Figure 5-1 Introduction to Embedded Core Test and Test Integration
### WHAT IS A CORE?

**SOFT**

- **HDL** Model with No Test
- **HDL** Model with Modeled Test
- **RTL** Model with No Test
- **RTL** Model with Modeled Test

**FIRM**

- **Gate-Level Netlist with No Test**
- **Gate-Level Netlist with Synthesized Test**
- **Gate-Level Netlist with Inserted Test**
- **Gate-Level Netlist with Mixed Test**

**HARD**

- **Layout GDSII with No Test**
- **Layout with Test from Synthesis**
- **Layout with Test from Gate-Level**
- **Layout with Test Optimization**

*Figure 5-2 What is a CORE?*
Figure 5-3 Chip Designed with Core

- A Core-Based Device May Include -

1. Core(s) with Test Wrapper + Embedded Memory Arrays
2. Chip-Level User Defined Logic + Embedded Memory Arrays
3. Chip-Level Test Selection and Control Logic
4. Dedicated Chip-Level Test Pins
5. Chip-Level Clock Generation and Clock Control Logic
6. IEEE 1149.1 Controller and Boundary Scan Logic
Figure 5-4 Reuse Core Deliverables

Business Deliverables

1. The Core
2. The Specification or Data Sheet
3. The Various Models
4. The Integration Guide
5. The Reuse Vectors
CORE-BASED DESIGN DFT ISSUES

A KNOWN STIMULUS

A KNOWN EXPECTED RESPONSE

ACCESS TO THE EMBEDDED CORE

Other Chip-Level Logic

Chip-Level Device

• If the Core is HARD — DFT must exist before delivery — how is access provided at the chip level?

• If the Core is HARD — and delivered with pre-generated vectors — how are vectors merged in the whole test program?

• If the Core is HARD — and part of the overall chip test environment — how is the core test scheduled?

• If the Core is HARD — and part of the overall chip test environment — what defaults are applied when not active?

• If the Core is HARD — what is the most economical and effective test mix — Scan? LBIST? MBIST? Functional?

• If the Core is SOFT — is the overall chip test environment developed as a Core and UDL or as a unified design?

• If the Core operates at a different frequency from the pin I/O or other chip logic — how does this affect DFT and Test?

Figure 5-5 Core DFT Issues
• DFT Drivers During Core Development
  Target Market/business — Turnkey versus Customer Design
  Target Cost-Performance Profile — Low to High
  Potential Packages — Plastic versus Ceramic
  Potential Pin Counts

• Core Test Architectures and Interfaces
  Direct Access — Mux Out Core Terminals
  Add-On Test Wrapper — Virtual Test Socket
  Interface Share-Wrapper — Scanned Registered Core I/O
  At-Speed Scan Or Logic Built-in Self-test (LBIST)

• Design For Reuse Considerations
  Dedicated Core Test Ports — Access Via IC Pins
  Reference Clocks — Test and Functional
  Test Wrapper — Signal Reduction/No JTAG/No Bidi’s
  Virtual Test Socket — Vector Reuse

Figure 5-6 Core Development DFT Considerations
Figure 5-7 DFT Core Interface Considerations

- Core DFT Interface Considerations
  Note — none of this is known a priori

  Access to core test ports via IC pins (integration)
  I/O port count less restrictive than IC pin count
  Impact of routing core signals to the chip edge
    - Dedicated test signals to place in test mode
    - Number of test signals needed to test core
    - Frequency requirements of test signals
At the time of Core Development, the UDL logic is not available and its configuration is not known.

For example:
- registered inputs or outputs
- combinational logic
- bidirectional signals or tristate busses

How are vectors generated for a Hard Core before integration?

How are vectors delivered that can assess the signal timing or frequency?

How is test access planned to be provided — through the UDL or directly from the package pins?
• Core DFT Interface Considerations

Wrapper for interface signal reduction
Wrapper for frequency assessment
Wrapper as frequency boundary
Wrapper as a virtual test socket (for ATPG)

Note: bidirectional functional signals can’t cross the boundary if wrapper or scan

Figure 5-9 DFT Core Interface Considerations
Figure 5-10 Registered Isolation Test Wrapper

where the wrapper is the registered core functional I/F that is scan-inserted separately

Note: Wrapper and core are on same clock and path delay is used to generate vectors
Figure 5-11 Slice Isolation Test Wrapper

where the wrapper is an added “slice” between the core functional I/F and the UDL functional I/F

Wrapper and core are on different clocks and path delay is used to generate vectors
"Land between the Lakes"
The Isolation Test Wrapper

DQ

UDL Scan Domain
UDL Logic

Core_Test
TR_SDO

Embedded Hard Core

DQ

QD

UDL Scan Domain

TR_SD
TR_MODE
TR_CLK
TR_SE

Wrapper Scan Domain

System Clock

the wrapper is an added “slice” between the core functional I/F and the UDL functional I/F

Wrapper and core are on different clocks and path delay is used to generate vectors

Figure 5-12 Slice Isolation Test Wrapper Cell
Figure 5-13 Core DFT Connections through the Test Wrapper

All Core Test Interface Signals pass through the Test Wrapper without being acted upon.
All Core I/O are part of the Wrapper Scan Chain.

So Total Core Test I/F is:
- Internal Scan
- Internal MBIST
- Wrapper Scan

“Land between the Lakes”
The Isolation Test Wrapper
Figure 5-14 Core DFT Connections with Test Mode Gating

All Core Test Interface Signals pass through the Test Wraper and may be acted upon by a Test Mode
All Core I/O are part of the Wrapper Scan Chain
So Total Core Test I/F is:
Gated Internal Scan
Gated Internal MBIST
Gated Wraper Scan
• DFT Considerations
  Can’t Support Bidirectional Core Ports
  Input and Reference Clocks

Figure 5-15 Other Core Interface Signal Concerns
• Core DFT Frequency Considerations

Wrapper for frequency boundary
Test signals designed for low frequency
Package interface designed for high frequency
Wrapper as a multi-frequency ATPG test socket

Note: functional high/low frequency signals can cross the wrapper—the test I/F is the concern

**Figure 5-16** DFT Core Interface Frequency Considerations
• Core DFT Goals and Features

Embedded Memory Test by MBIST
- Few Signals — High Coverage — Less Test Time
- Bitmap Characterization Support

Structure by Stuck-At Scan
- High Coverage — Fewer Vectors — Ease of Application

Frequency by At-Speed Scan (Path & Transition Delay)
- Deterministic — Fewer Vectors — Ease of Application

Reuse of Core Patterns Independent of Integration

Test Insulation from Customer Logic

Embedded Core I/O Timing Specifications with Wrapper

Minimize Test Logic Area Impact

Minimize Test Logic Performance Penalty

DFT Scannability Logic
- Full-Scan Single-Edge Triggered MUX DFF
- Tristate Busses - Contention/Float Prevention
- Negedge Inputs and Outputs

Iddq—No Active Logic and Clock Stop Support

Figure 5-17 A Reuse Embedded Core’s DFT Features
• Core Economic Considerations

Test Integration (Time-to-Market)

Core Area and Routing Impact (Silicon/Package Cost)

Core Power and Frequency Impact (Package/Pin Cost)

Core Test Program Time/Size/Complexity (Tester Cost)

---

**Figure 5-18** Core Test Economics
Chapter 5  Embedded Core Test Fundamentals

Design-for-Test for Digital IC’s and Embedded Core Systems
Alfred L. Crouch
© 1999 Prentice Hall, All Rights Reserved

Figure 5-19 Chip with Core Test Architecture
Figure 5-20 Isolated Scan-Based Core-Testing
Figure 5-21 Scan Testing the Non-Core Logic
Figure 5-22 Scan Testing the Non-Core Logic

I/O specification testing—bus_SE
Tristate busses - contention/float prevention
Iddq—HighZ pin
Pin requirements—(open drains)
Figure 5-23 Memory Testing the Device
• Chip-level DFT integration considerations each core/vector set must have:

1. Power Rating during Test
2. Frequency/Data Rate of Test Vectors
3. Fault Coverage of the Test Vectors
4. Required Test Architecture to Reuse Vectors
5. ATPG Test Wrapper or Encrypted Sim Model
6. The Vector Set’s Format
7. The Vector Set Sizing

Figure 5-24 DFT Integration Architecture
Figure 5-25 Test Program Components
• **Receiving Core DFT Specification**

• **Driven by Fab and Integration Requirements**

• **Core DFT Specification Items**
  - Test Mix
  - Style of Test
  - Maximum Number of Integration Signals
  - Minimum-Maximum Test Frequency
  - Maximum Vector Sizing
  - Minimum Fault Coverage
  - Clock Source

*Figure 5-26 Selecting or Receiving a Core*
• Core Test Driven by Cost-of-Test and TTM

• Two Concerns: Reuse and Integration

• Reuse: Interface, Clocks, Test Features
  - number of dedicated test signal
  - size of test integration interface
  - ability to test interface timing
  - no functional bidirectional ports
  - specifications and vectors based on clock-in
  - specifications and vectors based on clock-out
  - ability to stop clock for retention or Idq
  - number of clock domains
  - at-speed full scan
  - at-speed memory BIST
  - use of a scan test wrapper
  - self-defaulting safety logic

• Integration: Core Connections, Chip Test Modes
  - simple core integration
  - reuse of pre-existing vectors
  - application of test signal defaults
  - shared resources (pins and control logic)
  - shared testing (parallel scheduling)
  - chip level test controller

Figure 5-27 Embedded Core DFT Summary