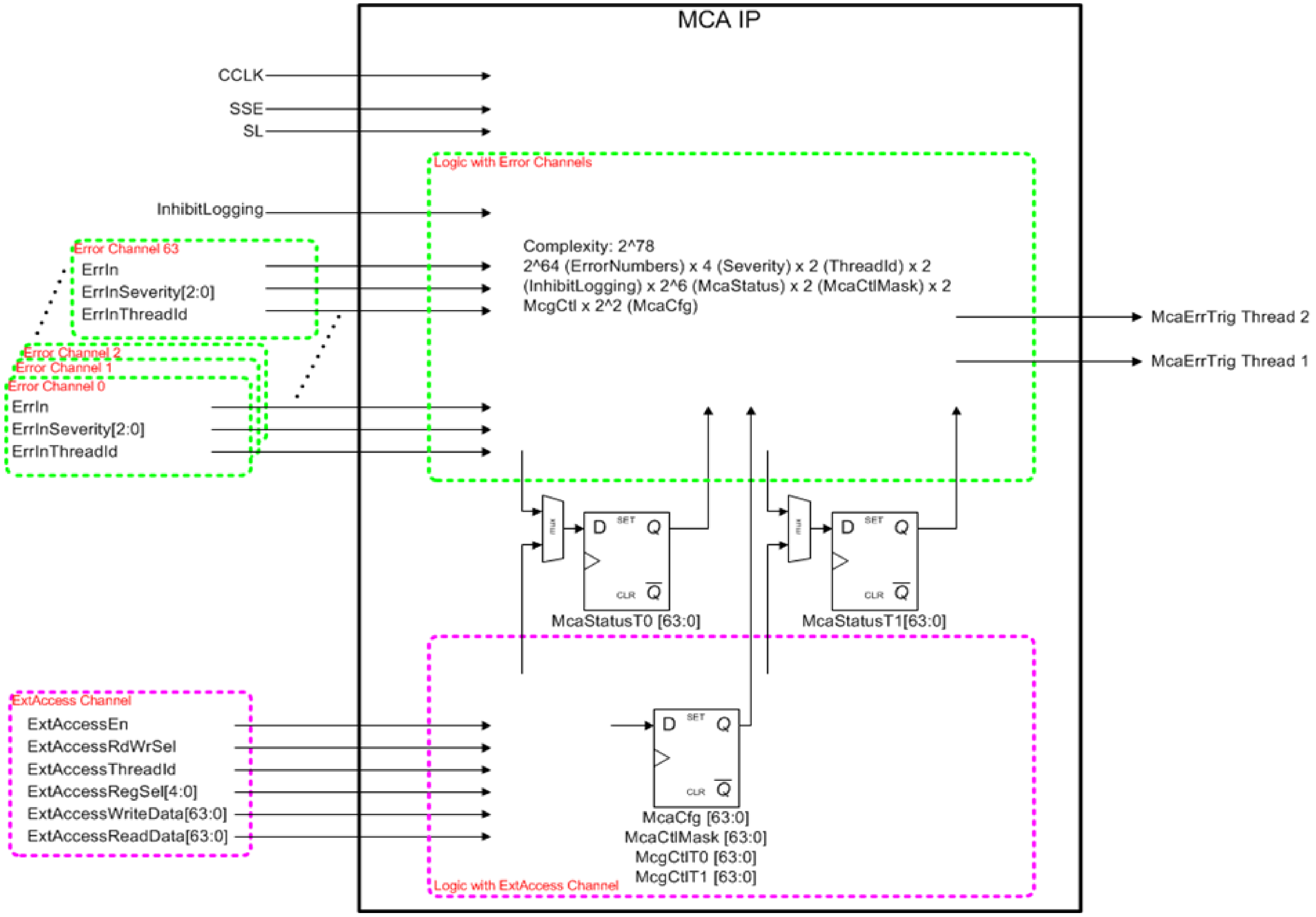

Specification
Behavioral modeling with SVA
RTL
Equivalence check with formal verification tool
Function Verified

Verification Challenge

- MCA (Machine Check Architecture)
  - A cross functional IP over any applicable processor
  - Collecting error information and generate relevant request for OS to react accordingly

- Challenge
  - High complexity and combination of error inputs, multiple register bits, and associated logic
  - The complexity level can go to 2^{78} combinations

Verification Challenge



Verification Challenge

- Advantage of MCA IP design
  - Well defined end-to-end (source-to-destination) points
  - All error inputs (source) contribute to internal registers and outputs (destination)
  - The spec clearly defines logical relations between source and destination

- Formal verification can check if “input to output of RTL ” is identical to the “behavioral model (property) of the spec”

Implementation

- Synopsys vc-static: configuration before run

Mode	SSE	SL
Cold reset (reset for all registers)	1	1
Warm reset (reset for some registers)	1	0
Functional	0	0
Illegal	0	1

```
create_clock CCLK -period 100
sim_force SSE -apply 1'b1
sim_force SL  -apply 1'b1
sim_run 2
sim_save_reset
fvassume nc -expr {{SSE, SL} != 2'b01}
```

  – Fewer constraints are recommended for higher coverage

Implementation

- 3 examples
  - Case 1: cold reset
  - Case 2: falsified
  - Case 3: vacuous

Implementation

- Case 1: cold reset
  - Spec

```
After cold reset, McaDefAddr register should have all 0's for bit[63:0]
```

  - Property: behavioral model using SVA

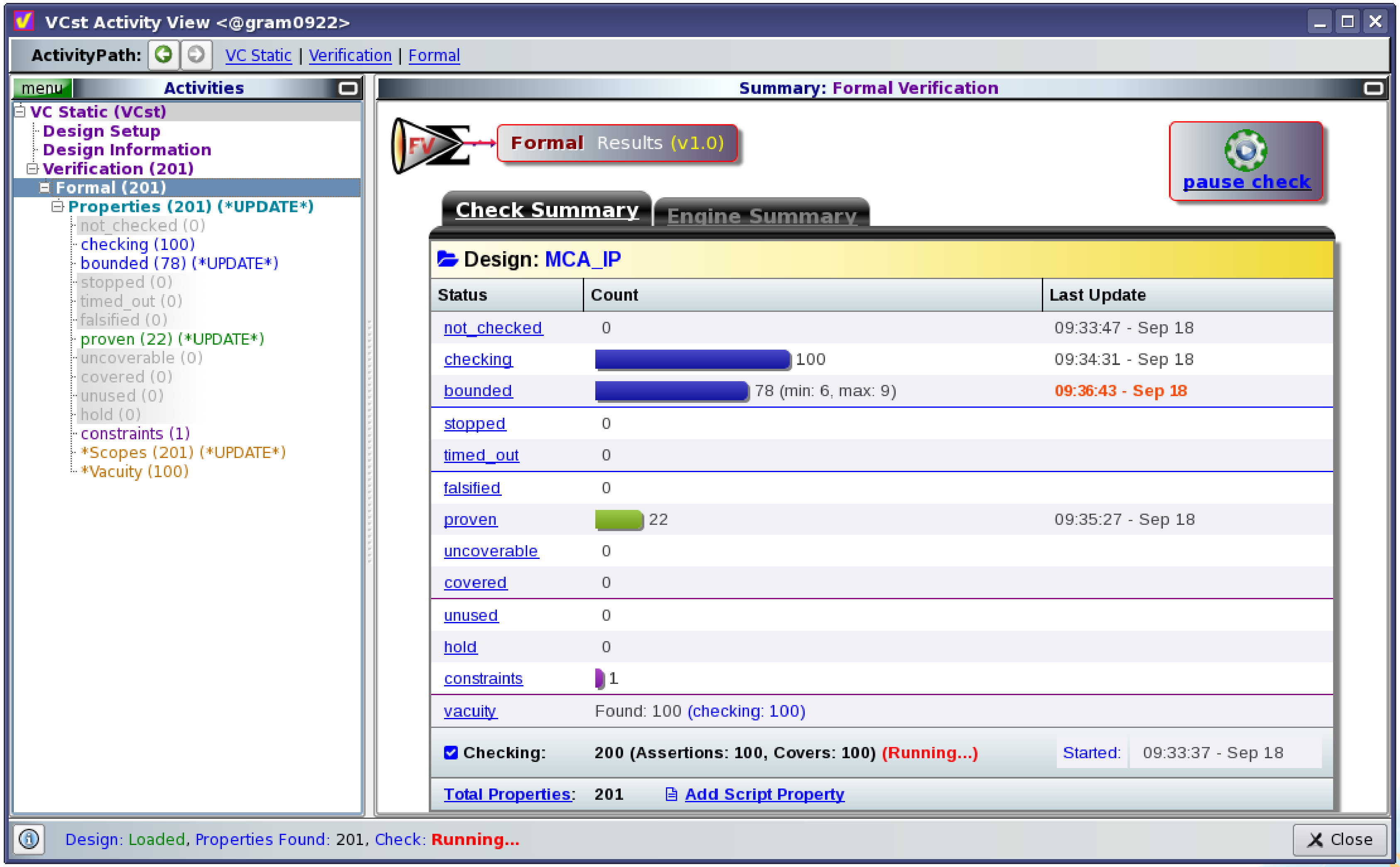
```
property hardreset (logic SSE, logic coldreset, logic [63:0] regs);
  @(posedge clk) disable iff (SSE)  $fell(coldreset) |-> ##0 (regs[63:0] == 0);
endproperty
```

  - Assertion and coverage

```
aDef: assert property (hardreset(SSE, coldreset, McaDefAddr)) else $fatal
($stime,,,"%m assert FAIL");
cDef: cover  property (hardreset(SSE, coldreset, McaDefAddr)) $display
("%m cover PASS @%0d", $time);
```

Implementation

- Case 1: cold reset → UI during run



VCst Activity View <@gram0922>

ActivityPath: VC Static | Verification | Formal

Activities

- VC Static (VCst)
  - Design Setup
  - Design Information
  - Verification (201)
    - Formal (201)
      - Properties (201) (*UPDATE*)
        - not_checked (0)
        - checking (100)
        - bounded (78) (*UPDATE*)
        - stopped (0)
        - timed_out (0)
        - falsified (0)
        - proven (22) (*UPDATE*)
        - uncoverable (0)
        - covered (0)
        - unused (0)
        - hold (0)
        - constraints (1)
        - *Scopes (201) (*UPDATE*)
        - *Vacuity (100)

Summary: Formal Verification

Formal Results (v1.0)

pause check

Check Summary | Engine Summary

Design: MCA_IP

Status	Count	Last Update
not_checked	0	09:33:47 - Sep 18
checking	100	09:34:31 - Sep 18
bounded	78 (min: 6, max: 9)	09:36:43 - Sep 18
stopped	0	
timed_out	0	
falsified	0	
proven	22	09:35:27 - Sep 18
uncoverable	0	
covered	0	
unused	0	
hold	0	
constraints	1	
vacuity	Found: 100 (checking: 100)	

☑ Checking: 200 (Assertions: 100, Covers: 100) (Running...) Started: 09:33:37 - Sep 18

Total Properties: 201 Add Script Property

Design: Loaded, Properties Found: 201, Check: Running...

Close

Implementation

- Case 2: falsified → modeling fault
  - Spec

The status register is stable when the error is transparent and the configuration bit[33] is low.

  - Property

```
property Status_stable_disable_cfg (logic allreset, logic error_valid, logic cfg,
logic [63:0] regs);
  @(posedge clk) disable iff (allreset) (error_valid & ~cfg) |-> ##0
$stable(regs);
endproperty
```

  - This failed, because reset or extaccess can happen at the same time errors happen. → missing functions in the model

Implementation

- Case 2: falsified → modeling fault
  - 2 more conditions added for reset and extaccess

```
property Status_stable_disable_cfg (logic allreset, logic block_reset, logic
block_ext, logic error_valid, logic cfg, logic [63:0] regs);
  @(posedge clk) disable iff (allreset) (block_reset & block_ext & error_valid
& ~cfg) |-> ##0 $stable(regs);
endproperty
```

Implementation

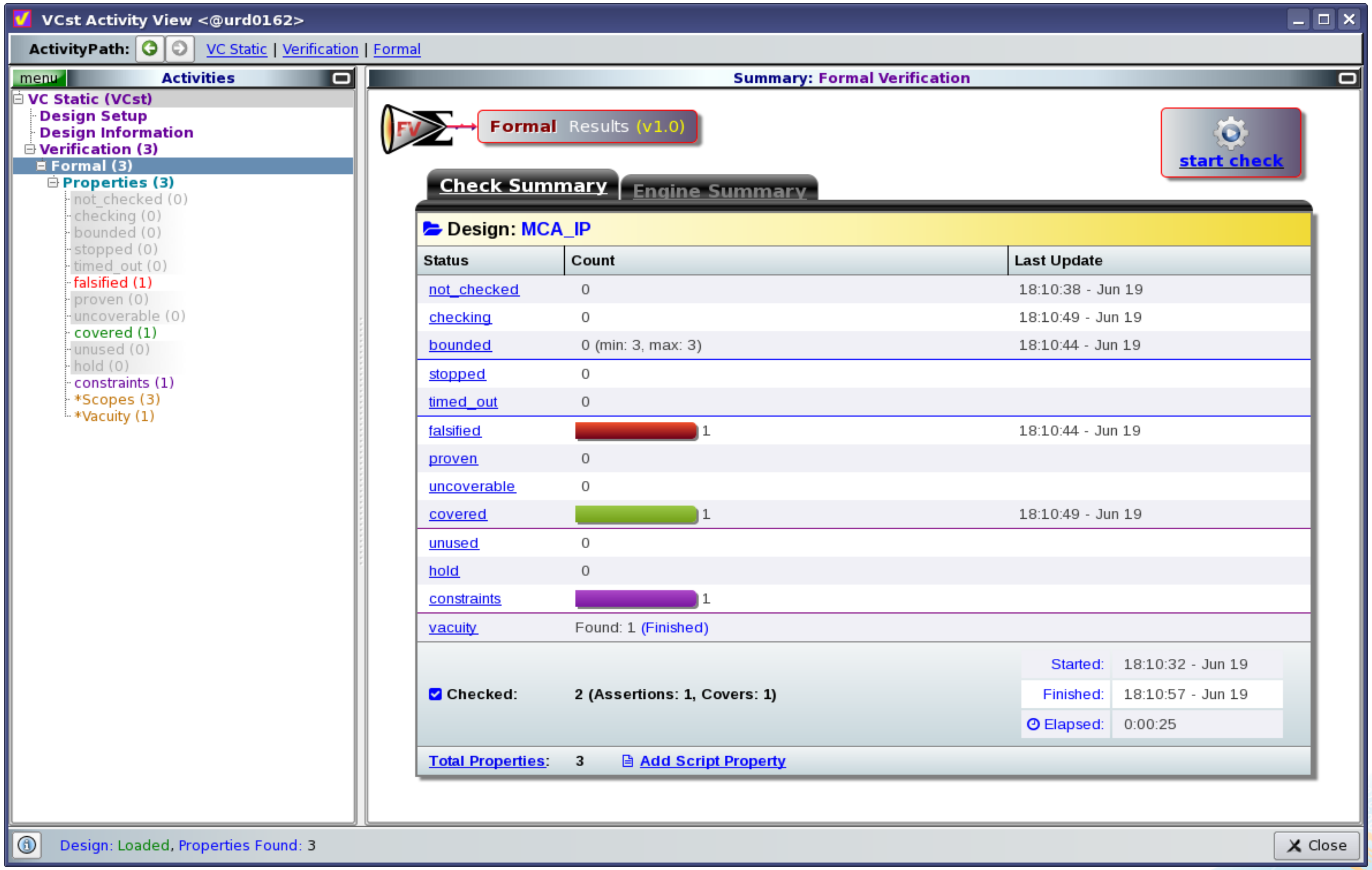
- Case 2: falsified → modeling fault

```
always @(posedge clk) begin
  block_reset_pos <= ~SSE && ~($past(SSE)) && ~($past(SSE,2)) && ~SL
&& ~($past(SL))  && ~($past(SL,2));
end

always @(negedge clk) begin
  block_reset_neg <= ~SSE && ~($past(SSE)) && ~($past(SSE,2)) && ~SL
&& ~($past(SL)) && ~($past(SL,2));
end

assign block_reset = block_reset_pos & block_reset_neg;
```

```
always @(posedge clk) begin
  block_ext <= ~ExtAccessEn && ~($past(ExtAccessEn)) &&
~($past(ExtAccessEn,2)) && ~($past(ExtAccessEn,3));
end
```

Debug

- Case 2: falsified → modeling fault


VCst Activity View <@urd0162>
ActivityPath: VC Static | Verification | Formal
menu
Activities
Summary: Formal Verification
Formal Results (v1.0)
start check
Check Summary
Engine Summary
Close

VC Static (VCst)
- Design Setup
- Design Information
- Verification (3)
  - Formal (3)
    - Properties (3)
      - not_checked (0)
      - checking (0)
      - bounded (0)
      - stopped (0)
      - timed_out (0)
      - falsified (1)
      - proven (0)
      - uncoverable (0)
      - covered (1)
      - unused (0)
      - hold (0)
      - constraints (1)
      - *Scopes (3)
      - *Vacuity (1)

Design: MCA_IP

Status	Count	Last Update
not_checked	0	18:10:38 - Jun 19
checking	0	18:10:49 - Jun 19
bounded	0 (min: 3, max: 3)	18:10:44 - Jun 19
stopped	0	
timed_out	0	
falsified	1	18:10:44 - Jun 19
proven	0	
uncoverable	0	
covered	1	18:10:49 - Jun 19
unused	0	
hold	0	
constraints	1	
vacuity	Found: 1 (Finished)	

Checked: 2 (Assertions: 1, Covers: 1)

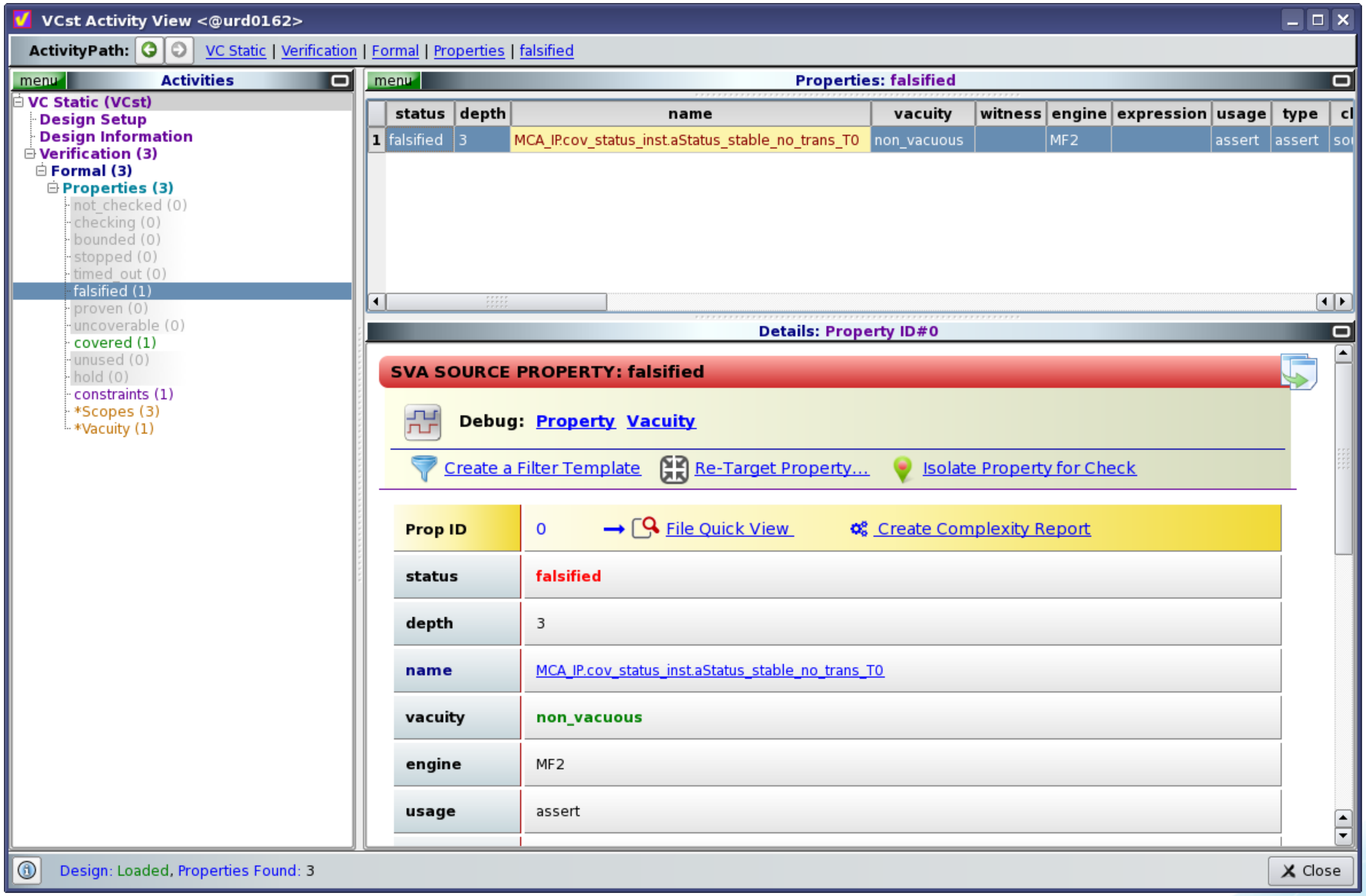
Started:	18:10:32 - Jun 19
Finished:	18:10:57 - Jun 19
Elapsed:	0:00:25

Total Properties: 3 Add Script Property

Design: Loaded, Properties Found: 3

Debug

- Case 2: falsified → modeling fault



VCst Activity View <@urd0162>

ActivityPath: VC Static | Verification | Formal | Properties | falsified

Activities

- VC Static (VCst)
  - Design Setup
  - Design Information
  - Verification (3)
    - Formal (3)
      - Properties (3)
        - not_checked (0)
        - checking (0)
        - bounded (0)
        - stopped (0)
        - timed_out (0)
        - falsified (1)
        - proven (0)
        - uncoverable (0)
        - covered (1)
        - unused (0)
        - hold (0)
        - constraints (1)
        - *Scopes (3)
        - *Vacuity (1)

Properties: falsified

	status	depth	name	vacuity	witness	engine	expression	usage	type	cl
1	falsified	3	MCA_IP.cov_status_inst.aStatus_stable_no_trans_T0	non_vacuous		MF2		assert	assert	so

Details: Property ID#0

SVA SOURCE PROPERTY: falsified

Debug: Property Vacuity

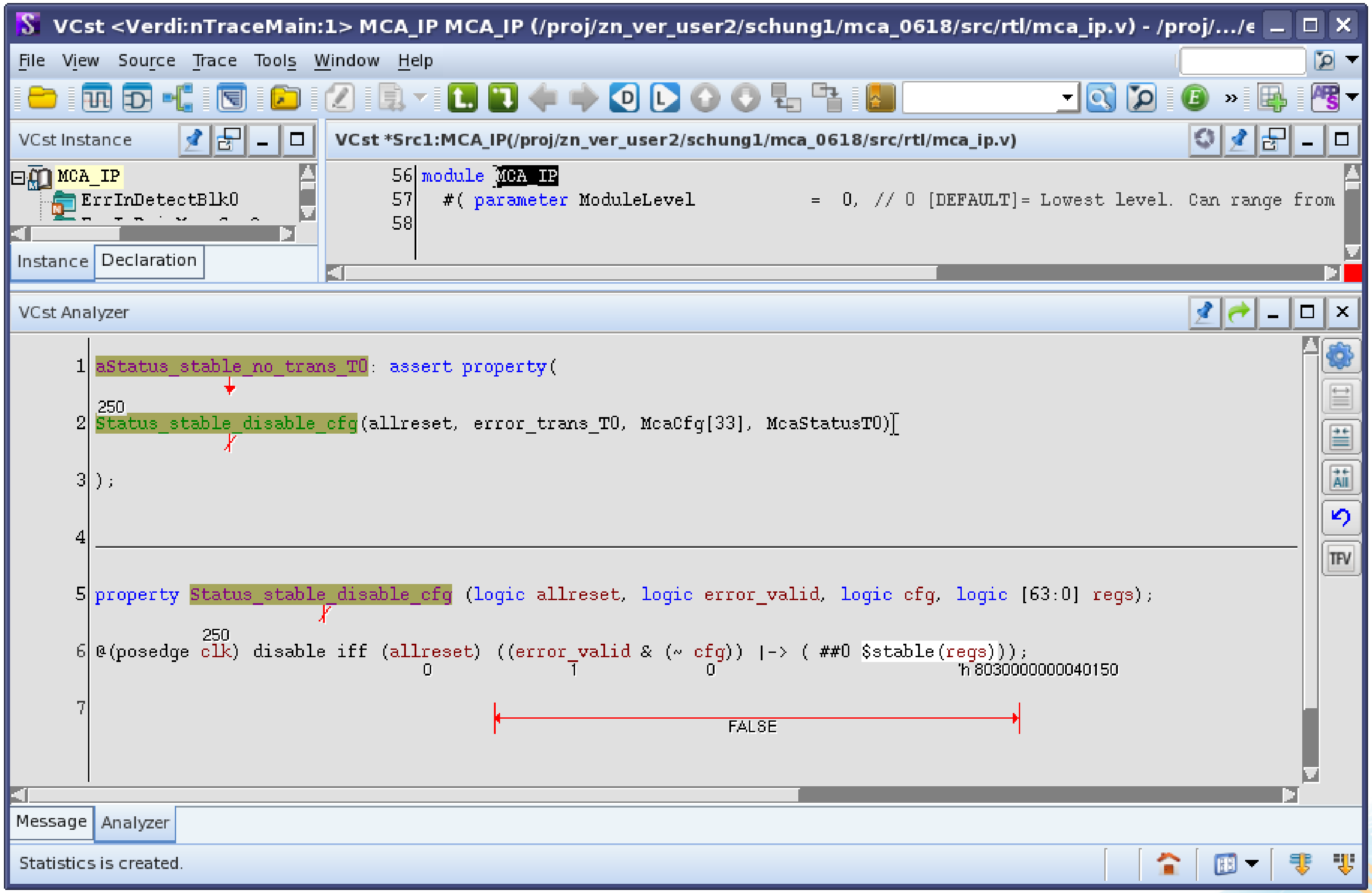
Create a Filter Template   Re-Target Property...   Isolate Property for Check

Prop ID	0 → File Quick View Create Complexity Report
status	falsified
depth	3
name	MCA_IP.cov_status_inst.aStatus_stable_no_trans_T0
vacuity	non_vacuous
engine	MF2
usage	assert

Design: Loaded, Properties Found: 3

Debug

- Case 2: falsified → modeling fault



VCst <Verdi:nTraceMain:1> MCA_IP MCA_IP (/proj/zn_ver_user2/schung1/mca_0618/src/rtl/mca_ip.v) - /proj/.../e

VCst *Src1:MCA_IP(/proj/zn_ver_user2/schung1/mca_0618/src/rtl/mca_ip.v)

```
56 module MCA_IP
57   #( parameter ModuleLevel          =  0, // 0 [DEFAULT]= Lowest level. Can range from
58
```

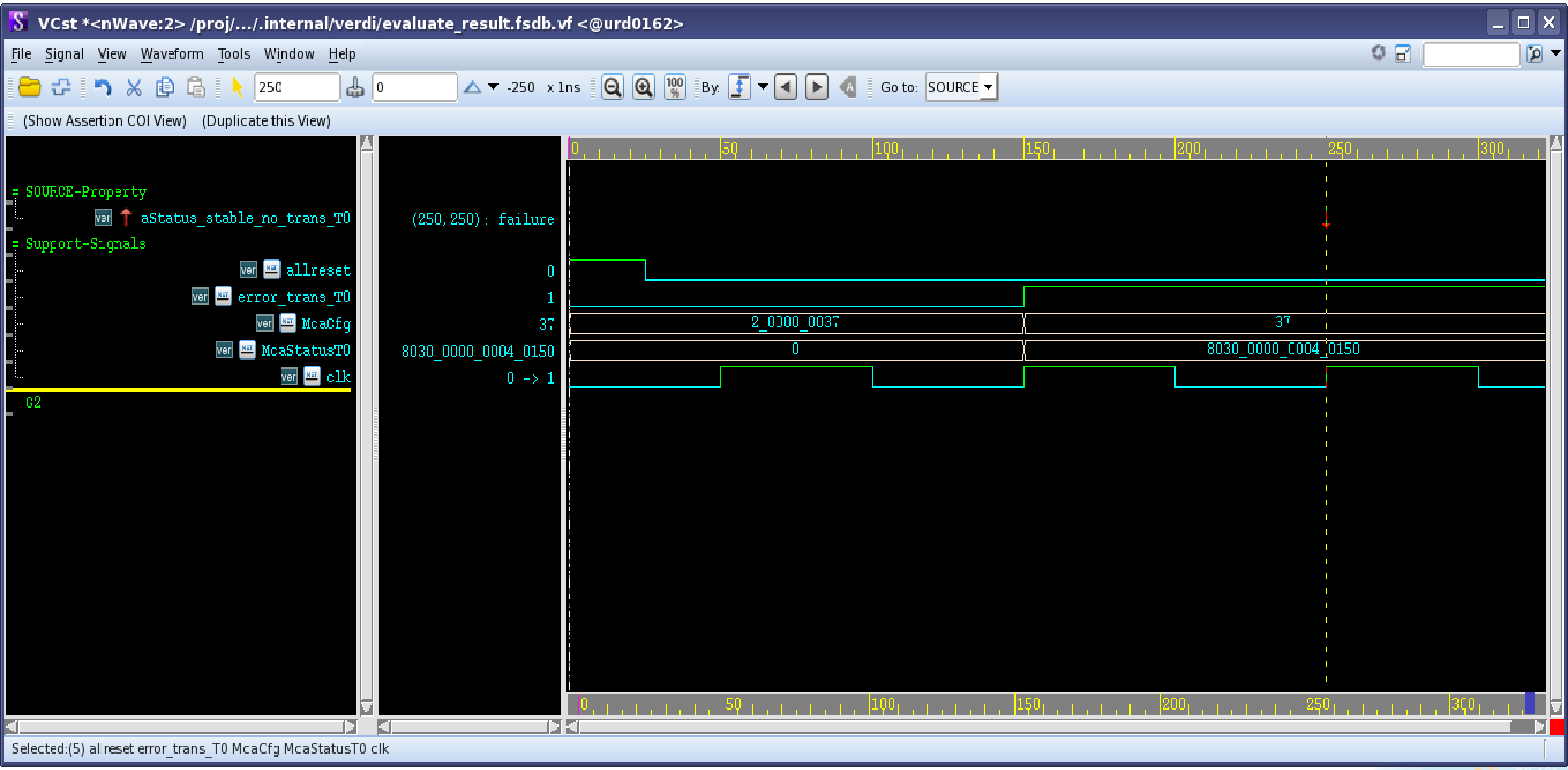
VCst Analyzer

```
aStatus_stable_no_trans_T0: assert property(
Status_stable_disable_cfg(allreset, error_trans_T0, McaCfg[33], McaStatusT0)
);

property Status_stable_disable_cfg (logic allreset, logic error_valid, logic cfg, logic [63:0] regs);
@(posedge clk) disable iff (allreset) ((error_valid & (~ cfg)) |-> ( ##0 $stable(regs)));
                              0              1           0                 'h 8030000000040150
                                         FALSE
```

Statistics is created.

Debug

- Case 2: falsified → modeling fault


VCst *<nWave:2> /proj/.../.internal/verdi/evaluate_result.fsdb.vf <@urd0162>
File Signal View Waveform Tools Window Help
250
0
-250 x 1ns
By:
Go to: SOURCE
(Show Assertion COI View) (Duplicate this View)
0
50
100
150
200
250
300

```
SOURCE-Property
    aStatus_stable_no_trans_T0    (250,250): failure
Support-Signals
    allreset                      0
    error_trans_T0                1
    McaCfg                        37
    McaStatusT0                   8030_0000_0004_0150
    clk                           0 -> 1
G2
```

McaCfg: 2_0000_0037 → 37

McaStatusT0: 0 → 8030_0000_0004_0150

Selected:(5) allreset error_trans_T0 McaCfg McaStatusT0 clk

Implementation

- Case 3: vacuous → uncoverable condition
  - Spec

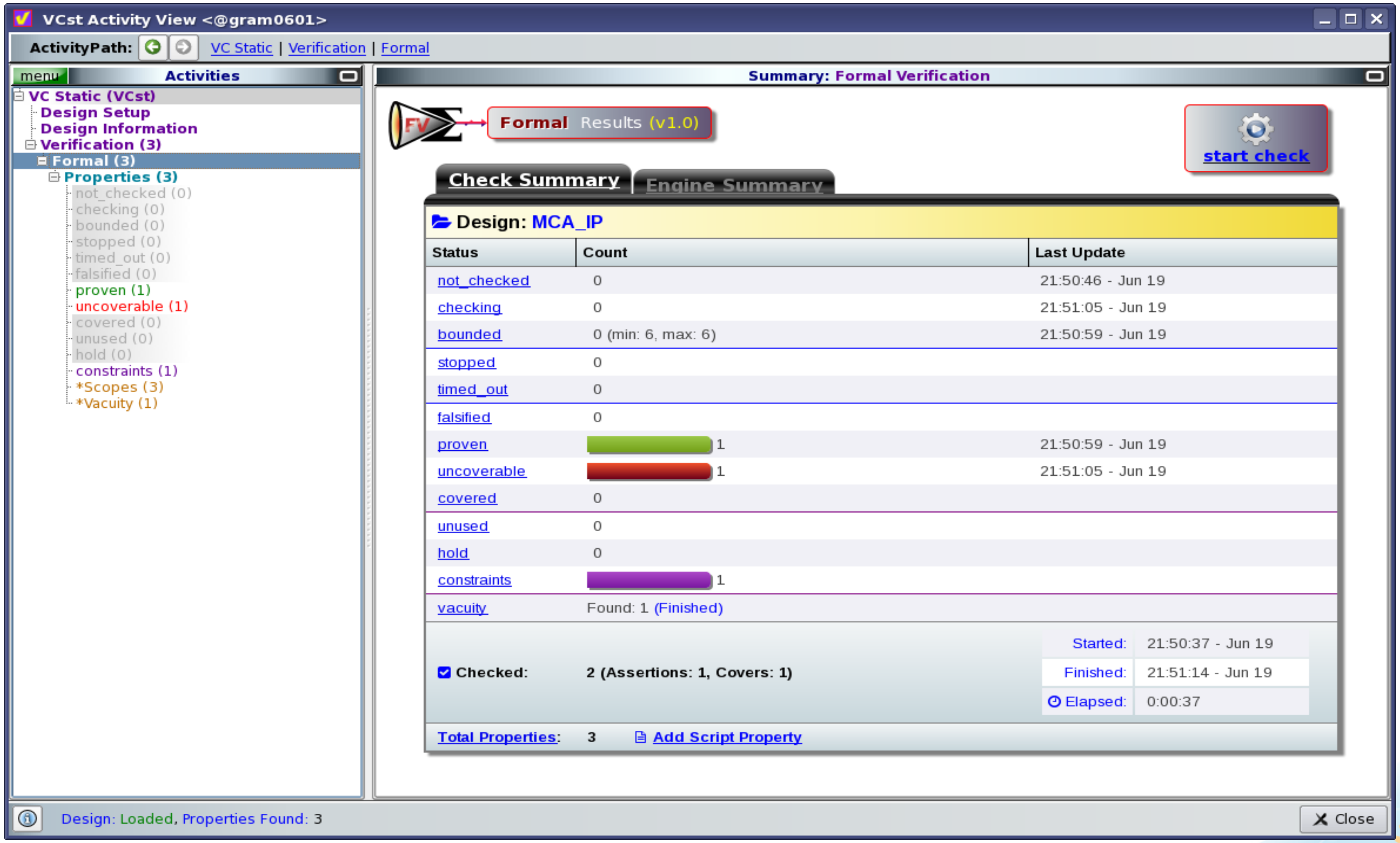
McaErrTrig (Interrupt) happens when a fatal, uncorrected, corrected or transparent error is received and McaStatus register overflow bit[63] is low.

  - Property

```
assign DetectMachineCheckExceptionT0 = int_fatal_T0 | int_uncor_T0 |
int_corre_T0 | int_trans_T0;

property Intreq3 (logic allreset, logic DetectMachineCheckException, logic
block_ext, logic ovf, logic intreq3);
  @(posedge clk) disable iff (allreset)  (~$past(intreq3) &
DetectMachineCheckException & ~ovf & block_ext) |-> ##0 intreq3;
endproperty
```

Debug

- Case 3: vacuous → uncoverable condition
  - ovf should be $past(ovf)



VCst Activity View <@gram0601>

ActivityPath: VC Static | Verification | Formal

Activities

- VC Static (VCst)
  - Design Setup
  - Design Information
  - Verification (3)
    - Formal (3)
      - Properties (3)
        - not_checked (0)
        - checking (0)
        - bounded (0)
        - stopped (0)
        - timed_out (0)
        - falsified (0)
        - proven (1)
        - uncoverable (1)
        - covered (0)
        - unused (0)
        - hold (0)
        - constraints (1)
        - *Scopes (3)
        - *Vacuity (1)

Summary: Formal Verification

Formal Results (v1.0)

start check

Check Summary | Engine Summary

Design: MCA_IP

Status	Count	Last Update
not_checked	0	21:50:46 - Jun 19
checking	0	21:51:05 - Jun 19
bounded	0 (min: 6, max: 6)	21:50:59 - Jun 19
stopped	0	
timed_out	0	
falsified	0	
proven	🟩 1	21:50:59 - Jun 19
uncoverable	🟥 1	21:51:05 - Jun 19
covered	0	
unused	0	
hold	0	
constraints	🟪 1	
vacuity	Found: 1 (Finished)	

Checked: 2 (Assertions: 1, Covers: 1)

Started:	21:50:37 - Jun 19
Finished:	21:51:14 - Jun 19
Elapsed:	0:00:37

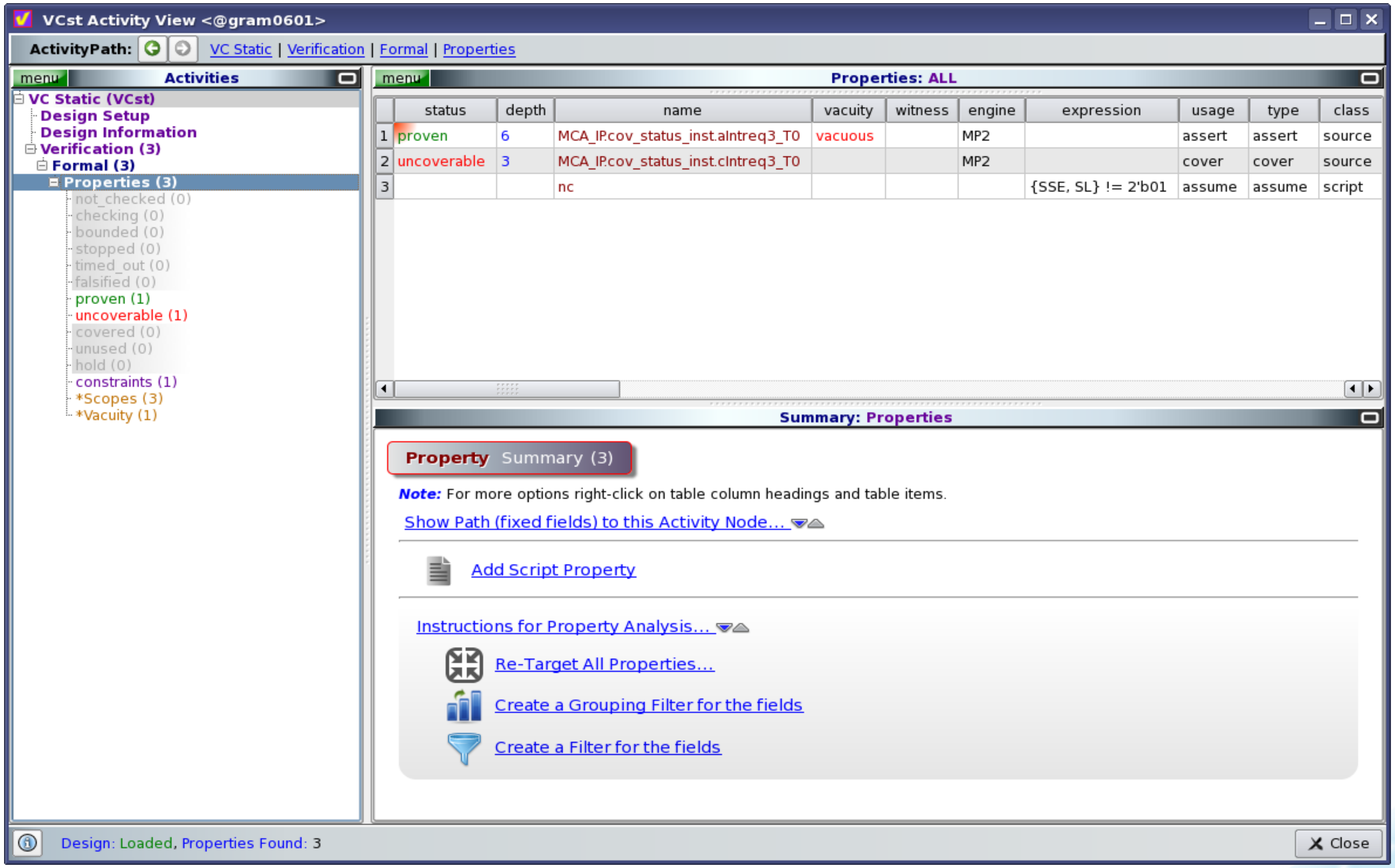
Total Properties: 3   Add Script Property

Design: Loaded, Properties Found: 3

Close

Debug

- Case 3: vacuous → uncoverable condition



VCst Activity View <@gram0601>

ActivityPath: VC Static | Verification | Formal | Properties

Activities

- VC Static (VCst)
  - Design Setup
  - Design Information
  - Verification (3)
    - Formal (3)
      - Properties (3)
        - not_checked (0)
        - checking (0)
        - bounded (0)
        - stopped (0)
        - timed_out (0)
        - falsified (0)
        - proven (1)
        - uncoverable (1)
        - covered (0)
        - unused (0)
        - hold (0)
        - constraints (1)
        - *Scopes (3)
        - *Vacuity (1)

Properties: ALL

	status	depth	name	vacuity	witness	engine	expression	usage	type	class
1	proven	6	MCA_IP.cov_status_inst.aIntreq3_T0	vacuous		MP2		assert	assert	source
2	uncoverable	3	MCA_IP.cov_status_inst.cIntreq3_T0			MP2		cover	cover	source
3			nc				{SSE, SL} != 2'b01	assume	assume	script

Summary: Properties

Property Summary (3)

Note: For more options right-click on table column headings and table items.

Show Path (fixed fields) to this Activity Node...

Add Script Property

Instructions for Property Analysis...

Re-Target All Properties...

Create a Grouping Filter for the fields

Create a Filter for the fields

Design: Loaded, Properties Found: 3

Close

Conclusion

- 100% functional coverage because:
  - All functions of the IP are end-to-end (source-to-destination).
  - Each function can be modeled with SVA.
  - The spec is well defined and documented before or in the middle of the project.
- The formal verification as the main verification methodology → simulation based verification as supplementary.

Conclusion

- vc-static: mature to use
  - About 35 min (total time via LSF) for 100 complex properties (100 assertion and 100 coverage to be verified by the tool)
  - Most of time spent → modeling with SVA
- Random simulation was still useful to find the "spec missing points" → Formal verification cannot do this!
  - System level free-run simulation is still important

Thank You