



Accelerate Verification, Streamline Challenges: A Comprehensive HBM Model Solution

Dharini SubashChandran, Ritesh Desai, Ujash Poshiya, Vatsal Patel



Verification Challenges

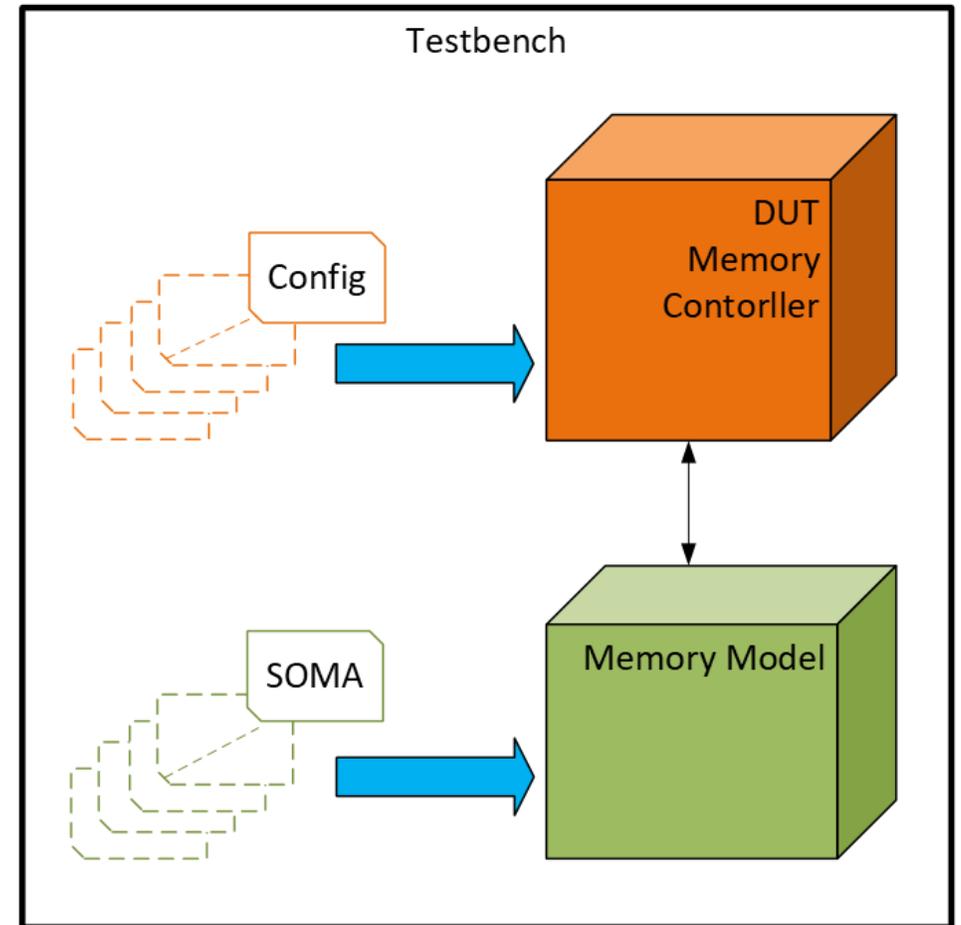
HBM is a pivotal piece of technology for AI training and inference

- AI models expanding in complexity, need efficient verification of memory devices
- IEEE 1500 port as integral part of HBM for testing and repair
- Crucial points for verification:
 - Refresh requirements changes due to temperature fluctuations
 - Multiple vendor devices with different configurations
 - Memory read data randomization
 - Validating high-speed data integrity, like parity, ECC, etc.

Multiple Memory Configurations (SOMA)

SOMA (Specification Of My Architecture)

- Memory Device variants are many!!!
 - Design configurations are based on:
 - 📦 Densities
 - 🚀 Speeds
 - 📐 Data Widths
 - 🏢 Vendors
- Poses challenges:
 - 🧠 Complexity
 - 💾 Storage
 - 📁 Multiple files/configs
- Traditionally, with multiple config files:
 - 🔧 Manual setups cause inconsistencies
 - 🐢⌚ Delays in verification



One Memory Configuration (SOMA)

- Unified HBM Configuration **1**

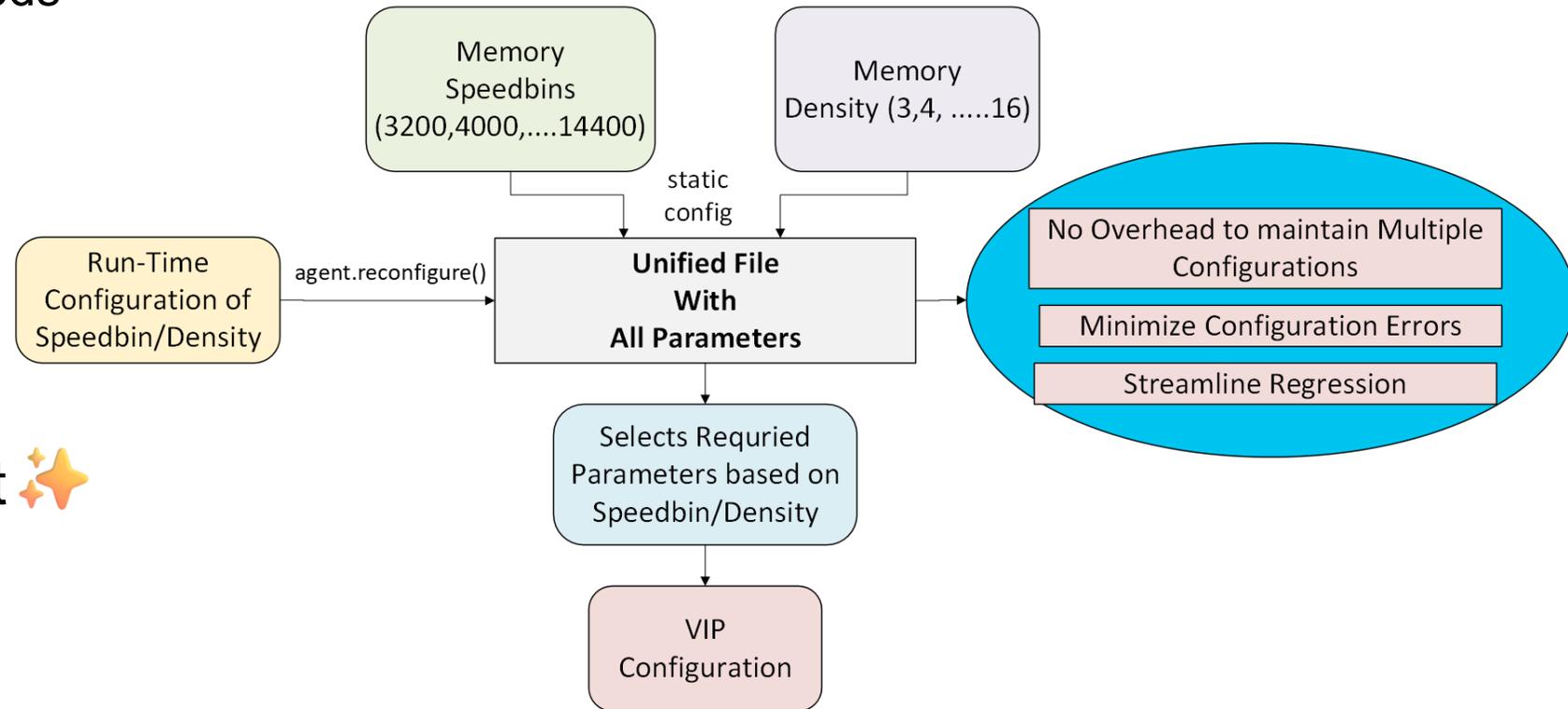
- Manage HBM densities, speeds with single config class

- Accelerated Integration **▶▶**

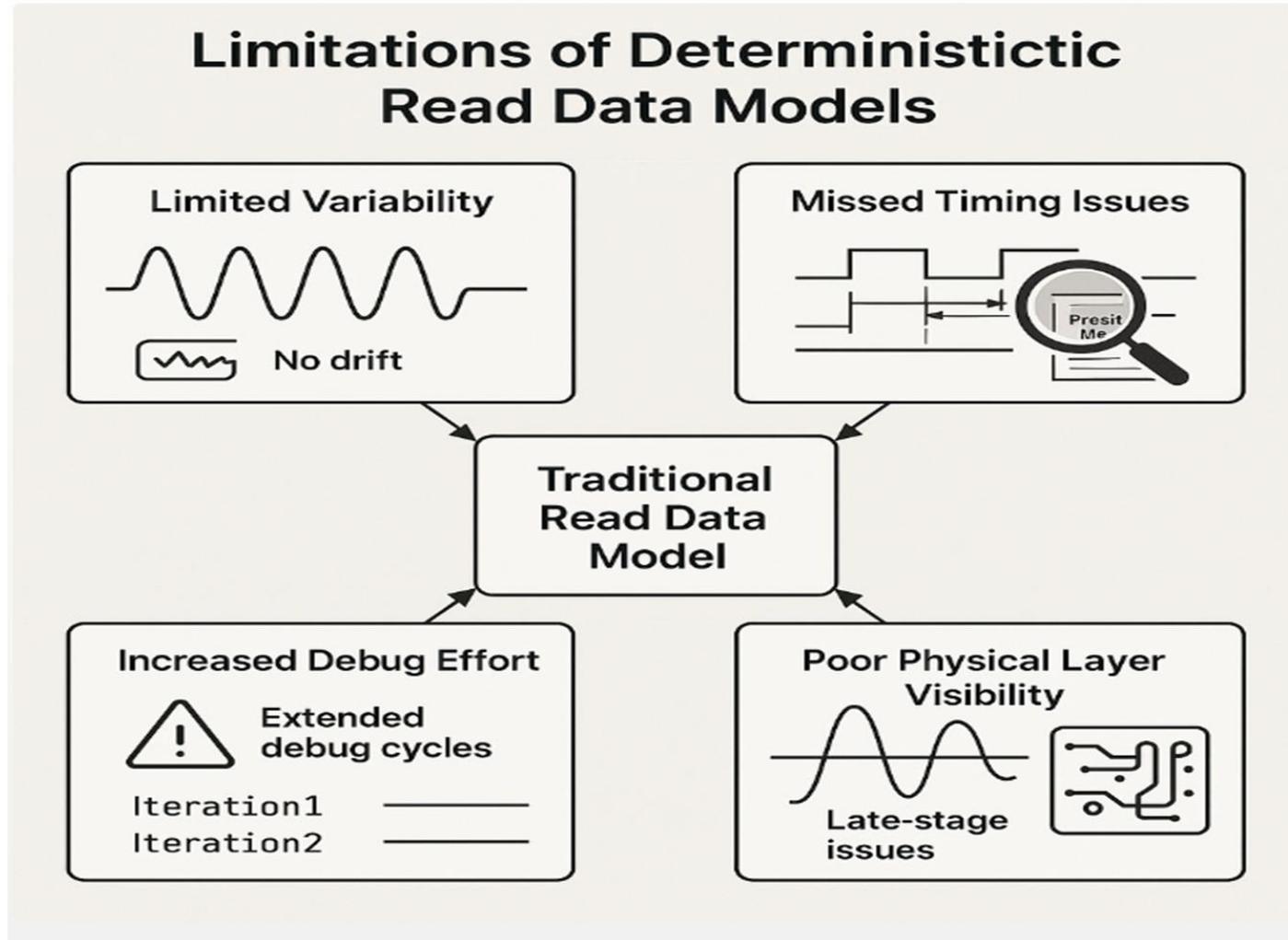
- Quickly adapt to new HBM variants and vendors

- Simplified File Management **✨**

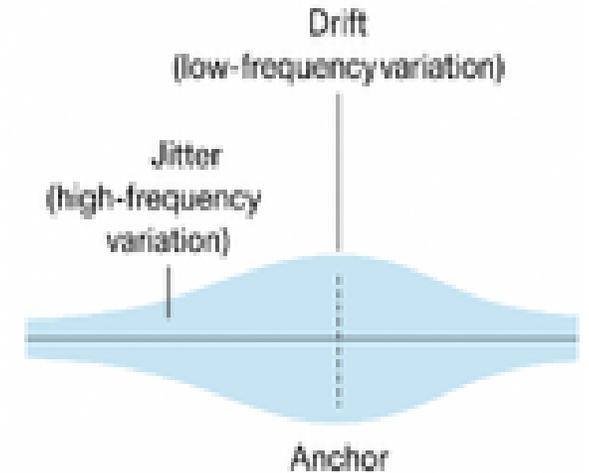
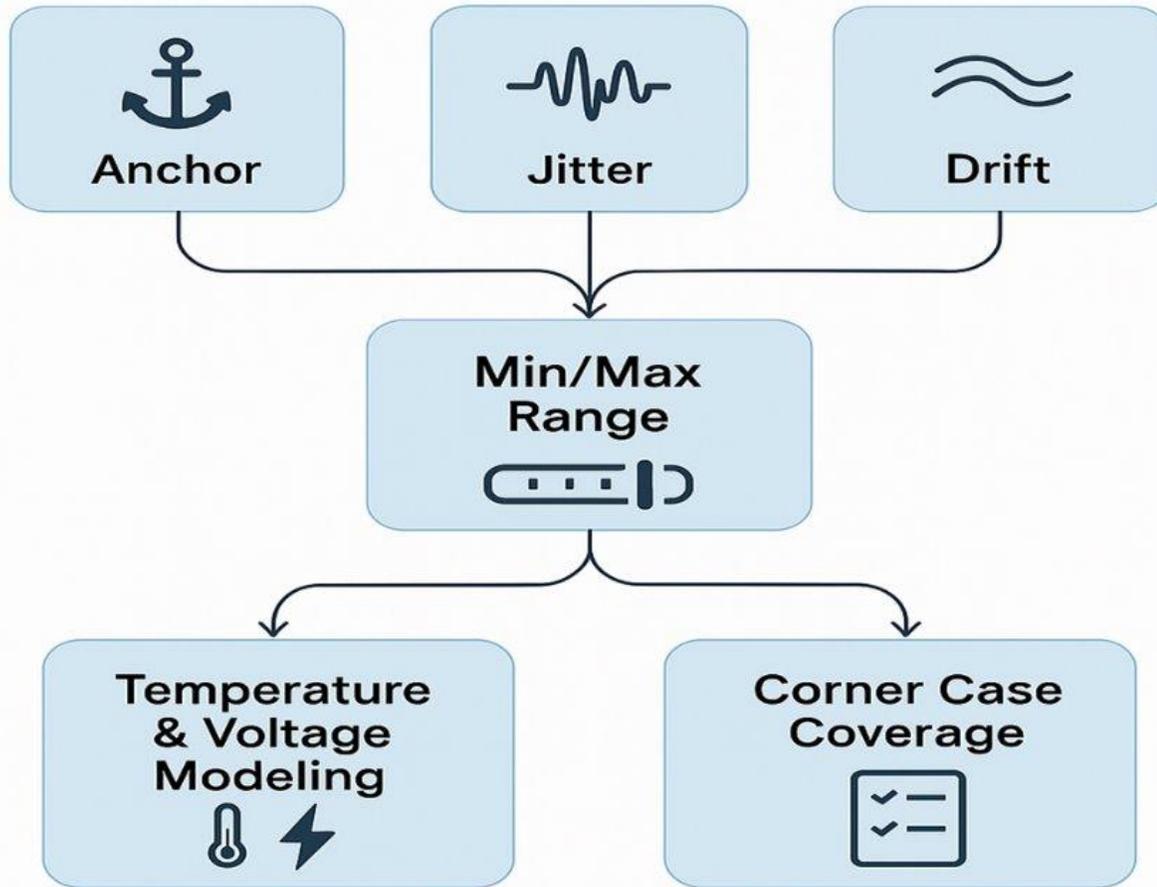
- Reduce file management overhead



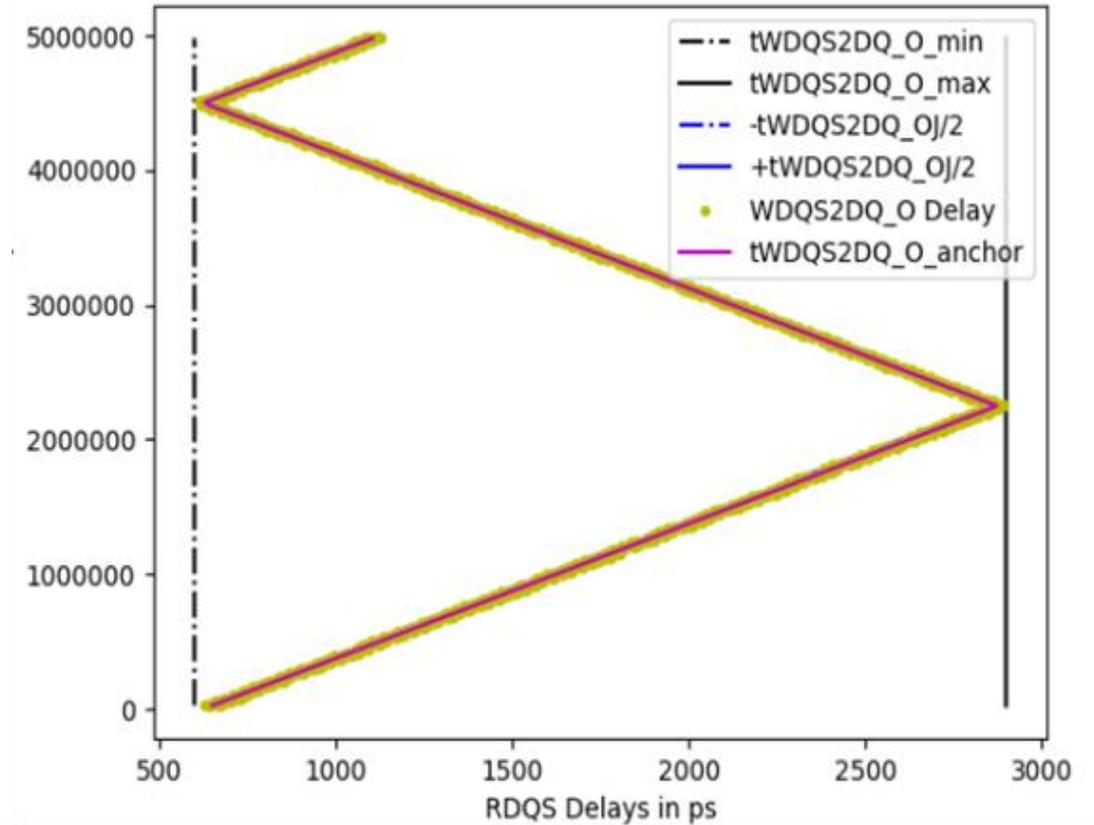
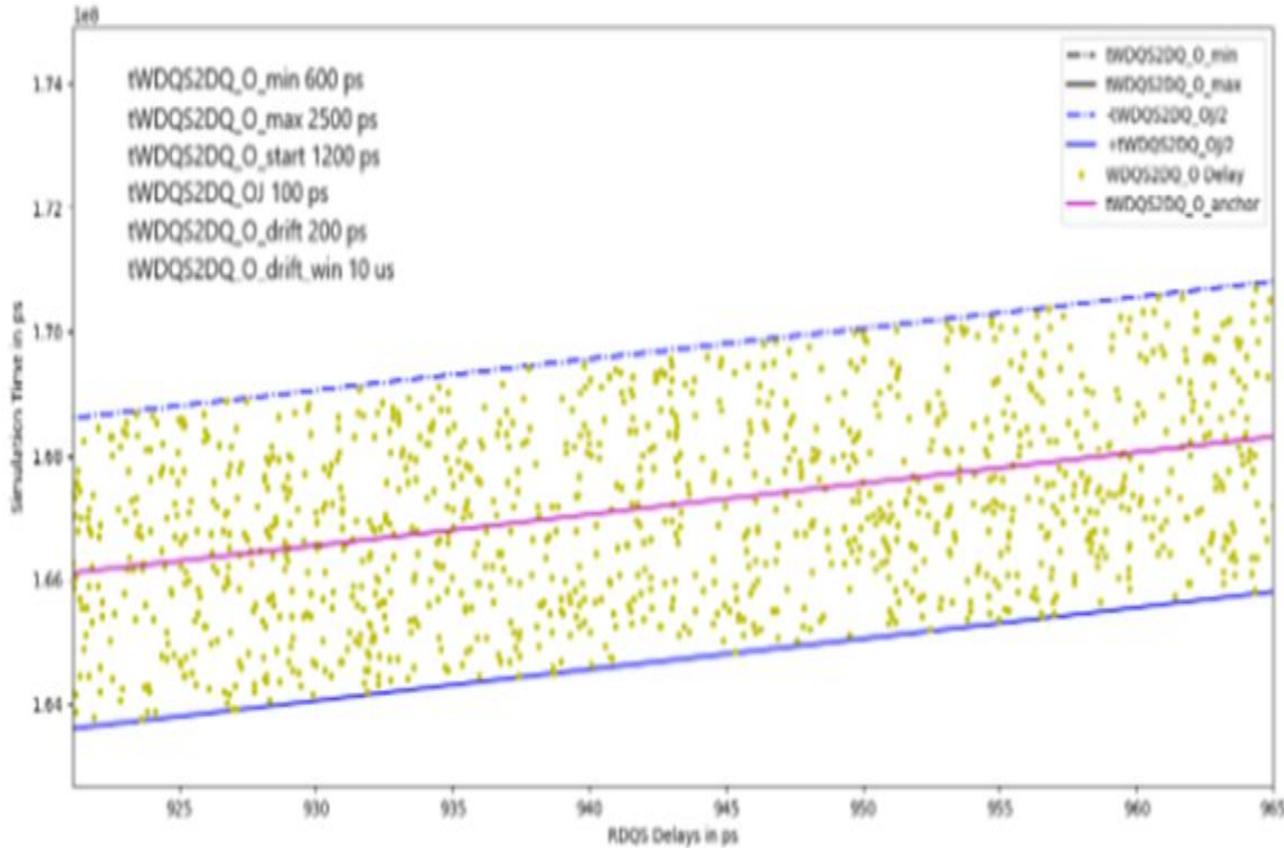
Sophisticated Read Data Randomization



Anchor, Jitter, Drift: The Foundation of Timing Randomization

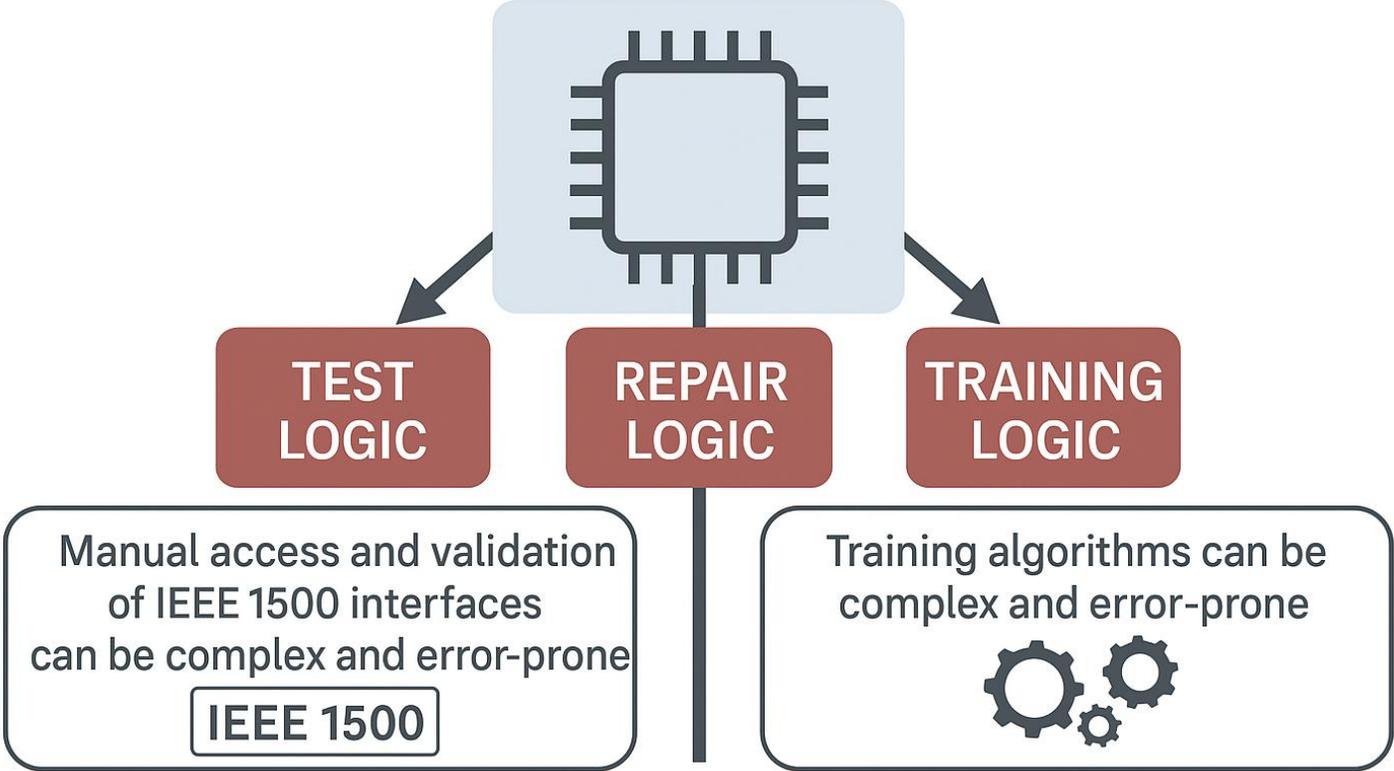


Actual Simulation Randomized Data

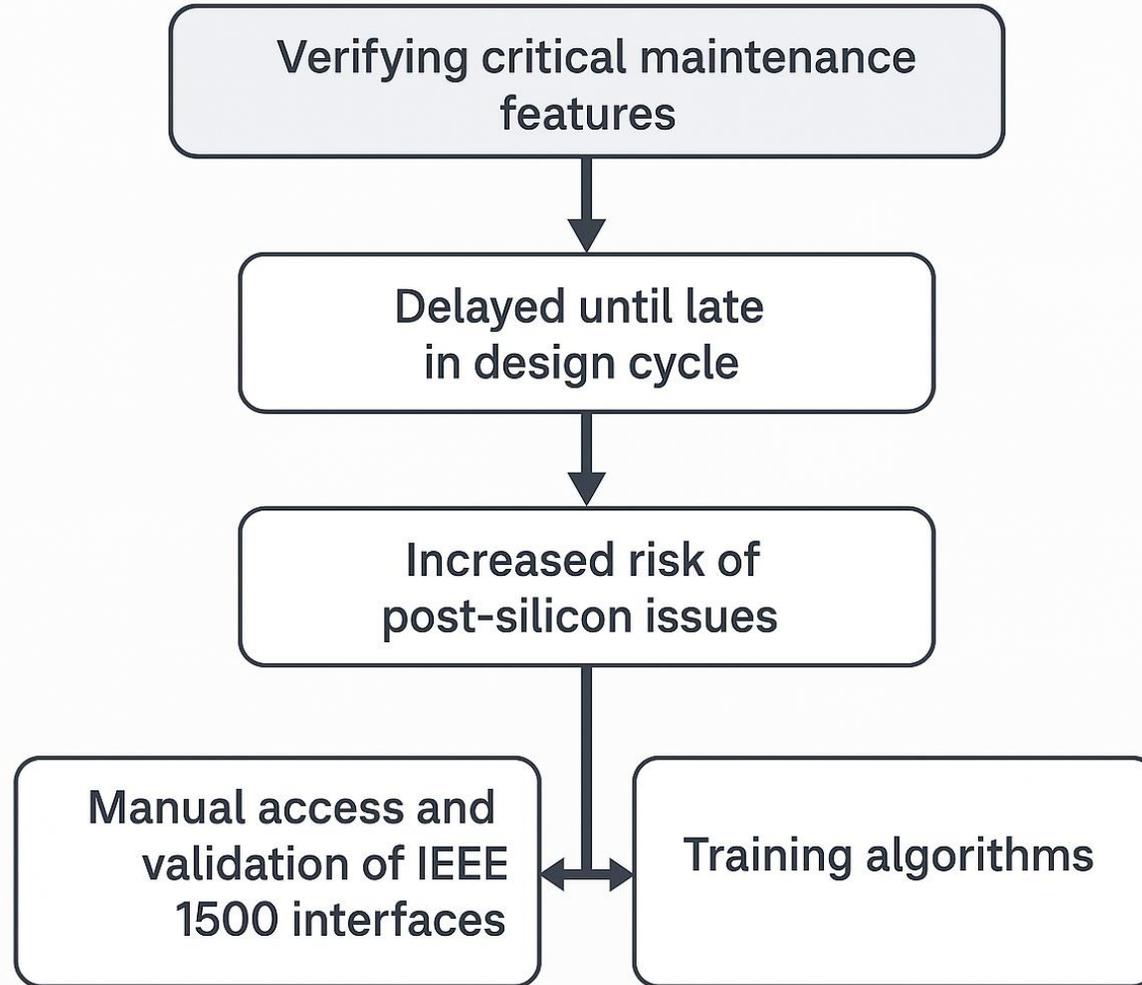


Integrated IEEE Backdoor Testing

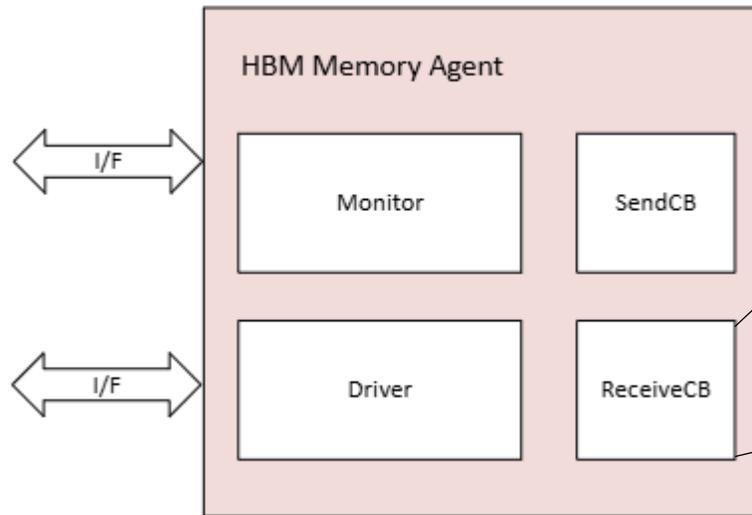
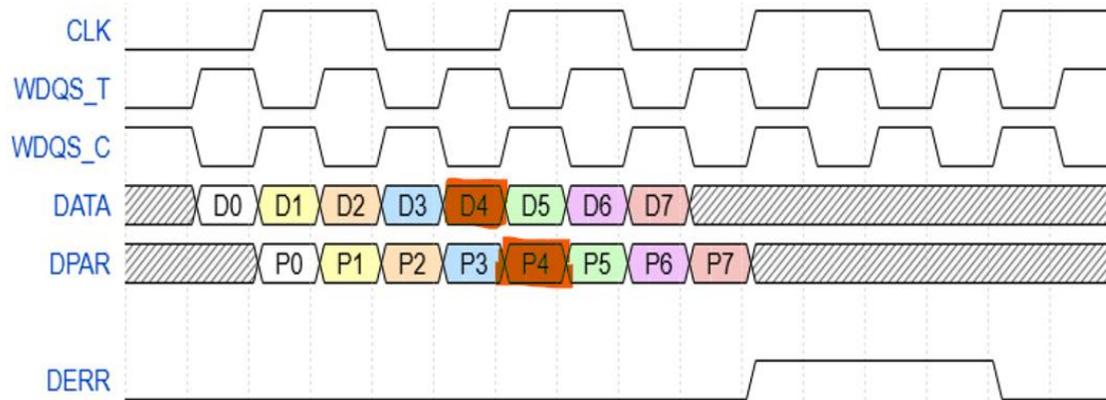
Verifying critical maintenance features like test, repair, and training logic is often delayed until late in the design cycle, increasing the risk of costly post-silicon issues.



Advantages



Error Injection



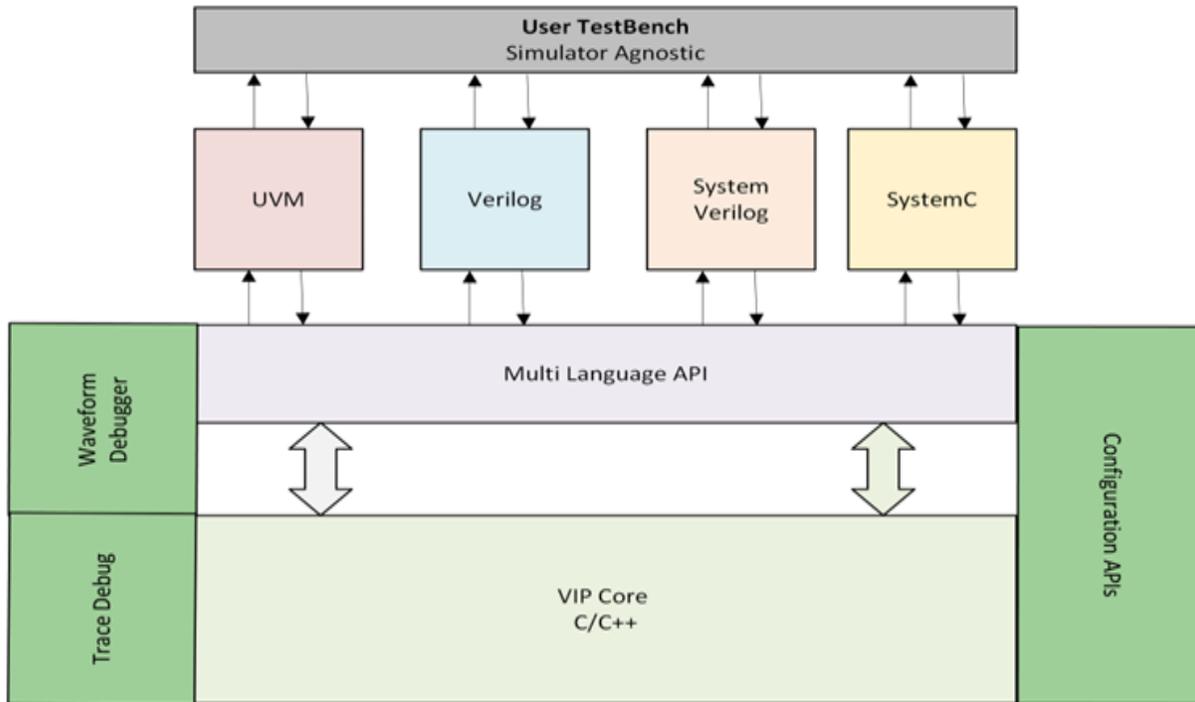
```
virtual function void ReceiveCbF (denaliHbmTransaction
trans);
  if ((trans.Type == DENALI_HBM_TR_Write) &&
(trans.PacketType == DENALI_HBM_PACKET_TYPE_Data))
  begin
    trans.Data[4] += 1;
    void'(trans.transSet());
  end
endfunction : ReceiveCbF
```

Debuggability: Real Time HBM display



- Comprehensive Debug Ports
- Real-Time Transaction Display
- Accelerated Root Cause Analysis

Methodology Agnostic



- No Methodology Lock-In
 - Use preferred verification framework (SV/UVM/SystemC) without limitations
- Streamlined Adoption
 - Easily integrate the solution
- Consistent Verification
 - Ensure consistent and efficient verification of any methodology

Questions

