Design for Verification (DFV) of System-on-Chips (SoCs)/ Chiplets

Sankaran Menon, Ph.D.
Intel Corp.

JULY 9–13, 2023
MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Outline

- Introduction to DFV
- DFX Features that are already main-stream
- Verification/Validation Challenges
  - Pre-Silicon Verification
  - Post-Silicon Validation
  - Manufacturing Validation
- Design for Verification (DFV) Requirements
- Achieving DFV Capabilities
- Next Steps and Future Work

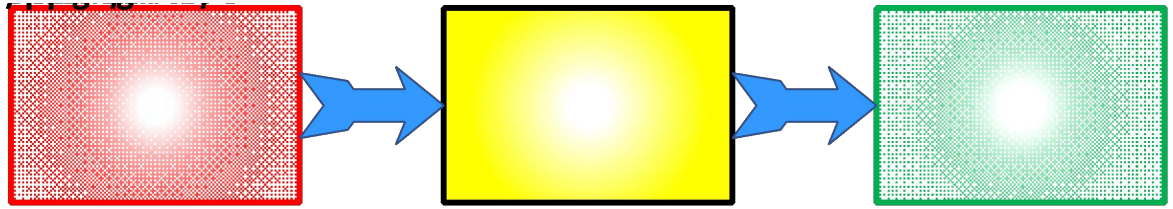
DFX Features

- DFX Features that are already main-stream
  - Design for Test (DFT)
    - Scan Test, Memory BIST, IO DFT etc
    - Structural tests are used to achieve high-coverage using lower test-times
  - Design for Debug (DFD)
    - Use of existing DFT features to achieve Debug
    - With some enhanced features added to achieve improved Debug for faster TTM
  - Design for Manufacture (DFM)
    - Features to ease Manufacturing
  - Design for Yield (DFY)
    - Features added in the design to improve Product Yield
  - Design for Verification (DFV)
    - Features to assist Verification

- A Few other Alphabet soup besides above constitutes DFX
  - This presentation will cover DFV for SoCs

Verification/Validation Phases

- Typical development of an SoC/Chiplet consists of three phases
  - Architecture/Specification
  - Design/Implementation
  - Post-Silicon Characterization/Testing

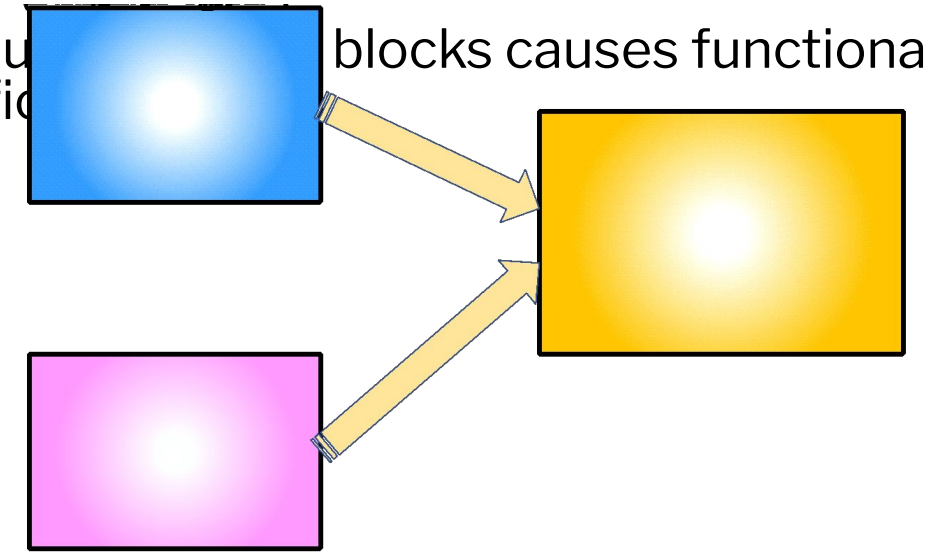


- Verification/Validation needs to be performed throughout this development process
  - Stimuli/Vectors at SoC/Chiplet-level or for each block/partition confirms design is correct
  - This is to find the bugs/flaws in each of the above phases

- Verification/Validation at each of the Phases:
  - High-level models are built for IPs, SoCs and the entire platform and verification run
  - In Pre-silicon phase, RTL design and verification is done all the way to tape-out
    - SoCs are then assembled with infrastructures like Fabric, PLLs, SRAMs, PHYs, IOs etc Verification done to ensure design is free of bugs
  - Post-Silicon Characterization stage reuses as much of the pre-silicon tests to ensure bug-free silicon
    - Tester-based patterns are created from the pre-silicon phase with the requisite pre-ambles

Verification/Validation Challenges

- Effectiveness of tests is paramount during Verification/Validation and Manufacturing
  - Need Functional and Structural content for high Verification/Validation coverage
  - Manufacturing relies on structural metrics but there are instances where functional content is required
  - Since gate-counts of designs are compounding exponentially Time/effort of fault-grading becoming impractical
  - Interactions across variou blocks causes functional and structural validation to become increasingly diffi

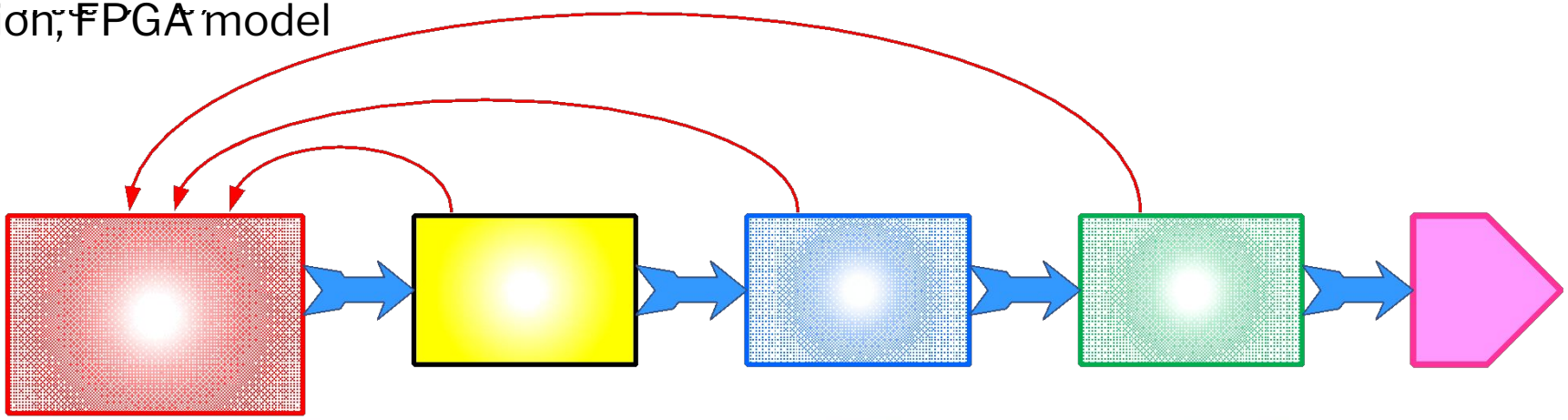


- These challenges are just the tip of the iceberg in estimating post-silicon coverage

Pre-silicon Verification and Coverage

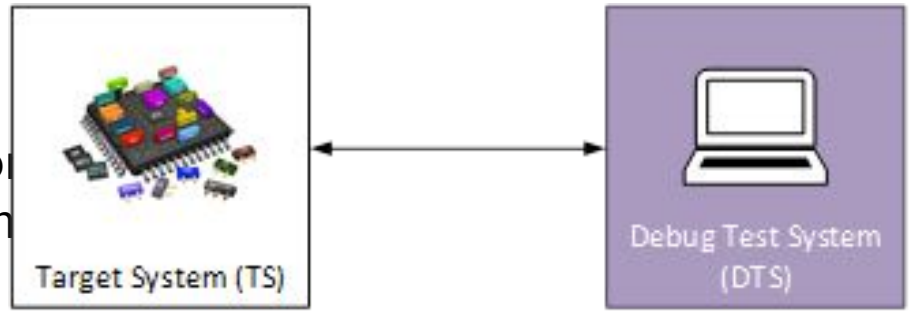
- Pre-silicon verification coverage begins by driving data stimulus and capturing results
  - Line and code level coverage are part of IP design/content of Chiplets/SoCs
  - These are assembled at SoC-level using various IPs with RTL integration tools, stimulus, scoreboards
  - Verification includes system-level checks along with IP-specific tests, such as:
    - Fabric bus protocols, clock cycle accurate delays, security, interrupt, clock gating, power domains etc.

- Stages of Pre-silicon validation
  - IP design and SoC/Chiplet Integration stage
  - RTL simulation
  - RTL Emulation/FPGA
  - System platform, Emulation, FPGA model



Design for Verification of SoCs/Chiplets

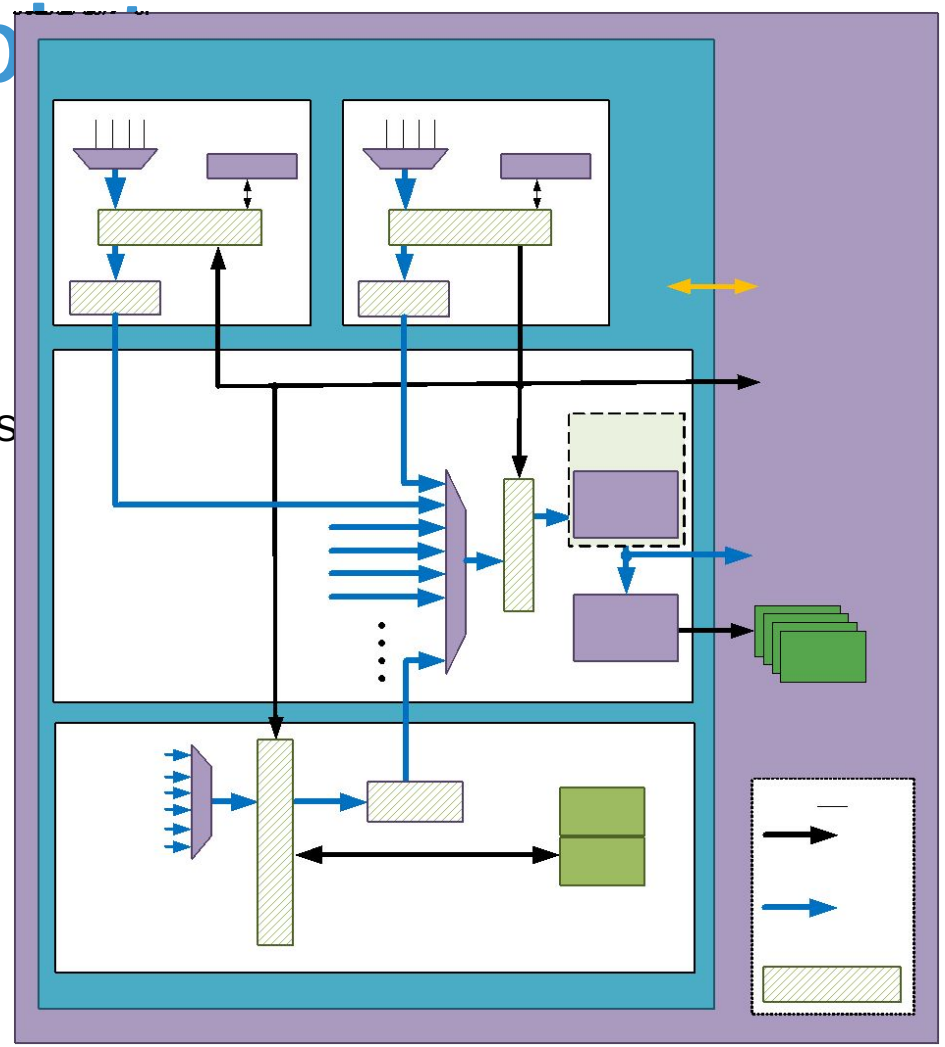
- Design for Verification needed to speed-up and improve Verification Quality and Time-To-Market (TTM)
  - Design for Verification is needed for various reasons
  - Help improve Verification:
    - Helps improve verification Quality,
    - Improves Verification Coverage,
    - Facilitate Emulation,
    - Avoid late bug discoveries etc.
  - DFV needs to be planned early in the design-cycle to bring awareness to the Architects, Designers, Implementers etc.
- DFV Infrastructure
  - Needs to be built into the design to enable Design for Verificatio
  - This needs to be taken into consideration as one of the requirem
  - Cannot be an After-Thought



- Leverage DFD infrastructure for DFV
  - Figure shows Debug Test System (DTS) connected to a Target System (TS) for Debug
  - DTS sends commands to enable Debug of the Target SoC or Target Platform/System

DFV Infrastructure of SoC/Chip

- DFV Infrastructure needs to be planned early-on
  - Need nodes to be identified for DFV
- Reuse of nodes identified for DFD
  - Design for Debug already has nodes identified for debug purpos
  - These can be reused for DFV purposes
  - Additional nodes for DFV:
    - Nodes identified for DFT, e.g., Scan Test Points, Array Observability nodes
    - IO Logic nodes identified for Test and Debug
    - PLL Observability nodes
    - Etc.,
- DFD Infrastructure reused for DFV
  - Figure shows the observability points identified for Debug
  - These Debug observability points can be reused for DFV
  - Provides dual-use purpose for DFD and DFV
  - Similarly, other DFT, DFM, DFY nodes can be reused for DFV purposes
  - Enables effective Verification early on for successful execution of Silicon/Tape-outs with High Quality, Faster TTM



Conclusions and Future Work

- We saw three phases of SoC/Chiplet Development
  - Architecture/Specification
  - Design/Implementation
  - Post-Silicon Characterization/Testing

- Verification/Validation needs to be done at every stage
  - To ensure high verification coverage, improved quality
  - Also, to attain Time-To-Market

- Reuse of nodes identified for DFD and DFT for DFV helps achieve goals much faster
  - Identifying nodes for DFV alone would be a hard-sell amongst the architects/implementers
  - Can be supplemented with small number of Must-Have DFV nodes if identified early-on with design team

- Planning DFV early-on in the design cycle
  - Helps get buy-in from project
  - Provides sufficient time to plan and add the required nodes for DFV
  - Enables adding DFD/DFT nodes and to identify additional nodes required for DFV

Last Slide

Dr. Sankaran Menon