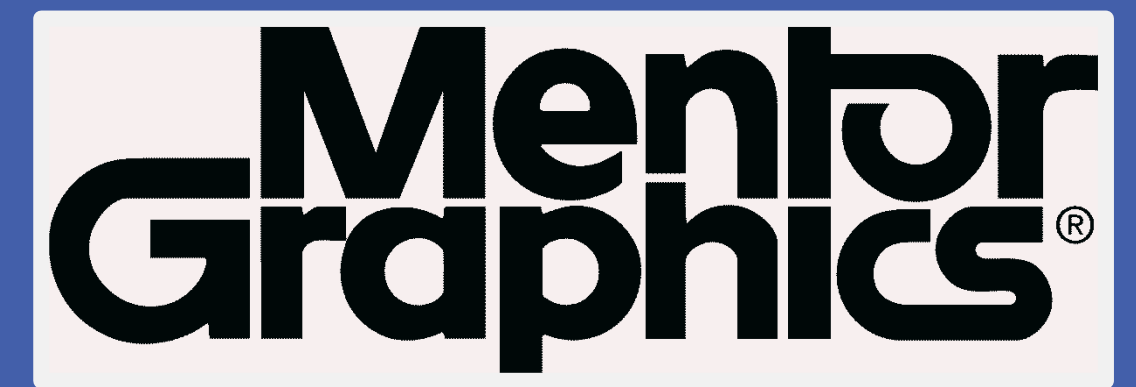


# Using Formal Techniques to Verify SoC Reset Schemes



Kaowen Liu, MediaTek Inc.  
 Penny Yang, MediaTek Inc.  
 Mark Eslinger, Mentor Graphics  
 Jeremy Levitt, Mentor Graphics  
 Matt Berman, Mentor Graphics

Kaowen.Liu@mediatek.com  
 Penny.Yang@mediatek.com  
 Mark\_Eslinger@mentor.com  
 Jeremy\_Levitt@mentor.com  
 Matt\_Berman@mentor.com



## Introduction

Complete verification of SoC initialization is a fundamental requirement and a prerequisite to ensure successful chip operation. At MediaTek we break this down into two components:

- 1) Global Reset Verification
- 2) X-State Detection Verification

At MediaTek we have setup an automated flow for verifying these initialization features using Formal techniques.

## Global Reset Sources

SoC initialization complexly utilizes many sources:

- Power-on Reset
- Hardware Reset
- Software Reset
- Interrupt Driven Reset
- Watchdog Timer Reset

## X-State Sources and X-Optimism

X-State in SoC designs can come from many sources:

- Primary: Uninitialized Registers
- Primary: X assignments in RTL
- Secondary: Bugs in the design due to RTL coding

Example: Out of Bound Array Selecting Generates an X

```
reg [9:0] w_slot;
assign y_data = w_slot[x_select[3:0]];
```

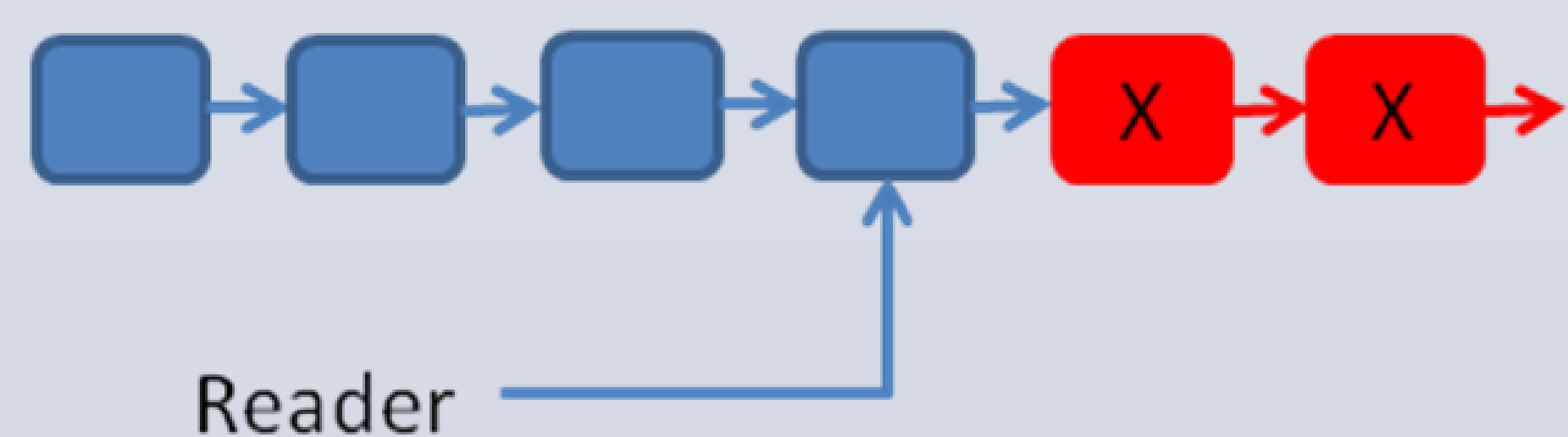
X-Optimism in RTL simulations causes potential bugs to be missed:

- Typically effects if/else and case logic in designs
- Most often cause bugs to be missed during initialization

Resettable Flip-Flop must not output an X-State after the initialization sequence is completed:

- X-States in a design are acceptable as long as the design functions properly
- Uninitialized registers and other X sources should not overwrite a Flip-Flop that has been initialized with a reset

Figure 1: Acceptable X in Shift Register after Reset (Reader gets good data)



## Global Reset Assertion

Users must verify that all resets reach their intended targets with correct polarities.

Example: Generated SVA Assertion

```
global_reset_check_id0: assert property (@(posedge clk)
    `GLOBAL_RESET |-> ##` DELAY !top.u1.rstn );
    • `GLOBAL_RESET is defined by the user: !top.watchdog_arstn
    • `DELAY is the expected delay from the source to destination: 0
```

These assertions are created for all resettable Flip-Flops in the design and the reset functions are verified by Questa Formal.

## X-State Detection Assertion

Users must verify that all Flip-Flops that have been reset aren't overwritten by an X value after initialization:

- All types of reset can be verified (typically asynchronous reset)

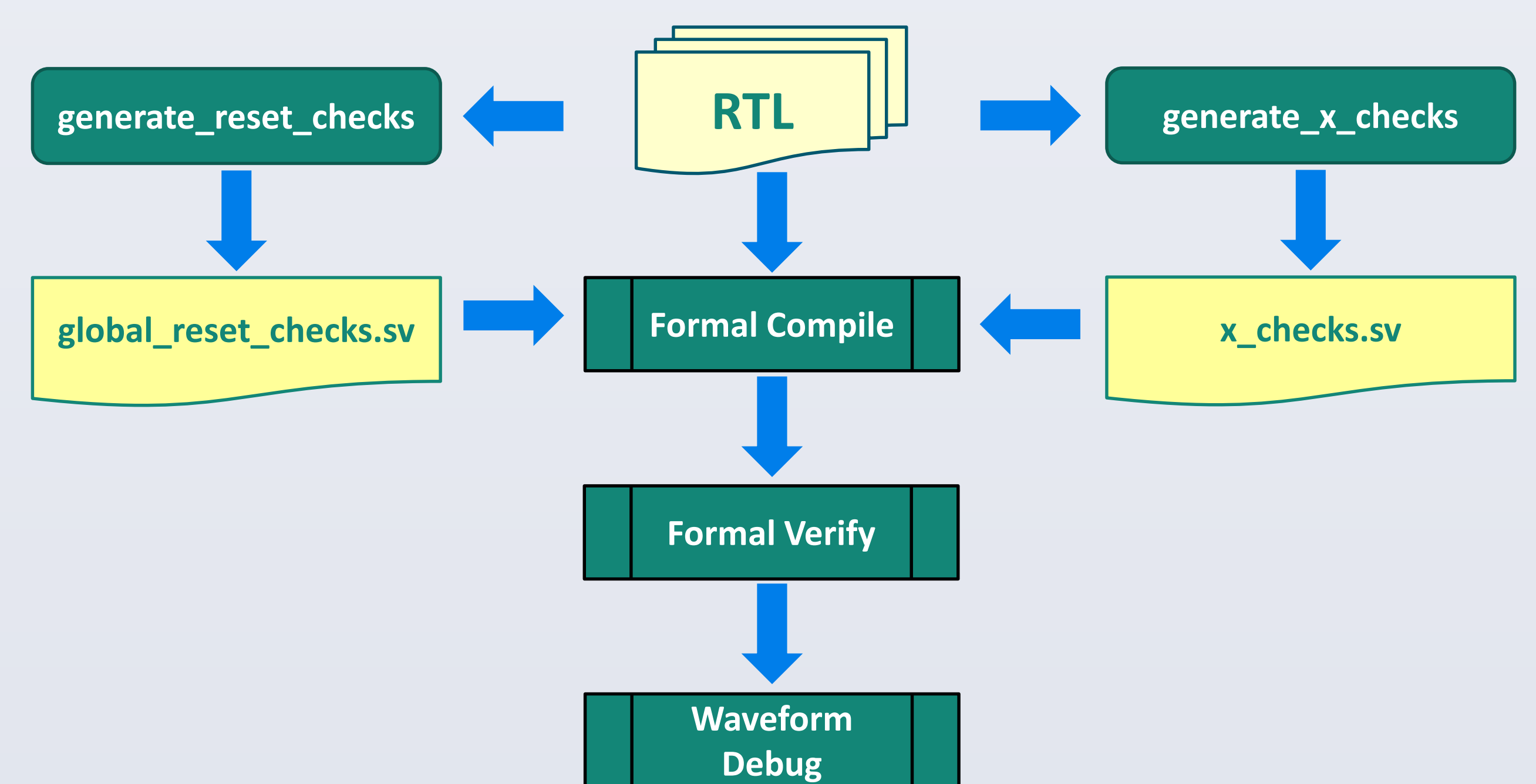
Example: Generated SVA Assertion

```
x_check_id0: assert property (@(posedge clk)
    ^top.u1.my_reg !== 1'bX );
```

These assertions are created for all resettable Flip-Flops in the design and the X-State detection is verified by Questa Formal.

## Formal Verification Flow

Figure 2: Formal Verification Flow used at MediaTek



## Formal Verification Results

Following is a summary of the results for the two types of initialization verification that was performed. The tool used in generating the SVA assertions and formally verifying the results was Questa Formal from Mentor Graphics.

Table 1: Global Reset Check Results

Category	Result
Design Size	3,738,047 register bits
SVA	14,835 assertions
Run Time	gen_sva: 61 min Compile: 6.5 hr Prove: 2.3 hr
Formal Result	Fired: 462 Proven: 14,373 Inconclusive: 0
Bug	248 reset connection errors from 3 modules

Table 2: X-State Detect Check Results

Design Size (reg bit)	SVA	Firing	Bug	Run Time
2,357	319	24	2	1 min
14,136	1,874	38	2	48 min
167,632	19,907	195	6	4 hr 20 min

## Conclusion

- Highly automated Formal techniques avoid the large amount of manual effort required by simulation-based verification to verify global reset schemes and X-state detection
  - Formal techniques are completed within hours without any inconclusive results
- Other automated applications for Formal techniques will be explored to reduce manual effort and achieve higher verification confidence