

IEEE P1500 Mergeable Core Test Task Force Update

May 3-4, 2001

Sudipta Bhawmik, Bret Stewart, Pradipta Ghosh,
Kamran Zarrineh, Magdy Abadir, Michael Wahl,
Saman Adham, Sandeep Koranne, Scott Davidson,
Peter Harrod, Rick Fissette

Copyright 2001 IEEE

Our Mission

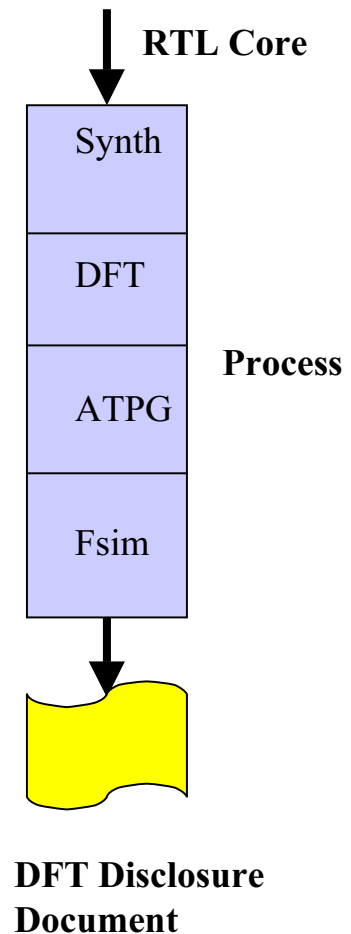
- To find solutions that will enable easy test interoperability of Mergeable (soft/firm) cores
- To find what needs to be done inside the core to make easy test integration

What will a Merged Core Test Standard Look Like?

- A synthesizable RTL core does not have any DFT incorporated, hence, cannot have a test interface specified
- The key objective of the standard is to allow for easy test interoperability of the IP when integrated with other cores and UDL

What will a Merged Core Test Standard Look Like?

- Merged core test standard will define a specific DFT **process** which a compliant core must undergo
- The outcome of the DFT process should be made available as the **“DFT Disclosure Document”**



Enforcement is the key issue.

DFT Process

- DFT Process to include
 - RTL Test interoperability rules
 - Synthesis
 - DFT insertion
 - ATPG
 - Fault Simulation
 - Iddq Test

IEEE 1500 DFT Disclosure Document

Core/Virtual Component Id:

Delivery Format :

Synthesis Tool:

Target Library:

List any constraints used:

DFT methodology used					
DFT Tools					
Scan					
BIST					
ATPG					
FSIM					
Iddq					

List any DFT rule violations (errors/warnings) reported by the tools:

Fault Coverage (mention fault model used)

Tool	Upper Bound	Lower Bound

Test Vectors:

Size:

Are the vectors validated? Yes No

Does the core have embedded memories: Yes No

(If Yes, fill in Memory Disclosure Doc)

List any special DFT considerations to be made to test the core:

Clocking Scheme	
<ul style="list-style-type: none"> • Number of clock domains: • Max clock speed the core has been tested for: • Any clock gated/muxed internally? • Any clock generated internally? • Any special test clock? 	
Flip-flops/Latches	
<ul style="list-style-type: none"> • Design consists of <ul style="list-style-type: none"> • Positive edge trig flip-flops <input type="checkbox"/> Yes <input type="checkbox"/> No • Negative edge trig flip-flops <input type="checkbox"/> Yes <input type="checkbox"/> No • Positive level sense latches <input type="checkbox"/> Yes <input type="checkbox"/> No • Negative level sense latches <input type="checkbox"/> Yes <input type="checkbox"/> No 	Number of Elements
Preset/Clear Signal	
<ul style="list-style-type: none"> • Design uses <ul style="list-style-type: none"> • Asynchronous Preset/Clear <input type="checkbox"/> Yes <input type="checkbox"/> No • Synchronous Preset/Clear <input type="checkbox"/> Yes <input type="checkbox"/> No 	

- Asynchronous reset controllable from core inputs? Yes No
- Asynchronous reset is gated and/or muxed ? Yes No

Internal Tri-State Bus

- Design consists of internal tri-state bus/drivers? Yes No
- No bus contention (more than one driver driving the bus at the same instant) occurs during
 - Deterministic test
 - Pseudo Random test

Inputs/Outputs

- Core inputs registered
 - None
 - All
 - Some (List the inputs)
- Core outputs registered
 - None
 - All
 - Some (list the outputs)

- Core includes wrapper? Yes No

If Yes

- Is the wrapper IEEE 1500 compliant? Yes No
- Can the wrapper be removed ? Yes No

If yes, how?

IDDQ Testing

- Is the design Iddq Testable? Yes No

If Yes :

- How many IDDQ stop vectors are there?
- What is the Iddq pass/fail threshold?
- Is there a special test configuration or mode for IDDQ? Yes No

If Yes, provide details

- Do any inputs have to be driven to specific values for IDDQ testing? Yes No

If Yes, list the inputs and their values.

Issues to be resolved

- Should we mandate a DFT process ?
 - What will be the process steps?
 - How do we enforce?
- Should DFT Disclosure be a separate document (may be XML) or be included as part of CTL?
[Michael Wahl working on the DTD development for XML version]
- Should mergeable core test standard (DFT process, DFT Disclosure) be part of IEEE 1500.0 or be a separate standard like IEEE 1500.n?