DFT METHODOLOGY AND SCAN INSERTION

By
Venugopal Annnavazzala
Associate Member Tech. Staff
Mentor Graphics (India)
Typical DFT Flow

- BSDArchitect
- MBISTArchitect
- RTL Simulation
- Logic Synthesis
- Testability Analysis & Test Synthesis
- Logic BIST Insertion
- ATPG & Fault Simulation
- DFTAdvisor
- DFTInsight
- LBISTArchitect
- FastScan
- FlexTest
- Logic and Timing Verification
- ASIC Vendor Interfaces
- Boundary Scan Insertion
- Memory BIST Synthesis
- Test Vector Translation
**DFTAdvisor Features**

- **Testability analysis**
  - Identifies and verifies test features, such as primary input clocks, sets and resets, and sequential power-up initialization of test logic
  - Identifies and can fix testability problems, such as gated sets/resets, uncontrollable clocks, and feedback loops

- **Design rules checking**
  - Checks over 100 rules--the same ones as FastScan and FlexTest, so problems can be caught earlier on in the design flow
  - Employs DFTInsight for graphical analysis

- **Test synthesis of a variety of structures**
  - Full scan and optimized partial scan
  - MUX-scan, clocked-scan, and LSSD scan methodologies
  - Test points, test logic, and partition scan
DFTAdvisor User Interface

DFTAdvisor Setup

Internal Circuitry

Existing Scan

Primary Outputs

RAM

RD
WR
Dout

Done With Setup

Primary Inputs

Clocks

DFTAdvisor Setup

Current Process

Process Pane

Graphic Pane

Button Pane

Setup

DRC and Circuit Learning

DRC Violation Debugging

Test Synthesis

Session Transcripting...

Modeling/DRC Setup...

Test Synthesis Setup...

Report Environment

Invoke DFTInsight

Dofile...

Exit...

Help...
Supported Test Structures

- Full Scan
- Partial Scan
- Partition Scan
- Test Points
Scan Basics: A Review

Premise: It is more difficult to create test sets for sequential circuits than for combinational circuits.

Goal: Make a sequential circuit appear combinational for testing purposes. Add control and observe points, to make the design more testable.

Method: Replace and connect sequential elements into a shift register (scan chain) of scan elements.

Result: All scan elements in the scan chain become accessible from the periphery of the chip (which makes them fully controllable and fully observable)
Scan Architectures

• A variety of scan architectures, or methodologies, exist in the industry

• Mentor Graphics DFT/ATPG tools fully support the following methodologies:
  o Mux-DFF (also called Mux-Scan)
  o LSSD
  o Clocked-Scan

• Benefits of each type:
  o Mux-DFF and Clocked-Scan best for edge-triggered flip-flops
  o Clocked-Scan insures data hold for non-scan cells during scan loading
  o LSSD is effective on latch-based designs
Mux-DFF Architecture

- Most common methodology
- Uses scan enable signal to multiplex between scan and circuit data input
- Gates data during shifting
- Also referred to as “Mux-Scan”

Example Mux-DFF scan cell:
LSSD Architecture

- Used at IBM since 1960-70s
- Uses polarity hold Shift Register Latches (SRLs)
- Non-overlapping clocks

Example LSSD scan cell:
Clocked-Scan Architecture

- Similar to MUX-Scan
- However, uses a dedicated test clock to shift in scan data

Example Clocked-Scan scan cell:
Basic DFTAdvisor Process Flow

Setup Mode

1. Set Up Circuit and Scan Info
2. Run Design Rules and Testability Analysis

DFT Mode

3. Existing Scan or Boundary Scan?
   - Yes: Rip Up/Connect to/Ignore Existing Scan Circuitry
   - No: Set Scan Chain Parameters

4. Perform Scan or Test Point Identification
5. Synthesize Desired Test Circuitry
6. ATPG Setup Files

ATPG Library
Synthesized Netlist
From Synthesis
DFTAdvisor Basic Full-Scan and Partial-Scan Flows

Setup Mode

- Set Up Circuit and Scan Info
- Run Design Rules and Testability Analysis

Partial-Scan

Scan Style?

Full-Scan

- Run Scan Identification

DFT Mode

- Synthesize Desired Scan Circuitry
- Set Scan Chain Parameters

Synthesized Netlist (with Scan)

- Setup Scan Identification
- Add Nonscan Instances and/or models

Write ATPG Setup Files

ATPG Setup Files
DFTAdvisor Fault Directed Partial-Scan Flow

**FlexTest Flow**
- Set Up Circuit Information
- Set System Mode FAULT
- Set Pattern Source to External
- Add Faults
- Run Fault Simulation
- Write UC and UD Faults to File
- Undetected Fault List

**DFTAdvisor Flow**
- Set Up Circuit and Scan Information
- Run Design Rules and Testability Analysis
- Setup Scan Id for Partial Scan Based on External Faults
- Identify Scannable Instances
- Set Scan Chain Parameters
- Synthesize Desired Scan Circuitry . . .

External Pattern Set
DFTAdvisor Inputs and Outputs

- Design (netlist)
- ATPG Library
- Scan Setup (or Dofile)
- ATPG Setup Files
- Test Procedure File
- Scan-Inserted Netlist
DFTAdvisor Inputs

DFTAdvisor utilizes the following inputs:

- Synthesized design (netlist)
  - EDIF
  - TDL
  - Verilog
  - VHDL

- ATPG Library (same as FastScan and FlexTest)

- Scan setup information
  DFTAdvisor commands entered either interactively or in batch mode (dofile)

- Test procedure file (sometimes necessary)
DFTAdvisor Outputs

DFTAdvisor produces the following outputs:

- **Scan-inserted netlist**
  - EDIF
  - TDL
  - Verilog
  - VHDL
  - NDL

- **ATPG setup files**
  - Dofile of scan setup commands that can be used in FastScan or FlexTest
  - Test procedure file for FastScan or FlexTest
DFTAdvisor System Modes

DFTAdvisor has two system modes:

- **SETUP**
  - Sets up design information
  - Sets up scan circuitry

- **DFT**
  - Runs scan identification
  - Performs test synthesis (scan insertion) of the specified circuitry
What is Test Logic?

Background:

- Sequential devices must be controllable to be converted to scan
- Other circuitry, such as RAM and tri-state devices must be controllable to be testable

Problem:

- Some designs contain uncontrollable (or internally-generated) clock, set, reset, tri-state enable, or RAM control signal
- Uncontrollable signals cause uncontrollable behavior of sequential devices and other parts of the circuit--reducing design testability

Solution:

- Add circuitry that makes clocks controllable from the primary inputs
- This circuitry is called Test Logic.
Examples of Test Logic

Before
Uncontrollable Clock

Added Test Logic

Before
Uncontrollable Signals

Added Test Logic

Before
Uncontrollable Clock

After
Uncontrollable Signals

Added Test Logic
Invoking DFTAdvisor

- Executable is at $MGC_HOME/bin/dftadvisor

- Invocation usage:

  $MGC_HOME/bin/dftadvisor {design_name
  {-Edif | -TDl | -VHdl | -VERIlog | -Genie | -SPice} {-LIBrary filename} [-SEnsitive]
  [-LOg filename] [-Replace] [-TOp module_name]
  [-Dofile dofile_name] [-LICense retry_limit]
  [-NOGui]} | -Help | -VERSion

- Example:

  shell> $MGC_HOME/bin/dftadvisor \n  design1.edif -edif -lib /net/jdoe/usr1/work \n  /mitsulib10 -d design1_coms.do -nog -logfile \n  my_session.log -replace

- When finished invoking, the “SETUP>” prompt is displayed
Setting Up the Circuit for Scan

Setting up the circuit for scan includes several tasks, such as:

- Selecting the scan methodology
- Specifying test logic circuitry
- Specifying circuit clocks
- Specifying existing scan (if necessary)
Scan Setup: Selecting the Scan Methodology

- DFTAdvisor supports insertion of the following methodologies:
  - MUX-scan (default)
  - Clocked-scan
  - LSSD

- To specify the scan methodology, use the Set Scan Type command
Scan Setup: Specifying Test Logic Circuitry

- Test logic can be added to the design to:
  - Gate uncontrollable set, reset, and clock signals
  - Disable tristate enable lines during scan loading and unloading
  - Control RAM clocks
  - Multiplex scan output with functional output
  - Add test points to the design

- Components that can be used as test logic exist in the design library (marked with “cell_type” attribute in model definition) or can be specified with Add Cell Model command

- Specify addition of test logic with the Set Test Logic command
Scan Setup: Specifying Circuit Clocks

- For DFTAdvisor, “clocks” are any signals that can alter the state of a sequential device.
- Clocks include system clocks, sets, and resets.
- Clocks must be specified with their “off-state” (inactive value).
- If you do not specify a clock signal, the sequential devices controlled by that clock will not be considered for scan.
- You specify clocks with the Add Clocks command.
- You may need to constrain other pins to make the clocks work, using the Add Pin Constraints command.
Scan Setup: Specifying Existing Scan

If you have existing scan circuitry you want to use:

- You must specify the scan group and scan chain information prior to entering DFT mode
- **Specify this information with the** Add Scan Group and Add Scan Chain **commands**

If you have existing scan circuitry you do not want to use:

- Do not specify the scan information to DFTAdvisor

If you have existing scan circuitry you want to delete:

- Specify the scan groups and chains
- **Delete existing scan with the** Ripup Scan Chains command (in DFT mode)
- Your design is treated as though it never had any scan
Running Rules Checking and Testability Analysis

- Testability analysis is run during rules checking
- It performs full ATPG analysis of clock rules C1, C3, C4, and C5
- Turn testability analysis on with the Atpg_analysis option of the Set DRC Handling command
- Additional rules checks are performed on designs with existing scan circuitry
Specifying Scan Port Names

In DFT mode, you can specify additional scan chain parameters, such as:

- **Names of the scan input and output ports**
  - By default, DFTAdvisor names ports of chainX, scan_inX and scan_outX (where X is a number, such as “1”)
  - You can change the default naming using the Setup Scan Insertion command
  - You can specify existing input and output port names with the Add Scan Pins command

- **Names of the scan enable and clock ports**
  - By default, DFTAdvisor names enable and clock ports as follows:
    
    ```
    scan enable -- scan_en
    test enable -- test_en
    test clock -- test_clk
    scan clock -- scan_clk
    scan master clock -- scan_mclk
    scan slave clock -- scan_sclk
    ```
  - You can specify other enable and clock names with the Setup Scan Insertion command
DFTAdvisor has two methods for scan identification:

- **User-defined**
  
  o Gives you the ability to choose which instances or models you want (or don’t want) converted to scan instances
  
  o **To identify which instances should be considered for scan, use the** Add Scan Instances, Add Scan Models, Add Nonscan Instances, and Add Nonscan Models **commands**

- **System-defined**
  
  o **Use the** Setup Scan Identification **command (in combination with several other “setup” type commands) to let DFTAdvisor identify scannable instances**
  
  o You can combine user-defined and system-defined methods by specifying some instances to be included (or excluded) and letting DFTAdvisor choose the rest
Running and Reporting on Scan Identification

- To run scan identification (after setting up for scan identification), enter the Run command

- To report on the results of scan identification, use the Report Statistics or Report Scan Identification command

- The Report Statistics command displays the number of:
  - Sequential instances
  - User-defined non-scan and scan instances
  - System-defined non-scan and scan instances
  - Instances scannable with test logic
  - Scan instances in pre-existing chains

- Report Scan Identification command displays identified and defined scan instances along with other pertinent information
Inserting Scan Chains

- After scan identification and the proper scan chain setup, you can insert scan chains

- To insert scan chains, use the Insert Test Logic command

- The Insert Test Logic command has a number of options that let you specify:
  - Instance order in the scan chain (via a file)
  - Type of test structures to insert
  - Maximum length of the scan chain
  - Number of scan chains
  - Whether to merge instances controlled by different clock edges onto the same chain
  - Whether to merge instances controlled by different clocks onto the same chain
  - Whether to connect the scan instances into a chain
Writing the Scan-Based Netlist

- After scan is inserted, you can save the scan-based netlist with the Write Netlist command.

- The **Write Netlist** command can save the design in any of the following netlist formats:
  - EDIF (default)
  - TDL
  - Verilog
  - VHDL
  - NDL
Writing the ATPG Setup Files

• If you are going to use FastScan or FlexTest for ATPG, DFTAdvisor can automatically write ATPG setup files

• To automatically generate the ATPG setup files, use the **Write Atpg Setup command**

• The **Write Atpg Setup command creates**:
  
  o A dofile, named `<name>.dofile`, which specifies to FastScan and FlexTest the circuit and scan information
  
  o A test procedure file named `<name>.testproc`, which specifies to FastScan and FlexTest the operation of the scan circuitry in the design