Truth Beneath the trace : Formal Revealing Silicon Secrets

Chepuri Venkatesh, Engineer, Senior
Mahesh Shinde, Lead Engineer, Senior
Satish Singh, Engineer, Senior Staff/Manager
Nitin Neralkar, Principal/Manager

Problem Statement

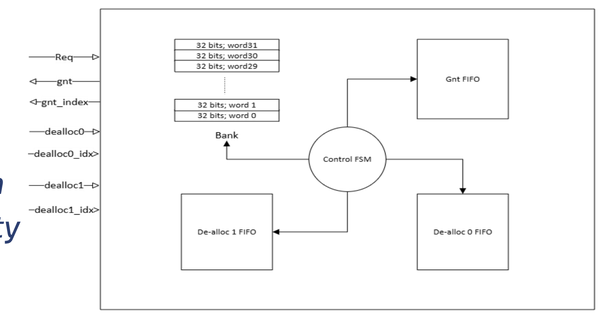
- *Root cause analysis is one of the most challenging aspects of post-silicon validation*
- *Triaging post-silicon bugs is complex, as symptoms often appear in one module while the root cause lies in another*
- *During post-silicon validation of a high-speed subsystem, SVE found two issues*
  - ***While processing concurrent large data, sub-system is getting hang***
  - ***Non-zero live resource count at the end of the test, suggesting resource leakage***
- *The sub-system process high speed data operations with upto 4K command ID trackers*
- *Resource IDs are managed by a HW managed Freelist module*
- *Requires millions of simulation cycles, demands long hours to root cause po-si issue*
- *Critical for maintaining project timelines due to high visibility and faster product release*

Proposed Methodology

- *Design suspected the resource ID trackers called Freelist as the potential blocks causing these post silicon issues*
- *Formal verification (FV) is a highly effective and preferred method for thoroughly validating intricate hardware designs, especially at module level*
- *Using what-if analysis & assertions, formal can expedite the root cause and triaging of post silicon issues*
- *Formal will exhaustively verify the fixes providing higher confidence to the designers*

Freelist Introduction

- *Grants unique IDs for new requests*
- *Uses depth-first search for ID allocation*
- *Search happens at word level granularity*
- *All the operations are handled by FSM*



Freelist block diagram

Post Silicon Issue 1: Sub-system deadlock

- *A Freelist level hang occurs when it fails to grant ID's despite having Freelist*

```
logic [CT_FREELIST_LUT_MEM_DEPTH-1:0] valid_locations;

always @(posedge Clk, posedge Reset) begin
    if (Reset) begin
        valid_locations <= 0;
    end

    else begin
        if (lut_alloc_gnt) begin
            valid_locations[lut_index] <= 1;
        end

        if (lut0_rddealloc_vld) begin
            valid_locations[lut0_rddealloc_index] <= 0;
        end

        if (lut1_rddealloc_vld) begin
            valid_locations[lut1_rddealloc_index] <= 0;
        end
    end
end
```

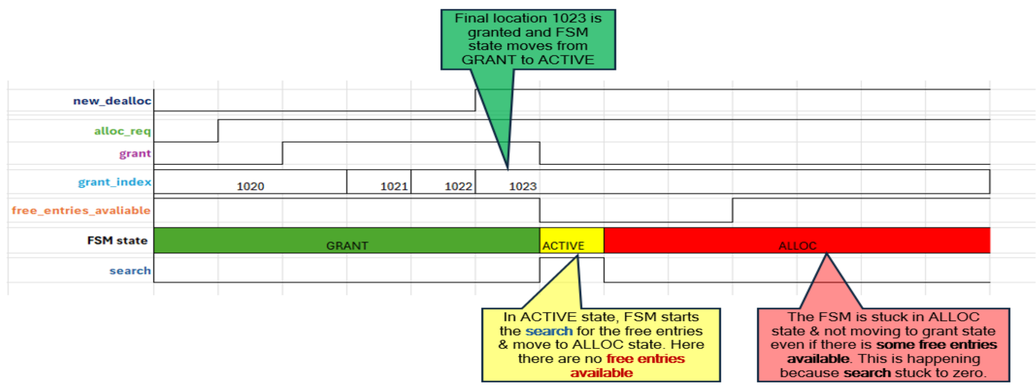
assert property (alloc_req & ($countones(~valid_locations) >= 1) |-> ##[0:6] alloc_gnt)

- *Created over-constraints to reach the higher entry depths*

assume property (~alloc_req |-> ##[1:5] alloc_req[*10])
assume property (de_alloc |=> ~de_alloc[*10])

- *Convergence time reduced from 12 hrs to ~30 mins*

Post Silicon Issue 1: Deadlock/hang

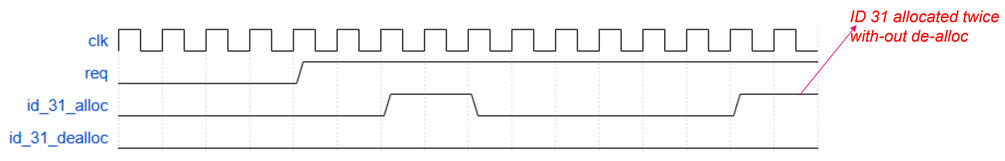


Hang issue bug description

- *Validated RTL fix & found a hidden starvation issue using the same assertion*
- ***Starvation Issue** : Freelist is waiting for new de-allocations to grant new requests despite free locations available*

Post Silicon Issue 2: Resource leakage

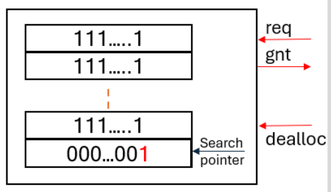
- *Live Counter values of Freelist should be equal to number of zeroes in the bank*
- *What-if cover:* ***Cover property (Live_count - bank_count == 3)***
- *Cover is proven & debug shows an ID allocation twice with out de-allocation*



Duplicate ID 31 allocated twice

- *Corner case happens only when search pointer is at the last location of a word, new request and de-allocation ID of same word comes in the same cycle*

assert property (alloc_gnt |-> ~valid_locations[gnt_index])



- *The assertions resulted in counter examples for three iterations to achieve correct functionality*

Results & Conclusion

- *Formal Verification was instrumental in*
  - ***efficiently triaging and root causing post-silicon bugs***
  - ***enabling systematic framework for complex SoC issues***
- *Formal methods enabled fast post-silicon debug, completing resolution in just one week*
- *Targeted what-if accelerated the root causing post silicon issues*
- *FV assertions accelerated RTL fix validation and uncovered additional hidden bugs*
- *Traditional simulation struggles to catch subsystem hangs from bulk 64B data transfers*
- *Issues like duplicate de-allocation are hard to catch at sub-system level DV*
- *All post-silicon control logic issues should be reproduced within the formal verification environment to mitigate risks*
- *Bug fixes must be formally validated to ensure reliability before design re-spin*

Qualcomm

Engineering Human Progress