ITC 2023 PO.26: Memory test automation using Shared Bus interface aids with turnaround time

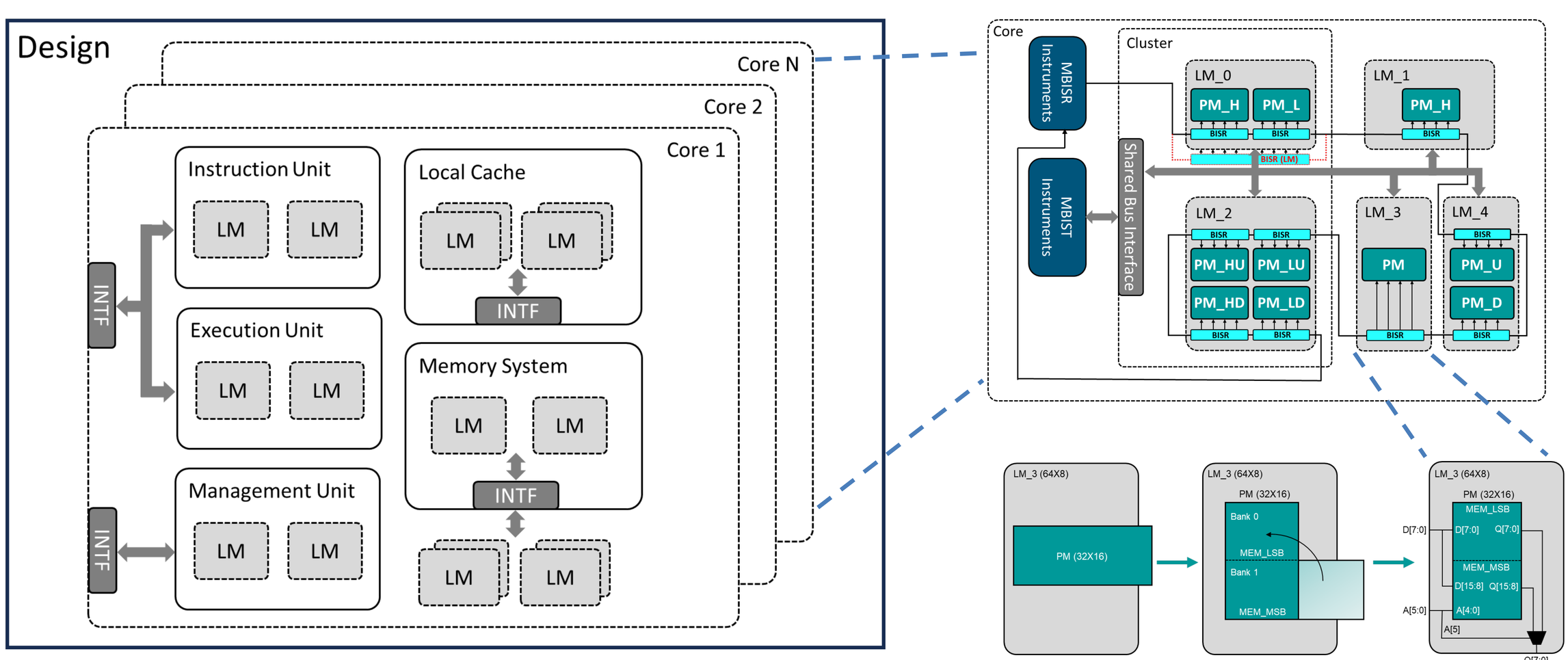
Prashant Seetharaman Albert Au Sebastian Bromberek Harshitha Kodali

SIEMENS

Introduction

Shared Bus technology is a well-known standard used in memory test introduced over a decade ago by commercial IP core vendors in cooperation with EDA companies. Leading-edge designs complicate the DFT modelling to apply MBIST through these access points. This poster presents how EDA can meet this challenge, facilitate accurate MBIST setup of the design with Shared Bus interface and consequently reduce the turnaround time.

Challenges of modern Shared Bus designs

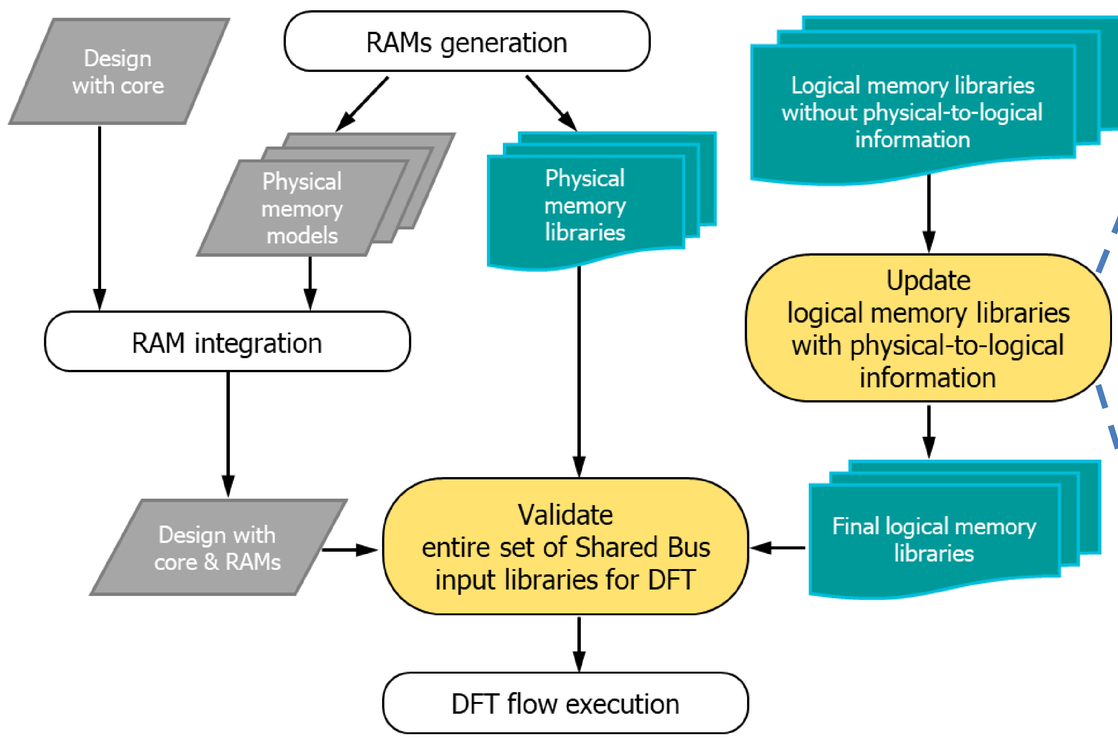


Leading-edge design contains numerous processing cores. MBIST is applied through Shared Bus interfaces (INTF). IP core vendor defines the logical memories (LM), which impose address space and data bus width. User chooses and integrates physical memories (PM) into each LM, and prepares the MBIST models.

Challenges of RAM integration and modeling process

- Memory repair and diagnosis is now common in the Shared Bus context.
- Complex mappings between logical and physical layers, different from conventional horizontal and vertical stacking of physical memories.
- Logical arrays composed of many physical memories (up to 64 in some of the designs).
- Multi-port memories behind Shared Bus interface with possible extra levels of multiplexing.
- Multiple cluster interfaces and multiple cluster instantiations in a single design.

RTL tracing-based automation for memory test setup through Shared Bus



Strategy for extracting and validating physical-to-logical structure

1. Prepare flatten design data model
2. Activate & confirm MBIST mode with vendor-specific protocol
3. **foreach (LM select codes) do** // From IP core documentation
4. Apply LM's select code to Shared Bus interface
5. Trace to determine signals controlling vertical stack of PMs
6. **foreach (PM activation signals) do**
7. Apply PM's activation signals to Shared Bus interface
8. Collect paths to data input/output, address, bit WE, etc.
9. **foreach (Collected paths) do**
10. Simulate read or write access
11. Trace connections and pipelines to/from PM
12. **end**
13. **end**
14. **end**

Conclusion

Shared Bus architectures are evolving to become more important and complex. Therefore, automation of the entire memory test setup through Shared Bus is crucial not only from the reliability point of view, but also from the turnaround time reduction perspective. The presented RTL learning approach was successfully applied in many examples helping to significantly automate DFT tasks.