2023
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

FSM Minesweeper – Scalable FV Methodology for Detecting Hangs in Interacting FSMs

Anshul Jain, Achutha KiranKumar V M, Harbaksh Gupta, Shashwat Singh

Intel Corporation

accellera
SYSTEMS INITIATIVE

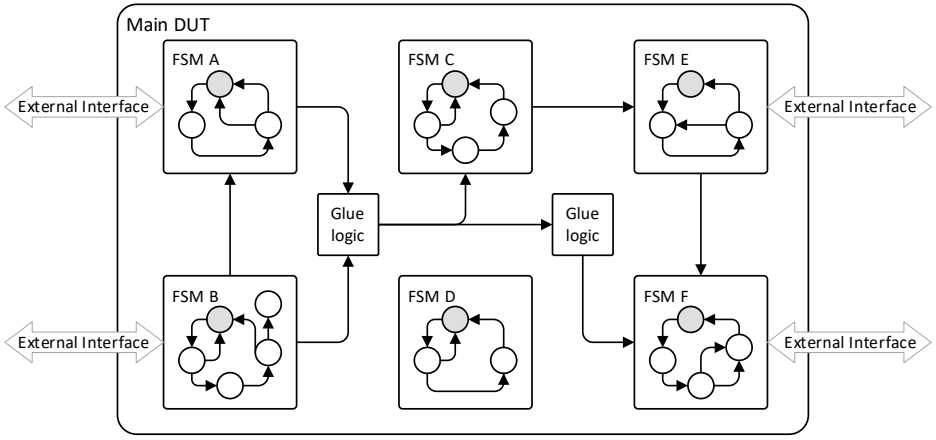
About Us – FVCTO @Intel

- Formal Verification Central Tech Office (FVCTO)
  - 100+ formal verification engineers/experts/pioneers
  - Applying FV across Intel Leadership Products throughout design cycle

- Today, we present the application of FV for security verification
  - In post-silicon phase (of a server CPU)
  - In pre-silicon phase (of a client CPU)

Key Message

- Formal Property Verification is a powerful technique for exposing hangs in designs due to “**Interacting FSMs**”



- Traditional formal techniques fails to scale on large designs due to tool capacity issues and the burden of deep design knowledge

- In this talk, we explain **FSM Minesweeper** – a straightforward method to overcome the barriers of capacity and detailed design know-how

Agenda

- Overview
- Problem Statement
- Methodology
- Case Studies
- Conclusions

Overview

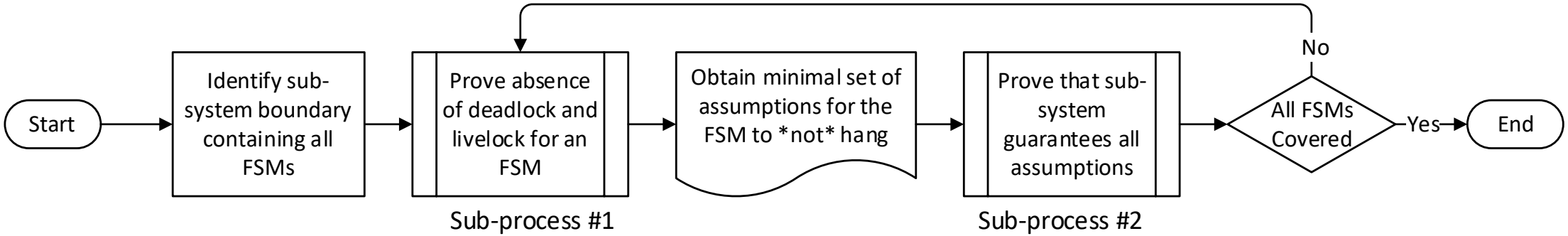
- Hangs have traditionally evaded simulations
  - Cause panic in late stages of IP/SOC design signoff
  - Generally rooted in complex control logic of designs
  - Need specifically timed, unlikely sequence of events to be discovered
- FV capable of finding hangs effectively using its breadth-first search
- FSM minesweeper – Efficient FV methodology for catching hangs
  - Focus on identifying all deadlocks & livelocks in a system of interacting FSMs

Problem Statement

- Modern SOCs & IPs are getting increasingly complex
  - More features → Higher complexity
- Corner-case coverage in simulations is predictably low
  - Especially in case of interacting FSMs
- Deep sequence of events required to warm-up all the states of FSMs
  - Hard to reach a cross-combination of FSM states which is buggy
- Traditional FV is incapable of solving this problem
  - Impractical to apply FV on entire IP
  - Capacity issues and convergence challenges are evident

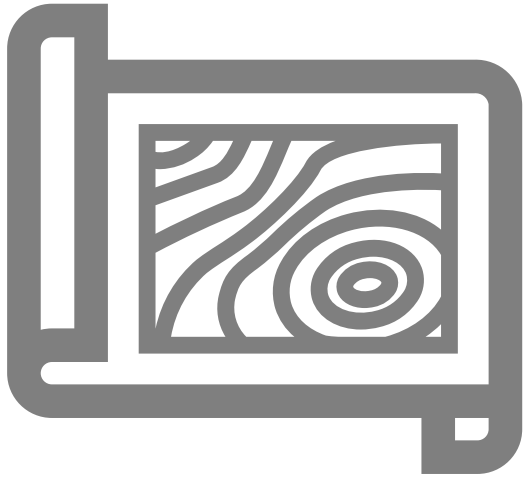
FSM Minesweeper – Overall Process

- FSM Minesweeper is a novel & straightforward solution for mining bugs in system of interacting FSMs
- Eminently scalable to large IPs and capable of managing formal complexity barriers using the process explained below



Step A

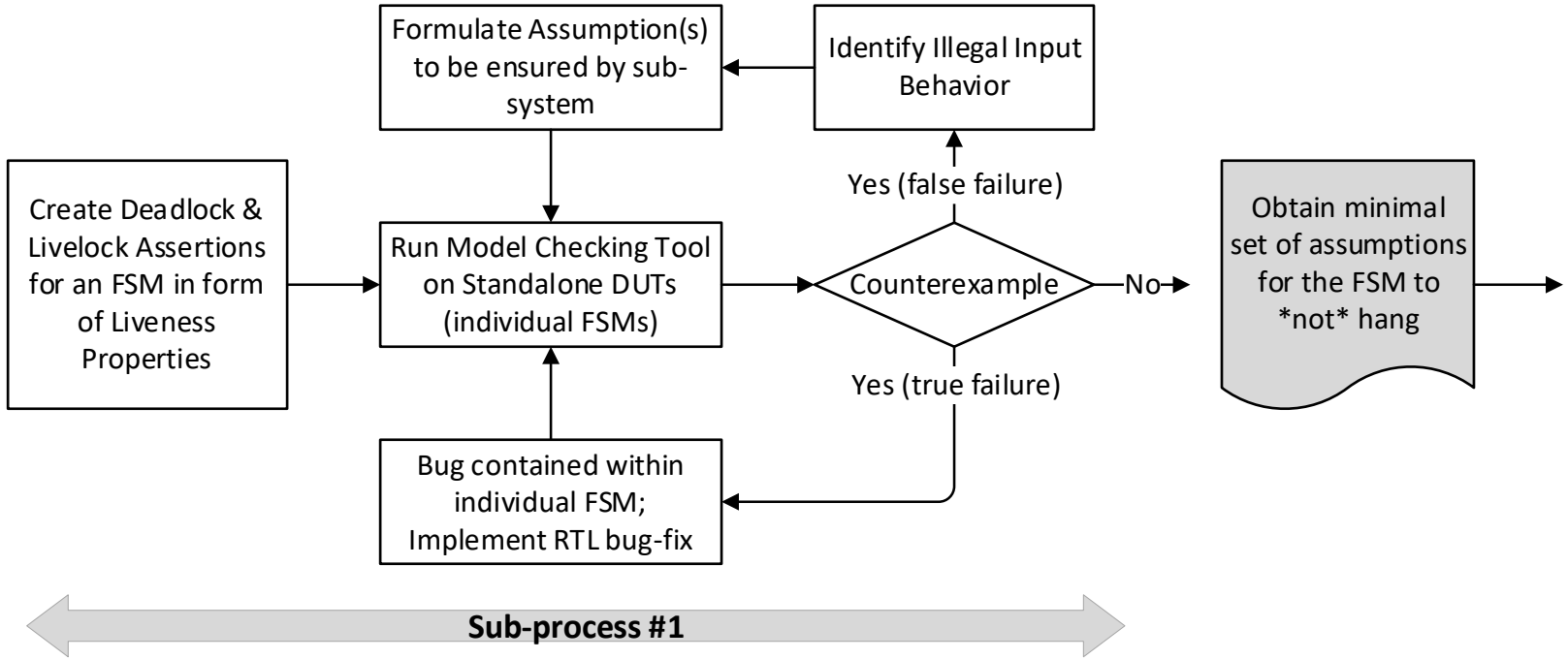
Identify design boundary containing interacting FSMs i.e., DUT



- Require FSMs and glue logic facilitating interaction b/w FSMs
- Entire RTL implementation is irrelevant (unnecessary complexity)
- Identify all the FSMs of your design (major industrial tools capable)
- Pick the module in design hierarchy instantiating all FSMs
- Black-box modules known to be independent of FSMs

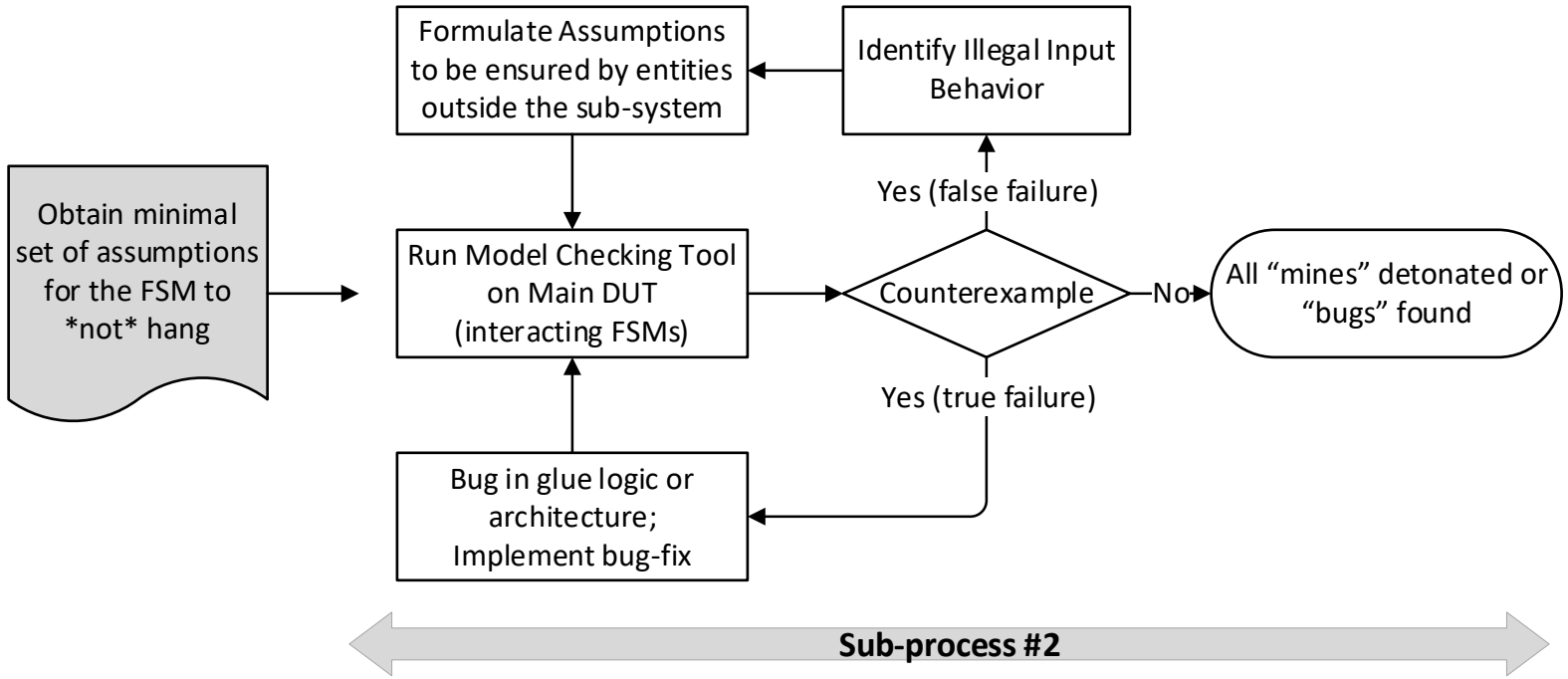
Step B

Prove absence of deadlock & livelock for individual FSMs



Step C

Prove that assumptions made by individual FSM is guaranteed by DUT



2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

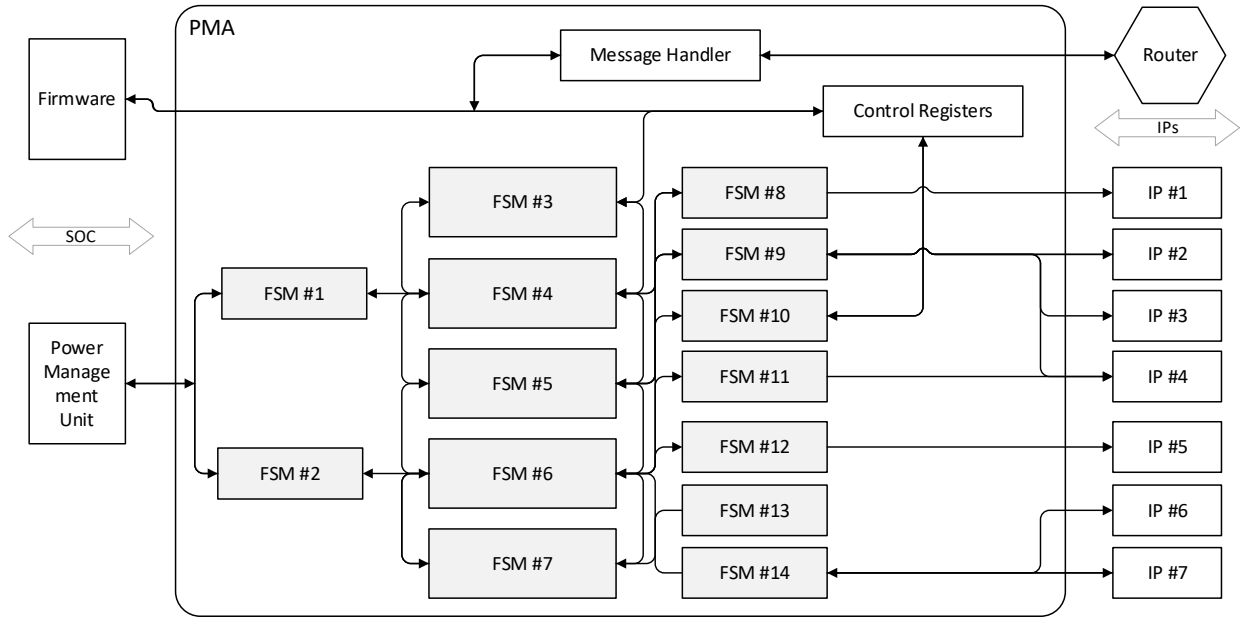
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Case Studies

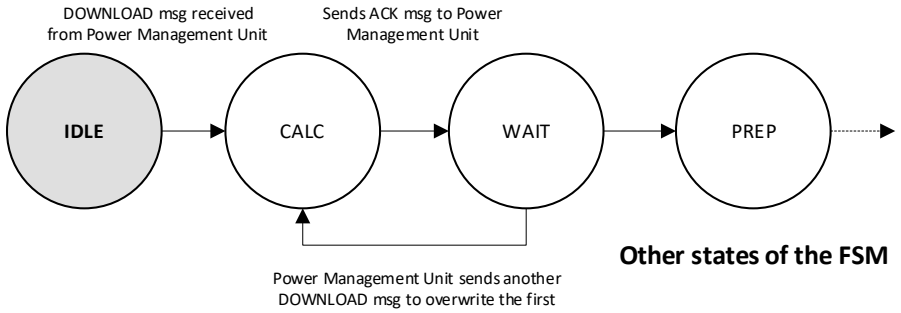
accellera SYSTEMS INITIATIVE

Case Study 1: Client CPU Memory Subsystem

- Memory Subsystem contains PMA (Power Management Agent)
- PMA integrates multiple IPs (like memory controller, inband ECC, fabric interface)
- PMA handles interactions between IPs for reset and power management
- PMA houses 14 interacting FSMs

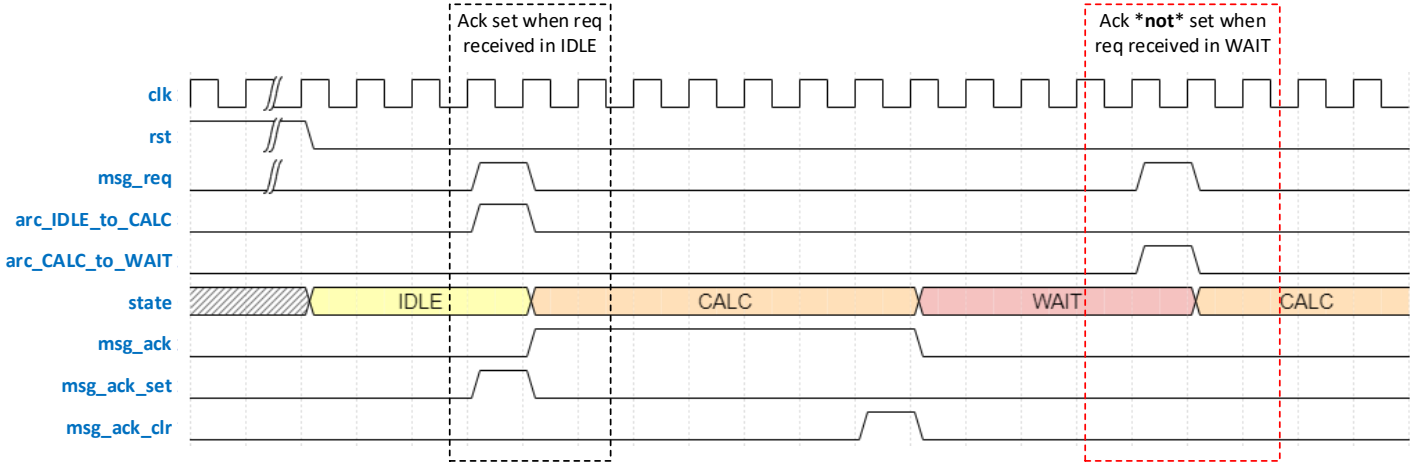


Case Study 1: PMA Hang



Failing Deadlock Assertion

```
(state == CALC) |-> s_eventually (state != CALC)
```

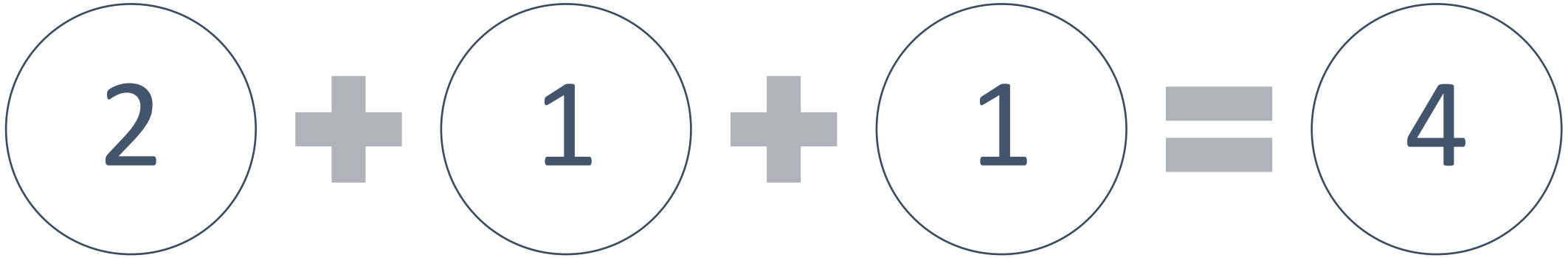


Buggy RTL Implementation

```
arc_IDLE_to_CALC = (state == IDLE) & msg_req_rise;
arc_WAIT_to_CALC = (state == WAIT) & msg_req_rise;
msg_ack_set = (arc_IDLE_to_CALC);

always_ff @(posedge clk) begin
   if (msg_ack_clr) msg_ack <= '0;
   else if (msg_ack_set) msg_ack <= '1;
end
```

Case Study 1: PMA Results



Bugs in Individual FSM

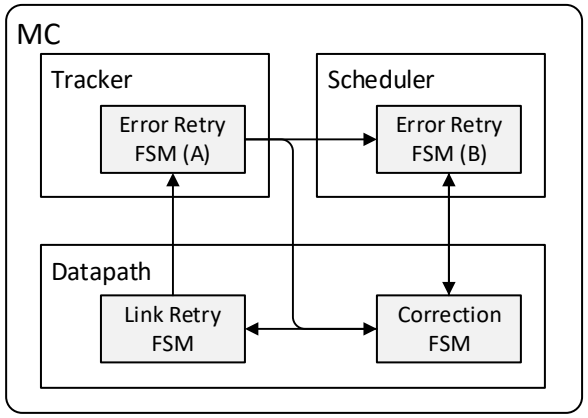
Bug in glue logic b/w FSMs

Bug in architecture of FSMs

Bugs

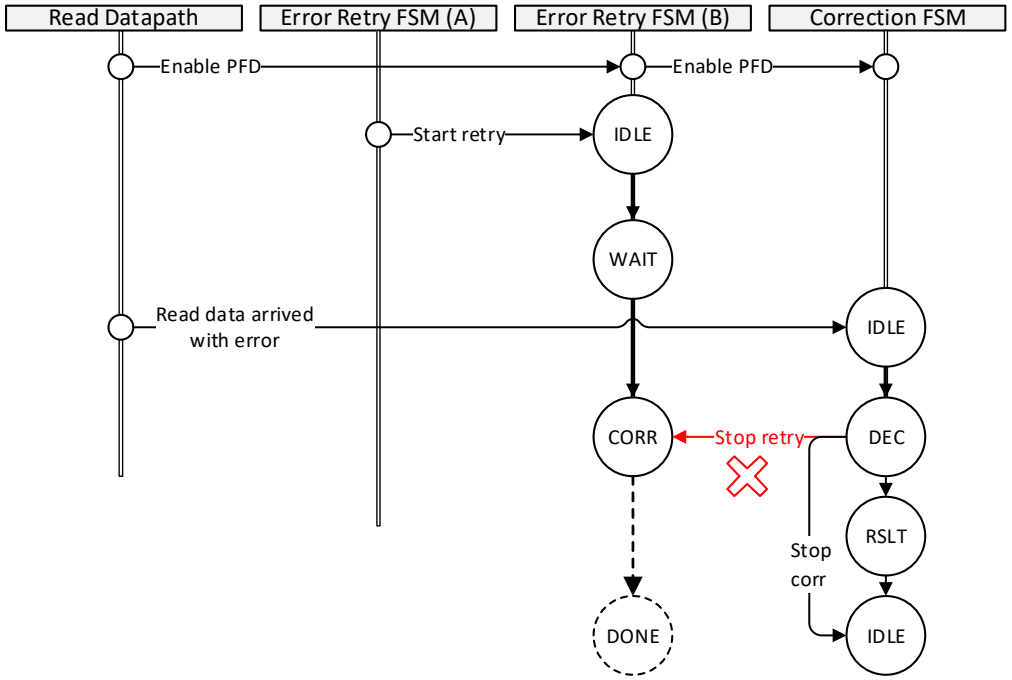
Case Study 2: Server CPU Memory Controller

- MC contains 4 interacting FSMs for implementing multiple error flows
- These FSMs work in tandem in various modes for correcting different types of errors



Case Study 2: MC Hang

- Architectural hang found in DV
  - Error retry FSM (B) and correction FSM went out-of-sync in persistent fault detection (PFD) mode when an uncorrectable error is encountered
- FSM Minesweeper reproduced the architectural hang
  - Successfully completed proof-of-concept
  - Proved the robustness of the bug fix

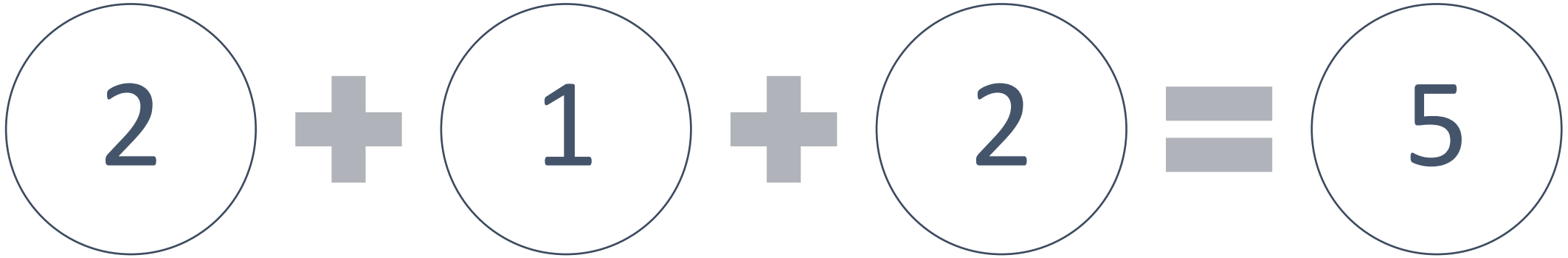


Buggy RTL Implementation:

```
stop_corr = corr_err_pfd_en & // Persistent fault detection enabled
   ~corr_err_req_uc &         // Do not correct uncorrectable error
   ~corr_err_skip;            // Skip correction when interrupted

stop_retry_set = corr_fsm_exit_dec & stop_corr
```

Case Study 1: MC Results



Bugs in Individual FSM

Bug in glue logic b/w FSMs

Bug in architecture of FSMs

Bugs

Conclusions

- Proving absence of hangs at IP-level is critical for today's DV teams
- Hangs are not well-addressed by traditional verification methods including end-to-end FV
- FSM minesweeper is a targeted application of FV
  - Applicable early on at IP-level RTL to guarantee individual FSM stability
  - Identify bugs in glue logic and
  - Find architectural flaws in FSMs interactions
- FSM Minesweeper is partly automated – moving to full automation for wider adoption

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Questions?

Thank you!

accellera SYSTEMS INITIATIVE